Enhancing Conformal Prediction via Class Similarity

Ariel Fargion Lahav Dabah Tom Tirer

Faculty of Engineering
Bar Ilan University

June 1, 2026

Introduction

- Suppose we train an image classifier to distinguish between animal species. Given an image, the model predicts “wolf”.

Introduction

- Suppose we train an image classifier to distinguish between animal species. Given an image, the model predicts “wolf”.
- But if the image is blurry or ambiguous, the correct label could also plausibly be “dog” or “fox”.

Introduction

- Suppose we train an image classifier to distinguish between animal species. Given an image, the model predicts “wolf”.
- But if the image is blurry or ambiguous, the correct label could also plausibly be “dog” or “fox”.
- Instead of outputting a single prediction, we would like the model to return a **set of candidate labels** likely to contain the true answer.

Introduction

- Suppose we train an image classifier to distinguish between animal species. Given an image, the model predicts "wolf".
- But if the image is blurry or ambiguous, the correct label could also plausibly be "dog" or "fox".
- Instead of outputting a single prediction, we would like the model to return a **set of candidate labels** likely to contain the true answer.
- **Conformal Prediction (CP)** provides exactly this capability: under exchangeability assumptions, CP constructs prediction sets with a formal **coverage guarantee**.

Introduction

- Suppose we train an image classifier to distinguish between animal species. Given an image, the model predicts "wolf".
- But if the image is blurry or ambiguous, the correct label could also plausibly be "dog" or "fox".
- Instead of outputting a single prediction, we would like the model to return a **set of candidate labels** likely to contain the true answer.
- **Conformal Prediction (CP)** provides exactly this capability: under exchangeability assumptions, CP constructs prediction sets with a formal **coverage guarantee**.
- For a user-specified confidence level (e.g., 90%), the prediction set contains the correct label with at least that probability.

Introduction

- Suppose we train an image classifier to distinguish between animal species. Given an image, the model predicts "wolf".
- But if the image is blurry or ambiguous, the correct label could also plausibly be "dog" or "fox".
- Instead of outputting a single prediction, we would like the model to return a **set of candidate labels** likely to contain the true answer.
- **Conformal Prediction (CP)** provides exactly this capability: under exchangeability assumptions, CP constructs prediction sets with a formal **coverage guarantee**.
- For a user-specified confidence level (e.g., 90%), the prediction set contains the correct label with at least that probability.
- This makes CP especially valuable in **high-stakes applications**, such as medical diagnosis and autonomous driving.

Introduction

- Suppose we train an image classifier to distinguish between animal species. Given an image, the model predicts "wolf".
- But if the image is blurry or ambiguous, the correct label could also plausibly be "dog" or "fox".
- Instead of outputting a single prediction, we would like the model to return a **set of candidate labels** likely to contain the true answer.
- **Conformal Prediction (CP)** provides exactly this capability: under exchangeability assumptions, CP constructs prediction sets with a formal **coverage guarantee**.
- For a user-specified confidence level (e.g., 90%), the prediction set contains the correct label with at least that probability.
- This makes CP especially valuable in **high-stakes applications**, such as medical diagnosis and autonomous driving.
- Beyond coverage, a key challenge is improving the **efficiency** of CP methods, commonly measured by the average prediction set size.

Introduction

- Suppose we train an image classifier to distinguish between animal species. Given an image, the model predicts "wolf".
- But if the image is blurry or ambiguous, the correct label could also plausibly be "dog" or "fox".
- Instead of outputting a single prediction, we would like the model to return a **set of candidate labels** likely to contain the true answer.
- **Conformal Prediction (CP)** provides exactly this capability: under exchangeability assumptions, CP constructs prediction sets with a formal **coverage guarantee**.
- For a user-specified confidence level (e.g., 90%), the prediction set contains the correct label with at least that probability.
- This makes CP especially valuable in **high-stakes applications**, such as medical diagnosis and autonomous driving.
- Beyond coverage, a key challenge is improving the **efficiency** of CP methods, commonly measured by the average prediction set size.
- Our work improves efficiency and semantic coherence by leveraging **class similarity information**.

Background: Conformal Prediction

- **Prediction Sets & Coverage:** Given a user-specified miscoverage level $\alpha \in (0, 1)$ (e.g., 10%), CP outputs a set of candidate labels mathematically guaranteed to contain the true class with at least $1 - \alpha$ probability. [Vovk et al., 1999]

Background: Conformal Prediction

- **Prediction Sets & Coverage:** Given a user-specified miscoverage level $\alpha \in (0, 1)$ (e.g., 10%), CP outputs a set of candidate labels mathematically guaranteed to contain the true class with at least $1 - \alpha$ probability. [Vovk et al., 1999]
- **The Mechanism:** Using a held-out calibration set, CP computes a quantile threshold $\hat{q}$ based on a heuristic non-conformity score. The only assumption required is that the calibration and test data are *exchangeable*.

Background: Conformal Prediction

- **Prediction Sets & Coverage:** Given a user-specified miscoverage level $\alpha \in (0, 1)$ (e.g., 10%), CP outputs a set of candidate labels mathematically guaranteed to contain the true class with at least $1 - \alpha$ probability. [Vovk et al., 1999]
- **The Mechanism:** Using a held-out calibration set, CP computes a quantile threshold $\hat{q}$ based on a heuristic non-conformity score. The only assumption required is that the calibration and test data are *exchangeable*.

Figure: Prediction sets progressively grow for increasingly difficult or ambiguous examples to maintain the strict coverage guarantee.

Conformal Prediction

Let us state the general process of conformal prediction given the *CP set* $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ and its deployment for a new test sample $\boldsymbol{x}_{test}$:

Conformal Prediction

Let us state the general process of conformal prediction given the *CP set* $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ and its deployment for a new test sample $\boldsymbol{x}_{test}$:

1. Define a heuristic score function $s(\boldsymbol{x}, y) \in \mathbb{R}$ based on some output of the model. A higher score should encode a lower level of agreement between $\boldsymbol{x}$ and y.

Conformal Prediction

Let us state the general process of conformal prediction given the *CP set* $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and its deployment for a new test sample $\boldsymbol{x}_{test}$:

1. Define a heuristic score function $s(\boldsymbol{x}, y) \in \mathbb{R}$ based on some output of the model. A higher score should encode a lower level of agreement between $\boldsymbol{x}$ and y.
2. Compute $\hat{q}$ as the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ quantile of the scores $\{s(\boldsymbol{x}_1, y_1), \ldots, s(\boldsymbol{x}_n, y_n)\}$.

Conformal Prediction

Let us state the general process of conformal prediction given the *CP set* $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and its deployment for a new test sample $\boldsymbol{x}_{test}$:

1. Define a heuristic score function $s(\boldsymbol{x}, y) \in \mathbb{R}$ based on some output of the model. A higher score should encode a lower level of agreement between $\boldsymbol{x}$ and y.
2. Compute $\hat{q}$ as the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ quantile of the scores $\{s(\boldsymbol{x}_1, y_1), \ldots, s(\boldsymbol{x}_n, y_n)\}$.
3. At deployment, use $\hat{q}$ to create prediction sets for new samples: $\mathcal{C}_\alpha(\boldsymbol{x}_{test}) = \{y : s(\boldsymbol{x}_{test}, y) \leq \hat{q}\}$.

Conformal Prediction

Let us state the general process of conformal prediction given the *CP set* $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and its deployment for a new test sample $\boldsymbol{x}_{test}$:

1. Define a heuristic score function $s(\boldsymbol{x}, y) \in \mathbb{R}$ based on some output of the model. A higher score should encode a lower level of agreement between $\boldsymbol{x}$ and y.
2. Compute $\hat{q}$ as the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ quantile of the scores $\{s(\boldsymbol{x}_1, y_1), \ldots, s(\boldsymbol{x}_n, y_n)\}$.
3. At deployment, use $\hat{q}$ to create prediction sets for new samples: $\mathcal{C}_\alpha(\boldsymbol{x}_{test}) = \{y : s(\boldsymbol{x}_{test}, y) \leq \hat{q}\}$.

Theorem (Marginal Coverage [Vovk et al., 1999; Angelopoulos & Bates, 2021])

Suppose that $\{(X_i, Y_i)\}_{i=1}^n$ *and* (X_{test}, Y_{test}) *are i.i.d., and define* $\hat{q}$ *as in step 2 above and* $\mathcal{C}_\alpha(X_{test})$ *as in step 3 above. Then the following holds:*

$$\mathbb{P}\left(Y_{test} \in \mathcal{C}_\alpha(X_{test})\right) \geq 1 - \alpha.$$

Motivation

- In many practical scenarios, classes can be grouped into semantic categories.

Motivation

- In many practical scenarios, classes can be grouped into semantic categories.
- Users can benefit from prediction sets that are not only small on average but also contain semantically similar classes.

Motivation

- In many practical scenarios, classes can be grouped into semantic categories.
- Users can benefit from prediction sets that are not only small on average but also contain semantically similar classes.
- Example: In species classification, confusing two species within the same family (e.g., two types of sparrows) is often less consequential for a typical user than confusing species from different families (e.g., a sparrow versus a hawk)

Motivation

- In many practical scenarios, classes can be grouped into semantic categories.
- Users can benefit from prediction sets that are not only small on average but also contain semantically similar classes.
- Example: In species classification, confusing two species within the same family (e.g., two types of sparrows) is often less consequential for a typical user than confusing species from different families (e.g., a sparrow versus a hawk)
- Existing works:
  - Clustered and group-conditional coverage CP [Vovk, 2012; Ding et al., 2023]: Apply CP per group to improve group conditional coverage — yields larger sets than the baselines.
  - Hierarchical selective classification/CP [Goren et al., 2023; Mortier et al., 2025]: Climb in the hierarchical tree of the classes — yields significantly larger sets than the baselines.

Motivation

- In many practical scenarios, classes can be grouped into semantic categories.
- Users can benefit from prediction sets that are not only small on average but also contain semantically similar classes.
- Example: In species classification, confusing two species within the same family (e.g., two types of sparrows) is often less consequential for a typical user than confusing species from different families (e.g., a sparrow versus a hawk)
- Existing works:
  - Clustered and group-conditional coverage CP [Vovk, 2012; Ding et al., 2023]: Apply CP per group to improve group conditional coverage — yields larger sets than the baselines.
  - Hierarchical selective classification/CP [Goren et al., 2023; Mortier et al., 2025]: Climb in the hierarchical tree of the classes — yields significantly larger sets than the baselines.
- Our research question: **"Can we reduce prediction set sizes and #groups simultaneously, without compromising coverage?"**

Method

- Let $g : [C] \to [G]$ denote a **given** map of classes to groups.
  $g(y) \in [G]$ is the index of the group that contains class $y \in [C]$.

Method

- Let $g : [C] \to [G]$ denote a **given** map of classes to groups.
  $g(y) \in [G]$ is the index of the group that contains class $y \in [C]$.
- Given a sample x, all common CP methods preserve the ranking of the softmax vector $\hat{\pi}(x)$ and, in particular, include the estimated class $\hat{y}(x)$ in the prediction set before including any other class.

Method

- Let $g : [C] \to [G]$ denote a **given** map of classes to groups.
  $g(y) \in [G]$ is the index of the group that contains class $y \in [C]$.
- Given a sample x, all common CP methods preserve the ranking of the softmax vector $\hat{\pi}(x)$ and, in particular, include the estimated class $\hat{y}(x)$ in the prediction set before including any other class.
- Very simple method:
  $$s_\lambda(x, y) := s(x, y) + \lambda\, d(y, \hat{y}(x))$$
  with hyperparameter $\lambda > 0$ and $d(y, \hat{y}(x)) = \mathbb{I}(g(y) \neq g(\hat{y}(x)))$.

Method

- Let $g : [C] \to [G]$ denote a **given** map of classes to groups.
  $g(y) \in [G]$ is the index of the group that contains class $y \in [C]$.
- Given a sample x, all common CP methods preserve the ranking of the softmax vector $\hat{\pi}(x)$ and, in particular, include the estimated class $\hat{y}(x)$ in the prediction set before including any other class.
- Very simple method:
  $$s_\lambda(x, y) := s(x, y) + \lambda\, d(y, \hat{y}(x))$$
  with hyperparameter $\lambda > 0$ and $d(y, \hat{y}(x)) = \mathbb{I}(g(y) \neq g(\hat{y}(x)))$.
- The score of a candidate y that is "semantically far" from $\hat{y}(x)$ is penalized by λ.

Method – Illustration

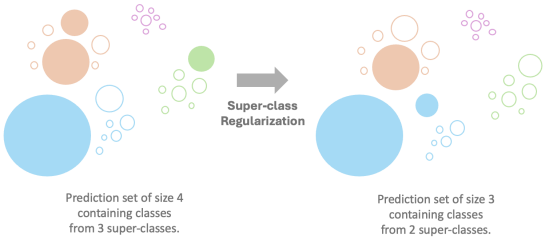

Figure: Illustration of prediction sets for an example before and after applying our proposed regularization. Each circle corresponds to a class, with colors indicating superclasses. Filled circles denote classes included in the prediction set, and circle size reflects the softmax value. In this example, the prediction set size decreases from 4 to 3, and the number of superclasses represented decreases from 3 to 2. We show that our regularization reduces the average prediction set size, regardless of the baseline score function.

Properties of Our Method

- **The coverage property is maintained.**
  s_λ is a valid score that preserves the exchangeability of the calibration and test samples.

Properties of Our Method

- **The coverage property is maintained.**
  s_λ is a valid score that preserves the exchangeability of the calibration and test samples.
- **The number of out-of-group labels in the prediction set cannot increase.**

Properties of Our Method

- **The coverage property is maintained.**
  s_λ is a valid score that preserves the exchangeability of the calibration and test samples.
- **The number of out-of-group labels in the prediction set cannot increase.**

Lemma

We have $\hat{q} \leq \hat{q}_\lambda \leq \hat{q} + \lambda$.

Proposition

Let $\mathcal{Y}_1(x) := \{y : d(y, \hat{y}(x)) \neq 0\}$. $\forall x$ and $\lambda > 0$ we have $\mathcal{C}_\lambda(x) \cap \mathcal{Y}_1(x) \subseteq \mathcal{C}(x) \cap \mathcal{Y}_1(x)$.

Properties of Our Method

Proposition

Let $\mathcal{G}_\lambda(x)$ and $\mathcal{G}(x)$ denote the groups represented in $\mathcal{C}_\lambda(x)$ and $\mathcal{C}(x)$, respectively. For any x such that $\hat{y}(x) \in \mathcal{C}(x)$ and $\lambda > 0$ we have $\mathcal{G}_\lambda(x) \subseteq \mathcal{G}(x)$.

Properties of Our Method

Proposition

Let $\mathcal{G}_\lambda(x)$ and $\mathcal{G}(x)$ denote the groups represented in $\mathcal{C}_\lambda(x)$ and $\mathcal{C}(x)$, respectively. For any x such that $\hat{y}(x) \in \mathcal{C}(x)$ and $\lambda > 0$ we have $\mathcal{G}_\lambda(x) \subseteq \mathcal{G}(x)$.

- The corollary implies $\mathbb{E}[|\mathcal{G}_\lambda(X)|] \leq \mathbb{E}[|\mathcal{G}(X)|]$ (empirically, #Groups strictly decreases with a substantial margin).

Properties of Our Method

Proposition

Let $\mathcal{G}_\lambda(x)$ and $\mathcal{G}(x)$ denote the groups represented in $\mathcal{C}_\lambda(x)$ and $\mathcal{C}(x)$, respectively. For any x such that $\hat{y}(x) \in \mathcal{C}(x)$ and $\lambda > 0$ we have $\mathcal{G}_\lambda(x) \subseteq \mathcal{G}(x)$.

- The corollary implies $\mathbb{E}[|\mathcal{G}_\lambda(X)|] \leq \mathbb{E}[|\mathcal{G}(X)|]$ (empirically, #Groups strictly decreases with a substantial margin).
- **Surprising behavior: The average prediction set size also decreases in practice.** With well-tuned λ (small enough), we observe $\hat{\mathbb{E}}[|\mathcal{C}_\lambda(X)|] < \hat{\mathbb{E}}[|\mathcal{C}(X)|]$, in benchmark settings, even though the penalty does not imply it directly.

Properties of Our Method – Theory for Set Size Reduction

- **Assumption** (large CP set): For small $\lambda \geq 0$, let the CP threshold be the statistical quantile q_λ of the CDF of $s_\lambda(X, Y)$, i.e., $\mathbb{P}(s_\lambda(X, Y) \leq q_\lambda) = 1 - \alpha$.

Properties of Our Method – Theory for Set Size Reduction

- **Assumption** (large CP set): For small $\lambda \geq 0$, let the CP threshold be the statistical quantile q_λ of the CDF of $s_\lambda(X, Y)$, i.e., $\mathbb{P}(s_\lambda(X, Y) \leq q_\lambda) = 1 - \alpha$.
- **Definitions**: for $z \in \{0, 1\}$, $\mathcal{Y}_z(x) := \{y : d(y, \hat{y}(x)) = z\}$.
  - $\overline{n}_0 := \mathbb{E}[|\mathcal{Y}_0(X)|]$ (Average number of "in-group" classes)
  - $\overline{n}_1 := \mathbb{E}[|\mathcal{Y}_1(X)|]$ (Average number of "out-of-group" classes)
  - $p_0 = \mathbb{P}(Y \in \mathcal{Y}_0(X))$ (Probability of "in-group" true label)
  - $p_1 = \mathbb{P}(Y \in \mathcal{Y}_1(X)) = 1 - p_0$ (Probability of "out-of-group" true label)

Properties of Our Method – Theory for Set Size Reduction

- **Assumption** (large CP set): For small $\lambda \geq 0$, let the CP threshold be the statistical quantile q_λ of the CDF of $s_\lambda(X, Y)$, i.e., $\mathbb{P}(s_\lambda(X, Y) \leq q_\lambda) = 1 - \alpha$.
- **Definitions**: for $z \in \{0, 1\}$, $\mathcal{Y}_z(x) := \{y : d(y, \hat{y}(x)) = z\}$.
  - $\overline{n}_0 := \mathbb{E}[|\mathcal{Y}_0(X)|]$ (Average number of "in-group" classes)
  - $\overline{n}_1 := \mathbb{E}[|\mathcal{Y}_1(X)|]$ (Average number of "out-of-group" classes)
  - $p_0 = \mathbb{P}(Y \in \mathcal{Y}_0(X))$ (Probability of "in-group" true label)
  - $p_1 = \mathbb{P}(Y \in \mathcal{Y}_1(X)) = 1 - p_0$ (Probability of "out-of-group" true label)
- In practical settings $\overline{n}_0 \ll \overline{n}_1$, and modern classifiers are quite powerful so p_0 is not small.

Properties of Our Method – Theory for Set Size Reduction

- **Assumption** (large CP set): For small $\lambda \geq 0$, let the CP threshold be the statistical quantile q_λ of the CDF of $s_\lambda(X, Y)$, i.e., $\mathbb{P}(s_\lambda(X, Y) \leq q_\lambda) = 1 - \alpha$.
- **Definitions**: for $z \in \{0, 1\}$, $\mathcal{Y}_z(x) := \{y : d(y, \hat{y}(x)) = z\}$.
  - $\overline{n}_0 := \mathbb{E}[|\mathcal{Y}_0(X)|]$ (Average number of "in-group" classes)
  - $\overline{n}_1 := \mathbb{E}[|\mathcal{Y}_1(X)|]$ (Average number of "out-of-group" classes)
  - $p_0 = \mathbb{P}(Y \in \mathcal{Y}_0(X))$ (Probability of "in-group" true label)
  - $p_1 = \mathbb{P}(Y \in \mathcal{Y}_1(X)) = 1 - p_0$ (Probability of "out-of-group" true label)
- In practical settings $\overline{n}_0 \ll \overline{n}_1$, and modern classifiers are quite powerful so p_0 is not small.

Theorem (informal)

Under mild differentiability assumptions, for **any** score function we have

$$\operatorname{sign}\left(\frac{\mathrm{d}}{\mathrm{d}\lambda}\mathbb{E}[|\mathcal{C}_\lambda(X)|]\big|_{\lambda=0}\right) = \operatorname{sign}\left(a p_1 \overline{n}_0 - b p_0 \overline{n}_1\right),$$

where $a, b \in \mathbb{R}_+$ have formulas based on "empirical"/statistical densities of $s(X, Y)$ given $Y \in \mathcal{Y}_z(x)$ (can be $a \approx b$).

Model-Specific Variant

- The theory shows: to reduce $\mathbb{E}[|\mathcal{C}_\lambda(X)|]$ via a group-wise penalty, partition the classes into groups that are as small as possible (low $\overline{n}_0$ and high $\overline{n}_1$), as long as the probability of making out-of-group mistakes ($p_1 = 1 - p_0$) is kept low.

Model-Specific Variant

- The theory shows: to reduce $\mathbb{E}[|\mathcal{C}_\lambda(X)|]$ via a group-wise penalty, partition the classes into groups that are as small as possible (low $\overline{n}_0$ and high $\overline{n}_1$), as long as the probability of making out-of-group mistakes ($p_1 = 1 - p_0$) is kept low.
- No need for human-related semantic similarity between classes within a group.

Model-Specific Variant

- The theory shows: to reduce $\mathbb{E}[|\mathcal{C}_\lambda(X)|]$ via a group-wise penalty, partition the classes into groups that are as small as possible (low $\overline{n}_0$ and high $\overline{n}_1$), as long as the probability of making out-of-group mistakes ($p_1 = 1 - p_0$) is kept low.
- No need for human-related semantic similarity between classes within a group.
- **Model-specific partition**:
  - Base the penalty on the class similarity *perceived by the classifier*.
  - Applicable for arbitrary datasets.

Model-Specific Variant

- The theory shows: to reduce $\mathbb{E}[|\mathcal{C}_\lambda(X)|]$ via a group-wise penalty, partition the classes into groups that are as small as possible (low $\overline{n}_0$ and high $\overline{n}_1$), as long as the probability of making out-of-group mistakes ($p_1 = 1 - p_0$) is kept low.
- No need for human-related semantic similarity between classes within a group.
- **Model-specific partition**:
  - Base the penalty on the class similarity *perceived by the classifier*.
  - Applicable for arbitrary datasets.
- For a given classifier $f(x) = Wh_\theta(x) + b$, the proposed extension is based on similarity between class means (computed on the training set) at the deepest feature space $h_\theta(\cdot) : \mathcal{X} \to \mathbb{R}^p$ (with $p \geq C$).

Model-Specific Variant

- Denote by $\{x_{c,i}\}$, $i \in [n_c]$, the training samples associated with class $c \in [C]$. Compute the class means and the global mean of the features:

$$\overline{h}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} h_\theta(x_{c,i}), \quad c \in [C]. \qquad \overline{h}_G = \frac{1}{C} \sum_{c=1}^{C} \overline{h}_c.$$

Model-Specific Variant

- Denote by $\{x_{c,i}\}$, $i \in [n_c]$, the training samples associated with class $c \in [C]$. Compute the class means and the global mean of the features:

$$\overline{h}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} h_\theta(x_{c,i}), \quad c \in [C]. \qquad \overline{h}_G = \frac{1}{C} \sum_{c=1}^{C} \overline{h}_c.$$

- We compute "soft" similarity matrix $\boldsymbol{M} \in \mathbb{R}^{C \times C}$ using cosine similarity:

$$M_{c,c'} = \frac{\langle \overline{h}_c - \overline{h}_G, \overline{h}_{c'} - \overline{h}_G \rangle}{\|\overline{h}_c - \overline{h}_G\| \|\overline{h}_{c'} - \overline{h}_G\|}$$

  Empirically, $M_{c,c'} \leq 0$ for most $c' \neq c$, aligned with the Neural Collapse phenomenon [Papyan et al., 2020; Tirer et al., 2023].

Model-Specific Variant

- Denote by $\{x_{c,i}\}$, $i \in [n_c]$, the training samples associated with class $c \in [C]$. Compute the class means and the global mean of the features:

$$\overline{h}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} h_\theta(x_{c,i}), \quad c \in [C]. \qquad \overline{h}_G = \frac{1}{C} \sum_{c=1}^{C} \overline{h}_c.$$

- We compute "soft" similarity matrix $\boldsymbol{M} \in \mathbb{R}^{C \times C}$ using cosine similarity:

$$M_{c,c'} = \frac{\langle \overline{h}_c - \overline{h}_G, \overline{h}_{c'} - \overline{h}_G \rangle}{\|\overline{h}_c - \overline{h}_G\| \|\overline{h}_{c'} - \overline{h}_G\|}$$

  Empirically, $M_{c,c'} \leq 0$ for most $c' \neq c$, aligned with the Neural Collapse phenomenon [Papyan et al., 2020; Tirer et al., 2023].

- We define the soft model-specific penalty function:

$$d^{MS}(y, y') := 1 - M_{y,y'}$$

Model-Specific Variant

- Denote by $\{x_{c,i}\}$, $i \in [n_c]$, the training samples associated with class $c \in [C]$. Compute the class means and the global mean of the features:

$$\overline{h}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} h_\theta(x_{c,i}), \quad c \in [C]. \qquad \overline{h}_G = \frac{1}{C} \sum_{c=1}^{C} \overline{h}_c.$$

- We compute "soft" similarity matrix $\boldsymbol{M} \in \mathbb{R}^{C \times C}$ using cosine similarity:

$$M_{c,c'} = \frac{\langle \overline{h}_c - \overline{h}_G, \overline{h}_{c'} - \overline{h}_G \rangle}{\|\overline{h}_c - \overline{h}_G\| \|\overline{h}_{c'} - \overline{h}_G\|}$$

  Empirically, $M_{c,c'} \leq 0$ for most $c' \neq c$, aligned with the Neural Collapse phenomenon [Papyan et al., 2020; Tirer et al., 2023].

- We define the soft model-specific penalty function:

$$d^{MS}(y, y') := 1 - M_{y,y'}$$

- The penalized score function: $s_\lambda^{MS}(x, y) := s(x, y) + \lambda\, d^{MS}(y, \hat{y}(x))$

Experiments

Performance evaluation of our Model-Agnostic (MA) and Model-Specific (MS) class-similarity approaches for $\alpha = 0.05$

	#Superclasses ↓			Size ↓				
Method	CIFAR100, RN50	CIFAR100, RN34	L17, RN50	ImageNet, ViT	m-ImageNet, RN50	CIFAR100, RN50	CIFAR100, RN34	L17, RN50
LAC								
Standard	2.27 (±0.276)	2.41 (±0.274)	1.26 (±0.068)	1.88 (±0.216)	4.73 (±0.767)	3.68 (±0.759)	3.82 (±0.707)	1.77 (±0.205)
Clustered	2.28 (±0.248)	2.34 (±0.200)	1.24 (±0.035)	2.73 (±0.978)	4.50 (±0.823)	3.70 (±0.676)	3.62 (±0.485)	1.69 (±0.101)
AIR	**1.36** (±0.097)	**1.43** (±0.091)	**1.16** (±0.040)	N/A	N/A	6.80 (±0.101)	7.15 (±0.089)	5.80 (±0.061)
MA-CS	1.85 (±0.183)	1.92 (±0.108)	1.19 (±0.068)	N/A	N/A	3.17 (±0.424)	3.51 (±0.749)	1.71 (±0.183)
MS-CS	1.83 (±0.137)	1.87 (±0.126)	1.19 (±0.058)	**1.80** (±0.161)	**3.82** (±0.696)	**2.92** (±0.339)	**2.94** (±0.339)	**1.70** (±0.156)
RAPS								
Standard	2.49 (±0.145)	3.40 (±0.112)	1.35 (±0.052)	3.27 (±0.176)	8.67 (±2.398)	3.83 (±0.276)	5.79 (±0.223)	1.98 (±0.167)
Clustered	2.42 (±0.103)	3.32 (±0.115)	1.33 (±0.027)	4.25 (±2.505)	8.19 (±2.224)	3.67 (±0.200)	5.62 (±0.232)	1.90 (±0.090)
AIR	**1.95** (±0.077)	3.03 (±0.085)	**1.20** (±0.026)	N/A	N/A	9.75 (±0.121)	15.15 (±0.095)	6.00 (±0.051)
MA-CS	2.01 (±0.160)	2.29 (±0.250)	1.23 (±0.081)	N/A	N/A	3.50 (±0.305)	4.52 (±0.501)	1.97 (±0.295)
MS-CS	**1.95** (±0.122)	**2.22** (±0.182)	1.24 (±0.050)	**2.55** (±0.284)	**7.35** (±2.495)	**3.17** (±0.265)	**3.79** (±0.387)	**1.81** (±0.222)
SAPS								
Standard	2.33 (±0.242)	2.39 (±0.17)	1.32 (±0.048)	2.13 (±0.107)	5.93 (±1.480)	3.45 (±0.458)	3.57 (±0.342)	1.94 (±0.164)
Clustered	2.55 (±0.293)	2.62 (±0.276)	1.31 (±0.045)	10.0 (±4.351)	8.03 (±2.173)	3.86 (±0.547)	4.02 (±0.521)	2.01 (±0.172)
AIR	**1.51** (±0.163)	**1.59** (±0.124)	**1.11** (±0.032)	N/A	N/A	7.55 (±0.324)	7.95 (±0.274)	5.55 (±0.062)
MA-CS	1.88 (±0.286)	1.26 (±0.033)	1.93 (±0.162)	N/A	N/A	**3.14** (±0.464)	**3.32** (±0.271)	1.94 (±0.147)
MS-CS	1.97 (±0.187)	2.14 (±0.175)	1.24 (±0.035)	**2.08** (±0.110)	**4.70** (±1.026)	**3.14** (±0.361)	**3.32** (±0.371)	**1.87** (±0.146)

Experiments – Key Findings

- **In terms of prediction set size: MS-CS and MA-CS are the best and second-best.** This is consistent in all experiments (improvement even over standard LAC). Clustered (group-wise CP) and AIR (hierarchical CP) are worse than the standard versions.
- **In terms of #Superclasses: MA-CS, and surprisingly also MS-CS, are competitive with AIR (but AIR yields awful prediction sizes).** AIR, MA-CS, MS-CS significantly outperform the standard and clustered versions.
- **Graceful dependence of the metrics on λ.** Stability also for class-conditional coverage.

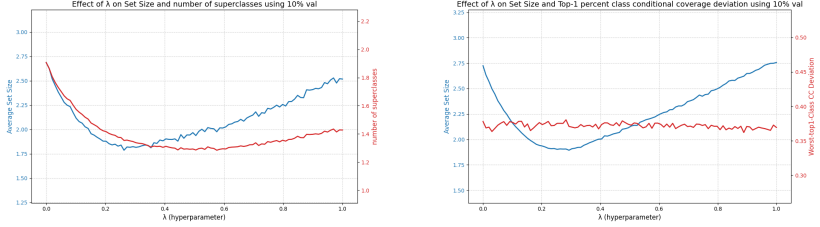

Figure: MA-CS for CIFAR-100, ResNet50, and RAPS; Effect of λ on: AvgSize (blue), #Superclasses (red, left), worst-class-covGap (red, right).

Thank You!

- **Our Paper:** Fargion, A., Dabah, L., & Tirer, T., "Enhancing Conformal Prediction via Class Similarity," Submitted, 2025. (arXiv:2511.19359)

- **Code & Implementation:**
  - Full implementation and benchmarks available.
  - Scan the QR code for the repository.

Scan for Code

arielfar77@gmail.com