Human review packet: GKGMM19IterativeLocalVoting

Generated from byte-pinned source excerpts, the expanded PaperInterface specifications, and hash-bound raw-source-to-Spec screening records.

Paper: GKGMM19IterativeLocalVoting
Paper id: GKGMM19IterativeLocalVoting
Source version: JAIR 64 (2019), 315-355; published 2019-02-18
Source record: audit/paper_statement_map.json
Rows: 11
Generated: 2026-09-06
Source URL: official source

How to use this packet

For each source claim, compare the exact source excerpts with the expanded PaperInterface specification. Mark up this PDF or fill the blank reviewer box in the TeX source; return the annotated file for reconciliation into the paper-local human-review log.

Open the interactive dashboard

The interactive dashboard is an optional alternative to using this PDF; it presents the same review surface rather than an additional review requirement. After forking and cloning (or pulling) AppliedModelingLib, run the following from the repository root. The cache command is needed only the first time in a new clone.

```
cd <your-repository-checkout>
lake exe cache get
python3 scripts/review_dashboard.py --paper GKGMM19IterativeLocalVoting --serve
```

Open http://127.0.0.1:8765/ in a browser and keep that terminal running while reviewing. The dashboard records saved annotations in this paper's local reviewer-owned review record.

Contents

Governing Lean declarations

These exact graph-bound declaration bodies are retained by the displayed specifications or reviewed prerequisites. They are supporting context and do not add source-result rows or semantic judgments.

AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → [Fintype ι] → (ι → ℝ) → (ι → ℝ) →L[ℝ] ℝ

value:
fun {ι : Type u_1} [Fintype ι] (g : ι → ℝ) => ∑ i : ι, g i • ContinuousLinearMap.proj i
```

AppliedModelingLib.FiniteDimensionalNorms.l1

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → [Fintype ι] → (ι → ℝ) → ℝ

value:
fun {ι : Type u_1} [Fintype ι] (x : ι → ℝ) => ∑ i : ι, |x i|
```

AppliedModelingLib.FiniteDimensionalNorms.l2Sq

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → [Fintype ι] → (ι → ℝ) → ℝ

value:
fun {ι : Type u_1} [Fintype ι] (x : ι → ℝ) => ∑ i : ι, x i ^ 2
```

AppliedModelingLib.FiniteDimensionalNorms.l2

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → [Fintype ι] → (ι → ℝ) → ℝ

value:
fun {ι : Type u_1} [Fintype ι] (x : ι → ℝ) => √(AppliedModelingLib.FiniteDimensionalNorms.l2Sq x)
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.l2Sq

AppliedModelingLib.FiniteDimensionalNorms.linf

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → [Fintype ι] → [Nonempty ι] → (ι → ℝ) → ℝ

value:
fun {ι : Type u_1} [Fintype ι] [Nonempty ι] (x : ι → ℝ) => Finset.univ.sup' Finset.univ_nonempty fun (i : ι) => |x i|
```

AppliedModelingLib.FiniteDimensionalNorms.lpPower

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → [Fintype ι] → ℝ → (ι → ℝ) → ℝ

value:
fun {ι : Type u_1} [Fintype ι] (p : ℝ) (x : ι → ℝ) => ∑ i : ι, |x i| ^ p
```

AppliedModelingLib.FiniteDimensionalNorms.lp

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → [Fintype ι] → ℝ → (ι → ℝ) → ℝ

value:
fun {ι : Type u_1} [Fintype ι] (p : ℝ) (x : ι → ℝ) => AppliedModelingLib.FiniteDimensionalNorms.lpPower p x ^ (1 / p)
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.lpPower

AppliedModelingLib.Optimization.EuclideanProjectionOnto

Declaration location: reusable library

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → Set (Coord → ℝ) → ((Coord → ℝ) → Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] (X : Set (Coord → ℝ)) (project : (Coord → ℝ) → Coord → ℝ) =>
  ∀ (y : Coord → ℝ),
    project y ∈ X ∧
      IsMinOn (fun (x : Coord → ℝ) => AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) => x i - y i) X
        (project y)
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.l2

AppliedModelingLib.Optimization.FiniteSubgradientOn

Declaration location: reusable library

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → ((Coord → ℝ) → ℝ) → Set (Coord → ℝ) → (Coord → ℝ) → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] (cost : (Coord → ℝ) → ℝ) (X : Set (Coord → ℝ)) (x g : Coord → ℝ) =>
  ∀ y ∈ X,
    (cost x + (AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional g) fun (i : Coord) => y i - x i) ≤
      cost y
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional

AppliedModelingLib.Optimization.OutcomeIndexedConvergesToSet

Declaration location: reusable library

Lean declaration body

```
type:
{Omega : Type u_1} →
  {Point : Type u_2} →
    [inst : MeasurableSpace Omega] →
      [TopologicalSpace Point] → MeasureTheory.Measure Omega → (ℕ → Omega → Point) → Set Point → Prop

value:
fun {Omega : Type u_1} {Point : Type u_2} [MeasurableSpace Omega] [TopologicalSpace Point]
    (mu : MeasureTheory.Measure Omega) (trajectory : ℕ → Omega → Point) (targetSet : Set Point) =>
  ∀ᵐ (omega : Omega) ∂mu,
    ∃ target ∈ targetSet, Filter.Tendsto (fun (n : ℕ) => trajectory n omega) Filter.atTop (nhds target)
```

AppliedModelingLib.Probability.HasBoundedDensity

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → [inst : MeasurableSpace α] → MeasureTheory.Measure α → MeasureTheory.Measure α → ENNReal → Prop

value:
fun {α : Type u_1} [MeasurableSpace α] (ν μ : MeasureTheory.Measure α) (C : ENNReal) =>
  ∃ (D : α → ENNReal), μ = ν.withDensity D ∧ ∀ᵐ (x : α) ∂ν, D x ≤ C
```

AppliedModelingLib.iidSequenceMeasure

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → [inst : MeasurableSpace α] → MeasureTheory.Measure α → MeasureTheory.Measure (ℕ → α)

value:
fun {α : Type u_1} [MeasurableSpace α] (ν : MeasureTheory.Measure α) =>
  MeasureTheory.Measure.infinitePi fun (x : ℕ) => ν
```

AppliedModelingLib.iidSequenceNaturalFiltration

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} →
  [inst : MetricSpace α] →
    [SecondCountableTopology α] →
      [inst_2 : MeasurableSpace α] → [BorelSpace α] → MeasureTheory.Filtration ℕ inferInstance

value:
fun {α : Type u_1} [MetricSpace α] [SecondCountableTopology α] [MeasurableSpace α] [BorelSpace α] =>
  MeasureTheory.Filtration.natural (fun (n : ℕ) (omega : ℕ → α) => omega n)
    AppliedModelingLib.iidSequenceNaturalFiltration._proof_2
```

AppliedModelingLib.iidSequencePastPrefix

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → (n : ℕ) → (ℕ → α) → Fin (n + 1) → α

value:
fun {α : Type u_1} (n : ℕ) (omega : ℕ → α) (a : Fin (n + 1)) => omega ↑a
```

AppliedModelingLib.iidSequenceSample

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → ℕ → (ℕ → α) → α

value:
fun {α : Type u_1} (n : ℕ) (omega : ℕ → α) => omega (n + 1)
```

GKGMM19IterativeLocalVoting.DecomposableStructure

Declaration location: paper-local

Lean declaration body

```
field coords:
{Voter : Type u_1} →
  {Point : Type u_2} →
    {Coord : Type u_3} → GKGMM19IterativeLocalVoting.DecomposableStructure Voter Point Coord → Finset Coord

field coordinate:
{Voter : Type u_1} →
  {Point : Type u_2} →
    {Coord : Type u_3} → GKGMM19IterativeLocalVoting.DecomposableStructure Voter Point Coord → Coord → Point → ℝ

field coordinateUtility:
{Voter : Type u_1} →
  {Point : Type u_2} →
    {Coord : Type u_3} → GKGMM19IterativeLocalVoting.DecomposableStructure Voter Point Coord → Coord → Voter → ℝ → ℝ

field coordinateUtilitiesConcave:
∀ {Voter : Type u_1} {Point : Type u_2} {Coord : Type u_3}
  (self : GKGMM19IterativeLocalVoting.DecomposableStructure Voter Point Coord) (coordinate : Coord) (voter : Voter),
  ConcaveOn ℝ Set.univ (self.coordinateUtility coordinate voter)
```

GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData

Declaration location: paper-local

Lean declaration body

```
field idealMeasure:
{Coord : Type u_1} →
  [inst : Fintype Coord] →
    GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord → MeasureTheory.Measure (Coord → ℝ)

field probability:
∀ {Coord : Type u_1} [inst : Fintype Coord]
  (self : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord),
  MeasureTheory.IsProbabilityMeasure self.idealMeasure

field densityBound:
{Coord : Type u_1} →
  [inst : Fintype Coord] → GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord → ENNReal

field densityBound_ne_top:
∀ {Coord : Type u_1} [inst : Fintype Coord]
  (self : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord), self.densityBound ≠ ⊤

field hasBoundedDensity:
∀ {Coord : Type u_1} [inst : Fintype Coord]
  (self : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord),
  AppliedModelingLib.Probability.HasBoundedDensity MeasureTheory.volume self.idealMeasure self.densityBound

field density_measurable:
∀ {Coord : Type u_1} [inst : Fintype Coord]
  (self : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord),
  ∃ (density : (Coord → ℝ) → ENNReal),
    Measurable density ∧
      self.idealMeasure = MeasureTheory.volume.withDensity density ∧
        ∀ᵐ (ideal : Coord → ℝ), density ideal ≤ self.densityBound
```

Direct retained dependencies

- AppliedModelingLib.Probability.HasBoundedDensity

GKGMM19IterativeLocalVoting.FiniteSourceSubgradientAt

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → ((Coord → ℝ) → ℝ) → (Coord → ℝ) → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] (utility : (Coord → ℝ) → ℝ) (x gradient : Coord → ℝ) =>
  ∀ (y : Coord → ℝ),
    utility y - utility x ≥
      (AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional gradient) fun (i : Coord) => y i - x i
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional

GKGMM19IterativeLocalVoting.FiniteSubgradientAt

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → ((Coord → ℝ) → ℝ) → (Coord → ℝ) → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] (cost : (Coord → ℝ) → ℝ) (x g : Coord → ℝ) =>
  ∀ (y : Coord → ℝ),
    (cost x + (AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional g) fun (i : Coord) => y i - x i) ≤
      cost y
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional

GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → ((Coord → ℝ) → ℝ) → Set (Coord → ℝ) → (Coord → ℝ) → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] (cost : (Coord → ℝ) → ℝ) (solutionSpace : Set (Coord → ℝ)) (x g : Coord → ℝ) =>
  AppliedModelingLib.Optimization.FiniteSubgradientOn cost solutionSpace x g
```

Direct retained dependencies

- AppliedModelingLib.Optimization.FiniteSubgradientOn

GKGMM19IterativeLocalVoting.HolderDualFinite

Declaration location: paper-local

Lean declaration body

```
type:
ℝ → ℝ → Prop

value:
fun (p q : ℝ) => 0 < p ∧ 0 < q ∧ 1 / p + 1 / q = 1
```

GKGMM19IterativeLocalVoting.ModelBCoordinatewiseBoundaryResponseAt

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → (Coord → ℝ) → (Coord → ℝ) → ℝ → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] (center ideal : Coord → ℝ) (r : ℝ) (response : Coord → ℝ) =>
  response = fun (i : Coord) => center i - r * ((center i - ideal i) / |center i - ideal i|)
```

GKGMM19IterativeLocalVoting.PopulationTheorem3OutcomeConvergesTo

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → {Omega : Type u_2} → [Fintype Coord] → (ℕ → Omega → Coord → ℝ) → Omega → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} {Omega : Type u_2} [Fintype Coord] (trajectory : ℕ → Omega → Coord → ℝ) (omega : Omega)
    (xstar : Coord → ℝ) =>
  ∀ (i : Coord), Filter.Tendsto (fun (n : ℕ) => trajectory n omega i) Filter.atTop (nhds (xstar i))
```

GKGMM19IterativeLocalVoting.SourceNorm

Declaration location: paper-local

Lean declaration body

type:
Type

constructor GKGMM19IterativeLocalVoting.SourceNorm.l1:
GKGMM19IterativeLocalVoting.SourceNorm

constructor GKGMM19IterativeLocalVoting.SourceNorm.l2:
GKGMM19IterativeLocalVoting.SourceNorm

constructor GKGMM19IterativeLocalVoting.SourceNorm.linfty:
GKGMM19IterativeLocalVoting.SourceNorm

constructor GKGMM19IterativeLocalVoting.SourceNorm.lp:
ℝ → GKGMM19IterativeLocalVoting.SourceNorm

GKGMM19IterativeLocalVoting.ILVUtilityFormulaData

Declaration location: paper-local

Lean declaration body

field solutionSpace:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVUtilityFormulaData Voter Point → Set Point

field utility:
{Voter : Type u_1} →
  {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVUtilityFormulaData Voter Point → Voter → Point → ℝ

field normDistance:
{Voter : Type u_1} →
  {Point : Type u_2} →
    GKGMM19IterativeLocalVoting.ILVUtilityFormulaData Voter Point →
      GKGMM19IterativeLocalVoting.SourceNorm → Point → Point → ℝ

field utility_bounded_on_solutionSpace:
∀ {Voter : Type u_1} {Point : Type u_2} (self : GKGMM19IterativeLocalVoting.ILVUtilityFormulaData Voter Point)
  (voter : Voter), ∃ (bound : ℝ), ∀ x ∈ self.solutionSpace, |self.utility voter x| ≤ bound

Direct retained dependencies

- GKGMM19IterativeLocalVoting.SourceNorm

GKGMM19IterativeLocalVoting.VoterResponseModel

Declaration location: paper-local

Lean declaration body

type:
Type

constructor GKGMM19IterativeLocalVoting.VoterResponseModel.modelA:
GKGMM19IterativeLocalVoting.VoterResponseModel

constructor GKGMM19IterativeLocalVoting.VoterResponseModel.modelB:
GKGMM19IterativeLocalVoting.VoterResponseModel

GKGMM19IterativeLocalVoting.ILVEnvironment

Declaration location: paper-local

Lean declaration body

field solutionSpace:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Set Point

field utility:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Voter → Point → ℝ

field ideal:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Voter → Point

field normDistance:
{Voter : Type u_1} →
  {Point : Type u_2} →
    GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → GKGMM19IterativeLocalVoting.SourceNorm → Point → Point → ℝ

field trajectory:
{Voter : Type u_1} →
  {Point : Type u_2} →
    GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point →

```
  GKGMM19IterativeLocalVoting.SourceNorm → GKGMM19IterativeLocalVoting.VoterResponseModel → ℕ → Point

field societalUtility:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Point → ℝ

field socialOptimal:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Set Point

field medianSet:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Set Point

field directionalField:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Point → Point

field zeroDirection:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Point

field utilityGradient:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Voter → Point → Point

field scalarDirection:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → ℝ → Point → Point

field voterExpectation:
{Voter : Type u_1} →
  {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → (Voter → Point) → Point

field convergesWithProbabilityOne:
{Voter : Type u_1} →
  {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → (ℕ → Point) → Set Point → Prop

field convergesToPoint:
{Voter : Type u_1} →
  {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → (ℕ → Point) → Point → Prop

field respondsAccordingTo:
{Voter : Type u_1} →
  {Point : Type u_2} →
    GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → GKGMM19IterativeLocalVoting.VoterResponseModel → Prop

field solutionSpace_nonempty_bounded_closed_convex:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Prop

field uniqueIdealSolutions:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Prop

field idealDistribution_bounded_measurable_density:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Prop

field directionalFieldUniformlyContinuous:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Prop
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.VoterResponseModel

GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier

Declaration location: paper-local

Lean declaration body

```
field data:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : Fintype Coord] →
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
        GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E →
          GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.ILVEnvironment

GKGMM19IterativeLocalVoting.HasUniqueIdealSolution

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Prop

value:
fun {Voter : Type u_1} {Point : Type u_2} (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point) =>
  ∀ (voter : Voter),
    E.ideal voter ∈ E.solutionSpace ∧
      (∀ x ∈ E.solutionSpace, E.utility voter x ≤ E.utility voter (E.ideal voter)) ∧
        ∀ x ∈ E.solutionSpace, E.utility voter x = E.utility voter (E.ideal voter) → x = E.ideal voter
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment

GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source

Declaration location: paper-local

Lean declaration body

```
field solutionSpace_nonempty:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
  GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project → E.solutionSpace.Nonempty

field geometry:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
  GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project →
    AppliedModelingLib.Optimization.EuclideanProjectionSourceGeometry Coord E.solutionSpace project

field project_measurable:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
  GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project → Measurable project

field ideal_is_unique_utility_maximizer:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
  GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project → GKGMM19IterativeLocalVoting.HasUniqueIdealSolution E

field ideal_mem_solutionSpace_ae:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
  GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project →
    ∀ᵐ (ideal : Coord → ℝ) ∂D.idealMeasure, ideal ∈ E.solutionSpace
```

Direct retained dependencies

- AppliedModelingLib.Optimization.EuclideanProjectionSourceGeometry
- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.HasUniqueIdealSolution
- GKGMM19IterativeLocalVoting.ILVEnvironment

GKGMM19IterativeLocalVoting.IsDirectionalEquilibrium

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Point → Prop

value:
fun {Voter : Type u_1} {Point : Type u_2} (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point)
    (xstar : Point) =>
  E.directionalField xstar = E.zeroDirection
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment

GKGMM19IterativeLocalVoting.LocalNeighborhood

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Point : Type u_2} →
    GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point →
      GKGMM19IterativeLocalVoting.SourceNorm → Point → ℝ → Point → Prop

value:
fun {Voter : Type u_1} {Point : Type u_2} (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point)
    (q : GKGMM19IterativeLocalVoting.SourceNorm) (center : Point) (r : ℝ) (a : Point) =>
  {candidate : Point | candidate ∈ E.solutionSpace ∧ E.normDistance q candidate center ≤ r} a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.SourceNorm

GKGMM19IterativeLocalVoting.ModelAResponseAt

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Point : Type u_2} →
    GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point →
      GKGMM19IterativeLocalVoting.SourceNorm → Point → ℝ → Voter → Point → Prop

value:
fun {Voter : Type u_1} {Point : Type u_2} (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point)
    (q : GKGMM19IterativeLocalVoting.SourceNorm) (center : Point) (r : ℝ) (voter : Voter) (response : Point) =>
  response ∈ GKGMM19IterativeLocalVoting.LocalNeighborhood E q center r ∧
    ∀ candidate ∈ GKGMM19IterativeLocalVoting.LocalNeighborhood E q center r,
      E.utility voter candidate ≤ E.utility voter response
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.LocalNeighborhood
- GKGMM19IterativeLocalVoting.SourceNorm

GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Omega : Type u_2} →
    {Point : Type u_3} →
      [inst : MeasurableSpace Omega] →
        [TopologicalSpace Point] →
          GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point →
            MeasureTheory.Measure Omega → (ℕ → Omega → Point) → Prop

value:
fun {Voter : Type u_1} {Omega : Type u_2} {Point : Type u_3} [MeasurableSpace Omega] [TopologicalSpace Point]
    (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point) (mu : MeasureTheory.Measure Omega)
    (trajectory : ℕ → Omega → Point) =>
  AppliedModelingLib.Optimization.OutcomeIndexedConvergesToSet mu trajectory E.socialOptimal
```

Direct retained dependencies

- AppliedModelingLib.Optimization.OutcomeIndexedConvergesToSet
- GKGMM19IterativeLocalVoting.ILVEnvironment

GKGMM19IterativeLocalVoting.RawLocalNeighborhood

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Point : Type u_2} →
    GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point →
      GKGMM19IterativeLocalVoting.SourceNorm → Point → ℝ → Point → Prop

value:
fun {Voter : Type u_1} {Point : Type u_2} (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point)
    (q : GKGMM19IterativeLocalVoting.SourceNorm) (center : Point) (r : ℝ) (a : Point) =>
  {candidate : Point | E.normDistance q candidate center ≤ r} a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.SourceNorm

GKGMM19IterativeLocalVoting.ModelARawResponseAt

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Point : Type u_2} →
    GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point →
      GKGMM19IterativeLocalVoting.SourceNorm → Point → ℝ → Voter → Point → Prop

value:
fun {Voter : Type u_1} {Point : Type u_2} (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point)
    (q : GKGMM19IterativeLocalVoting.SourceNorm) (center : Point) (r : ℝ) (voter : Voter) (response : Point) =>
  response ∈ GKGMM19IterativeLocalVoting.RawLocalNeighborhood E q center r ∧
    ∀ candidate ∈ GKGMM19IterativeLocalVoting.RawLocalNeighborhood E q center r,
      E.utility voter candidate ≤ E.utility voter response
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.RawLocalNeighborhood
- GKGMM19IterativeLocalVoting.SourceNorm

GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData

Declaration location: paper-local

Lean declaration body

```
field jointMeasure:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component →
          MeasureTheory.Measure ((Component → ℝ) × (Coord → ℝ))

field probability:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component),
  MeasureTheory.IsProbabilityMeasure self.jointMeasure

field densityBound:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component → ENNReal

field densityBound_ne_top:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component), self.densityBound ≠ ⊤

field hasBoundedDensity:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
```

```
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component),
  AppliedModelingLib.Probability.HasBoundedDensity (MeasureTheory.volume.prod MeasureTheory.volume) self.jointMeasure
    self.densityBound

field joint_density_measurable:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component),
  ∃ (density : (Component → ℝ) × (Coord → ℝ) → ENNReal),
    Measurable density ∧
      self.jointMeasure = (MeasureTheory.volume.prod MeasureTheory.volume).withDensity density ∧
        ∀ᵐ (sample : (Component → ℝ) × (Coord → ℝ)) ∂MeasureTheory.volume.prod MeasureTheory.volume,
          density sample ≤ self.densityBound

field idealDistribution:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component →
          GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord

field ideal_marginal:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component),
  MeasureTheory.Measure.map Prod.snd self.jointMeasure = self.idealDistribution.idealMeasure

field weightSet:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component → Set (Component → ℝ)

field weightSet_nonempty:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component), self.weightSet.Nonempty

field weightSet_bounded:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component),
  Bornology.IsBounded self.weightSet

field weightSet_closed:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component), IsClosed self.weightSet

field weightSet_convex:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component), Convex ℝ self.weightSet

field weightSet_nonnegative:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component),
  ∀ weight ∈ self.weightSet, ∀ (k : Component), 0 ≤ weight k

field weight_ae_mem_set:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component),
  ∀ᵐ (sample : (Component → ℝ) × (Coord → ℝ)) ∂self.jointMeasure, sample.1 ∈ self.weightSet

field weight_nonzero_ae:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component),
  ∀ᵐ (sample : (Component → ℝ) × (Coord → ℝ)) ∂self.jointMeasure, sample.1 ≠ 0

field component_nonempty:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component), Nonempty Component

field coordinate_nonempty:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component), Nonempty Coord

field componentBlock:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component → Component → Finset Coord

field componentBlocks_nonempty:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component) (k : Component),
  (self.componentBlock k).Nonempty

field componentBlocks_disjoint:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component) (k l : Component),
  k ≠ l → Disjoint (self.componentBlock k) (self.componentBlock l)

field componentBlocks_cover:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component) (i : Coord),
  ∃ (k : Component), i ∈ self.componentBlock k
```

Direct retained dependencies

- AppliedModelingLib.Probability.HasBoundedDensity
- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData

GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleIdeal

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component →
          (Component → ℝ) × (Coord → ℝ) → Component → Coord → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
    (a : (Component → ℝ) × (Coord → ℝ)) (a_1 : Component) (a_2 : Coord) =>
  a.2 a_2
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData

GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeight

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component →
          (Component → ℝ) × (Coord → ℝ) → Component → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
    (a : (Component → ℝ) × (Coord → ℝ)) (a_1 : Component) =>
  a.1 a_1
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData

GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeightNorm2

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component → (Component → ℝ) × (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
    (a : (Component → ℝ) × (Coord → ℝ)) =>
  AppliedModelingLib.FiniteDimensionalNorms.l2 (D.sampleWeight a)
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.l2
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeight

GKGMM19IterativeLocalVoting.decomposableUtilityFormula

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Point : Type u_2} →
    {Coord : Type u_3} → GKGMM19IterativeLocalVoting.DecomposableStructure Voter Point Coord → Voter → Point → ℝ

value:
fun {Voter : Type u_1} {Point : Type u_2} {Coord : Type u_3}
    (D : GKGMM19IterativeLocalVoting.DecomposableStructure Voter Point Coord) (v : Voter) (x : Point) =>
  ∑ m ∈ D.coords, D.coordinateUtility m v (D.coordinate m x)
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.DecomposableStructure

GKGMM19IterativeLocalVoting.IsDecomposableUtilitiesWith

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Point : Type u_2} →
    {Coord : Type u_3} →
      GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point →
        GKGMM19IterativeLocalVoting.DecomposableStructure Voter Point Coord → Prop

value:
fun {Voter : Type u_1} {Point : Type u_2} {Coord : Type u_3}
    (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point)
    (D : GKGMM19IterativeLocalVoting.DecomposableStructure Voter Point Coord) =>
  ∀ (v : Voter) (x : Point), E.utility v x = GKGMM19IterativeLocalVoting.decomposableUtilityFormula D v x
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.DecomposableStructure
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.decomposableUtilityFormula

GKGMM19IterativeLocalVoting.finiteCoordinateBlockL2Distance

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → Finset Coord → (Coord → ℝ) → (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (block : Finset Coord) (x ideal : Coord → ℝ) =>
  AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : ↥block) => x ↑i - ideal ↑i
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.l2

GKGMM19IterativeLocalVoting.finiteCoordinateC3CarrierFormula

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : Fintype Coord] →
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
        GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) =>
  GKGMM19IterativeLocalVoting.finiteCoordinateIdealDistributionFormulaData C3.data
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.finiteCoordinateIdealDistributionFormulaData

GKGMM19IterativeLocalVoting.finiteCoordinateNorm

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → [Nonempty Coord] → GKGMM19IterativeLocalVoting.SourceNorm → (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (p : GKGMM19IterativeLocalVoting.SourceNorm) (x : Coord → ℝ) =>
  match p with
  | GKGMM19IterativeLocalVoting.SourceNorm.l1 => AppliedModelingLib.FiniteDimensionalNorms.l1 x
  | GKGMM19IterativeLocalVoting.SourceNorm.l2 => AppliedModelingLib.FiniteDimensionalNorms.l2 x
  | GKGMM19IterativeLocalVoting.SourceNorm.linfty => AppliedModelingLib.FiniteDimensionalNorms.linf x
  | GKGMM19IterativeLocalVoting.SourceNorm.lp p => AppliedModelingLib.FiniteDimensionalNorms.lp p x
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.l1
- AppliedModelingLib.FiniteDimensionalNorms.l2
- AppliedModelingLib.FiniteDimensionalNorms.linf
- AppliedModelingLib.FiniteDimensionalNorms.lp
- GKGMM19IterativeLocalVoting.SourceNorm

GKGMM19IterativeLocalVoting.finiteCoordinateDistance

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [Fintype Coord] → [Nonempty Coord] → GKGMM19IterativeLocalVoting.SourceNorm → (Coord → ℝ) → (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (p : GKGMM19IterativeLocalVoting.SourceNorm)
    (x y : Coord → ℝ) =>
  GKGMM19IterativeLocalVoting.finiteCoordinateNorm p fun (m : Coord) => x m - y m
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateNorm

GKGMM19IterativeLocalVoting.UsesFiniteCoordinateNormDistance

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [Fintype Coord] → [Nonempty Coord] → GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ) → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord] [Nonempty Coord]
    (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)) =>
  ∀ (p : GKGMM19IterativeLocalVoting.SourceNorm) (x y : Coord → ℝ),
    E.normDistance p x y = GKGMM19IterativeLocalVoting.finiteCoordinateDistance p x y
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance

GKGMM19IterativeLocalVoting.finiteDot

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → (Coord → ℝ) → (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (x y : Coord → ℝ) => ∑ i : Coord, x i * y i
```

GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : Fintype Coord] →
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
        {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} →
          {project : (Coord → ℝ) → Coord → ℝ} → GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ}
    (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project) =>
  E.solutionSpace.Nonempty ∧
    IsClosed E.solutionSpace ∧
      Bornology.IsBounded E.solutionSpace ∧
        Convex ℝ E.solutionSpace ∧
          AppliedModelingLib.Optimization.EuclideanProjectionOnto E.solutionSpace project ∧
            Measurable project ∧
              (∀ (voter : Voter),
                  E.ideal voter ∈ E.solutionSpace ∧
                    (∀ x ∈ E.solutionSpace, E.utility voter x ≤ E.utility voter (E.ideal voter)) ∧
                      ∀ x ∈ E.solutionSpace, E.utility voter x = E.utility voter (E.ideal voter) → x = E.ideal voter) ∧
                ∀ᵐ (ideal : Coord → ℝ) ∂D.idealMeasure, ideal ∈ E.solutionSpace
```

Direct retained dependencies

- AppliedModelingLib.Optimization.EuclideanProjectionOnto
- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.ILVEnvironment

GKGMM19IterativeLocalVoting.finiteModelBExpectedLinfCost

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] →
    [Nonempty Coord] → GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord → (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord]
    (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord) (state : Coord → ℝ) =>
  ∫ (ideal : Coord → ℝ),
    AppliedModelingLib.FiniteDimensionalNorms.linf fun (j : Coord) => state j - ideal j ∂D.idealMeasure
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.linf
- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData

GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource

Declaration location: paper-local

Lean declaration body

```
field societalUtility_eq_neg_expectedLinfCost:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord},
  GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E D →
    ∀ (x : Coord → ℝ), E.societalUtility x = -GKGMM19IterativeLocalVoting.finiteModelBExpectedLinfCost D x

field mem_socialOptimal_iff_societalUtility_isMaxOn:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord},
  GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E D →
    ∀ (x : Coord → ℝ), x ∈ E.socialOptimal ↔ x ∈ E.solutionSpace ∧ IsMaxOn E.societalUtility E.solutionSpace x
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.finiteModelBExpectedLinfCost

GKGMM19IterativeLocalVoting.finiteModelBExpectedLpCost

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] → GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord → ℝ → (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord)
    (p : ℝ) (state : Coord → ℝ) =>
  ∫ (ideal : Coord → ℝ),
    AppliedModelingLib.FiniteDimensionalNorms.lp p fun (j : Coord) => state j - ideal j ∂D.idealMeasure
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.lp
- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData

GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource

Declaration location: paper-local

Lean declaration body

```
field societalUtility_eq_neg_expectedLpCost:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {p : ℝ},
  GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E D p →
    ∀ (x : Coord → ℝ), E.societalUtility x = -GKGMM19IterativeLocalVoting.finiteModelBExpectedLpCost D p x

field mem_socialOptimal_iff_societalUtility_isMaxOn:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {p : ℝ},
  GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E D p →
    ∀ (x : Coord → ℝ), x ∈ E.socialOptimal ↔ x ∈ E.solutionSpace ∧ IsMaxOn E.societalUtility E.solutionSpace x
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.finiteModelBExpectedLpCost

GKGMM19IterativeLocalVoting.finiteModelBExpectedLpMinimizerSet

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] →
    GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord → ℝ → Set (Coord → ℝ) → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord)
    (p : ℝ) (solutionSpace : Set (Coord → ℝ)) (a : Coord → ℝ) =>
  {x : Coord → ℝ |
      x ∈ solutionSpace ∧ IsMinOn (GKGMM19IterativeLocalVoting.finiteModelBExpectedLpCost D p) solutionSpace x}
    a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.finiteModelBExpectedLpCost

GKGMM19IterativeLocalVoting.finiteModelBIdealNaturalFiltration

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → MeasureTheory.Filtration ℕ inferInstance

value:
fun {Coord : Type u_1} [Fintype Coord] => AppliedModelingLib.iidSequenceNaturalFiltration
```

Direct retained dependencies

- AppliedModelingLib.iidSequenceNaturalFiltration

GKGMM19IterativeLocalVoting.finiteModelBIdealSample

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → ℕ → (ℕ → Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (n : ℕ) (omega : ℕ → Coord → ℝ) (a : Coord) =>
  AppliedModelingLib.iidSequenceSample n omega a
```

Direct retained dependencies

- AppliedModelingLib.iidSequenceSample

GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] →
    GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord → MeasureTheory.Measure (ℕ → Coord → ℝ)

value:
fun {Coord : Type u_1} [Fintype Coord] (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord) =>
  AppliedModelingLib.iidSequenceMeasure D.idealMeasure
```

Direct retained dependencies

- AppliedModelingLib.iidSequenceMeasure
- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData

GKGMM19IterativeLocalVoting.finiteModelBLinfSocialObjectiveSourceFormula

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Nonempty Coord] →
        {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
          {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} →
            GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E D → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord] [Nonempty Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord}
    (T : GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E D) =>
  (∀ (x : Coord → ℝ), E.societalUtility x = -GKGMM19IterativeLocalVoting.finiteModelBExpectedLinfCost D x) ∧
    ∀ (x : Coord → ℝ), x ∈ E.socialOptimal ↔ x ∈ E.solutionSpace ∧ IsMaxOn E.societalUtility E.solutionSpace x
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.finiteModelBExpectedLinfCost

GKGMM19IterativeLocalVoting.finiteModelBLinfSourceInputsFormula

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Nonempty Coord] →
        {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
          (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) →
            {project : (Coord → ℝ) → Coord → ℝ} →
              GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project →
                GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E C3.data → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord] [Nonempty Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
    (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project)
    (T : GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E C3.data) =>
  GKGMM19IterativeLocalVoting.finiteCoordinateC3CarrierFormula C3 ∧
    GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula S ∧
      GKGMM19IterativeLocalVoting.finiteModelBLinfSocialObjectiveSourceFormula T
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.finiteCoordinateC3CarrierFormula
- GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula
- GKGMM19IterativeLocalVoting.finiteModelBLinfSocialObjectiveSourceFormula

GKGMM19IterativeLocalVoting.finiteModelBLpSocialObjectiveSourceFormula

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : Fintype Coord] →
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
        {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} →
          {p : ℝ} → GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E D p → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {p : ℝ}
    (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E D p) =>
  (∀ (x : Coord → ℝ), E.societalUtility x = -GKGMM19IterativeLocalVoting.finiteModelBExpectedLpCost D p x) ∧
    ∀ (x : Coord → ℝ), x ∈ E.socialOptimal ↔ x ∈ E.solutionSpace ∧ IsMaxOn E.societalUtility E.solutionSpace x
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.finiteModelBExpectedLpCost

GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : Fintype Coord] →
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
        (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) →
          {project : (Coord → ℝ) → Coord → ℝ} →
            GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project →
              {p : ℝ} → GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data p → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
    (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {p : ℝ}
    (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data p) =>
  GKGMM19IterativeLocalVoting.finiteCoordinateC3CarrierFormula C3 ∧
    GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula S ∧
      GKGMM19IterativeLocalVoting.finiteModelBLpSocialObjectiveSourceFormula T
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.finiteCoordinateC3CarrierFormula
- GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula
- GKGMM19IterativeLocalVoting.finiteModelBLpSocialObjectiveSourceFormula

GKGMM19IterativeLocalVoting.finiteScalarDirection

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → ℝ → (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} (a : ℝ) (x : Coord → ℝ) (a_1 : Coord) => a * x a_1
```

GKGMM19IterativeLocalVoting.ilvRadius

Declaration location: paper-local

Lean declaration body

```
type:
ℝ → ℕ → ℝ

value:
fun (r0 : ℝ) (t : ℕ) => r0 / ↑t
```

GKGMM19IterativeLocalVoting.l1LinfBallResponse

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → (Coord → ℝ) → (Coord → ℝ) → ℝ → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (center ideal : Coord → ℝ) (r : ℝ) (a : Coord) =>
  max (center a - r) (min (center a + r) (ideal a))
```

GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeRawResponse

Declaration location: paper-local

Lean declaration body

```
type:
{Omega : Type u_1} →
  {Coord : Type u_2} → [Fintype Coord] → ℝ → (ℕ → Omega → Coord → ℝ) → (ℕ → Omega → Coord → ℝ) → ℕ → Omega → Coord → ℝ

value:
fun {Omega : Type u_1} {Coord : Type u_2} [Fintype Coord] (r0 : ℝ) (trajectory idealSample : ℕ → Omega → Coord → ℝ)
    (n : ℕ) (omega : Omega) (a : Coord) =>
  GKGMM19IterativeLocalVoting.l1LinfBallResponse (trajectory n omega) (idealSample n omega)
    (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)) a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.l1LinfBallResponse

GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeStep

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → ℝ → ((Coord → ℝ) → Coord → ℝ) → ℕ → (Coord → ℝ) × (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (r0 : ℝ) (project : (Coord → ℝ) → Coord → ℝ) (n : ℕ)
    (stateIdeal : (Coord → ℝ) × (Coord → ℝ)) (a : Coord) =>
  project
    (GKGMM19IterativeLocalVoting.l1LinfBallResponse stateIdeal.1 stateIdeal.2
      (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)))
    a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.l1LinfBallResponse

GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] →
    GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord →
      ℝ → ((Coord → ℝ) → Coord → ℝ) → (Coord → ℝ) → ℕ → (ℕ → Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord)
    (r0 : ℝ) (project : (Coord → ℝ) → Coord → ℝ) (initial : Coord → ℝ) (a : ℕ) (a_1 : ℕ → Coord → ℝ) (a_2 : Coord) =>
  Nat.brecOn (motive := fun (x : ℕ) => (ℕ → Coord → ℝ) → Coord → ℝ) a
    (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory._f D r0 project initial) a_1 a_2
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeStep
- GKGMM19IterativeLocalVoting.finiteModelBIdealSample

GKGMM19IterativeLocalVoting.l1LinfBallDirection

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → (Coord → ℝ) → (Coord → ℝ) → ℝ → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (center ideal : Coord → ℝ) (r : ℝ) (a : Coord) =>
  (center a - GKGMM19IterativeLocalVoting.l1LinfBallResponse center ideal r a) / r
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.l1LinfBallResponse

GKGMM19IterativeLocalVoting.l2BlockDistanceNormalizedGradient

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → Finset Coord → (Coord → ℝ) → (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (block : Finset Coord) (x ideal : Coord → ℝ) (a : Coord) =>
  if a ∈ block then (x a - ideal a) / GKGMM19IterativeLocalVoting.finiteCoordinateBlockL2Distance block x ideal else 0
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.finiteCoordinateBlockL2Distance

GKGMM19IterativeLocalVoting.l2DistanceNormalizedGradient

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → (Coord → ℝ) → (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (x ideal : Coord → ℝ) (a : Coord) =>
  (x a - ideal a) / AppliedModelingLib.FiniteDimensionalNorms.l2 fun (j : Coord) => x j - ideal j
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.l2

GKGMM19IterativeLocalVoting.l2BallResponse

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → [Nonempty Coord] → (Coord → ℝ) → (Coord → ℝ) → ℝ → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (center ideal : Coord → ℝ) (r : ℝ) (a : Coord) =>
  (if
      GKGMM19IterativeLocalVoting.finiteCoordinateDistance GKGMM19IterativeLocalVoting.SourceNorm.l2 center ideal ≤
        r then
      ideal
    else fun (i : Coord) => center i - r * GKGMM19IterativeLocalVoting.l2DistanceNormalizedGradient center ideal i)
    a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance
- GKGMM19IterativeLocalVoting.l2DistanceNormalizedGradient

GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeStep

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [Fintype Coord] → [Nonempty Coord] → ℝ → ((Coord → ℝ) → Coord → ℝ) → ℕ → (Coord → ℝ) × (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (r0 : ℝ) (project : (Coord → ℝ) → Coord → ℝ) (n : ℕ)
    (stateIdeal : (Coord → ℝ) × (Coord → ℝ)) (a : Coord) =>
  project
    (GKGMM19IterativeLocalVoting.l2BallResponse stateIdeal.1 stateIdeal.2
      (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)))
    a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.l2BallResponse

GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] →
    [Nonempty Coord] →
      GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord →
        ℝ → ((Coord → ℝ) → Coord → ℝ) → (Coord → ℝ) → ℕ → (ℕ → Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord]
    (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord) (r0 : ℝ)
    (project : (Coord → ℝ) → Coord → ℝ) (initial : Coord → ℝ) (a : ℕ) (a_1 : ℕ → Coord → ℝ) (a_2 : Coord) =>
  Nat.brecOn (motive := fun (x : ℕ) => (ℕ → Coord → ℝ) → Coord → ℝ) a
    (GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory._f D r0 project initial) a_1 a_2
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeStep
- GKGMM19IterativeLocalVoting.finiteModelBIdealSample

GKGMM19IterativeLocalVoting.l2BallDirection

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → [Nonempty Coord] → (Coord → ℝ) → (Coord → ℝ) → ℝ → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (center ideal : Coord → ℝ) (r : ℝ) (a : Coord) =>
  (center a - GKGMM19IterativeLocalVoting.l2BallResponse center ideal r a) / r
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.l2BallResponse

GKGMM19IterativeLocalVoting.linfL1ActiveCoordinates

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → [Nonempty Coord] → (Coord → ℝ) → (Coord → ℝ) → Finset Coord

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (center ideal : Coord → ℝ) =>
  {i : Coord |
    |center i - ideal i| = AppliedModelingLib.FiniteDimensionalNorms.linf fun (j : Coord) => center j - ideal j}
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.linf

GKGMM19IterativeLocalVoting.linfL1RawDirection

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → (Coord → ℝ) → (Coord → ℝ) → ℝ → Coord → ℝ

value:
fun {Coord : Type u_1} (center raw : Coord → ℝ) (r : ℝ) (a : Coord) => (center a - raw a) / r
```

GKGMM19IterativeLocalVoting.linfL1SymmetricActiveDirection

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → [Nonempty Coord] → [DecidableEq Coord] → (Coord → ℝ) → (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] [DecidableEq Coord] (center ideal : Coord → ℝ) (a : Coord) =>
  (have active : Finset Coord := GKGMM19IterativeLocalVoting.linfL1ActiveCoordinates center ideal;
    if (AppliedModelingLib.FiniteDimensionalNorms.linf fun (i : Coord) => center i - ideal i) = 0 then
      fun (x : Coord) => 0
    else fun (i : Coord) => if i ∈ active then (center i - ideal i) / |center i - ideal i| / ↑active.card else 0)
    a
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.linf
- GKGMM19IterativeLocalVoting.linfL1ActiveCoordinates

GKGMM19IterativeLocalVoting.finiteModelBLinfOutcomeStep

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [Fintype Coord] →
    [Nonempty Coord] → [DecidableEq Coord] → ℝ → ((Coord → ℝ) → Coord → ℝ) → ℕ → (Coord → ℝ) × (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] [DecidableEq Coord] (r0 : ℝ) (project : (Coord → ℝ) → Coord → ℝ)
    (n : ℕ) (stateIdeal : (Coord → ℝ) × (Coord → ℝ)) (a : Coord) =>
  project
    (fun (i : Coord) =>
      stateIdeal.1 i -
        GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1) *
          GKGMM19IterativeLocalVoting.linfL1SymmetricActiveDirection stateIdeal.1 stateIdeal.2 i)
    a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.linfL1SymmetricActiveDirection

GKGMM19IterativeLocalVoting.finiteModelBLinfOutcomeTrajectory

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] →
    [Nonempty Coord] →
      [DecidableEq Coord] →
        GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord →
          ℝ → ((Coord → ℝ) → Coord → ℝ) → (Coord → ℝ) → ℕ → (ℕ → Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] [DecidableEq Coord]
    (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord) (r0 : ℝ)
    (project : (Coord → ℝ) → Coord → ℝ) (initial : Coord → ℝ) (a : ℕ) (a_1 : ℕ → Coord → ℝ) (a_2 : Coord) =>
  Nat.brecOn (motive := fun (x : ℕ) => (ℕ → Coord → ℝ) → Coord → ℝ) a
    (GKGMM19IterativeLocalVoting.finiteModelBLinfOutcomeTrajectory._f D r0 project initial) a_1 a_2
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.finiteModelBIdealSample
- GKGMM19IterativeLocalVoting.finiteModelBLinfOutcomeStep

GKGMM19IterativeLocalVoting.linfL1WaterfillSubsets

Declaration location: paper-local

Lean declaration body

```
type:
(Coord : Type u_1) → [Fintype Coord] → Finset (Finset Coord)

value:
fun (Coord : Type u_1) [Fintype Coord] => Finset.univ.powerset.erase ∅
```

GKGMM19IterativeLocalVoting.linfL1WaterfillLevel

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → [Nonempty Coord] → (Coord → ℝ) → (Coord → ℝ) → ℝ → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (center ideal : Coord → ℝ) (r : ℝ) =>
  max 0
    ((GKGMM19IterativeLocalVoting.linfL1WaterfillSubsets Coord).sup'
      (GKGMM19IterativeLocalVoting.linfL1WaterfillSubsets_nonempty Coord) fun (A : Finset Coord) =>
      (∑ i ∈ A, |center i - ideal i| - r) / ↑A.card)
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.linfL1WaterfillSubsets

GKGMM19IterativeLocalVoting.linfL1WaterfillRawResponse

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → [Nonempty Coord] → (Coord → ℝ) → (Coord → ℝ) → ℝ → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (center ideal : Coord → ℝ) (r : ℝ) (a : Coord) =>
  center a -
    (center a - ideal a) / |center a - ideal a| *
      max (|center a - ideal a| - GKGMM19IterativeLocalVoting.linfL1WaterfillLevel center ideal r) 0
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.linfL1WaterfillLevel

GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeRawResponse

Declaration location: paper-local

Lean declaration body

```
type:
{Omega : Type u_1} →
  {Coord : Type u_2} →
    [Fintype Coord] → [Nonempty Coord] → ℝ → (ℕ → Omega → Coord → ℝ) → (ℕ → Omega → Coord → ℝ) → ℕ → Omega → Coord → ℝ

value:
fun {Omega : Type u_1} {Coord : Type u_2} [Fintype Coord] [Nonempty Coord] (r0 : ℝ)
    (trajectory idealSample : ℕ → Omega → Coord → ℝ) (n : ℕ) (omega : Omega) (a : Coord) =>
  GKGMM19IterativeLocalVoting.linfL1WaterfillRawResponse (trajectory n omega) (idealSample n omega)
    (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)) a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.linfL1WaterfillRawResponse

GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeStep

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [Fintype Coord] → [Nonempty Coord] → ℝ → ((Coord → ℝ) → Coord → ℝ) → ℕ → (Coord → ℝ) × (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (r0 : ℝ) (project : (Coord → ℝ) → Coord → ℝ) (n : ℕ)
    (stateIdeal : (Coord → ℝ) × (Coord → ℝ)) (a : Coord) =>
  project
    (GKGMM19IterativeLocalVoting.linfL1WaterfillRawResponse stateIdeal.1 stateIdeal.2
      (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)))
    a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.linfL1WaterfillRawResponse

GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] →
    [Nonempty Coord] →
      GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord →
        ℝ → ((Coord → ℝ) → Coord → ℝ) → (Coord → ℝ) → ℕ → (ℕ → Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord]
    (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord) (r0 : ℝ)
    (project : (Coord → ℝ) → Coord → ℝ) (initial : Coord → ℝ) (a : ℕ) (a_1 : ℕ → Coord → ℝ) (a_2 : Coord) =>
  Nat.brecOn (motive := fun (x : ℕ) => (ℕ → Coord → ℝ) → Coord → ℝ) a
    (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory._f D r0 project initial) a_1 a_2
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeStep
- GKGMM19IterativeLocalVoting.finiteModelBIdealSample

GKGMM19IterativeLocalVoting.lpGradientCandidate

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → ℝ → (Coord → ℝ) → (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (p : ℝ) (x absDeriv : Coord → ℝ) (a : Coord) =>
  |x a| ^ (p - 1) * absDeriv a / AppliedModelingLib.FiniteDimensionalNorms.lpPower p x ^ ((p - 1) / p)
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.lpPower

GKGMM19IterativeLocalVoting.lpCostGradientCandidate

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} → [Fintype Coord] → ℝ → (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (p : ℝ) (d : Coord → ℝ) (a : Coord) =>
  GKGMM19IterativeLocalVoting.lpGradientCandidate p d (fun (i : Coord) => d i / |d i|) a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.lpGradientCandidate

GKGMM19IterativeLocalVoting.finiteModelBOutcomeRawResponse

Declaration location: paper-local

Lean declaration body

```
type:
{Omega : Type u_1} →
  {Coord : Type u_2} →
    [Fintype Coord] → ℝ → ℝ → (ℕ → Omega → Coord → ℝ) → (ℕ → Omega → Coord → ℝ) → ℕ → Omega → Coord → ℝ

value:
fun {Omega : Type u_1} {Coord : Type u_2} [Fintype Coord] (p r0 : ℝ) (trajectory idealSample : ℕ → Omega → Coord → ℝ)
    (n : ℕ) (omega : Omega) (a : Coord) =>
  trajectory n omega a -
    GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1) *
      GKGMM19IterativeLocalVoting.lpCostGradientCandidate p
        (fun (j : Coord) => trajectory n omega j - idealSample n omega j) a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.lpCostGradientCandidate

GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] →
    GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord →
      ℝ → ℝ → ((Coord → ℝ) → Coord → ℝ) → (Coord → ℝ) → ℕ → (ℕ → Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord)
    (p r0 : ℝ) (project : (Coord → ℝ) → Coord → ℝ) (initial : Coord → ℝ) (a : ℕ) (a_1 : ℕ → Coord → ℝ) (a_2 : Coord) =>
  Nat.brecOn (motive := fun (x : ℕ) => (ℕ → Coord → ℝ) → Coord → ℝ) a
    (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory._f D p r0 project initial) a_1 a_2
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.finiteModelBIdealSample
- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.lpCostGradientCandidate

GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [Fintype Coord] → [Nonempty Coord] → GKGMM19IterativeLocalVoting.SourceNorm → (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (q : GKGMM19IterativeLocalVoting.SourceNorm)
    (gradient : Coord → ℝ) (a : Coord) =>
  (if gradient = 0 then fun (x : Coord) => 0
    else fun (i : Coord) => gradient i / GKGMM19IterativeLocalVoting.finiteCoordinateNorm q gradient)
    a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateNorm

GKGMM19IterativeLocalVoting.ModelBFiniteResponseAt

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  [Fintype Coord] →
    [Nonempty Coord] → GKGMM19IterativeLocalVoting.SourceNorm → (Coord → ℝ) → ℝ → (Coord → ℝ) → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (q : GKGMM19IterativeLocalVoting.SourceNorm)
    (center : Coord → ℝ) (r : ℝ) (gradient response : Coord → ℝ) =>
  response = fun (i : Coord) => center i + r * GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection q gradient i
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection

GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MeasurableSpace Voter] →
      [Fintype Coord] →
        [Nonempty Coord] → MeasureTheory.Measure Voter → (Voter → (Coord → ℝ) → Coord → ℝ) → (Coord → ℝ) → Coord → ℝ

value:
fun {Voter : Type u_1} {Coord : Type u_2} [MeasurableSpace Voter] [Fintype Coord] [Nonempty Coord]
    (population : MeasureTheory.Measure Voter) (utilityGradient : Voter → (Coord → ℝ) → Coord → ℝ) (x : Coord → ℝ)
    (a : Coord) =>
  ∫ (voter : Voter),
    GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection GKGMM19IterativeLocalVoting.SourceNorm.l2
      (utilityGradient voter x) a ∂population
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection

GKGMM19IterativeLocalVoting.theorem3NormalizedGradientDirection

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point → Voter → Point → Point

value:
fun {Voter : Type u_1} {Point : Type u_2} (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter Point) (voter : Voter)
    (x : Point) =>
  if E.utilityGradient voter x = E.zeroDirection then E.zeroDirection
  else
    E.scalarDirection
      (E.normDistance GKGMM19IterativeLocalVoting.SourceNorm.l2 (E.utilityGradient voter x) E.zeroDirection)⁻¹
      (E.utilityGradient voter x)
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.SourceNorm

GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel

Declaration location: paper-local

Lean declaration body

```
field population:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MeasurableSpace Voter] →
      [inst_1 : Fintype Coord] →
        [inst_2 : Nonempty Coord] →
          {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
            GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E → MeasureTheory.Measure Voter

field population_probability:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E),
  MeasureTheory.IsProbabilityMeasure self.population

field utilityGradient:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MeasurableSpace Voter] →
      [inst_1 : Fintype Coord] →
        [inst_2 : Nonempty Coord] →
          {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
            GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E → Voter → (Coord → ℝ) → Coord → ℝ

field utilityGradient_eq:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E),
  E.utilityGradient = self.utilityGradient

field utilityGradient_source_subgradient:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E) (voter : Voter) (x : Coord → ℝ),
  GKGMM19IterativeLocalVoting.FiniteSourceSubgradientAt (E.utility voter) x (self.utilityGradient voter x)

field scalarDirection_eq:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E),
  E.scalarDirection = GKGMM19IterativeLocalVoting.finiteScalarDirection

field zeroDirection_eq:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E), E.zeroDirection = fun (x : Coord) => 0

field normDistance_l2_zero_eq:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E) (g : Coord → ℝ),
  E.normDistance GKGMM19IterativeLocalVoting.SourceNorm.l2 g E.zeroDirection =
    GKGMM19IterativeLocalVoting.finiteCoordinateNorm GKGMM19IterativeLocalVoting.SourceNorm.l2 g

field normalized_gradient_integrable:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E) (x : Coord → ℝ) (i : Coord),
  MeasureTheory.Integrable
    (fun (voter : Voter) =>
      GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection GKGMM19IterativeLocalVoting.SourceNorm.l2
        (self.utilityGradient voter x) i)
    self.population

field directionalField_eq:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E),
  E.directionalField =
    GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField self.population self.utilityGradient

field voterExpectation_normalized_eq:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E) (x : Coord → ℝ),
  (E.voterExpectation fun (voter : Voter) =>
      GKGMM19IterativeLocalVoting.theorem3NormalizedGradientDirection E voter x) =
    GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField self.population self.utilityGradient x
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteSourceSubgradientAt
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateNorm
- GKGMM19IterativeLocalVoting.finiteScalarDirection
- GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection

- GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField
- GKGMM19IterativeLocalVoting.theorem3NormalizedGradientDirection

GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldUniformContinuity

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MeasurableSpace Voter] →
      [inst_1 : Fintype Coord] →
        [inst_2 : Nonempty Coord] →
          {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
            GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [MeasurableSpace Voter] [Fintype Coord] [Nonempty Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E) =>
  ∀ (ε : ℝ),
    0 < ε →
      ∃ (δ : ℝ),
        0 < δ ∧
          ∀ (x y : Coord → ℝ),
            GKGMM19IterativeLocalVoting.finiteCoordinateDistance GKGMM19IterativeLocalVoting.SourceNorm.l2 x y < δ →
              GKGMM19IterativeLocalVoting.finiteCoordinateDistance GKGMM19IterativeLocalVoting.SourceNorm.l2
                  (GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField M.population M.utilityGradient x)
                  (GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField M.population M.utilityGradient y) <
                ε
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance
- GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField

GKGMM19IterativeLocalVoting.populationTheorem3CenteredIncrement

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    {Omega : Type u_3} →
      [inst : MeasurableSpace Voter] →
        [inst_1 : Fintype Coord] →
          [inst_2 : Nonempty Coord] →
            {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
              GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E →
                ℝ → (ℕ → Omega → Coord → ℝ) → (ℕ → Omega → Voter) → (Coord → ℝ) → ℕ → Omega → ℝ

value:
fun {Voter : Type u_1} {Coord : Type u_2} {Omega : Type u_3} [MeasurableSpace Voter] [Fintype Coord] [Nonempty Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E) (r0 : ℝ)
    (trajectory : ℕ → Omega → Coord → ℝ) (sample : ℕ → Omega → Voter) (a : Coord → ℝ) (n : ℕ) (omega : Omega) =>
  GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1) *
    (GKGMM19IterativeLocalVoting.finiteDot a
        (GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField M.population M.utilityGradient
          (trajectory n omega)) -
      GKGMM19IterativeLocalVoting.finiteDot a
        (GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection GKGMM19IterativeLocalVoting.SourceNorm.l2
          (M.utilityGradient (sample n omega) (trajectory n omega))))
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel
- GKGMM19IterativeLocalVoting.SourceNorm

- GKGMM19IterativeLocalVoting.finiteDot
- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection
- GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField

GKGMM19IterativeLocalVoting.populationTheorem3PastCenteredTest

Declaration location: paper-local

Lean declaration body

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MeasurableSpace Voter] →
      [inst_1 : Fintype Coord] →
        [inst_2 : Nonempty Coord] →
          {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
            GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E →
              ℝ → (Coord → ℝ) → (Coord → ℝ) → Voter → ℝ

value:
fun {Voter : Type u_1} {Coord : Type u_2} [MeasurableSpace Voter] [Fintype Coord] [Nonempty Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E) (r : ℝ) (a state : Coord → ℝ)
    (voter : Voter) =>
  r *
    (GKGMM19IterativeLocalVoting.finiteDot a
        (GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField M.population M.utilityGradient state) -
      GKGMM19IterativeLocalVoting.finiteDot a
        (GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection GKGMM19IterativeLocalVoting.SourceNorm.l2
          (M.utilityGradient voter state)))
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteDot
- GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection
- GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField

GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource

Declaration location: paper-local

Lean declaration body

```
field initial:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MetricSpace Voter] →
      [inst_1 : SecondCountableTopology Voter] →
        [inst_2 : MeasurableSpace Voter] →
          [inst_3 : BorelSpace Voter] →
            [inst_4 : StandardBorelSpace Voter] →
              [inst_5 : Nonempty Voter] →
                [inst_6 : Fintype Coord] →
                  [inst_7 : Nonempty Coord] →
                    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
                      {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E} →
                        GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M → Coord → ℝ

field trajectory:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MetricSpace Voter] →
      [inst_1 : SecondCountableTopology Voter] →
        [inst_2 : MeasurableSpace Voter] →
          [inst_3 : BorelSpace Voter] →
            [inst_4 : StandardBorelSpace Voter] →
              [inst_5 : Nonempty Voter] →
                [inst_6 : Fintype Coord] →
                  [inst_7 : Nonempty Coord] →
                    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
                      {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E} →
                        GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M → ℕ → (ℕ → Voter) → Coord → ℝ
```

```
field state:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MetricSpace Voter] →
      [inst_1 : SecondCountableTopology Voter] →
        [inst_2 : MeasurableSpace Voter] →
          [inst_3 : BorelSpace Voter] →
            [inst_4 : StandardBorelSpace Voter] →
              [inst_5 : Nonempty Voter] →
                [inst_6 : Fintype Coord] →
                  [inst_7 : Nonempty Coord] →
                    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
                      {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E} →
                        GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M →
                          (n : ℕ) → (Fin (n + 1) → Voter) → Coord → ℝ

field r0:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MetricSpace Voter] →
      [inst_1 : SecondCountableTopology Voter] →
        [inst_2 : MeasurableSpace Voter] →
          [inst_3 : BorelSpace Voter] →
            [inst_4 : StandardBorelSpace Voter] →
              [inst_5 : Nonempty Voter] →
                [inst_6 : Fintype Coord] →
                  [inst_7 : Nonempty Coord] →
                    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
                      {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E} →
                        GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M → ℝ

field r0_pos:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M), 0 < self.r0

field solutionSpace_univ:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M), E.solutionSpace = Set.univ

field field_continuity:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M),
  GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldUniformContinuity M

field initial_eq:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M) (omega : ℕ → Voter),
  self.trajectory 0 omega = self.initial

field trajectory_from_past:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M) (n : ℕ) (omega : ℕ → Voter),
  self.trajectory n omega = self.state n (AppliedModelingLib.iidSequencePastPrefix n omega)

field raw_update:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M) (n : ℕ) (omega : ℕ → Voter),
  self.trajectory (n + 1) omega = fun (i : Coord) =>
    self.trajectory n omega i +
      GKGMM19IterativeLocalVoting.ilvRadius self.r0 (n + 1) *
        GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection GKGMM19IterativeLocalVoting.SourceNorm.l2
          (M.utilityGradient (AppliedModelingLib.iidSequenceSample n omega) (self.trajectory n omega)) i

field centered_test_integrable:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M) (a : Coord → ℝ) (n : ℕ),
  MeasureTheory.Integrable
    (fun (z : (Fin (n + 1) → Voter) × Voter) =>
```

```
    GKGMM19IterativeLocalVoting.populationTheorem3PastCenteredTest M
      (GKGMM19IterativeLocalVoting.ilvRadius self.r0 (n + 1)) a (self.state n z.1) z.2)
  (MeasureTheory.Measure.map
    (fun (omega : ℕ → Voter) =>
      (AppliedModelingLib.iidSequencePastPrefix n omega, AppliedModelingLib.iidSequenceSample n omega))
    (AppliedModelingLib.iidSequenceMeasure M.population))

field centered_partial_sum_adapted:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M) (a : Coord → ℝ),
  MeasureTheory.StronglyAdapted AppliedModelingLib.iidSequenceNaturalFiltration fun (n : ℕ) (omega : ℕ → Voter) =>
    ∑ i ∈ Finset.range n,
      GKGMM19IterativeLocalVoting.populationTheorem3CenteredIncrement M self.r0 self.trajectory
        AppliedModelingLib.iidSequenceSample a i omega

field centered_aestronglyMeasurable:
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E}
  (self : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M) (a : Coord → ℝ) (n : ℕ),
  MeasureTheory.AEStronglyMeasurable
    (GKGMM19IterativeLocalVoting.populationTheorem3CenteredIncrement M self.r0 self.trajectory
      AppliedModelingLib.iidSequenceSample a n)
    (AppliedModelingLib.iidSequenceMeasure M.population)
```

Direct retained dependencies

- AppliedModelingLib.iidSequenceMeasure
- AppliedModelingLib.iidSequenceNaturalFiltration
- AppliedModelingLib.iidSequencePastPrefix
- AppliedModelingLib.iidSequenceSample
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel
- GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldUniformContinuity
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection
- GKGMM19IterativeLocalVoting.populationTheorem3CenteredIncrement
- GKGMM19IterativeLocalVoting.populationTheorem3PastCenteredTest

GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component →
          (Component → ℝ) × (Coord → ℝ) → (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
    (sample : (Component → ℝ) × (Coord → ℝ)) (state : Coord → ℝ) =>
  ∑ k : Component,
    D.sampleWeight sample k / D.sampleWeightNorm2 sample *
      GKGMM19IterativeLocalVoting.finiteCoordinateBlockL2Distance (D.componentBlock k) state (D.sampleIdeal sample k)
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleIdeal
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeight
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeightNorm2
- GKGMM19IterativeLocalVoting.finiteCoordinateBlockL2Distance

GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelARawResponseAt

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        [Nonempty Coord] →
          GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component →
            (Coord → ℝ) → ℝ → (Component → ℝ) × (Coord → ℝ) → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component] [Nonempty Coord]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component) (state : Coord → ℝ) (radius : ℝ)
    (sample : (Component → ℝ) × (Coord → ℝ)) (response : Coord → ℝ) =>
  GKGMM19IterativeLocalVoting.finiteCoordinateDistance GKGMM19IterativeLocalVoting.SourceNorm.l2 response state ≤
      radius ∧
    ∀ (candidate : Coord → ℝ),
      GKGMM19IterativeLocalVoting.finiteCoordinateDistance GKGMM19IterativeLocalVoting.SourceNorm.l2 candidate state ≤
          radius →
        GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost D sample response ≤
          GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost D sample candidate
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost

GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource

Declaration location: paper-local

Lean declaration body

```
field rawResponse:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        [inst_2 : Nonempty Coord] →
          {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component} →
            GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D →
              (Coord → ℝ) → (Component → ℝ) × (Coord → ℝ) → ℝ → Coord → ℝ

field rawResponse_measurable:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  [inst_2 : Nonempty Coord] {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component}
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D) (radius : ℝ),
  Measurable fun (z : (Coord → ℝ) × (Component → ℝ) × (Coord → ℝ)) => self.rawResponse z.1 z.2 radius

field rawResponse_exact_of_valid:
∀ {Coord : Type u_1} {Component : Type u_2} [inst : Fintype Coord] [inst_1 : Fintype Component]
  [inst_2 : Nonempty Coord] {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component}
  (self : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D) (state : Coord → ℝ)
  (radius : ℝ) (sample : (Component → ℝ) × (Coord → ℝ)),
  0 < radius →
    (∀ (k : Component), 0 ≤ D.sampleWeight sample k) →
      0 < D.sampleWeightNorm2 sample →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelARawResponseAt D state radius sample
          (self.rawResponse state sample radius)
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeight
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeightNorm2
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelARawResponseAt

GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeStep

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        [inst_2 : Nonempty Coord] →
          {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component} →
            GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D →
              ℝ → ((Coord → ℝ) → Coord → ℝ) → ℕ → (Coord → ℝ) × (Component → ℝ) × (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component] [Nonempty Coord]
    {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component}
    (A : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D) (r0 : ℝ)
    (project : (Coord → ℝ) → Coord → ℝ) (n : ℕ) (stateSample : (Coord → ℝ) × (Component → ℝ) × (Coord → ℝ))
    (a : Coord) =>
  project (A.rawResponse stateSample.1 stateSample.2 (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))) a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource
- GKGMM19IterativeLocalVoting.ilvRadius

GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        [inst_2 : Nonempty Coord] →
          {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component} →
            GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D →
              ℝ → ((Coord → ℝ) → Coord → ℝ) → (Coord → ℝ) → ℕ → (ℕ → (Component → ℝ) × (Coord → ℝ)) → Coord → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component] [Nonempty Coord]
    {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component}
    (A : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D) (r0 : ℝ)
    (project : (Coord → ℝ) → Coord → ℝ) (initial : Coord → ℝ) (a : ℕ) (a_1 : ℕ → (Component → ℝ) × (Coord → ℝ))
    (a_2 : Coord) =>
  Nat.brecOn (motive := fun (x : ℕ) => (ℕ → (Component → ℝ) × (Coord → ℝ)) → Coord → ℝ) a
    (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory._f A r0 project initial) a_1 a_2
```

Direct retained dependencies

- AppliedModelingLib.iidSequenceSample
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeStep

GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelARawDirection

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        [inst_2 : Nonempty Coord] →
          {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component} →
            GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D →
              (Coord → ℝ) → ℝ → (Component → ℝ) × (Coord → ℝ) → Coord → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component] [Nonempty Coord]
    {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component}
    (A : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D) (state : Coord → ℝ)
    (radius : ℝ) (sample : (Component → ℝ) × (Coord → ℝ)) (a : Coord) =>
  (state a - A.rawResponse state sample radius a) / radius
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource

GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAResponseSourceFormula

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        [inst_2 : Nonempty Coord] →
          (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component) →
            GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D → Prop

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component] [Nonempty Coord]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
    (A : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D) =>
  (∀ (radius : ℝ), Measurable fun (z : (Coord → ℝ) × (Component → ℝ) × (Coord → ℝ)) => A.rawResponse z.1 z.2 radius) ∧
    ∀ (state : Coord → ℝ) (radius : ℝ) (sample : (Component → ℝ) × (Coord → ℝ)),
      0 < radius →
        (∀ (k : Component), 0 ≤ D.sampleWeight sample k) →
          0 < D.sampleWeightNorm2 sample →
            GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelARawResponseAt D state radius sample
              (A.rawResponse state sample radius)
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeight
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeightNorm2
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelARawResponseAt
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource

GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationCost

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component → (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component) (a : Coord → ℝ) =>
  ∫ (sample : (Component → ℝ) × (Coord → ℝ)),
    GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost D sample a ∂D.jointMeasure
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost

GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationMinimizerSet

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component →
          Set (Coord → ℝ) → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component) (solutionSpace : Set (Coord → ℝ))
    (a : Coord → ℝ) =>
  {x : Coord → ℝ |
      x ∈ solutionSpace ∧
        IsMinOn (GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationCost D) solutionSpace x}
    a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationCost

GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleSocialObjectiveSource

Declaration location: paper-local

Lean declaration body

```
field minimizerSet_eq_socialOptimal:
∀ {Voter : Type u_1} {Coord : Type u_2} {Component : Type u_3} [inst : Fintype Coord] [inst_1 : Fintype Component]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component},
  GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleSocialObjectiveSource E D →
    GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationMinimizerSet D E.solutionSpace = E.socialOptimal
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationMinimizerSet

GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleUtility

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component →
          (Component → ℝ) × (Coord → ℝ) → (Coord → ℝ) → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
    (sample : (Component → ℝ) × (Coord → ℝ)) (state : Coord → ℝ) =>
  -GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost D sample state
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost

GKGMM19IterativeLocalVoting.weightedEuclideanL2BlockSampleDirection

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    {Sample : Type u_3} →
      [Fintype Coord] →
        Finset Component →
          (Component → Finset Coord) →
            (Sample → Component → ℝ) →
              (Sample → ℝ) → (Sample → Component → Coord → ℝ) → (Coord → ℝ) → Sample → Coord → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} {Sample : Type u_3} [Fintype Coord] (components : Finset Component)
    (componentBlock : Component → Finset Coord) (sampleWeight : Sample → Component → ℝ) (sampleWeightNorm2 : Sample → ℝ)
    (sampleIdeal : Sample → Component → Coord → ℝ) (state : Coord → ℝ) (sample : Sample) (a : Coord) =>
  ∑ k ∈ components,
    sampleWeight sample k / sampleWeightNorm2 sample *
      GKGMM19IterativeLocalVoting.l2BlockDistanceNormalizedGradient (componentBlock k) state (sampleIdeal sample k) a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.l2BlockDistanceNormalizedGradient

GKGMM19IterativeLocalVoting.weightedEuclideanL2BlockOutcomeStep

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    {Sample : Type u_3} →
      [Fintype Coord] →
        Finset Component →
          (Component → Finset Coord) →
            (Sample → Component → ℝ) →
              (Sample → ℝ) →
                (Sample → Component → Coord → ℝ) → ((Coord → ℝ) → Coord → ℝ) → ℝ → (Coord → ℝ) × Sample → Coord → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} {Sample : Type u_3} [Fintype Coord] (components : Finset Component)
    (componentBlock : Component → Finset Coord) (sampleWeight : Sample → Component → ℝ) (sampleWeightNorm2 : Sample → ℝ)
    (sampleIdeal : Sample → Component → Coord → ℝ) (project : (Coord → ℝ) → Coord → ℝ) (radius : ℝ)
    (stateSample : (Coord → ℝ) × Sample) (a : Coord) =>
  project
    (fun (i : Coord) =>
      stateSample.1 i -
        radius *
          GKGMM19IterativeLocalVoting.weightedEuclideanL2BlockSampleDirection components componentBlock sampleWeight
            sampleWeightNorm2 sampleIdeal stateSample.1 stateSample.2 i)
    a
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.weightedEuclideanL2BlockSampleDirection

GKGMM19IterativeLocalVoting.weightedEuclideanL2BlockOutcomeTrajectory

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    {Sample : Type u_3} →
      [Fintype Coord] →
        Finset Component →
          (Component → Finset Coord) →
            (Sample → Component → ℝ) →
              (Sample → ℝ) →
                (Sample → Component → Coord → ℝ) →
                  ℝ → ((Coord → ℝ) → Coord → ℝ) → (Coord → ℝ) → ℕ → (ℕ → Sample) → Coord → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} {Sample : Type u_3} [Fintype Coord] (components : Finset Component)
    (componentBlock : Component → Finset Coord) (sampleWeight : Sample → Component → ℝ) (sampleWeightNorm2 : Sample → ℝ)
    (sampleIdeal : Sample → Component → Coord → ℝ) (r0 : ℝ) (project : (Coord → ℝ) → Coord → ℝ) (initial : Coord → ℝ)
    (a : ℕ) (a_1 : ℕ → Sample) (a_2 : Coord) =>
  Nat.brecOn (motive := fun (x : ℕ) => (ℕ → Sample) → Coord → ℝ) a
    (GKGMM19IterativeLocalVoting.weightedEuclideanL2BlockOutcomeTrajectory._f components componentBlock sampleWeight
      sampleWeightNorm2 sampleIdeal r0 project initial)
    a_1 a_2
```

Direct retained dependencies

- AppliedModelingLib.iidSequenceSample
- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.weightedEuclideanL2BlockOutcomeStep

GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleOutcomeTrajectory

Declaration location: paper-local

Lean declaration body

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] →
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component →
          ℝ → ((Coord → ℝ) → Coord → ℝ) → (Coord → ℝ) → ℕ → (ℕ → (Component → ℝ) × (Coord → ℝ)) → Coord → ℝ

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component) (r0 : ℝ)
    (project : (Coord → ℝ) → Coord → ℝ) (initial : Coord → ℝ) (a : ℕ) (a_1 : ℕ → (Component → ℝ) × (Coord → ℝ))
    (a_2 : Coord) =>
  GKGMM19IterativeLocalVoting.weightedEuclideanL2BlockOutcomeTrajectory Finset.univ D.componentBlock D.sampleWeight
    D.sampleWeightNorm2 D.sampleIdeal r0 project initial a a_1 a_2
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleIdeal
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeight
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeightNorm2
- GKGMM19IterativeLocalVoting.weightedEuclideanL2BlockOutcomeTrajectory

Material library prerequisites

AppliedModelingLib.Optimization.EuclideanProjectionSourceGeometry

Source locator: cited publication, lines 449-452

Verbatim paper-source connection

```
C1 The solution space X ⊆ RM is non-empty, bounded, closed, and convex.
 C2 Each voter v has a unique ideal solution xv ∈ X .
 C3 The ideal point xv of each voter is drawn independently from a probability distribution
      with a bounded and measurable density function hX .

[Next verbatim source excerpt]

 Algorithm 1: Iterative Local Voting (ILV)
  Inputs: Initial solution x0 ∈ X , tolerance ϵ > 0, an integer N , initial radius r0 > 0,
    termination time T , norm q for local neighborhood.
   Output:     Solution x.

     • For t ≥ 1, sample a voter vt ∈ V at random from the population; set rt = r0 /t
        and elicit


                             x0t = arg maxx∈{s:ks−xt−1 kq ≤rt } fvt (x),            (1)
                                         0
        and then compute xt = [xt ]X , where [·]X is a projection onto space X ; i.e. ask
        the voter to move to her favorite point within constraints, and then project the
        reported point onto X .

[Next verbatim source excerpt]

   More formally, consider a M -dimensional societal decision problem in X  ⊂ RM and a
population of voters V , where each voter v ∈ V has bounded utility fv (x) ∈ R, ∀ x ∈ X .
Then we consider the class of algorithms described in Algorithm 1. We study the algorithm
class under two plausible models of how voters respond to query (1), which asks for the
voter's favorite point in a local region.
```

Lean declaration type (metadata)

```
field closed_solutionSpace:
∀ {Coord : Type u_1} [inst : Fintype Coord] {solutionSpace : Set (Coord → ℝ)} {project : (Coord → ℝ) → Coord → ℝ},
  AppliedModelingLib.Optimization.EuclideanProjectionSourceGeometry Coord solutionSpace project → IsClosed solutionSpace

field bounded_solutionSpace:
∀ {Coord : Type u_1} [inst : Fintype Coord] {solutionSpace : Set (Coord → ℝ)} {project : (Coord → ℝ) → Coord → ℝ},
  AppliedModelingLib.Optimization.EuclideanProjectionSourceGeometry Coord solutionSpace project →
    Bornology.IsBounded solutionSpace

field convex_solutionSpace:
∀ {Coord : Type u_1} [inst : Fintype Coord] {solutionSpace : Set (Coord → ℝ)} {project : (Coord → ℝ) → Coord → ℝ},
  AppliedModelingLib.Optimization.EuclideanProjectionSourceGeometry Coord solutionSpace project → Convex ℝ solutionSpace

field closest_point_projection:
∀ {Coord : Type u_1} [inst : Fintype Coord] {solutionSpace : Set (Coord → ℝ)} {project : (Coord → ℝ) → Coord → ℝ},
  AppliedModelingLib.Optimization.EuclideanProjectionSourceGeometry Coord solutionSpace project →
    AppliedModelingLib.Optimization.EuclideanProjectionOnto solutionSpace project
```

Exact Lean library declaration

```
/--
The finite-dimensional Euclidean feasible-set data shared by projected
optimization algorithms.  A closest-point projection onto a closed bounded
convex set is both compactly supported and nonexpansive in squared distance;
the lemmas below expose those consequences once for all applications.
-/
structure EuclideanProjectionSourceGeometry
    (Coord : Type*) [Fintype Coord]
    (solutionSpace : Set (Coord → ℝ))
    (project : (Coord → ℝ) → Coord → ℝ) : Prop where
  closed_solutionSpace : IsClosed solutionSpace
  bounded_solutionSpace : Bornology.IsBounded solutionSpace
  convex_solutionSpace : Convex ℝ solutionSpace
  closest_point_projection : EuclideanProjectionOnto solutionSpace project
```

Direct retained dependencies

- AppliedModelingLib.Optimization.EuclideanProjectionOnto

Recorded source-to-library screening **Verdict:** matches

Reason: The structure packages the closed, bounded, convex C1 solution space together with Algorithm 1’s Euclidean closest-point projection; each field appears in the selected raw source bundle.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Paper-specific semantic prerequisites

GKGMM19IterativeLocalVoting.sourceNormFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.sourceNormFormulaData
Source locator: cited publication, lines 100-124

Verbatim paper-source connection

local neighborhoods are defined – in this paper we focus on neighborhoods that are balls in the L
q norm, and in particular on the cases where q = 1, 2 or ∞. (For M < ∞ dimensional

316

[form-feed in source extraction]
Iterative Local Voting for Collective Decision-making in Continuous Spaces

q norm kxk ,
p P
vectors, the L q
m |xm | . q = 1, 2 and ∞ neighborhoods correspond to
q
q
bounds on the sum of absolute values of the changes, the sum of the square of the changes, and the maximum change, respectively.)

More formally, consider a M -dimensional societal decision problem in X ⊂ RM and a population of voters V , where each voter v ∈ V has bounded utility fv (x) ∈ R, ∀ x ∈ X . Then we consider the class of algorithms described in Algorithm 1. We study the algorithm class under two plausible models of how voters respond to query (1), which asks for the voter's favorite point in a local region.

[Next verbatim source excerpt]

Theorem 2. Suppose that conditions C1 , C2 , and C3 are satisfied, the voter utilities are Lp normed, and voters respond to query (1) according to Model B. Then, ILV with Lq neighborhoods converges to the societal optimal point w.p. 1 for any p > 0 and q > 0 such that 1/p + 1/q = 1.

Lean-expanded paper semantic target

```
type:
GKGMM19IterativeLocalVoting.SourceNorm → Prop

value:
fun (a : GKGMM19IterativeLocalVoting.SourceNorm) =>
  match a with
  | GKGMM19IterativeLocalVoting.SourceNorm.l1 => True
  | GKGMM19IterativeLocalVoting.SourceNorm.l2 => True
  | GKGMM19IterativeLocalVoting.SourceNorm.linfty => True
  | GKGMM19IterativeLocalVoting.SourceNorm.lp p => 0 < p
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.SourceNorm

Recorded source-to-declaration screening **Verdict:** matches

Reason: The source defines local neighborhoods using the finite-dimensional Lq formula and explicitly singles out q = 1, 2, or infinity, while Theorem 2 permits finite dual Lp/Lq parameters with p > 0 and q > 0. The Lean target has exactly the L1, L2, Linfinity, and finite Lp constructors and requires 0 < p in the finite-exponent case; its displayed uses apply the same predicate to either p or q. The duality equation belongs to the theorem-level relation between two parameters, not to this single-parameter admissibility predicate.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GKGMM19IterativeLocalVoting.finiteCoordinateModelARawResponseFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.finiteCoordinateModelARawResponseFormulaData
Source locator: cited publication, lines 126-136

Verbatim paper-source connection

• Model A: One possibility is that voters exactly perform the maximization asked of them, responding with their favorite point in the given L
q norm constraint set. In other

words, they return a point arg maxx∈{s:ks−x fvt (x). Note that by definition of this
t−1 kq ≤rt }
movement, the algorithm is myopically incentive compatible : if a voter is the last voter and no projections are used, then truthfully performing this movement is the dominant strategy. In general, the mechanism is not globally incentive compatible, nor incentive compatible with projections onto the feasible region. Simple examples of manipulations in both instances exist.

[Next verbatim source excerpt]

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution x0 ∈ X , tolerance ϵ > 0, an integer N , initial radius r0 > 0,
termination time T , norm q for local neighborhood.
Output: Solution x.

• For t ≥ 1, sample a voter vt ∈ V at random from the population; set rt = r0 /t and elicit

x0t = arg maxx∈{s:ks−xt−1 kq ≤rt } fvt (x), (1)
0
and then compute xt = [xt]X , where [·]X is a projection onto space X ; i.e. ask the voter to move to her favorite point within constraints, and then project the reported point onto X .

Lean-expanded paper semantic target

type:
{Voter : Type u_1} →
{Coord : Type u_2} →
[Fintype Coord] →
[Nonempty Coord] →
GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ) →
GKGMM19IterativeLocalVoting.SourceNorm → (Coord → ℝ) → ℝ → Voter → (Coord → ℝ) → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord] [Nonempty Coord]
(E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)) (q : GKGMM19IterativeLocalVoting.SourceNorm)
(center : Coord → ℝ) (radius : ℝ) (voter : Voter) (response : Coord → ℝ) =>
GKGMM19IterativeLocalVoting.sourceNormFormulaData q ∧
GKGMM19IterativeLocalVoting.finiteCoordinateDistance q response center ≤ radius ∧
∀ (candidate : Coord → ℝ),
GKGMM19IterativeLocalVoting.finiteCoordinateDistance q candidate center ≤ radius →
E.utility voter candidate ≤ E.utility voter response

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance
- GKGMM19IterativeLocalVoting.sourceNormFormulaData

Recorded source-to-declaration screening **Verdict:** matches

Reason: The Model A source response is an arbitrary maximizer over the raw Lq ball centered at the current point. The target requires the response to lie in that ball and to have utility at least that of every candidate in the same ball, with sourceNormFormulaData and finiteCoordinateDistance supplying the displayed norm formula. Projection is correctly absent from this raw-response relation and occurs later in Algorithm 1.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GKGMM19IterativeLocalVoting.modelBFiniteResponseFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.modelBFiniteResponseFormulaData
Source locator: cited publication, lines 140-148

Verbatim paper-source connection

```
• Model B:      On the other hand, voters may not actually search within the constraint set
  to find their favorite point inside of it. Rather, a voter v may have an idea about how
  to best improve the current point and then move in that direction to the boundary of
  the given constraint set. This model leads to a voter moving the current solution in the
                                                                    gt
  direction of the gradient of her utility function, returning a point xt−1 + rt
                                                                  kgt kq , for some
  gt ∈ ∂fvt (xt−1 ). Note that ∂f (x) denotes the set of subgradients of a function f at x, i.e.
  g ∈ ∂f (x) if ∀y , f (y) − f (x) ≥ g T (y − x).
```

Settled source reading

When the displayed Model B gradient is zero, the voter remains at the current point.

Lean-expanded paper semantic target

```
type:
{Coord : Type u_1} →
  [Fintype Coord] →
    [Nonempty Coord] → ((Coord → ℝ) → ℝ) → GKGMM19IterativeLocalVoting.SourceNorm → (Coord → ℝ) → ℝ → (Coord → ℝ) → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] [Nonempty Coord] (utility : (Coord → ℝ) → ℝ)
    (q : GKGMM19IterativeLocalVoting.SourceNorm) (center : Coord → ℝ) (r : ℝ) (response : Coord → ℝ) =>
  GKGMM19IterativeLocalVoting.sourceNormFormulaData q ∧
    ∃ (gradient : Coord → ℝ),
      (∀ (y : Coord → ℝ),
          utility y - utility center ≥
            (AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional gradient) fun (i : Coord) =>
              y i - center i) ∧
        (gradient = 0 ∧ response = center ∨
          gradient ≠ 0 ∧
            response = fun (i : Coord) =>
              center i + r * (gradient i / GKGMM19IterativeLocalVoting.finiteCoordinateNorm q gradient))
```

Direct retained dependencies

- AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateNorm
- GKGMM19IterativeLocalVoting.sourceNormFormulaData

Recorded source-to-declaration screening

Verdict: matches
Reason: The target requires sourceNormFormulaData q before choosing a source subgradient, so q is restricted to the source's admissible L1, L2, Linfinity, or positive-exponent finite Lp cases. Its subgradient inequality and nonzero-gradient response center + r * gradient/||gradient||_q reproduce Model B, while the zero-gradient response=center branch is exactly the approved stay-put convention.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GKGMM19IterativeLocalVoting.finiteCoordinateAlgorithm1ILVFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.finiteCoordinateAlgorithm1ILVFormulaData
Source locator: cited publication, lines 140-148

Verbatim paper-source connection

• Model B: On the other hand, voters may not actually search within the constraint set to find their favorite point inside of it. Rather, a voter v may have an idea about how to best improve the current point and then move in that direction to the boundary of the given constraint set. This model leads to a voter moving the current solution in the direction of the gradient of her utility function, returning a point $x_{t-1} + r_t \frac{g_t}{\|g_t\|_q}$, for some $g_t \in \partial f_{v_t}(x_{t-1})$. Note that $\partial f(x)$ denotes the set of subgradients of a function f at x, i.e. $g \in \partial f(x)$ if $\forall y$, $f(y) - f(x) \geq g^T(y - x)$.

[Next verbatim source excerpt]

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution $x_0 \in X$, tolerance $\epsilon > 0$, an integer N, initial radius $r_0 > 0$, termination time T, norm q for local neighborhood.
Output: Solution x.

- For $t \geq 1$, sample a voter $v_t \in V$ at random from the population; set $r_t = r_0/t$ and elicit

$$x'_t = \arg\max_{x \in \{s : \|s - x_{t-1}\|_q \leq r_t\}} f_{v_t}(x), \qquad (1)$$

  and then compute $x_t = [x'_t]_X$, where $[\cdot]_X$ is a projection onto space X; i.e. ask the voter to move to her favorite point within constraints, and then project the reported point onto X.

- Stop when either $t = T$, in which case return x_T, or when $\max_{l,\, m \in \{t-N,\ldots,t\}} |x_l - x_m| \leq \epsilon$, in which case return $x = x_t$.

[Next verbatim source excerpt]

More formally, consider a M-dimensional societal decision problem in $X \subset \mathbb{R}^M$ and a population of voters V, where each voter $v \in V$ has bounded utility $f_v(x) \in \mathbb{R}$, $\forall x \in X$. Then we consider the class of algorithms described in Algorithm 1. We study the algorithm class under two plausible models of how voters respond to query (1), which asks for the voter's favorite point in a local region.

• Model A: One possibility is that voters exactly perform the maximization asked of them, responding with their favorite point in the given L_q norm constraint set. In other words, they return a point $\arg\max_{x \in \{s : \|s - x_{t-1}\|_q \leq r_t\}} f_{v_t}(x)$. Note that by definition of this movement, the algorithm is myopically incentive compatible : if a voter is the last voter and no projections are used, then truthfully performing this movement is the dominant strategy. In general, the mechanism is not globally incentive compatible, nor incentive compatible with projections onto the feasible region. Simple examples of manipulations in both instances exist.

• Model B: On the other hand, voters may not actually search within the constraint set to find their favorite point inside of it. Rather, a voter v may have an idea about how to best improve the current point and then move in that direction to the boundary of the given constraint set. This model leads to a voter moving the current solution in the direction of the gradient of her utility function, returning a point $x_{t-1} + r_t \frac{g_t}{\|g_t\|_q}$, for some $g_t \in \partial f_{v_t}(x_{t-1})$. Note that $\partial f(x)$ denotes the set of subgradients of a function f at x, i.e. $g \in \partial f(x)$ if $\forall y$, $f(y) - f(x) \geq g^T(y - x)$.

Lean-expanded paper semantic target

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : MeasurableSpace Voter] →
      [Fintype Coord] →
        [Nonempty Coord] →
          MeasureTheory.Measure Voter →
            MeasureTheory.Measure (ℕ → Voter) →
              GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ) →
                GKGMM19IterativeLocalVoting.SourceNorm →
                  (Coord → ℝ) →
                    ℝ →
                      ℕ →
                        ℝ →
                          ℕ →
                            ((Coord → ℝ) → Coord → ℝ) →
                              ((ℕ → Voter) → ℕ → Coord → ℝ) →
```

```
                    ((ℕ → Voter) → ℕ → Coord → ℝ) → ((ℕ → Voter) → ℕ) → ((ℕ → Voter) → Coord → ℝ) → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [MeasurableSpace Voter] [Fintype Coord] [Nonempty Coord]
  (population : MeasureTheory.Measure Voter) (executionLaw : MeasureTheory.Measure (ℕ → Voter))
  (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)) (q : GKGMM19IterativeLocalVoting.SourceNorm)
  (initial : Coord → ℝ) (epsilon : ℝ) (N : ℕ) (r0 : ℝ) (T : ℕ) (project : (Coord → ℝ) → Coord → ℝ)
  (rawResponse trajectory : (ℕ → Voter) → ℕ → Coord → ℝ) (stopTime : (ℕ → Voter) → ℕ)
  (output : (ℕ → Voter) → Coord → ℝ) =>
 GKGMM19IterativeLocalVoting.sourceNormFormulaData q ∧
  MeasureTheory.IsProbabilityMeasure population ∧
   (executionLaw = MeasureTheory.Measure.infinitePi fun (x : ℕ) => population) ∧
    initial ∈ E.solutionSpace ∧
     0 < epsilon ∧
      0 < r0 ∧
       0 < T ∧
        AppliedModelingLib.Optimization.EuclideanProjectionOnto E.solutionSpace project ∧
         ((∀ (omega : ℕ → Voter) (t : ℕ),
            GKGMM19IterativeLocalVoting.finiteCoordinateModelARawResponseFormulaData E q
             (trajectory omega t) (GKGMM19IterativeLocalVoting.ilvRadius r0 (t + 1)) (omega (t + 1))
             (rawResponse omega (t + 1))) ∨
           ∀ (omega : ℕ → Voter) (t : ℕ),
            GKGMM19IterativeLocalVoting.modelBFiniteResponseFormulaData (E.utility (omega (t + 1))) q
             (trajectory omega t) (GKGMM19IterativeLocalVoting.ilvRadius r0 (t + 1))
             (rawResponse omega (t + 1))) ∧
          ∀ (omega : ℕ → Voter),
           trajectory omega 0 = initial ∧
            (∀ (t : ℕ), trajectory omega (t + 1) = project (rawResponse omega (t + 1))) ∧
             0 < stopTime omega ∧
              stopTime omega ≤ T ∧
               (stopTime omega = T ∨
                 ∀ (l m : ℕ),
                  l ∈ Finset.Icc (stopTime omega - N) (stopTime omega) →
                   m ∈ Finset.Icc (stopTime omega - N) (stopTime omega) →
                    GKGMM19IterativeLocalVoting.finiteCoordinateDistance
                      GKGMM19IterativeLocalVoting.SourceNorm.l2 (trajectory omega l)
                      (trajectory omega m) ≤
                     epsilon) ∧
                (∀ (t : ℕ),
                  0 < t →
                   t < stopTime omega →
                    ¬∀ (l m : ℕ),
                      l ∈ Finset.Icc (t - N) t →
                       m ∈ Finset.Icc (t - N) t →
                        GKGMM19IterativeLocalVoting.finiteCoordinateDistance
                          GKGMM19IterativeLocalVoting.SourceNorm.l2 (trajectory omega l)
                          (trajectory omega m) ≤
                         epsilon) ∧
                 output omega = trajectory omega (stopTime omega)
```

Direct retained dependencies

- AppliedModelingLib.Optimization.EuclideanProjectionOnto
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance
- GKGMM19IterativeLocalVoting.finiteCoordinateModelARawResponseFormulaData
- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.modelBFiniteResponseFormulaData
- GKGMM19IterativeLocalVoting.sourceNormFormulaData

Recorded source-to-declaration screening **Verdict:** matches

Reason: The target now places the Model A/Model B alternative outside every execution-path and time quantifier, so one fixed response model governs the whole execution. In the Model A branch every raw response is a utility maximizer over the source Lq ball; in the Model B branch every raw response is the source normalized subgradient update with the approved zero-gradient convention. Both branches then use the same independent population draws, r0/t schedule, Euclidean projection, earliest T-or-window stop, and output specified by Algorithm 1.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

GKGMM19IterativeLocalVoting.finiteCoordinateC1C2SourceFormula

Paper-local declaration:
GKGMM19IterativeLocalVoting.finiteCoordinateC1C2SourceFormula
Source locator: cited publication, lines 449-452

Verbatim paper-source connection

```
C1 The solution space X ⊆ RM is non-empty, bounded, closed, and convex.
 C2 Each voter v has a unique ideal solution xv ∈ X .
 C3 The ideal point xv of each voter is drawn independently from a probability distribution
      with a bounded and measurable density function hX .
```

Lean-expanded paper semantic target

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [inst : Fintype Coord] →
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)} →
        {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} →
          {project : (Coord → ℝ) → Coord → ℝ} → GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ}
    (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project) =>
  E.solutionSpace.Nonempty ∧
    IsClosed E.solutionSpace ∧
      Bornology.IsBounded E.solutionSpace ∧
        Convex ℝ E.solutionSpace ∧
          ∀ (voter : Voter),
            E.ideal voter ∈ E.solutionSpace ∧
              (∀ x ∈ E.solutionSpace, E.utility voter x ≤ E.utility voter (E.ideal voter)) ∧
                ∀ x ∈ E.solutionSpace, E.utility voter x = E.utility voter (E.ideal voter) → x = E.ideal voter
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.ILVEnvironment

Recorded source-to-declaration screening **Verdict:** matches

Reason: The source C1 clauses are exactly the target's nonempty, closed, bounded, convex solution space, and source C2 is exactly the target's feasible designated ideal that weakly maximizes utility with equality only at that ideal for every voter. The FiniteModelBC1C2Source prerequisite supplies the same geometry and unique-maximizer data; its Euclidean projection realizes Algorithm 1's approved projection reading and does not alter the displayed C1/C2 formula.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GKGMM19IterativeLocalVoting.finiteCoordinateDecomposableUtilityFamilyFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.finiteCoordinateDecomposableUtilityFamilyFormulaData
Source locator: cited publication, lines 582-594

Verbatim paper-source connection

Definition 3. Decomposable utilities. A voter utility functionPis decomposable if there
m=1 fv (x).
exists concave functions fvm for m ∈ {1 . . . M } such that fv (x) = M m m

If the utility functions for the voters are decomposable, then we can show that our
algorithm under L
∞ neighborhoods converges to the vector of medians of voters' ideal points
m
on each dimension. Suppose that hX is the marginal density function of the random variable
xm m m
v , and let x̄ be the set of medians of xv . (By set of medians, we mean the set of points
such that, on each dimension, the mass of voters with ideal points above and below.)

Lean-expanded paper semantic target

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [Fintype Coord] →
      [Nonempty Coord] → GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ) → (Coord → Voter → ℝ → ℝ) → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord] [Nonempty Coord]
    (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)) (componentUtility : Coord → Voter → ℝ → ℝ) =>
  (∀ (coordinate : Coord) (voter : Voter), ConcaveOn ℝ Set.univ (componentUtility coordinate voter)) ∧
    ∀ (voter : Voter) (x : Coord → ℝ),
      E.utility voter x = ∑ coordinate : Coord, componentUtility coordinate voter (x coordinate)
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment

Recorded source-to-declaration screening **Verdict:** matches

Reason: Definition 3 requires, for every voter and coordinate, a concave one-dimensional component and the equality f_v(x) = sum_m f_v^m(x_m). The Lean target quantifies exactly those component functions, concavity on their real domain, and the finite-coordinate sum. Its ambient-state quantifier agrees with the identity-bound approved raw-query-domain reading.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GKGMM19IterativeLocalVoting.finiteCoordinateIdealDistributionFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.finiteCoordinateIdealDistributionFormulaData
Source locator: cited publication, lines 449-452

Verbatim paper-source connection

```
C1 The solution space X ⊆ RM is non-empty, bounded, closed, and convex.
 C2 Each voter v has a unique ideal solution xv ∈ X .
 C3 The ideal point xv of each voter is drawn independently from a probability distribution
     with a bounded and measurable density function hX .
```

Lean-expanded paper semantic target

```
type:
{Coord : Type u_1} →
  [inst : Fintype Coord] → GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord → Prop

value:
fun {Coord : Type u_1} [Fintype Coord] (D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord) =>
  MeasureTheory.IsProbabilityMeasure D.idealMeasure ∧
    D.densityBound ≠ ⊤ ∧
      ∃ (density : (Coord → ℝ) → ENNReal),
        Measurable density ∧
          D.idealMeasure = MeasureTheory.volume.withDensity density ∧
            ∀ᵐ (ideal : Coord → ℝ), density ideal ≤ D.densityBound
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData

Recorded source-to-declaration screening **Verdict:** matches

Reason: Source C3 specifies a probability law for finite-dimensional ideal points with a bounded measurable density. The target requires a probability measure and a finite ENNReal bound, together with a measurable density whose withDensity measure is the ideal law and which is bounded almost everywhere relative to finite-dimensional Lebesgue measure. The displayed HasBoundedDensity and C3 carrier prerequisites preserve that formula.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

GKGMM19IterativeLocalVoting.finiteCoordinateLpNormedUtilitiesOnSolutionSpaceFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.finiteCoordinateLpNormedUtilitiesOnSolutionSpaceFormulaData
Source locator: cited publication, lines 474-475

Verbatim paper-source connection

Definition 1. Lp normed utilities. The voter utility function is Lp normed if fv (x) =
−kx − xv kp , ∀x ∈ X .

Lean-expanded paper semantic target

```
type:
{Voter : Type u_1} →
  {Coord : Type u_2} →
    [Fintype Coord] →
      [Nonempty Coord] →
        GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ) → GKGMM19IterativeLocalVoting.SourceNorm → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord] [Nonempty Coord]
    (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)) (p : GKGMM19IterativeLocalVoting.SourceNorm) =>
  GKGMM19IterativeLocalVoting.sourceNormFormulaData p ∧
    ∀ (voter : Voter),
      ∀ x ∈ E.solutionSpace,
        E.utility voter x = -GKGMM19IterativeLocalVoting.finiteCoordinateDistance p x (E.ideal voter)
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance
- GKGMM19IterativeLocalVoting.sourceNormFormulaData

Recorded source-to-declaration screening **Verdict:** matches

Reason: Source Definition 1 states f_v(x) = -||x-x_v||_p for every x in X. The target has exactly that voter/state quantification restricted to E.solutionSpace, with finiteCoordinateDistance expanding to the corresponding finite L1, L2, Linfinity, or positive-exponent Lp formula through sourceNormFormulaData.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GKGMM19IterativeLocalVoting.finiteCoordinateLpNormedUtilitiesRawQueryFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.finiteCoordinateLpNormedUtilitiesRawQueryFormulaData
Source locator: cited publication, lines 177-190

Verbatim paper-source connection

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution x0 ∈ X , tolerance ϵ > 0, an integer N , initial radius r0 > 0,
termination time T , norm q for local neighborhood.
Output: Solution x.

- For t ≥ 1, sample a voter vt ∈ V at random from the population; set rt = r0 /t
  and elicit

  x0t = arg maxx∈{s:ks−xt−1 kq ≤rt } fvt (x), (1)
  0
  and then compute xt = [xt]X , where [·]X is a projection onto space X ; i.e. ask
  the voter to move to her favorite point within constraints, and then project the
  reported point onto X .

[Next verbatim source excerpt]

Definition 1. Lp normed utilities. The voter utility function is Lp normed if fv (x) =
−kx − xv kp , ∀x ∈ X .

[Next verbatim source excerpt]

Definition 2. Weighted Euclidean utilities. Let the solution space X P be decomposable
into K different sub-spaces, so that x = (x1 , . . . , xK) for each x ∈ X (where K k
k=1 dim(x) =
M). Suppose that the utility function of the voter v is
K
X wvk
fv (x) = − kxk − xkv k2 .
kwv k2
k=1

where wv is a voter-specific weight vector, then the function is a Weighted Euclidean utility
function. We further assume that wv ∈ W ⊂ RK + and xv are independently drawn for each
voter v from a joint probability distribution with a bounded and measurable density function,
with W nonempty, bounded, closed, and convex.
This utility function can be interpreted as follows: the decision-making problem is de-
k
composable into K sub-problems, and each voter v has an ideal point xv and a weight wv

[Next verbatim source excerpt]

Next consider the general class of decomposable utilities, motivated by the fact that the
algorithm with L
∞ neighborhoods is of special interest since they are easy for humans to

understand: one can change each dimension up to a certain amount, independent of the
others.

Definition 3. Decomposable utilities. A voter utility functionPis decomposable if there
m=1 fv (x).
exists concave functions fvm for m ∈ {1 . . . M } such that fv (x) = M m m

Lean-expanded paper semantic target

type:
{Voter : Type u_1} →
{Coord : Type u_2} →
[Fintype Coord] →
[Nonempty Coord] →
GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ) → GKGMM19IterativeLocalVoting.SourceNorm → Prop

value:
fun {Voter : Type u_1} {Coord : Type u_2} [Fintype Coord] [Nonempty Coord]
(E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)) (p : GKGMM19IterativeLocalVoting.SourceNorm) =>
GKGMM19IterativeLocalVoting.sourceNormFormulaData p ∧
∀ (voter : Voter) (x : Coord → ℝ),
E.utility voter x = -GKGMM19IterativeLocalVoting.finiteCoordinateDistance p x (E.ideal voter)

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance
- GKGMM19IterativeLocalVoting.sourceNormFormulaData

Recorded source-to-declaration screening **Verdict:** matches

Reason: The approved raw-query-domain record explicitly extends Definition 1's displayed utility equality from X to every ambient candidate queried before projection. The Lean target does exactly this by dropping only the solution-space membership premise while retaining the same admissible source norm and negative finite-coordinate distance to the ideal.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

GKGMM19IterativeLocalVoting.paper_definition4_dlcd_formula

Paper-local declaration:
GKGMM19IterativeLocalVoting.paper_definition4_dlcd_formula
Source locator: cited publication, lines 688-716

Verbatim paper-source connection

Rather, we consider an unconstrained budget allocation problem, one in which a voter's utility includes a term for the budget deficit. Let $E \subseteq \{1 \ldots M\}$, $I = \{1 \ldots M\} \setminus E$ be the expenditure and income items, respectively. Then the general budget utility function is $f_v(x) = g_v(x) - d(\sum_{e \in E} x_e - \sum_{i \in I} x_i)$, where d is an increasing function on the deficit.
For example, suppose a voter's disutility was proportional to the square of the budget deficit (she especially dislikes large budget deficits); then, this term adds complex dependencies between the budget items. In general, nothing is known about convergence of Algorithm 1 with such utilities, as the deficit term may add complex dependencies between the dimensions. However, if the voter utility functions are decomposable across the dimensions and L_∞ neighborhoods used, then the results of Section 3.2 can be applied. We propose the following class of decomposable utility functions for the budgeting problem, achieved by assuming that the cost for the deficit is linear, and call the class "decomposable with a linear cost for deficit," or DLCD.

Definition 4. Let $f_v(x)$ be DLCD if

$$f_v(x) = \sum_{m=1}^{M} f_{vm}(x_m) - w_v \left(\sum_{e \in E} x_e - \sum_{i \in I} x_i \right),$$

where f_{vm} is a concave function for each m and $w_v \in R_+$.

Lean-expanded paper semantic target

```
type:
{Dim : Type u_1} → [Fintype Dim] → ((Dim → ℝ) → ℝ) → (Dim → Bool) → (Dim → ℝ → ℝ) → ℝ → Prop

value:
fun {Dim : Type u_1} [Fintype Dim] (utility : (Dim → ℝ) → ℝ) (isExpense : Dim → Bool) (componentUtility : Dim → ℝ → ℝ)
    (deficitWeight : ℝ) =>
  0 ≤ deficitWeight ∧
    (∀ (m : Dim), ConcaveOn ℝ Set.univ (componentUtility m)) ∧
      ∀ (x : Dim → ℝ),
        utility x =
          ∑ m : Dim, componentUtility m (x m) -
            deficitWeight *
              ((∑ m : Dim, if isExpense m = true then x m else 0) - ∑ m : Dim, if isExpense m = true then 0 else x m)
```

Recorded source-to-declaration screening **Verdict:** matches

Reason: Definition 4 is exactly a sum of concave coordinate utilities minus a nonnegative scalar weight times total expenditure less total income. The Boolean expense partition in Lean is exhaustive, the two finite sums reproduce the source's E/I sums, and 0 <= deficitWeight matches the approved reading that R+ includes the zero-cost boundary.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

GKGMM19IterativeLocalVoting.theorem3_population_directional_field_model_formula

Paper-local declaration:

GKGMM19IterativeLocalVoting.theorem3_population_directional_field_model_formula
Source locator: cited publication, lines 118-194

Verbatim paper-source connection

More formally, consider a M -dimensional societal decision problem in X ⊂ RM and a population of voters V , where each voter v ∈ V has bounded utility fv (x) ∈ R, ∀ x ∈ X . Then we consider the class of algorithms described in Algorithm 1. We study the algorithm class under two plausible models of how voters respond to query (1), which asks for the voter's favorite point in a local region.

- Model A: One possibility is that voters exactly perform the maximization asked of them, responding with their favorite point in the given L
q norm constraint set. In other
words, they return a point arg maxx∈{s:ks−x fvt (x). Note that by definition of this
t−1 kq ≤rt }
movement, the algorithm is myopically incentive compatible : if a voter is the last voter and no projections are used, then truthfully performing this movement is the dominant strategy. In general, the mechanism is not globally incentive compatible, nor incentive compatible with projections onto the feasible region. Simple examples of manipulations in both instances exist.

- Model B: On the other hand, voters may not actually search within the constraint set to find their favorite point inside of it. Rather, a voter v may have an idea about how to best improve the current point and then move in that direction to the boundary of the given constraint set. This model leads to a voter moving the current solution in the
gt
direction of the gradient of her utility function, returning a point xt−1 + rt
kgt kq , for some
gt ∈ ∂fvt (xt−1). Note that ∂f (x) denotes the set of subgradients of a function f at x, i.e. g ∈ ∂f (x) if ∀y , f (y) − f (x) ≥ g T (y − x).

ILV is directly inspired by the stochastic approximation approach to solve optimization problems (Robbins & Monro, 1951), especially stochastic gradient descent (SGD) and the stochastic subgradient method (SSGM). The idea is that if (a) voter preferences are drawn from some probability distribution and (b) the response of a voter to the query (1) moves the solution approximately in the direction of her utility gradient, then this procedure almost implements stochastic gradient descent for minimizing negative expected utility.

The caveat is that although the procedure can potentially obtain the direction of the gradient of the voter utilities, it cannot in general obtain any information about its magnitude since the movement norm is chosen by the procedure itself. However, we show that for certain plausible utility and voter response models, the algorithm does indeed converge to a unique point with desirable properties, including cases in which it converges to the societal optimum.

Note that with such feedback and without any additional assumptions on voter preferences (e.g. that voter utilities are normalized to the same scale), no algorithm has any hope of finding a desirable solution that depends on the cardinal values of voters' utilities, e.g., the social welfare maximizing solution (the solution that maximizes the sum of agent utilities). This is because an algorithm that uses only ordinal information about voter preferences is insensitive to any scaling or even monotonic transformations of those preferences.

317

[form-feed in source extraction]
Garg, Kamble, Goel, Marn, & Munagala

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution x0 ∈ X , tolerance ϵ > 0, an integer N , initial radius r0 > 0, termination time T , norm q for local neighborhood.
Output: Solution x.

- For t ≥ 1, sample a voter vt ∈ V at random from the population; set rt = r0 /t and elicit

x0t = arg maxx∈{s:ks−xt−1 kq ≤rt } fvt (x), (1)
0
and then compute xt = [xt]X , where [·]X is a projection onto space X ; i.e. ask the voter to move to her favorite point within constraints, and then project the reported point onto X .

- Stop when either t = T , in which case return xT , or when maxl, m∈{t−N,...,t} |xl − xm | ≤ ϵ, in which case return x = xt .

[Next verbatim source excerpt]

C1 The solution space X ⊆ RM is non-empty, bounded, closed, and convex.
C2 Each voter v has a unique ideal solution xv ∈ X .
C3 The ideal point xv of each voter is drawn independently from a probability distribution with a bounded and measurable density function hX .

Under this model, for a solution x ∈ X , the societal utility is given by Ev [fv (x)]. and the social optimal (SO) solution is any x
∗ ∈ arg max
x∈X Ev [fv (x)].
"Convergence" of ILV refers to the convergence of the sequence of random variables {xt }t≥1 to some x ∈ X with probability 1, assuming that the algorithm is allowed to run indefinitely (this notion of convergence also implies the termination of the algorithm with probability 1).
In the following subsections, we present several classes of utility functions for which the algorithm converges, summarized in Table 1. We further formalize the relationship to directional equilibria in Section 3.3.

[Next verbatim source excerpt]

Theorem 3. Suppose that C1 , C2 , and C3 are satisfied, and let G(x) , Ev k∇f ∇fv (x)
v (x)k2
.
Suppose, G(x) is uniformly continuous, L movement norm constraints are used, and voters
2

move according to Model B. If a trajectory {x}∞ t=1 of the algorithm converges to x , i.e.
∗

xt → x , then x is a directional equilibrium, i.e. G(x) = 0.
∗ ∗ ∗

Settled source reading When the displayed Model B gradient is zero, the voter remains at the current point.

Settled source reading The checked Theorem 3 zero-field result uses an explicit full-space condition; the constrained-space result gives zero field or no feasible aggregate direction. The printed theorem invokes C1–C3, including bounded convex X; these results alone do not prove or refute zero field on every such X.

Lean-elaborated paper signature

```
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Voter] [inst_1 : Fintype Coord]
  [inst_2 : Nonempty Coord] (E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)),
  Nonempty (GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E) ↔
    ∃ (population : MeasureTheory.Measure Voter),
      MeasureTheory.IsProbabilityMeasure population ∧
        ∃ (utilityGradient : Voter → (Coord → ℝ) → Coord → ℝ),
          E.utilityGradient = utilityGradient ∧
            (∀ (voter : Voter) (x : Coord → ℝ),
                GKGMM19IterativeLocalVoting.FiniteSourceSubgradientAt (E.utility voter) x (utilityGradient voter x)) ∧
              E.scalarDirection = GKGMM19IterativeLocalVoting.finiteScalarDirection ∧
                (E.zeroDirection = fun (x : Coord) => 0) ∧
                  (∀ (g : Coord → ℝ),
                      E.normDistance GKGMM19IterativeLocalVoting.SourceNorm.l2 g E.zeroDirection =
                        GKGMM19IterativeLocalVoting.finiteCoordinateNorm GKGMM19IterativeLocalVoting.SourceNorm.l2 g) ∧
                    (∀ (x : Coord → ℝ) (i : Coord),
                        MeasureTheory.Integrable
                          (fun (voter : Voter) =>
                            GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection
                              GKGMM19IterativeLocalVoting.SourceNorm.l2 (utilityGradient voter x) i)
                          population) ∧
                      E.directionalField =
                          GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField population utilityGradient ∧
                        ∀ (x : Coord → ℝ),
                          (E.voterExpectation fun (voter : Voter) =>
                              GKGMM19IterativeLocalVoting.theorem3NormalizedGradientDirection E voter x) =
                            GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField population utilityGradient x
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteSourceSubgradientAt
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateNorm
- GKGMM19IterativeLocalVoting.finiteScalarDirection
- GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection
- GKGMM19IterativeLocalVoting.populationTheorem3DirectionalField
- GKGMM19IterativeLocalVoting.theorem3NormalizedGradientDirection

Recorded source-to-declaration screening **Verdict:** matches

Reason: Theorem 3 defines G(x) as the population expectation of a voter-utility gradient or printed source subgradient normalized in L2. The target requires FiniteSourceSubgradientAt (E.utility voter) x (utilityGradient voter x) for every voter and x, then defines the coordinatewise integrable population field from precisely that function. The L2 normalization and approved zero-gradient zero contribution are preserved, so the field cannot be unrelated to voter utility.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

GKGMM19IterativeLocalVoting.theorem3_population_iid_trace_source_formula

Paper-local declaration:
GKGMM19IterativeLocalVoting.theorem3_population_iid_trace_source_formula
Source locator: cited publication, lines 118-194

Verbatim paper-source connection

More formally, consider a M -dimensional societal decision problem in X ⊂ RM and a population of voters V , where each voter v ∈ V has bounded utility fv (x) ∈ R, ∀ x ∈ X . Then we consider the class of algorithms described in Algorithm 1. We study the algorithm class under two plausible models of how voters respond to query (1), which asks for the voter's favorite point in a local region.

- Model A: One possibility is that voters exactly perform the maximization asked of them, responding with their favorite point in the given L
q norm constraint set. In other
words, they return a point arg maxx∈{s:ks−x fvt (x). Note that by definition of this
t−1 kq ≤rt }
movement, the algorithm is myopically incentive compatible : if a voter is the last voter and no projections are used, then truthfully performing this movement is the dominant strategy. In general, the mechanism is not globally incentive compatible, nor incentive compatible with projections onto the feasible region. Simple examples of manipulations in both instances exist.

- Model B: On the other hand, voters may not actually search within the constraint set to find their favorite point inside of it. Rather, a voter v may have an idea about how to best improve the current point and then move in that direction to the boundary of the given constraint set. This model leads to a voter moving the current solution in the
gt
direction of the gradient of her utility function, returning a point xt−1 + rt
kgt kq , for some
gt ∈ ∂fvt (xt−1). Note that ∂f (x) denotes the set of subgradients of a function f at x, i.e. g ∈ ∂f (x) if ∀y , f (y) − f (x) ≥ g T (y − x).

ILV is directly inspired by the stochastic approximation approach to solve optimization problems (Robbins & Monro, 1951), especially stochastic gradient descent (SGD) and the stochastic subgradient method (SSGM). The idea is that if (a) voter preferences are drawn from some probability distribution and (b) the response of a voter to the query (1) moves the solution approximately in the direction of her utility gradient, then this procedure almost implements stochastic gradient descent for minimizing negative expected utility.

The caveat is that although the procedure can potentially obtain the direction of the gradient of the voter utilities, it cannot in general obtain any information about its magnitude since the movement norm is chosen by the procedure itself. However, we show that for certain plausible utility and voter response models, the algorithm does indeed converge to a unique point with desirable properties, including cases in which it converges to the societal optimum.

Note that with such feedback and without any additional assumptions on voter preferences (e.g. that voter utilities are normalized to the same scale), no algorithm has any hope of finding a desirable solution that depends on the cardinal values of voters' utilities, e.g., the social welfare maximizing solution (the solution that maximizes the sum of agent utilities). This is because an algorithm that uses only ordinal information about voter preferences is insensitive to any scaling or even monotonic transformations of those preferences.

317

[form-feed in source extraction]
Garg, Kamble, Goel, Marn, & Munagala

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution x0 ∈ X , tolerance ϵ > 0, an integer N , initial radius r0 > 0, termination time T , norm q for local neighborhood.
Output: Solution x.

- For t ≥ 1, sample a voter vt ∈ V at random from the population; set rt = r0 /t and elicit

x0t = arg maxx∈{s:ks−xt−1 kq ≤rt } fvt (x), (1)
0
and then compute xt = [xt]X , where [·]X is a projection onto space X ; i.e. ask the voter to move to her favorite point within constraints, and then project the reported point onto X .

- Stop when either t = T , in which case return xT , or when maxl, m∈{t−N,...,t} |xl − xm | ≤ ϵ, in which case return x = xt .

[Next verbatim source excerpt]

C1 The solution space X ⊆ RM is non-empty, bounded, closed, and convex.
C2 Each voter v has a unique ideal solution xv ∈ X .
C3 The ideal point xv of each voter is drawn independently from a probability distribution with a bounded and measurable density function hX .

Under this model, for a solution $x \in X$, the societal utility is given by $E_v[f_v(x)]$. and the social optimal (SO) solution is any $x^* \in \arg\max_{x\in X} E_v[f_v(x)]$.

"Convergence" of ILV refers to the convergence of the sequence of random variables $\{x_t\}_{t\geq 1}$ to some $x \in X$ with probability 1, assuming that the algorithm is allowed to run indefinitely (this notion of convergence also implies the termination of the algorithm with probability 1).

In the following subsections, we present several classes of utility functions for which the algorithm converges, summarized in Table 1. We further formalize the relationship to directional equilibria in Section 3.3.

[Next verbatim source excerpt]

Theorem 3. Suppose that C1, C2, and C3 are satisfied, and let $G(x) \triangleq E_v\left[\frac{\nabla f_v(x)}{\|\nabla f_v(x)\|_2}\right]$.

Suppose, $G(x)$ is uniformly continuous, L_2 movement norm constraints are used, and voters move according to Model B. If a trajectory $\{x\}_{t=1}^{\infty}$ of the algorithm converges to x^*, i.e. $x_t \to x^*$, then x^* is a directional equilibrium, i.e. $G(x^*) = 0$.

Settled source reading When the displayed Model B gradient is zero, the voter remains at the current point.

Settled source reading The checked Theorem 3 zero-field result uses an explicit full-space condition; the constrained-space result gives zero field or no feasible aggregate direction. The printed theorem invokes C1–C3, including bounded convex X; these results alone do not prove or refute zero field on every such X.

Lean-elaborated paper signature

```
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E),
  Nonempty (GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M) ↔
    ∃ (initial : Coord → ℝ) (trajectory : ℕ → (ℕ → Voter) → Coord → ℝ) (state :
      (n : ℕ) → (Fin (n + 1) → Voter) → Coord → ℝ) (r0 : ℝ),
      0 < r0 ∧
        E.solutionSpace = Set.univ ∧
          GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldUniformContinuity M ∧
            (∀ (omega : ℕ → Voter), trajectory 0 omega = initial) ∧
              (∀ (n : ℕ) (omega : ℕ → Voter),
                  trajectory n omega = state n (AppliedModelingLib.iidSequencePastPrefix n omega)) ∧
                (∀ (n : ℕ) (omega : ℕ → Voter),
                    trajectory (n + 1) omega = fun (i : Coord) =>
                      trajectory n omega i +
                        GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1) *
                          GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection
                            GKGMM19IterativeLocalVoting.SourceNorm.l2
                            (M.utilityGradient (AppliedModelingLib.iidSequenceSample n omega) (trajectory n omega)) i) ∧
                  (∀ (a : Coord → ℝ) (n : ℕ),
                      MeasureTheory.Integrable
                        (fun (z : (Fin (n + 1) → Voter) × Voter) =>
                          GKGMM19IterativeLocalVoting.populationTheorem3PastCenteredTest M
                            (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)) a (state n z.1) z.2)
                        (MeasureTheory.Measure.map
                          (fun (omega : ℕ → Voter) =>
                            (AppliedModelingLib.iidSequencePastPrefix n omega,
                              AppliedModelingLib.iidSequenceSample n omega))
                          (AppliedModelingLib.iidSequenceMeasure M.population))) ∧
                    (∀ (a : Coord → ℝ),
                        MeasureTheory.StronglyAdapted AppliedModelingLib.iidSequenceNaturalFiltration
                          fun (n : ℕ) (omega : ℕ → Voter) =>
                          ∑ i ∈ Finset.range n,
                            GKGMM19IterativeLocalVoting.populationTheorem3CenteredIncrement M r0 trajectory
                              AppliedModelingLib.iidSequenceSample a i omega) ∧
                      ∀ (a : Coord → ℝ) (n : ℕ),
                        MeasureTheory.AEStronglyMeasurable
                          (GKGMM19IterativeLocalVoting.populationTheorem3CenteredIncrement M r0 trajectory
                            AppliedModelingLib.iidSequenceSample a n)
                          (AppliedModelingLib.iidSequenceMeasure M.population)
```

Direct retained dependencies

- AppliedModelingLib.iidSequenceMeasure
- AppliedModelingLib.iidSequenceNaturalFiltration
- AppliedModelingLib.iidSequencePastPrefix
- AppliedModelingLib.iidSequenceSample

- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel
- GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldUniformContinuity
- GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.modelBFiniteNormalizedDirection
- GKGMM19IterativeLocalVoting.populationTheorem3CenteredIncrement
- GKGMM19IterativeLocalVoting.populationTheorem3PastCenteredTest

Recorded source-to-declaration screening **Verdict:** matches
Reason: Under the approved full-space Theorem 3 reading, the target gives a positive r0, the exact r0/(n+1) Model B update, iid fresh voter omega(n+1), and a state determined by the preceding finite history. Its PopulationTheorem3DirectionalFieldModel prerequisite supplies the source subgradient relation for every voter and state. The additional integrability, adaptedness, and measurability clauses are exactly the approved iid-trace well-definedness conditions and assume no convergence, equilibrium, or martingale conclusion.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GKGMM19IterativeLocalVoting.utilityBoundedOnSolutionSpaceFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.utilityBoundedOnSolutionSpaceFormulaData
Source locator: cited publication, lines 100-124

Verbatim paper-source connection

local neighborhoods are defined – in this paper we focus on neighborhoods that are balls in
the L
q norm, and in particular on the cases where q = 1, 2 or ∞. (For M < ∞ dimensional

316

[form-feed in source extraction]
Iterative Local Voting for Collective Decision-making in Continuous Spaces

q norm kxk ,
p P
vectors, the L q
m |xm | . q = 1, 2 and ∞ neighborhoods correspond to
q
q
bounds on the sum of absolute values of the changes, the sum of the square of the changes,
and the maximum change, respectively.)

More formally, consider a M -dimensional societal decision problem in X ⊂ RM and a
population of voters V , where each voter v ∈ V has bounded utility fv (x) ∈ R, ∀ x ∈ X .
Then we consider the class of algorithms described in Algorithm 1. We study the algorithm
class under two plausible models of how voters respond to query (1), which asks for the
voter's favorite point in a local region.

[Next verbatim source excerpt]

Theorem 2. Suppose that conditions C1 , C2 , and C3 are satisfied, the voter utilities are
Lp normed, and voters respond to query (1) according to Model B. Then, ILV with Lq
neighborhoods converges to the societal optimal point w.p. 1 for any p > 0 and q > 0 such
that 1/p + 1/q = 1.

Lean-expanded paper semantic target

type:
{Voter : Type u_1} → {Point : Type u_2} → GKGMM19IterativeLocalVoting.ILVUtilityFormulaData Voter Point → Prop

value:
fun {Voter : Type u_1} {Point : Type u_2} (U : GKGMM19IterativeLocalVoting.ILVUtilityFormulaData Voter Point) =>
∀ (voter : Voter), ∃ (bound : ℝ), ∀ x ∈ U.solutionSpace, |U.utility voter x| ≤ bound

Direct retained dependencies

- GKGMM19IterativeLocalVoting.ILVUtilityFormulaData

Recorded source-to-declaration screening **Verdict:** matches

Reason: The source states that each voter v has bounded real-valued utility f_v(x) for every x in the feasible space X. The Lean target quantifies voter by voter, supplies a real bound on the absolute utility value, and restricts that bound to x in U.solutionSpace, exactly representing bounded voter utilities on X. The ILVUtilityFormulaData prerequisite exposes the identical clause and adds no conclusion.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleFormulaData

Paper-local declaration:
GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleFormulaData
Source locator: cited publication, lines 521-534

Verbatim paper-source connection

Definition 2. Weighted Euclidean utilities. Let the solution space X P be decomposable
into K different sub-spaces, so that x = (x1 , . . . , xK) for each x ∈ X (where K k
k=1 dim(x) =
M). Suppose that the utility function of the voter v is
K
X wvk
fv (x) = − kxk − xkv k2 .
kwv k2
k=1

where wv is a voter-specific weight vector, then the function is a Weighted Euclidean utility
function. We further assume that wv ∈ W ⊂ RK + and xv are independently drawn for each
voter v from a joint probability distribution with a bounded and measurable density function,
with W nonempty, bounded, closed, and convex.

Lean-expanded paper semantic target

```
type:
{Coord : Type u_1} →
  {Component : Type u_2} →
    [inst : Fintype Coord] →
      [inst_1 : Fintype Component] → GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component → Prop

value:
fun {Coord : Type u_1} {Component : Type u_2} [Fintype Coord] [Fintype Component]
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component) =>
  MeasureTheory.IsProbabilityMeasure D.jointMeasure ∧
    D.densityBound ≠ ⊤ ∧
      (∃ (density : (Component → ℝ) × (Coord → ℝ) → ENNReal),
          Measurable density ∧
            D.jointMeasure = (MeasureTheory.volume.prod MeasureTheory.volume).withDensity density ∧
              ∀ᵐ (sample : (Component → ℝ) × (Coord → ℝ)) ∂MeasureTheory.volume.prod MeasureTheory.volume,
                density sample ≤ D.densityBound) ∧
        MeasureTheory.Measure.map Prod.snd D.jointMeasure = D.idealDistribution.idealMeasure ∧
          D.weightSet.Nonempty ∧
            Bornology.IsBounded D.weightSet ∧
              IsClosed D.weightSet ∧
                Convex ℝ D.weightSet ∧
                  (∀ weight ∈ D.weightSet, ∀ (k : Component), 0 ≤ weight k) ∧
                    (∀ᵐ (sample : (Component → ℝ) × (Coord → ℝ)) ∂D.jointMeasure, sample.1 ∈ D.weightSet) ∧
                      (∀ᵐ (sample : (Component → ℝ) × (Coord → ℝ)) ∂D.jointMeasure, sample.1 ≠ 0) ∧
                        (∀ (k : Component), (D.componentBlock k).Nonempty) ∧
                          (∀ (k l : Component), k ≠ l → Disjoint (D.componentBlock k) (D.componentBlock l)) ∧
                            (∀ (i : Coord), ∃ (k : Component), i ∈ D.componentBlock k) ∧
                              ∀ (sample : (Component → ℝ) × (Coord → ℝ)) (state : Coord → ℝ),
                                GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleUtility D sample state =
                                  -∑ k : Component,
                                      D.sampleWeight sample k / D.sampleWeightNorm2 sample *
                                        GKGMM19IterativeLocalVoting.finiteCoordinateBlockL2Distance (D.componentBlock k)
                                          state (D.sampleIdeal sample k)
```

Direct retained dependencies

- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleIdeal
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeight
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData.sampleWeightNorm2
- GKGMM19IterativeLocalVoting.finiteCoordinateBlockL2Distance
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleUtility

Recorded source-to-declaration screening **Verdict:** matches

Reason: Definition 2's joint ideal/weight population, bounded measurable joint density, nonempty bounded closed convex nonnegative weight set, coordinate-block decomposition, and normalized weighted sum of Euclidean block distances all appear in the target. The ideal marginal and almost-everywhere nonzero weight enforce the source distribution and the approved nonzero-vector denominator convention, while allowing zero individual coordinates; the utility equality is available for all ambient query states as approved.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Appendix and supplement source claims

Review row 1

Source locator: cited publication, lines 2158-2160

Verbatim source input

Lemma 3. ∀ (p, q) s.t. p > 0, q > 0, and 1/p + 1/q = 1, k∇kx − xv kp kq = 1, ∀ x s.t.
v for any m.
xm 6= xm

[Next verbatim source excerpt]

Lemma 3 ∀ (p, q) s.t. p > 0, q > 0, 1/p+1/q = 1, k∇kx−xv kp kq = 1, ∀ x s.t. ∀ m, xm 6= xm
v .

Proof. If xm 6= xm
v ,∀m :
!1/p
X
∇m kx − xv kp = ∇m |xm − xm
v |
p

m
1 ∇m |xm − xmv |
p
= P
p(|xm − xm p 1−1/p
m v |)
|xm − xm
v |
p−1 (∇ |xm − xm |)
m v
=
kx − xv kp−1
p

Then

|xm − xm
v |
p−1 (∇ |xm − xm |)
m v
k∇kx − xv kp kq = k kq
kx − xv kp−1
p
!1/q
1 X q
m
= |x − xm
v |
p−1
(∇m |xm − xm
v |)
kx − xv kp−1
p m
!1/q
1 X
m
= |x − xm
v |
(p−1)q
kx − xv kp−1
p m
1 p/q
= p−1 kx − xv kp (p − 1)q = p
kx − xv kp
=1

Expanded PaperInterface specification

∀ {Coord : Type u_1} [inst : Fintype Coord] [inst_1 : Nonempty Coord] {p q : ℝ},
GKGMM19IterativeLocalVoting.HolderDualFinite p q →
∀ {x ideal : Coord → ℝ},
(∀ (i : Coord), x i ≠ ideal i) →
GKGMM19IterativeLocalVoting.finiteCoordinateNorm (GKGMM19IterativeLocalVoting.SourceNorm.lp q)
(GKGMM19IterativeLocalVoting.lpCostGradientCandidate p fun (i : Coord) => x i - ideal i) =
1

Retained governing declarations

- GKGMM19IterativeLocalVoting.HolderDualFinite
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateNorm
- GKGMM19IterativeLocalVoting.lpCostGradientCandidate

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Fintype Coord
3. assumption: Nonempty Coord
4. parameter: ℝ
5. parameter: ℝ
6. assumption: GKGMM19IterativeLocalVoting.HolderDualFinite p q
7. parameter: Coord → ℝ
8. parameter: Coord → ℝ
9. assumption: ∀ (i : Coord), x i ≠ ideal i
10. conclusion: GKGMM19IterativeLocalVoting.finiteCoordinateNorm (GKGMM19IterativeLocalVoting.SourceNorm.lp q)
    (GKGMM19IterativeLocalVoting.lpCostGradientCandidate p fun (i : Coord) => x i - ideal i) =
  1

Lean proof endpoint (built separately)

GKGMM19IterativeLocalVoting.lemma3_finite_holder_dual_gradient_candidate_norm_formula

Recorded source-to-Spec screening **Verdict:** matches

Reason: For positive finite Holder-dual p and q and coordinatewise noncollision, the expanded Spec computes the source gradient candidate for the p-norm distance and states that its q norm is exactly one, which is the complete Lemma 3 claim.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Main-text source claims

Review row 2

Source locator: cited publication, lines 556-559

Verbatim source input

Proposition 1. Suppose that conditions C1 , C2 , and C3 are satisfied, the voter utilities are Weighted Euclidean, and voters correctly respond to query (1) according to either Model A or Model B. Then, ILV with L2 neighborhoods converges with probability 1 to the societal optimal point.

[Next verbatim source excerpt]

More formally, consider a M -dimensional societal decision problem in X ⊂ RM and a population of voters V , where each voter v ∈ V has bounded utility fv (x) ∈ R, ∀ x ∈ X . Then we consider the class of algorithms described in Algorithm 1. We study the algorithm class under two plausible models of how voters respond to query (1), which asks for the voter's favorite point in a local region.

- Model A: One possibility is that voters exactly perform the maximization asked of them, responding with their favorite point in the given L
q norm constraint set. In other
words, they return a point arg maxx∈{s:ks−x fvt (x). Note that by definition of this
t−1 kq ≤rt }
movement, the algorithm is myopically incentive compatible : if a voter is the last voter and no projections are used, then truthfully performing this movement is the dominant strategy. In general, the mechanism is not globally incentive compatible, nor incentive compatible with projections onto the feasible region. Simple examples of manipulations in both instances exist.

- Model B: On the other hand, voters may not actually search within the constraint set to find their favorite point inside of it. Rather, a voter v may have an idea about how to best improve the current point and then move in that direction to the boundary of the given constraint set. This model leads to a voter moving the current solution in the
gt
direction of the gradient of her utility function, returning a point xt−1 + rt
kgt kq , for some
gt ∈ ∂fvt (xt−1). Note that ∂f (x) denotes the set of subgradients of a function f at x, i.e. g ∈ ∂f (x) if ∀y , f (y) − f (x) ≥ g T (y − x).

ILV is directly inspired by the stochastic approximation approach to solve optimization problems (Robbins & Monro, 1951), especially stochastic gradient descent (SGD) and the stochastic subgradient method (SSGM). The idea is that if (a) voter preferences are drawn from some probability distribution and (b) the response of a voter to the query (1) moves the solution approximately in the direction of her utility gradient, then this procedure almost implements stochastic gradient descent for minimizing negative expected utility.

The caveat is that although the procedure can potentially obtain the direction of the gradient of the voter utilities, it cannot in general obtain any information about its magni-tude since the movement norm is chosen by the procedure itself. However, we show that for certain plausible utility and voter response models, the algorithm does indeed converge to a unique point with desirable properties, including cases in which it converges to the societal optimum.

Note that with such feedback and without any additional assumptions on voter prefer-ences (e.g. that voter utilities are normalized to the same scale), no algorithm has any hope of finding a desirable solution that depends on the cardinal values of voters' utilities, e.g., the social welfare maximizing solution (the solution that maximizes the sum of agent utilities). This is because an algorithm that uses only ordinal information about voter preferences is insensitive to any scaling or even monotonic transformations of those preferences.

317

[form-feed in source extraction]
Garg, Kamble, Goel, Marn, & Munagala

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution x0 ∈ X , tolerance ϵ > 0, an integer N , initial radius r0 > 0, termination time T , norm q for local neighborhood.
Output: Solution x.

- For t ≥ 1, sample a voter vt ∈ V at random from the population; set rt = r0 /t and elicit

x0t = arg maxx∈{s:ks−xt−1 kq ≤rt } fvt (x), (1)
0
and then compute xt = [xt]X , where [·]X is a projection onto space X ; i.e. ask the voter to move to her favorite point within constraints, and then project the reported point onto X .

- Stop when either t = T , in which case return xT , or when maxl, m∈{t−N,...,t} |xl − xm | ≤ ϵ, in which case return x = xt .

[Next verbatim source excerpt]

C1 The solution space X ⊆ RM is non-empty, bounded, closed, and convex.
C2 Each voter v has a unique ideal solution xv ∈ X .
C3 The ideal point xv of each voter is drawn independently from a probability distribution with a bounded and measurable density function hX .

Under this model, for a solution x ∈ X , the societal utility is given by Ev [fv (x)]. and the social optimal (SO) solution is any x
∗ ∈ arg max
x∈X Ev [fv (x)].

"Convergence" of ILV refers to the convergence of the sequence of random variables {xt }t≥1 to some x ∈ X with probability 1, assuming that the algorithm is allowed to run indefinitely (this notion of convergence also implies the termination of the algorithm with probability 1).

In the following subsections, we present several classes of utility functions for which the algorithm converges, summarized in Table 1. We further formalize the relationship to directional equilibria in Section 3.3.

[Next verbatim source excerpt]

Definition 2. Weighted Euclidean utilities. Let the solution space X P be decomposable into K different sub-spaces, so that x = (x1 , . . . , xK) for each x ∈ X (where K k
k=1 dim(x) =
M). Suppose that the utility function of the voter v is

K
X wvk
fv (x) = − kxk − xkv k2 .
kwv k2
k=1

where wv is a voter-specific weight vector, then the function is a Weighted Euclidean utility function. We further assume that wv ∈ W ⊂ RK + and xv are independently drawn for each voter v from a joint probability distribution with a bounded and measurable density function, with W nonempty, bounded, closed, and convex.

This utility function can be interpreted as follows: the decision-making problem is de-
k
composable into K sub-problems, and each voter v has an ideal point xv and a weight wv
k

for each sub-problem k , so that the voter's disutility for a solution is the weighted sum of the Euclidean distances to the ideal points in each sub-problems. Such utility functions may emerge in facility location problems, for example, where voters have preferences on

323

[form-feed in source extraction]
Garg, Kamble, Goel, Marn, & Munagala

the locations of multiple facilities on a map. This utility form is also the one most closely related to the existing literature on Directional Equilibria and Quadratic Voting, in which preferences are linear. To recover the weighted linear preferences case, set K = M , with each sub-space of dimension 1. In this case, the following holds:

Expanded PaperInterface specification

```
(∀ {Voter : Type u_1} {Coord : Type u_2} {Component : Type u_3} [inst : Fintype Coord] [inst_1 : Fintype Component]
    [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
    {project : (Coord → ℝ) → Coord → ℝ}
    (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D.idealDistribution project)
    (A : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D) {r0 : ℝ},
    0 < r0 →
      ∀ initial ∈ E.solutionSpace,
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleSocialObjectiveSource E D →
          GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula S ∧
            GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAResponseSourceFormula D A ∧
              GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationMinimizerSet D E.solutionSpace =
                  E.socialOptimal ∧
                GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
                  (AppliedModelingLib.iidSequenceMeasure D.jointMeasure)
                  (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory A r0 project
                    initial)) ∧
  ∀ {Voter : Type u_4} {Coord : Type u_5} {Component : Type u_6} [inst : Fintype Coord] [inst_1 : Fintype Component]
    [Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
    {project : (Coord → ℝ) → Coord → ℝ}
    (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D.idealDistribution project) {r0 : ℝ},
    0 < r0 →
      ∀ initial ∈ E.solutionSpace,
        GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleSocialObjectiveSource E D →
          GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula S ∧
            GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationMinimizerSet D E.solutionSpace =
                E.socialOptimal ∧
              GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
                (AppliedModelingLib.iidSequenceMeasure D.jointMeasure)
                (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleOutcomeTrajectory D r0 project initial)
```

Retained governing declarations

- AppliedModelingLib.iidSequenceMeasure
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleSocialObjectiveSource
- GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAResponseSourceFormula
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleOutcomeTrajectory
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationMinimizerSet

Lean-elaborated claim roles

```
1. conclusion: (∀ {Voter : Type u_1} {Coord : Type u_2} {Component : Type u_3} [inst : Fintype Coord] [inst_1 : Fintype Component]
   [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
   (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
   {project : (Coord → ℝ) → Coord → ℝ}
   (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D.idealDistribution project)
   (A : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D) {r0 : ℝ},
   0 < r0 →
     ∀ initial ∈ E.solutionSpace,
       GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleSocialObjectiveSource E D →
         GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula S ∧
           GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAResponseSourceFormula D A ∧
             GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationMinimizerSet D E.solutionSpace =
                 E.socialOptimal ∧
               GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
                 (AppliedModelingLib.iidSequenceMeasure D.jointMeasure)
                 (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory A r0 project
                   initial)) ∧
  ∀ {Voter : Type u_4} {Coord : Type u_5} {Component : Type u_6} [inst : Fintype Coord] [inst_1 : Fintype Component]
   [Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
   (D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component)
   {project : (Coord → ℝ) → Coord → ℝ}
   (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D.idealDistribution project) {r0 : ℝ},
   0 < r0 →
     ∀ initial ∈ E.solutionSpace,
       GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleSocialObjectiveSource E D →
         GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula S ∧
           GKGMM19IterativeLocalVoting.weightedEuclideanJointSamplePopulationMinimizerSet D E.solutionSpace =
               E.socialOptimal ∧
             GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
               (AppliedModelingLib.iidSequenceMeasure D.jointMeasure)
               (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleOutcomeTrajectory D r0 project initial)
```

Lean proof endpoint (built separately)

```
GKGMM19IterativeLocalVoting.proposition1_weighted_euclidean_joint_c3
```

Recorded source-to-Spec screening **Verdict:** matches

Reason: Both Proposition 1 response branches are present on the same iid joint law of the full nonnegative, almost-surely nonzero weight vector and ideal point: an exact measurable Model-A maximizer followed by projection, and the normalized weighted-Euclidean Model-B trajectory. The displayed C1-C3 geometry and density data, approved raw-query utility domain, normalized block-distance cost, and social-objective identification restrict both branches to Definition 2, and each concludes almost-sure convergence to the societal-optimal set under L2 neighborhoods.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 3

Source locator: cited publication, lines 587-600

Verbatim source input

If the utility functions for the voters are decomposable, then we can show that our algorithm under L_∞ neighborhoods converges to the vector of medians of voters' ideal points on each dimension. Suppose that h_X^m is the marginal density function of the random variable x_v^m, and let $\bar{x}^m$ be the set of medians of x_v. (By set of medians, we mean the set of points such that, on each dimension, the mass of voters with ideal points above and below.)

Proposition 2. Suppose that conditions C1 , C2 , and C3 are satisfied, the voter utilities are decomposable, and voters respond to query (1) according to either Model A or Model B. Then, ILV with L_∞ neighborhoods converges with probability 1 to a point in the set of medians $\bar{x}$.

[Next verbatim source excerpt]

More formally, consider a M-dimensional societal decision problem in $X \subset R^M$ and a population of voters V, where each voter $v \in V$ has bounded utility $f_v(x) \in R$, $\forall x \in X$. Then we consider the class of algorithms described in Algorithm 1. We study the algorithm class under two plausible models of how voters respond to query (1), which asks for the voter's favorite point in a local region.

- Model A: One possibility is that voters exactly perform the maximization asked of them, responding with their favorite point in the given L_q norm constraint set. In other words, they return a point $\arg\max_{x \in \{s : \|s - x_{t-1}\|_q \le r_t\}} f_{v_t}(x)$. Note that by definition of this movement, the algorithm is myopically incentive compatible : if a voter is the last voter and no projections are used, then truthfully performing this movement is the dominant strategy. In general, the mechanism is not globally incentive compatible, nor incentive compatible with projections onto the feasible region. Simple examples of manipulations in both instances exist.

- Model B: On the other hand, voters may not actually search within the constraint set to find their favorite point inside of it. Rather, a voter v may have an idea about how to best improve the current point and then move in that direction to the boundary of the given constraint set. This model leads to a voter moving the current solution in the direction of the gradient of her utility function, returning a point $x_{t-1} + r_t \frac{g_t}{\|g_t\|_q}$, for some $g_t \in \partial f_{v_t}(x_{t-1})$. Note that $\partial f(x)$ denotes the set of subgradients of a function f at x, i.e. $g \in \partial f(x)$ if $\forall y$, $f(y) - f(x) \ge g^T(y - x)$.

ILV is directly inspired by the stochastic approximation approach to solve optimization problems (Robbins & Monro, 1951), especially stochastic gradient descent (SGD) and the stochastic subgradient method (SSGM). The idea is that if (a) voter preferences are drawn from some probability distribution and (b) the response of a voter to the query (1) moves the solution approximately in the direction of her utility gradient, then this procedure almost implements stochastic gradient descent for minimizing negative expected utility.

The caveat is that although the procedure can potentially obtain the direction of the gradient of the voter utilities, it cannot in general obtain any information about its magnitude since the movement norm is chosen by the procedure itself. However, we show that for certain plausible utility and voter response models, the algorithm does indeed converge to a unique point with desirable properties, including cases in which it converges to the societal optimum.

Note that with such feedback and without any additional assumptions on voter preferences (e.g. that voter utilities are normalized to the same scale), no algorithm has any hope of finding a desirable solution that depends on the cardinal values of voters' utilities, e.g., the social welfare maximizing solution (the solution that maximizes the sum of agent utilities). This is because an algorithm that uses only ordinal information about voter preferences is insensitive to any scaling or even monotonic transformations of those preferences.

317

[form-feed in source extraction]

Garg, Kamble, Goel, Marn, & Munagala

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution $x_0 \in X$, tolerance $\epsilon > 0$, an integer N, initial radius $r_0 > 0$, termination time T, norm q for local neighborhood.
Output: Solution x.

- For $t \ge 1$, sample a voter $v_t \in V$ at random from the population; set $r_t = r_0/t$ and elicit

$$x_t^0 = \arg\max_{x \in \{s : \|s - x_{t-1}\|_q \le r_t\}} f_{v_t}(x), \qquad (1)$$

and then compute $x_t = [x_t]_X$, where $[\cdot]_X$ is a projection onto space X; i.e. ask the voter to move to her favorite point within constraints, and then project the reported point onto X.

- Stop when either $t = T$, in which case return x_T, or when $\max_{l, m \in \{t-N,\ldots,t\}} |x_l - x_m| \le \epsilon$, in which case return $x = x_t$.

[Next verbatim source excerpt]

C1 The solution space $X \subseteq \mathbb{R}^M$ is non-empty, bounded, closed, and convex.
C2 Each voter v has a unique ideal solution $x_v \in X$.
C3 The ideal point x_v of each voter is drawn independently from a probability distribution with a bounded and measurable density function h_X.

Under this model, for a solution $x \in X$, the societal utility is given by $E_v[f_v(x)]$. and the social optimal (SO) solution is any $x^* \in \arg\max_{x \in X} E_v[f_v(x)]$.

"Convergence" of ILV refers to the convergence of the sequence of random variables $\{x_t\}_{t \ge 1}$ to some $x \in X$ with probability 1, assuming that the algorithm is allowed to run indefinitely (this notion of convergence also implies the termination of the algorithm with probability 1).

In the following subsections, we present several classes of utility functions for which the algorithm converges, summarized in Table 1. We further formalize the relationship to directional equilibria in Section 3.3.

[Next verbatim source excerpt]

Next consider the general class of decomposable utilities, motivated by the fact that the algorithm with L_∞ neighborhoods is of special interest since they are easy for humans to understand: one can change each dimension up to a certain amount, independent of the others.

Definition 3. Decomposable utilities. A voter utility function is decomposable if there exists concave functions f_v^m for $m \in \{1 \ldots M\}$ such that $f_v(x) = \sum_{m=1}^{M} f_v^m(x^m)$.

[Next verbatim source excerpt]

Now, we sketch the proof for fully decomposable utility functions and L_∞ neighborhoods.

Proposition 2. Suppose that conditions C1, C2, and C3 are satisfied, the voter utilities are decomposable, and voters respond to query (1) according to either Model A or Model B. Then, ILV with L_∞ neighborhoods converges with probability 1 to a point in the set of medians $\bar{x}$.

Proof. Consider each dimension separately. If $x_{t-1}^m < x_v^m$, then the sampled voter increases x_{t-1}^m by r_t as long as $x_{t-1}^m + r_t \le x_v^m$. On the other hand if $x_{t-1}^m > x_v^m$, then the sampled voter decreases x_{t-1}^m by r_t as long as $x_{t-1}^m - r_t \ge x_v^m$. Thus except for when a voter's ideal solution is too close to the current point, the algorithm can be seen as performing SSGM on each dimension separately as if the utility function was L_1 (the absolute value) on each dimension. Thus a proof akin to that of Theorem 1 with $p = 1$, $q = \infty$ holds.

Settled source reading For Proposition 2 with a Linfinity neighborhood, Model B moves every non-tied decomposition coordinate by the full current radius toward that coordinate of the sampled voter's ideal; tied coordinates do not move.

Settled source reading The checked Proposition 2 assumes that replacing one coordinate of a feasible point by that of another feasible point preserves feasibility. This is stronger than C1's closed convex domain; it is the current statement scope, not a proved necessary condition for the conclusion.

Expanded PaperInterface specification

```
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E)
  (D : GKGMM19IterativeLocalVoting.DecomposableStructure Voter (Coord → ℝ) Coord),
  GKGMM19IterativeLocalVoting.IsDecomposableUtilitiesWith E D →
    D.coords = Finset.univ →
      (∀ (m : Coord) (x : Coord → ℝ), D.coordinate m x = x m) →
        GKGMM19IterativeLocalVoting.UsesFiniteCoordinateNormDistance E →
          (∀ {x y : Coord → ℝ},
            x ∈ E.solutionSpace →
              y ∈ E.solutionSpace →
                ∀ (m : Coord),
                  ∃ replacement ∈ E.solutionSpace, replacement m = y m ∧ ∀ (l : Coord), l ≠ m → replacement l = x l) →
            ∀ {project : (Coord → ℝ) → Coord → ℝ}
              (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
```

```
0 < r0 →
  ∀ initial ∈ E.solutionSpace,
    (∃ (sampledVoter : (Coord → ℝ) → Voter),
        (∀ᵐ (ideal : Coord → ℝ) ∂C3.data.idealMeasure, E.ideal (sampledVoter ideal) = ideal) ∧
          ∀ᵐ (omega :
            ℕ → Coord → ℝ) ∂GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data,
            ∀ (n : ℕ),
              GKGMM19IterativeLocalVoting.ModelARawResponseAt E
                GKGMM19IterativeLocalVoting.SourceNorm.linfty
                (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project
                  initial n omega)
                (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))
                (sampledVoter (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega))
                (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeRawResponse r0
                  (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project
                    initial)
                  GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)) →
      GKGMM19IterativeLocalVoting.finiteCoordinateC3CarrierFormula C3 ∧
        GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula S ∧
          (∃ (sampledVoter : (Coord → ℝ) → Voter),
              (∀ᵐ (ideal : Coord → ℝ) ∂C3.data.idealMeasure, E.ideal (sampledVoter ideal) = ideal) ∧
                ∀ᵐ (omega :
                  ℕ → Coord → ℝ) ∂GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data,
                  ∀ (n : ℕ),
                    GKGMM19IterativeLocalVoting.ModelARawResponseAt E
                      GKGMM19IterativeLocalVoting.SourceNorm.linfty
                      (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project
                        initial n omega)
                      (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))
                      (sampledVoter (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega))
                      (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeRawResponse r0
                        (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0
                          project initial)
                        GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)) ∧
            (∀ {center response : Coord → ℝ} {r : ℝ} {voter : Voter},
                GKGMM19IterativeLocalVoting.ModelAResponseAt E
                    GKGMM19IterativeLocalVoting.SourceNorm.linfty center r voter response →
                  ∀ (m : Coord),
                    IsMaxOn (fun (z : ℝ) => D.coordinateUtility m voter z)
                      {z : ℝ |
                        ∃
                          candidate ∈
                            GKGMM19IterativeLocalVoting.LocalNeighborhood E
                              GKGMM19IterativeLocalVoting.SourceNorm.linfty center r,
                          D.coordinate m candidate = z}
                      (D.coordinate m response)) ∧
              (∀ (n : ℕ) (omega : ℕ → Coord → ℝ),
                  GKGMM19IterativeLocalVoting.ModelBCoordinatewiseBoundaryResponseAt
                    (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 1 r0 project
                      initial n omega)
                    (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)
                    (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))
                    (GKGMM19IterativeLocalVoting.finiteModelBOutcomeRawResponse 1 r0
                      (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 1 r0 project
                        initial)
                      GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)) ∧
                AppliedModelingLib.Optimization.OutcomeIndexedConvergesToSet
                    (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
                    (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project
                      initial)
                    (GKGMM19IterativeLocalVoting.finiteModelBExpectedLpMinimizerSet C3.data 1
                      E.solutionSpace) ∧
                  AppliedModelingLib.Optimization.OutcomeIndexedConvergesToSet
                    (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
                    (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 1 r0 project
                      initial)
                    (GKGMM19IterativeLocalVoting.finiteModelBExpectedLpMinimizerSet C3.data 1
                      E.solutionSpace)
```

Retained governing declarations

- AppliedModelingLib.Optimization.OutcomeIndexedConvergesToSet
- GKGMM19IterativeLocalVoting.DecomposableStructure
- GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier
- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.IsDecomposableUtilitiesWith
- GKGMM19IterativeLocalVoting.LocalNeighborhood
- GKGMM19IterativeLocalVoting.ModelARawResponseAt
- GKGMM19IterativeLocalVoting.ModelAResponseAt
- GKGMM19IterativeLocalVoting.ModelBCoordinatewiseBoundaryResponseAt

- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.UsesFiniteCoordinateNormDistance
- GKGMM19IterativeLocalVoting.finiteCoordinateC3CarrierFormula
- GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeRawResponse
- GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory
- GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula
- GKGMM19IterativeLocalVoting.finiteModelBExpectedLpMinimizerSet
- GKGMM19IterativeLocalVoting.finiteModelBIdealSample
- GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
- GKGMM19IterativeLocalVoting.finiteModelBOutcomeRawResponse
- GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory
- GKGMM19IterativeLocalVoting.ilvRadius

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Fintype Coord
4. assumption: Nonempty Coord
5. parameter: GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)
6. parameter: GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E
7. parameter: GKGMM19IterativeLocalVoting.DecomposableStructure Voter (Coord → ℝ) Coord
8. assumption: GKGMM19IterativeLocalVoting.IsDecomposableUtilitiesWith E D
9. assumption: D.coords = Finset.univ
10. assumption: ∀ (m : Coord) (x : Coord → ℝ), D.coordinate m x = x m
11. assumption: GKGMM19IterativeLocalVoting.UsesFiniteCoordinateNormDistance E
12. assumption: ∀ {x y : Coord → ℝ},
  x ∈ E.solutionSpace →
    y ∈ E.solutionSpace →
      ∀ (m : Coord), ∃ replacement ∈ E.solutionSpace, replacement m = y m ∧ ∀ (l : Coord), l ≠ m → replacement l = x l
13. parameter: (Coord → ℝ) → Coord → ℝ
14. assumption: GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project
15. parameter: ℝ
16. assumption: 0 < r0
17. parameter: Coord → ℝ
18. assumption: initial ∈ E.solutionSpace
19. assumption: ∃ (sampledVoter : (Coord → ℝ) → Voter),
  (∀ᵐ (ideal : Coord → ℝ) ∂C3.data.idealMeasure, E.ideal (sampledVoter ideal) = ideal) ∧
    ∀ᵐ (omega : ℕ → Coord → ℝ) ∂GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data,
      ∀ (n : ℕ),
        GKGMM19IterativeLocalVoting.ModelARawResponseAt E GKGMM19IterativeLocalVoting.SourceNorm.linfty
          (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project initial n omega)
          (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))
          (sampledVoter (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega))
          (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeRawResponse r0
            (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project initial)
            GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)
20. conclusion: GKGMM19IterativeLocalVoting.finiteCoordinateC3CarrierFormula C3 ∧
  GKGMM19IterativeLocalVoting.finiteModelBC1C2SourceFormula S ∧
    (∃ (sampledVoter : (Coord → ℝ) → Voter),
        (∀ᵐ (ideal : Coord → ℝ) ∂C3.data.idealMeasure, E.ideal (sampledVoter ideal) = ideal) ∧
          ∀ᵐ (omega : ℕ → Coord → ℝ) ∂GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data,
            ∀ (n : ℕ),
              GKGMM19IterativeLocalVoting.ModelARawResponseAt E GKGMM19IterativeLocalVoting.SourceNorm.linfty
                (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project initial n omega)
                (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))
                (sampledVoter (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega))
                (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeRawResponse r0
                  (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project initial)
                  GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)) ∧
      (∀ {center response : Coord → ℝ} {r : ℝ} {voter : Voter},
          GKGMM19IterativeLocalVoting.ModelAResponseAt E GKGMM19IterativeLocalVoting.SourceNorm.linfty center r voter
              response →
            ∀ (m : Coord),
              IsMaxOn (fun (z : ℝ) => D.coordinateUtility m voter z)
                {z : ℝ |
                  ∃
                    candidate ∈
                      GKGMM19IterativeLocalVoting.LocalNeighborhood E GKGMM19IterativeLocalVoting.SourceNorm.linfty
                        center r,
                    D.coordinate m candidate = z}
                (D.coordinate m response)) ∧
        (∀ (n : ℕ) (omega : ℕ → Coord → ℝ),
            GKGMM19IterativeLocalVoting.ModelBCoordinatewiseBoundaryResponseAt
              (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 1 r0 project initial n omega)
              (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)
              (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))
              (GKGMM19IterativeLocalVoting.finiteModelBOutcomeRawResponse 1 r0
                (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 1 r0 project initial)
                GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)) ∧
          AppliedModelingLib.Optimization.OutcomeIndexedConvergesToSet
```

```
  (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
  (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project initial)
  (GKGMM19IterativeLocalVoting.finiteModelBExpectedLpMinimizerSet C3.data 1 E.solutionSpace) ∧
AppliedModelingLib.Optimization.OutcomeIndexedConvergesToSet
  (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
  (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 1 r0 project initial)
  (GKGMM19IterativeLocalVoting.finiteModelBExpectedLpMinimizerSet C3.data 1 E.solutionSpace)
```

Lean proof endpoint (built separately)

```
GKGMM19IterativeLocalVoting.proposition2_coordinatewise_boundary_finite_c3_median
```

Recorded source-to-Spec screening **Verdict:** matches

Reason: The target quantifies decomposable coordinate utilities with concave coordinate components and applies the approved ambient raw-query reading. It records the approved product-coordinate feasible geometry, ties the canonical clamp trajectory to exact Model-A responses, and uses the approved Proposition 2 Model-B rule that every non-tied coordinate moves the full Linfinity radius toward the sampled ideal while ties stay fixed. Both projected iid traces converge almost surely to the feasible minimizers of expected coordinatewise L1 distance, the displayed representation of the coordinatewise median set approved for this row.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Appendix and supplement source claims (continued)

Review row 4

Source locator: cited publication, lines 2082-2090

Verbatim source input

Lemma 1. For q ∈ {1, 2, ∞}, there exists K2 ∈ R+ s.t. kg̃v (x) − gt k2 ≤ K2 , ∀ gt ∈ ∂fv (x)
for any v and x.

The lemma bounds the error in the movement direction from the gradient direction, by
noting that both the movement direction and the gradient direction have bounded norms.

We also need the following lemma, which is proved separately for each case in the following
sections.

Expanded PaperInterface specification

```
(∀ {Coord : Type u_1} [inst : Fintype Coord] [inst_1 : Nonempty Coord] {center ideal : Coord → ℝ} {r : ℝ}
  (g : Coord → ℝ),
  0 < r →
   GKGMM19IterativeLocalVoting.FiniteSubgradientAt
     (fun (y : Coord → ℝ) => AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) => y i - ideal i) center
     g →
    (AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) =>
      GKGMM19IterativeLocalVoting.l2BallDirection center ideal r i - g i) ≤
     2 * ↑(Fintype.card Coord)) ∧
 (∀ {Coord : Type u_2} [inst : Fintype Coord] [Nonempty Coord] {center ideal : Coord → ℝ} {r : ℝ} (g : Coord → ℝ),
   0 < r →
    GKGMM19IterativeLocalVoting.FiniteSubgradientAt
      (fun (y : Coord → ℝ) => AppliedModelingLib.FiniteDimensionalNorms.l1 fun (i : Coord) => y i - ideal i)
      center g →
     (AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) =>
       GKGMM19IterativeLocalVoting.l1LinfBallDirection center ideal r i - g i) ≤
      2 * ↑(Fintype.card Coord)) ∧
  ∀ {Coord : Type u_3} [inst : Fintype Coord] [inst_1 : Nonempty Coord] {center ideal raw : Coord → ℝ} {r : ℝ}
   (g : Coord → ℝ),
   0 < r →
    GKGMM19IterativeLocalVoting.finiteCoordinateDistance GKGMM19IterativeLocalVoting.SourceNorm.l1 raw center ≤ r →
     GKGMM19IterativeLocalVoting.FiniteSubgradientAt
       (fun (y : Coord → ℝ) => AppliedModelingLib.FiniteDimensionalNorms.linf fun (i : Coord) => y i - ideal i)
       center g →
      (AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) =>
        GKGMM19IterativeLocalVoting.linfL1RawDirection center raw r i - g i) ≤
       2 * ↑(Fintype.card Coord)
```

Retained governing declarations

- AppliedModelingLib.FiniteDimensionalNorms.l1
- AppliedModelingLib.FiniteDimensionalNorms.l2
- AppliedModelingLib.FiniteDimensionalNorms.linf
- GKGMM19IterativeLocalVoting.FiniteSubgradientAt
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance
- GKGMM19IterativeLocalVoting.l1LinfBallDirection
- GKGMM19IterativeLocalVoting.l2BallDirection
- GKGMM19IterativeLocalVoting.linfL1RawDirection

Lean-elaborated claim roles

```
1. conclusion: (∀ {Coord : Type u_1} [inst : Fintype Coord] [inst_1 : Nonempty Coord] {center ideal : Coord → ℝ} {r : ℝ}
  (g : Coord → ℝ),
  0 < r →
   GKGMM19IterativeLocalVoting.FiniteSubgradientAt
     (fun (y : Coord → ℝ) => AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) => y i - ideal i) center
     g →
    (AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) =>
      GKGMM19IterativeLocalVoting.l2BallDirection center ideal r i - g i) ≤
     2 * ↑(Fintype.card Coord)) ∧
 (∀ {Coord : Type u_2} [inst : Fintype Coord] [Nonempty Coord] {center ideal : Coord → ℝ} {r : ℝ} (g : Coord → ℝ),
   0 < r →
    GKGMM19IterativeLocalVoting.FiniteSubgradientAt
      (fun (y : Coord → ℝ) => AppliedModelingLib.FiniteDimensionalNorms.l1 fun (i : Coord) => y i - ideal i)
      center g →
     (AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) =>
       GKGMM19IterativeLocalVoting.l1LinfBallDirection center ideal r i - g i) ≤
```

```
      2 * ↑(Fintype.card Coord)) ∧
  ∀ {Coord : Type u_3} [inst : Fintype Coord] [inst_1 : Nonempty Coord] {center ideal raw : Coord → ℝ} {r : ℝ}
    (g : Coord → ℝ),
    0 < r →
      GKGMM19IterativeLocalVoting.finiteCoordinateDistance GKGMM19IterativeLocalVoting.SourceNorm.l1 raw center ≤ r →
        GKGMM19IterativeLocalVoting.FiniteSubgradientAt
            (fun (y : Coord → ℝ) => AppliedModelingLib.FiniteDimensionalNorms.linf fun (i : Coord) => y i - ideal i)
            center g →
          (AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) =>
              GKGMM19IterativeLocalVoting.linfL1RawDirection center raw r i - g i) ≤
            2 * ↑(Fintype.card Coord)
```

Lean proof endpoint (built separately)

GKGMM19IterativeLocalVoting.appendix_lemma1_uniform_direction_error

Recorded source-to-Spec screening **Verdict:** matches

Reason: The three conjuncts instantiate exactly q = 2, infinity, and 1 for the L2, L1, and Linfinity sampled costs, respectively; each keeps the source's universal quantifier over every subgradient at every center and ideal, requires the positive query radius needed to form the movement direction, and supplies the uniform voter- and point-independent finite bound K2 = 2 * card(Coord) on the L2 direction error. The displayed direction and distance dependencies spell the actual raw displacement used in each case, so no source case or quantifier is lost.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 5

Source locator: cited publication, lines 2093-2104

Verbatim source input

Lemma 2. Suppose that fv (x) , kxv − xkp and define the function

At , I{g̃vt (xt) ∈
/ ∂fvt (xt)},

where g̃vt (xt) is as defined in (2). Then there exists C ∈ R s.t. ∀ n, P(At = 1|Ft) ≤ Crt ,
when (p = 2, q = 2), (p = 1, q = ∞), or (p = ∞, q = 1).

The lemma can be interpreted as follows: At indicates a `bad' event, when a voter may not
be providing a true subgradient of her utility function. However, the probability of the event
occurring vanishes with rt , which, as we will see below, is the right rate for the algorithm
to converge.

Expanded PaperInterface specification

```
(∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
  GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project →
   ∀ {r0 : ℝ},
    0 < r0 →
     ∀ (initial : Coord → ℝ) (A : ℕ → Set (ℕ → Coord → ℝ)),
      (∀ (n : ℕ), MeasurableSet (A n)) →
       (∀ (n : ℕ) (omega : ℕ → Coord → ℝ),
         omega ∈ A n ↔
          ¬GKGMM19IterativeLocalVoting.FiniteSubgradientAt
            (fun (y : Coord → ℝ) =>
             AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) =>
              y i - GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega i)
            (GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory D r0 project initial n omega)
            (GKGMM19IterativeLocalVoting.l2BallDirection
             (GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory D r0 project initial n omega)
             (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)
             (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)))) →
        ∃ (C : ℝ),
         ∀ (n : ℕ),
          (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
             D)[(A n).indicator fun (x : ℕ → Coord → ℝ) => 1 |
            ↑GKGMM19IterativeLocalVoting.finiteModelBIdealNaturalFiltration
             n] ≤ᵐ[GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure D]
           fun (x : ℕ → Coord → ℝ) => C * GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)) ∧
 (∀ {Voter : Type u_3} {Coord : Type u_4} [inst : Fintype Coord] [Nonempty Coord]
   {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
   {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
   GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project →
    ∀ {r0 : ℝ},
     0 < r0 →
      ∀ (initial : Coord → ℝ) (A : ℕ → Set (ℕ → Coord → ℝ)),
       (∀ (n : ℕ), MeasurableSet (A n)) →
        (∀ (n : ℕ) (omega : ℕ → Coord → ℝ),
          omega ∈ A n ↔
           ¬GKGMM19IterativeLocalVoting.FiniteSubgradientAt
             (fun (y : Coord → ℝ) =>
              AppliedModelingLib.FiniteDimensionalNorms.l1 fun (i : Coord) =>
               y i - GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega i)
             (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory D r0 project initial n omega)
             (GKGMM19IterativeLocalVoting.l1LinfBallDirection
              (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory D r0 project initial n
               omega)
              (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)
              (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)))) →
         ∃ (C : ℝ),
          ∀ (n : ℕ),
           (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
              D)[(A n).indicator fun (x : ℕ → Coord → ℝ) => 1 |
             ↑GKGMM19IterativeLocalVoting.finiteModelBIdealNaturalFiltration
              n] ≤ᵐ[GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure D]
            fun (x : ℕ → Coord → ℝ) => C * GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)) ∧
  ∀ {Voter : Type u_5} {Coord : Type u_6} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
   {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
   {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
   GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project →
    ∀ {r0 : ℝ},
     0 < r0 →
      ∀ (initial : Coord → ℝ) (A : ℕ → Set (ℕ → Coord → ℝ)),
       (∀ (n : ℕ), MeasurableSet (A n)) →
        (∀ (n : ℕ) (omega : ℕ → Coord → ℝ),
          omega ∈ A n ↔
           ¬GKGMM19IterativeLocalVoting.FiniteSubgradientAt
             (fun (y : Coord → ℝ) =>
              AppliedModelingLib.FiniteDimensionalNorms.linf fun (i : Coord) =>
               y i - GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega i)
             (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory D r0 project initial n omega)
             (GKGMM19IterativeLocalVoting.linfL1RawDirection
              (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory D r0 project initial n
               omega)
```

```
              (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeRawResponse r0
                (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory D r0 project initial)
                GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)
              (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)))) →
        ∃ (C : ℝ),
          ∀ (n : ℕ),
            (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
                D)[(A n).indicator fun (x : ℕ → Coord → ℝ) => 1 |
                ↑GKGMM19IterativeLocalVoting.finiteModelBIdealNaturalFiltration
                  n] ≤ᵐ[GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure D]
              fun (x : ℕ → Coord → ℝ) => C * GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)
```

Retained governing declarations

- AppliedModelingLib.FiniteDimensionalNorms.l1
- AppliedModelingLib.FiniteDimensionalNorms.l2
- AppliedModelingLib.FiniteDimensionalNorms.linf
- GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.FiniteSubgradientAt
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory
- GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory
- GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeRawResponse
- GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory
- GKGMM19IterativeLocalVoting.finiteModelBIdealNaturalFiltration
- GKGMM19IterativeLocalVoting.finiteModelBIdealSample
- GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.l1LinfBallDirection
- GKGMM19IterativeLocalVoting.l2BallDirection
- GKGMM19IterativeLocalVoting.linfL1RawDirection

Lean-elaborated claim roles

```
1. conclusion: (∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
    GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project →
      ∀ {r0 : ℝ},
        0 < r0 →
          ∀ (initial : Coord → ℝ) (A : ℕ → Set (ℕ → Coord → ℝ)),
            (∀ (n : ℕ), MeasurableSet (A n)) →
              (∀ (n : ℕ) (omega : ℕ → Coord → ℝ),
                  omega ∈ A n ↔
                    ¬GKGMM19IterativeLocalVoting.FiniteSubgradientAt
                        (fun (y : Coord → ℝ) =>
                          AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) =>
                            y i - GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega i)
                        (GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory D r0 project initial n omega)
                        (GKGMM19IterativeLocalVoting.l2BallDirection
                          (GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory D r0 project initial n omega)
                          (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)
                          (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)))) →
                ∃ (C : ℝ),
                  ∀ (n : ℕ),
                    (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
                          D)[(A n).indicator fun (x : ℕ → Coord → ℝ) => 1 |
                        ↑GKGMM19IterativeLocalVoting.finiteModelBIdealNaturalFiltration
                          n] ≤ᵐ[GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure D]
                      fun (x : ℕ → Coord → ℝ) => C * GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)) ∧
  (∀ {Voter : Type u_3} {Coord : Type u_4} [inst : Fintype Coord] [Nonempty Coord]
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
      {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
      GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project →
        ∀ {r0 : ℝ},
          0 < r0 →
            ∀ (initial : Coord → ℝ) (A : ℕ → Set (ℕ → Coord → ℝ)),
              (∀ (n : ℕ), MeasurableSet (A n)) →
                (∀ (n : ℕ) (omega : ℕ → Coord → ℝ),
                    omega ∈ A n ↔
                      ¬GKGMM19IterativeLocalVoting.FiniteSubgradientAt
```

```
              (fun (y : Coord → ℝ) =>
                AppliedModelingLib.FiniteDimensionalNorms.l1 fun (i : Coord) =>
                  y i - GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega i)
              (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory D r0 project initial n omega)
              (GKGMM19IterativeLocalVoting.l1LinfBallDirection
                (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory D r0 project initial n
                  omega)
                (GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)
                (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)))) →
          ∃ (C : ℝ),
            ∀ (n : ℕ),
              (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
                  D)[(A n).indicator fun (x : ℕ → Coord → ℝ) => 1 |
                ↑GKGMM19IterativeLocalVoting.finiteModelBIdealNaturalFiltration
                  n] ≤ᵐ[GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure D]
                fun (x : ℕ → Coord → ℝ) => C * GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)) ∧
  ∀ {Voter : Type u_5} {Coord : Type u_6} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    {D : GKGMM19IterativeLocalVoting.FiniteCoordinateIdealDistributionData Coord} {project : (Coord → ℝ) → Coord → ℝ},
    GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D project →
      ∀ {r0 : ℝ},
        0 < r0 →
          ∀ (initial : Coord → ℝ) (A : ℕ → Set (ℕ → Coord → ℝ)),
            (∀ (n : ℕ), MeasurableSet (A n)) →
              (∀ (n : ℕ) (omega : ℕ → Coord → ℝ),
                  omega ∈ A n ↔
                    ¬GKGMM19IterativeLocalVoting.FiniteSubgradientAt
                      (fun (y : Coord → ℝ) =>
                        AppliedModelingLib.FiniteDimensionalNorms.linf fun (i : Coord) =>
                          y i - GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega i)
                      (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory D r0 project initial n omega)
                      (GKGMM19IterativeLocalVoting.linfL1RawDirection
                        (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory D r0 project initial n
                          omega)
                        (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeRawResponse r0
                          (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory D r0 project initial)
                          GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)
                        (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)))) →
          ∃ (C : ℝ),
            ∀ (n : ℕ),
              (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
                  D)[(A n).indicator fun (x : ℕ → Coord → ℝ) => 1 |
                ↑GKGMM19IterativeLocalVoting.finiteModelBIdealNaturalFiltration
                  n] ≤ᵐ[GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure D]
                fun (x : ℕ → Coord → ℝ) => C * GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)
```

Lean proof endpoint (built separately)

GKGMM19IterativeLocalVoting.appendix_lemma2_bad_event_linear_rate

Recorded source-to-Spec screening **Verdict:** matches

Reason: For each listed norm pair, the target uses exactly the actual-response non-subgradient event and gives the conditional O(r_t) indicator bound.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 6

Source locator: cited publication, lines 2183-2190

Verbatim source input

Lemma 4. Suppose that fv (x) , wvk k k
k=1 kwv k2 kx − xv k2 , and define the function

/ ∂fvt (xt)},
At , I{g̃vt (xt) ∈

where g̃vt (xt) is as defined in (2) for q = 2. Then there exists C ∈ R s.t. ∀ n, P(At =
1|Ft) ≤ Crt .

Expanded PaperInterface specification

```
∀ {Voter : Type u_1} {Coord : Type u_2} {Component : Type u_3} [inst : Fintype Coord] [inst_1 : Fintype Component]
  [inst_2 : Nonempty Coord] {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  {D : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component}
  {project : (Coord → ℝ) → Coord → ℝ},
  GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D.idealDistribution project →
    ∀ (Aresponse : GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D) {r0 : ℝ},
      0 < r0 →
        ∀ (initial : Coord → ℝ) (A : ℕ → Set (ℕ → (Component → ℝ) × (Coord → ℝ))),
          (∀ (n : ℕ), MeasurableSet (A n)) →
            (∀ (n : ℕ) (omega : ℕ → (Component → ℝ) × (Coord → ℝ)),
                omega ∈ A n ↔
                  ¬GKGMM19IterativeLocalVoting.FiniteSubgradientAt
                      (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost D
                        (AppliedModelingLib.iidSequenceSample n omega))
                      (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory Aresponse r0
                        project initial n omega)
                      (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelARawDirection Aresponse
                        (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory Aresponse r0
                          project initial n omega)
                        (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))
                        (AppliedModelingLib.iidSequenceSample n omega))) →
              ∃ (C : ℝ),
                ∀ (n : ℕ),
                  (AppliedModelingLib.iidSequenceMeasure
                        D.jointMeasure)[(A n).indicator fun (x : ℕ → (Component → ℝ) × (Coord → ℝ)) => 1 |
                      ↑AppliedModelingLib.iidSequenceNaturalFiltration
                        n] ≤ᵐ[AppliedModelingLib.iidSequenceMeasure D.jointMeasure]
                    fun (x : ℕ → (Component → ℝ) × (Coord → ℝ)) => C * GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)
```

Retained governing declarations

- AppliedModelingLib.iidSequenceMeasure
- AppliedModelingLib.iidSequenceNaturalFiltration
- AppliedModelingLib.iidSequenceSample
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.FiniteSubgradientAt
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData
- GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource
- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory
- GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelARawDirection

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Type u_3
4. parameter: Fintype Coord
5. parameter: Fintype Component
6. assumption: Nonempty Coord
7. parameter: GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)
8. parameter: GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleData Coord Component
9. parameter: (Coord → ℝ) → Coord → ℝ
10. assumption: GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E D.idealDistribution project
11. parameter: GKGMM19IterativeLocalVoting.WeightedEuclideanJointSampleModelAResponseSource D
12. parameter: ℝ

```
13. assumption: 0 < r0
14. parameter: Coord → ℝ
15. parameter: ℕ → Set (ℕ → (Component → ℝ) × (Coord → ℝ))
16. assumption: ∀ (n : ℕ), MeasurableSet (A n)
17. assumption: ∀ (n : ℕ) (omega : ℕ → (Component → ℝ) × (Coord → ℝ)),
  omega ∈ A n ↔
    ¬GKGMM19IterativeLocalVoting.FiniteSubgradientAt
        (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleCost D (AppliedModelingLib.iidSequenceSample n omega))
        (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory Aresponse r0 project initial n
          omega)
        (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelARawDirection Aresponse
          (GKGMM19IterativeLocalVoting.weightedEuclideanJointSampleModelAOutcomeTrajectory Aresponse r0 project initial
            n omega)
          (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)) (AppliedModelingLib.iidSequenceSample n omega))
18. conclusion: ∃ (C : ℝ),
  ∀ (n : ℕ),
    (AppliedModelingLib.iidSequenceMeasure
          D.jointMeasure)[(A n).indicator fun (x : ℕ → (Component → ℝ) × (Coord → ℝ)) => 1 |
        ↑AppliedModelingLib.iidSequenceNaturalFiltration n] ≤ᵐ[AppliedModelingLib.iidSequenceMeasure D.jointMeasure]
      fun (x : ℕ → (Component → ℝ) × (Coord → ℝ)) => C * GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1)
```

Lean proof endpoint (built separately)

GKGMM19IterativeLocalVoting.appendix_lemma4_weighted_bad_event_linear_rate

Recorded source-to-Spec screening **Verdict:** matches

Reason: The target instantiates the stated weighted Euclidean Definition-2 cost, q=2 response event, and conditional O(r_t) bound.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 7

Source locator: cited publication, lines 1976-1991

Verbatim source input

Theorem 4. (Nemirovski et al., 2009; Strassen,
R 1965) Let θ ∈ Θ be a random vector with
distribution P. Let f (x) = E[f (x, θ)] = Θ f (x, θ)dP (θ), for x ∈ X , a non-empty bounded

closed convex set, and assume the expectation is well-defined and finite valued. Suppose that
f (·, θ), θ ∈ Θ is convex and f¯(·) is continuous and finite valued in a neighborhood of point
x.
For each θ, choose any g(x, θ) ∈ ∂f (x, θ). Then, there exists ḡ(x) ∈ ∂ f¯(x) s.t. ḡ(x) =
Eθ [g(x, θ)].

This theorem says that the expected value of the sub-gradient of the utility at any point x across voters is a subgradient of the societal utility at x, irrespective of how the voters choose the subgradient when there are multiple subgradients, i.e., when the utility function is not differentiable. This key result allows us to use the subgradient of utility function of a sampled voter as an unbaised estimate of the societal subgradient.

Expanded PaperInterface specification

```
∀ {Theta : Type u_1} {Coord : Type u_2} [inst : MeasurableSpace Theta] [inst_1 : Fintype Coord]
  (mu : MeasureTheory.Measure Theta) [MeasureTheory.IsProbabilityMeasure mu] (solutionSpace : Set (Coord → ℝ))
  (sampleCost : Theta → (Coord → ℝ) → ℝ) (x : Coord → ℝ) (sampleGradient : Theta → Coord → ℝ),
  solutionSpace.Nonempty →
    Bornology.IsBounded solutionSpace →
      IsClosed solutionSpace →
        Convex ℝ solutionSpace →
          x ∈ solutionSpace →
            (∀ y ∈ solutionSpace, MeasureTheory.Integrable (fun (theta : Theta) => sampleCost theta y) mu) →
              (∀ (i : Coord), MeasureTheory.Integrable (fun (theta : Theta) => sampleGradient theta i) mu) →
                (∀ (theta : Theta), ConvexOn ℝ solutionSpace (sampleCost theta)) →
                  (∃ (epsilon : ℝ),
                      0 < epsilon ∧
                        ∀ (y : Coord → ℝ),
                          dist y x < epsilon →
                            MeasureTheory.Integrable (fun (theta : Theta) => sampleCost theta y) mu ∧
                              ContinuousAt (fun (z : Coord → ℝ) => ∫ (theta : Theta), sampleCost theta z ∂mu) y) →
                    (∀ (theta : Theta),
                        GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt (sampleCost theta) solutionSpace x
                          (sampleGradient theta)) →
                      ∀ y ∈ solutionSpace,
                        (∫ (theta : Theta), sampleCost theta x ∂mu +
                            (AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional fun (i : Coord) =>
                                ∫ (theta : Theta), sampleGradient theta i ∂mu)
                              fun (i : Coord) => y i - x i) ≤
                          ∫ (theta : Theta), sampleCost theta y ∂mu
```

Retained governing declarations

- AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional
- GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: MeasurableSpace Theta
4. parameter: Fintype Coord
5. parameter: MeasureTheory.Measure Theta
6. assumption: MeasureTheory.IsProbabilityMeasure mu
7. parameter: Set (Coord → ℝ)
8. parameter: Theta → (Coord → ℝ) → ℝ
9. parameter: Coord → ℝ
10. parameter: Theta → Coord → ℝ
11. assumption: solutionSpace.Nonempty
12. assumption: Bornology.IsBounded solutionSpace
13. assumption: IsClosed solutionSpace
14. assumption: Convex ℝ solutionSpace
15. assumption: x ∈ solutionSpace
16. assumption: ∀ y ∈ solutionSpace, MeasureTheory.Integrable (fun (theta : Theta) => sampleCost theta y) mu
17. assumption: ∀ (i : Coord), MeasureTheory.Integrable (fun (theta : Theta) => sampleGradient theta i) mu
18. assumption: ∀ (theta : Theta), ConvexOn ℝ solutionSpace (sampleCost theta)
19. assumption: ∃ (epsilon : ℝ),
  0 < epsilon ∧
    ∀ (y : Coord → ℝ),
      dist y x < epsilon →
        MeasureTheory.Integrable (fun (theta : Theta) => sampleCost theta y) mu ∧
          ContinuousAt (fun (z : Coord → ℝ) => ∫ (theta : Theta), sampleCost theta z ∂mu) y
20. assumption: ∀ (theta : Theta),
  GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt (sampleCost theta) solutionSpace x (sampleGradient theta)
21. parameter: Coord → ℝ
22. assumption: y ∈ solutionSpace
```

```
23. conclusion: (∫ (theta : Theta), sampleCost theta x ∂mu +
    (AppliedModelingLib.FiniteDimensionalNorms.coordinateLinearFunctional fun (i : Coord) =>
        ∫ (theta : Theta), sampleGradient theta i ∂mu)
      fun (i : Coord) => y i - x i) ≤
  ∫ (theta : Theta), sampleCost theta y ∂mu
```

Lean proof endpoint (built separately)

GKGMM19IterativeLocalVoting.appendix_theorem4_expected_subgradient_boundary

Recorded source-to-Spec screening **Verdict:** matches

Reason: The binders give a probability law on theta, a nonempty bounded closed convex feasible set, finite integrable sample costs, convexity of every sample cost, local finiteness and continuity of the expected cost near x, and an arbitrary integrable selected within-set subgradient at x. The conclusion is exactly the finite-coordinate subgradient inequality for the coordinatewise expectation of that selection against the expected objective, which is the source's existence and identification of g-bar(x) = E[g(x,theta)] in the domain X.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 8

Source locator: cited publication, lines 2004-2024

Verbatim source input

```
Theorem 5. (Jiang & Walrand, 2010) Consider the above update rule. If
                              f (·) has a unique minimizer x∗ ∈ X
                                     X           X
                              rt > 0,     rt = ∞,    rt2 < ∞
                                      t           t
                              ∃C1 ∈ R < ∞ s.t. k∂f (x)k2 ≤ C1 , ∀ x ∈ X
                              ∃C2 ∈ R < ∞ s.t. Et [kzt k2 ] ≤ C2 , ∀ t
                              ∃C3 ∈ R < ∞ s.t. kbt k2 ≤ C3 , ∀ t
                              X
                                 rt kbt k < ∞ w.p. 1
                               t

Then xt    → x∗   w.p. 1 as t → ∞.
   Note: Jiang and Walrand (2010) prove the result for gradients, though the same proof
follows for subgradients. Only the inequality [x
                                                            ∗ − x ]T g ≤ f (x∗ ) − f (x ) for gradient g
                                                                 t    t                t                t
at iteration t is used, which holds for subgradients.              Boyd and Mutapcic (2006) provide
a general discussion of subgradient methods, along with similar results.                  Shor (1998), in
Theorem 46, provide a convergence proof for the stochastic subgradient method without
projections and the extra noise terms.
```

Settled source reading The checked Appendix Theorem 5 is specialized to a deterministic summable bias sequence. This suffices for the zero-bias main-text application; the broader adapted-bias statement is not established by this specialization.

Expanded PaperInterface specification

```
∀ {Omega : Type u_1} {Coord : Type u_2} {mOmega : MeasurableSpace Omega} [inst : Fintype Coord]
  [inst_1 : Nonempty Coord] (mu : MeasureTheory.Measure Omega) [MeasureTheory.IsProbabilityMeasure mu]
  (filtration : MeasureTheory.Filtration ℕ mOmega) (solutionSpace : Set (Coord → ℝ)) (objective : (Coord → ℝ) → ℝ)
  (project : (Coord → ℝ) → Coord → ℝ) (trajectory meanSubgradient noise : ℕ → Omega → Coord → ℝ) (bias : ℕ → Coord → ℝ)
  (radius : ℕ → ℝ) (xstar : Coord → ℝ),
  solutionSpace.Nonempty →
    Bornology.IsBounded solutionSpace →
      IsClosed solutionSpace →
        Convex ℝ solutionSpace →
          ConvexOn ℝ solutionSpace objective →
            xstar ∈ solutionSpace →
              IsMinOn objective solutionSpace xstar →
                (∀ y ∈ solutionSpace, objective y = objective xstar → y = xstar) →
                  (∀ (t : ℕ), 0 < t → 0 < radius t) →
                    (Summable fun (t : ℕ) => radius (t + 1) ^ 2) →
                      Filter.Tendsto (fun (n : ℕ) => ∑ t ∈ Finset.range n, radius (t + 1)) Filter.atTop Filter.atTop →
                        (∀ (y : Coord → ℝ),
                            project y ∈ solutionSpace ∧
                              IsMinOn
                                (fun (z : Coord → ℝ) =>
                                  GKGMM19IterativeLocalVoting.finiteCoordinateDistance
                                    GKGMM19IterativeLocalVoting.SourceNorm.l2 z y)
                                solutionSpace (project y)) →
                          (∀ (t : ℕ),
                              ∀ᵐ (omega : Omega) ∂mu,
                                trajectory (t + 1) omega =
                                  project fun (i : Coord) =>
                                    trajectory t omega i -
                                      radius (t + 1) * (meanSubgradient t omega i + noise t omega i + bias t i)) →
                            (∀ (i : Coord),
                                MeasureTheory.StronglyAdapted filtration fun (t : ℕ) (omega : Omega) =>
                                  trajectory t omega i) →
                              (∀ (i : Coord),
                                  MeasureTheory.StronglyAdapted filtration fun (t : ℕ) (omega : Omega) =>
                                    meanSubgradient t omega i) →
                                (∀ (t : ℕ),
                                    ∀ᵐ (omega : Omega) ∂mu,
                                      trajectory t omega ∈ solutionSpace ∧
                                        GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt objective solutionSpace
                                          (trajectory t omega) (meanSubgradient t omega)) →
                                  (∃ (C1 : ℝ),
                                      0 ≤ C1 ∧
                                        ∀ y ∈ solutionSpace,
                                          ∀ (g : Coord → ℝ),
                                            GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt objective
                                                solutionSpace y g →
                                              GKGMM19IterativeLocalVoting.finiteCoordinateNorm
                                                  GKGMM19IterativeLocalVoting.SourceNorm.l2 g ≤
                                                C1) →
                                    (∀ (t : ℕ) (i : Coord),
                                        MeasureTheory.Integrable (fun (omega : Omega) => noise t omega i) mu) →
                                      (∀ (t : ℕ) (i : Coord),
                                          mu[fun (omega : Omega) => noise t omega i | ↑filtration t] =ᵐ[mu] 0) →
                                        (∀ (t : ℕ),
                                            MeasureTheory.Integrable
                                              (fun (omega : Omega) =>
```

```
                GKGMM19IterativeLocalVoting.finiteCoordinateNorm
                  GKGMM19IterativeLocalVoting.SourceNorm.l2 (noise t omega) ^
                 2)
               mu) →
             (∃ (C2 : ℝ),
               0 ≤ C2 ∧
                ∀ (t : ℕ),
                 mu[fun (omega : Omega) =>
                   GKGMM19IterativeLocalVoting.finiteCoordinateNorm
                     GKGMM19IterativeLocalVoting.SourceNorm.l2 (noise t omega) ^
                    2 |
                   ↑filtration t] ≤ᵐ[mu]
                  fun (x : Omega) => C2) →
              (∃ (C3 : ℝ),
                0 ≤ C3 ∧
                 ∀ (t : ℕ),
                  GKGMM19IterativeLocalVoting.finiteCoordinateNorm
                    GKGMM19IterativeLocalVoting.SourceNorm.l2 (bias t) ≤
                   C3) →
               (Summable fun (t : ℕ) =>
                 radius (t + 1) *
                  GKGMM19IterativeLocalVoting.finiteCoordinateNorm
                   GKGMM19IterativeLocalVoting.SourceNorm.l2 (bias t)) →
                (MeasureTheory.StronglyAdapted filtration fun (t : ℕ) (omega : Omega) =>
                  objective (trajectory t omega) - objective xstar) →
                 (∀ (t : ℕ),
                   MeasureTheory.Integrable
                    (fun (omega : Omega) =>
                     AppliedModelingLib.FiniteDimensionalNorms.l2Sq
                      fun (i : Coord) => trajectory t omega i - xstar i)
                    mu) →
                  (∀ (t : ℕ),
                    MeasureTheory.Integrable
                     (fun (omega : Omega) =>
                      objective (trajectory t omega) - objective xstar)
                     mu) →
                   (∀ (t : ℕ) (i : Coord),
                     MeasureTheory.Integrable
                      (fun (omega : Omega) =>
                       (trajectory t omega i - xstar i) * noise t omega i)
                      mu) →
                    (∃ (distanceBound : ℝ),
                      0 ≤ distanceBound ∧
                       ∀ (t : ℕ),
                        ∀ᵐ (omega : Omega) ∂mu,
                         (AppliedModelingLib.FiniteDimensionalNorms.l2
                           fun (i : Coord) =>
                           trajectory t omega i - xstar i) ≤
                          distanceBound) →
                     (∀ y ∈ solutionSpace,
                       ∃ (g : Coord → ℝ),
                        GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt
                         objective solutionSpace y g) →
                      ∀ᵐ (omega : Omega) ∂mu,
                       Filter.Tendsto (fun (t : ℕ) => trajectory t omega)
                        Filter.atTop (nhds xstar)
```

Retained governing declarations

- AppliedModelingLib.FiniteDimensionalNorms.l2
- AppliedModelingLib.FiniteDimensionalNorms.l2Sq
- GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteCoordinateDistance
- GKGMM19IterativeLocalVoting.finiteCoordinateNorm

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: MeasurableSpace Omega
4. parameter: Fintype Coord
5. assumption: Nonempty Coord
6. parameter: MeasureTheory.Measure Omega
7. assumption: MeasureTheory.IsProbabilityMeasure mu
8. parameter: MeasureTheory.Filtration ℕ mOmega
9. parameter: Set (Coord → ℝ)
10. parameter: (Coord → ℝ) → ℝ
11. parameter: (Coord → ℝ) → Coord → ℝ
12. parameter: ℕ → Omega → Coord → ℝ
13. parameter: ℕ → Omega → Coord → ℝ
14. parameter: ℕ → Omega → Coord → ℝ
15. parameter: ℕ → Coord → ℝ
16. parameter: ℕ → ℝ
17. parameter: Coord → ℝ
18. assumption: solutionSpace.Nonempty
19. assumption: Bornology.IsBounded solutionSpace

20. assumption: IsClosed solutionSpace
21. assumption: Convex ℝ solutionSpace
22. assumption: ConvexOn ℝ solutionSpace objective
23. assumption: xstar ∈ solutionSpace
24. assumption: IsMinOn objective solutionSpace xstar
25. assumption: ∀ y ∈ solutionSpace, objective y = objective xstar → y = xstar
26. assumption: ∀ (t : ℕ), 0 < t → 0 < radius t
27. assumption: Summable fun (t : ℕ) => radius (t + 1) ^ 2
28. assumption: Filter.Tendsto (fun (n : ℕ) => ∑ t ∈ Finset.range n, radius (t + 1)) Filter.atTop Filter.atTop
29. assumption: ∀ (y : Coord → ℝ),
  project y ∈ solutionSpace ∧
    IsMinOn
      (fun (z : Coord → ℝ) =>
        GKGMM19IterativeLocalVoting.finiteCoordinateDistance GKGMM19IterativeLocalVoting.SourceNorm.l2 z y)
      solutionSpace (project y)
30. assumption: ∀ (t : ℕ),
  ∀ᵐ (omega : Omega) ∂mu,
    trajectory (t + 1) omega =
      project fun (i : Coord) =>
        trajectory t omega i - radius (t + 1) * (meanSubgradient t omega i + noise t omega i + bias t i)
31. assumption: ∀ (i : Coord), MeasureTheory.StronglyAdapted filtration fun (t : ℕ) (omega : Omega) => trajectory t omega i
32. assumption: ∀ (i : Coord), MeasureTheory.StronglyAdapted filtration fun (t : ℕ) (omega : Omega) => meanSubgradient t omega i
33. assumption: ∀ (t : ℕ),
  ∀ᵐ (omega : Omega) ∂mu,
    trajectory t omega ∈ solutionSpace ∧
      GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt objective solutionSpace (trajectory t omega)
        (meanSubgradient t omega)
34. assumption: ∃ (C1 : ℝ),
  0 ≤ C1 ∧
    ∀ y ∈ solutionSpace,
      ∀ (g : Coord → ℝ),
        GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt objective solutionSpace y g →
          GKGMM19IterativeLocalVoting.finiteCoordinateNorm GKGMM19IterativeLocalVoting.SourceNorm.l2 g ≤ C1
35. assumption: ∀ (t : ℕ) (i : Coord), MeasureTheory.Integrable (fun (omega : Omega) => noise t omega i) mu
36. assumption: ∀ (t : ℕ) (i : Coord), mu[fun (omega : Omega) => noise t omega i | ↑filtration t] =ᵐ[mu] 0
37. assumption: ∀ (t : ℕ),
  MeasureTheory.Integrable
    (fun (omega : Omega) =>
      GKGMM19IterativeLocalVoting.finiteCoordinateNorm GKGMM19IterativeLocalVoting.SourceNorm.l2 (noise t omega) ^ 2)
    mu
38. assumption: ∃ (C2 : ℝ),
  0 ≤ C2 ∧
    ∀ (t : ℕ),
      mu[fun (omega : Omega) =>
          GKGMM19IterativeLocalVoting.finiteCoordinateNorm GKGMM19IterativeLocalVoting.SourceNorm.l2 (noise t omega) ^
            2 |
          ↑filtration t] ≤ᵐ[mu]
        fun (x : Omega) => C2
39. assumption: ∃ (C3 : ℝ),
  0 ≤ C3 ∧
    ∀ (t : ℕ), GKGMM19IterativeLocalVoting.finiteCoordinateNorm GKGMM19IterativeLocalVoting.SourceNorm.l2 (bias t) ≤ C3
40. assumption: Summable fun (t : ℕ) =>
  radius (t + 1) * GKGMM19IterativeLocalVoting.finiteCoordinateNorm GKGMM19IterativeLocalVoting.SourceNorm.l2 (bias t)
41. assumption: MeasureTheory.StronglyAdapted filtration fun (t : ℕ) (omega : Omega) => objective (trajectory t omega) - objective xstar
42. assumption: ∀ (t : ℕ),
  MeasureTheory.Integrable
    (fun (omega : Omega) =>
      AppliedModelingLib.FiniteDimensionalNorms.l2Sq fun (i : Coord) => trajectory t omega i - xstar i)
    mu
43. assumption: ∀ (t : ℕ), MeasureTheory.Integrable (fun (omega : Omega) => objective (trajectory t omega) - objective xstar) mu
44. assumption: ∀ (t : ℕ) (i : Coord),
  MeasureTheory.Integrable (fun (omega : Omega) => (trajectory t omega i - xstar i) * noise t omega i) mu
45. assumption: ∃ (distanceBound : ℝ),
  0 ≤ distanceBound ∧
    ∀ (t : ℕ),
      ∀ᵐ (omega : Omega) ∂mu,
        (AppliedModelingLib.FiniteDimensionalNorms.l2 fun (i : Coord) => trajectory t omega i - xstar i) ≤ distanceBound
46. assumption: ∀ y ∈ solutionSpace,
  ∃ (g : Coord → ℝ), GKGMM19IterativeLocalVoting.FiniteSubgradientWithinAt objective solutionSpace y g
47. conclusion: ∀ᵐ (omega : Omega) ∂mu, Filter.Tendsto (fun (t : ℕ) => trajectory t omega) Filter.atTop (nhds xstar)

Lean proof endpoint (built separately)

GKGMM19IterativeLocalVoting.appendix_theorem5_ssgm_convergence_boundary

Recorded source-to-Spec screening **Verdict:** matches

Reason: Under the displayed projected stochastic-subgradient update, the target requires the source’s unique feasible minimizer, positive nonsummable steps with summable squares, uniformly bounded objective subgradients, conditionally mean-zero noise with bounded conditional second moment, uniformly bounded deterministic bias, and summability of radius times the bias norm. The extra adaptedness, measurability, integrability, projection, and bounded-trajectory clauses make those source expectations and update well defined. With the approved deterministic-bias reading, the conclusion is precisely almost-sure convergence of x_t to xstar.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Main-text source claims (continued)

Review row 9

Source locator: cited publication, lines 481-484

Verbatim source input

Theorem 1. Suppose that conditions C1 , C2 , and C3 are satisfied, the voter utilities are Lp
normed, and voters respond to query (1) according to either Model A or Model B. Then,
ILV with Lq neighborhoods converges to the societal optimal point w.p. 1 when (p, q) = (2, 2),
(1, ∞), or (∞, 1).

[Next verbatim source excerpt]

More formally, consider a M -dimensional societal decision problem in X ⊂ RM and a
population of voters V , where each voter v ∈ V has bounded utility fv (x) ∈ R, ∀ x ∈ X .
Then we consider the class of algorithms described in Algorithm 1. We study the algorithm
class under two plausible models of how voters respond to query (1), which asks for the
voter's favorite point in a local region.

• Model A: One possibility is that voters exactly perform the maximization asked of
them, responding with their favorite point in the given L
q norm constraint set. In other

words, they return a point arg maxx∈{s:ks−x fvt (x). Note that by definition of this
t−1 kq ≤rt }
movement, the algorithm is myopically incentive compatible : if a voter is the last voter
and no projections are used, then truthfully performing this movement is the dominant
strategy. In general, the mechanism is not globally incentive compatible, nor incentive
compatible with projections onto the feasible region. Simple examples of manipulations
in both instances exist.

• Model B: On the other hand, voters may not actually search within the constraint set
to find their favorite point inside of it. Rather, a voter v may have an idea about how
to best improve the current point and then move in that direction to the boundary of
the given constraint set. This model leads to a voter moving the current solution in the
gt
direction of the gradient of her utility function, returning a point xt−1 + rt
kgt kq , for some
gt ∈ ∂fvt (xt−1). Note that ∂f (x) denotes the set of subgradients of a function f at x, i.e.
g ∈ ∂f (x) if ∀y , f (y) − f (x) ≥ g T (y − x).

ILV is directly inspired by the stochastic approximation approach to solve optimization
problems (Robbins & Monro, 1951), especially stochastic gradient descent (SGD) and the
stochastic subgradient method (SSGM). The idea is that if (a) voter preferences are drawn
from some probability distribution and (b) the response of a voter to the query (1) moves the
solution approximately in the direction of her utility gradient, then this procedure almost
implements stochastic gradient descent for minimizing negative expected utility.

The caveat is that although the procedure can potentially obtain the direction of the
gradient of the voter utilities, it cannot in general obtain any information about its magni-
tude since the movement norm is chosen by the procedure itself. However, we show that for
certain plausible utility and voter response models, the algorithm does indeed converge to a
unique point with desirable properties, including cases in which it converges to the societal
optimum.

Note that with such feedback and without any additional assumptions on voter prefer-
ences (e.g. that voter utilities are normalized to the same scale), no algorithm has any hope
of finding a desirable solution that depends on the cardinal values of voters' utilities, e.g., the
social welfare maximizing solution (the solution that maximizes the sum of agent utilities).
This is because an algorithm that uses only ordinal information about voter preferences is
insensitive to any scaling or even monotonic transformations of those preferences.

317

[form-feed in source extraction]
Garg, Kamble, Goel, Marn, & Munagala

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution x0 ∈ X , tolerance ϵ > 0, an integer N , initial radius r0 > 0,
termination time T , norm q for local neighborhood.
Output: Solution x.

• For t ≥ 1, sample a voter vt ∈ V at random from the population; set rt = r0 /t
and elicit

x0t = arg maxx∈{s:ks−xt−1 kq ≤rt } fvt (x), (1)
0
and then compute xt = [xt]X , where [·]X is a projection onto space X ; i.e. ask
the voter to move to her favorite point within constraints, and then project the
reported point onto X .

• Stop when either t = T , in which case return xT , or when
maxl, m∈{t−N,...,t} |xl − xm | ≤ ϵ, in which case return x = xt .

[Next verbatim source excerpt]

C1 The solution space X ⊆ RM is non-empty, bounded, closed, and convex.
C2 Each voter v has a unique ideal solution xv ∈ X .
C3 The ideal point xv of each voter is drawn independently from a probability distribution with a bounded and measurable density function hX .

Under this model, for a solution x ∈ X , the societal utility is given by Ev [fv (x)]. and the social optimal (SO) solution is any x
∗ ∈ arg max
x∈X Ev [fv (x)].

"Convergence" of ILV refers to the convergence of the sequence of random variables {xt }t≥1 to some x ∈ X with probability 1, assuming that the algorithm is allowed to run indefinitely (this notion of convergence also implies the termination of the algorithm with probability 1).

In the following subsections, we present several classes of utility functions for which the algorithm converges, summarized in Table 1. We further formalize the relationship to directional equilibria in Section 3.3.

3.1 Spatial Utilities
Here we consider spatial utility functions, where the utilities of each voters can be expressed in the form of some kind of spatial distance from their ideal solutions. First, we consider the following kind of utilities.

Definition 1. Lp normed utilities. The voter utility function is Lp normed if fv (x) = −kx − xv kp , ∀x ∈ X .

Expanded PaperInterface specification

```
(∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
  (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
  0 < r0 →
   ∀ initial ∈ E.solutionSpace,
    ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data 2),
     GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
      GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
       (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
       (GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory C3.data r0 project initial)) ∧
 (∀ {Voter : Type u_3} {Coord : Type u_4} [inst : Fintype Coord] [Nonempty Coord]
   {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
   (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
   (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
   0 < r0 →
    ∀ initial ∈ E.solutionSpace,
     ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data 2),
      GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
       GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
        (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
        (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 2 r0 project initial)) ∧
  (∀ {Voter : Type u_5} {Coord : Type u_6} [inst : Fintype Coord] [Nonempty Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
    (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
    0 < r0 →
     ∀ initial ∈ E.solutionSpace,
      ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data 1),
       GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
        GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
         (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
         (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project initial)) ∧
   (∀ {Voter : Type u_7} {Coord : Type u_8} [inst : Fintype Coord] [Nonempty Coord]
     {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
     (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
     (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
     0 < r0 →
      ∀ initial ∈ E.solutionSpace,
       ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data 1),
        GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
         GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
          (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
          (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 1 r0 project initial)) ∧
    (∀ {Voter : Type u_9} {Coord : Type u_10} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
      (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
      (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
      0 < r0 →
       ∀ initial ∈ E.solutionSpace,
        ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E C3.data),
         GKGMM19IterativeLocalVoting.finiteModelBLinfSourceInputsFormula C3 S T ∧
          GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
           (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
           (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory C3.data r0 project initial)) ∧
     ∀ {Voter : Type u_11} {Coord : Type u_12} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
      (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
      (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
      0 < r0 →
       ∀ initial ∈ E.solutionSpace,
        ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E C3.data),
```

```
GKGMM19IterativeLocalVoting.finiteModelBLinfSourceInputsFormula C3 S T ∧
  GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
    (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
    (GKGMM19IterativeLocalVoting.finiteModelBLinfOutcomeTrajectory C3.data r0 project initial)
```

Retained governing declarations

- GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource
- GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal
- GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory
- GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory
- GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory
- GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
- GKGMM19IterativeLocalVoting.finiteModelBLinfOutcomeTrajectory
- GKGMM19IterativeLocalVoting.finiteModelBLinfSourceInputsFormula
- GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula
- GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory

Lean-elaborated claim roles

```
1. conclusion: (∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
    {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
    (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
    (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
    0 < r0 →
      ∀ initial ∈ E.solutionSpace,
        ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data 2),
          GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
            GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
              (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
              (GKGMM19IterativeLocalVoting.finiteModelAL2OutcomeTrajectory C3.data r0 project initial)) ∧
  (∀ {Voter : Type u_3} {Coord : Type u_4} [inst : Fintype Coord] [Nonempty Coord]
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
      (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
      (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
      0 < r0 →
        ∀ initial ∈ E.solutionSpace,
          ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data 2),
            GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
              GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
                (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
                (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 2 r0 project initial)) ∧
    (∀ {Voter : Type u_5} {Coord : Type u_6} [inst : Fintype Coord] [Nonempty Coord]
        {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
        (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
        (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
        0 < r0 →
          ∀ initial ∈ E.solutionSpace,
            ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data 1),
              GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
                GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
                  (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
                  (GKGMM19IterativeLocalVoting.finiteModelAL1LinfOutcomeTrajectory C3.data r0 project initial)) ∧
      (∀ {Voter : Type u_7} {Coord : Type u_8} [inst : Fintype Coord] [Nonempty Coord]
          {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
          (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
          (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
          0 < r0 →
            ∀ initial ∈ E.solutionSpace,
              ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data 1),
                GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
                  GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
                    (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
                    (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data 1 r0 project initial)) ∧
        (∀ {Voter : Type u_9} {Coord : Type u_10} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
            {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
            (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
            (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
            0 < r0 →
              ∀ initial ∈ E.solutionSpace,
                ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E C3.data),
                  GKGMM19IterativeLocalVoting.finiteModelBLinfSourceInputsFormula C3 S T ∧
                    GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
```

```
            (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
            (GKGMM19IterativeLocalVoting.finiteModelALinfL1OutcomeTrajectory C3.data r0 project initial)) ∧
     ∀ {Voter : Type u_11} {Coord : Type u_12} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
      {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
      (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
      (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {r0 : ℝ},
      0 < r0 →
        ∀ initial ∈ E.solutionSpace,
          ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLinfSocialObjectiveSource E C3.data),
            GKGMM19IterativeLocalVoting.finiteModelBLinfSourceInputsFormula C3 S T ∧
              GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
                (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
                (GKGMM19IterativeLocalVoting.finiteModelBLinfOutcomeTrajectory C3.data r0 project initial)
```

Lean proof endpoint (built separately)

GKGMM19IterativeLocalVoting.theorem1_finite_c3_six_cases

Recorded source-to-Spec screening **Verdict:** matches

Reason: The six conjuncts are exactly Model A and Model B for each source pair (p,q) = (2,2), (1,infinity), and (infinity,1). In every branch the displayed C1-C3 carrier, feasible Euclidean projection, iid ideal sampling, approved raw utility-query domain, Lp or Linfinity social-objective identification, r_t = r0/t trajectory, and feasible initial point lead to almost-sure convergence to the societal-optimal set. No norm pair or response model from Theorem 1 is omitted.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 10

Source locator: cited publication, lines 505-508

Verbatim source input

Theorem 2. Suppose that conditions C1 , C2 , and C3 are satisfied, the voter utilities are Lp normed, and voters respond to query (1) according to Model B. Then, ILV with Lq neighborhoods converges to the societal optimal point w.p. 1 for any p > 0 and q > 0 such that 1/p + 1/q = 1.

[Next verbatim source excerpt]

More formally, consider a M -dimensional societal decision problem in X ⊂ RM and a population of voters V , where each voter v ∈ V has bounded utility fv (x) ∈ R, ∀ x ∈ X . Then we consider the class of algorithms described in Algorithm 1. We study the algorithm class under two plausible models of how voters respond to query (1), which asks for the voter's favorite point in a local region.

• Model A: One possibility is that voters exactly perform the maximization asked of them, responding with their favorite point in the given L
q norm constraint set. In other

words, they return a point arg maxx∈{s:ks−x fvt (x). Note that by definition of this
t−1 kq ≤rt }
movement, the algorithm is myopically incentive compatible : if a voter is the last voter and no projections are used, then truthfully performing this movement is the dominant strategy. In general, the mechanism is not globally incentive compatible, nor incentive compatible with projections onto the feasible region. Simple examples of manipulations in both instances exist.

• Model B: On the other hand, voters may not actually search within the constraint set to find their favorite point inside of it. Rather, a voter v may have an idea about how to best improve the current point and then move in that direction to the boundary of the given constraint set. This model leads to a voter moving the current solution in the
gt
direction of the gradient of her utility function, returning a point xt−1 + rt
kgt kq , for some
gt ∈ ∂fvt (xt−1). Note that ∂f (x) denotes the set of subgradients of a function f at x, i.e. g ∈ ∂f (x) if ∀y , f (y) − f (x) ≥ g T (y − x).

ILV is directly inspired by the stochastic approximation approach to solve optimization problems (Robbins & Monro, 1951), especially stochastic gradient descent (SGD) and the stochastic subgradient method (SSGM). The idea is that if (a) voter preferences are drawn from some probability distribution and (b) the response of a voter to the query (1) moves the solution approximately in the direction of her utility gradient, then this procedure almost implements stochastic gradient descent for minimizing negative expected utility.

The caveat is that although the procedure can potentially obtain the direction of the gradient of the voter utilities, it cannot in general obtain any information about its magnitude since the movement norm is chosen by the procedure itself. However, we show that for certain plausible utility and voter response models, the algorithm does indeed converge to a unique point with desirable properties, including cases in which it converges to the societal optimum.

Note that with such feedback and without any additional assumptions on voter preferences (e.g. that voter utilities are normalized to the same scale), no algorithm has any hope of finding a desirable solution that depends on the cardinal values of voters' utilities, e.g., the social welfare maximizing solution (the solution that maximizes the sum of agent utilities). This is because an algorithm that uses only ordinal information about voter preferences is insensitive to any scaling or even monotonic transformations of those preferences.

317

[form-feed in source extraction]
Garg, Kamble, Goel, Marn, & Munagala

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution x0 ∈ X , tolerance ϵ > 0, an integer N , initial radius r0 > 0, termination time T , norm q for local neighborhood.
Output: Solution x.

• For t ≥ 1, sample a voter vt ∈ V at random from the population; set rt = r0 /t and elicit

x0t = arg maxx∈{s:ks−xt−1 kq ≤rt } fvt (x), (1)
0
and then compute xt = [xt]X , where [·]X is a projection onto space X ; i.e. ask the voter to move to her favorite point within constraints, and then project the reported point onto X .

• Stop when either t = T , in which case return xT , or when maxl, m∈{t−N,...,t} |xl − xm | ≤ ϵ, in which case return x = xt .

[Next verbatim source excerpt]

C1 The solution space $X \subseteq RM$ is non-empty, bounded, closed, and convex.
C2 Each voter v has a unique ideal solution $xv \in X$.
C3 The ideal point xv of each voter is drawn independently from a probability distribution with a bounded and measurable density function hX .

Under this model, for a solution $x \in X$, the societal utility is given by Ev [fv (x)]. and the social optimal (SO) solution is any x
∗ ∈ arg max
x∈X Ev [fv (x)].
"Convergence" of ILV refers to the convergence of the sequence of random variables {xt }t≥1 to some $x \in X$ with probability 1, assuming that the algorithm is allowed to run indefinitely (this notion of convergence also implies the termination of the algorithm with probability 1).
In the following subsections, we present several classes of utility functions for which the algorithm converges, summarized in Table 1. We further formalize the relationship to directional equilibria in Section 3.3.

3.1 Spatial Utilities
Here we consider spatial utility functions, where the utilities of each voters can be expressed in the form of some kind of spatial distance from their ideal solutions. First, we consider the following kind of utilities.

Definition 1. Lp normed utilities. The voter utility function is Lp normed if fv (x) = −kx − xv kp , ∀x ∈ X .

Expanded PaperInterface specification

```
∀ {Voter : Type u_1} {Coord : Type u_2} [inst : Fintype Coord] [inst_1 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (C3 : GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E) {project : (Coord → ℝ) → Coord → ℝ}
  (S : GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project) {p q r0 : ℝ},
  GKGMM19IterativeLocalVoting.HolderDualFinite p q →
    0 < r0 →
      ∀ initial ∈ E.solutionSpace,
        ∀ (T : GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data p),
          GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
            (∀ᵐ (omega : ℕ → Coord → ℝ) ∂GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data,
                ∀ (n : ℕ),
                  GKGMM19IterativeLocalVoting.ModelBFiniteResponseAt (GKGMM19IterativeLocalVoting.SourceNorm.lp q)
                    (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data p r0 project initial n omega)
                    (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))
                    (fun (i : Coord) =>
                      -GKGMM19IterativeLocalVoting.lpCostGradientCandidate p
                          (fun (j : Coord) =>
                            GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data p r0 project initial n
                                omega j -
                              GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega j)
                          i)
                    (GKGMM19IterativeLocalVoting.finiteModelBOutcomeRawResponse p r0
                      (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data p r0 project initial)
                      GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)) ∧
              GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
                (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
                (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data p r0 project initial)
```

Retained governing declarations

- GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier
- GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source
- GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource
- GKGMM19IterativeLocalVoting.HolderDualFinite
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.ModelBFiniteResponseAt
- GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal
- GKGMM19IterativeLocalVoting.SourceNorm
- GKGMM19IterativeLocalVoting.finiteModelBIdealSample
- GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure
- GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula
- GKGMM19IterativeLocalVoting.finiteModelBOutcomeRawResponse
- GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory
- GKGMM19IterativeLocalVoting.ilvRadius
- GKGMM19IterativeLocalVoting.lpCostGradientCandidate

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Fintype Coord
4. assumption: Nonempty Coord
5. parameter: GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)
6. parameter: GKGMM19IterativeLocalVoting.FiniteCoordinateC3Carrier E
7. parameter: (Coord → ℝ) → Coord → ℝ
8. assumption: GKGMM19IterativeLocalVoting.FiniteModelBC1C2Source E C3.data project
9. parameter: ℝ
10. parameter: ℝ
11. parameter: ℝ
12. assumption: GKGMM19IterativeLocalVoting.HolderDualFinite p q
13. assumption: 0 < r0
14. parameter: Coord → ℝ
15. assumption: initial ∈ E.solutionSpace
16. assumption: GKGMM19IterativeLocalVoting.FiniteModelBLpSocialObjectiveSource E C3.data p
17. conclusion: GKGMM19IterativeLocalVoting.finiteModelBLpSourceInputsFormula C3 S T ∧
  (∀ᵐ (omega : ℕ → Coord → ℝ) ∂GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data,
      ∀ (n : ℕ),
        GKGMM19IterativeLocalVoting.ModelBFiniteResponseAt (GKGMM19IterativeLocalVoting.SourceNorm.lp q)
          (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data p r0 project initial n omega)
          (GKGMM19IterativeLocalVoting.ilvRadius r0 (n + 1))
          (fun (i : Coord) =>
            -GKGMM19IterativeLocalVoting.lpCostGradientCandidate p
                (fun (j : Coord) =>
                  GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data p r0 project initial n omega j -
                    GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega j)
                i)
          (GKGMM19IterativeLocalVoting.finiteModelBOutcomeRawResponse p r0
            (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data p r0 project initial)
            GKGMM19IterativeLocalVoting.finiteModelBIdealSample n omega)) ∧
    GKGMM19IterativeLocalVoting.OutcomeIndexedILVConvergesToSocietalOptimal E
      (GKGMM19IterativeLocalVoting.finiteModelBIdealSequenceMeasure C3.data)
      (GKGMM19IterativeLocalVoting.finiteModelBOutcomeTrajectory C3.data p r0 project initial)
```

Lean proof endpoint (built separately)

GKGMM19IterativeLocalVoting.theorem2_modelB_finite_holder_dual_c3

Recorded source-to-Spec screening **Verdict:** matches

Reason: HolderDualFinite expands to exactly p > 0, q > 0, and 1/p + 1/q = 1. Under the finite-coordinate C1-C3 data and the approved ambient raw utility formula, the target uses the source Lp cost subgradient, the Lq-normalized Model-B raw response at almost every iid execution step, Algorithm 1's r0/(n+1) radius and projection, and the exact expected-Lp social optimum; it concludes almost-sure convergence of that trajectory to the societal-optimal set.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 11

Source locator: cited publication, lines 627-637

Verbatim source input

Theorem 3. Suppose that C1 , C2 , and C3 are satisfied, and let G(x) , Ev k∇f ∇fv (x)
v (x)k2
.
Suppose, G(x) is uniformly continuous, L movement norm constraints are used, and voters
2

move according to Model B. If a trajectory {x}∞ t=1 of the algorithm converges to x , i.e.
∗

xt → x , then x is a directional equilibrium, i.e. G(x) = 0.
∗ ∗ ∗

[Next verbatim source excerpt]

More formally, consider a M -dimensional societal decision problem in X ⊂ RM and a population of voters V , where each voter v ∈ V has bounded utility fv (x) ∈ R, ∀ x ∈ X . Then we consider the class of algorithms described in Algorithm 1. We study the algorithm class under two plausible models of how voters respond to query (1), which asks for the voter's favorite point in a local region.

- Model A: One possibility is that voters exactly perform the maximization asked of them, responding with their favorite point in the given L
q norm constraint set. In other

  words, they return a point arg maxx∈{s:ks−x fvt (x). Note that by definition of this
  t−1 kq ≤rt }
  movement, the algorithm is myopically incentive compatible : if a voter is the last voter and no projections are used, then truthfully performing this movement is the dominant strategy. In general, the mechanism is not globally incentive compatible, nor incentive compatible with projections onto the feasible region. Simple examples of manipulations in both instances exist.

- Model B: On the other hand, voters may not actually search within the constraint set to find their favorite point inside of it. Rather, a voter v may have an idea about how to best improve the current point and then move in that direction to the boundary of the given constraint set. This model leads to a voter moving the current solution in the
  gt
  direction of the gradient of her utility function, returning a point xt−1 + rt
  kgt kq , for some
  gt ∈ ∂fvt (xt−1). Note that ∂f (x) denotes the set of subgradients of a function f at x, i.e. g ∈ ∂f (x) if ∀y , f (y) − f (x) ≥ g T (y − x).

ILV is directly inspired by the stochastic approximation approach to solve optimization problems (Robbins & Monro, 1951), especially stochastic gradient descent (SGD) and the stochastic subgradient method (SSGM). The idea is that if (a) voter preferences are drawn from some probability distribution and (b) the response of a voter to the query (1) moves the solution approximately in the direction of her utility gradient, then this procedure almost implements stochastic gradient descent for minimizing negative expected utility.

The caveat is that although the procedure can potentially obtain the direction of the gradient of the voter utilities, it cannot in general obtain any information about its magnitude since the movement norm is chosen by the procedure itself. However, we show that for certain plausible utility and voter response models, the algorithm does indeed converge to a unique point with desirable properties, including cases in which it converges to the societal optimum.

Note that with such feedback and without any additional assumptions on voter preferences (e.g. that voter utilities are normalized to the same scale), no algorithm has any hope of finding a desirable solution that depends on the cardinal values of voters' utilities, e.g., the social welfare maximizing solution (the solution that maximizes the sum of agent utilities). This is because an algorithm that uses only ordinal information about voter preferences is insensitive to any scaling or even monotonic transformations of those preferences.

317

[form-feed in source extraction]
Garg, Kamble, Goel, Marn, & Munagala

Algorithm 1: Iterative Local Voting (ILV)
Inputs: Initial solution x0 ∈ X , tolerance ϵ > 0, an integer N , initial radius r0 > 0, termination time T , norm q for local neighborhood.
Output: Solution x.

- For t ≥ 1, sample a voter vt ∈ V at random from the population; set rt = r0 /t and elicit

  x0t = arg maxx∈{s:ks−xt−1 kq ≤rt } fvt (x), (1)
  0
  and then compute xt = [xt]X , where [·]X is a projection onto space X ; i.e. ask the voter to move to her favorite point within constraints, and then project the reported point onto X .

• Stop when either $t = T$, in which case return x_T, or when $\max_{l,m\in\{t-N,\ldots,t\}} |x_l - x_m| \le \epsilon$, in which case return $x = x_t$.

[Next verbatim source excerpt]

C1 The solution space $X \subseteq R^M$ is non-empty, bounded, closed, and convex.
C2 Each voter v has a unique ideal solution $x_v \in X$.
C3 The ideal point x_v of each voter is drawn independently from a probability distribution with a bounded and measurable density function h_X.

Under this model, for a solution $x \in X$, the societal utility is given by $E_v[f_v(x)]$. and the social optimal (SO) solution is any $x^* \in \arg\max_{x\in X} E_v[f_v(x)]$.

"Convergence" of ILV refers to the convergence of the sequence of random variables $\{x_t\}_{t\ge1}$ to some $x \in X$ with probability 1, assuming that the algorithm is allowed to run indefinitely (this notion of convergence also implies the termination of the algorithm with probability 1).

In the following subsections, we present several classes of utility functions for which the algorithm converges, summarized in Table 1. We further formalize the relationship to directional equilibria in Section 3.3.

Settled source reading When the displayed Model B gradient is zero, the voter remains at the current point.

Settled source reading The checked Theorem 3 zero-field result uses an explicit full-space condition; the constrained-space result gives zero field or no feasible aggregate direction. The printed theorem invokes C1–C3, including bounded convex X; these results alone do not prove or refute zero field on every such X.

Expanded PaperInterface specification

```
∀ {Voter : Type u} {Coord : Type v} [inst : MetricSpace Voter] [inst_1 : SecondCountableTopology Voter]
  [inst_2 : MeasurableSpace Voter] [inst_3 : BorelSpace Voter] [inst_4 : StandardBorelSpace Voter]
  [inst_5 : Nonempty Voter] [inst_6 : Fintype Coord] [inst_7 : Nonempty Coord]
  {E : GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)}
  (M : GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E)
  (S : GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M) (xstar : Coord → ℝ),
  (∀ᵐ (omega : ℕ → Voter) ∂AppliedModelingLib.iidSequenceMeasure M.population,
    GKGMM19IterativeLocalVoting.PopulationTheorem3OutcomeConvergesTo S.trajectory omega xstar) →
   ∀ᵐ (omega : ℕ → Voter) ∂AppliedModelingLib.iidSequenceMeasure M.population,
    GKGMM19IterativeLocalVoting.IsDirectionalEquilibrium E xstar
```

Retained governing declarations

- AppliedModelingLib.iidSequenceMeasure
- GKGMM19IterativeLocalVoting.ILVEnvironment
- GKGMM19IterativeLocalVoting.IsDirectionalEquilibrium
- GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel
- GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource
- GKGMM19IterativeLocalVoting.PopulationTheorem3OutcomeConvergesTo

Lean-elaborated claim roles

1. parameter: Type u
2. parameter: Type v
3. parameter: MetricSpace Voter
4. assumption: SecondCountableTopology Voter
5. parameter: MeasurableSpace Voter
6. assumption: BorelSpace Voter
7. assumption: StandardBorelSpace Voter
8. assumption: Nonempty Voter
9. parameter: Fintype Coord
10. assumption: Nonempty Coord
11. parameter: GKGMM19IterativeLocalVoting.ILVEnvironment Voter (Coord → ℝ)
12. parameter: GKGMM19IterativeLocalVoting.PopulationTheorem3DirectionalFieldModel E
13. parameter: GKGMM19IterativeLocalVoting.PopulationTheorem3IidTraceSource M
14. parameter: Coord → ℝ
15. assumption: ∀ᵐ (omega : ℕ → Voter) ∂AppliedModelingLib.iidSequenceMeasure M.population, GKGMM19IterativeLocalVoting.PopulationTheorem3OutcomeConvergesTo S.trajectory omega xstar
16. conclusion: ∀ᵐ (omega : ℕ → Voter) ∂AppliedModelingLib.iidSequenceMeasure M.population, GKGMM19IterativeLocalVoting.IsDirectionalEquilibrium E xstar

Lean proof endpoint (built separately)

GKGMM19IterativeLocalVoting.theorem3_population_fullspace

Recorded source-to-Spec screening **Verdict:** matches

Reason: Under the approved full-space and zero-gradient conventions, it states that an iid Model-B trajectory converging almost surely has zero aggregate normalized directional field.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation