REPROCP: REFINED PROTOTYPICAL CONFORMAL PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Conformal prediction has become a popular framework for distribution-free uncertainty quantification, but its effectiveness in low-data regimes is limited by the data splitting required for model fitting and calibration. Full conformal prediction avoids this split by reusing all labeled examples but typically incurs the prohibitive cost of repeatedly refitting the model. We introduce *ReProCP* (Refined Prototypical Conformal Prediction), an efficient full conformal method built on pre-trained self-supervised vision models. Our key insight is that self-supervised embeddings provide not only a strong foundation for accurate transfer learning, but also an effective representation for uncertainty quantification. Operating directly in this embedding space, ReProCP employs a training-free, prototype-based nonconformity score that can be efficiently evaluated across candidate labels, eliminating repeated model fitting. ReProCP further leverages unlabeled images to refine the embedding geometry through dimensionality reduction, whitening, and neighborhood smoothing, yielding more compact prediction sets without additional annotations. Across diverse image-classification datasets, self-supervised backbones, and labeling budgets, ReProCP consistently produces smaller prediction sets than competing conformal baselines, with increasingly pronounced gains as the labeling budget decreases.

1 INTRODUCTION

Reliable image classification requires both accurate predictions and principled uncertainty quantification. Conformal prediction (CP) constructs prediction sets with distribution-free, finite-sample coverage guarantees under exchangeability (Vovk et al., 2005; Angelopoulos & Bates, 2023). Its most common implementation, split CP, partitions labeled data into model-fitting and calibration subsets (Papadopoulos et al., 2002). When only a few labeled examples are available, this split limits both predictive accuracy and calibration efficiency. Full CP avoids this tradeoff by reusing all labeled examples, but typically requires computationally expensive model refitting for each test example and candidate label.

Existing approaches address limited supervision by transferring information from related labeled tasks or by assuming sufficient training data but limited calibration data. Alongside these approaches, recent large-scale evaluation has highlighted the potential of pre-trained vision representations for CP (Fillioux et al., 2026). Our focus is the more restrictive setting in which labels are scarce for both task learning and conformal inference, while an unlabeled image pool is available. This calls for methods that exploit pre-trained representations and unlabeled data while avoiding both label splitting and repeated model fitting.

We introduce *ReProCP* (Refined Prototypical Conformal Prediction), an efficient full conformal method built on pre-trained self-supervised vision embeddings. Our key insight is that these embeddings provide not only a strong foundation for learning with few labels but also a useful representation for uncertainty quantification. ReProCP represents each class by the mean of its labeled embeddings and uses prototype similarity to define a training-free nonconformity score. This prototype-based formulation enables efficient full conformal inference directly in the embedding space, eliminating the need for repeated model fitting.

To obtain more informative prediction sets, ReProCP further uses unlabeled images to refine the embedding geometry. As illustrated in Figure 1, dimensionality reduction, whitening, and neigh-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

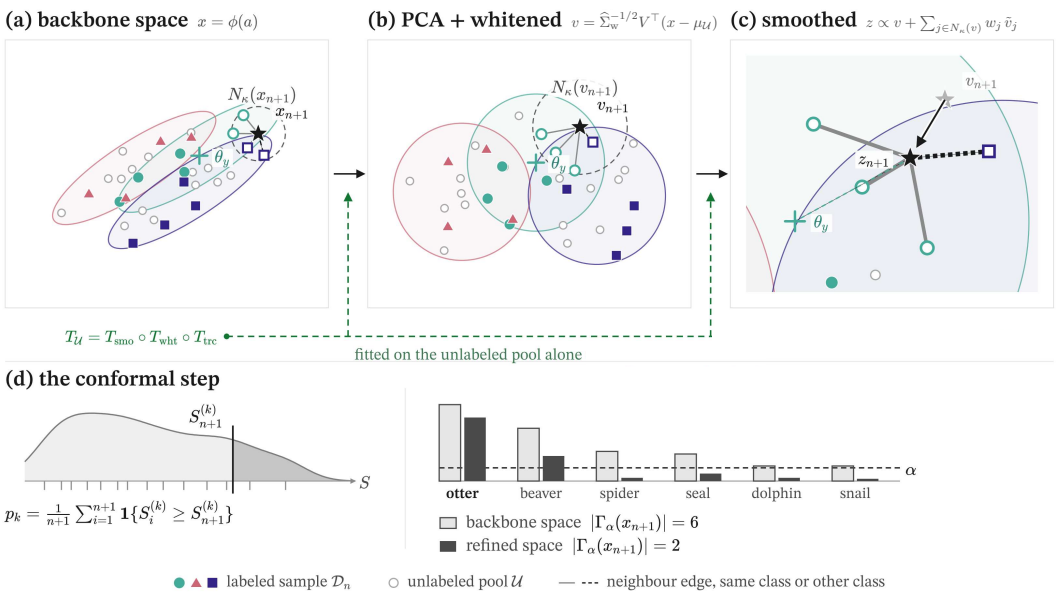


Figure 1: A demonstration of the refinement effect on prediction sets. Panels (a) to (c) are schematic. (a) Backbone classes are elongated along shared directions. (b) Truncation and whitening rescale the space, shrinking the directions shared across classes. (c) Smoothing then moves v_{n+1} toward the class center θ_y, yielding z_{n+1}. Panel (d) shows a measured example for one CIFAR-100 image using frozen DINOv2 embeddings. Six of the 100 candidate labels have p-values above α in the backbone space, compared with two in the refined space.

borhood smoothing suppress noisy directions, balance variation across the retained subspace, and reduce local image-specific variability. This label-free refinement improves the representation used for conformal scoring without additional annotations or task-specific model training. Under the stated sampling assumptions, fixing the refinement using the unlabeled pool and applying it identically to labeled and test examples preserves conformal validity.

Across diverse image-classification datasets, backbones, and labeling budgets, ReProCP produces smaller prediction sets than competing baselines in nearly all scarce-label comparisons, with the largest gains at the lowest budgets. Representation refinement further reduces set size by up to 79% relative to full CP on the original embeddings.

Our contributions are summarized as follows:

- We introduce ReProCP, an efficient full conformal method using a training-free, prototype-based nonconformity score on pre-trained self-supervised embeddings, reusing all labeled examples without repeated model fitting.
- We develop a label-free representation refinement based on dimensionality reduction, whitening, and neighborhood smoothing, with theoretical analysis addressing its validity and benefit.
- We demonstrate that ReProCP yields more compact prediction sets than competing conformal baselines across diverse benchmarks, with particularly pronounced gains in label-scarce settings.

2 Related Work

Frozen self-supervised representations. Vision transformers (Dosovitskiy et al., 2021), combined with large-scale pre-training, provide strong frozen representations for transfer learning. DINOv2 and its successors learn general-purpose visual features that support diverse downstream tasks through lightweight adaptation or probing (Oquab et al., 2024; Siméoni et al., 2025), while CLIP learns transferable representations from image–text supervision (Radford et al., 2021). ReProCP adopts this frozen-backbone paradigm but uses the embedding space directly for conformal inference via a nearest-prototype nonconformity score.

Representation refinement. Post-hoc refinement of pre-trained descriptors has long been effective in image retrieval. Principal component analysis (PCA) and whitening can improve descriptor matching by suppressing correlated and uninformative directions (Jégou & Chum, 2012). This practice remains relevant in modern training-free systems; for example, Min et al. (2026) apply PCA-based whitening to DINOv2 embeddings for zero-shot cross-view retrieval. Related benefits have been observed in few-shot classification, where Simple CNAPS uses a regularized Mahalanobis metric—equivalently, a whitened feature space—to outperform Euclidean and cosine distances (Bateni et al., 2020). These findings motivate refining the geometry of pre-trained representations when conformal scores rely on embedding similarity. ReProCP therefore applies truncation, whitening, and neighborhood smoothing, all fitted exclusively on an unlabeled image pool.

Conformal prediction. CP converts a nonconformity score into prediction sets with finite-sample coverage under exchangeability (Vovk et al., 2005; Angelopoulos & Bates, 2023). Full, or transductive, CP treats each candidate-labeled test point symmetrically with the training data, whereas split conformal separates model fitting from calibration (Papadopoulos et al., 2002; Lei et al., 2018). Although split conformal dominates practical classification (Sadinle et al., 2019; Romano et al., 2020; Angelopoulos et al., 2021), full conformal remains attractive when its computational cost can be controlled, for example through closed-form updates or incremental learning (Nouretdinov et al., 2001; Cherubin et al., 2021). ReProCP advances this direction with a training-free prototype score that enables efficient full conformal inference directly in the embedding space.

Conformal prediction with few labels or unlabeled data. Prior work has reduced the labeled-data requirements of CP through meta-learned scores (Fisch et al., 2021) or unlabeled examples assigned approximate scores via nearest-neighbor matching (Zhou et al., 2026). Full Conformal Adaptation (FCA) reuses few-shot labels for both adapting and conformalizing medical vision–language models (Silva-Rodríguez et al., 2025). In contrast, ReProCP uses vision-only prototype scores and a tailored refinement pipeline derived from unlabeled data to improve set-size efficiency.

3 PRELIMINARIES

3.1 PROBLEM FORMULATION

Let $\mathcal{A}$ denote the image space, $\mathcal{X} = \mathbb{R}^d$ the embedding space, and $\mathcal{Y} = [K] := \{1, \dots, K\}$ the label space. A fixed pre-trained encoder $\phi : \mathcal{A} \to \mathcal{X}$ maps each image $a \in \mathcal{A}$ to its embedding $x = \phi(a)$. We observe a labeled sample $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n \subseteq \mathcal{X} \times \mathcal{Y}$, where $x_i = \phi(a_i)$, and an unlabeled pool $\mathcal{U} = \{\tilde{x}_j\}_{j=1}^m \subseteq \mathcal{X}$, where $\tilde{x}_j = \phi(\tilde{a}_j)$. The unlabeled images are drawn from the same image distribution, independently of the labeled sample and test example.

Given a test image a_{n+1} with embedding $x_{n+1} = \phi(a_{n+1})$, our goal is to construct a prediction set $\Gamma_\alpha(x_{n+1}) \subseteq \mathcal{Y}$ for its unknown label y_{n+1}. Assuming that $(x_1, y_1), \dots, (x_{n+1}, y_{n+1})$ are exchangeable, we seek the finite-sample marginal coverage guarantee

$$\Pr[y_{n+1} \in \Gamma_\alpha(x_{n+1})] \geq 1 - \alpha, \tag{1}$$

where $\alpha \in (0, 1)$ is the target miscoverage level. Our objective is to obtain prediction sets that are as small as possible while maintaining this guarantee.

3.2 BACKGROUND ON CONFORMAL PREDICTION

CP constructs prediction sets satisfying Eq. (1) using nonconformity scores, with larger scores indicating poorer agreement between an input and a candidate label (Vovk et al., 2005; Angelopoulos & Bates, 2023). Let $S(x, y; \mathcal{D})$ denote the score of a labeled example (x, y) relative to a reference sample $\mathcal{D}$.

Split conformal prediction. Split CP partitions the labeled indices into disjoint training and calibration sets $\mathcal{I}_{\text{tr}}$ and $\mathcal{I}_{\text{cal}}$, with sizes n_{tr} and n_{cal}, respectively, where $n_{\text{tr}} + n_{\text{cal}} = n$. A predictor is fitted on $\mathcal{D}_{\text{tr}} = \{(x_i, y_i) : i \in \mathcal{I}_{\text{tr}}\}$, yielding the fixed scoring function $S_{\text{tr}}(x, y) := S(x, y; \mathcal{D}_{\text{tr}})$. The calibration scores are $s_i = S_{\text{tr}}(x_i, y_i)$ for $i \in \mathcal{I}_{\text{cal}}$. Consider their ordered values as $s_{(1)} \leq \cdots \leq s_{(n_{\text{cal}})}$, the conformal threshold is given by

$$\hat{q}_\alpha = s_{(\lceil (n_{\text{cal}}+1)(1-\alpha) \rceil)}. \tag{2}$$

The resulting prediction set is

$$\Gamma_\alpha^{\text{split}}(x_{n+1}) = \{k \in [K] : S_{\text{tr}}(x_{n+1}, k) \le \hat{q}_\alpha\}. \tag{3}$$

Split CP is computationally efficient because the predictor is fitted only once. However, when labels are scarce, partitioning the set limits the data available for both model fitting and calibration.

Full conformal prediction. Full CP avoids this split by incorporating the test example into the sample set used for scoring (Vovk et al., 2005). For each candidate label $k \in [K]$, we form the augmented sample

$$\mathcal{D}_{n+1}^{(k)} = \mathcal{D}_n \cup \{(x_{n+1}, k)\}. \tag{4}$$

Let $y_i^{(k)} = y_i$ for $i \le n$ and $y_{n+1}^{(k)} = k$. In the leave-one-out formulation considered here, each example is scored relative to the other n examples:

$$s_i^{(k)} = S\Big(x_i, y_i^{(k)}; \mathcal{D}_{-i}^{(k)}\Big), \qquad i = 1, \dots, n+1, \tag{5}$$

where $\mathcal{D}_{-i}^{(k)}$ denotes the augmented sample with its i-th example removed. The same scoring rule is used for every example and is invariant to the ordering of the reference sample.

The conformal p-value compares the test score with all $n+1$ scores:

$$p_k = \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbf{1}\Big\{s_i^{(k)} \ge s_{n+1}^{(k)}\Big\}, \tag{6}$$

and the full CP set is

$$\Gamma_\alpha^{\text{full}}(x_{n+1}) = \{k \in [K] : p_k > \alpha\}. \tag{7}$$

For the true candidate label, exchangeability and symmetric scoring ensure that the p-value is super-uniform, yielding the coverage guarantee in Eq. (1).

Full CP reuses all labeled examples, but the scores generally need to be recomputed for each candidate label. When scoring requires fitting a predictor, these computations can make inference expensive. Scores that admit simple updates enable efficient full conformal inference without repeated model fitting.

4 METHOD

Our method combines pool-based representation refinement with an efficient full conformal predictor for label-scarce image classification. In the first stage, we use an unlabeled image pool to adapt the geometry of frozen self-supervised embeddings, without training a task-specific model or accessing additional labels. In the second stage, we apply full CP using a training-free prototype score whose leave-one-out updates can be computed in closed form. This design exploits all available labeled examples while retaining finite-sample coverage and avoiding the repeated model fitting typically required by full CP.

4.1 POOL-FITTED REPRESENTATION REFINEMENT

We refine the frozen embeddings using three complementary operations fitted entirely on the unlabeled pool. First, truncation removes directions that cannot be estimated reliably from the available data. Second, within-group whitening suppresses nuisance variation while preserving directions that distinguish groups. Third, neighborhood smoothing reduces image-specific noise by exploiting the local geometry of the pool. Together, these operations define

$$T_{\mathcal{U}} = T_{\text{smo}} \circ T_{\text{wht}} \circ T_{\text{trc}}, \tag{8}$$

which is applied identically to labeled and test embeddings. Thus, the representation is refined without using class labels or fitting a task-specific model.

Truncation. We first restrict the embeddings to directions that are reliably supported by the unlabeled pool. Let

$$\mu_{\mathcal{U}} = \frac{1}{m} \sum_{j=1}^{m} \tilde{x}_j, \qquad \widehat{\Sigma}_{\mathcal{U}} = \frac{1}{m} \sum_{j=1}^{m} (\tilde{x}_j - \mu_{\mathcal{U}})(\tilde{x}_j - \mu_{\mathcal{U}})^\top$$

denote the pool mean and covariance, and let $V \in \mathbb{R}^{d\times d'}$ contain the leading d' eigenvectors of $\widehat{\Sigma}_{\mathcal{U}}$. The truncated representation is

$$T_{\mathrm{trc}}(x) = V^\top (x - \mu_{\mathcal{U}}). \tag{9}$$

Here, truncation is used primarily for statistical stability rather than dimensionality reduction. Low-variance directions are estimated unreliably from a finite pool, yet whitening rescales directions inversely to their estimated variability and can therefore strongly amplify noise in these components. Similar degradation from full-dimensional whitening has been observed in image retrieval (Jégou & Chum, 2012). Truncating first confines the subsequent whitening step to a subspace that is well supported by the pool.

Within-group whitening. We next rescale the retained directions according to their within-group variability. Ideally, the distance underlying the conformal score should emphasize class-discriminative variation while discounting nuisance variation caused by factors such as pose, illumination, or background. Whitening by the total pool covariance does not distinguish between these two sources of variation. We therefore estimate a within-group covariance, using clusters as label-free proxies for semantic classes.

Specifically, we partition the truncated pool embeddings $\tilde{t}_j = T_{\mathrm{trc}}(\tilde{x}_j)$ into C groups using k-means and compute

$$\widehat{\Sigma}_{\mathrm{w}} = \frac{1}{m}\sum_{g=1}^{C}\sum_{j\in G_g}(\tilde{t}_j - \bar{t}_g)(\tilde{t}_j - \bar{t}_g)^\top, \tag{10}$$

where G_g is the set of examples assigned to group g and $\bar{t}_g$ is their mean. Pooling deviations across groups yields a single, stable covariance estimate based on all m examples. We then define

$$T_{\mathrm{wht}}(t) = \widehat{\Sigma}_{\mathrm{w}}^{-1/2} t. \tag{11}$$

The transformation expresses each direction in units of its estimated within-group spread. Consequently, Euclidean distance after whitening is equivalent to a Mahalanobis distance under $\widehat{\Sigma}_{\mathrm{w}}$, down-weighting directions dominated by nuisance variation. This construction is motivated by the effectiveness of whitened and regularized Mahalanobis metrics in few-shot classification and training-free retrieval with frozen representations (Bateni et al., 2020; Min et al., 2026).

Neighborhood smoothing. Finally, we exploit the local structure of the refined pool to reduce residual image-specific variation. Let $v = T_{\mathrm{wht}}(t)/\|T_{\mathrm{wht}}(t)\|_2$ denote the normalized truncated-and-whitened embedding, and define $\tilde{v}$ analogously for each embedding in the unlabeled pool. We denote by $N_\kappa(v) \subseteq \{1,\dots,m\}$ the set of κ indices of pool embeddings most similar to v in terms of cosine similarity. We define the smoothed representation as

$$T_{\mathrm{smo}}(v) = (1-\beta)\, v + \beta\, \nu(v), \qquad \nu(v) = \sum_{j\in N_\kappa(v)} \frac{w_j}{W}\,\tilde{v}_j, \qquad w_j = \max\{v^\top \tilde{v}_j,\, 0\}^\gamma, \tag{12}$$

where $W = \sum_{j\in N_\kappa(v)} w_j$ and γ controls how strongly the closest neighbors are emphasized. Truncating similarities at zero prevents negatively aligned neighbors from contributing. The smoothed representation is a convex combination of v and the weighted average $\nu(v)$ of its neighbors, and $\beta \in [0,1]$ sets how far it moves toward that average.

This operation is a weighted form of query expansion commonly used in image retrieval (Radenović et al., 2019). It is also related to conformal methods such as DAPS and SNAPS, which improve efficiency by smoothing nonconformity scores over neighboring nodes (Zargarbashi et al., 2023; Song et al., 2024). In contrast, we smooth the representation before computing the score. The resulting embedding can therefore be reused across all candidate labels, avoiding label-dependent neighborhood computations. Performing smoothing after truncation and whitening further ensures that neighborhoods are formed in a geometry that suppresses unreliable or nuisance directions.

4.2 Full conformal prediction with a training-free prototype score

We next construct a full conformal predictor directly in the refined embedding space, without training a classifier. Let $z_i = T_{\mathcal{U}}(x_i)$ for each labeled example $i \in [n]$, and let $z_{n+1} = T_{\mathcal{U}}(x_{n+1})$ denote the refined test embedding. For each candidate label $k \in [K]$, we augment the labeled sample with (z_{n+1}, k). Let $\ell_i^{(k)}$ denote the resulting label of point i, such that $\ell_i^{(k)} = y_i$ for $i \le n$ and $\ell_{n+1}^{(k)} = k$.

To preserve the symmetry required by full CP, each point is scored using prototypes computed without that point. For each point i, let

$$\mathcal{I}_c^{(k,-i)} = \left\{ j \in [n+1] \setminus \{i\} : \ell_j^{(k)} = c \right\}$$

denote the remaining examples assigned to class c. The corresponding leave-one-out prototype is

$$\mu_c^{(k,-i)} = \frac{1}{\left|\mathcal{I}_c^{(k,-i)}\right|} \sum_{j \in \mathcal{I}_c^{(k,-i)}} z_j. \tag{13}$$

We then define the nonconformity score via a temperature-scaled softmax over cosine similarities:

$$S_i^{(k)} = 1 - \frac{\exp\left(\cos\left(z_i, \mu_{\ell_i^{(k)}}^{(k,-i)}\right)/T\right)}{\sum_{c=1}^{K} \exp\left(\cos\left(z_i, \mu_c^{(k,-i)}\right)/T\right)}. \tag{14}$$

This score is small when z_i is more closely aligned with the prototype of its assigned class than with those of competing classes. The temperature $T > 0$ controls the sensitivity of this comparison.

Crucially, evaluating these leave-one-out scores does not require model refitting. For a candidate label k, adding the test embedding changes only the prototype of class k, while leaving out point i changes only the prototype of its assigned class $\ell_i^{(k)}$. All other prototypes remain equal to their labeled-sample means and can be reused. The candidate p-values and final prediction set are obtained by substituting the scores in Eq. (6) and Eq. (7). Under exchangeability of the labeled and test examples, the resulting set satisfies the finite-sample coverage guarantee in Eq. (1) (Vovk et al., 2005).

Computational cost. The closed-form prototype updates make the full conformal sweep computationally practical. Let $z = z_{n+1}$ denote the refined test embedding. The inner products $\langle z_i, z \rangle$ for all n labeled examples are computed once through a matrix–vector product and shared across the K candidate labels. The remaining candidate-dependent quantities follow from rank-one updates of the relevant prototypes, after which each p-value is obtained by a comparison count.

A naive implementation evaluates K class scores for each of the n labeled points and each of the K candidate labels, requiring $O(nK^2)$ scalar operations. Appendix A.1 derives an equivalent organization that reuses all candidate-independent terms and reduces the sweep to $O(nK)$ scalar operations after the shared matrix–vector product. The total cost per test example is therefore $O(n(d' + K))$, while multiple test examples can be handled efficiently using batched matrix operations.

4.3 Theoretical analysis

We establish two complementary properties of the proposed refinement. First, fitting the transformation on an independent unlabeled pool preserves the exchangeability required for conformal validity. Second, we characterize when neighborhood smoothing moves an embedding closer to its class center.

Proposition 1 (Representation refinement preserves validity)**.** *Let $\mathcal{U}$ be independent of the exchangeable pairs $(x_1, y_1), \ldots, (x_{n+1}, y_{n+1})$. Suppose that $T_{\mathcal{U}}$ is a measurable transformation fitted exclusively on $\mathcal{U}$ and applied identically to every input. Then, conditional on $\mathcal{U}$, the refined pairs*

$$\left(T_{\mathcal{U}}(x_i), y_i\right)_{i=1}^{n+1}$$

remain exchangeable. Consequently, applying the full conformal procedure of Section 4.2 yields

$$\Pr[y_{n+1} \in \Gamma_\alpha(x_{n+1}) \,|\, \mathcal{U}] \geq 1 - \alpha, \tag{15}$$

and hence also satisfies the marginal coverage guarantee in Eq. (1).

The result follows by conditioning on $\mathcal{U}$: the transformation then becomes fixed and is applied symmetrically to all labeled and test examples. A formal proof is provided in Appendix A.2.

Algorithm 1 ReProCP: pool-fitted refinement + full CP

Require: pool $\mathcal{U}$, labeled sample $\mathcal{D}_n$, test point x_{n+1}, level α, configuration (d', C, κ, γ), temperature T

Stage 1: fit $T_{\mathcal{U}}$ on the pool (label-free, once)

$\mu_{\mathcal{U}} \leftarrow \mathrm{mean}(\mathcal{U})$, $V \leftarrow$ leading d' eigenvectors of $\mathrm{cov}(\mathcal{U})$ ▷ truncation

$\tilde{t}_j \leftarrow V^\top(\tilde{x}_j - \mu_{\mathcal{U}})$ for all $\tilde{x}_j \in \mathcal{U}$

$\widehat{\Sigma}_{\mathrm{w}} \leftarrow$ pooled within-group covariance of $\{\tilde{t}_j\}$ over C k-means groups, Eq. (10)

$\tilde{v}_j \leftarrow \widehat{\Sigma}_{\mathrm{w}}^{-1/2}\tilde{t}_j \,/\, \big\|\widehat{\Sigma}_{\mathrm{w}}^{-1/2}\tilde{t}_j\big\|$ for all j ▷ whitening

$T_{\mathcal{U}}(x) \leftarrow T_{\mathrm{smo}}\big(\widehat{\Sigma}_{\mathrm{w}}^{-1/2}V^\top(x - \mu_{\mathcal{U}})\big)$ by Eq. (12) ▷ smoothing

Stage 2: full CP on the refined embeddings

$z_i \leftarrow T_{\mathcal{U}}(x_i)$ for $i = 1, \ldots, n$, and $z_{n+1} \leftarrow T_{\mathcal{U}}(x_{n+1})$

$F_{ic} \leftarrow \cos\big(z_i, \mu_c^{(-i)}\big)$ for all $i \in [n]$, $c \in [K]$ ▷ once per labeled sample

$g_i \leftarrow \langle z_{n+1},\, z_i\rangle$ for $i = 1, \ldots, n$ ▷ once per test point

for each candidate label $k \in [K]$ **do**

  rebuild column k of F from g, folding in (z_{n+1}, k)

  $S_i^{(k)} \leftarrow$ Eq. (14) at temperature T, for all $i \le n{+}1$

  $p_k \leftarrow$ Eq. (6)

end for

return $\Gamma_\alpha(x_{n+1}) = \{k \in [K] : p_k > \alpha\}$

We next examine the effect of neighborhood smoothing. Let $\{\theta_c\}_{c\in[K]}$ be reference class centers in the whitened space, and let $c(j)$ denote the unknown class of pool neighbor $\tilde{v}_j$. For a point v from class y, define

$$\varepsilon = \|v - \theta_y\|_2, \qquad r_j = \|\tilde{v}_j - \theta_{c(j)}\|_2.$$

We measure neighborhood homophily, meaning the tendency of nearby points to share the same class, by the weighted fraction of neighborhood mass belonging to class y:

$$h(v) = \frac{1}{W} \sum_{j\in N_\kappa(v)\ c(j)=y} w_j. \tag{16}$$

We also define the largest class-center displacement among foreign-class neighbors as

$$\Delta(v) = \max_{j\in N_\kappa(v)\ c(j)\neq y} \|\theta_{c(j)} - \theta_y\|_2, \tag{17}$$

with $\Delta(v) = 0$ when the neighborhood contains no foreign-class points.

Proposition 2 (Sufficient condition for beneficial smoothing)**.** *For any $\beta \in (0, 1]$, the smoothed representation $z = T_{\mathrm{smo}}(v)$ in Eq. (12) satisfies*

$$\|z - \theta_y\|_2 \le (1-\beta)\,\varepsilon + \beta\left[\sum_{j\in N_\kappa(v)} \frac{w_j}{W}\, r_j + \big(1 - h(v)\big)\,\Delta(v)\right]. \tag{18}$$

Consequently, smoothing strictly decreases the distance to the reference center whenever

$$\sum_{j\in N_\kappa(v)} \frac{w_j}{W}\, r_j + \big(1 - h(v)\big)\,\Delta(v) < \varepsilon. \tag{19}$$

The bound exposes the two properties that determine whether smoothing is beneficial. The first term in Eq. (19) measures how tightly the neighbors concentrate around their respective class centers. The second penalizes contamination by other classes, combining the total weight assigned to foreign neighbors, $1 - h(v)$, with their separation from the target center. Thus, smoothing is favored by compact, homophilous neighborhoods — the same structural principle underlying graph-based conformal score diffusion (Zargarbashi et al., 2023). The proof is given in Appendix A.2.

5 Experiments

5.1 Experimental setup

Datasets and encoders. We evaluate the proposed method on three standard image-classification benchmarks. CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) contain natural images and are relatively close to the pre-training domain, whereas EuroSAT (Helber et al., 2019) consists of Sentinel-2

Table 1: Mean prediction-set size ($\pm$ standard error) with marginal coverage in parentheses, on DINOv2 ViT-B at target miscoverage $\alpha = 0.1$ over 50 trials.

dataset	method	2 shots	4 shots	6 shots	8 shots
CIFAR-10	Split CP (best)	9.81±0.04 (0.988)	2.62±0.25 (0.951)	1.85±0.20 (0.965)	1.11±0.03 (0.944)
	CV+	3.59±0.26 (0.953)	1.21±0.04 (0.940)	1.04±0.01 (0.954)	1.01±0.00 (0.951)
	SemiCP (best)	6.36±0.46 (0.930)	1.65±0.15 (0.907)	1.51±0.19 (0.939)	1.08±0.03 (0.935)
	ReProCP (ours)	**1.02±0.01 (0.978)**	**1.00±0.00 (0.984)**	**1.00±0.00 (0.983)**	**1.00±0.00 (0.976)**
CIFAR-100	Split CP (best)	92.92±0.15 (0.931)	8.76±1.09 (0.945)	2.73±0.14 (0.925)	1.93±0.05 (0.919)
	CV+	8.34±0.50 (0.952)	2.09±0.03 (0.915)	1.70±0.02 (0.917)	1.51±0.02 (0.916)
	SemiCP (best)	91.94±0.23 (0.920)	5.83±0.76 (0.927)	2.32±0.12 (0.912)	1.79±0.04 (0.911)
	ReProCP (ours)	**2.43±0.09 (0.940)**	**1.42±0.02 (0.910)**	**1.31±0.01 (0.904)**	**1.28±0.01 (0.903)**
EuroSAT	Split CP (best)	9.75±0.06 (0.980)	3.97±0.27 (0.944)	3.44±0.26 (0.952)	2.04±0.13 (0.915)
	CV+	4.76±0.23 (0.948)	2.19±0.08 (0.902)	1.77±0.06 (0.913)	1.52±0.03 (0.916)
	SemiCP (best)	7.48±0.38 (0.956)	3.84±0.19 (0.949)	2.35±0.25 (0.893)	1.66±0.10 (0.885)
	ReProCP (ours)	**3.55±0.18 (0.937)**	**2.02±0.09 (0.910)**	**1.69±0.05 (0.910)**	**1.51±0.04 (0.901)**

satellite imagery and represents a larger domain shift. Unless stated otherwise, we extract embeddings using a frozen DINOv2 ViT-B/14 encoder (Oquab et al., 2024). For each dataset, the unlabeled pool $\mathcal{U}$ contains 10,000 held-out images drawn from the same distribution and disjoint from all labeled and test samples.

Method configuration. We use a single configuration across all datasets and encoders. Truncation retains $d' = 128$ dimensions, within-group whitening uses $C = K$ k-means groups, and smoothing uses the $\kappa = 10$ nearest pool neighbors with weight exponent $\gamma = 3$. The smoothing weight is set to $\beta = W/(1+W)$. This counts the input as one more neighbor at full similarity, so the representation moves further toward its neighborhood when more neighbors align closely with it. The temperature T is the only quantity selected separately for each dataset–encoder pair. It is determined solely from the unlabeled pool, before observing any labels, and then fixed across all trials and labeling budgets. Appendix B.1 specifies this selection rule and reports sensitivity analyses for all configuration choices.

Baselines. We compare ReProCP with split CP applied to a trained linear classifier using three standard nonconformity scores: LAC (Sadinle et al., 2019), APS (Romano et al., 2020), and RAPS (Angelopoulos et al., 2021). We additionally compare with SemiCP (Zhou et al., 2026), which supplements the calibration set with scores from nearest-neighbor-matched unlabeled examples, and CV+ (Barber et al., 2021). All split-based methods divide the available labels between classifier training and conformal calibration. For each experimental setting, we report their best result over $n_{\mathrm{tr}}/n \in \{0.25, 0.5, 0.75\}$ and, where applicable, over the three conformal scores. This oracle selection favors the baselines and therefore provides a conservative comparison with ReProCP. All split-based variants use the same linear-classifier architecture, and SemiCP follows its official implementation. ReProCP and SemiCP have access to the unlabeled pool, whereas the remaining baselines use labeled data only. Further implementation details, including the classifier, CV+ folds, and the low-data adaptation required by SemiCP, are provided in Appendix B.2.

Evaluation protocol. We consider $s \in \{2, 4, 6, 8\}$ labeled examples per class. Each result is averaged over 50 independent trials, using class-balanced labeled samples and five test examples per class. We report marginal coverage and mean prediction-set size at $\alpha = 0.1$ in the main text; results for $\alpha = 0.05$ are provided in Appendix C.2. We adopt the non-empty prediction-set convention: whenever a conformal set is empty, it is replaced by the method's highest-ranked label.

5.2 Results

ReProCP outperforms all baselines across all datasets and labeling budgets. As shown in Table 1, ReProCP consistently produces the smallest prediction sets, with the largest gains in the most label-scarce regime. At $s = 2$, it reduces the mean set size by 25–72% relative to the strongest baseline, CV+. The gap narrows as more labels become available, as expected, because splitting-based methods benefit from having sufficient data for both model training and conformal calibration. All methods generally achieve the target coverage, with only slight undercoverage for SemiCP

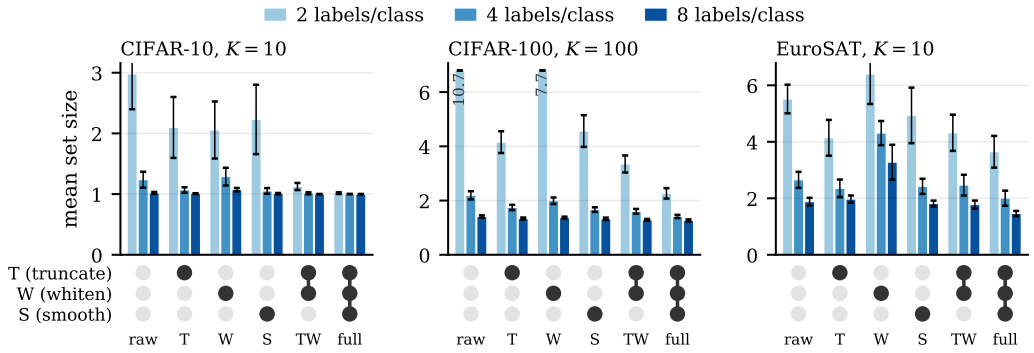


Figure 2: Ablation of the representation-refinement stages. We report mean prediction-set size for the raw embeddings, each individual stage, truncation followed by whitening (T+W), and the complete pipeline (T+W+S), where T, W, and S denote truncation, whitening, and smoothing, respectively. Results are shown for $s \in \{2, 4, 8\}$ labels per class at $\alpha = 0.1$, averaged over 20 trials. Bars exceeding a panel's vertical range are clipped and annotated with their values.

on EuroSAT at $s = 8$, where coverage reaches 0.885. Similar trends hold with CLIP ViT-B (Appendix C.1), where ReProCP outperforms all baselines in most settings and again provides its largest gains when labeled data are most scarce.

Ablation of the representation-refinement pipeline. To assess the contribution of each refinement stage, we compare the raw embeddings with truncation, whitening, and smoothing applied individually, as well as truncation followed by whitening and the complete three-stage pipeline. Figure 2 shows that the complete pipeline consistently yields the smallest mean prediction sets across all datasets and labeling budgets. The gains are largest at $s = 2$, where the raw embeddings produce particularly large sets: the full pipeline reduces the mean set size from approximately 3.0 to 1.0 on CIFAR-10, from more than 10.0 to 2.2 on CIFAR-100, and from 5.5 to 3.7 on EuroSAT. Truncation provides the most consistent improvement among the individual stages, whereas whitening alone is dataset dependent and can degrade performance, most notably on EuroSAT. However, whitening becomes beneficial when applied after truncation, indicating that poorly supported directions should be removed before the retained representation is rescaled. Adding smoothing yields a further improvement in nearly every setting, demonstrating that the three stages are complementary. As the number of labels increases, the differences narrow and prediction sets approach their minimum size, particularly on CIFAR-10, leaving less room for representation refinement to improve efficiency.

Additional results. The appendices provide further evidence for the robustness, effectiveness, and efficiency of ReProCP. Experiments with the stricter miscoverage level $\alpha = 0.05$ show that the same trends persist: ReProCP is best in 11 of 12 settings with DINOv2 and 8 of 12 with CLIP, while generally maintaining the target coverage (Appendix C.2). Further analysis validates the role of smoothing: refinement increases neighborhood homophily and moves at least 99.6% of the evaluated examples closer to their class center (Appendix A.2), while representation-level smoothing outperforms score-level smoothing in 10 of 12 settings (Appendix C.3). Score-distribution visualizations additionally show that refinement improves the separation between true- and false-class scores, reducing the inclusion of incorrect labels (Appendix C.4). Finally, the closed-form score updates return exactly the same conformal p-values as the naive implementation while providing an 80–130$\times$ test-time speedup (Appendices A.1 and C.5).

6 Conclusion

We introduce ReProCP, a training-free conformal framework for the few-label regime. It separates the roles of the available resources: an unlabeled pool is used to refine a frozen self-supervised representation, while the labeled sample is used entirely for conformal inference. A prototype-based score makes full conformal prediction practical in this setting through a closed-form candidate sweep. Across the benchmarks and label budgets studied, ReProCP produces the smallest prediction sets, with the largest gains when labels are scarce and diminishing gains as the labeled sample grows.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance and polishing, and for literature retrieval and discovery. We have not used generative AI tools for generating experimental results or making experimental decisions, and the remaining required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work, manually verified the implementation and experimental results, and performed a manual literature survey to validate AI-assisted literature discovery. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Anastasios N. Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations (ICLR)*, 2021.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021.

Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Giovanni Cherubin, Konstantinos Chatzikokolakis, and Martin Jaggi. Exact optimization of conformal predictors via incremental and decremental learning. In *International Conference on Machine Learning (ICML)*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Leo Fillioux, Julio Silva-Rodríguez, Ismail Ben Ayed, Paul-Henry Cournède, Maria Vakalopoulou, Stergios Christodoulidis, and Jose Dolz. Are foundation models for computer vision good conformal predictors? *Transactions on Machine Learning Research*, 2026.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Few-shot conformal prediction with auxiliary tasks. In *International Conference on Machine Learning (ICML)*, 2021.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Hervé Jégou and Ondřej Chum. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *European Conference on Computer Vision (ECCV)*, 2012.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018.

Jeongho Min, Dongyoung Kim, and Jaehyup Lee. From street to orbit: Training-free cross-view retrieval via location semantics and LLM guidance. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026.

Ilia Nouretdinov, Thomas Melluish, and Volodya Vovk. Ridge regression confidence machine. In *International Conference on Machine Learning (ICML)*, 2001.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning (ECML)*, pp. 345–356, 2002.

Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525):223–234, 2019.

Julio Silva-Rodríguez, Leo Fillioux, Paul-Henry Cournède, Maria Vakalopoulou, Stergios Christodoulidis, Ismail Ben Ayed, and Jose Dolz. Full conformal adaptation of medical vision-language models. In *Information Processing in Medical Imaging (IPMI)*, 2025.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Jianqing Song, Jianguo Huang, Wenyu Jiang, Baoming Zhang, Shuangjie Li, and Chongjun Wang. Similarity-navigated conformal prediction for graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Soroush H. Zargarbashi, Simone Antonelli, and Aleksandar Bojchevski. Conformal prediction sets for graph neural networks. In *International Conference on Machine Learning (ICML)*, 2023.

Xuanning Zhou, Zihao Shi, Hao Zeng, Xiaobo Xia, Bingyi Jing, and Hongxin Wei. Semi-supervised conformal prediction with unlabeled nonconformity score. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

APPENDIX

A TECHNICAL DERIVATIONS AND PROOFS

A.1 CLOSED-FORM UPDATES FOR FULL CONFORMAL SCORES

Section 4.2 states that one matrix-vector product per test point supplies every quantity needed for the sweep over candidate labels. We derive this claim below.

Let z denote the refined test embedding z_{n+1}, let $Q_c = \sum_{j:y_j=c} z_j$ be the sum of the refined labeled embeddings of class c, and let Z be the matrix whose rows are the unit-norm labeled embeddings. Since cosine similarity is invariant to positive rescaling, we work with the class sums Q_c rather than the corresponding means.

Candidate k augments the sample with (z, k), which replaces Q_k by $Q_k + z$ and leaves every other class sum untouched. Logits for classes $c \neq k$ therefore depend only on the labeled sample and can be computed once for the entire candidate sweep. For class k, define

$$b_i \;=\; \langle z_i, Q_k\rangle + \langle z_i, z\rangle, \qquad A \;=\; \|Q_k\|^2 + 2\langle Q_k, z\rangle + \|z\|^2, \tag{20}$$

so that $b_i = \langle z_i, Q_k + z\rangle$ and $A = \|Q_k + z\|^2$. A calibration point outside class k is scored against $Q_k + z$, giving the logit $b_i/\sqrt{A}$. A point inside class k is scored against $Q_k + z - z_i$, which gives

$$\cos\big(z_i,\, \mu_k^{(k,-i)}\big) \;=\; \frac{b_i - 1}{\sqrt{A - 2b_i + 1}}, \tag{21}$$

where the unit norm of z_i has been used twice.

The required quantities fall into three types. The quantities $\langle z_i, Q_c\rangle$ and $||Q_c||^2$ depend only on the labeled sample and can be computed once, in $O(nKd')$. The quantities $\langle Q_c, z\rangle$ and $||z||^2$ depend on the test point but not on i, at a cost of $O(Kd')$ per test point. The only quantities coupling a calibration point to the test point are the inner products $\langle z_i, z\rangle$, all of which are obtained together as the matrix-vector product Zz in $O(nd')$.

The product Zz is independent of the candidate label. For candidate k, only the class-k quantities in Eq. (20) and Eq. (21) change, so the same n inner products serve all K candidates and the remaining sweep consists of scalar arithmetic. Since only the class-k column of the cached logit matrix F changes, each calibration point's softmax denominator can be updated as

$$D_i \;\leftarrow\; D_i - \exp(F_{ik}^{\text{old}}/T) + \exp(F_{ik}^{\text{new}}/T),$$

where F_{ik} is the class-k logit of point i. The denominator update and the subsequent comparison count for the p-value both require $O(n)$ operations per candidate. When computing $S_{n+1}^{(k)}$, the test point is excluded from its own prototype, so its class-k logit is the unaugmented $\langle z, Q_k\rangle$. The candidate-specific test scores therefore follow from the cached quantities in $O(K)$. With $n = sK$ and $s \geq 2$ in every reported run, the per-test-point cost $O(nd'+Kd'+nK)$ reduces to $O(n(d'+K))$.

A.2 Proofs

Proposition 1

Fix the unlabeled pool at $\mathcal{U} = u$. Conditional on this pool, T_u is a fixed measurable map applied according to the same rule to every observation. Since $\mathcal{U}$ is independent of the original exchangeable sequence, that sequence remains exchangeable conditional on $\mathcal{U} = u$, and therefore so does the transformed sequence. The standard full conformal validity argument then gives

$$\Pr[y_{n+1} \in \Gamma_\alpha(x_{n+1}) \,|\, \mathcal{U} = u] \geq 1 - \alpha.$$

Taking the expectation of the conditional coverage probability over $\mathcal{U}$ yields the stated marginal guarantee. □

Proposition 2

The weights w_j/W sum to one, so

$$\nu(v) - \theta_y = \sum_j \frac{w_j}{W}\big(\tilde{v}_j - \theta_{c(j)}\big) + \sum_j \frac{w_j}{W}\big(\theta_{c(j)} - \theta_y\big).$$

By the triangle inequality,

$$\Big\|\sum_j \frac{w_j}{W}\big(\tilde{v}_j - \theta_{c(j)}\big)\Big\| \;\leq\; \sum_j \frac{w_j}{W}\, r_j.$$

The terms with $c(j) = y$ vanish in the second sum, while the remaining difference vectors have norm at most $\Delta(v)$ and carry total weight $1 - h$. Hence

$$\|\nu(v) - \theta_y\| \;\leq\; \sum_j \frac{w_j}{W}\, r_j + (1 - h)\,\Delta(v).$$

Finally, since

$$z - \theta_y = (1 - \beta)(v - \theta_y) + \beta\big(\nu(v) - \theta_y\big),$$

the triangle inequality then gives Eq. (18). If the second term in the resulting bound is strictly smaller than ε, then for $\beta > 0$ the bound is strictly smaller than ε. □

Table 2: Mean homophily $h(v)$ on raw embeddings and on the deployed truncation+whitening space the smoothing stage sees, with the distance to the class center $\varepsilon = \|v - \theta_y\|$ in that space. Improved $= \|z - \theta_y\| < \varepsilon$ after smoothing.

	$h(v)$		improved %	$\bar{\varepsilon}$ (w)
dataset	raw	whit.	whit.	before→after
CIFAR-10	0.971	0.976	100.0	0.766→0.548
CIFAR-100	0.806	0.826	99.6	0.665→0.442
EuroSAT	0.845	0.860	100.0	0.870→0.638

Homophily and the distance to the class center. Table 2 instantiates the quantities in Proposition 2 on the deployed pipeline. The mean homophily $h(v)$ from Eq. (16), averaged over labeled points, increases after truncation and whitening on all three datasets, although the increase is modest, ranging from 0.005 to 0.020, with the largest gains where homophily is initially lowest. The more pronounced effect is on the distance to the class center. After smoothing, the mean distance decreases on every dataset, from 0.665 to 0.442 on CIFAR-100, and at least 99.6% of labeled points end closer to their own center than before smoothing.

B Additional experimental settings

B.1 Configuration

The same configuration is used for every dataset and both encoders: truncation from $d = 768$ to $d' = 128$ dimensions, whitening pooled over $C = K$ k-means groups, and smoothing over $\kappa = 10$ pool neighbors with weight exponent $\gamma = 3$. For CLIP ViT-B/16, the 768-dimensional embedding is the pooled image feature taken before the projection into the joint image-text space.

Temperature. T is the only quantity resolved separately for each dataset and encoder, using the unlabeled pool. The pool is passed through the frozen refinement map, and the $C = K$ k-means groups from the whitening stage serve as class proxies. Let $\tilde{z}_j$ denote pool point j after refinement, $c(j)$ its group, μ_c the mean of group c, and $\mu_{c(j)}^{(-j)}$ the mean of point j's group with j removed. We define the margin of point j and the resulting temperature as

$$g_j = \cos\big(\tilde{z}_j, \mu_{c(j)}^{(-j)}\big) - \max_{c \neq c(j)} \cos\big(\tilde{z}_j, \mu_c\big), \qquad T = \frac{|\mathrm{median}_j\, g_j|}{4}.$$

The factor $1/4$ is fixed across all datasets and encoders. The resulting temperature falls on the flat part of the sensitivity curve in Figure 3 throughout. On CIFAR-100 it is at the lower end of the tested grid, where the prediction sets contain 2.26 labels at two shots, compared with 3.11 at the grid's lowest temperature. On CIFAR-10 and EuroSAT it lies within a plateau spanning more than an order of magnitude, with set size changing only at much smaller or larger temperatures. On EuroSAT, for example, the smallest tested temperature gives 9.9 labels, compared with 3.6 on the plateau.

The remaining hyperparameters. Figure 4 varies the two stage pairs around their deployed values, holding everything else fixed: the smoothing neighborhood κ against the weight exponent γ, and the truncation dimension d' against the whitening group count C. The surface is flat in the region of interest. The smoothing pair changes the set size by at most a few percent on every dataset, with CIFAR-100 spanning 1.40 to 1.49 across the full grid. The truncation and grouping pair remains on a plateau covering d' from 64 to 256 and C from K to $4K$. Two failure modes recur across datasets and occur well outside the deployed region: too few whitening groups ($C = K/2$), which inflates the prediction sets, and severe truncation with many groups ($d' = 32$ and $C = 4K$), which increases the CIFAR-100 set size to 71.8 labels. A few EuroSAT cells improve on the deployed configuration by up to 14%, with no consistent direction across datasets. This is the trade-off accepted by using a single shared configuration.

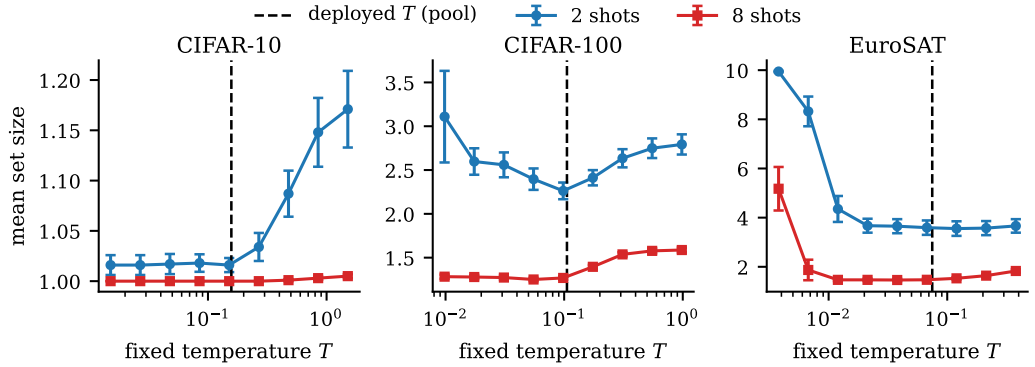


Figure 3: Temperature sensitivity. Mean prediction-set size under fixed temperatures on a log grid (20 trials, $\alpha = 0.1$, DINOv2 ViT-B), at 2 and 8 labels per class. The dashed line marks the deployed pool-resolved T.

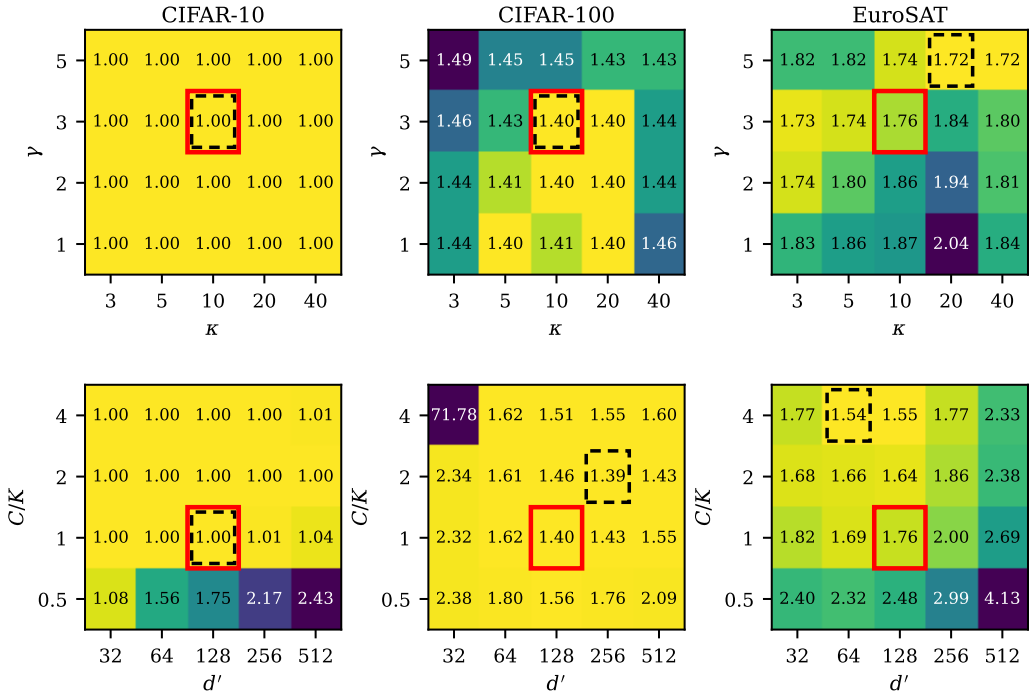


Figure 4: Configuration sensitivity. Mean prediction-set size ($\alpha = 0.1$, four labels per class, 10 trials). Deployed configuration (red box), and optimum (dashed box) marked. Top: smoothing neighborhood κ against weight exponent γ. Bottom: truncation dimension d' against whitening groups per class C/K.

B.2 Baseline implementations

The head. Every baseline uses the same probe: a multinomial logistic regression on L2-normalized raw backbone embeddings, standardized per dimension using its own training split. For n_{tr} training points, we minimize the cross-entropy objective with respect to W:

$$\widehat{W} = \arg\min_{W} \frac{1}{n_{\text{tr}}} \sum_{i} \ell\big(W^{\top} z_i,\, y_i\big) + \lambda \|W\|_F^2, \qquad \lambda = 10^{-2}.$$

The same λ is used for every method, budget, and dataset, so the baselines differ only in their conformal construction. A class absent from a training split receives probability zero, so its calibration

Table 3: CLIP ViT-B results: mean prediction-set size (± SE, 50 trials), with marginal coverage in parentheses, at $\alpha = 0.1$. Split CP and SemiCP use their best score (THR/APS/RAPS) and train fraction per cell.

dataset	method	2 shots	4 shots	6 shots	8 shots
CIFAR-10	Split CP (best)	9.68±0.06 (0.981)	3.08±0.27 (0.940)	2.22±0.17 (0.954)	1.43±0.08 (0.933)
	CV+	3.93±0.28 (0.946)	1.53±0.07 (0.916)	1.25±0.03 (0.926)	1.14±0.02 (0.920)
	SemiCP (best)	6.33±0.53 (0.926)	2.51±0.34 (0.899)	1.44±0.11 (0.894)	1.31±0.09 (0.916)
	ReProCP (ours)	**1.84±0.10 (0.948)**	**1.26±0.04 (0.924)**	**1.19±0.02 (0.931)**	**1.13±0.02 (0.929)**
CIFAR-100	Split CP (best)	92.86±0.17 (0.929)	24.17±1.14 (0.944)	11.70±0.52 (0.923)	8.30±0.26 (0.918)
	CV+	34.07±0.79 (0.957)	10.67±0.22 (0.924)	7.43±0.12 (0.919)	**5.77±0.09 (0.914)**
	SemiCP (best)	92.34±0.23 (0.924)	22.19±1.30 (0.934)	10.22±0.51 (0.909)	7.58±0.27 (0.908)
	ReProCP (ours)	**21.08±0.58 (0.948)**	**8.89±0.21 (0.915)**	**6.73±0.12 (0.905)**	6.02±0.12 (0.903)
EuroSAT	Split CP (best)	9.75±0.06 (0.984)	4.50±0.26 (0.942)	3.98±0.23 (0.943)	2.48±0.15 (0.910)
	CV+	5.85±0.22 (0.949)	**2.91±0.11 (0.918)**	**2.20±0.08 (0.918)**	**1.78±0.05 (0.909)**
	SemiCP (best)	7.77±0.46 (0.936)	4.29±0.44 (0.893)	2.61±0.31 (0.870)	1.95±0.13 (0.872)
	ReProCP (ours)	**4.64±0.20 (0.931)**	3.09±0.12 (0.906)	2.68±0.10 (0.908)	2.36±0.08 (0.901)

points attain the maximal score, pushing the baseline prediction sets toward the full label space at the smallest budgets. The rank-penalized score uses the penalty 0.01 and $k_{\text{reg}} = 2$ from its published configuration.

CV+. CV+ uses the same head with the THR score. The number of folds is $\min(5, s)$, with stratification so that every fold contains every class. Thus, at two shots, each of the two folds trains on a single labeled example per class. Each fold's model scores its held-out points and the test point. Let R_j denote the held-out score of labeled point j, $f(j)$ its fold, and $R^{(k)}_{f(j)}$ the test score for candidate k under that fold's model. The corresponding candidate p-value is

$$p_k = \frac{1 + \#\{\, j \in [n] : R_j \geq R^{(k)}_{f(j)} \,\}}{n+1}.$$

SemiCP. The implementation follows the official method: the non-randomized scores used for its main results, one-nearest-neighbor matching in predicted-score space with the associated bias correction for each unlabeled point, and a quantile computed over the labeled and matched scores jointly. We make one deliberate substitution. The published method assumes a fully trained classifier, which is not available at our label budgets, so we use the probe above in its place. This choice is consistent with their analysis, which reports that matching quality tracks classifier accuracy.

C Additional results

C.1 Backbone transfer

Table 3 repeats the comparison in Table 1 using CLIP ViT-B (Radford et al., 2021) embeddings. The pipeline and configuration from Appendix B.1 are unchanged, with only the pool statistics refitted and the temperature re-resolved for this encoder. ReProCP is smallest in 8 of the 12 cells. The remaining gaps are concentrated on the harder EuroSAT benchmark, highlighting a limitation of the method when the underlying backbone provides a weaker representation.

C.2 Results at the tighter miscoverage level

Tables 4 and 5 repeat both comparisons at the tighter miscoverage level. Prediction sets grow for every method and dataset, while the ordering remains unchanged. ReProCP is smallest in 11 of the 12 DINOv2 cells, with the exception of EuroSAT at eight shots, and in 8 of the 12 CLIP ViT-B cells, with the same four cells favoring CV+ as at the looser level.

C.3 Score-level versus representation-level smoothing

A pool correction can act on the representation or on the score. Table 6 compares the two.

Table 4: DINOv2 ViT-B results: mean prediction-set size (± SE, 50 trials), with marginal coverage in parentheses, at $\alpha = 0.05$. Split CP and SemiCP use their best score (THR/APS/RAPS) and train fraction per cell.

dataset	method	2 shots	4 shots	6 shots	8 shots
CIFAR-10	Split CP (best)	10.00±0.00 (1.000)	6.68±0.37 (0.978)	4.85±0.38 (0.982)	1.72±0.14 (0.969)
	CV+	5.43±0.32 (0.976)	1.73±0.09 (0.963)	1.21±0.03 (0.970)	1.08±0.01 (0.964)
	SemiCP (best)	7.46±0.43 (0.961)	2.63±0.30 (0.945)	2.45±0.40 (0.959)	1.48±0.20 (0.950)
	ReProCP (ours)	**1.13±0.05 (0.982)**	**1.03±0.01 (0.986)**	**1.00±0.00 (0.983)**	**1.00±0.00 (0.977)**
CIFAR-100	Split CP (best)	96.71±0.10 (0.969)	26.90±3.04 (0.977)	6.26±0.46 (0.966)	3.40±0.11 (0.961)
	CV+	24.06±1.40 (0.984)	3.97±0.10 (0.962)	2.91±0.06 (0.960)	2.39±0.04 (0.958)
	SemiCP (best)	95.98±0.16 (0.962)	21.66±3.19 (0.969)	4.67±0.33 (0.954)	3.27±0.15 (0.957)
	ReProCP (ours)	**5.05±0.34 (0.969)**	**2.27±0.06 (0.952)**	**2.06±0.04 (0.950)**	**2.01±0.03 (0.951)**
EuroSAT	Split CP (best)	10.00±0.00 (1.000)	6.76±0.36 (0.974)	6.24±0.30 (0.984)	3.15±0.22 (0.964)
	CV+	6.34±0.25 (0.977)	3.44±0.15 (0.956)	2.60±0.10 (0.958)	**2.25±0.08 (0.956)**
	SemiCP (best)	8.22±0.33 (0.973)	4.59±0.24 (0.961)	3.94±0.43 (0.944)	2.56±0.23 (0.936)
	ReProCP (ours)	**4.86±0.24 (0.971)**	**3.12±0.16 (0.949)**	**2.55±0.12 (0.954)**	2.28±0.10 (0.946)

Table 5: CLIP ViT-B results at $\alpha = 0.05$ under the same evaluation protocol as table 4.

dataset	method	2 shots	4 shots	6 shots	8 shots
CIFAR-10	Split CP (best)	10.00±0.00 (1.000)	6.65±0.42 (0.975)	5.45±0.34 (0.985)	2.17±0.19 (0.964)
	CV+	5.65±0.33 (0.979)	2.51±0.14 (0.973)	1.77±0.07 (0.968)	1.49±0.05 (0.960)
	SemiCP (best)	7.31±0.52 (0.943)	4.22±0.51 (0.932)	2.51±0.34 (0.940)	1.79±0.22 (0.941)
	ReProCP (ours)	**2.76±0.17 (0.970)**	**1.80±0.10 (0.965)**	**1.49±0.06 (0.963)**	**1.42±0.04 (0.959)**
CIFAR-100	Split CP (best)	96.67±0.15 (0.967)	46.45±2.82 (0.975)	21.39±0.90 (0.963)	15.27±0.49 (0.961)
	CV+	58.81±1.30 (0.985)	20.41±0.42 (0.965)	14.07±0.24 (0.963)	**11.02±0.19 (0.960)**
	SemiCP (best)	95.88±0.20 (0.960)	45.21±3.28 (0.964)	18.78±1.11 (0.953)	14.08±0.55 (0.955)
	ReProCP (ours)	**33.71±0.88 (0.975)**	**17.68±0.42 (0.961)**	**13.65±0.28 (0.954)**	12.17±0.26 (0.953)
EuroSAT	Split CP (best)	10.00±0.00 (1.000)	7.69±0.28 (0.975)	6.90±0.28 (0.979)	3.85±0.24 (0.954)
	CV+	7.34±0.22 (0.974)	**4.44±0.17 (0.966)**	**3.51±0.14 (0.963)**	**2.78±0.10 (0.956)**
	SemiCP (best)	8.58±0.32 (0.973)	5.85±0.33 (0.966)	3.69±0.39 (0.912)	3.49±0.34 (0.923)
	ReProCP (ours)	**5.68±0.22 (0.956)**	4.46±0.17 (0.956)	4.15±0.14 (0.952)	3.98±0.11 (0.958)

Representation-level smoothing gives the smaller set in 10 of the 12 cells, with differences below 0.02 in the two exceptions. The largest gap occurs on CIFAR-100 at two shots, where representation smoothing gives 2.43 labels versus 2.82 for score-level smoothing, compared with 3.51 without either correction. Combining the two offers little additional benefit on CIFAR-10 and CIFAR-100, while on EuroSAT it gives the smallest set at every budget, reaching 1.40 labels versus 1.50 at eight shots. Overall, the comparison supports applying the local correction before conformal scoring.

C.4 Refinement and the score distribution

Figure 5 illustrates the effect of refinement on the score distribution. On CIFAR-10 and CIFAR-100, refinement increases the separation between true-class and false-class scores, placing more true scores below the calibration threshold and more false scores above it. EuroSAT retains substantially more overlap, consistent with its smaller set-size improvement in Table 1.

C.5 Runtime analysis

Table 7 reports the time required to construct prediction sets for a batch of 500 test examples after all precomputation is complete. The naive full conformal sweep, which re-scores the labeled sample for every candidate label and test example, requires 35–83 seconds, while the cached implementation in Algorithm 1 produces bit-for-bit identical p-values in approximately half a second, giving an 80–130× speedup. The closed-form computation is derived in Appendix A.1. The result is that exact full conformal inference becomes practical at the reported batch size, while the inductive baselines remain faster because they only threshold the outputs of an already-trained classifier.

Table 6: Score-level versus representation-level smoothing: mean prediction-set size (± SE, 50 trials), with marginal coverage in parentheses, at $\alpha = 0.1$. All arms start from truncation+whitening and differ only in where the pool correction acts, at the representation level (+ smoothing) or the score level (+ SNAPS). SNAPS uses its leave-one-out variant at $\eta = 0.5$ and $k = 10$.

dataset	arm	2 shots	4 shots	6 shots	8 shots
CIFAR-10	T+W	1.19±0.05 (0.967)	1.03±0.01 (0.979)	**1.00±0.00 (0.978)**	**1.00±0.00 (0.972)**
	+ smoothing	**1.02±0.01 (0.978)**	**1.00±0.00 (0.984)**	**1.00±0.00 (0.983)**	**1.00±0.00 (0.976)**
	+ SNAPS	1.08±0.02 (0.974)	1.01±0.00 (0.984)	**1.00±0.00 (0.983)**	**1.00±0.00 (0.977)**
	+ both	**1.02±0.01 (0.978)**	**1.00±0.00 (0.984)**	**1.00±0.00 (0.984)**	**1.00±0.00 (0.978)**
CIFAR-100	T+W	3.51±0.13 (0.952)	1.58±0.02 (0.914)	1.39±0.01 (0.909)	1.32±0.01 (0.905)
	+ smoothing	2.43±0.09 (0.940)	**1.42±0.02 (0.910)**	**1.31±0.01 (0.904)**	1.28±0.01 (0.903)
	+ SNAPS	2.82±0.09 (0.950)	1.46±0.02 (0.914)	1.32±0.01 (0.909)	**1.26±0.01 (0.904)**
	+ both	**2.41±0.09 (0.939)**	**1.42±0.02 (0.908)**	**1.31±0.01 (0.902)**	1.27±0.01 (0.901)
EuroSAT	T+W	4.28±0.22 (0.935)	2.48±0.12 (0.907)	2.07±0.07 (0.912)	1.77±0.05 (0.904)
	+ smoothing	3.54±0.18 (0.941)	2.01±0.09 (0.913)	1.69±0.05 (0.909)	1.50±0.04 (0.899)
	+ SNAPS	3.52±0.17 (0.940)	2.06±0.08 (0.910)	1.74±0.06 (0.912)	1.54±0.04 (0.901)
	+ both	**3.43±0.19 (0.945)**	**1.85±0.07 (0.910)**	**1.63±0.06 (0.914)**	**1.40±0.03 (0.898)**

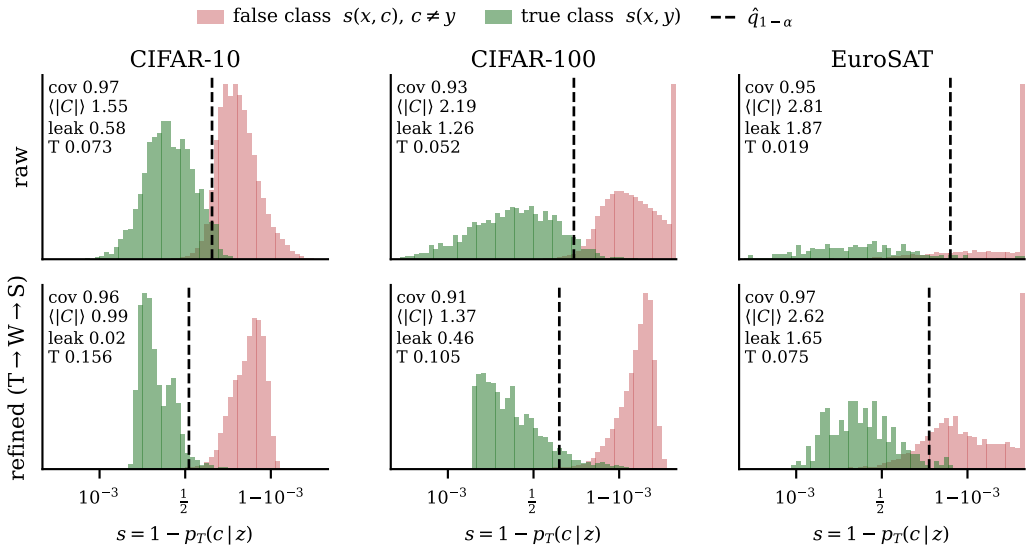


Figure 5: Distribution of the deployed prototype-softmax score $s = 1 - p_T(c|z)$ over the test set before (top) and after (bottom) frozen refinement T→W→S. True-class and false-class scores are shown at four labels per class and $\alpha = 0.1$. The dashed line marks the calibration quantile $\hat{q}_{1-\alpha}$. The score axis is logit-scaled and bar heights denote per-class probability mass per bin. Annotations report coverage, mean set size, and leak (mean number of false labels in the set).

Table 7: Test-time compute in seconds on CIFAR-100 ($K = 100$, 500 test points, median over 10 trials): the time to return prediction sets for a new test batch, with all fitting and calibration done beforehand. The naive sweep returns identical p-values through the unvectorized per-candidate loop.

method	2 shots	4 shots	6 shots	8 shots
Split CP	0.04	0.04	0.04	0.04
CV+	0.03	0.04	0.04	0.05
SemiCP	0.04	0.04	0.03	0.04
full CP, naive sweep	35.39	45.70	62.37	83.16
ReProCP (ours)	0.44	0.44	0.51	0.63