Human review packet: DSWG24DiscretizationBias

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

Paper: DSWG24DiscretizationBias
Paper id: DSWG24DiscretizationBias
Source version: arXiv:2405.16762; byte-pinned publication source record
Source record: audit/paper_statement_map.json
Rows: 9
Generated: 2026-09-07
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 DSWG24DiscretizationBias --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.Decision.IsPointwiseMax

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → {α : Type u_2} → (ι → α → ℝ) → (ι → α) → Prop

value:
fun {ι : Type u_1} {α : Type u_2} (score : ι → α → ℝ) (choose : ι → α) =>
  ∀ (i : ι) (a : α), score i a ≤ score i (choose i)
```

AppliedModelingLib.Decision.MonotoneExpectation

Declaration location: reusable library

Lean declaration body

```
type:
{ω : Type u_1} → ((ω → ℝ) → ℝ) → Prop

value:
fun {ω : Type u_1} (expect : (ω → ℝ) → ℝ) => ∀ (f g : ω → ℝ), (∀ (x : ω), f x ≤ g x) → expect f ≤ expect g
```

AppliedModelingLib.Decision.FiniteLinearExpectation

Declaration location: reusable library

Lean declaration body

```
type:
{ω : Type u_1} → ((ω → ℝ) → ℝ) → Prop

value:
fun {ω : Type u_1} (expect : (ω → ℝ) → ℝ) =>
  AppliedModelingLib.Decision.MonotoneExpectation expect ∧
    (∀ (c : ℝ) (f : ω → ℝ), (expect fun (x : ω) => c * f x) = c * expect f) ∧
      ∀ (f g : ω → ℝ), (expect fun (x : ω) => f x + g x) = expect f + expect g
```

Direct retained dependencies

- AppliedModelingLib.Decision.MonotoneExpectation

AppliedModelingLib.Decision.averageScore

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → {α : Type u_2} → [Fintype ι] → (ι → α → ℝ) → (ι → α) → ℝ

value:
fun {ι : Type u_1} {α : Type u_2} [Fintype ι] (score : ι → α → ℝ) (choose : ι → α) =>
  (∑ i : ι, score i (choose i)) / ↑(Fintype.card ι)
```

AppliedModelingLib.Decision.datasetAccuracy

Declaration location: reusable library

Lean declaration body

```
type:
{ι : Type u_1} → {α : Type u_2} → [Fintype ι] → [DecidableEq α] → (ι → α) → (ι → α) → ℝ

value:
fun {ι : Type u_1} {α : Type u_2} [Fintype ι] [DecidableEq α] (truth decision : ι → α) =>
  AppliedModelingLib.Decision.averageScore (fun (i : ι) (a : α) => if a = truth i then 1 else 0) decision
```

Direct retained dependencies

- AppliedModelingLib.Decision.averageScore

AppliedModelingLib.Decision.expectedDecisionAccuracy

Declaration location: reusable library

Lean declaration body

```
type:
{ω : Type u_1} →
  {σ : Type u_2} →
    {ι : Type u_3} →
      {α : Type u_4} → [Fintype ι] → [DecidableEq α] → ((ω → ℝ) → ℝ) → (ω → σ) → (ω → ι → α) → (σ → ι → α) → ℝ

value:
fun {ω : Type u_1} {σ : Type u_2} {ι : Type u_3} {α : Type u_4} [Fintype ι] [DecidableEq α] (expect : (ω → ℝ) → ℝ)
    (obs : ω → σ) (truth : ω → ι → α) (rule : σ → ι → α) =>
  expect fun (x : ω) => AppliedModelingLib.Decision.datasetAccuracy (truth x) (rule (obs x))
```

Direct retained dependencies

- AppliedModelingLib.Decision.datasetAccuracy

AppliedModelingLib.Decision.expectedObjective

Declaration location: reusable library

Lean declaration body

```
type:
{ω : Type u_1} → {σ : Type u_2} → {β : Type u_3} → ((ω → ℝ) → ℝ) → (ω → σ) → (σ → β → ℝ) → (σ → β) → ℝ

value:
fun {ω : Type u_1} {σ : Type u_2} {β : Type u_3} (expect : (ω → ℝ) → ℝ) (obs : ω → σ) (objective : σ → β → ℝ)
    (rule : σ → β) =>
  expect fun (x : ω) => objective (obs x) (rule (obs x))
```

AppliedModelingLib.Learning.Prediction.Model

Declaration location: reusable library

Lean declaration body

```
type:
Type u_1 → Type u_2 → Type (max u_1 u_2)

value:
fun (X : Type u_1) (Y : Type u_2) => X → Y
```

AppliedModelingLib.Learning.Prediction.MulticlassScore

Declaration location: reusable library

Lean declaration body

```
type:
Type u_1 → Type u_2 → Type (max u_1 u_2)

value:
fun (X : Type u_1) (Y : Type u_2) => AppliedModelingLib.Learning.Prediction.Model X (Y → ℝ)
```

Direct retained dependencies

- AppliedModelingLib.Learning.Prediction.Model

AppliedModelingLib.Learning.Prediction.IsArgmaxRule

Declaration location: reusable library

Lean declaration body

```
type:
{X : Type u_1} →
  {Y : Type u_2} →
    AppliedModelingLib.Learning.Prediction.MulticlassScore X Y → AppliedModelingLib.Learning.Prediction.Model X Y → Prop

value:
fun {X : Type u_1} {Y : Type u_2} (q : AppliedModelingLib.Learning.Prediction.MulticlassScore X Y)
    (rule : AppliedModelingLib.Learning.Prediction.Model X Y) =>
  AppliedModelingLib.Decision.IsPointwiseMax q rule
```

Direct retained dependencies

- AppliedModelingLib.Decision.IsPointwiseMax
- AppliedModelingLib.Learning.Prediction.Model
- AppliedModelingLib.Learning.Prediction.MulticlassScore

AppliedModelingLib.Learning.Prediction.IsSimplexValued

Declaration location: reusable library

Lean declaration body

```
type:
{X : Type u_1} → {Y : Type u_2} → [Fintype Y] → AppliedModelingLib.Learning.Prediction.MulticlassScore X Y → Prop

value:
fun {X : Type u_1} {Y : Type u_2} [Fintype Y] (q : AppliedModelingLib.Learning.Prediction.MulticlassScore X Y) =>
  (∀ (x : X), ∑ y : Y, q x y = 1) ∧ (∀ (x : X) (y : Y), 0 ≤ q x y) ∧ ∀ (x : X) (y : Y), q x y ≤ 1
```

Direct retained dependencies

- AppliedModelingLib.Learning.Prediction.MulticlassScore

AppliedModelingLib.pmfExp

Declaration location: reusable library

Lean declaration body

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

value:
fun {α : Type u_1} [Fintype α] (μ : PMF α) (f : α → ℝ) => ∑ a : α, (μ a).toReal * f a
```

AppliedModelingLib.pmfProb

Declaration location: reusable library

Lean declaration body

```
type:
{α : Type u_1} → [Fintype α] → PMF α → (p : α → Prop) → [DecidablePred p] → ℝ

value:
fun {α : Type u_1} [Fintype α] (μ : PMF α) (p : α → Prop) [DecidablePred p] =>
  AppliedModelingLib.pmfExp μ fun (a : α) => if p a then 1 else 0
```

Direct retained dependencies

- AppliedModelingLib.pmfExp

AppliedModelingLib.uniformPMF

Declaration location: reusable library

Lean declaration body

```
type:
(α : Type u_1) → [Fintype α] → [Nonempty α] → PMF α

value:
fun (α : Type u_1) [Fintype α] [Nonempty α] =>
  PMF.ofFintype (fun (x : α) => (↑(Fintype.card α))⁻¹) (AppliedModelingLib.uniformPMF._proof_1 α)
```

DSWG24DiscretizationBias.Finite.featureMarginal

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_3} → {Y : Type u_4} → [Fintype X] → [DecidableEq X] → [Fintype Y] → [DecidableEq Y] → PMF (X × Y) → PMF X

value:
fun {X : Type u_3} {Y : Type u_4} [Fintype X] [DecidableEq X] [Fintype Y] [DecidableEq Y] (μ : PMF (X × Y)) =>
  PMF.map Prod.fst μ
```

DSWG24DiscretizationBias.Finite.labelMarginal

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_3} → {Y : Type u_4} → [Fintype X] → [DecidableEq X] → [Fintype Y] → [DecidableEq Y] → PMF (X × Y) → PMF Y

value:
fun {X : Type u_3} {Y : Type u_4} [Fintype X] [DecidableEq X] [Fintype Y] [DecidableEq Y] (μ : PMF (X × Y)) =>
  PMF.map Prod.snd μ
```

DSWG24DiscretizationBias.Finite.paperAggregatePosterior

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_3} →
  {Y : Type u_4} → [Fintype X] → [DecidableEq X] → [Fintype Y] → [DecidableEq Y] → PMF (X × Y) → (X → Y → ℝ) → Y → ℝ

value:
fun {X : Type u_3} {Y : Type u_4} [Fintype X] [DecidableEq X] [Fintype Y] [DecidableEq Y] (μ : PMF (X × Y))
    (q : X → Y → ℝ) (y : Y) =>
  AppliedModelingLib.pmfExp (DSWG24DiscretizationBias.Finite.featureMarginal μ) fun (x : X) => q x y
```

Direct retained dependencies

- AppliedModelingLib.pmfExp
- DSWG24DiscretizationBias.Finite.featureMarginal

DSWG24DiscretizationBias.Finite.paperMarginalLabelShare

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_3} →
  {Y : Type u_4} → [Fintype X] → [DecidableEq X] → [Fintype Y] → [DecidableEq Y] → PMF (X × Y) → (X → Y) → Y → ℝ

value:
fun {X : Type u_3} {Y : Type u_4} [Fintype X] [DecidableEq X] [Fintype Y] [DecidableEq Y] (μ : PMF (X × Y))
    (rule : X → Y) (y : Y) =>
  AppliedModelingLib.pmfProb (DSWG24DiscretizationBias.Finite.featureMarginal μ) fun (x : X) => rule x = y
```

Direct retained dependencies

- AppliedModelingLib.pmfProb
- DSWG24DiscretizationBias.Finite.featureMarginal

DSWG24DiscretizationBias.Finite.paperBias

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_3} →
  {Y : Type u_4} →
    [Fintype X] → [DecidableEq X] → [Fintype Y] → [DecidableEq Y] → PMF (X × Y) → (X → Y) → (Y → ℝ) → Y → ℝ

value:
fun {X : Type u_3} {Y : Type u_4} [Fintype X] [DecidableEq X] [Fintype Y] [DecidableEq Y] (μ : PMF (X × Y))
    (rule : X → Y) (pref : Y → ℝ) (y : Y) =>
  DSWG24DiscretizationBias.Finite.paperMarginalLabelShare μ rule y - pref y
```

Direct retained dependencies

- DSWG24DiscretizationBias.Finite.paperMarginalLabelShare

DSWG24DiscretizationBias.Finite.paperAggregateBias

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_3} →
  {Y : Type u_4} →
    [Fintype X] → [DecidableEq X] → [Fintype Y] → [DecidableEq Y] → PMF (X × Y) → (X → Y → ℝ) → (X → Y) → Y → ℝ

value:
fun {X : Type u_3} {Y : Type u_4} [Fintype X] [DecidableEq X] [Fintype Y] [DecidableEq Y] (μ : PMF (X × Y))
    (q : X → Y → ℝ) (rule : X → Y) (y : Y) =>
  DSWG24DiscretizationBias.Finite.paperBias μ rule (DSWG24DiscretizationBias.Finite.paperAggregatePosterior μ q) y
```

Direct retained dependencies

- DSWG24DiscretizationBias.Finite.paperAggregatePosterior
- DSWG24DiscretizationBias.Finite.paperBias

DSWG24DiscretizationBias.Finite.paperTheorem1TightArgmax

Declaration location: paper-local

Lean declaration body

```
type:
Unit → Fin 2

value:
fun (x : Unit) => 0
```

DSWG24DiscretizationBias.Finite.paperTheorem1TightMu

Declaration location: paper-local

Lean declaration body

```
type:
PMF (Unit × Fin 2)

value:
AppliedModelingLib.uniformPMF (Unit × Fin 2)
```

Direct retained dependencies

- AppliedModelingLib.uniformPMF

DSWG24DiscretizationBias.Finite.paperTheorem1TightQ

Declaration location: paper-local

Lean declaration body

```
type:
Unit → Fin 2 → ℝ

value:
fun (x : Unit) (x_1 : Fin 2) => 1 / 2
```

DSWG24DiscretizationBias.Pareto.ParetoDominates

Declaration location: paper-local

Lean declaration body

```
type:
{Rule : Type u_1} → (Rule → ℝ) → (Rule → ℝ) → Rule → Rule → Prop

value:
fun {Rule : Type u_1} (accuracy fidelity : Rule → ℝ) (better worse : Rule) =>
  accuracy worse ≤ accuracy better ∧
    fidelity worse ≤ fidelity better ∧ (accuracy worse < accuracy better ∨ fidelity worse < fidelity better)
```

DSWG24DiscretizationBias.Pareto.ParetoOptimal

Declaration location: paper-local

Lean declaration body

```
type:
{Rule : Type u_1} → (Rule → ℝ) → (Rule → ℝ) → Rule → Prop

value:
fun {Rule : Type u_1} (accuracy fidelity : Rule → ℝ) (rule : Rule) =>
  ¬∃ (better : Rule), DSWG24DiscretizationBias.Pareto.ParetoDominates accuracy fidelity better rule
```

Direct retained dependencies

- DSWG24DiscretizationBias.Pareto.ParetoDominates

DSWG24DiscretizationBias.Pareto.weightedObjective

Declaration location: paper-local

Lean declaration body

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

value:
fun {Rule : Type u_1} (γ : ℝ) (accuracy fidelity : Rule → ℝ) (rule : Rule) =>
  γ * accuracy rule + (1 - γ) * fidelity rule
```

DSWG24DiscretizationBias.ProofBridge.aggregatePosterior

Declaration location: paper-local

Lean declaration body

```
type:
{N K : ℕ} → (Fin N → Fin K → ℝ) → Fin K → ℝ

value:
fun {N K : ℕ} (q : Fin N → Fin K → ℝ) (y : Fin K) => (∑ i : Fin N, q i y) / ↑N
```

DSWG24DiscretizationBias.ProofBridge.classifierMAE

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} →
  {Y : Type u_2} → [Fintype X] → [DecidableEq X] → [Fintype Y] → [DecidableEq Y] → PMF (X × Y) → (X → Y → ℝ) → ℝ

value:
fun {X : Type u_1} {Y : Type u_2} [Fintype X] [DecidableEq X] [Fintype Y] [DecidableEq Y] (μ : PMF (X × Y))
    (q : X → Y → ℝ) =>
  AppliedModelingLib.pmfExp (DSWG24DiscretizationBias.Finite.featureMarginal μ) fun (x : X) =>
    ∑ y : Y, q x y * (1 - q x y)
```

Direct retained dependencies

- AppliedModelingLib.pmfExp
- DSWG24DiscretizationBias.Finite.featureMarginal

DSWG24DiscretizationBias.ProofBridge.continuousJointAggregatePosterior

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} → {Y : Type u_2} → [inst : MeasurableSpace (X × Y)] → MeasureTheory.Measure (X × Y) → (X → Y → ℝ) → Y → ℝ

value:
fun {X : Type u_1} {Y : Type u_2} [MeasurableSpace (X × Y)] (μ : MeasureTheory.Measure (X × Y)) (q : X → Y → ℝ)
    (y : Y) =>
  ∫ (xy : X × Y), q xy.1 y ∂μ
```

DSWG24DiscretizationBias.ProofBridge.continuousJointAggregateBias

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} →
  {Y : Type u_2} →
    [inst : MeasurableSpace (X × Y)] → [DecidableEq Y] → MeasureTheory.Measure (X × Y) → (X → Y → ℝ) → (X → Y) → Y → ℝ

value:
fun {X : Type u_1} {Y : Type u_2} [MeasurableSpace (X × Y)] [DecidableEq Y] (μ : MeasureTheory.Measure (X × Y))
    (q : X → Y → ℝ) (rule : X → Y) (y : Y) =>
  ∫ (xy : X × Y), if rule xy.1 = y then 1 else 0 ∂μ -
    DSWG24DiscretizationBias.ProofBridge.continuousJointAggregatePosterior μ q y
```

Direct retained dependencies

- DSWG24DiscretizationBias.ProofBridge.continuousJointAggregatePosterior

DSWG24DiscretizationBias.ProofBridge.continuousJointClassifierMAE

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} →
  {Y : Type u_2} → [inst : MeasurableSpace (X × Y)] → [Fintype Y] → MeasureTheory.Measure (X × Y) → (X → Y → ℝ) → ℝ

value:
fun {X : Type u_1} {Y : Type u_2} [MeasurableSpace (X × Y)] [Fintype Y] (μ : MeasureTheory.Measure (X × Y))
    (q : X → Y → ℝ) =>
  ∫ (xy : X × Y), ∑ y : Y, q xy.1 y * (1 - q xy.1 y) ∂μ
```

DSWG24DiscretizationBias.ProofBridge.continuousJointPriorBias

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} →
  {Y : Type u_2} → [inst : MeasurableSpace (X × Y)] → [DecidableEq Y] → MeasureTheory.Measure (X × Y) → (X → Y) → Y → ℝ

value:
fun {X : Type u_1} {Y : Type u_2} [MeasurableSpace (X × Y)] [DecidableEq Y] (μ : MeasureTheory.Measure (X × Y))
    (rule : X → Y) (y : Y) =>
  ∫ (xy : X × Y), if rule xy.1 = y then 1 else 0 ∂μ - ∫ (xy : X × Y), if xy.2 = y then 1 else 0 ∂μ
```

DSWG24DiscretizationBias.ProofBridge.expectedObjective

Declaration location: paper-local

Lean declaration body

```
type:
{ω : Type u_1} →
  {σ : Type u_2} →
    {N K : ℕ} →
      [NeZero K] →
        ((ω → ℝ) → ℝ) → (ω → σ) → (ω → Fin N → Fin K) → ℝ → (σ → (Fin N → Fin K) → ℝ) → (σ → Fin N → Fin K) → ℝ
```

value:
```
fun {ω : Type u_1} {σ : Type u_2} {N K : ℕ} [NeZero K] (expect : (ω → ℝ) → ℝ) (observedDataset : ω → σ)
    (trueLabels : ω → Fin N → Fin K) (γ : ℝ) (fidelityTerm : σ → (Fin N → Fin K) → ℝ) (rule : σ → Fin N → Fin K) =>
  γ * AppliedModelingLib.Decision.expectedDecisionAccuracy expect observedDataset trueLabels rule +
    (1 - γ) * AppliedModelingLib.Decision.expectedObjective expect observedDataset fidelityTerm rule
```

Direct retained dependencies

- AppliedModelingLib.Decision.expectedDecisionAccuracy
- AppliedModelingLib.Decision.expectedObjective

DSWG24DiscretizationBias.ProofBridge.isFirstArgmax

Declaration location: paper-local

Lean declaration body

type:
```
{K : ℕ} → (Fin K → ℝ) → Fin K → Prop
```

value:
```
fun {K : ℕ} (score : Fin K → ℝ) (chosen : Fin K) =>
  (∀ (y : Fin K), score y ≤ score chosen) ∧ ∀ (y : Fin K), score y = score chosen → ↑chosen ≤ ↑y
```

DSWG24DiscretizationBias.ProofBridge.marginalLabelShare

Declaration location: paper-local

Lean declaration body

type:
```
{N K : ℕ} → (Fin N → Fin K) → Fin K → ℝ
```

value:
```
fun {N K : ℕ} (decision : Fin N → Fin K) (y : Fin K) => (∑ i : Fin N, if decision i = y then 1 else 0) / ↑N
```

DSWG24DiscretizationBias.ProofBridge.bias

Declaration location: paper-local

Lean declaration body

type:
```
{N K : ℕ} → (Fin N → Fin K) → (Fin K → ℝ) → Fin K → ℝ
```

value:
```
fun {N K : ℕ} (decision : Fin N → Fin K) (pref : Fin K → ℝ) (y : Fin K) =>
  DSWG24DiscretizationBias.ProofBridge.marginalLabelShare decision y - pref y
```

Direct retained dependencies

- DSWG24DiscretizationBias.ProofBridge.marginalLabelShare

DSWG24DiscretizationBias.ProofBridge.fidelity

Declaration location: paper-local

Lean declaration body

type:
```
{N K : ℕ} → (Fin N → Fin K) → (Fin K → ℝ) → ℝ
```

value:
```
fun {N K : ℕ} (decision : Fin N → Fin K) (pref : Fin K → ℝ) =>
  -∑ y : Fin K, |DSWG24DiscretizationBias.ProofBridge.bias decision pref y|
```

Direct retained dependencies

- DSWG24DiscretizationBias.ProofBridge.bias

DSWG24DiscretizationBias.ProofBridge.equation1Objective

Declaration location: paper-local

Lean declaration body

```
type:
{N K : ℕ} → ℝ → (Fin N → Fin K → ℝ) → (Fin K → ℝ) → (Fin N → Fin K) → ℝ

value:
fun {N K : ℕ} (γ : ℝ) (q : Fin N → Fin K → ℝ) (pref : Fin K → ℝ) (decision : Fin N → Fin K) =>
  γ * AppliedModelingLib.Decision.averageScore q decision +
    (1 - γ) * DSWG24DiscretizationBias.ProofBridge.fidelity decision pref
```

Direct retained dependencies

- AppliedModelingLib.Decision.averageScore
- DSWG24DiscretizationBias.ProofBridge.fidelity

DSWG24DiscretizationBias.ProofBridge.perfectClassifierOnSupport

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} →
  {Y : Type u_2} →
    [Fintype X] → [DecidableEq X] → [Fintype Y] → [DecidableEq Y] → PMF (X × Y) → (X → Y → ℝ) → (X → Y) → Prop

value:
fun {X : Type u_1} {Y : Type u_2} [Fintype X] [DecidableEq X] [Fintype Y] [DecidableEq Y] (μ : PMF (X × Y))
    (q : X → Y → ℝ) (rule : X → Y) =>
  ∀ (x : X) (y : Y), μ (x, y) ≠ 0 → rule x = y ∧ q x y = 1 ∧ ∀ (a : Y), a ≠ y → q x a = 0
```

DSWG24DiscretizationBias.ProofBridge.prior

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} → {Y : Type u_2} → [Fintype X] → [DecidableEq X] → [Fintype Y] → [DecidableEq Y] → PMF (X × Y) → Y → ℝ

value:
fun {X : Type u_1} {Y : Type u_2} [Fintype X] [DecidableEq X] [Fintype Y] [DecidableEq Y] (μ : PMF (X × Y)) (y : Y) =>
  ((DSWG24DiscretizationBias.Finite.labelMarginal μ) y).toReal
```

Direct retained dependencies

- DSWG24DiscretizationBias.Finite.labelMarginal

DSWG24DiscretizationBias.ProofBridge.sourcePNq

Declaration location: paper-local

Lean declaration body

```
type:
{Sample : Type u_1} → {N K : ℕ} → (Sample → Fin N → Fin K → ℝ) → (Sample → Fin K) → (Sample → Fin K → ℝ) → Prop

value:
fun {Sample : Type u_1} {N K : ℕ} (posterior : Sample → Fin N → Fin K → ℝ) (aggregateArgmax : Sample → Fin K)
    (prefAt : Sample → Fin K → ℝ) =>
  DSWG24DiscretizationBias.ProofBridge.referenceSimplexAt prefAt ∧
    (∀ (sample : Sample),
        DSWG24DiscretizationBias.ProofBridge.isAggregateFirstArgmax (posterior sample) (aggregateArgmax sample)) ∧
      ∀ (sample : Sample), 1 / ↑N ≤ prefAt sample (aggregateArgmax sample)
```

Direct retained dependencies

- DSWG24DiscretizationBias.ProofBridge.isAggregateFirstArgmax
- DSWG24DiscretizationBias.ProofBridge.referenceSimplexAt

DSWG24DiscretizationBias.Theorem2iii.datasetAccuracyScore

Declaration location: paper-local

Lean declaration body

```
type:
{N K : ℕ} → (Fin N → Fin K → ℝ) → (Fin N → Fin K) → ℝ

value:
fun {N K : ℕ} (posterior : Fin N → Fin K → ℝ) (decision : Fin N → Fin K) =>
  AppliedModelingLib.Decision.averageScore posterior decision
```

Direct retained dependencies

- AppliedModelingLib.Decision.averageScore

DSWG24DiscretizationBias.Theorem2iii.datasetLabelShare

Declaration location: paper-local

Lean declaration body

```
type:
{N K : ℕ} → (Fin N → Fin K) → Fin K → ℝ

value:
fun {N K : ℕ} (decision : Fin N → Fin K) (y : Fin K) => (∑ i : Fin N, if decision i = y then 1 else 0) / ↑N
```

DSWG24DiscretizationBias.Theorem2iii.datasetFidelity

Declaration location: paper-local

Lean declaration body

```
type:
{N K : ℕ} → (Fin K → ℝ) → (Fin N → Fin K) → ℝ

value:
fun {N K : ℕ} (pref : Fin K → ℝ) (decision : Fin N → Fin K) =>
  -∑ y : Fin K, |DSWG24DiscretizationBias.Theorem2iii.datasetLabelShare decision y - pref y|
```

Direct retained dependencies

- DSWG24DiscretizationBias.Theorem2iii.datasetLabelShare

DSWG24DiscretizationBias.Theorem2iii.expectedDatasetAccuracy

Declaration location: paper-local

Lean declaration body

```
type:
{Sample : Type u_1} →
  [Fintype Sample] →
    [DecidableEq Sample] → {N K : ℕ} → PMF Sample → (Sample → Fin N → Fin K → ℝ) → (Sample → Fin N → Fin K) → ℝ

value:
fun {Sample : Type u_1} [Fintype Sample] [DecidableEq Sample] {N K : ℕ} (sampleLaw : PMF Sample)
    (posterior : Sample → Fin N → Fin K → ℝ) (policy : Sample → Fin N → Fin K) =>
  AppliedModelingLib.pmfExp sampleLaw fun (sample : Sample) =>
    DSWG24DiscretizationBias.Theorem2iii.datasetAccuracyScore (posterior sample) (policy sample)
```

Direct retained dependencies

- AppliedModelingLib.pmfExp
- DSWG24DiscretizationBias.Theorem2iii.datasetAccuracyScore

DSWG24DiscretizationBias.Theorem2iii.expectedDatasetFidelityAt

Declaration location: paper-local

Lean declaration body

```
type:
{Sample : Type u_1} →
  [Fintype Sample] → [DecidableEq Sample] → {N K : ℕ} → PMF Sample → (Sample → Fin K → ℝ) → (Sample → Fin N → Fin K) → ℝ

value:
fun {Sample : Type u_1} [Fintype Sample] [DecidableEq Sample] {N K : ℕ} (sampleLaw : PMF Sample)
    (prefAt : Sample → Fin K → ℝ) (policy : Sample → Fin N → Fin K) =>
  AppliedModelingLib.pmfExp sampleLaw fun (sample : Sample) =>
    DSWG24DiscretizationBias.Theorem2iii.datasetFidelity (prefAt sample) (policy sample)
```

Direct retained dependencies

- AppliedModelingLib.pmfExp
- DSWG24DiscretizationBias.Theorem2iii.datasetFidelity

DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} → [Fintype X] → [DecidableEq X] → PMF X → (N : ℕ) → PMF (Fin N → X)

value:
fun {X : Type u_1} [Fintype X] [DecidableEq X] (μ : PMF X) (N : ℕ) =>
  PMF.ofFintype (fun (sample : Fin N → X) => ∏ i : Fin N, μ (sample i))
    (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF._proof_1 μ N)
```

DSWG24DiscretizationBias.Theorem2iii.sampledDecision

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} → {N K : ℕ} → (X → Fin K) → (Fin N → X) → Fin N → Fin K

value:
fun {X : Type u_1} {N K : ℕ} (rule : X → Fin K) (sample : Fin N → X) (a : Fin N) => rule (sample a)
```

DSWG24DiscretizationBias.Theorem2iii.sampledPosterior

Declaration location: paper-local

Lean declaration body

```
type:
{X : Type u_1} → {N K : ℕ} → (X → Fin K → ℝ) → (Fin N → X) → Fin N → Fin K → ℝ

value:
fun {X : Type u_1} {N K : ℕ} (posterior : X → Fin K → ℝ) (sample : Fin N → X) (a : Fin N) (a_1 : Fin K) =>
  posterior (sample a) a_1
```

Paper-specific semantic prerequisites

DSWG24DiscretizationBias.ProofBridge.bayesOptimal

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.bayesOptimal
Source locator: cited publication, lines 597-607

Verbatim paper-source connection

3.1.1 Continuous classifier and discretization rules
Suppose we have a machine learning classifier q (e.g., trained on some other dataset) that makes continuous probability predictions for each class, given the data. Let q(y, x) denote the probability placed on class y

10

[form-feed in source extraction]
given data x. For example, the Bayes optimal classifier is
q ∗ (y, x) ≜ Pr(Y = y|X = x).
We will refer to p as the prior and q as our learned posterior. Let the simplex of possible probability vectors over the support Y be △Y , and let q(x) ∈ △Y be the vector of probabilities (q(y, x))y .

Lean-expanded paper semantic target

```
type:
{Ω : Type u_1} →
  {σ : Type u_2} →
    [Fintype Ω] →
      [DecidableEq Ω] →
        [Fintype σ] →
          [DecidableEq σ] → {N K : ℕ} → PMF Ω → (Ω → σ) → (Ω → Fin N → Fin K) → (σ → Fin N → Fin K → ℝ) → Prop

value:
fun {Ω : Type u_1} {σ : Type u_2} [Fintype Ω] [DecidableEq Ω] [Fintype σ] [DecidableEq σ] {N K : ℕ} (μ : PMF Ω)
    (observedDataset : Ω → σ) (trueLabels : Ω → Fin N → Fin K) (posterior : σ → Fin N → Fin K → ℝ) =>
  ∀ (xs : σ) (i : Fin N) (y : Fin K),
    (AppliedModelingLib.pmfProb μ fun (w : Ω) => observedDataset w = xs ∧ trueLabels w i = y) =
      posterior xs i y * AppliedModelingLib.pmfProb μ fun (w : Ω) => observedDataset w = xs
```

Direct retained dependencies

- AppliedModelingLib.pmfProb

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

Reason: The source defines the Bayes classifier by q-star(y,x)=Pr(Y=y given X=x); the expanded finite target states the equivalent atom factorization Pr(X=xs and Y_i=y)=posterior(xs,i,y) times Pr(X=xs), preserving every positive-mass conditional and leaving zero-mass conditionals unconstrained.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.calibrated

Paper-local declaration:

DSWG24DiscretizationBias.ProofBridge.calibrated
Source locator: cited publication, lines 635-658

Verbatim paper-source connection

3.2 Theoretical Results: Argmax bias and continuous classifier
We first study the relationship between argmax bias and the properties of the continuous classifier. We show that discretization bias can occur even with unbiased (calibrated) continuous classifiers q, as long as it is uncertain (has non-zero error).
The following result holds for all calibrated classifiers. A given classifier q is calibrated when its continuous predictions are correct on average, Pr(Y = y|q(y, x) = c) = c.10 Note that the Bayes optimal classifier is calibrated.
Theorem 1. Argmax bias depends on predictive uncertainty, i.e., how much information features x provide about the true class label. Consider calibrated classifier q and the argmax decision rule Dargmax . Let N > K and consider a reference distribution of either the aggregate posterior or the prior. Then,
(i). Suppose x provides no information and q(y, x) = Pr(y) for all x, y. Then, amplification bias is maximized and all data points are classified as a plurality class:
(
1 − Pr(y), if y = arg maxw p(w),
BIAS(y, Dargmax) =
− Pr(y), otherwise.
9 Note that, though related, mean average error of the continuous predictor q(y, x) is not equivalent to expected zero-one

loss of the discrete decisions. The latter, our measure of decision accuracy, also depends on how the continuous predictor is discretized. n o
10 For all c of non-zero measure: ∀c ∈
R
c : (x,y) I[q(y, x) = c]f (x, y)d(y, x) > 0 . Further note that calibration implies
Bayesian consistency, i.e., the continuous model cumulatively assigns the proper mass to each class y: EX [q(y, x)] = Pr(y).

Settled source reading

Calibration uses the standard conditional-expectation event identity intended by the source proof.

Lean-expanded paper semantic target

```
type:
{X : Type u_1} →
  {Y : Type u_2} →
    [inst : MeasurableSpace (X × Y)] → [DecidableEq Y] → MeasureTheory.Measure (X × Y) → (X → Y → ℝ) → Prop

value:
fun {X : Type u_1} {Y : Type u_2} [MeasurableSpace (X × Y)] [DecidableEq Y] (μ : MeasureTheory.Measure (X × Y))
    (q : X → Y → ℝ) =>
  ∀ (y : Y) (s : Set ℝ),
    MeasurableSet s →
      ∫ (xy : X × Y), if q xy.1 y ∈ s then if xy.2 = y then 1 else 0 else 0 ∂μ =
        ∫ (xy : X × Y), if q xy.1 y ∈ s then q xy.1 y else 0 ∂μ
```

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

Reason: The approved source reading requires, for every label y and measurable score-value event A, equality between the mass of observations with Y = y and q_y in A and the integral of q_y over that same event. The expanded calibrated predicate states exactly this event-preimage integral identity for every y and measurable set s. Its displayed use in the Theorem 1(iii) Spec adds posteriorSimplex and coordinate measurability before assuming calibrated, so the paper use remains within the approved probabilistic score domain.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.isAggregateFirstArgmax

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.isAggregateFirstArgmax
Source locator: cited publication, lines 749-755

Verbatim paper-source connection

q
For the result, we define the notion of non-trivial reference distributions PN : a reference distribution
1
pq : {xi } → △NY is in P N if it assigns mass at least N to the most likely class in the dataset, i.e., at least 1
data point would be discretized to the most likely class arg maxy i q(y, xi).13
P

[Next verbatim source excerpt]

Aggregate posterior reference distribution What is an appropriate reference distribution pref ? Of course, if we knew the true distribution of labels (e.g., the true fraction of each group in the voter file), we could measure and optimize fidelity with respect to it. However, this information is not often known. In this paper, we will thus primarily compare to the aggregate posterior: for a given set {xi }, what would the decision distribution be if we could make continuous decisions corresponding to the continuous classifier q,

1 X
pqagg (y, {xi }) = q(y, xi).
N i

Lean-expanded paper semantic target

type:
{N K : ℕ} → (Fin N → Fin K → ℝ) → Fin K → Prop

value:
fun {N K : ℕ} (posterior : Fin N → Fin K → ℝ) (target : Fin K) =>
DSWG24DiscretizationBias.ProofBridge.isFirstArgmax (DSWG24DiscretizationBias.ProofBridge.aggregatePosterior posterior)
target

Direct retained dependencies

- DSWG24DiscretizationBias.ProofBridge.aggregatePosterior
- DSWG24DiscretizationBias.ProofBridge.isFirstArgmax

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

Reason: The source's dataset-most-likely class maximizes the sum of row posteriors, with a consistent tie order; the expanded target takes the first maximizer of the aggregate posterior, and division by positive dataset size does not change that argmax in the displayed N>K uses.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.isArgmaxRule

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.isArgmaxRule
Source locator: cited publication, lines 218

Verbatim paper-source connection

Argmax rule assigns, for each data point, the most likely2 class: Dargmax ({q(xi }) ≜ {ŷi = arg maxy q(y, xi)}.

[Next verbatim source excerpt]

2 Suppose ties are broken according to some consistent ordering over the classes. In the event of ties, we will use arg maxy q(y, x) to denote the single class that is in the argmax set and is first in the consistent ordering among classes in the argmax set.

Lean-expanded paper semantic target

type:
{X : Type u_1} → {Y : Type u_2} → (X → Y → ℝ) → (X → Y) → Prop

value:
fun {X : Type u_1} {Y : Type u_2} (q : X → Y → ℝ) (rule : X → Y) =>
  AppliedModelingLib.Learning.Prediction.IsArgmaxRule q rule

Direct retained dependencies

- AppliedModelingLib.Learning.Prediction.IsArgmaxRule

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

Reason: The expanded predicate requires the selected label to attain the row-wise maximum. It relaxes the source's fixed tie ordering, but every source tie-broken argmax satisfies it and the displayed theorem1iii use therefore proves a valid strengthening covering every source-admissible rule without changing behavior on that domain.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.isIndependentRandomizedRule

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.isIndependentRandomizedRule
Source locator: cited publication, lines 615-618

Verbatim paper-source connection

Decision (discretization) rules are as defined in Section 1.1. Our theoretical results distinguish between independent (e.g., argmax, threhsolding, Thompson sampling) and joint (e.g., our optimization approach) rules. Formally, with independent rules, there exists a (potentially randomized) function d such that Dd ({q(xi)}) = {ŷi = d(q(y, xi))}.

Lean-expanded paper semantic target

```
type:
{X : Type u_1} → {N K : ℕ} → ((Fin N → X) → (Fin N → Fin K) → ℝ) → Prop

value:
fun {X : Type u_1} {N K : ℕ} (jointProbability : (Fin N → X) → (Fin N → Fin K) → ℝ) =>
  ∃ (rowProbability : X → Fin K → ℝ),
    (∀ (x : X) (y : Fin K), 0 ≤ rowProbability x y) ∧
      (∀ (x : X), ∑ y : Fin K, rowProbability x y = 1) ∧
        ∀ (xs : Fin N → X) (decision : Fin N → Fin K),
          jointProbability xs decision = ∏ i : Fin N, rowProbability (xs i) (decision i)
```

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

Reason: The source defines an independent potentially randomized rule by one common row-level randomized function d; the expanded target represents d as a nonnegative normalized row kernel and factors the conditional joint decision probability into the product of its row probabilities.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.isIndependentRule

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.isIndependentRule
Source locator: cited publication, lines 225-228

Verbatim paper-source connection

These rules are all independent: the assigned label for a data point i depends on $q(x_i)$ but not on other data points or their probability vectors $\{q(x_j)\}_{j\neq i}$.
In contrast, our proposed approach also includes joint decision optimization rules, which depend on the entire set of data points, as a solution to discretization bias.

[Next verbatim source excerpt]

data point x_i. A (potentially randomized) decision (discretization) rule $D : \triangle_Y^N \to Y^N$ is a rule that, given a set of probability vectors $\{q(x_i)\}_{i=1}^N$ associated with N data points, assigns each data point i a label $\hat{y}_i \in Y$.

[Next verbatim source excerpt]

Decision (discretization) rules are as defined in Section 1.1. Our theoretical results distinguish between independent (e.g., argmax, threhsolding, Thompson sampling) and joint (e.g., our optimization approach) rules. Formally, with independent rules, there exists a (potentially randomized) function d such that $D_d(\{q(x_i)\}) = \{\hat{y}_i = d(q(y, x_i))\}$.

Lean-expanded paper semantic target

type:
{X : Type u_1} → {N K : ℕ} → ((Fin N → X) → Fin N → Fin K) → Prop

value:
fun {X : Type u_1} {N K : ℕ} (rule : (Fin N → X) → Fin N → Fin K) =>
  ∃ (d : X → Fin K), ∀ (xs : Fin N → X) (i : Fin N), rule xs i = d (xs i)

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

Reason: The source says each assigned label depends only on its own row probability vector through one common function d; the expanded deterministic target is exactly the existence of d with rule(xs,i)=d(xs i) for every dataset and row.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.isThompsonSamplingRule

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.isThompsonSamplingRule
Source locator: cited publication, lines 222-223

Verbatim paper-source connection

Thompson sampling for each data point i, samples a class y based on the probabilities q(y, xi), i.e., ŷi = y with probability q(y, xi), and so in expectation will have labels matching the aggregate posterior.

Lean-expanded paper semantic target

```
type:
{N K : ℕ} → (Fin N → Fin K → ℝ) → (Fin N → Fin K → ℝ) → Prop

value:
fun {N K : ℕ} (q selectionProbability : Fin N → Fin K → ℝ) =>
  ∀ (i : Fin N) (y : Fin K), selectionProbability i y = q i y
```

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

Reason: The source defines Thompson sampling by selecting label y for row i with probability q(y,x_i); the expanded target equates every row-label selection probability with the corresponding posterior probability.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.isTieBrokenArgmaxRule

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.isTieBrokenArgmaxRule
Source locator: cited publication, lines 215-231

Verbatim paper-source connection

1.1.2 Overview of decision rules
Several rules are used in academia and in practice:

Argmax rule assigns, for each data point, the most likely2 class: $D_{argmax}(\{q(x_i\}) \triangleq \{\hat{y}_i = \arg\max_y q(y, x_i)\}$.
Threshold at t rule assigns a label only if the argmax class has a probability of at least t, i.e., $\hat{y}_i = \arg\max_y q(y, x_i)$ if $\max_y q(y, x_i) \geq t$, otherwise i is left uncoded.

Thompson sampling for each data point i, samples a class y based on the probabilities $q(y, x_i)$, i.e., $\hat{y}_i = y$ with probability $q(y, x_i)$, and so in expectation will have labels matching the aggregate posterior.

These rules are all independent: the assigned label for a data point i depends on $q(x_i)$ but not on other data points or their probability vectors $\{q(x_j)\}_{j\neq i}$.
In contrast, our proposed approach also includes joint decision optimization rules, which depend on the entire set of data points, as a solution to discretization bias.
2 Suppose ties are broken according to some consistent ordering over the classes. In the event of ties, we will use $\arg\max_y q(y, x)$ to denote the single class that is in the argmax set and is first in the consistent ordering among classes in the argmax set.

[Next verbatim source excerpt]

Argmax rule assigns, for each data point, the most likely2 class: $D_{argmax}(\{q(x_i\}) \triangleq \{\hat{y}_i = \arg\max_y q(y, x_i)\}$.
Threshold at t rule assigns a label only if the argmax class has a probability of at least t, i.e., $\hat{y}_i = \arg\max_y q(y, x_i)$ if $\max_y q(y, x_i) \geq t$, otherwise i is left uncoded.

Thompson sampling for each data point i, samples a class y based on the probabilities $q(y, x_i)$, i.e., $\hat{y}_i = y$ with probability $q(y, x_i)$, and so in expectation will have labels matching the aggregate posterior.

These rules are all independent: the assigned label for a data point i depends on $q(x_i)$ but not on other data points or their probability vectors $\{q(x_j)\}_{j\neq i}$.
In contrast, our proposed approach also includes joint decision optimization rules, which depend on the entire set of data points, as a solution to discretization bias.
2 Suppose ties are broken according to some consistent ordering over the classes. In the event of ties, we will use $\arg\max_y q(y, x)$ to denote the single class that is in the argmax set and is first in the consistent ordering among classes in the argmax set.

Lean-expanded paper semantic target

```
type:
{N K : ℕ} → (Fin N → Fin K → ℝ) → (Fin N → Fin K) → Prop

value:
fun {N K : ℕ} (q : Fin N → Fin K → ℝ) (rule : Fin N → Fin K) =>
  ∀ (i : Fin N), DSWG24DiscretizationBias.ProofBridge.isFirstArgmax (q i) (rule i)
```

Direct retained dependencies

- DSWG24DiscretizationBias.ProofBridge.isFirstArgmax

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

Reason: The source chooses a row-wise maximum and resolves ties by a fixed consistent class ordering; the expanded target uses the natural Fin ordering, requires maximal score, and requires the chosen index to be first among all tied maxima for every row.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.maximizesEquation1

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.maximizesEquation1
Source locator: cited publication, lines 252-264

Verbatim paper-source connection

Integer optimization rules directly optimize given objectives. We consider the rule that corresponds to solving the following optimization problem, for a given γ, data points $\{x_i\}$, and reference p_{ref}:

$$D^{\gamma}(\{q(x_i)\}, p_{ref}) = \arg\max_{\{\hat{y}_i\}} \left(\gamma \frac{1}{N} \sum_{i=1}^{N} q(\hat{y}_i, x_i) + (1-\gamma)\text{fid}(p_{ref}, \hat{y}_{1:N}) \right). \quad (1)$$

The rule balances, parameterized by $\gamma \in [0, 1]$, row-level scores (accuracy as measured by q) and distributional fidelity.
Aggregate posterior matching A special case of the joint optimization decision rules defined in Equation (1) is as $\gamma \to 0$, leading to a rule that maximizes accuracy subject to the constraint that distributional fidelity be as high as possible. We call such rules matching solutions, e.g., aggregate posterior

[Next verbatim source excerpt]

[form-feed in source extraction]
(2a) To measure bias, we use the distance between the marginal distribution of labels and a reference distribution p_{ref}. Let $\hat{p}_{marg}(y)$ denote the fraction of datapoints with the label y:

$$\hat{p}_{marg}(y, \hat{y}_{1:N}) = \frac{1}{N} \sum_{i=1}^{N} 1[\hat{y}_i = y].$$

Then, bias is how much a class y is amplified by labels relative to the reference:

$$\text{bias}(y, \hat{y}_{1:N}, p_{ref}) \triangleq \hat{p}_{marg}(y, \hat{y}_{1:N}) - p_{ref}(y).$$

(2b) As a summary statistic, we will further consider distributional fidelity, the negative sum of the absolute value of bias across classes (equivalently, the negative of the ell1 distance between the label and reference distributions).

$$\text{fid}(p_{ref}, \hat{y}_{1:N}) \triangleq - \sum_{y \in \mathcal{Y}} \text{bias}(y, \hat{y}_{1:N}, p_{ref}).$$

Lean-expanded paper semantic target

```
type:
{N K : ℕ} → ℝ → (Fin N → Fin K → ℝ) → (Fin K → ℝ) → (Fin N → Fin K) → Prop

value:
fun {N K : ℕ} (γ : ℝ) (q : Fin N → Fin K → ℝ) (pref : Fin K → ℝ) (decision : Fin N → Fin K) =>
  ∀ (other : Fin N → Fin K),
    DSWG24DiscretizationBias.ProofBridge.equation1Objective γ q pref other ≤
      DSWG24DiscretizationBias.ProofBridge.equation1Objective γ q pref decision
```

Direct retained dependencies

- DSWG24DiscretizationBias.ProofBridge.equation1Objective

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

Reason: Equation (1) maximizes gamma times average selected posterior score plus one-minus-gamma times fidelity, where fidelity is the negative L1 bias; the expanded dependencies implement those formulas and the target compares the chosen decision against every alternative with the correct maximizing inequality.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.posteriorSimplex

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.posteriorSimplex
Source locator: cited publication, lines 597-607

Verbatim paper-source connection

3.1.1 Continuous classifier and discretization rules
Suppose we have a machine learning classifier q (e.g., trained on some other dataset) that makes continuous probability predictions for each class, given the data. Let q(y, x) denote the probability placed on class y

10

[form-feed in source extraction]
given data x. For example, the Bayes optimal classifier is
$q*(y, x) \triangleq \Pr(Y = y|X = x)$.
We will refer to p as the prior and q as our learned posterior. Let the simplex of possible probability vectors over the support Y be △Y , and let q(x) ∈ △Y be the vector of probabilities (q(y, x))y .

Lean-expanded paper semantic target

type:
{X : Type u_1} → {Y : Type u_2} → [Fintype Y] → (X → Y → ℝ) → Prop

value:
fun {X : Type u_1} {Y : Type u_2} [Fintype Y] (q : X → Y → ℝ) =>
  AppliedModelingLib.Learning.Prediction.IsSimplexValued q

Direct retained dependencies

- AppliedModelingLib.Learning.Prediction.IsSimplexValued

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

Reason: The source places every posterior vector q(x) in the label simplex; the expanded IsSimplexValued predicate requires its coordinates to be nonnegative, sum to one, and lie at most one, exactly the probability-vector condition.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

DSWG24DiscretizationBias.ProofBridge.referenceSimplexAt

Paper-local declaration:
DSWG24DiscretizationBias.ProofBridge.referenceSimplexAt
Source locator: cited publication, lines 749-755

Verbatim paper-source connection

q
For the result, we define the notion of non-trivial reference distributions PN : a reference distribution
1
pq : {xi } → △NY is in P N if it assigns mass at least N to the most likely class in the dataset, i.e., at least 1
data point would be discretized to the most likely class arg maxy i q(y, xi).13
P

Lean-expanded paper semantic target

type:
{Sample : Type u_1} → {K : ℕ} → (Sample → Fin K → ℝ) → Prop

value:
fun {Sample : Type u_1} {K : ℕ} (prefAt : Sample → Fin K → ℝ) =>
∀ (sample : Sample), (∀ (y : Fin K), 0 ≤ prefAt sample y) ∧ ∑ y : Fin K, prefAt sample y = 1

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

Reason: The source's dataset-indexed reference distribution takes values in the label simplex; the expanded target states pointwise nonnegativity and unit total mass, while the displayed sourcePNq use separately supplies the required at-least-1/N mass on the tie-broken aggregate-posterior maximizer.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Source claims

Review row 1

Source locator: cited publication, lines 642-666

Verbatim source input

```
Theorem 1. Argmax bias depends on predictive uncertainty, i.e., how much information features x provide
about the true class label. Consider calibrated classifier q and the argmax decision rule Dargmax . Let N > K
and consider a reference distribution of either the aggregate posterior or the prior. Then,
  (i). Suppose x provides no information and q(y, x) = Pr(y) for all x, y. Then, amplification bias is
      maximized and all data points are classified as a plurality class:
                                          (
                                            1 − Pr(y), if y = arg maxw p(w),
                        BIAS(y, Dargmax ) =
                                            − Pr(y),      otherwise.
   9 Note that, though related, mean average error of the continuous predictor q(y, x) is not equivalent to expected zero-one

loss of the discrete decisions. The latter, our measure of decision accuracy, also depends on how the continuous predictor is
discretized.                              n                                     o
  10 For all c of non-zero measure: ∀c ∈
                                         R
                                     c : (x,y) I[q(y, x) = c]f (x, y)d(y, x) > 0 . Further note that calibration implies
Bayesian consistency, i.e., the continuous model cumulatively assigns the proper mass to each class y: EX [q(y, x)] = Pr(y).


                                                   11

[form-feed in source extraction]
                                                                              (
                                                                               1,                                        if z = y,
  (ii). Suppose the classifier is perfect, i.e., ∀(x, y) ∼ FXY , we have q(z, x) =                                                                     Then, there
                                                                               0,                                        otherwise.
       is no amplification bias, BIAS(y, Dargmax ) = 0.
```

Expanded PaperInterface specification

```
∀ {X : Type u_1} {Y : Type u_2} [inst : Fintype X] [inst_1 : DecidableEq X] [inst_2 : Fintype Y]
  [inst_3 : DecidableEq Y] (μ : PMF (X × Y)) (q : X → Y → ℝ) (rule : X → Y) (y : Y),
  DSWG24DiscretizationBias.ProofBridge.perfectClassifierOnSupport μ q rule →
    DSWG24DiscretizationBias.Finite.paperBias μ rule (DSWG24DiscretizationBias.ProofBridge.prior μ) y = 0 ∧
      DSWG24DiscretizationBias.Finite.paperBias μ rule (DSWG24DiscretizationBias.Finite.paperAggregatePosterior μ q) y =
        0
```

Retained governing declarations

- DSWG24DiscretizationBias.Finite.paperAggregatePosterior
- DSWG24DiscretizationBias.Finite.paperBias
- DSWG24DiscretizationBias.ProofBridge.perfectClassifierOnSupport
- DSWG24DiscretizationBias.ProofBridge.prior

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Fintype X
4. parameter: DecidableEq X
5. parameter: Fintype Y
6. parameter: DecidableEq Y
7. parameter: PMF (X × Y)
8. parameter: X → Y → ℝ
9. parameter: X → Y
10. parameter: Y
11. assumption: DSWG24DiscretizationBias.ProofBridge.perfectClassifierOnSupport μ q rule
12. conclusion: DSWG24DiscretizationBias.Finite.paperBias μ rule (DSWG24DiscretizationBias.ProofBridge.prior μ) y = 0 ∧
  DSWG24DiscretizationBias.Finite.paperBias μ rule (DSWG24DiscretizationBias.Finite.paperAggregatePosterior μ q) y = 0
```

Lean proof endpoint (built separately)

```
DSWG24DiscretizationBias.PaperInterface.theorem1ii_perfect_classifier_zero_bias
```

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

Reason: The support-sensitive perfect-classifier premise says that the induced decision is the true label and the score vector is one-hot on every positive-probability joint atom. The target then concludes zero amplification bias for every label under both the prior and aggregate-posterior references, exactly as in Theorem 1(ii).

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 2

Source locator: cited publication, lines 756-763

Verbatim source input

```
Theorem 2. Consider Bayes optimal q, and N > K. For all F :
   (i). For every γ, pref there exists a joint decision-making rule that maximizes Oγ (D, q, pref ) in Equa-
        tion (3).

  (ii). The argmax decision rule Dargmax maximizes O1 (D, q, pref ), i.e., is accuracy maximizing.
 (iii). Dargmax is the only Pareto optimal independent decision rule for non-trivial pref ∈ P. No independent
        decision rule D maximizes objective Oγ for any γ, unless γ = 1 and D agrees with Dargmax with
        probability 1.
```

Expanded PaperInterface specification

```
∀ {Z : Type u_1} {X : Type u_2} [inst : Fintype Z] [inst_1 : DecidableEq Z] {N K : ℕ} (ν : PMF Z) (feature : Z → X)
  (posterior : X → Fin K → ℝ) (rule : Z → Fin K) (argmaxRule : X → Fin K) (aggregateArgmax : (Fin N → X) → Fin K)
  (prefAt : (Fin N → X) → Fin K → ℝ),
  2 ≤ K →
    K < N →
      (∀ (x : X), DSWG24DiscretizationBias.ProofBridge.isFirstArgmax (posterior x) (argmaxRule x)) →
        (0 < AppliedModelingLib.pmfProb ν fun (z : Z) => rule z ≠ argmaxRule (feature z)) →
          DSWG24DiscretizationBias.ProofBridge.sourcePNq
              (fun (sample : Fin N → X) => DSWG24DiscretizationBias.Theorem2iii.sampledPosterior posterior sample)
              aggregateArgmax prefAt →
            0 < N →
              ¬DSWG24DiscretizationBias.Pareto.ParetoOptimal
                  (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetAccuracy
                    (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
                    DSWG24DiscretizationBias.Theorem2iii.sampledPosterior
                      (fun (z : Z) (y : Fin K) => posterior (feature z) y) sample)
                  (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetFidelityAt
                    (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
                    prefAt fun (i : Fin N) => feature (sample i))
                  fun (sample : Fin N → Z) => DSWG24DiscretizationBias.Theorem2iii.sampledDecision rule sample
```

Retained governing declarations

- AppliedModelingLib.pmfProb
- DSWG24DiscretizationBias.Pareto.ParetoOptimal
- DSWG24DiscretizationBias.ProofBridge.isFirstArgmax
- DSWG24DiscretizationBias.ProofBridge.sourcePNq
- DSWG24DiscretizationBias.Theorem2iii.expectedDatasetAccuracy
- DSWG24DiscretizationBias.Theorem2iii.expectedDatasetFidelityAt
- DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF
- DSWG24DiscretizationBias.Theorem2iii.sampledDecision
- DSWG24DiscretizationBias.Theorem2iii.sampledPosterior

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Fintype Z
4. parameter: DecidableEq Z
5. parameter: ℕ
6. parameter: ℕ
7. parameter: PMF Z
8. parameter: Z → X
9. parameter: X → Fin K → ℝ
10. parameter: Z → Fin K
11. parameter: X → Fin K
12. parameter: (Fin N → X) → Fin K
13. parameter: (Fin N → X) → Fin K → ℝ
14. assumption: 2 ≤ K
15. assumption: K < N
16. assumption: ∀ (x : X), DSWG24DiscretizationBias.ProofBridge.isFirstArgmax (posterior x) (argmaxRule x)
17. assumption: 0 < AppliedModelingLib.pmfProb ν fun (z : Z) => rule z ≠ argmaxRule (feature z)
18. assumption: DSWG24DiscretizationBias.ProofBridge.sourcePNq
  (fun (sample : Fin N → X) => DSWG24DiscretizationBias.Theorem2iii.sampledPosterior posterior sample) aggregateArgmax
  prefAt
19. assumption: 0 < N
20. conclusion: ¬DSWG24DiscretizationBias.Pareto.ParetoOptimal
    (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetAccuracy
      (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
      DSWG24DiscretizationBias.Theorem2iii.sampledPosterior (fun (z : Z) (y : Fin K) => posterior (feature z) y) sample)
    (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetFidelityAt
      (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
      prefAt fun (i : Fin N) => feature (sample i))
    fun (sample : Fin N → Z) => DSWG24DiscretizationBias.Theorem2iii.sampledDecision rule sample
```

Lean proof endpoint (built separately)

DSWG24DiscretizationBias.PaperInterface.theorem2iii_randomized_non_argmax_not_pareto

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

Reason: The finite augmented state Z represents a feature draw together with any realized randomization; positive disagreement is therefore the source probability-one agreement condition in contrapositive form. The posterior-score expected-accuracy expression is the Bayes-reduced source metric, and the target retains K>=2, N>K, the nontrivial reference family, and Pareto nonoptimality.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 3

Source locator: cited publication, lines 756-763

Verbatim source input

Theorem 2. Consider Bayes optimal q, and N > K. For all F :
(i). For every γ, pref there exists a joint decision-making rule that maximizes Oγ (D, q, pref) in Equation (3).

(ii). The argmax decision rule Dargmax maximizes O1 (D, q, pref), i.e., is accuracy maximizing.
(iii). Dargmax is the only Pareto optimal independent decision rule for non-trivial pref ∈ P. No independent decision rule D maximizes objective Oγ for any γ, unless γ = 1 and D agrees with Dargmax with probability 1.

Expanded PaperInterface specification

```
∀ {Z : Type u_1} {X : Type u_2} [inst : Fintype Z] [inst_1 : DecidableEq Z] {N K : ℕ} (ν : PMF Z) (feature : Z → X)
  (posterior : X → Fin K → ℝ) (rule : Z → Fin K) (argmaxRule : X → Fin K) (aggregateArgmax : (Fin N → X) → Fin K)
  (prefAt : (Fin N → X) → Fin K → ℝ) (γ : ℝ),
  2 ≤ K →
    K < N →
      (∀ (x : X), DSWG24DiscretizationBias.ProofBridge.isFirstArgmax (posterior x) (argmaxRule x)) →
        (0 < AppliedModelingLib.pmfProb ν fun (z : Z) => rule z ≠ argmaxRule (feature z)) →
          DSWG24DiscretizationBias.ProofBridge.sourcePNq
              (fun (sample : Fin N → X) => DSWG24DiscretizationBias.Theorem2iii.sampledPosterior posterior sample)
              aggregateArgmax prefAt →
            0 ≤ γ →
              γ < 1 →
                0 < N →
                  ¬∀ (other : (Fin N → Z) → Fin N → Fin K),
                      DSWG24DiscretizationBias.Pareto.weightedObjective γ
                          (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetAccuracy
                            (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
                            DSWG24DiscretizationBias.Theorem2iii.sampledPosterior
                              (fun (z : Z) (y : Fin K) => posterior (feature z) y) sample)
                          (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetFidelityAt
                            (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
                            prefAt fun (i : Fin N) => feature (sample i))
                          other ≤
                        DSWG24DiscretizationBias.Pareto.weightedObjective γ
                          (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetAccuracy
                            (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
                            DSWG24DiscretizationBias.Theorem2iii.sampledPosterior
                              (fun (z : Z) (y : Fin K) => posterior (feature z) y) sample)
                          (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetFidelityAt
                            (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
                            prefAt fun (i : Fin N) => feature (sample i))
                          fun (sample : Fin N → Z) => DSWG24DiscretizationBias.Theorem2iii.sampledDecision rule sample
```

Retained governing declarations

- AppliedModelingLib.pmfProb
- DSWG24DiscretizationBias.Pareto.weightedObjective
- DSWG24DiscretizationBias.ProofBridge.isFirstArgmax
- DSWG24DiscretizationBias.ProofBridge.sourcePNq
- DSWG24DiscretizationBias.Theorem2iii.expectedDatasetAccuracy
- DSWG24DiscretizationBias.Theorem2iii.expectedDatasetFidelityAt
- DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF
- DSWG24DiscretizationBias.Theorem2iii.sampledDecision
- DSWG24DiscretizationBias.Theorem2iii.sampledPosterior

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Fintype Z
4. parameter: DecidableEq Z
5. parameter: ℕ
6. parameter: ℕ
7. parameter: PMF Z
8. parameter: Z → X
9. parameter: X → Fin K → ℝ
10. parameter: Z → Fin K
11. parameter: X → Fin K
12. parameter: (Fin N → X) → Fin K
13. parameter: (Fin N → X) → Fin K → ℝ
14. parameter: ℝ

```
15. assumption: 2 ≤ K
16. assumption: K < N
17. assumption: ∀ (x : X), DSWG24DiscretizationBias.ProofBridge.isFirstArgmax (posterior x) (argmaxRule x)
18. assumption: 0 < AppliedModelingLib.pmfProb ν fun (z : Z) => rule z ≠ argmaxRule (feature z)
19. assumption: DSWG24DiscretizationBias.ProofBridge.sourcePNq
  (fun (sample : Fin N → X) => DSWG24DiscretizationBias.Theorem2iii.sampledPosterior posterior sample) aggregateArgmax
  prefAt
20. assumption: 0 ≤ γ
21. assumption: γ < 1
22. assumption: 0 < N
23. conclusion: ¬∀ (other : (Fin N → Z) → Fin N → Fin K),
    DSWG24DiscretizationBias.Pareto.weightedObjective γ
        (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetAccuracy
          (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
          DSWG24DiscretizationBias.Theorem2iii.sampledPosterior (fun (z : Z) (y : Fin K) => posterior (feature z) y)
            sample)
        (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetFidelityAt
          (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
          prefAt fun (i : Fin N) => feature (sample i))
        other ≤
      DSWG24DiscretizationBias.Pareto.weightedObjective γ
        (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetAccuracy
          (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
          DSWG24DiscretizationBias.Theorem2iii.sampledPosterior (fun (z : Z) (y : Fin K) => posterior (feature z) y)
            sample)
        (DSWG24DiscretizationBias.Theorem2iii.expectedDatasetFidelityAt
          (DSWG24DiscretizationBias.Theorem2iii.iidSamplePMF ν N) fun (sample : Fin N → Z) =>
          prefAt fun (i : Fin N) => feature (sample i))
        fun (sample : Fin N → Z) => DSWG24DiscretizationBias.Theorem2iii.sampledDecision rule sample
```

Lean proof endpoint (built separately)

DSWG24DiscretizationBias.PaperInterface.theorem2iii_randomized_weighted_objective_maximizer_agrees_argmax

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

Reason: This is the paired Bayes-reduced finite randomized endpoint. It retains the source regime, positive disagreement, and nontrivial reference condition, and says that the rule cannot maximize the weighted objective for 0<=gamma<1. That interval is the accepted clarification of the source's unless gamma=1 statement, not an economic caveat.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 4

Source locator: cited publication, lines 744-755

Verbatim source input

3.3 Results: Individual discretization versus joint optimization
Above, we study how the continuous classifier's properties induce discretization bias, when using the argmax rule. We now study discrete decision-making: given an imperfect but Bayes optimal classifier, how can we make unbiased decisions? We show that joint decision-making, as in the program in Equation (1), is necessary for Pareto optimality.

q
For the result, we define the notion of non-trivial reference distributions PN : a reference distribution
1
pq : {xi } → △NY is in P N if it assigns mass at least N to the most likely class in the dataset, i.e., at least 1 data point would be discretized to the most likely class arg maxy i q(y, xi).13
P

[Next verbatim source excerpt]

q
For the result, we define the notion of non-trivial reference distributions PN : a reference distribution
1
pq : {xi } → △NY is in P N if it assigns mass at least N to the most likely class in the dataset, i.e., at least 1 data point would be discretized to the most likely class arg maxy i q(y, xi).13
P

Expanded PaperInterface specification

```
∀ {Sample : Type u_1} {N K : ℕ} (posterior : Sample → Fin N → Fin K → ℝ) (aggregateArgmax : Sample → Fin K)
  (prefAt : Sample → Fin K → ℝ),
  DSWG24DiscretizationBias.ProofBridge.sourcePNq posterior aggregateArgmax prefAt ↔
    DSWG24DiscretizationBias.ProofBridge.referenceSimplexAt prefAt ∧
      (∀ (sample : Sample),
          DSWG24DiscretizationBias.ProofBridge.isAggregateFirstArgmax (posterior sample) (aggregateArgmax sample)) ∧
        ∀ (sample : Sample), 1 / ↑N ≤ prefAt sample (aggregateArgmax sample)
```

Retained governing declarations

- DSWG24DiscretizationBias.ProofBridge.isAggregateFirstArgmax
- DSWG24DiscretizationBias.ProofBridge.referenceSimplexAt
- DSWG24DiscretizationBias.ProofBridge.sourcePNq

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: ℕ
3. parameter: ℕ
4. parameter: Sample → Fin N → Fin K → ℝ
5. parameter: Sample → Fin K
6. parameter: Sample → Fin K → ℝ
7. conclusion: DSWG24DiscretizationBias.ProofBridge.sourcePNq posterior aggregateArgmax prefAt ↔
  DSWG24DiscretizationBias.ProofBridge.referenceSimplexAt prefAt ∧
    (∀ (sample : Sample),
        DSWG24DiscretizationBias.ProofBridge.isAggregateFirstArgmax (posterior sample) (aggregateArgmax sample)) ∧
      ∀ (sample : Sample), 1 / ↑N ≤ prefAt sample (aggregateArgmax sample)
```

Lean proof endpoint (built separately)

DSWG24DiscretizationBias.PaperInterface.nontrivial_reference_family_definition

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

Reason: The source family is a dataset-indexed reference distribution in the label simplex that assigns at least 1/N mass to the dataset aggregate-posterior maximizer. The target states precisely the simplex, fixed-order aggregate argmax, and 1/N mass conditions for every dataset.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 5

Source locator: cited publication, lines 642-650

Verbatim source input

```
Theorem 1. Argmax bias depends on predictive uncertainty, i.e., how much information features x provide
about the true class label. Consider calibrated classifier q and the argmax decision rule Dargmax . Let N > K
and consider a reference distribution of either the aggregate posterior or the prior. Then,
  (i). Suppose x provides no information and q(y, x) = Pr(y) for all x, y. Then, amplification bias is
       maximized and all data points are classified as a plurality class:
                                                    (
                                                      1 − Pr(y), if y = arg maxw p(w),
                            BIAS(y, Dargmax ) =
                                                      − Pr(y),      otherwise.
```

Expanded PaperInterface specification

```
∀ {X : Type u_1} {Y : Type u_2} [inst : Fintype X] [inst_1 : DecidableEq X] [Nonempty X] [inst_3 : Fintype Y]
  [inst_4 : DecidableEq Y] (μ : PMF (X × Y)) (q : X → Y → ℝ) (z y : Y),
  (∀ (x : X) (a : Y), q x a = DSWG24DiscretizationBias.ProofBridge.prior μ a) →
    (∀ (a : Y), DSWG24DiscretizationBias.ProofBridge.prior μ a ≤ DSWG24DiscretizationBias.ProofBridge.prior μ z) →
      (DSWG24DiscretizationBias.Finite.paperBias μ (fun (x : X) => z) (DSWG24DiscretizationBias.ProofBridge.prior μ) y =
          if z = y then 1 - DSWG24DiscretizationBias.ProofBridge.prior μ y
          else -DSWG24DiscretizationBias.ProofBridge.prior μ y) ∧
        DSWG24DiscretizationBias.Finite.paperBias μ (fun (x : X) => z)
            (DSWG24DiscretizationBias.Finite.paperAggregatePosterior μ q) y =
          if z = y then 1 - DSWG24DiscretizationBias.ProofBridge.prior μ y
          else -DSWG24DiscretizationBias.ProofBridge.prior μ y
```

Retained governing declarations

- DSWG24DiscretizationBias.Finite.paperAggregatePosterior
- DSWG24DiscretizationBias.Finite.paperBias
- DSWG24DiscretizationBias.ProofBridge.prior

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: Fintype X
4. parameter: DecidableEq X
5. assumption: Nonempty X
6. parameter: Fintype Y
7. parameter: DecidableEq Y
8. parameter: PMF (X × Y)
9. parameter: X → Y → ℝ
10. parameter: Y
11. parameter: Y
12. assumption: ∀ (x : X) (a : Y), q x a = DSWG24DiscretizationBias.ProofBridge.prior μ a
13. assumption: ∀ (a : Y), DSWG24DiscretizationBias.ProofBridge.prior μ a ≤ DSWG24DiscretizationBias.ProofBridge.prior μ z
14. conclusion: (DSWG24DiscretizationBias.Finite.paperBias μ (fun (x : X) => z) (DSWG24DiscretizationBias.ProofBridge.prior μ) y =
    if z = y then 1 - DSWG24DiscretizationBias.ProofBridge.prior μ y
    else -DSWG24DiscretizationBias.ProofBridge.prior μ y) ∧
  DSWG24DiscretizationBias.Finite.paperBias μ (fun (x : X) => z)
      (DSWG24DiscretizationBias.Finite.paperAggregatePosterior μ q) y =
    if z = y then 1 - DSWG24DiscretizationBias.ProofBridge.prior μ y
    else -DSWG24DiscretizationBias.ProofBridge.prior μ y
```

Lean proof endpoint (built separately)

```
DSWG24DiscretizationBias.PaperInterface.theorem1i_no_information_bias
```

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

Reason: The no-information and plurality premises expose the source’s q(y,x)=Pr(y) and plurality choice. The target gives the displayed bias formula for both source-permitted reference choices. The source’s word maximized has no separate formal objective in the pinned statement, so the displayed formula is the complete mathematical conclusion here.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 6

Source locator: cited publication, lines 667-673

Verbatim source input

(iii). More generally, in the intermediate regime, argmax bias is upper bounded by the predictive error of the classifier q:

$$\mathrm{BIAS}(y, D_{\mathrm{argmax}}) \le \mathrm{MAE}(q). \quad (4)$$

The bound is tight: there exist FXY and q such that Equation (4) holds with equality for the plurality class.

Settled source reading Calibration uses the standard conditional-expectation event identity intended by the source proof.

Expanded PaperInterface specification

```
∀ {X : Type u_1} {Y : Type u_2} [inst : MeasurableSpace X] [inst_1 : MeasurableSpace Y] [MeasurableSingletonClass Y]
  [inst_3 : Fintype Y] [inst_4 : DecidableEq Y] (μ : MeasureTheory.Measure (X × Y)) [MeasureTheory.IsFiniteMeasure μ]
  (q : X → Y → ℝ) (argmaxRule : X → Y) (y : Y),
  Measurable argmaxRule →
    DSWG24DiscretizationBias.ProofBridge.isArgmaxRule q argmaxRule →
      DSWG24DiscretizationBias.ProofBridge.posteriorSimplex q →
        (∀ (y : Y), Measurable fun (xy : X × Y) => q xy.1 y) →
          DSWG24DiscretizationBias.ProofBridge.calibrated μ q →
            DSWG24DiscretizationBias.ProofBridge.continuousJointPriorBias μ argmaxRule y ≤
                DSWG24DiscretizationBias.ProofBridge.continuousJointClassifierMAE μ q ∧
              DSWG24DiscretizationBias.ProofBridge.continuousJointAggregateBias μ q argmaxRule y ≤
                DSWG24DiscretizationBias.ProofBridge.continuousJointClassifierMAE μ q
```

Retained governing declarations

- DSWG24DiscretizationBias.ProofBridge.calibrated
- DSWG24DiscretizationBias.ProofBridge.continuousJointAggregateBias
- DSWG24DiscretizationBias.ProofBridge.continuousJointClassifierMAE
- DSWG24DiscretizationBias.ProofBridge.continuousJointPriorBias
- DSWG24DiscretizationBias.ProofBridge.isArgmaxRule
- DSWG24DiscretizationBias.ProofBridge.posteriorSimplex

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: MeasurableSpace X
4. parameter: MeasurableSpace Y
5. assumption: MeasurableSingletonClass Y
6. parameter: Fintype Y
7. parameter: DecidableEq Y
8. parameter: MeasureTheory.Measure (X × Y)
9. assumption: MeasureTheory.IsFiniteMeasure μ
10. parameter: X → Y → ℝ
11. parameter: X → Y
12. parameter: Y
13. assumption: Measurable argmaxRule
14. assumption: DSWG24DiscretizationBias.ProofBridge.isArgmaxRule q argmaxRule
15. assumption: DSWG24DiscretizationBias.ProofBridge.posteriorSimplex q
16. assumption: ∀ (y : Y), Measurable fun (xy : X × Y) => q xy.1 y
17. assumption: DSWG24DiscretizationBias.ProofBridge.calibrated μ q
18. conclusion: DSWG24DiscretizationBias.ProofBridge.continuousJointPriorBias μ argmaxRule y ≤
    DSWG24DiscretizationBias.ProofBridge.continuousJointClassifierMAE μ q ∧
  DSWG24DiscretizationBias.ProofBridge.continuousJointAggregateBias μ q argmaxRule y ≤
    DSWG24DiscretizationBias.ProofBridge.continuousJointClassifierMAE μ q
```

Lean proof endpoint (built separately)

DSWG24DiscretizationBias.PaperInterface.theorem1iii_argmax_bias_le_mae

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

Reason: The selected source clause bounds each label's argmax bias by classifier MAE for either the prior or aggregate-posterior reference. The expanded target proves both corresponding inequalities: predicted argmax mass minus true-label mass and predicted argmax mass minus aggregate posterior are each at most the integral of the simplex score error sum q_y(1-q_y). Its weak pointwise-argmax premise includes the source argmax rule, posteriorSimplex supplies probability-vector scores, and the calibrated event identity is exactly the approved usual calibration reading. Finiteness and measurability are analytic well-formedness restrictions, while allowing any finite measure and any maximizing tie choice strengthens the bound on the source-admissible probability-distribution case.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 7

Source locator: cited publication, lines 667-673

Verbatim source input

(iii). More generally, in the intermediate regime, argmax bias is upper bounded by the predictive error of the classifier q:

$$\mathrm{BIAS}(y, D_{\mathrm{argmax}}) \leq \mathrm{MAE}(q). \tag{4}$$

The bound is tight: there exist FXY and q such that Equation (4) holds with equality for the plurality class.

Expanded PaperInterface specification

DSWG24DiscretizationBias.Finite.paperAggregateBias DSWG24DiscretizationBias.Finite.paperTheorem1TightMu DSWG24DiscretizationBias.Finite.paperTheorem1TightQ DSWG24DiscretizationBias.Finite.paperTheorem1TightArgmax 0 = DSWG24DiscretizationBias.ProofBridge.classifierMAE DSWG24DiscretizationBias.Finite.paperTheorem1TightMu DSWG24DiscretizationBias.Finite.paperTheorem1TightQ

Retained governing declarations

- DSWG24DiscretizationBias.Finite.paperAggregateBias
- DSWG24DiscretizationBias.Finite.paperTheorem1TightArgmax
- DSWG24DiscretizationBias.Finite.paperTheorem1TightMu
- DSWG24DiscretizationBias.Finite.paperTheorem1TightQ
- DSWG24DiscretizationBias.ProofBridge.classifierMAE

Lean-elaborated claim roles

1. conclusion: DSWG24DiscretizationBias.Finite.paperAggregateBias DSWG24DiscretizationBias.Finite.paperTheorem1TightMu DSWG24DiscretizationBias.Finite.paperTheorem1TightQ DSWG24DiscretizationBias.Finite.paperTheorem1TightArgmax 0 = DSWG24DiscretizationBias.ProofBridge.classifierMAE DSWG24DiscretizationBias.Finite.paperTheorem1TightMu DSWG24DiscretizationBias.Finite.paperTheorem1TightQ

Lean proof endpoint (built separately)

DSWG24DiscretizationBias.PaperInterface.theorem1iii_tight_binary_example

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

Reason: The repaired named witness has one feature, two uniformly likely labels, posterior q(0)=q(1)=1/2, and the fixed first-class tie break. It is calibrated at the sole score value, its aggregate-posterior bias for class 0 is 1/2, and its classifier MAE is 1/2, exactly witnessing the source tightness assertion.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 8

Source locator: cited publication, lines 756-758

Verbatim source input

```
Theorem 2. Consider Bayes optimal q, and N > K. For all F :
   (i). For every γ, pref there exists a joint decision-making rule that maximizes Oγ (D, q, pref ) in Equa-
        tion (3).

[Next verbatim source excerpt]

   Now, note that the objective can be decomposed using the tower property, as follows:
 γ
ON (D, q, pref ) = γACCN (D, q) + (1 − γ)FIDN (D, q, pref )
                        [U+0002 control character]                                                                                   [U+0003 control character]
                 = EF{X} EF{Y }|{X} [γacc(yi , D({q(xi )})) + (1 − γ)fid(pref , D({q(xi )})) | {Xi } = {xi }]
                        "            "     N
                                                                                                                         ##
                                         1 X
                 = EF{X} EF{Y }|{X} γ          I [D({q(xi )}) = yi ] + (1 − γ)fid(pref , D({q(xi )})) | {Xi } = {xi }
                                        N i
                        "            "     N
                                                                                                                        ##
                                         1 X
                 = EF{X} EF{Y }|{X} γ          q(D({q(xj )})i , xi ) + (1 − γ)fid(pref , D({q(xi )})) | {Xi } = {xi }
                                        N i

   Where the last line follows from, given Bayes optimal q, we have for all ŷi

                                                q(ŷi , xi ) = EFY |X [Y = ŷi |X = xi ].

Thus, the optimization decision rule maximizes the inner expectation in the final line, for each given dataset
                                                   γ
{xi }. Thus, the corresponding rule maximizes ON  (D, q, pref ), as desired.
```

Expanded PaperInterface specification

```
∀ {ω : Type u_1} {σ : Type u_2} {N K : ℕ} [inst : NeZero K],
  2 ≤ K →
    K < N →
      ∀ (expect : (ω → ℝ) → ℝ),
        AppliedModelingLib.Decision.FiniteLinearExpectation expect →
          ∀ (observedDataset : ω → σ) (trueLabels : ω → Fin N → Fin K) (γ : ℝ) (posterior : σ → Fin N → Fin K → ℝ)
            (fidelityTerm : σ → (Fin N → Fin K) → ℝ),
            (∀ (i : Fin N) (choose : σ → Fin K),
                (expect fun (x : ω) => if choose (observedDataset x) = trueLabels x i then 1 else 0) =
                  expect fun (x : ω) => posterior (observedDataset x) i (choose (observedDataset x))) →
              ∃ (optRule : σ → Fin N → Fin K),
                ∀ (rule : σ → Fin N → Fin K),
                  DSWG24DiscretizationBias.ProofBridge.expectedObjective expect observedDataset trueLabels γ
                      fidelityTerm rule ≤
                    DSWG24DiscretizationBias.ProofBridge.expectedObjective expect observedDataset trueLabels γ
                      fidelityTerm optRule
```

Retained governing declarations

- AppliedModelingLib.Decision.FiniteLinearExpectation
- DSWG24DiscretizationBias.ProofBridge.expectedObjective

Lean-elaborated claim roles

```
1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: ℕ
4. parameter: ℕ
5. assumption: NeZero K
6. assumption: 2 ≤ K
7. assumption: K < N
8. parameter: (ω → ℝ) → ℝ
9. assumption: AppliedModelingLib.Decision.FiniteLinearExpectation expect
10. parameter: ω → σ
11. parameter: ω → Fin N → Fin K
12. parameter: ℝ
13. parameter: σ → Fin N → Fin K → ℝ
14. parameter: σ → (Fin N → Fin K) → ℝ
15. assumption: ∀ (i : Fin N) (choose : σ → Fin K),
  (expect fun (x : ω) => if choose (observedDataset x) = trueLabels x i then 1 else 0) =
    expect fun (x : ω) => posterior (observedDataset x) i (choose (observedDataset x))
16. conclusion: ∃ (optRule : σ → Fin N → Fin K),
  ∀ (rule : σ → Fin N → Fin K),
    DSWG24DiscretizationBias.ProofBridge.expectedObjective expect observedDataset trueLabels γ fidelityTerm rule ≤
      DSWG24DiscretizationBias.ProofBridge.expectedObjective expect observedDataset trueLabels γ fidelityTerm optRule
```

Lean proof endpoint (built separately)

DSWG24DiscretizationBias.PaperInterface.theorem2i_joint_rule_exists

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

Reason: The governing source model at lines 590-595 defines |Y| = K >= 2, and Theorem 2 at lines 756-758 assumes N > K; these are exactly the target premises hK : 2 <= K and hNK : K < N. The Bayes-row identity and finite-linear expectation encode the tower-property conversion used in the displayed proof. The target quantifies over a generic gamma and fidelityTerm, so the paper's Equation (3) objective for every stated gamma and pref is a direct specialization, and the existential rule dominates every competing joint rule as claimed.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 9

Source locator: cited publication, lines 760

Verbatim source input

(ii). The argmax decision rule Dargmax maximizes O1 (D, q, pref), i.e., is accuracy maximizing.

[Next verbatim source excerpt]

Proof for Part (ii)
The result follows directly from the classifier being Bayes optimal – for each row, you cannot do better than picking the most likely class for that row.
Recall that D({q(xi)}) refers to the set of decisions ŷ1:N with dataset x1:N , where the decisions can be jointly made across data points. Note that, for the Bayes optimal classifier q ∗ , we have

EFY |X [I [Y = y] | X = x] = q ∗ (y, x)

Thus, we have:
1
ON (D, q ∗ , ·) = ACCN (D, q ∗) (26)
∗
= EF [acc(y1:N , D({q (xi)}))] (27)
" #
1 X
= EF I [ŷi = y] let ŷi ≜ D({q ∗ (xj)}N
j=1)i
N i
(28)
" " ##
1 X
= EF{X} EF{Y }|{X} I [Yi = ŷi] | {Xi } = {xi } tower property
N i
(29)
" #
1 X
= EF{X} EFY |X [I [Y = ŷi] | X = xi] independent sampling of datapoints
N i
(30)
" #
1 X ∗
= EF{X} q (ŷi , xi) (31)
N i
" #
1 X ∗ ∗
≤ EF{X} q (arg max q (y, xi), xi) (32)
N i y

1
≜ ON (Dargmax , q ∗ , ·). (33)

Expanded PaperInterface specification

∀ {ω : Type u_1} {σ : Type u_2} {N K : ℕ} [NeZero K],
2 ≤ K →
K < N →
∀ (expect : (ω → ℝ) → ℝ),
AppliedModelingLib.Decision.FiniteLinearExpectation expect →
∀ (observedDataset : ω → σ) (trueLabels : ω → Fin N → Fin K) (posterior : σ → Fin N → Fin K → ℝ)
{decisionRule argmaxRule : σ → Fin N → Fin K},
(∀ (xs : σ), AppliedModelingLib.Decision.IsPointwiseMax (posterior xs) (argmaxRule xs)) →
(∀ (i : Fin N) (choose : σ → Fin K),
(expect fun (x : ω) => if choose (observedDataset x) = trueLabels x i then 1 else 0) =
expect fun (x : ω) => posterior (observedDataset x) i (choose (observedDataset x))) →
AppliedModelingLib.Decision.expectedDecisionAccuracy expect observedDataset trueLabels decisionRule ≤
AppliedModelingLib.Decision.expectedDecisionAccuracy expect observedDataset trueLabels argmaxRule

Retained governing declarations

- AppliedModelingLib.Decision.FiniteLinearExpectation
- AppliedModelingLib.Decision.IsPointwiseMax
- AppliedModelingLib.Decision.expectedDecisionAccuracy

Lean-elaborated claim roles

1. parameter: Type u_1
2. parameter: Type u_2
3. parameter: ℕ
4. parameter: ℕ
5. assumption: NeZero K
6. assumption: 2 ≤ K
7. assumption: K < N
8. parameter: (ω → ℝ) → ℝ
9. assumption: AppliedModelingLib.Decision.FiniteLinearExpectation expect
10. parameter: ω → σ
11. parameter: ω → Fin N → Fin K

12. parameter: σ → Fin N → Fin K → ℝ
13. parameter: σ → Fin N → Fin K
14. parameter: σ → Fin N → Fin K
15. assumption: ∀ (xs : σ), AppliedModelingLib.Decision.IsPointwiseMax (posterior xs) (argmaxRule xs)
16. assumption: ∀ (i : Fin N) (choose : σ → Fin K),
  (expect fun (x : ω) => if choose (observedDataset x) = trueLabels x i then 1 else 0) =
    expect fun (x : ω) => posterior (observedDataset x) i (choose (observedDataset x))
17. conclusion: Real.le✝ (AppliedModelingLib.Decision.expectedDecisionAccuracy expect observedDataset trueLabels decisionRule)
  (AppliedModelingLib.Decision.expectedDecisionAccuracy expect observedDataset trueLabels argmaxRule)

Lean proof endpoint (built separately)

DSWG24DiscretizationBias.PaperInterface.theorem2ii_argmax_accuracy_maximizing

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

Reason: The governing source model at lines 590-595 defines |Y| = K >= 2, and Theorem 2 at line 756 assumes N > K; these are exactly hK : 2 <= K and hNK : K < N. The pointwise-max premise and Bayes-row identity give the source proof's rowwise posterior comparison, and the conclusion compares the argmax rule against an arbitrary decision rule in expected accuracy, exactly the O_1 claim. Source-permitted rule randomness is included by instantiating the generic expectation and observation carrier with the rule's random seed; the target does not exclude that source case.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation
