Synthetic-Powered Predictive Inference

Meshi Bashari[*1] **Roy Maor Lotan**[*1] **Yonghoon Lee**[*2] **Edgar Dobriban** [2] **Yaniv Romano**[1,3]

[1]Department of Electrical and Computer Engineering, Technion—Israel Institute of Technology
[2]Department of Statistics and Data Science, The Wharton School, University of Pennsylvania, USA
[3]Department of Computer Science, Technion—Israel Institute of Technology

Abstract

Conformal prediction is a framework for predictive inference with a distribution-free, finite-sample guarantee. However, it tends to provide uninformative prediction sets when calibration data are scarce. This paper introduces Synthetic-powered predictive inference (`SPI`), a novel framework that incorporates synthetic data—e.g., from a generative model—to improve sample efficiency. At the core of our method is a score transporter: an empirical quantile mapping that aligns nonconformity scores from trusted, real data with those from synthetic data. By carefully integrating the score transporter into the calibration process, `SPI` provably achieves finite-sample coverage guarantees without making any assumptions about the real and synthetic data distributions. When the score distributions are well aligned, `SPI` yields substantially tighter and more informative prediction sets than standard conformal prediction. Experiments on image classification—augmenting data with synthetic diffusion-model generated images—and on tabular regression demonstrate notable improvements in predictive efficiency in data-scarce settings.

1 Introduction

1.1 Background and motivation

Conformal prediction [48, 58, 59] is a general framework for quantifying predictive uncertainty, providing finite-sample statistical guarantees for any machine learning model. Given a test instance with an unknown label (e.g., an image), conformal prediction constructs a prediction set—a collection of plausible labels guaranteed to include the true label with a user-specified coverage probability (e.g., 95%). To do so, it relies on a labeled holdout calibration set to compute nonconformity scores, which measure how well a model's prediction aligns with the true labeled outcome. These scores are then used to assess uncertainty in future predictions. Crucially, the coverage guarantee holds whenever the calibration and test data are exchangeable (e.g., i.i.d.), without any assumption on the sampling distribution.

While conformal prediction offers a powerful coverage guarantee, its reliance on a holdout set limits its effectiveness when labeled data is scarce—becoming unstable and highly variable in coverage, or overly conservative and uninformative. As a result, it offers limited value in applications where labeled data is inherently limited, such as those requiring personalization or subgroup-specific guarantees. Importantly, this is not merely an abstract concern [7]—for example, in medical settings, it is natural to seek valid inference tailored to specific patient characteristics such as age, health condition, and/or other group identifiers of interest, see e.g., [13, 38]. Similarly, in image classification tasks, one may wish to ensure that coverage holds for the true class label, see e.g., [57]. In these cases and many others, we often have only a few representative holdout examples for each group or class, which severely restricts the applicability of standard conformal prediction.

[*]Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Meanwhile, we are witnessing rapid progress in the ability to train accurate machine learning models even under data-scarce settings, driven by the rising quality of synthetic data produced by modern generative models and by advances in domain adaptation, see e.g., [11]. These developments inspire the question we pursue in this work: *Can we rigorously enhance the sample efficiency of conformal prediction by leveraging a large pool of synthetic data—such as labeled datapoints from related subpopulations, or even data sampled from generative models?*

At first glance, it may appear hard to use synthetic data to boost sample efficiency in a statistically valid way. After all, the distribution of synthetic data can be completely different from that of the data of interest.[1] Overcoming this challenge, we propose a principled framework that unlocks conformal prediction with the ability to incorporate synthetic data while preserving rigorous, model-agnostic, non-asymptotic coverage guarantees. Crucially, our method—`SPI`—provides a coverage bound that requires no assumptions about the similarity between the real and synthetic data distributions. Still, when the distribution of synthetic and real scores is close, our approach yields a substantial boost in sample efficiency—resulting in more informative prediction sets than those produced by standard conformal prediction. A discussion of related literature is deferred to Appendix B.

1.2 Preview of the proposed method and our key contributions

Our key innovation is the introduction of the *score transporter*: a data-driven empirical quantile mapping function that transports the real calibration scores to resemble the synthetic scores. This mapping enables the construction of prediction sets for new test datapoints, leveraging the abundance of synthetic data. Crucially, the score transporter does not require data splitting, allowing full use of the real and synthetic calibration data. Furthermore, we develop a computationally efficient algorithm with a runtime complexity similar to that of standard conformal prediction. A pictorial illustration of our proposed calibration framework is provided in Figure 1.

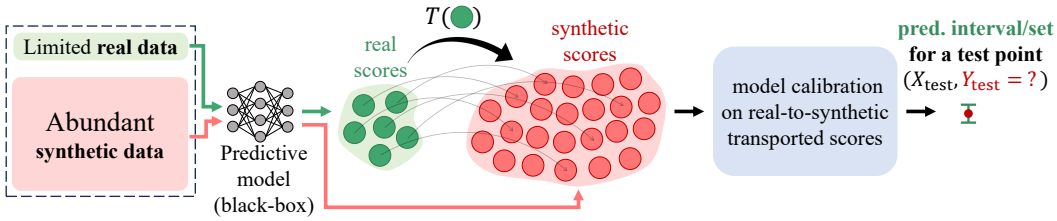


Figure 1: **A high-level overview of the proposed method**. The approach leverages a small labeled real dataset alongside a large labeled synthetic dataset. The *score transporter* maps scores from the real domain to the synthetic one. Calibration is then performed using the transported real scores and the synthetic scores.

We support our proposed `SPI` framework with two theoretical guarantees, where the final coverage rate bound is the tighter of the two. The first shows that when the synthetic and real score distributions are close, the achieved coverage closely matches the desired level. More generally, it characterizes how the distributional shift between the real and synthetic scores and the construction of the score transporter affect the realized coverage.

The second theoretical result complements the first by providing worst-case bounds on the coverage probability, even if the synthetic data are of poor quality. This bound is directly controlled by the user, allowing them to set a "guardrail," i.e., a lower bound on the coverage probability (say 90%) that holds regardless of the distribution of the synthetic data. Remarkably, this bound holds even when the synthetic data depend on the real calibration set. This flexibility enables users to adapt or filter the synthetic data to improve efficiency, for example, by selecting datapoints that resemble the real ones—all without requiring any data splitting.

We demonstrate the practicality of our method on multi-class classification and regression tasks. For image classification on ImageNet, we explore two practical strategies for constructing synthetic data. The first leverages a generative model (Stable Diffusion [46] or FLUX [30]) to generate artificial images for each class. The second uses another set of real data, drawn from a different

[1]We refer to the limited dataset of interest as the *real calibration set*, to distinguish it from the (potentially synthetic) calibration data.

distribution, as the synthetic data. In the regression setting, we consider tabular panel data, using past panels as synthetic data and a recent panel as the real calibration set. Across all experiments, our method shows improvements in statistical efficiency—even when the real calibration set is very small, with as few as 15 datapoints. Software for reproducing the experiments is available at https://github.com/Meshiba/spi.

2 Problem setup

Consider the standard setting of a prediction problem where we have m i.i.d. (real) calibration datapoints[2] $(X_i, Y_i)_{i\in[m]} \overset{\text{iid}}{\sim} P_{X,Y} = P_X \times P_{Y|X}$, for $i \in [m] := \{1, \ldots, m\}$ on $\mathcal{X} \times \mathcal{Y}$, where each $X_i \in \mathcal{X}$ represents the features and $Y_i \in \mathcal{Y}$ denotes the label or outcome for the i-th datapoint. Given a new test input $X_{m+1} \sim P_X$, the task is to construct a prediction set $\widehat{C}(X_{m+1})$ for the unknown label Y_{m+1} with the following distribution-free coverage guarantee:

$$\mathbb{P}_{(X_i,Y_i)_{i\in[m+1]}\overset{\text{iid}}{\sim}P_{X,Y}}\left\{Y_{m+1} \in \widehat{C}(X_{m+1})\right\} \geq 1-\alpha, \text{ for any distribution } P_{X,Y} \text{ on } \mathcal{X}\times\mathcal{Y}, \quad (1)$$

where $1-\alpha \in (0,1)$ is a predetermined target level of coverage. Here and below, we abbreviate by $(a_i)_{i\in[k]}$ vectors $(a_1, a_2, \cdots, a_k)$. While standard conformal prediction [4, 39, 58, 59] has this property, its efficiency can be limited when the calibration sample size m is small.

Suppose now that we also have access to a set of synthetic datapoints $(\tilde{X}_j, \tilde{Y}_j)_{j\in[N]} \overset{\text{iid}}{\sim} Q_{X,Y}$. These could be datapoints collected from related distributions, sampled from a generative model, or obtained otherwise. We hope that $Q_{X,Y}$ is close to $P_{X,Y}$, but do not assume this. We are interested in the setting where $m \ll N$, aiming to improve inference with a small calibration set by leveraging a large synthetic dataset. To make this concrete, we aim to construct a prediction set map $\widehat{C} : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$—where $\mathcal{P}(\mathcal{Y})$ is the set of subsets of $\mathcal{Y}$—as a function of the datasets $(X_i, Y_i)_{i\in[m]}$ and $(\tilde{X}_j, \tilde{Y}_j)_{j\in[N]}$, such that the prediction set $\widehat{C}(X_{m+1})$ satisfies, for any distributions $P_{X,Y}$ and $Q_{X,Y}$ on $\mathcal{X} \times \mathcal{Y}$,

$$\mathbb{P}_{(X_i,Y_i)_{i\in[m+1]}\overset{\text{iid}}{\sim}P_{X,Y},(\tilde{X}_i,\tilde{Y}_i)_{i\in[N]}\overset{\text{iid}}{\sim}Q_{X,Y}}\left\{Y_{m+1} \in \widehat{C}(X_{m+1})\right\} \geq 1-\alpha. \quad (2)$$

For the classification task where Y is discrete, we extend our discussion beyond the marginal coverage guarantee in (1) and consider the following label-conditional coverage guarantee [57, 58]:

$$\mathbb{P}\left\{Y_{m+1} \in \widehat{C}(X_{m+1}) \,\middle|\, Y_{m+1} = y\right\} \geq 1-\alpha, \text{ for all } y \in \mathcal{Y}, \quad (3)$$

where, as before, we aim for a distribution-free guarantee, under any distributions $P_{X,Y}$ and $Q_{X,Y}$—although the inequality in (3) is written in a simplified form.

2.1 Background: split conformal prediction

Split conformal prediction [39] is an approach to attain the coverage guarantee (1). The first step is to construct a nonconformity score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ from an independent dataset.[3] Next, we compute the scores on the calibration datapoints: $S_i = s(X_i, Y_i)$ for $i \in [m]$. The prediction set is then given as

$$\widehat{C}(X_{m+1}) := \left\{y \in \mathcal{Y} : s(X_{m+1}, y) \leq \hat{Q}_{1-\alpha}\right\}, \quad (4)$$

where $\hat{Q}_{1-\alpha}$ denotes the $\lceil (1-\alpha)(m+1) \rceil$-th smallest score from the (multi-)set $(S_i)_{i\in[m]}$.

If the scores $(S_i)_{i\in[m]}$ are distinct almost surely, then the split conformal prediction set (4) attains the following coverage bounds [39, 58, 59]:

$$1-\alpha \leq \mathbb{P}\left\{Y_{m+1} \in \widehat{C}(X_{m+1})\right\} \leq 1-\alpha+1/(m+1). \quad (5)$$

If m is very small, the split conformal set might be conservative. In particular, if $m+1 < 1/\alpha$, then the only way to achieve $1-\alpha$ coverage with m datapoints is by producing a trivial prediction set that

[2] Some of our results rely on a weaker assumption than i.i.d.—namely, exchangeability of the real calibration datapoints.

[3] A typical example for regression problems is $s(x, y) = |y - \hat{\mu}(x)|$, where $\hat{\mu}$ is a predictor pre-trained on a separate dataset; see e.g., [4, 58], etc.

includes all labels. For a typical value of $\alpha = 0.05$, this is the case when $m < 19$. Since we aim to handle situations with very low sample sizes, this motivates us to develop a procedure capable of producing more informative prediction sets by leveraging synthetic data.

3 Methodology

3.1 Synthetic-powered predictive inference

In this section, we introduce our method—SPI—which is designed to leverage the synthetic datapoints to effectively increase the sample size, thereby producing a non-conservative prediction set. We construct a split-conformal-type method that performs inference based on pre-constructed nonconformity scores. Throughout the section, we assume that the score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is fixed, and denote the real and synthetic scores as $S_i = s(X_i, Y_i)$ for $i \in [m+1]$ and $\tilde{S}_j = s(\tilde{X}_j, \tilde{Y}_j)$ for $j \in [N]$, respectively. Here S_{m+1} is the unobserved test score.

Our strategy is to construct a score-transporter T that maps a real score to a synthetic score—as a function of the observed scores. We then run split conformal prediction on the synthetic scores and apply T to obtain a prediction set for the real score S_{m+1}. A carefully constructed map T can generate a prediction set with a theoretically controlled coverage rate, while effectively leveraging the large synthetic dataset. The procedure has three steps.

Step 1. Construct windows in the space of synthetic scores. Denote by $S_{(1)}, \ldots, S_{(m+1)}$ the real scores arranged in increasing order. We first define a "window" $I_m(r)$ designed to contain the r-th score $S_{(r)}$ for each $r \in [m+1]$, as follows:

$$R_r^- = \max\left\{t \in [N+1] : F(t-1) \leq \tfrac{\beta}{2}\right\}, \; R_r^+ = \min\left\{t \in [N+1] : F(t) \geq 1 - \tfrac{\beta}{2}\right\}, \quad (6)$$

where $\beta \in (0,1)$ is a predefined level, and $F := F_{m,N,r}$ is defined as[4]

$$F(t) = \sum_{k=1}^{t} p_{m,N,r}(k), \text{ with } p_{m,N,r}(k) = \tbinom{k+r-2}{r-1}\tbinom{N+m-k-r+2}{m-r+1}/\tbinom{N+m+1}{m+1}.$$

Then, with the synthetic scores $\tilde{S}_{(1)}, \ldots, \tilde{S}_{(N)}$ in increasing order, we construct the window as

$$I_m(r) = [L_m(r), U_m(r)], \quad \text{where} \quad L_m(r) := \tilde{S}_{(R_r^-)} \text{ and } U_m(r) := \tilde{S}_{(R_r^+)}, \quad (7)$$

and where $\tilde{S}_{(N+1)} = +\infty$. This window is designed to satisfy the following property, where we denote the distribution of $s(X,Y)$ under $P_{X,Y}$ and $Q_{X,Y}$ by P and Q, respectively:

Lemma 3.1. *If $P = Q$ and both are continuous distributions, then $\mathbb{P}\left\{S_{(r)} \in I_m(r)\right\} \geq 1 - \beta$ for all $r \in [m+1]$.*

The proof is deferred to Appendix G. Intuitively, the window $I_m(r)$ represents a region in the synthetic score space where $S_{(r)}$ is likely to lie, and the transporter we construct in the next step maps the real score to an element within its corresponding window.

Step 2. Construct the score-transporter. We now define the map $T(\cdot) = T(\cdot; (S_i)_{i\in[m]}, (\tilde{S}_j)_{j\in[N]})$ mapping real to synthetic scores as follows. For a scalar η, let $r_\eta = \sum_{i=1}^m \mathbb{1}\{S_i < \eta\} + 1$ denote the rank of η among $(S_1, \ldots, S_m, \eta)$ in increasing order, and with L_m, U_m from (7), define

$$T(\eta) = \begin{cases} U_m(r_\eta), & \text{if } \eta \geq U_m(r_\eta), \\ \mathrm{NN}_m^-(r_\eta, \eta), & \text{if } L_m(r_\eta) \leq \eta < U_m(r_\eta), \\ L_m(r_\eta), & \text{if } \eta < L_m(r_\eta), \end{cases} \quad (8)$$

where the *lower nearest neighbor* NN_m^- is defined as

$$\mathrm{NN}_m^-(r, \eta) := \max_{R_r^- \leq j \leq R_r^+} \left\{\tilde{S}_{(j)} : \tilde{S}_{(j)} \leq \eta\right\}.$$

[4]Here, for non-negative integers $a \leq b$, $\binom{b}{a} = b!/(a!(b-a)!)$ denotes the binomial coefficient, where $x! = x \cdot (x-1) \cdot \ldots \cdot 1$ is the factorial of a non-negative integer x. Also, $p_{m,N,r}(k)$ is the *probability mass function of the r-th order statistic* from a random sample of size $m+1$ drawn *without replacement* from a finite population of size $N+m+1$ [e.g., 62, p. 243].

Roughly speaking, the score-transporter T maps η to a synthetic score in the corresponding window $I_m(r_\eta)$ that is closest to η. The lower nearest neighbor NN_m^- is chosen carefully to act as a lower bound on the score, ensuring that the coverage can be tightly controlled.

Step 3. Conformal prediction after transport-mapping. Applying the score-transporter T to a hypothetical score $s(X_{m+1}, y)$, we construct the prediction set as those y values for which this mapped value lies in the conformal prediction region constructed from the synthetic data:

$$\widehat{C}(X_{m+1}) = \left\{ y \in \mathcal{Y} : T\left(s(X_{m+1}, y)\right) \leq \tilde{Q}_{1-\alpha} \right\}. \tag{9}$$

Here $\tilde{Q}_{1-\alpha}$ is the $\lceil (1-\alpha)(N+1) \rceil$-th smallest score in $(\tilde{S}_j)_{j\in[N]}$. We term this procedure *Synthetic-powered predictive inference* (`SPI`). Figure S1 presents a schematic overview of `SPI` with two candidate labels, illustrating each of the steps discussed above. Building on the ideas of [57, 58], we extend our proposed method to achieve label-conditional coverage guarantees in Appendix F.

3.2 Simplifying the computation of `SPI`

Since $T(\cdot) = T(\cdot; (S_i)_{i\in[m]}, (\tilde{S}_j)_{j\in[N]})$ depends on $(S_i)_{i\in[m]}$ and $(\tilde{S}_j)_{j\in[N]}$, the prediction set $\widehat{C}(X_{m+1})$ in (9) has an *a priori* potentially complex dependence on y. Fortunately, the prediction set simplifies to the following formula, which is fast to compute:

$$\widehat{C}^{\mathrm{fast}}(x) = \left\{ y \in \mathcal{Y} : s(x,y) \leq \max\{\min\{\tilde{Q}'_{1-\alpha}, S_{(\tilde{R}^-)}\}, S_{(\tilde{R}^+)}\} \right\}, \; S_{(m+1)} = +\infty. \tag{10}$$

Here $\tilde{Q}'_{1-\alpha}$ is the $(\lceil (1-\alpha)(N+1) \rceil + 1)$-th smallest score among $(\tilde{S}_j)_{j\in[N]}$, and

$$\tilde{R}^{\pm} = \max\{r \in [m+1] : R_r^{\pm} \leq \lceil (1-\alpha)(N+1) \rceil\}. \tag{11}$$

The following result shows that the prediction set $\widehat{C}^{\mathrm{fast}}$ is equivalent to the prediction set (9)—here, for two sets A and B, $A \triangle B$ denotes the symmetric set difference $(A \cap B^c) \cup (A^c \cup B)$.

Proposition 3.2. *Recall the prediction sets $\widehat{C}$ from* (9) *and $\widehat{C}^{\mathrm{fast}}$ from* (10)*. If Q is continuous, then*

$$\mathbb{P}\left\{ \{Y_{m+1} \in \widehat{C}(X_{m+1})\} \triangle \{Y_{m+1} \in \widehat{C}^{\mathrm{fast}}(X_{m+1})\} \right\} = 0.$$

Based on this simplification, we present the complete `SPI` procedure in Algorithm 1.

3.3 Theoretical guarantees

We now derive bounds on the coverage rate of the `SPI` prediction set (9). The first bound shows that when the real and synthetic scores are similar (as measured by total variation distance), our method has a tight coverage around the desired level.

Theorem 3.3 (Coverage depending on the closeness of real and synthetic distributions)**.** *Suppose the real calibration set $(X_i, Y_i)_{i\in[m]}$ is exchangeable with the test point (X_{m+1}, Y_{m+1}) and the synthetic calibration datapoints $(\tilde{X}_j, \tilde{Y}_j)_{j\in[N]}$ are drawn i.i.d., where the distribution Q of their scores is continuous. Let $P_{(r)}^{m+1}$ and $Q_{(r)}^{m+1}$ denote the distribution of the r-th order statistic among $m+1$ i.i.d. draws from P and Q, respectively. Then the prediction set $\widehat{C}(X_{m+1})$ from* (9) *satisfies*

$$1 - \alpha - \beta - \varepsilon_{P,Q}^{m+1} \leq \mathbb{P}\left\{ Y_{m+1} \in \widehat{C}(X_{m+1}) \right\} \leq 1 - \alpha + \beta + \varepsilon_{P,Q}^{m+1} + 1/(N+1),$$

where $\varepsilon_{P,Q}^{m+1} = \frac{1}{m+1} \sum_{i=1}^{m+1} \mathrm{d_{TV}}(P_{(i)}^{m+1}, Q_{(i)}^{m+1})$ and $\mathrm{d_{TV}}$ denotes the total variation distance.

Note that $\varepsilon_{P,Q}^{m+1}$ is bounded above by the total variation distance between P and Q. When $P = Q$, we have $\varepsilon_{P,Q}^{m+1} = 0$, and thus our procedure provides a tighter upper bound than split conformal prediction using only the real calibration data (5) when $\beta + 1/(N+1) \leq 1/(m+1)$. When $N \gg m$ and $\beta \ll 1/m$, our method offers a tighter coverage. In practice, however, we often observe tight coverage even for relatively large β—in the proof, the $\pm\beta$ term arises from a union bound that accounts for the case where the test score S_{m+1} is not covered by the corresponding window.

Remark 3.4. The continuity of the score distribution required in Theorem 3.3 can generally be attained conveniently. For example, in settings where the originally constructed score outputs discrete values, one can simply add a negligible amount of i.i.d. Uniform$[-\delta, \delta]$ noise to the scores, so that the perturbed scores—which are nearly identical to the original ones—have a continuous distribution.

When the distributions P and Q differ greatly, the bounds in Theorem 3.3 may be loose, as they do not sufficiently account for the adjustment introduced by the map T under distribution shift. Below, we provide alternative worst-case bounds for the coverage rate of the `SPI` prediction set, which depend only on the sample sizes, and hold regardless of the relationship between Q and P.

Theorem 3.5 (Worst-case coverage)**.** *Suppose that the real calibration set $(X_1, Y_1), \ldots, (X_m, Y_m)$ is exchangeable with the test point (X_{m+1}, Y_{m+1}), and that the synthetic score distribution Q is continuous. Then the prediction set $\widehat{C}(X_{m+1})$ in* (9) *satisfies*

$$\frac{|\{j \in [m+1] : R_j^+ \leq \lceil (1-\alpha)(N+1) \rceil\}|}{m+1} \leq \mathbb{P}\left\{Y_{m+1} \in \widehat{C}(X_{m+1})\right\} \leq \frac{|\{j \in [m+1] : R_j^- \leq \lceil (1-\alpha)(N+1) \rceil\}|}{m+1}.$$

While this result is somewhat non-explicit, the bounds can be computed fast, and remain close to the target level $1-\alpha$, as illustrated in Section D.1 through a set of plots. We emphasize that the bounds hold due to the careful construction of the score transport map from (8), and would not hold if we were to simply mix together the real and synthetic data. Moreover, the bounds impose no condition on the distribution of the synthetic scores—it is even allowed for the synthetic scores to depend on the real calibration set. This provides significant flexibility in the choice of synthetic data even with a separate score function; see Section 3.4.

The bounds in Theorem 3.5 depend solely on the constants m, N, α, and β. This allows us to adjust the levels α and β to achieve a specific lower bound, say $1-\alpha'$, for a predetermined value of α'—which implies that the guarantee (2) can be achieved. However, we advise using our procedure without level adjustment, in the spirit of Theorem 3.3, since Theorem 3.5 provides worst-case bounds. In practice, we recommend setting β to meet a user-tolerable guardrail level—for instance, 90% worst-case coverage when the target level is $1-\alpha = 95\%$. Algorithm 4 outlines a procedure to compute such a β for given N, m, and α, ensuring the user-specified guardrail coverage bound is satisfied.

Having stated the two theoretical guarantees separately, we pause to highlight the following:

Remark 3.6 (Effective coverage guarantee of `SPI`). The coverage guarantees of `SPI` are adaptive to the quality of the synthetic data, in the sense that the effective theoretical bound automatically takes the tighter of Theorems 3.3 and 3.5.

3.4 Improving the quality of synthetic scores

The quality of the `SPI` prediction set depends on how well the distribution of the synthetic score Q approximates the true score distribution P, as supported by Theorem 3.3. In practice, the alignment between synthetic and real scores can be assessed using standard goodness-of-fit tests on the empirical cumulative distribution functions (CDFs), such as Cramér–von Mises or Kolmogorov–Smirnov tests. These tests can help detect when the synthetic scores deviate substantially from the real scores, indicating that the synthetic data may be less useful for the inference task or could be modified to better align with the real data by carefully constructing the synthetic scores.

For instance, we can seek a map g such that the distribution of the adjusted synthetic score $\tilde{S}'_j = g(\tilde{S}_j)$ better approximates the true distribution P. More generally, we may construct a separate score function $\tilde{s} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ for the synthetic data, so that the distribution of the synthetic score $\tilde{s}(\tilde{X}, \tilde{Y})$ better approximates that of the real score $s(X, Y)$. Or, we may select a subset of the synthetic scores that is expected to provide a better approximation. Below, we present two approaches to improve the quality of synthetic scores.

3.4.1 Constructing a separate synthetic score function

We first discuss the approach of constructing a separate synthetic score function $\tilde{s}$. For example, one might choose to construct $\tilde{s}$ using a split of the real data and a split of the synthetic data. However, if

the original data sample size m is small—which is the main focus of this work—we may prefer to reuse the data both for constructing the adjustment function or synthetic scores and for performing inference. Therefore, in this section, we focus on data-dependent score construction, while the details of the data-splitting-based approach are deferred to Section E.

For example, one might consider constructing an adjustment function g as $s \mapsto g(s) := \hat{\theta}_1 s + \hat{\theta}_2$, where the parameters $(\hat{\theta}_1, \hat{\theta}_2)$ are fitted using the calibration scores $(S_i)_{i\in[m]}$ and $(\tilde{S}_j)_{j\in[N]}$ via least squares:

$$(\hat{\theta}_1, \hat{\theta}_2) = \underset{a,b}{\operatorname{argmin}} \sum_{i=1}^{m} |a \cdot \tilde{S}_{(\lfloor iN/m \rfloor)} + b - S_{(i)}|^2,$$

and then constructing the adjusted synthetic scores $(g(\tilde{S}_j))_{j\in[N]}$, setting $\tilde{s} = g \circ s$; see [63] for a more sophisticated approach to learning the score function $\tilde{s}$.

For such a synthetic score $\tilde{s}$ constructed in a data-dependent manner, can we still expect a provable coverage bound? The answer is yes, since the bounds in Theorem 3.5 hold for synthetic scores with arbitrary dependence on the real calibration set.

Corollary 3.7. *Suppose the synthetic score function $\tilde{s}$ is constructed using both the real data $(X_i, Y_i)_{i\in[m]}$ and the calibration data $(\tilde{X}_j, \tilde{Y}_j)_{j\in[N]}$. Then the prediction set $\widehat{C}$ from* (9)*, constructed using $\tilde{S}_j = \tilde{s}(\tilde{X}_j, \tilde{Y}_j)$ for $j \in [N]$, attains the bounds stated in Theorem 3.5.*

3.4.2 Constructing a subset of synthetic data

Now, we shift to a different approach for improving the quality of synthetic scores: constructing a subset of the synthetic data that is more relevant for inference on the real data. This approach is particularly useful when the synthetic data comes from different sources, rather than sampled from a generative model. The idea is to select synthetic datapoints based on how well they approximate the real data. Then, we form a subset consisting of points with high approximation quality. Again, since Theorem 3.5 imposes no condition on the joint distribution of the synthetic scores, the bounds also hold for the `SPI` prediction set constructed with this subset of synthetic scores.

Corollary 3.8. *Let $I_{subset} = \{j_1, \cdots, j_{\tilde{N}}\} \subset [N]$ denote the indices of a subset of synthetic data points, and suppose that $\tilde{N} = |I_{subset}|$ is fixed. Then the prediction set $\widehat{C}$ from* (9)*, constructed using $(\tilde{S}_{j_l})_{l\in[\tilde{N}]}$ as the synthetic scores, satisfies the bounds stated in Theorem 3.5, with N replaced by $\tilde{N}$.*

Note that this result requires the number of selected points $\tilde{N}$ to be fixed.[5] For example, one can use a nearest-neighbor procedure, in which we partition the synthetic data into subsets of a fixed size n, and then select k subsets whose score distributions most closely resemble that of the real data, resulting in $\tilde{N} = nk$ synthetic data points (see Algorithm 2).

4 Experiments

In this section, we compare the performance of the proposed `SPI` procedure to that of standard conformal prediction in a setting where a small real calibration set and a large synthetic calibration set are available, each drawn from distinct and unknown distributions. Further experiments on simulated data, where the underlying distributions are known, are presented in Appendix I.

Setup and performance metrics We randomly sample two disjoint subsets from the real data, assigning one as the real calibration set and the other as the test set. Additionally, we sample a synthetic calibration set from the synthetic data, which is intentionally larger than the real calibration set, aligning with the focus of this paper. The test set is used to evaluate the procedure based on two metrics: the coverage rate, and the prediction set size (for classification problems) or prediction interval width (for regression problems). We report the results from 100 repeated trials, each with different random calibration, test, and synthetic datasets.

Methods We compare the following methods: `OnlyReal`—standard split conformal prediction [39] using the real calibration set; `OnlySynth`—conformal prediction applied to the synthetic calibration

[5] More generally, if $\tilde{N}$ is random but independent of the real scores, the same bounds hold for the conditional probability $\mathbb{P}\{Y_{m+1} \in \widehat{C}(X_{m+1}) \mid \tilde{N}\}$.

set as if it were real, which does not provide coverage guarantees; and SPI (ours)—the proposed procedure outlined in Algorithm 1, applied with $\beta = 0.4$.

4.1 Multi-class classification on the ImageNet data

We begin by evaluating our method on a multi-class classification task using the ImageNet dataset [14]. In particular, we aim for marginal (1) and label-conditional coverage guarantees (3); the latter requires hold-out data for each class. Since our experiments involve generating thousands of images per class, we restrict our study to a subset of 30 classes (listed in Table S1) that form the real population.

We consider two scenarios for constructing the synthetic data. In the first scenario, we apply a generative model to produce synthetic images. In the second scenario, the synthetic set is formed using real images drawn from classes not included in the real population.

Across all experiments and methods, we use a CLIP model [43] as the predictive model, along with the adaptive prediction sets (APS) score function [45]. We also include additional experiments with the homogeneous prediction sets (HPS) score function [58] in Appendix J.3. Importantly, CLIP is not trained on ImageNet images. Additional details on the score functions and pre-trained model are provided in Appendices C.1 and H.3, respectively.

4.1.1 SPI with generated synthetic data

We use Stable Diffusion [46] to generate synthetic images resembling those in ImageNet. Figure 2 shows representative examples, including images from an additional generative model discussed later. Additional examples and further details are provided in Figure S5 and Appendix H.4.

Figure 2: Examples of real and generated images for the *golden retriever* class. The first column displays real ImageNet images, while the remaining columns show generated samples. The top row contains images generated by Stable Diffusion [46], and the bottom row by FLUX [30].

For the marginal coverage experiments, we randomly select $m = 15$ ImageNet images from the real data, chosen from among 30 classes, to construct the real calibration set. The test set consists of 15,000 real images, and the synthetic calibration set includes $N = 1,000$ generated images, sampled uniformly across all classes. For the label-conditional experiments, we randomly select $m = 15$ real images for each of the $k = 30$ classes to form the real calibration set (resulting in $mk = 450$ real data points), 500 real images per class to form the test set, and $n = 1,000$ generated images per class to form the synthetic calibration set (resulting in $N = nk = 30,000$ synthetic data points).

Figure 3 presents the performance of various methods under both marginal and label-conditional guarantees at target coverage level $1 - \alpha = 0.95$. The label-conditional results are shown for five representative classes. The observations below apply to both the marginal and label-conditional settings. We observe that OnlyReal controls the coverage at level $1 - \alpha = 0.95$. However, it remains conservative due to the small size of the real calibration set, which results in trivial prediction sets. The OnlySynth approach fails to achieve the target coverage level of $1 - \alpha$, exhibiting undercoverage for some classes. This violation arises from the distribution shift between the real and synthetic data.

In contrast, the proposed method, SPI, achieves coverage within the theoretical bounds established in Theorem 3.5. For example, for the "Siberian husky" class, where the synthetic images differ significantly from the real ones, SPI still produces informative prediction sets. For classes where the synthetic and real data are more aligned, such as the "lighter" class, SPI shows low variance in coverage with smaller prediction set sizes.

We provide results for additional α levels in Appendix J.1. Further experiments on the effect of the size of the real calibration set and the parameter β on the performance of SPI are presented

in Appendices J.1.1 and J.1.2, respectively. In addition, we provide experiments using the FLUX generative model [30], which exhibit similar trends to those observed with Stable Diffusion; see Appendix J.1.3. Examples of generated images and details are provided in Figures 2 and S6 and Appendix H.4, respectively.

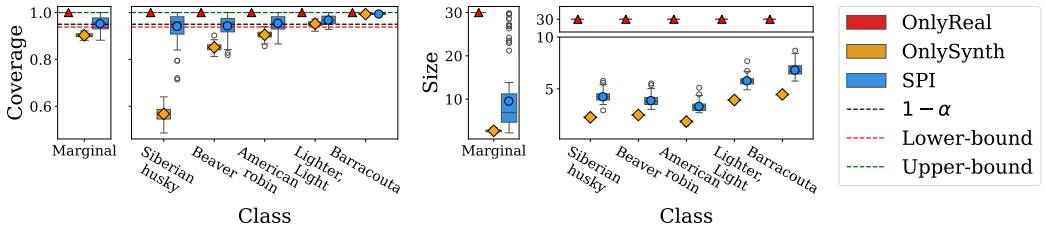


Figure 3: Results for the ImageNet data: Coverage rates of `OnlyReal`, `OnlySynth`, and `SPI` at target level $1 - \alpha = 0.95$, averaged over 100 trials. Left: Average coverage. Right: Average prediction set size, both under marginal (leftmost box in each group) and label-conditional coverage settings. Label-conditional results are shown for selected classes; see Table S3 for results across all classes.

4.1.2 `SPI` with synthetic data from k-nearest subset selection

We now explore the performance of the subset-based variant of our approach, referred to as `SPI-Subset` and described in Algorithm 2. The experiments in this section reflect scenarios where a generative model is unavailable.

As before, we aim to control both marginal and label-conditional coverage. In the marginal setting, we randomly select $m = 15$ real images, across 30 classes, to form the real calibration set, and 15,000 real images from the same classes to form the test set. In the label-conditional setting, we randomly sample $m = 15$ real images per class to form the real calibration set and 500 real images per class for testing. In both cases, the synthetic calibration set consists of $N = 1,500$ annotated ImageNet images, drawn from 100 classes that are disjoint from the real classes, with $n = 15$ images per class.

We apply the subset selection approach to improve the quality of the synthetic data, using a $k = 20$ nearest-subset selection strategy, leading to $\tilde{N} = nk = 300$ selected synthetic datapoints (see Algorithm 2). We compare this `SPI-Subset` variant of our method to `SPI-Whole`, where the latter denotes the `SPI` procedure run with the entire synthetic set. Additionally, as a baseline, we include standard conformal prediction applied to the real set, `OnlyReal`.

Figure 4 shows the performance of different methods with marginal and label-conditional guarantees at target coverage level $1 - \alpha = 0.98$. The label-conditional results are shown for five representative classes. We see that `OnlyReal` controls the coverage at the $1 - \alpha$ level as expected, but it produces overly conservative—in fact, trivial—prediction sets that contain all 30 possible labels. This is not surprising as split conformal prediction needs at least 50 datapoints to produce a nontrivial prediction set at level $\alpha = 0.02$.

Both `SPI-Whole` and `SPI-Subset` achieve coverage within the theoretical bounds, generating smaller prediction sets compared to `OnlyReal`. In the label-conditional setting, `SPI-Subset` achieves coverage that more closely aligns with the target $1 - \alpha$ and produces smaller prediction sets, outperforming `SPI-Whole`. This highlights the benefit of aligning the synthetic set more closely with the real distribution through k-nearest subset selection.

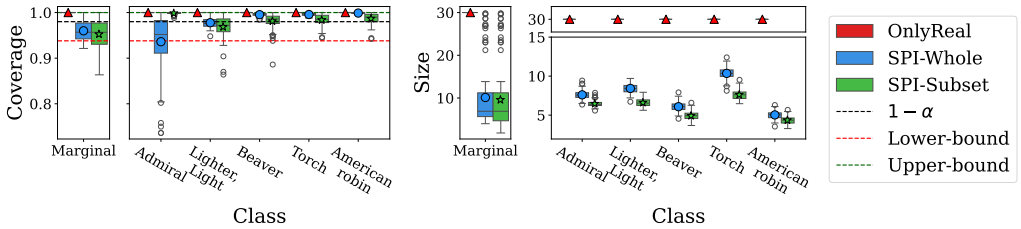


Figure 4: Results for the ImageNet data: Coverage rates of `OnlyReal`, `SPI-Whole`, and `SPI-Subset` at level $1 - \alpha = 0.98$, averaged over 100 trials. Left: Average coverage. Right: Average prediction set size, both under marginal (leftmost box in each group) and label-conditional coverage settings. Label-conditional results are shown for selected classes; see Table S8 for results across all classes.

However, in the marginal setting, where the real calibration set includes images from 30 different classes, selecting a small subset of k synthetic classes does not necessarily improve alignment with the real distribution. Consequently, `SPI-Subset`, which uses only a subset of the synthetic data (300 images), exhibits higher variance in coverage compared to `SPI-Whole`, which leverages the entire synthetic calibration set of 1,500 images.

We provide results for all classes appearing in the real data at additional values of the target level α in Appendix J.2. In Appendix J.2.1, we further illustrate the performance of the `SPI-Subset` procedure for different values of the hyperparameter k.

4.2 Regression on the MEPS dataset

In this experiment, we evaluate our method on a regression task using the Medical Expenditure Panel Survey (MEPS) datasets [3]. We first fit a regression model on MEPS panel survey number 19. MEPS panel survey number 20 is then used as the synthetic data, and panel survey number 21 serves as the real data. This setup reflects a scenario in which large historical panels are leveraged as synthetic data to improve calibration on a recent, smaller real-world population. For all methods, we use the conformalized quantile regression (CQR) score function [44]. Further details on the score function and the regression model are provided in Appendices C.1 and H.3, respectively. For each age group, we construct a real calibration set with $m = 15$ examples and a synthetic calibration set with $N = 1,000$ examples.

Figure 5 presents the coverage and interval length results for `OnlyReal`, `OnlySynth`, and `SPI` at target coverage level $1 - \alpha = 0.9$, across different age-groups. Similarly to the classification experiments, `OnlyReal` attains valid coverage but has a higher variance due to the small size of the real calibration set. Following that figure, we see that the synthetic and real data are well aligned. However, `OnlySynth`, which relies solely on synthetic data, lacks formal coverage guarantees. In contrast, `SPI` achieves coverage close to the nominal level of 0.9, as predicted by Theorem 3.3. We provide results for additional α levels and further experiments in Appendix K.

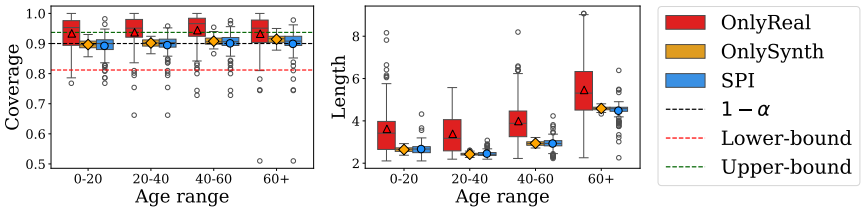


Figure 5: Results for the MEPS dataset: Marginal coverage and interval length for each age-group, obtained by `OnlyReal`, `OnlySynth`, and `SPI`. Target coverage is $1 - \alpha = 0.9$; experiments are repeated for 100 trials.

5 Discussion

In this work, we presented a novel framework that enhances the sample efficiency of conformal prediction by leveraging synthetic data in a theoretically grounded manner. While we focused on marginal and label-conditional coverage, many applications require feature-conditional guarantees. Extending our approach to such settings—e.g., by drawing on ideas from Gibbs et al. [22]—is an important direction for future work. Another limitation is the assumption that the real calibration data and the test point are i.i.d., which may not hold in practice. We believe our results can be extended beyond the i.i.d. setting by building on techniques developed in [8, 28, 42, 51, 53].

Naturally, the quality of the synthetic data affects the performance of `SPI`. While we found our data-dependent k-nearest subset approach to be effective, exploring alternative strategies—particularly those suited to settings with multiple synthetic data sources—may further enhance performance. Another future direction that we are pursuing is to generalize `SPI` to enable the broader use of synthetic data across statistical inference methods, including conformal risk control, hypothesis testing, and multiple hypothesis testing [9].

As this work aims to advance the field of uncertainty quantification in machine learning, it has potential social implications, similar to other research in the field.

Acknowledgments and Disclosure of Funding

M. B., R. M. L., and Y. R. were supported by the European Union (ERC, SafetyBounds, 101163414). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This research was also partially supported by the Israel Science Foundation (ISF grant 729/21). E. D. and Y. L. were partially supported by the US NSF, NIH, ARO, AFOSR, ONR, and the Sloan Foundation. Y. R. acknowledges additional support from the Career Advancement Fellowship at the Technion, and is deeply grateful to Shai Feldman and Jeremias Sulam for their insightful discussions and valuable feedback.

References

[1] Agency for Healthcare Research and Quality. Medical expenditure panel survey, panel 20, 2020.

[2] Agency for Healthcare Research and Quality. Medical expenditure panel survey, panel 21, 2020.

[3] Agency for Healthcare Research and Quality. Medical expenditure panel survey, panel 19, 2025.

[4] Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

[5] Anastasios N Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I Jordan, and Tijana Zrnic. Prediction-powered inference. *Science*, 2023.

[6] Konstantina Bairaktari, Jiayun Wu, and Steven Wu. Kandinsky conformal prediction: Beyond class- and covariate-conditional coverage. In *Forty-second International Conference on Machine Learning*, 2025.

[7] Christopher RS Banerji, Tapabrata Chakraborti, Chris Harbron, and Ben D MacArthur. Clinical ai tools must convey predictive uncertainty for each individual patient. *Nature medicine*, 2023.

[8] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 2023.

[9] Meshi Bashari, Yonghoon Lee, Roy Maor Lotan, Edgar Dobriban, and Yaniv Romano. Statistical inference leveraging synthetic data with distribution-free guarantees. *arXiv preprint arXiv:2509.20345*, 2025.

[10] Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *The Annals of Statistics*, 2023.

[11] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[12] Victor Chernozhukov, Kaspar Wuthrich, and Yinchu Zhu. Exact and Robust Conformal Inference Methods for Predictive Machine Learning With Dependent Data. In *Proceedings of the 31st Conference On Learning Theory, PMLR*, 2018.

[13] Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Toward personalized inference on individual treatment effects. *Proceedings of the National Academy of Sciences*, 2023.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, 2009.

[15] Tiffany Ding, Anastasios Angelopoulos, Stephen Bates, Michael Jordan, and Ryan J Tibshirani. Class-conditional conformal prediction with many classes. *Advances in neural information processing systems*, 2023.

[16] Robin Dunn, Larry Wasserman, and Aaditya Ramdas. Distribution-free prediction sets for two-layer hierarchical models. *Journal of the American Statistical Association*, 2022.

[17] Shiladitya Dutta, Hongbo Wei, Lars van der Laan, and Ahmed Alaa. Estimating uncertainty in multimodal foundation models using public internet data. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2024.

[18] Bat-Sheva Einbinder, Yaniv Romano, Matteo Sesia, and Yanfei Zhou. Training uncertainty-aware classifiers with conformalized deep learning. *Advances in Neural Information Processing Systems*, 2022.

[19] Bat-Sheva Einbinder, Liran Ringel, and Yaniv Romano. Semi-supervised risk control via prediction-powered inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[20] Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Few-shot conformal prediction with auxiliary tasks. In *International Conference on Machine Learning*. PMLR, 2021.

[21] Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 2021.

[22] Isaac Gibbs, John J Cherian, and Emmanuel J Candès. Conformal prediction with conditional guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 2025.

[23] Leying Guan. Localized conformal prediction: A generalized inference framework for conformal prediction. *Biometrika*, 2023.

[24] Leying Guan. A conformal test of linear models via permutation-augmented regressions. *The Annals of Statistics*, 2024.

[25] Leying Guan and Robert Tibshirani. Prediction and outlier detection in classification problems. *Journal of the Royal Statistical Society: Series B*, 2022.

[26] Rohan Hore and Rina Foygel Barber. Conformal prediction with local weights: randomization enables robust guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 2025.

[27] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. Version 0.1, Zenodo.

[28] Sunay Joshi, Shayan Kiyani, George Pappas, Edgar Dobriban, and Hamed Hassani. Likelihood-ratio regularized quantile regression: Adapting conformal prediction to high-dimensional covariate shifts. *arXiv preprint arXiv:2502.13030*, 2025.

[29] Christopher Jung, Georgy Noarov, Ramya Ramalingam, and Aaron Roth. Batch multivalid conformal prediction. In *International Conference on Learning Representations*, 2023.

[30] Black Forest Labs. Flux: High-fidelity text-to-image generation with transformer diffusion models, 2024.

[31] Yonghoon Lee, Eric Tchetgen Tchetgen, and Edgar Dobriban. Batch predictive inference. *arXiv preprint arXiv:2409.13990*, 2024.

[32] Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2014.

[33] Jing Lei, James Robins, and Larry Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, 2013.

[34] Jing Lei, Alessandro Rinaldo, and Larry Wasserman. A conformal prediction approach to explore functional data. *Annals of Mathematics and Artificial Intelligence*, 2015.

[35] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 2018.

[36] Ziyi Liang, Matteo Sesia, and Wenguang Sun. Integrative conformal p-values for powerful out-of-distribution testing with labeled outliers. *arXiv preprint arXiv:2208.11111*, 2022.

[37] Ziyi Liang, Yanfei Zhou, and Matteo Sesia. Conformal inference is (almost) free for neural networks trained with early stopping. In *International Conference on Machine Learning*, 2023.

[38] Keli Liu and Xiao-Li Meng. There is individualized treatment. why not individualized inference? *Annual Review of Statistics and Its Application*, 2016.

[39] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning*. Springer, 2002.

[40] Sangdon Park, Edgar Dobriban, Insup Lee, and Osbert Bastani. PAC prediction sets under covariate shift. In *International Conference on Learning Representations*, 2022.

[41] Sangwoo Park, Kfir M Cohen, and Osvaldo Simeone. Few-shot calibration of set predictors via meta-learned cross-validation-based conformal prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[42] Aleksandr Podkopaev and Aaditya Ramdas. Distribution-free uncertainty quantification for classification under label shift. In *Uncertainty in artificial intelligence*. PMLR, 2021.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PLMR, 2021.

[44] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 2019.

[45] Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 2020.

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[47] Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least Ambiguous Set-Valued Classifiers With Bounded Error Levels. *Journal of the American Statistical Association*, 2019.

[48] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Transduction with confidence and credibility. In *IJCAI*, 1999.

[49] Henry Scheffe and John W Tukey. Non-parametric estimation. i. validation of order statistics. *The Annals of Mathematical Statistics*, 1945.

[50] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 2022.

[51] Matteo Sesia, YX Rachel Wang, and Xin Tong. Adaptive conformal classification with noisy labels. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 2024.

[52] David Stutz, Abhijit Guha Roy, Tatiana Matejovicova, Patricia Strachan, Ali Taylan Cemgil, and Arnaud Doucet. Conformal prediction under ambiguous ground truth. *Transactions on Machine Learning Research*, 2024.

[53] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel J Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 2019.

[54] John W Tukey. Non-parametric estimation ii. statistically equivalent blocks and tolerance regions–the continuous case. *The Annals of Mathematical Statistics*, 1947.

[55] John W Tukey. Nonparametric estimation, iii. statistically equivalent blocks and multivariate tolerance regions–the discontinuous case. *The Annals of Mathematical Statistics*, 1948.

[56] Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*. PMLR, 2012.

[57] Vladimir Vovk, David Lindsay, Ilia Nouretdinov, and Alex Gammerman. Mondrian confidence machine. *Technical Report*, 2003.

[58] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

[59] Volodya Vovk, Alexander Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. In *International Conference on Machine Learning*, 1999.

[60] Abraham Wald. An extension of wilks' method for setting tolerance limits. *The Annals of Mathematical Statistics*, 1943.

[61] Samuel S Wilks. Determination of sample sizes for setting tolerance limits. *The Annals of Mathematical Statistics*, 1941.

[62] Samuel S Wilks. *Mathematical statistics*. Wiley, 1962.

[63] Ran Xie, Rina Barber, and Emmanuel Candes. Boosted conformal prediction intervals. *Advances in Neural Information Processing Systems*, 2024.

[64] Yao Zhang and Emmanuel J Candès. Posterior conformal prediction. *arXiv preprint arXiv:2409.19712*, 2024.

A Algorithmic details

Algorithm 1 Synthetic-powered predictive inference (`SPI`)

Input: Real calibration set $(X_i, Y_i)_{i\in[m]}$; synthetic calibration set $(\tilde{X}_i, \tilde{Y}_i)_{i\in[N]}$; test input X_{m+1}; score function s; target coverage level $1-\alpha$; parameter for window construction β.
Compute the real scores $S_i = s(X_i, Y_i)$, for $i \in [m]$.
Compute the synthetic scores $\tilde{S}_j = s(\tilde{X}_j, \tilde{Y}_j)$, for $j \in [N]$, and let $\tilde{Q}'_{1-\alpha} = \tilde{S}_{(\lceil (N+1)(1-\alpha)\rceil+1)}$.
Compute R_r^- and R_r^+ for $r \in [m+1]$, according to (6).
Compute $\tilde{R}^-$ and $\tilde{R}^+$, according to (11).
Compute the bound $Q = \max\{\min\{\tilde{Q}'_{1-\alpha}, S_{(\tilde{R}^-)}\}, S_{(\tilde{R}_+)}\}$.
Compute $\widehat{C}(X_{m+1}) = \{y \in \mathcal{Y} : s(X_{m+1}, y) \le Q\}$.
Output: Prediction set $\widehat{C}(X_{m+1})$.

Algorithm 2 `SPI` with data-dependent k-nearest subset selection

Input: Real calibration set $(X_i, Y_i)_{i\in[m]}$; subsets of synthetic calibration set $(\tilde{X}_j^l, \tilde{Y}_j^l)_{j\in[n]}$, $l = 1, 2, \cdots, L$; test input X_{m+1}; score function s; target coverage level $1-\alpha$; parameter for window construction β; parameter for selection k.
Compute the real scores $S_i = s(X_i, Y_i)$, for $i \in [m]$.
Compute the synthetic scores $\tilde{S}_j^l = s(\tilde{X}_j^l, \tilde{Y}_j^l)$, for $j \in [N]$ and $l \in [L]$.
for l in $[L]$ **do**
    Distances$[l] \leftarrow$ Cramer-von-Mises-Statistic$(\{S_i : i \in [m]\}, \{\tilde{S}_j^l : j \in [n]\})$ {Algorithm 3}
end for
Let $\mathcal{L}$ be the set of k subsets in $[L]$ with the smallest values in *Distances*.
Apply Algorithm 1 with $\{(\tilde{X}_j^l, \tilde{Y}_j^l) : j \in [n], l \in \mathcal{L}\}$ as the synthetic calibration data.
Output: Prediction set $\widehat{C}(X_{m+1})$.

Algorithm 3 Cramer-von Mises two-sample test statistic

Input: $(X_i)_{i\in[N]}$; $(Y_i)_{i\in[M]}$ (all distinct)
Let $W = \{X_1, \ldots, X_N\} \cup \{Y_1, \ldots, Y_M\}$ be the set of all datapoints.
Compute the ranks:
$r_i =$ (the rank of X_i in W) for $i \in [N]$, and $s_i =$ (the rank of Y_i in W) for $i \in [M]$.
Let $r_{(1)} < \cdots < r_{(N)}$ and $s_{(1)} < \cdots < s_{(M)}$ be the order statistics of $(r_i)_{i\in[N]}$ and $(s_i)_{i\in[M]}$, respectively.
Compute

$$U = N\sum_{i=1}^{N}(r_{(i)} - i)^2 + M\sum_{j=1}^{M}(s_{(j)} - j)^2.$$

Compute the Cramer–von Mises test statistic T as:

$$T = \frac{U}{NM(N+M)} - \frac{4MN-1}{6(M+N)}.$$

Output: T.

Algorithm 4 β-selection

Input: Real calibration set size m; synthetic calibration set size N; target coverage level $1-\alpha$; desired worst-case lower bound L; step size ϵ.
Set $\beta \leftarrow \epsilon$.
Compute R_r^+ with β for $r \in [m+1]$, according to (6).
Compute $\tilde{L} \leftarrow |\{i \in [m+1] : R_i^+ \leq \lceil (1-\alpha)(N+1) \rceil\}| \,/\, (m+1)$.
while $\tilde{L} < L$ **do**
    $\beta += \epsilon$
    Compute R_r^+ with β for $r \in [m+1]$, according to (6).
    Compute $\tilde{L} \leftarrow |\{i \in [m+1] : R_i^+ \leq \lceil (1-\alpha)(N+1) \rceil\}| \,/\, (m+1)$.
end while
Output: β.

B Related work

The concept of prediction sets dates back to foundational works such as Wilks [61], Wald [60], Scheffe and Tukey [49], and Tukey [54, 55]. The initial ideas behind conformal prediction were introduced by Saunders et al. [48] and Vovk et al. [59]. Since then, with the rise of machine learning, conformal prediction has emerged as a widely used framework for constructing distribution-free prediction sets [e.g., 4, 10, 12, 16, 18, 21, 23–25, 32–37, 39, 40, 45, 47, 56, 58].

More recently, there has been growing interest in extending conformal prediction to offer more refined guarantees beyond standard marginal coverage. In particular, several works aim to offer approximate local coverage guarantees in the feature space [23, 26, 64]; group-conditional coverage, which aims to guarantee valid coverage across pre-defined groups based on features and/or labels [6, 22, 29, 57]; and cluster-conditional coverage, which focuses on label-conditioned subgroups [15]. However, these approaches still face the inherent limitations of conformal inference in settings where labeled data for the group-of-interest is limited, as previously discussed.

In contrast, we are interested in obtaining exact label- or group-of-interest conditional coverage guarantees even when the dataset from our distribution of interest is small. To this end, we take a different approach, aiming to enhance sample efficiency by incorporating synthetic data.

A related line of work explores the use of unlabeled data to improve sample efficiency [5, 19]. These methods assume that the unlabeled data is drawn from the same distribution as the labeled calibration set. In contrast, we consider settings where this assumption is violated and develop methods that remain valid under such unknown distributional shifts. Moreover, the above methods cannot be applied in the label-conditional setting, as they require knowing the labels of the unlabeled data.

Another related line of work is few-shot conformal prediction [20, 41], which addresses settings where only limited data is available for the target task, along with additional auxiliary tasks. These approaches leverage related but distinct tasks to improve sample efficiency. Fisch et al. [20] provides asymptotic task-conditional coverage guarantees, whereas our focus is on finite-sample guarantees. Park et al. [41] mitigate the small-sample challenge using cross-validation, but their methods remain constrained by the overall number of available datapoints—which we assume to be small in our setting. Dutta et al. [17] propose to retrieve web images to enable conformal prediction in zero-shot settings, by leveraging conformal prediction with ambiguous ground truth [52], but do not provide coverage guarantees for their method.

C Technical background

C.1 Score functions

Adaptive prediction sets (APS) [45] For classification tasks, we assume that the pre-trained model outputs an estimated probability vector $\hat{\pi} \in [0,1]^K$, where K is the number of classes and each entry represents the estimated probability of the corresponding class. We consider the APS score function that is defined for a given pair (X, Y) as follows: Let $\hat{\pi}_{(1)}(X) \geq \hat{\pi}_{(2)}(X) \geq \cdots \geq \hat{\pi}_{(K)}(X)$ be the sorted values of the probability vector $\hat{\pi}(X)$, and let $r(Y, \hat{\pi}(X))$ denote the rank of the label Y

within this sorted vector. The nonconformity score is then given by:

$$s(X,Y) = \hat{\pi}_{(1)}(X) + \hat{\pi}_{(2)}(X) + \cdots + \hat{\pi}_{(r(Y,\hat{\pi}(X)))}(X) - U \cdot \hat{\pi}_{(r(Y,\hat{\pi}(X)))}(X), \tag{12}$$

where U is a uniform random variable on $[0,1]$, independent of everything else.

Homogeneous prediction sets (HPS) [58] As a complementary score function for classification tasks, we consider the HPS score. Using the same notations as in the APS paragraph, for a given pair (X,Y), the HPS nonconformity score is defined as

$$s(X,Y) = 1 - \hat{\pi}_{(r(Y,\hat{\pi}(X)))}(X). \tag{13}$$

Conformalized quantile regression [44] For the regression task, suppose we have a pre-trained quantile regression model that estimates the γ-th quantile of the distribution $Y \mid X$, denoted as $\hat{q}(X;\gamma)$. The conformalized quantile regression (CQR) score is then defined as

$$s(X,Y) = \max\{\hat{q}(X;\alpha/2) - Y, Y - \hat{q}(X;1-\alpha/2)\}. \tag{14}$$

Applying conformal prediction with this score, the prediction set takes the form

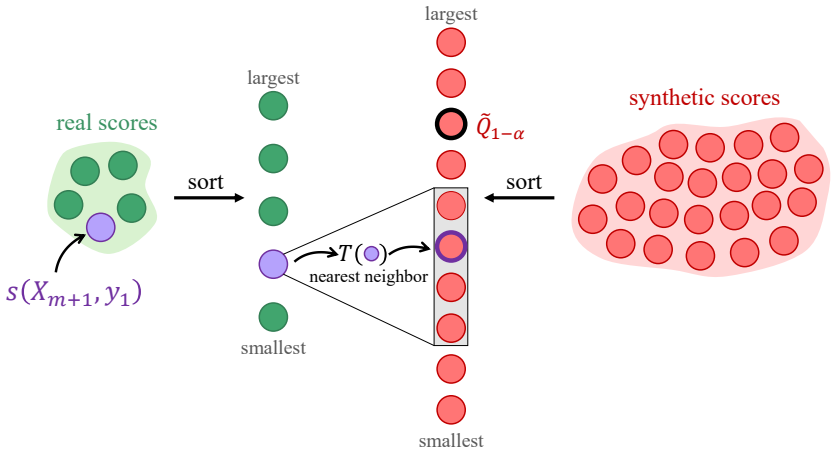
$$\widehat{C}(X_{n+1}) = \left[\hat{q}(X_{n+1};\alpha/2) - \hat{Q}_{1-\alpha}, \hat{q}(X_{n+1};1-\alpha/2) + \hat{Q}_{1-\alpha}\right].$$

D Explaining the *score transporter*

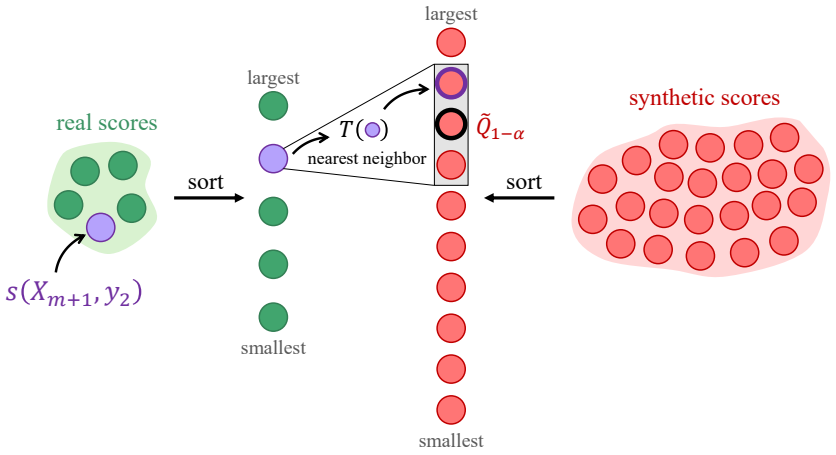
In this section, we provide further intuition about our proposed procedure, as well as the theoretical bounds established in Theorem 3.5.

Figure S1 provides a schematic overview of the procedure. For clarity, we illustrate the construction of the prediction set as described in Equation (9). In practice, we use the computationally efficient procedure described in Equation (10), which we have shown to be equivalent in Proposition 3.2.

We begin by computing nonconformity scores for both real and synthetic data points. We assume access to a fixed score function s (e.g., a pre-trained black-box model), along with a small real calibration set $(X_i, Y_i)_{i\in[m]}$, a large synthetic calibration set $(\tilde{X}_j, \tilde{Y}_j)_{j\in[N]}$, and a test point X_{m+1}.



(a) Candidate label y_1: the test score (purple) ranks second among the real scores. Its mapped synthetic neighbor—computed via (8) and outlined in purple—falls below the empirical quantile $\tilde{Q}_{1-\alpha}$, hence $y_1 \in \widehat{C}(X_{m+1})$.



(b) Candidate label y_2: the test score (purple) ranks fourth among the real scores. Its mapped synthetic neighbor—computed via (8) and outlined in purple—exceeds the empirical quantile $\tilde{Q}_{1-\alpha}$, thus $y_2 \notin \widehat{C}(X_{m+1})$.

Figure S1: **Illustration of the synthetic-powered predictive inference for two candidate labels.** Each panel displays sorted nonconformity scores: real scores on the left and synthetic scores on the right. The rectangle indicates the window in the synthetic space to which the test score can be mapped (as defined in (7)). The black-outlined circle indicates the $(1-\alpha)(1+\frac{1}{N})$th empirical quantile of the synthetic scores, $\tilde{Q}_{1-\alpha}$.

For each candidate label $y \in \mathcal{Y}$, we compute the nonconformity score of the test pair (X_{m+1}, y), along with those of the real and synthetic calibration sets. These scores are depicted as circles in

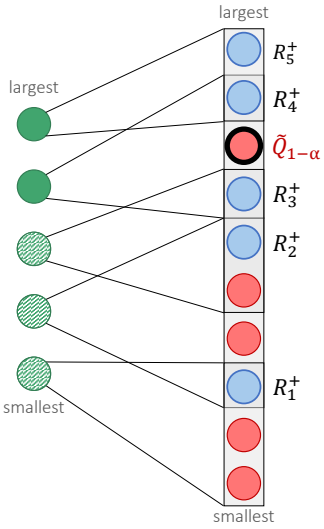
Figure S1, displayed in sorted order. Crucially, the scores for the real and synthetic calibration sets are computed once and then reused for all candidate labels.

Next, following Equation (7), each real score $S_{(i)}$, $i \in [m+1]$, is associated with a window in the synthetic score space. These windows (illustrated as rectangles connected to each real score) define the set of synthetic scores to which each real score can potentially be mapped. Importantly, these windows depend only on the sample sizes m, N, and the parameter β—not on the values of the scores themselves.

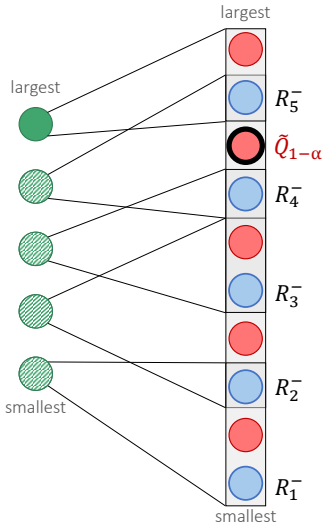
For the test score $s(X_{m+1}, y)$, we identify its synthetic neighbor after mapping, within its associated window according to (8), denoted by $T(s(X_{m+1}, y))$. This synthetic score is then compared to the $(1-\alpha)(1+\frac{1}{N})$th empirical quantile of the synthetic scores, $\tilde{Q}_{1-\alpha}$ (outlined in black). If $T(s(X_{m+1}, y)) \leq \tilde{Q}_{1-\alpha}$, the candidate label y is included in the prediction set $\widehat{C}(X_{m+1})$; otherwise, it is excluded.

To illustrate this, Figure S1a shows the procedure for candidate label $y_1 \in \mathcal{Y}$. The corresponding test score (marked in purple) ranks second among the real scores, and its mapped synthetic neighbor within the corresponding window (outlined in purple) lies below the quantile threshold: $T(s(X_{m+1}, y_1)) \leq \tilde{Q}_{1-\alpha}$. Thus, $y_1 \in \widehat{C}(X_{m+1})$. In contrast, Figure S1b depicts the case for label $y_2 \in \mathcal{Y}$ (also marked in purple), where the test score ranks fourth, and its mapped neighbor exceeds the threshold: $T(s(X_{m+1}, y_2)) > \tilde{Q}_{1-\alpha}$. Therefore, $y_2 \notin \widehat{C}(X_{m+1})$. This procedure is repeated for each $y \in \mathcal{Y}$ to construct the full prediction set $\widehat{C}(X_{m+1})$.

We now illustrate the theoretical bounds established in Theorem 3.5, using the same schematic from Figure S1. Real and synthetic nonconformity scores are shown as circles in sorted order, with each real score connected to a window in the synthetic score space (depicted as rectangles). The $(1-\alpha)(1+\frac{1}{N})$th empirical quantile of the synthetic scores, $\tilde{Q}_{1-\alpha}$, is outlined in black.



(a) Values of R_r^+ (in blue), representing the upper endpoints of the synthetic windows. The smallest three real scores satisfy $R_r^+ \leq \tilde{Q}_{1-\alpha}$ and are thus guaranteed to be mapped to synthetic scores below the threshold $\tilde{Q}_{1-\alpha}$.



(b) Values of R_r^- (in blue), representing the lower endpoints of the synthetic windows. The fifth real score satisfies $R_5^- > \tilde{Q}_{1-\alpha}$, meaning it is necessarily mapped above the threshold $\tilde{Q}_{1-\alpha}$.

Figure S2: **Illustration of the quantities used in the worst-case coverage bounds from Theorem 3.5**. Real and synthetic nonconformity scores are shown as circles in sorted order. Each real score is connected to a window in the synthetic score space (depicted as rectangles). The synthetic $(1-\alpha)(1+\frac{1}{N})$th empirical quantile $\tilde{Q}_{1-\alpha}$ is outlined in black.

Figure S2 visualizes the quantities used to derive the worst-case coverage bounds. For each real score $S_{(r)}$ for $r \in [m+1]$, we denote the endpoints of its associated window by R_r^- and R_r^+, as introduced

in (6). These correspond to the smallest and largest ranks, respectively, of the synthetic scores that $S_{(r)}$ can be mapped to. For convenience, we refer to R_r^- and R_r^+ as the synthetic scores at those ranks.

Figure S2a shows the values R_r^+ (in blue), which are used to compute the lower bound. Since the transported score $T(S_{(r)})$—defined as the nearest synthetic score within the window among those that are smaller than $S_{(r)}$—is always less than or equal to R_r^+, any real score satisfying $R_r^+ \leq \tilde{Q}_{1-\alpha}$ must necessarily satisfy $T(S_{(r)}) \leq \tilde{Q}_{1-\alpha}$. Consequently, the corresponding label will always be included in the prediction set.

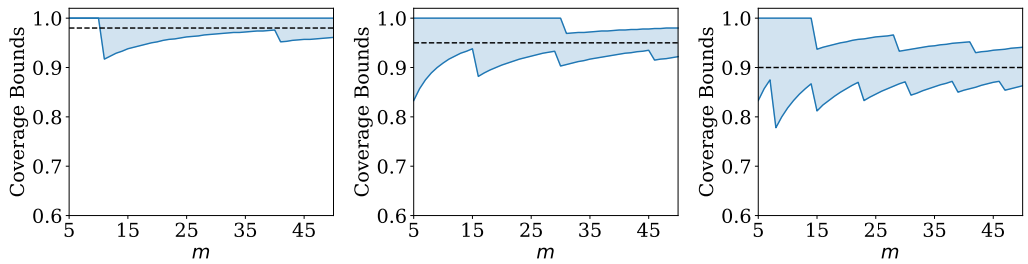
Figure S2b shows the values R_r^- (in blue), which are used to compute the upper bound. Real scores for which $R_r^- \leq \tilde{Q}_{1-\alpha}$ may be mapped to a synthetic score below the threshold and thus may be included in the prediction set—for example, the bottom four real scores in the figure. In contrast, if $R_r^- > \tilde{Q}_{1-\alpha}$ (as for the fifth score), then the transported score must exceed the threshold, and the corresponding label is guaranteed to be excluded.

By exchangeability, the test score is equally likely to take any of the $m+1$ possible ranks among the real calibration scores. Therefore, the coverage probability is bounded between the fraction of real scores whose $R_r^+ \leq \tilde{Q}_{1-\alpha}$ and the fraction whose $R_r^- \leq \tilde{Q}_{1-\alpha}$, as formalized in Theorem 3.5. In our example, these correspond to $3/5$ and $4/5$, respectively.

D.1 Coverage guarantee bounds

To illustrate the distribution-free bounds in Theorem 3.5, we present several visualizations. These bounds are determined solely by the sample sizes m and N, the parameter β, and the target coverage level $1-\alpha$.

Figure S3 presents the upper and lower bounds in Theorem 3.5 as functions of the calibration set size m and the level α, with fixed $N = 1000$ and $\beta = 0.4$. As m increases, the bounds become tighter around the target level $1-\alpha$.

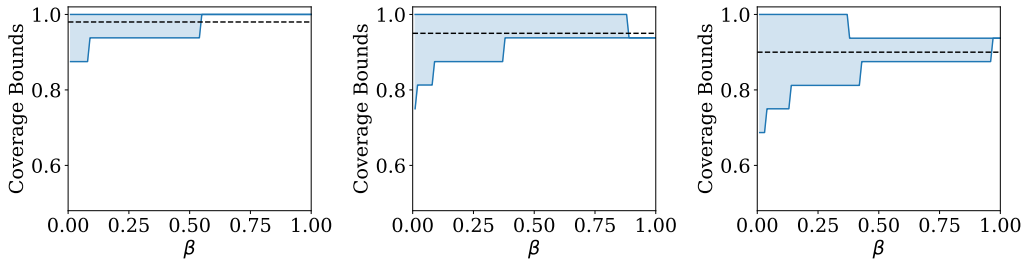


(a) $\alpha = 0.02$ (b) $\alpha = 0.05$ (c) $\alpha = 0.1$

Figure S3: Illustration of the coverage bounds in Theorem 3.5 as a function of the real calibration set size m. The synthetic calibration size is $N = 1000$, and we set $\beta = 0.4$. Results are presented for $\alpha = 0.02$ (a), 0.05 (b), and 0.1 (c). The shaded regions represent the area between the lower and upper bounds for each α level, with the dashed black lines indicating the target coverage level $1-\alpha$.

Next, Figure S4 illustrates how the bounds vary with the parameter β—under $m = 15$, $N = 1000$, and different values of α. As β decreases, the bounds become looser. This trend can be explained as follows: by the construction of the windows in (6), smaller values of β lead to wider windows. As a result, fewer R_r^+ values (for $r \in [m+1]$) fall below the $(1-\alpha)$th empirical quantile, loosening the lower bound. At the same time, more R_r^- values fall below this quantile, resulting in a looser upper bound. This trend is consistent across various α, as shown in the figure. Further, the bounds exhibit a stepwise pattern due to their discrete nature—their values change in increments of $1/(m+1)$.

In practice, one may be interested in using our method while ensuring that the lower bound guaranteed by Theorem 3.5 is no smaller than a user-specified level L. To this end, we provide an algorithm (see Algorithm 4) for selecting β based on the sample sizes m and N, the target miscoverage level α, step size ϵ (e.g., 0.01), and the desired lower bound L. As shown in Figure S4, multiple β values may



(a) $\alpha = 0.02$ (b) $\alpha = 0.05$ (c) $\alpha = 0.1$

Figure S4: Illustration of the coverage guarantee bounds from Theorem 3.5 as a function of β. The real calibration set contains $m = 15$ datapoints. Other details are as in Figure S3.

yield the same lower bound. In such cases, the algorithm selects the smallest β that results in a lower bound greater than or equal to L, inspired by the result in Theorem 3.3.

E Constructing a separate synthetic score function with data splitting

In this section, we provide details on constructing the synthetic score function independently of the calibration data, adding to the discussion in Section 3.4. For instance, suppose we apply data splitting (to both the real and the synthetic data) and use one split as training data to construct the synthetic score function $\tilde{s}$. If the data has already been split to construct s, the same split can be used for constructing $\tilde{s}$. Then we use both real and synthetic (training) data to construct $\tilde{s}$, to ensure that its distribution better approximates that of the real score. Throughout this section, we condition everything on the training datasets.

We begin with the method of constructing an adjustment function g and using the transformed score function $\tilde{s} = g \circ s$ for the synthetic data. In this case, the prediction set $\widehat{C}^g(X_{m+1})$, constructed according to (9), is given by

$$\widehat{C}^g(X_{m+1}) = \{y \in \mathcal{Y} : T(s(X_{m+1}, y); (S_i)_{i\in[m]}, (g(\tilde{S}_j))_{j\in N}) \leq \tilde{Q}^g_{1-\alpha}\},$$

where $\tilde{Q}^g_{1-\alpha}$ denotes the $\lceil (N+1)(1-\alpha)\rceil$-th smallest value among $\{g(\tilde{S}_j) : j \in [N]\}$.

Example 1. One option is to construct an affine transformation $g(s) = \theta_1 s + \theta_2$ to adjust the scale and bias of $s(\tilde{X}, \tilde{Y})$. Denote the training sets by $(X'_1, Y'_1), \ldots, (X'_{m_{\text{train}}}, Y'_{m_{\text{train}}})$ and $(\tilde{X}'_1, \tilde{Y}'_1), \ldots, (\tilde{X}'_{N_{\text{train}}}, \tilde{Y}'_{N_{\text{train}}})$, and denote the corresponding real and synthetic (training) scores by $(S'_i)_{i\in[m_{\text{train}}]}$ and $(\tilde{S}'_j)_{j\in[N_{\text{train}}]}$, respectively. Then, we can set θ_1 and θ_2 via least squares:

$$(\theta_1, \theta_2) = \operatorname*{argmin}_{a,b} \sum_{i=1}^{m_{\text{train}}} \left| a \cdot \tilde{S}'_{(\lfloor iN_{\text{train}}/m_{\text{train}}\rfloor)} + b - S'_{(i)} \right|^2,$$

where $S'_{(i)}$ and $\tilde{S}'_{(j)}$ denote the order statistics of the real and synthetic training scores, respectively.

More generally, suppose we construct a new score function $\tilde{s}$ using the training data, such that the distribution of $\tilde{s}(\tilde{X}, \tilde{Y})$ approximates that of $s(X, Y)$, where $(\tilde{X}, \tilde{Y}) \sim Q_{X,Y}$ and $(X, Y) \sim P$. The prediction set is then constructed according to (9), using the synthetic scores $\tilde{S}_j = \tilde{s}(\tilde{X}_j, \tilde{Y}_j)$:

$$\widehat{C}(X_{m+1}) = \{y \in \mathcal{Y} : T(s(X_{m+1}, y); (S_i)_{i\in[m]}, (\tilde{S}_j)_{j\in N}) \leq \tilde{Q}_{1-\alpha}\}. \tag{15}$$

Denoting the distribution of $\tilde{s}(\tilde{X}, \tilde{Y})$ as $\tilde{Q}$, we have the following result as a direct consequence of Theorem 3.3 and 3.5[6].

Corollary E.1. *Suppose the distribution $\tilde{Q}$ is continuous. Then the prediction set $\widehat{C}(X_{m+1})$ from* (15)*, constructed using $\tilde{S}_j = \tilde{s}(\tilde{X}_j, \tilde{Y}_j)$ for $j \in [N]$, satisfies*

$$1 - \alpha - \beta - \varepsilon^{m+1}_{P,\tilde{Q}} \leq \mathbb{P}\left\{Y_{m+1} \in \widehat{C}(X_{m+1})\right\} \leq 1 - \alpha + \beta + \varepsilon^{m+1}_{P,\tilde{Q}} + 1/(N+1).$$

Moreover, the bounds stated in Theorem 3.5 also hold for $\widehat{C}(X_{m+1})$ from (15)*.*

F Predictive inference with label-conditional coverage control

Here, we review the standard approach [58] for achieving the label-conditional coverage guarantee (3), and then discuss a variant of this approach based on `SPI`.

The basic idea is to partition the calibration set by classes, run conformal prediction within each class, and then combine the results to construct a prediction set. Specifically, the prediction set is constructed as follows:

$$\widehat{C}(X_{m+1}) = \left\{y \in \mathcal{Y} : s(X_{m+1}, y) \leq Q^y_{1-\alpha}\right\},$$

where $Q^y_{1-\alpha}$ denotes the $\lceil (1-\alpha)(n_y+1)\rceil$-th smallest element among $\{S_i : i \in [m], Y_i = y\}$, and n_y denotes the number of calibration points labeled with y.

[6]The proofs of these theorems build upon the setting $S_1, \ldots, S_{m+1} \overset{\text{iid}}{\sim} P$ and $\tilde{S}_1, \ldots, \tilde{S}_N \overset{\text{iid}}{\sim} Q$, and do not depend directly on the datasets or the score function. Therefore, the results in Corollary E.1 follow directly by applying the same arguments with $\tilde{S}_1, \ldots, \tilde{S}_N \overset{\text{iid}}{\sim} \tilde{Q}$.

Now, we introduce the SPI-based method that ensures the label-conditional coverage guarantee. The idea follows the same logic as the standard method: We run SPI within each class-specific partition of the real and synthetic calibration sets and then combine the results. For a class that does not appear in the synthetic dataset, we run the procedure with the entire synthetic dataset.

To formalize this idea, define

$$\mathcal{Y}_1 = \{y \in \mathcal{Y} : \tilde{Y}_j = y \text{ for some } j \in [N]\} \text{ and } \mathcal{Y}_0 = \{y \in \mathcal{Y} : \tilde{Y}_j \neq y \text{ for all } j \in [N]\}.$$

Let $I_y = \{i \in [m] : Y_i = y\}$ for each $y \in \mathcal{Y}$, and $J_y = \{j \in [N] : \tilde{Y}_j = y\}$ for each $y \in \mathcal{Y}_1$. Then for each $y \in \mathcal{Y}_1$, we define the function $T^y(\cdot) = T(\cdot; (S_i)_{i\in I_y}, (\tilde{S}_j)_{j\in J_y})$, following the definition in (8). For $y \in \mathcal{Y}_0$, we let $T^y(\cdot) = T(\cdot; (S_i)_{i\in I_y}, (\tilde{S}_j)_{j\in[N]})$. Then we define $\tilde{Q}^y_{1-\alpha}$ for each y as follows:

$$\tilde{Q}^y_{1-\alpha} := \begin{cases} \lceil (1-\alpha)(|J_y|+1)\rceil\text{-th smallest element in } \{\tilde{S}_j : j \in J_y\}, & \text{if } y \in \mathcal{Y}_1, \\ \lceil (1-\alpha)(N+1)\rceil\text{-th smallest element in } \{\tilde{S}_j : j \in [N]\}, & \text{if } y \in \mathcal{Y}_0. \end{cases}$$

Then we construct the prediction set as

$$\widehat{C}(X_{m+1}) = \left\{ y \in \mathcal{Y} : T^y(s(X_{m+1}, y)) \leq \tilde{Q}^y_{1-\alpha} \right\}. \tag{16}$$

As a direct consequence of Theorem 3.5, the prediction set (16) attains the following label-conditional coverage control:

$$\begin{aligned} \frac{|\{j \in [m+1] : R^+_j \leq \lceil (1-\alpha)(N_y+1)\rceil\}|}{m+1} &\leq \mathbb{P}\left\{ Y_{m+1} \in \widehat{C}(X_{m+1}) \,\middle|\, Y_{m+1} = y \right\} \\ &\leq \frac{|\{j \in [m+1] : R^-_j \leq \lceil (1-\alpha)(N_y+1)\rceil\}|}{m+1}, \qquad \text{for all } y \in \mathcal{Y}, \end{aligned}$$

where we let $N_y = |J_y|$ for $y \in \mathcal{Y}_1$ and $N_y = N$ for $y \in \mathcal{Y}_0$.

G Mathematical proofs

G.1 Proof of Lemma 3.1

The result follows directly from the work of Lee et al. [31], but we provide the proof for completeness. Define

$$R_r = \min\{\tau \in [N+1] : \tilde{S}_{(\tau)} \geq S_{(r)}\}$$

for each $r \in [m+1]$, where we let $R_r = N+1$ if $S_{(r)} \geq \tilde{S}_{(N)}$. Note that R_r is random, whereas R^-_r and R^+_r are not. Then by the exchangeability of $(S_r)_{r\in[m+1]}$ and $(\tilde{S}_j)_{j\in[N]}$, the distribution of the vector $(R_1, R_2, \cdots, R_{m+1})$ is given by

$$(R_1, R_2, \cdots, R_{m+1}) \sim \text{Unif}\big(\{(\zeta_1, \cdots, \zeta_{m+1}) : 1 \leq \zeta_1 \leq \cdots \leq \zeta_{m+1} \leq N+1\}\big).$$

Therefore, for each $k \in [N+1]$, we have

$$\begin{aligned} \mathbb{P}\{R_r = k\} &= \frac{|\{(\zeta_1, \cdots, \zeta_{m+1}) : 1 \leq \zeta_1 \leq \cdots \leq \zeta_{m+1} \leq N+1 \text{ and } \zeta_r = k\}|}{|\{(\zeta_1, \cdots, \zeta_{m+1}) : 1 \leq \zeta_1 \leq \cdots \leq \zeta_{m+1} \leq N+1\}|} \\ &= \frac{|\{\zeta_{1:(r-1)} : 1 \leq \zeta_1 \leq \cdots \leq \zeta_{r-1} \leq k\}| \cdot |\{\zeta_{(r+1):(m+1)} : k \leq \zeta_{r+1} \leq \cdots \leq \zeta_{m+1} \leq N+1\}|}{|\{(\zeta_1, \cdots, \zeta_{m+1}) : 1 \leq \zeta_1 \leq \cdots \leq \zeta_{m+1} \leq N+1\}|} \\ &= \frac{{}_k\mathrm{H}_{r-1} \cdot {}_{N-k+2}\mathrm{H}_{m-r+1}}{{}_{N+1}\mathrm{H}_{m+1}} = \frac{\binom{k+r-2}{r-1} \cdot \binom{N+m-k-r+2}{m-r+1}}{\binom{N+m+1}{m+1}}, \end{aligned}$$

where we use the notation ${}_n\mathrm{H}_r$ to denote the number of ways to select r items with replacement from n items. Therefore,

$$\begin{aligned} \mathbb{P}\left\{S_{(r)} \in I_m(r)\right\} = \mathbb{P}\left\{\tilde{S}_{(R^-_r)} \leq S_{(r)} \leq \tilde{S}_{(R^+_r)}\right\} &= \mathbb{P}\left\{\tilde{S}_{(R^-_r)} \leq \tilde{S}_{(R_r)} \leq \tilde{S}_{(R^+_r)}\right\} \\ &= \mathbb{P}\left\{R^-_r \leq R_r \leq R^+_r\right\} = F(R^+_r) - F(R^-_r - 1) \geq 1 - \beta, \end{aligned}$$

where the inequality follows from the definition of R^-_r and R^+_r.

G.2 Proof of Theorem 3.3

Let $r_{m+1} = \sum_{i=1}^{m} \mathbb{1}\{S_i < S_{m+1}\} + 1$ denote the rank of S_{m+1} in the increasing order among $S_1, \ldots, S_m, S_{m+1}$. Observe that $T(S_{m+1}) \leq S_{m+1}$ holds if $L_m(r_{m+1}) \leq S_{m+1}$, by the construction of the mapping T. Therefore, writing $L_r = L_m(r)$ and $U_r = U_m(r)$ for simplicity, we have

$$\begin{aligned}
&\mathbb{P}\left\{Y_{m+1} \in \widehat{C}(X_{m+1})\right\} = \mathbb{P}\left\{T(S_{m+1}) \leq \tilde{Q}_{1-\alpha}\right\} \\
&= \mathbb{P}\left\{T(S_{m+1}) \leq \tilde{Q}_{1-\alpha}, S_{m+1} \in [L_{r_{m+1}}, U_{r_{m+1}}]\right\} \\
&\qquad\qquad + \mathbb{P}\left\{T(S_{m+1}) \leq \tilde{Q}_{1-\alpha}, S_{m+1} \notin [L_{r_{m+1}}, U_{r_{m+1}}]\right\} \\
&\geq \mathbb{P}\left\{S_{m+1} \leq \tilde{Q}_{1-\alpha}, S_{m+1} \in [L_{r_{m+1}}, U_{r_{m+1}}]\right\}.
\end{aligned}$$

We can condition on r_{m+1} to write that this equals

$$\mathbb{E}\left[\mathbb{P}\left\{S_{m+1} \leq \tilde{Q}_{1-\alpha}, S_{m+1} \in [L_{r_{m+1}}, U_{r_{m+1}}] \;\middle|\; r_{m+1}\right\}\right].$$

Further, since $r_{m+1} \sim \mathrm{Unif}(1, 2, \ldots, m+1)$ by the exchangeability of $(S_i)_{i\in[m+1]}$, and since r_{m+1} is independent of the order statistics $S_{(1)}, \cdots, S_{(m+1)}$, the expression further simplifies to

$$\begin{aligned}
&\frac{1}{m+1}\sum_{r=1}^{m+1} \mathbb{P}\left\{S_{(r)} \leq \tilde{Q}_{1-\alpha}, L_r \leq S_{(r)} \leq U_r \;\middle|\; r_{m+1} = r\right\} \\
&\qquad\qquad = \frac{1}{m+1}\sum_{r=1}^{m+1} \mathbb{P}\left\{S_{(r)} \leq \tilde{Q}_{1-\alpha}, L_r \leq S_{(r)} \leq U_r\right\}.
\end{aligned}$$

Now we fix $r \in [m+1]$ and examine the probability in the summation. The event inside the probability is a function of $S_{(r)} \sim P_{(r)}^{m+1}$ and $\tilde{S}_1, \cdots, \tilde{S}_N \overset{\text{iid}}{\sim} Q$. Thus, we have

$$\begin{aligned}
&\mathbb{P}_{S_{(r)}\sim P_{(r)}^{m+1}, \tilde{S}_{1:N}\sim Q^N}\left\{S_{(r)} \leq \tilde{Q}_{1-\alpha}, L_r \leq S_{(r)} \leq U_r\right\} \\
&\geq \mathbb{P}_{S_{(r)}\sim Q_{(r)}^{m+1}, \tilde{S}_{1:N}\sim Q^N}\left\{S_{(r)} \leq \tilde{Q}_{1-\alpha}, L_r \leq S_{(r)} \leq U_r\right\} \\
&\qquad\qquad - \mathrm{d_{TV}}(P_{(r)}^{m+1} \times Q^N, Q_{(r)}^{m+1} \times Q^N) \\
&= \mathbb{P}_{S_{(r)}\sim Q_{(r)}^{m+1}, \tilde{S}_{1:N}\sim Q^N}\left\{S_{(r)} \leq \tilde{Q}_{1-\alpha}, L_r \leq S_{(r)} \leq U_r\right\} - \mathrm{d_{TV}}(P_{(r)}^{m+1}, Q_{(r)}^{m+1}).
\end{aligned}$$

Therefore, putting everything together, we have

$$\begin{aligned}
&\mathbb{P}\left\{Y_{m+1} \in \widehat{C}(X_{m+1})\right\} \\
&\geq \frac{1}{m+1}\sum_{r=1}^{m+1} \mathbb{P}_{S_{(r)}\sim Q_{(r)}^{m+1}, \tilde{S}_{1:N}\sim Q^N}\left\{S_{(r)} \leq \tilde{Q}_{1-\alpha}, L_r \leq S_{(r)} \leq U_r\right\} \\
&\qquad\qquad - \frac{1}{m+1}\sum_{r=1}^{m+1} \mathrm{d_{TV}}(P_{(r)}^{m+1}, Q_{(r)}^{m+1}) \\
&= \mathbb{P}_{S_{1:(m+1)}\sim Q^{m+1}, \tilde{S}_{1:N}\sim Q^N}\left\{S_{m+1} \leq \tilde{Q}_{1-\alpha}, S_{m+1} \in [L_{r_{m+1}}, U_{r_{m+1}}]\right\} - \varepsilon_{P,Q}^{m+1}.
\end{aligned}$$

The probability in the last term is equivalently taken with respect to $S_1, \cdots, S_{m+1}, \tilde{S}_1, \cdots, \tilde{S}_N \overset{\text{iid}}{\sim} Q$, and thus we have

$$\mathbb{P}_{S_{1:(m+1)}\sim Q^{m+1}, \tilde{S}_{1:N}\sim Q^N}\left\{S_{m+1} \leq \tilde{Q}_{1-\alpha}\right\} \geq 1 - \alpha,$$

by the standard conformal prediction coverage guarantee (5), and

$$\mathbb{P}_{S_{1:(m+1)}\sim Q^{m+1}, \tilde{S}_{1:N}\sim Q^N}\left\{S_{m+1} \in [L_{r_{m+1}}, U_{r_{m+1}}]\right\} = \frac{1}{m+1}\sum_{r=1}^{m+1} \mathbb{P}\left\{S_{(r)} \in [L_r, U_r]\right\} \geq 1 - \beta,$$

by Lemma 3.1. Therefore, by the union bound, we have

$$\mathbb{P}\left\{Y_{m+1} \in \widehat{C}(X_{m+1})\right\} \geq 1 - \alpha - \beta - \varepsilon_{P,Q}^{m+1}.$$

Next, defining $\tilde{Q}'_{1-\alpha}$ as in Section 3.2, the events $S_{m+1} \in [L_{r_{m+1}}, U_{r_{m+1}}]$ and $S_{m+1} < \tilde{Q}'_{1-\alpha}$ together imply $T(S_{m+1}) \leq \tilde{Q}_{1-\alpha}$, by the construction of T, and thus we have

$$\mathbb{P}\left\{T(S_{m+1}) \leq \tilde{Q}_{1-\alpha}\right\} \leq \mathbb{P}\left\{S_{m+1} < \tilde{Q}'_{1-\alpha} \text{ or } S_{m+1} \notin [L_{r_{m+1}}, U_{r_{m+1}}]\right\}.$$

Therefore, applying arguments analogous to the ones above, we have

$$\begin{aligned}
&\mathbb{P}\left\{Y_{m+1} \in \widehat{C}(X_{m+1})\right\} \\
&\leq \mathbb{P}_{S_{(r)} \sim Q_{(r)}^{m+1}, \tilde{S}_{1:N} \sim Q^N}\left\{S_{m+1} < \tilde{Q}'_{1-\alpha} \text{ or } S_{m+1} \notin [L_{r_{m+1}}, U_{r_{m+1}}]\right\} + \varepsilon_{P,Q}^{m+1} \\
&\leq 1 - \alpha + \beta + \varepsilon_{P,Q}^{m+1} + \frac{1}{N+1},
\end{aligned}$$

since $\mathbb{P}\left\{S_{m+1} < \tilde{Q}'_{1-\alpha}\right\} \leq 1 - \alpha + \frac{1}{N+1}$ under exchangeability due to the standard conformal prediction coverage guarantee (5).

G.3 Proof of Theorem 3.5

Let us define r_{m+1} as in the proof of Theorem 3.3. By the continuity assumption on Q, the synthetic scores are almost surely all distinct, and their order is well-defined. Now observe the deterministic relation

$$\begin{aligned}
\{R^+_{r_{m+1}} \leq \lceil (1-\alpha)(N+1) \rceil\} = \{U_m(r_{m+1}) \leq \tilde{Q}_{1-\alpha}\} \subset \{Y_{m+1} \in \widehat{C}(X_{m+1})\} \\
\subset \{L_m(r_{m+1}) \leq \tilde{Q}_{1-\alpha}\} = \{R^-_{r_{m+1}} \leq \lceil (1-\alpha)(N+1) \rceil\},
\end{aligned} \tag{17}$$

which holds by the construction of $\widehat{C}(X_{m+1})$ and the definition of the interval $[L_m(r_{m+1}), U_m(r_{m+1})]$. Therefore, the desired inequalities directly follow from the fact that $r_{m+1} \sim \text{Unif}([m+1])$ due to the exchangeability of the scores $(S_i)_{i \in [m+1]}$.

G.4 Proof of Proposition 3.2

We show that for any $x \in \mathcal{X}$, the following relation holds:

$$\widehat{C}(x) \triangle \widehat{C}^{\text{fast}}(x) \subset \{y \in \mathcal{Y} : s(x,y) \in \{\tilde{S}_j : j \in [N]\}\}.$$

The claim then follows directly from the continuity of Q.

Fix any $x \in \mathcal{X}$. It is sufficient to prove that for any y in the set $\Lambda := \{y' : s(x,y') \notin \{\tilde{S}_j : j \in [N]\}\}$, we have $y \in \widehat{C}(x)$ if and only if $y \in \widehat{C}^{\text{fast}}(x)$ holds.

Let us first take any $y \in \widehat{C}^{\text{fast}}(x) \cap \Lambda$, and define $r_{m+1}^{(x,y)} = \sum_{i=1}^m \mathbb{1}\{S_i < s(x,y)\} + 1$. Then we have the following:

$$\begin{aligned}
\left\{y \in \widehat{C}^{\text{fast}}(x)\right\} &= \left(\left\{s(x,y) \leq \tilde{Q}'_{1-\alpha}\right\} \cap \left\{s(x,y) \leq S_{(\tilde{R}^-)}\right\}\right) \cup \left\{s(x,y) \leq S_{(\tilde{R}^+)}\right\} \\
&= \left(\left\{s(x,y) \leq \tilde{Q}'_{1-\alpha}\right\} \cap \left\{r_{m+1}^{(x,y)} \leq \tilde{R}^-\right\}\right) \cup \left\{r_{m+1}^{(x,y)} \leq \tilde{R}^+\right\} \quad \text{since } y \in \Lambda \\
&= \left(\left\{s(x,y) \leq \tilde{Q}'_{1-\alpha}\right\} \cap \left\{L_m(r_{m+1}^{(x,y)}) \leq \tilde{Q}_{1-\alpha}\right\}\right) \cup \left\{U_m(r_{m+1}^{(x,y)}) \leq \tilde{Q}_{1-\alpha}\right\},
\end{aligned}$$

and the final set can be expressed as a disjoint union of two events:

$$\text{(i) } s(x,y) \leq \tilde{Q}'_{1-\alpha} \text{ and } L_m(r_{m+1}^{(x,y)}) \leq \tilde{Q}_{1-\alpha} < U_m(r_{m+1}^{(x,y)}), \quad \text{(ii) } U_m(r_{m+1}^{(x,y)}) \leq \tilde{Q}_{1-\alpha}.$$

Note that in the case (ii), $T(s(x,y)) \leq \tilde{Q}_{1-\alpha}$ directly follows, since $T(s(x,y)) \leq U_m(r_{m+1}^{(x,y)})$ holds deterministically. In the case (i), we have $s(x,y) \leq \tilde{Q}'_{1-\alpha} \leq U_m(r_{m+1}^{(x,y)})$, and thus $T(s(x,y))$ is

equal to either $L_m(r_{m+1}^{(x,y)})$ or $\mathrm{NN}_m^-(r_{m+1}^{(x,y)}, s(x,y))$, which are both less than or equal to $\tilde{Q}_{1-\alpha}$: the first by the condition (i), the second by the definition of NN_m^-. Therefore, in either case, we have $y \in \widehat{C}(x)$.

Next, to prove the contrapositive, let $y \notin \widehat{C}^{\mathrm{fast}}(x)$—more precisely, $y \in \widehat{C}^{\mathrm{fast}}(x)^c \cap \Lambda$. From the observations above, we have

$$\begin{aligned}\left\{y \notin \widehat{C}^{\mathrm{fast}}(x)\right\} &= \left(\left\{s(x,y) > \tilde{Q}'_{1-\alpha}\right\} \cup \left\{L_m(r_{m+1}^{(x,y)}) > \tilde{Q}_{1-\alpha}\right\}\right) \cap \left\{U_m(r_{m+1}^{(x,y)}) > \tilde{Q}_{1-\alpha}\right\} \\ &= \left(\left\{s(x,y) > \tilde{Q}'_{1-\alpha}\right\} \cap \left\{U_m(r_{m+1}^{(x,y)}) > \tilde{Q}_{1-\alpha}\right\}\right) \cup \left\{L_m(r_{m+1}^{(x,y)}) > \tilde{Q}_{1-\alpha}\right\},\end{aligned}$$

where the second equality applies De Morgan's law. The final set is a disjoint union of the following two events:

$$\text{(i) } s(x,y) > \tilde{Q}'_{1-\alpha} \text{ and } L_m(r_{m+1}^{(x,y)}) \le \tilde{Q}_{1-\alpha} < U_m(r_{m+1}^{(x,y)}), \quad \text{(ii) } L_m(r_{m+1}^{(x,y)}) > \tilde{Q}_{1-\alpha}.$$

In case (ii), we have $T(s(x,y)) > \tilde{Q}_{1-\alpha}$, since $T(s(x,y)) \ge L_m(r_{m+1}^{(x,y)})$ holds deterministically. In case (i), $T(s(x,y))$ is equal to either $U_m(r_{m+1}^{(x,y)})$ or $\mathrm{NN}_m^-(r_{m+1}^{(x,y)}, s(x,y))$, which are both larger than $\tilde{Q}_{1-\alpha}$. This can be concluded as follows: In this case, we have $s(x,y) > \tilde{Q}'_{1-\alpha} \ge \tilde{Q}_{1-\alpha}$. If $L_m(r_{m+1}^{(x,y)}) \le s(x,y) < U_m(r_{m+1}^{(x,y)})$, by the construction of $T(s(x,y))$, we have $T(s(x,y)) = \mathrm{NN}_m^-(r_{m+1}^{(x,y)}, s(x,y)) \ge \tilde{Q}'_{1-\alpha} \ge \tilde{Q}_{1-\alpha}$, and moreover we cannot have equality, since $y \in \Lambda$. Otherwise, $s(x,y) \ge U_m(r_{m+1}^{(x,y)})$, and therefore, $T(s(x,y)) = U_m(r_{m+1}^{(x,y)}) > \tilde{Q}_{1-\alpha}$. Therefore, in both cases, we have $y \notin \widehat{C}(x)$, as desired.

H Experimental details

H.1 Setup and environment

The experiments were conducted on a system running Ubuntu 20.04.6 LTS, with 192 CPU cores of Intel(R) Xeon(R) Gold CPUs at 2.40 GHz, 1 TB of RAM, and 16 NVIDIA A40 GPUs. The software environment used Python 3.11.5, PyTorch 2.6, and CUDA 12.2.

H.2 Datasets

Our experiments involve two datasets: ImageNet for image classification tasks and the Medical Expenditure Panel Survey (MEPS) for regression tasks.

- **ImageNet** [14]: We use the training split of ImageNet, focusing on 30 selected classes, which are listed in Table S1.
- **MEPS**: The MEPS dataset is a medical survey used for regression tasks, with the goal of predicting healthcare expenditures. For the regression experiments: MEPS-19 [3], MEPS-20 [1], and MEPS-21 [2]. Each survey includes 139 features, such as demographic information (e.g., age, gender), and clinical data (e.g., chronic conditions, medical history).

H.3 Model details

We have applied the following models to compute the nonconformity scores:

- **ImageNet experiments**: We employed a CLIP model based ViT-B/32 backbone, pre-trained on the LAION-2B dataset [27, 43, 50], using the HuggingFace API. Table S1 reports the top-1 and top-2 accuracies of this model on the ImageNet training set for the subset of classes used in our experiments.
- **MEPS experiments**: The dataset was filtered to include only non-Hispanic White and non-White individuals. Panel-specific variables were renamed for consistency across panels 19-21, and rows with missing or invalid values were removed. A healthcare utilization variable was computed as the sum of expenditures across outpatient, office-based, emergency room, inpatient, and home health services, serving as the regression target. Preprocessing steps included retaining common features across panels, standardizing covariates, and applying a log transformation to the target variable to reduce skewness.

  A deep neural network was trained to estimate the lower and upper quantile bounds of healthcare utilization using a quantile regression approach, with different α levels used for each experiment. The network architecture consisted of four hidden layers (256, 128, 64, and 32 units) with LeakyReLU activations, dropout regularization (rate = 0.3), and optional batch normalization. The model was optimized using the pinball loss function and trained on 2019 data with early stopping based on validation loss (up to 50 epochs, batch size = 128, learning rate = 1e-4).

H.4 Data generation

H.4.1 Stable Diffusion

We generated synthetic images using the Stable Diffusion v1.5 model [46]. For each class listed in Table S1, we generated 2,000 images using the following configuration:

- Prompt: "A photo of a {class name}", where {class name} refers to the corresponding ImageNet label, as shown to be effective in [43].
- Inference steps: 260
- Guidance scale: 7.5

Figure S5 presents examples of generated images alongside real ImageNet training images from the same class.

H.4.2 FLUX

We generated synthetic images using the FLUX.1 model [30] from Black Forest Labs. For each class listed in Table S1, we generated 2,000 images using the following configuration:

- Prompt: "A photo of a class name", where class name refers to the corresponding ImageNet label, as shown to be effective in [43].
- Inference steps: 50

Images were generated using the `FluxPipeline` from the `diffusers` library, utilizing NVIDIA GPUs for acceleration with mixed-precision (`float16`) computation. The generation process was parallelized across multiple GPUs. Figure S6 presents examples of generated images alongside real ImageNet training images from the same class.

Table S1: Per-class accuracies of the pre-trained CLIP model with a ViT backbone on ImageNet. The first two columns (Top-1 and Top-2 accuracy) are computed over all ImageNet classes, while the last two columns are computed only over the subset of classes shown in this table.

Class	Top-1 (%)	Top-2 (%)	Top-1 (%)	Top-2 (%)
Junco, snowbird	91.8	95.1	94.7	98.3
Bulbul	89.8	96.2	96	99.5
Jay	9.6	22	29	58.3
Magpie	88.2	93.2	94.5	97.8
Golden retriever	66.5	78.4	83.9	95.5
Labrador retriever	53.8	66.3	83.9	94.2
English springer	58.7	79.9	96.2	97.7
Kuvasz	65.5	83.5	93	97.9
Siberian husky	13.8	40	87.3	94.2
Marmot	47.5	66	75.4	98.5
Beaver	59.6	73.5	93.6	98.5
Bicycle	91.5	96.2	97.8	99.9
Lighter, Light	35.3	44.3	72.2	84.8
Muzzle	52.5	62.1	89.2	94.5
Tennis ball	65.4	76.8	88.4	93.8
Torch	44.8	60.7	85.2	95.2
Unicycle	66.3	80.5	83.2	96.5
White wolf	63.9	79.5	87.3	95.4
Water ouzel	88.9	93.5	93.1	96.1
American robin	87.2	92.3	94.5	98.2
Admiral	0.1	0.1	0.1	0.1
Rock beauty	4.6	28.9	66.6	89.5
Papillon	59.8	73	93.8	97.3
Lycaenid butterfly	70.1	92.8	95.3	99.4
Gyromitra	0.1	0.1	0.1	0.1
Coral fungus	78.9	91	87.6	98.8
Stinkhorn	65.3	79.2	87.1	97.7
Barracouta	1.4	7.1	4.4	41.8
Garfish	48.2	64.8	85.8	94.3
Tinca tinca	89.4	94.2	96.5	98.8

Figure S5: Comparison between real and Stable Diffusion-generated images for selected ImageNet classes. Each row corresponds to a class, with the first column showing a real ImageNet image and the remaining columns showing generated datapoints.

Figure S6: Comparison between real and FLUX-generated images for selected ImageNet classes. Each row corresponds to a class, with the first column showing a real ImageNet image and the remaining columns showing generated datapoints.

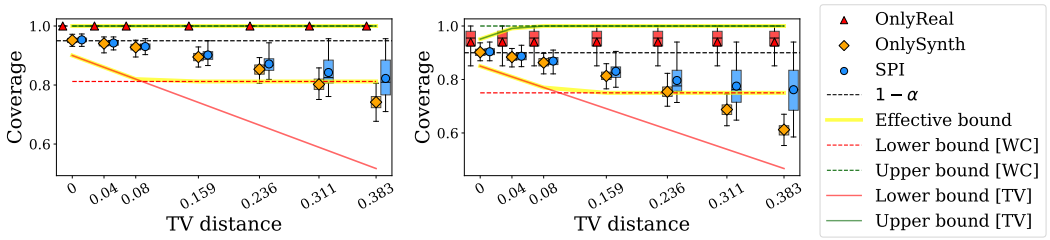
I Simulated data experiments

In this section, we present controlled experiments on simulated data, where the distributions of the real and synthetic scores are known. The goal is to compute and visualize the bounds from Theorem 3.3 and Theorem 3.5. The former provides bounds that depend on the total variation (TV) distance between the score distributions of the real and synthetic data, while the latter provides user-specified bounds (via the parameter β) that do not depend on these distributions. Our theoretical guarantees rely on both theorems, where the effective bound in each case is given by the tighter of the two, as illustrated in the following experiment.

Data and setup. We consider a simple regression setting with $\tilde{X} = X = 0$, real outcomes $Y \sim \mathcal{N}(5, 1)$, and synthetic outcomes $\tilde{Y} \sim \mathcal{N}(\mu, 1)$, where $\mu \in \{4, 4.2, 4.4, 4.6, 4.8, 4.9, 5\}$ controls the discrepancy between the two distributions. We use the absolute residual score function $s(X, Y) = |\hat{\mu}(X) - Y|$, where $\hat{\mu}(X) = X$. The real dataset includes $m = 15$ samples, while the synthetic dataset includes $N = 1,000$ samples. The test set consists of $1,000$ real samples. We set $\beta = 0.05$, and report results averaged over 100 independent trials.

Figure S7 shows the performance of all methods as a function of the TV distance for target levels $\alpha = 0.05$ and 0.1. The bounds derived from Theorem 3.5, denoted [WC], remain constant across different TV distances since they do not depend on the underlying data distributions. In contrast, the bounds from Theorem 3.3, denoted [TV], vary with the TV distance and become looser as the discrepancy between the real and synthetic distributions increases. The shaded yellow line indicates the effective bounds—defined as the tighter of the two in each case.

Following that figure, `SPI` achieves coverage within the effective bounds, as guaranteed by Theorems 3.3 and 3.5. Notably, for large TV distances—where the synthetic and real data differ substantially—the bounds from Theorem 3.5 dominate, providing the tighter guarantee even for a small β value. Moreover, as the user-specified parameter β increases, the [WC] bounds approach the nominal level $1 - \alpha$.



(a) $\alpha = 0.05$ (b) $\alpha = 0.1$

Figure S7: Results on simulated data: Coverage rate as a function of the TV distance between the real and synthetic score distributions. Bounds from Theorem 3.3 are labeled [TV], and those from Theorem 3.5 are labeled [WC]. Effective bounds (minimum/maximum of the two) are shown in yellow. Results are shown for $\alpha = 0.05$ (a) and 0.1 (b), with $\beta = 0.05$.

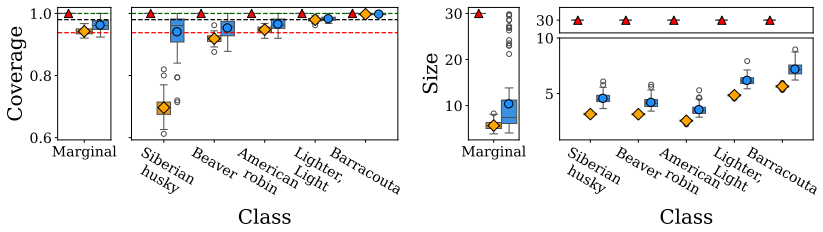
J Additional ImageNet experiments

This section provides supplementary results that complement those in Section 4.1, including additional experiments on the ImageNet dataset. The following two subsections—Appendices J.1 and J.2—correspond to Sections 4.1.1 and 4.1.2 of the main manuscript, respectively, and follow the same experimental settings.

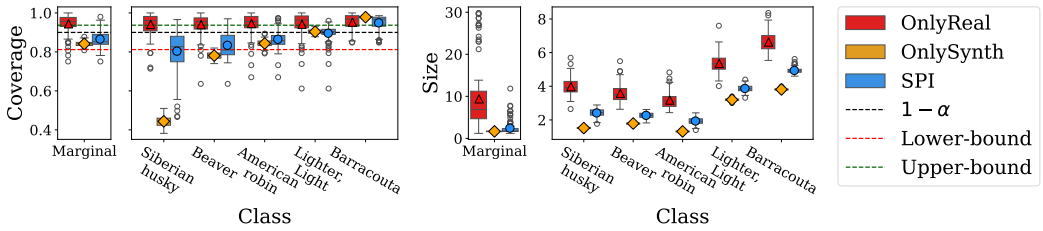
J.1 Experiments with generated synthetic data

Figure S8 presents the performance under both marginal and label-conditional guarantees at levels $\alpha = 0.02$ and 0.1. We observe a similar trend to that seen in Figure 3. Following that figure, we can see that the standard conformal prediction method, `OnlyReal`, controls the coverage at the $1 - \alpha$ level as expected, but it produces overly conservative prediction sets. `OnlySynth` method fails to

achieve the target coverage level of $1 - \alpha$, under-covering some classes, while in others, it becomes overly conservative, depending on the unknown distribution shift between the real and synthetic data. In contrast, the proposed method, `SPI`, stays within the theoretical bounds and produces informative prediction sets.



(a) $\alpha = 0.02$



(b) $\alpha = 0.1$

Figure S8: Results for the ImageNet data: Coverage rates of `OnlyReal`, `OnlySynth`, and `SPI` run at level $\alpha = 0.02$ (a) and 0.1 (b), averaged over 100 trials. Left: Average coverage. Right: Average prediction set size, both under marginal (leftmost box in each group) and label-conditional coverage settings. Label-conditional results are shown for selected classes; see Tables S2 and S4 for results across all classes.

Tables S2 to S4 present results for all 30 classes in the real calibration set corresponding to Figure 3 in the main manuscript and Figures S8a and S8b above.

Table S2: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials. Standard errors are shown in parentheses. The target coverage level is $1 - \alpha = 0.98$. The theoretical coverage guarantees for `SPI` are in the range $[93.7, 100]$. Other details are as in Figure S8.

	Coverage (%)			Size		
Class	`Only Real`	`Only Synth`	`SPI`	`Only Real`	`Only Synth`	`SPI`
Admiral	100 (± 0)	6.9 (± 0.3)	93.6 (± 0.6)	30 (± 0)	4.5 (± 0)	6.1 (± 0)
American robin	100 (± 0)	94.8 (± 0.1)	96.6 (± 0.2)	30 (± 0)	2.6 (± 0)	3.6 (± 0)
Barracouta	100 (± 0)	99.8 (± 0)	99.8 (± 0)	30 (± 0)	5.6 (± 0)	7.2 (± 0.1)
Beaver	100 (± 0)	91.9 (± 0.1)	95.3 (± 0.3)	30 (± 0)	3.1 (± 0)	4.2 (± 0)
Bicycle	100 (± 0)	96.3 (± 0.1)	97.3 (± 0.1)	30 (± 0)	2.8 (± 0)	3.5 (± 0)
Bulbul	100 (± 0)	99.4 (± 0)	99.5 (± 0)	30 (± 0)	2.6 (± 0)	3.7 (± 0)
Coral fungus	100 (± 0)	99.4 (± 0)	99.4 (± 0)	30 (± 0)	2.7 (± 0)	3.3 (± 0)
English springer	100 (± 0)	94.2 (± 0.1)	96.5 (± 0.2)	30 (± 0)	2.9 (± 0)	4.1 (± 0)
Garfish	100 (± 0)	92.1 (± 0.1)	95.3 (± 0.3)	30 (± 0)	4.6 (± 0)	5.9 (± 0)
Golden retriever	100 (± 0)	94.3 (± 0.1)	96.4 (± 0.2)	30 (± 0)	4.3 (± 0)	5.8 (± 0.1)
Gyromitra	100 (± 0)	92.4 (± 0.2)	96.4 (± 0.3)	30 (± 0)	3.1 (± 0)	3.7 (± 0)
Jay	100 (± 0)	92.0 (± 0.2)	95.6 (± 0.3)	30 (± 0)	6.4 (± 0)	8.0 (± 0.1)
Junco, snowbird	100 (± 0)	97.5 (± 0.1)	98.0 (± 0.1)	30 (± 0)	2.3 (± 0)	3.1 (± 0)
Kuvasz	100 (± 0)	95.1 (± 0.1)	96.5 (± 0.2)	30 (± 0)	2.8 (± 0)	4.0 (± 0)
Labrador retriever	100 (± 0)	96.2 (± 0.1)	97.2 (± 0.1)	30 (± 0)	4.6 (± 0)	6.3 (± 0.1)
Lighter, Light	100 (± 0)	98.0 (± 0.1)	98.3 (± 0.1)	30 (± 0)	4.9 (± 0)	6.2 (± 0)
Lycaenid butterfly	100 (± 0)	93.5 (± 0.1)	95.7 (± 0.2)	30 (± 0)	2.9 (± 0)	4.1 (± 0.1)
Magpie	100 (± 0)	96.7 (± 0.1)	97.4 (± 0.1)	30 (± 0)	2.5 (± 0)	3.4 (± 0)
Marmot	100 (± 0)	95.5 (± 0.1)	97.0 (± 0.2)	30 (± 0)	4.0 (± 0)	5.6 (± 0.1)
Muzzle	100 (± 0)	97.5 (± 0)	98.0 (± 0.1)	30 (± 0)	3.5 (± 0)	4.9 (± 0)
Papillon	100 (± 0)	89.7 (± 0.2)	94.9 (± 0.4)	30 (± 0)	3.1 (± 0)	4.3 (± 0)
Rock beauty	100 (± 0)	90.6 (± 0.2)	95.3 (± 0.3)	30 (± 0)	4.8 (± 0)	6.0 (± 0)
Siberian husky	100 (± 0)	69.7 (± 0.3)	94.1 (± 0.6)	30 (± 0)	3.2 (± 0)	4.6 (± 0)
Stinkhorn	100 (± 0)	97.9 (± 0.1)	98.2 (± 0.1)	30 (± 0)	4.6 (± 0)	5.5 (± 0)
Tennis ball	100 (± 0)	97.5 (± 0.1)	98.0 (± 0.1)	30 (± 0)	3.1 (± 0)	4.0 (± 0)
Tinca tinca	100 (± 0)	98.5 (± 0)	98.6 (± 0.1)	30 (± 0)	3.3 (± 0)	4.2 (± 0)
Torch	100 (± 0)	98.3 (± 0)	98.7 (± 0.1)	30 (± 0)	5.6 (± 0)	7.1 (± 0.1)
Unicycle	100 (± 0)	96.3 (± 0.1)	97.2 (± 0.1)	30 (± 0)	4.2 (± 0)	5.5 (± 0)
Water ouzel	100 (± 0)	95.7 (± 0.1)	97.0 (± 0.2)	30 (± 0)	2.5 (± 0)	3.3 (± 0)
White wolf	100 (± 0)	85.5 (± 0.2)	94.3 (± 0.5)	30 (± 0)	2.9 (± 0)	4.0 (± 0)

Table S3: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials. Standard errors are shown in parentheses. The target coverage level is $1 - \alpha = 0.95$. The theoretical coverage guarantees for `SPI` are in the range $[93.7, 100]$. Other details are as in Figure 3.

	Coverage (%)			Size		
Class	`Only Real`	`Only Synth`	`SPI`	`Only Real`	`Only Synth`	`SPI`
Admiral	100 (± 0)	0.6 (± 0)	93.6 (± 0.6)	30 (± 0)	3.6 (± 0)	5.8 (± 0)
American robin	100 (± 0)	90.5 (± 0.2)	95.4 (± 0.3)	30 (± 0)	1.8 (± 0)	3.3 (± 0)
Barracouta	100 (± 0)	99.3 (± 0)	99.4 (± 0)	30 (± 0)	4.5 (± 0)	6.8 (± 0.1)
Beaver	100 (± 0)	85.1 (± 0.2)	94.3 (± 0.5)	30 (± 0)	2.5 (± 0)	3.9 (± 0)
Bicycle	100 (± 0)	92.5 (± 0.1)	95.6 (± 0.3)	30 (± 0)	2.3 (± 0)	3.3 (± 0)
Bulbul	100 (± 0)	98.4 (± 0.1)	98.6 (± 0.1)	30 (± 0)	2.0 (± 0)	3.4 (± 0)
Coral fungus	100 (± 0)	98.7 (± 0)	98.8 (± 0.1)	30 (± 0)	2.1 (± 0)	3.1 (± 0)
English springer	100 (± 0)	90.1 (± 0.1)	95.2 (± 0.4)	30 (± 0)	2.2 (± 0)	3.8 (± 0)
Garfish	100 (± 0)	86.4 (± 0.1)	94.2 (± 0.4)	30 (± 0)	3.7 (± 0)	5.6 (± 0)
Golden retriever	100 (± 0)	88.8 (± 0.2)	94.9 (± 0.4)	30 (± 0)	3.3 (± 0)	5.3 (± 0.1)
Gyromitra	100 (± 0)	80.3 (± 0.3)	95.1 (± 0.5)	30 (± 0)	2.7 (± 0)	3.6 (± 0)
Jay	100 (± 0)	80.4 (± 0.2)	93.7 (± 0.6)	30 (± 0)	4.8 (± 0)	7.4 (± 0.1)
Junco, snowbird	100 (± 0)	94.7 (± 0.1)	96.4 (± 0.2)	30 (± 0)	1.7 (± 0)	2.9 (± 0)
Kuvasz	100 (± 0)	91.6 (± 0.1)	95.1 (± 0.3)	30 (± 0)	2.1 (± 0)	3.7 (± 0)
Labrador retriever	100 (± 0)	91.9 (± 0.1)	95.6 (± 0.3)	30 (± 0)	3.6 (± 0)	5.8 (± 0.1)
Lighter, Light	100 (± 0)	95.2 (± 0.1)	96.8 (± 0.2)	30 (± 0)	3.9 (± 0)	5.8 (± 0)
Lycaenid butterfly	100 (± 0)	88.2 (± 0.1)	94.5 (± 0.4)	30 (± 0)	2.3 (± 0)	4.0 (± 0.1)
Magpie	100 (± 0)	93.6 (± 0.1)	95.7 (± 0.2)	30 (± 0)	1.9 (± 0)	3.1 (± 0)
Marmot	100 (± 0)	93.0 (± 0.1)	96.2 (± 0.3)	30 (± 0)	3.2 (± 0)	5.2 (± 0.1)
Muzzle	100 (± 0)	95.7 (± 0.1)	96.9 (± 0.2)	30 (± 0)	2.8 (± 0)	4.6 (± 0)
Papillon	100 (± 0)	83.6 (± 0.2)	94.1 (± 0.5)	30 (± 0)	2.3 (± 0)	4.0 (± 0)
Rock beauty	100 (± 0)	68.5 (± 0.4)	94.2 (± 0.5)	30 (± 0)	3.5 (± 0)	5.5 (± 0)
Siberian husky	100 (± 0)	56.7 (± 0.3)	94.1 (± 0.6)	30 (± 0)	2.3 (± 0)	4.2 (± 0)
Stinkhorn	100 (± 0)	95.6 (± 0.1)	96.6 (± 0.2)	30 (± 0)	3.6 (± 0)	5.1 (± 0)
Tennis ball	100 (± 0)	94.3 (± 0.1)	96.0 (± 0.2)	30 (± 0)	2.5 (± 0)	3.7 (± 0)
Tinca tinca	100 (± 0)	96.5 (± 0.1)	97.2 (± 0.1)	30 (± 0)	2.6 (± 0)	4.0 (± 0)
Torch	100 (± 0)	96.7 (± 0.1)	97.6 (± 0.1)	30 (± 0)	4.5 (± 0)	6.6 (± 0.1)
Unicycle	100 (± 0)	92.9 (± 0.1)	95.7 (± 0.3)	30 (± 0)	3.3 (± 0)	5.1 (± 0)
Water ouzel	100 (± 0)	92.6 (± 0.1)	95.7 (± 0.3)	30 (± 0)	1.8 (± 0)	3.1 (± 0)
White wolf	100 (± 0)	80.5 (± 0.2)	93.9 (± 0.6)	30 (± 0)	2.1 (± 0)	3.7 (± 0)

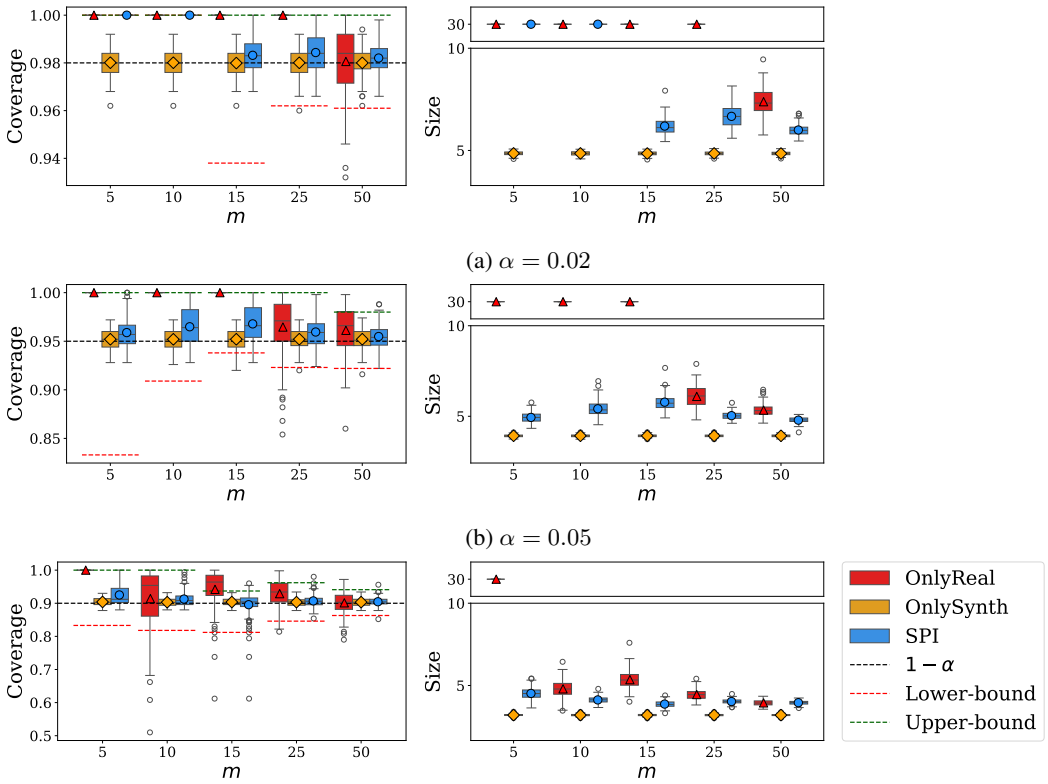
Table S4: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials. Standard errors are shown in parentheses. The target coverage level is $1 - \alpha = 0.9$. The theoretical coverage guarantees for `SPI` are in the range $[81.2, 93.7]$. Other details as in Figure S8.

	Coverage (%)			Size		
Class	`Only Real`	`Only Synth`	`SPI`	`Only Real`	`Only Synth`	`SPI`
Admiral	93.6 (± 0.6)	0.2 (± 0)	81.3 (± 0.9)	5.6 (± 0)	3.1 (± 0)	4.3 (± 0)
American robin	94.5 (± 0.5)	84.3 (± 0.2)	86.5 (± 0.5)	3.2 (± 0)	1.3 (± 0)	1.9 (± 0)
Barracouta	95.3 (± 0.4)	97.8 (± 0)	94.9 (± 0.4)	6.6 (± 0.1)	3.8 (± 0)	4.9 (± 0)
Beaver	94.0 (± 0.6)	78.1 (± 0.2)	83.4 (± 0.6)	3.6 (± 0.1)	1.8 (± 0)	2.3 (± 0)
Bicycle	94.2 (± 0.5)	87.1 (± 0.2)	88.2 (± 0.4)	3.2 (± 0)	2.0 (± 0)	2.3 (± 0)
Bulbul	93.8 (± 0.6)	95.8 (± 0.1)	92.8 (± 0.6)	3.3 (± 0)	1.6 (± 0)	2.1 (± 0)
Coral fungus	93.5 (± 0.6)	97.7 (± 0.1)	93.2 (± 0.6)	2.9 (± 0)	1.7 (± 0)	2.1 (± 0)
English springer	93.9 (± 0.6)	83.1 (± 0.2)	85.3 (± 0.4)	3.6 (± 0)	1.5 (± 0)	2.2 (± 0)
Garfish	93.5 (± 0.6)	79.2 (± 0.2)	83.4 (± 0.5)	5.4 (± 0.1)	3.2 (± 0)	4.0 (± 0)
Golden retriever	94.1 (± 0.6)	82.4 (± 0.2)	85.8 (± 0.5)	5.0 (± 0.1)	2.3 (± 0)	3.1 (± 0)
Gyromitra	95.0 (± 0.6)	64.1 (± 0.3)	85.2 (± 0.9)	3.3 (± 0)	2.3 (± 0)	2.5 (± 0)
Jay	93.3 (± 0.7)	67.1 (± 0.3)	80.9 (± 0.9)	6.8 (± 0.1)	3.7 (± 0)	4.8 (± 0)
Junco, snowbird	94.2 (± 0.5)	90.0 (± 0.1)	89.9 (± 0.4)	2.7 (± 0)	1.3 (± 0)	1.7 (± 0)
Kuvasz	93.4 (± 0.6)	85.0 (± 0.2)	86.3 (± 0.4)	3.5 (± 0)	1.5 (± 0)	2.0 (± 0)
Labrador retriever	93.5 (± 0.7)	84.6 (± 0.2)	85.9 (± 0.5)	5.4 (± 0.1)	2.7 (± 0)	3.5 (± 0)
Lighter, Light	94.2 (± 0.6)	90.4 (± 0.1)	89.6 (± 0.4)	5.4 (± 0.1)	3.2 (± 0)	3.9 (± 0)
Lycaenid butterfly	94.0 (± 0.5)	81.3 (± 0.2)	85.7 (± 0.5)	3.9 (± 0.1)	1.9 (± 0)	2.5 (± 0)
Magpie	93.5 (± 0.6)	88.3 (± 0.2)	88.3 (± 0.5)	3.0 (± 0)	1.4 (± 0)	1.9 (± 0)
Marmot	94.0 (± 0.6)	89.8 (± 0.1)	89.1 (± 0.4)	4.9 (± 0.1)	2.6 (± 0)	3.1 (± 0)
Muzzle	93.5 (± 0.6)	92.1 (± 0.1)	90.8 (± 0.5)	4.3 (± 0)	2.3 (± 0)	3.0 (± 0)
Papillon	93.8 (± 0.6)	75.8 (± 0.2)	82.4 (± 0.6)	3.8 (± 0.1)	1.7 (± 0)	2.4 (± 0)
Rock beauty	94.2 (± 0.5)	44.6 (± 0.3)	80.7 (± 1.0)	5.1 (± 0.1)	2.5 (± 0)	3.8 (± 0)
Siberian husky	94.1 (± 0.6)	44.4 (± 0.3)	80.4 (± 1.1)	4.0 (± 0)	1.5 (± 0)	2.4 (± 0)
Stinkhorn	93.4 (± 0.6)	92.2 (± 0.1)	90.7 (± 0.4)	4.5 (± 0.1)	2.8 (± 0)	3.2 (± 0)
Tennis ball	93.5 (± 0.6)	88.6 (± 0.1)	88.6 (± 0.3)	3.5 (± 0)	2.0 (± 0)	2.5 (± 0)
Tinca tinca	93.2 (± 0.6)	93.2 (± 0.1)	91.2 (± 0.5)	3.8 (± 0)	2.1 (± 0)	2.7 (± 0)
Torch	94.8 (± 0.5)	93.9 (± 0.1)	92.6 (± 0.4)	6.2 (± 0.1)	3.8 (± 0)	4.6 (± 0)
Unicycle	93.3 (± 0.6)	86.5 (± 0.2)	87.1 (± 0.4)	4.8 (± 0)	2.8 (± 0)	3.3 (± 0)
Water ouzel	94.3 (± 0.5)	87.7 (± 0.1)	88.3 (± 0.3)	3.0 (± 0)	1.4 (± 0)	1.9 (± 0)
White wolf	93.9 (± 0.6)	74.1 (± 0.2)	82.4 (± 0.7)	3.4 (± 0)	1.5 (± 0)	2.1 (± 0)

J.1.1 The effect of the real calibration set size

Here, we evaluate the performance of different methods as a function of the real calibration set size m, following the same setup described in Section 4.1.1. This parameter directly affects the performance of both the standard conformal prediction method, `OnlyReal`, and our proposed method, `SPI`, including the theoretical bounds established in Theorem 3.5. In contrast, `OnlySynth`, which relies solely on the synthetic calibration set, is unaffected by changes in m. As such, it serves as a useful baseline for assessing how well the synthetic calibration set aligns with the real one.

Figure S9 presents the performance of all methods for the "Lighter" class across varying values of m and α levels. Notably, although `OnlySynth` does not have formal coverage guarantees, its empirical coverage closely matches the target level $1-\alpha$. This alignment suggests that the synthetic calibration data approximate the real distribution well.



(a) $\alpha = 0.02$

(b) $\alpha = 0.05$

(c) $\alpha = 0.1$

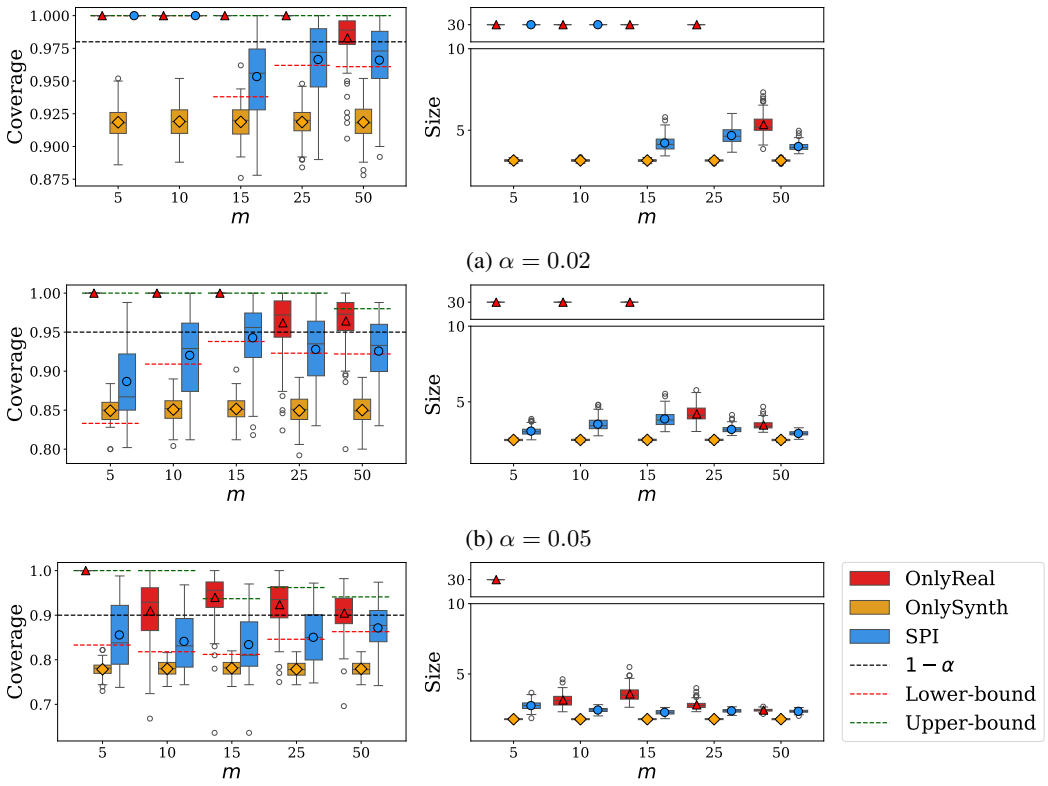
Figure S9: Results for the ImageNet data: Coverage rate for `OnlyReal`, `OnlySynth`, and `SPI` on the "Lighter" class as a function of the real calibration set size m, for levels $\alpha = 0.02$ (a), $\alpha = 0.05$ (b), and $\alpha = 0.1$ (c).

Figure S9a presents results for $\alpha = 0.02$. At this low level, the standard conformal prediction, `OnlyReal`, controls the coverage at level $1-\alpha$, but—as expected—produces trivial prediction sets when $m < 50$.

In contrast, our proposed method, `SPI`, achieves coverage within the theoretical bounds even for small m, with reduced variance in coverage and smaller prediction sets for $m \geq 15$. Interestingly, for $\alpha = 0.02$ and $m = 5$ or 10, the theoretical lower and upper coverage bounds are both equal to unity, indicating that we know *a priori* that the proposed method yields trivial prediction sets for this window construction.

For $\alpha = 0.05$ and $\alpha = 0.1$ (Figures S9b and S9c, respectively), we observe similar trends. Our method, `SPI`, consistently achieves coverage within the theoretical bounds, remaining close to the target coverage level $1-\alpha$, while also exhibiting reduced variance in coverage and producing smaller, more informative prediction sets compared to the baseline, `OnlyReal`.

Additionally, Figure S10 presents the same experiment as in Figure S9, but for the "Beaver" class. In this case, the `OnlySynth` method yields coverage that falls significantly below the target level $1-\alpha$, indicating that the synthetic calibration set differs substantially from the real data. Nevertheless, our proposed method, `SPI`, achieves coverage within the theoretical bounds across all α levels and calibration set sizes, while also producing informative prediction sets.



(a) $\alpha = 0.02$

(b) $\alpha = 0.05$

(c) $\alpha = 0.1$

Figure S10: Results for the ImageNet data: Coverage rate for `OnlyReal`, `OnlySynth`, and `SPI` on the Beaver class as a function of the real calibration set size m, for levels $\alpha = 0.02$ (a), $\alpha = 0.05$ (b), and $\alpha = 0.1$ (c).

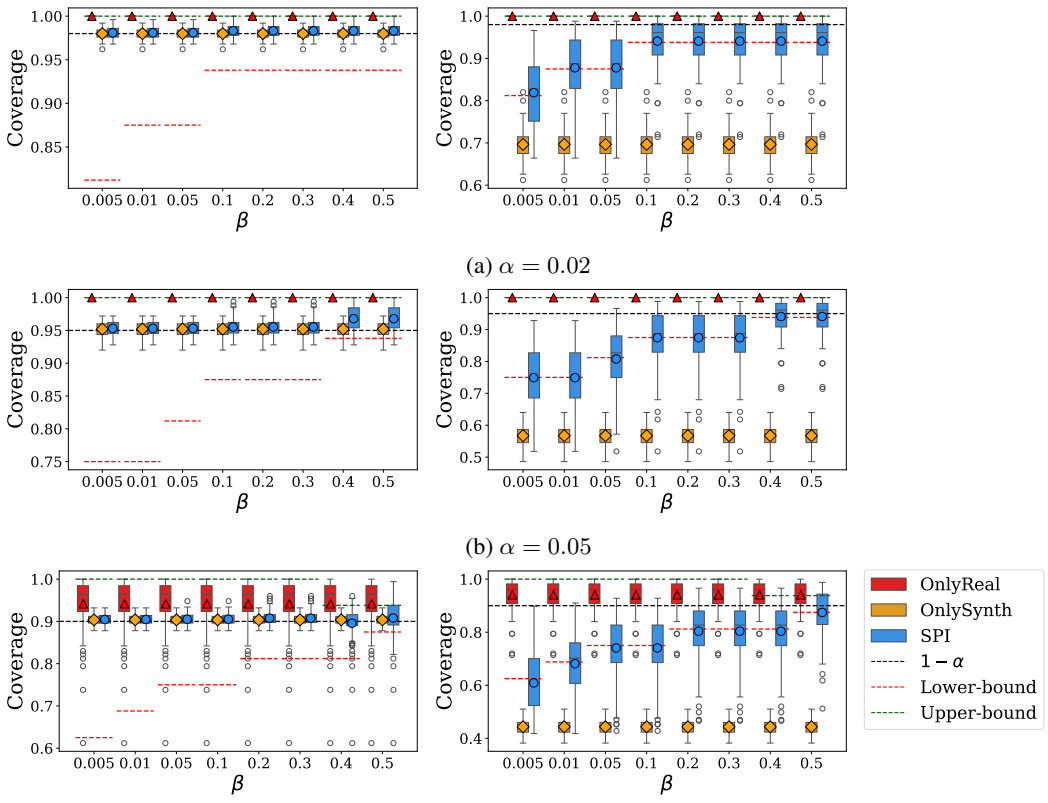
J.1.2 The effect of the hyperparameter β

In this section, we examine the effect of the hyperparameter β on the performance of our proposed method, `SPI`. At a theoretical level, the sensitivity of `SPI` to β depends on the similarity between the synthetic and real score distributions. Specifically, this can be seen in the score-transportation step: when the two score distributions differ substantially, the transported score $T(S_{m+1})$ is likely to lie at one of the endpoints of the corresponding window—even under a small β. As a result, the coverage of the `SPI` prediction set approaches the guardrail bound in Theorem 3.5, indicating that we gain little from the synthetic data.

On the other hand, when the synthetic scores closely resemble the real scores, the transported score is likely to lie within the corresponding window rather than in its endpoints—even for relatively large values of β. In this case, the `SPI` prediction set resembles conformal prediction with a larger sample size, achieving coverage close to the nominal $1-\alpha$ level, and the effect of the hyperparameter β becomes relatively minor.

Figure S11 illustrates this behavior, showing the coverage of all methods for different β values across two classes—lighter (left column) and Siberian husky (right column) classes—at target levels $\alpha = 0.02, 0.05$, and 0.1. Following that figure, for all β values, the empirical coverage of `SPI` lies within the theoretical bounds of Theorem 3.5.

The results align with the behavior described above: for the lighter class (left column), the synthetic scores closely resemble the real ones, and accordingly, the performance of `SPI` remains roughly the same across different β values, achieving coverage close to the nominal $1-\alpha$ level. In contrast, for the Siberian husky class (right column), where the synthetic scores deviate significantly from the real scores, the coverage of `SPI` closely follows the guardrail bounds. Note that this figure also presents how the lower bound of Theorem 3.5 increases with β, becoming closer to the nominal $1-\alpha$ level.



(a) $\alpha = 0.02$

(b) $\alpha = 0.05$

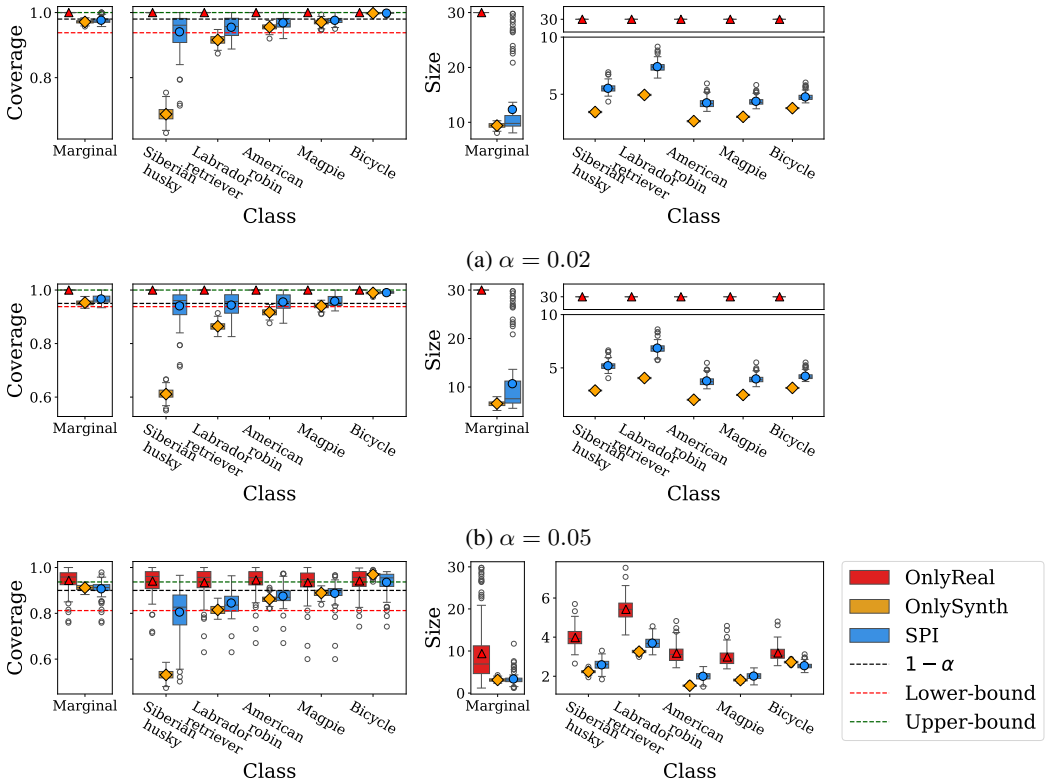
(c) $\alpha = 0.1$

Figure S11: Results for the ImageNet data: Coverage rate for `OnlyReal`, `OnlySynth`, and `SPI` as a function of β. Results are displayed for the Siberian husky (right column) and lighter (left column) classes, at levels $\alpha = 0.02$ (a), $\alpha = 0.05$ (b), and $\alpha = 0.1$ (c).

J.1.3 Results for SPI with FLUX-generated synthetic data

In this section, we evaluate the performance of the proposed method, SPI, using synthetic images generated by the FLUX.1 model [30]. The experimental setup follows the same procedure described in Section 4.1.1 of the main manuscript. As before, we aim for both marginal and label-conditional coverage guarantees.

Figure S12 presents the marginal and label-conditional coverage of various methods at levels $\alpha = 0.02$, 0.05, and 0.1. The results for label-conditional guarantees are presented for representative classes; results for all classes in the real population are detailed in Tables S5 to S7. We observe similar trends to those observed using synthetic images generated by Stable Diffusion. The standard conformal method, OnlyReal, controls the coverage at the $1 - \alpha$ level; however, it yields overly conservative prediction sets due to the small sample size. OnlySynth fails to control the coverage at the desired level, exhibiting under-coverage for some classes and over-coverage for others. In contrast, the proposed method, SPI, achieves coverage within the theoretical bounds while providing smaller, more informative prediction sets.



(a) $\alpha = 0.02$

(b) $\alpha = 0.05$

(c) $\alpha = 0.1$

Figure S12: Results for the ImageNet data using FLUX-generated synthetic images: Coverage rates of OnlyReal, OnlySynth, and SPI run at level $\alpha = 0.02$ (a), 0.05 (b), and 0.1 (b), averaged over 100 trials. Left: Average coverage. Right: Average prediction set size, both under marginal (leftmost box in each group) and label-conditional coverage settings. Label-conditional results are shown for selected classes; see Tables S5 to S7 for results across all classes.

Table S5: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials using FLUX-generated synthetic data. The target coverage level is $1 - \alpha = 0.98$. The theoretical coverage guarantees for `SPI` are in the range $[93.7, 100]$. Standard errors are shown in parentheses. Other experimental details follow Figure S12.

	Coverage (%)			Size		
Class	`Only Real`	`Only Synth`	`SPI`	`Only Real`	`Only Synth`	`SPI`
Admiral	100 (± 0)	0.2 (± 0)	93.6 (± 0.6)	30 (± 0)	4.7 (± 0)	6.5 (± 0)
American robin	100 (± 0)	95.5 (± 0.1)	96.8 (± 0.2)	30 (± 0)	2.7 (± 0)	4.3 (± 0)
Barracouta	100 (± 0)	99.9 (± 0)	99.9 (± 0)	30 (± 0)	5.6 (± 0)	8.2 (± 0.1)
Beaver	100 (± 0)	86.7 (± 0.2)	94.5 (± 0.4)	30 (± 0)	4.1 (± 0)	5.6 (± 0)
Bicycle	100 (± 0)	99.8 (± 0)	99.8 (± 0)	30 (± 0)	3.8 (± 0)	4.8 (± 0)
Bulbul	100 (± 0)	97.3 (± 0.1)	97.9 (± 0.1)	30 (± 0)	3.2 (± 0)	4.7 (± 0)
Coral fungus	100 (± 0)	99.6 (± 0)	99.6 (± 0)	30 (± 0)	2.9 (± 0)	3.9 (± 0)
English springer	100 (± 0)	96.2 (± 0.1)	97.3 (± 0.1)	30 (± 0)	3.3 (± 0)	5.1 (± 0)
Garfish	100 (± 0)	90.4 (± 0.1)	95.0 (± 0.3)	30 (± 0)	4.9 (± 0)	6.9 (± 0)
Golden retriever	100 (± 0)	93.6 (± 0.1)	96.0 (± 0.2)	30 (± 0)	4.6 (± 0)	6.9 (± 0)
Gyromitra	100 (± 0)	57.2 (± 0.3)	95.0 (± 0.6)	30 (± 0)	3.6 (± 0)	4.5 (± 0)
Jay	100 (± 0)	50.1 (± 0.5)	93.3 (± 0.7)	30 (± 0)	6.9 (± 0)	9.1 (± 0)
Junco, snowbird	100 (± 0)	99.2 (± 0)	99.3 (± 0)	30 (± 0)	2.4 (± 0)	3.6 (± 0)
Kuvasz	100 (± 0)	99.4 (± 0)	99.4 (± 0)	30 (± 0)	2.8 (± 0)	4.6 (± 0)
Labrador retriever	100 (± 0)	91.6 (± 0.2)	95.5 (± 0.3)	30 (± 0)	4.9 (± 0)	7.4 (± 0)
Lighter, Light	100 (± 0)	75.6 (± 0.2)	94.4 (± 0.6)	30 (± 0)	4.7 (± 0)	7.0 (± 0)
Lycaenid butterfly	100 (± 0)	93.4 (± 0.1)	95.7 (± 0.2)	30 (± 0)	3.8 (± 0)	4.7 (± 0)
Magpie	100 (± 0)	97.0 (± 0.1)	97.6 (± 0.1)	30 (± 0)	3.0 (± 0)	4.4 (± 0)
Marmot	100 (± 0)	98.0 (± 0.1)	98.3 (± 0.1)	30 (± 0)	4.9 (± 0)	6.9 (± 0.1)
Muzzle	100 (± 0)	96.2 (± 0.1)	97.2 (± 0.1)	30 (± 0)	3.7 (± 0)	5.8 (± 0)
Papillon	100 (± 0)	99.9 (± 0)	99.9 (± 0)	30 (± 0)	2.5 (± 0)	4.4 (± 0)
Rock beauty	100 (± 0)	93.6 (± 0.1)	96.0 (± 0.2)	30 (± 0)	4.9 (± 0)	7.1 (± 0)
Siberian husky	100 (± 0)	68.8 (± 0.2)	94.1 (± 0.6)	30 (± 0)	3.4 (± 0)	5.5 (± 0)
Stinkhorn	100 (± 0)	98.3 (± 0)	98.5 (± 0.1)	30 (± 0)	5.1 (± 0)	6.5 (± 0)
Tennis ball	100 (± 0)	89.5 (± 0.1)	94.5 (± 0.4)	30 (± 0)	3.4 (± 0)	4.8 (± 0)
Tinca tinca	100 (± 0)	99.4 (± 0)	99.4 (± 0)	30 (± 0)	3.3 (± 0)	5.0 (± 0)
Torch	100 (± 0)	91.5 (± 0.1)	95.8 (± 0.3)	30 (± 0)	6.0 (± 0)	8.5 (± 0)
Unicycle	100 (± 0)	99.8 (± 0)	99.8 (± 0)	30 (± 0)	4.8 (± 0)	6.6 (± 0)
Water ouzel	100 (± 0)	99.1 (± 0)	99.2 (± 0)	30 (± 0)	2.6 (± 0)	3.9 (± 0)
White wolf	100 (± 0)	83.9 (± 0.2)	94.2 (± 0.5)	30 (± 0)	3.4 (± 0)	5.1 (± 0)

Table S6: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials using FLUX-generated synthetic data. The target coverage level is $1 - \alpha = 0.95$. The theoretical coverage guarantees for `SPI` are in the range $[93.7, 100]$. Standard errors are shown in parentheses. Other experimental details follow Figure S12.

	Coverage (%)			Size		
Class	`Only Real`	`Only Synth`	`SPI`	`Only Real`	`Only Synth`	`SPI`
Admiral	100 (± 0)	0.2 (± 0)	93.6 (± 0.6)	30 (± 0)	4.4 (± 0)	6.3 (± 0)
American robin	100 (± 0)	91.7 (± 0.1)	95.6 (± 0.3)	30 (± 0)	2.0 (± 0)	3.8 (± 0)
Barracouta	100 (± 0)	99.9 (± 0)	99.9 (± 0)	30 (± 0)	4.8 (± 0)	7.7 (± 0.1)
Beaver	100 (± 0)	81.0 (± 0.2)	94.1 (± 0.5)	30 (± 0)	3.1 (± 0)	4.8 (± 0)
Bicycle	100 (± 0)	98.9 (± 0)	99.0 (± 0)	30 (± 0)	3.1 (± 0)	4.2 (± 0)
Bulbul	100 (± 0)	93.8 (± 0.1)	96.2 (± 0.2)	30 (± 0)	2.6 (± 0)	4.3 (± 0)
Coral fungus	100 (± 0)	99.4 (± 0)	99.4 (± 0)	30 (± 0)	2.4 (± 0)	3.6 (± 0)
English springer	100 (± 0)	95.1 (± 0.1)	96.8 (± 0.2)	30 (± 0)	3.0 (± 0)	4.9 (± 0)
Garfish	100 (± 0)	84.7 (± 0.2)	93.9 (± 0.5)	30 (± 0)	4.0 (± 0)	6.3 (± 0)
Golden retriever	100 (± 0)	89.9 (± 0.1)	95.0 (± 0.3)	30 (± 0)	3.8 (± 0)	6.5 (± 0.1)
Gyromitra	100 (± 0)	46.6 (± 0.2)	95.0 (± 0.6)	30 (± 0)	3.0 (± 0)	4.1 (± 0)
Jay	100 (± 0)	31.7 (± 0.2)	93.3 (± 0.7)	30 (± 0)	5.7 (± 0)	8.5 (± 0.1)
Junco, snowbird	100 (± 0)	97.9 (± 0.1)	98.3 (± 0.1)	30 (± 0)	2.0 (± 0)	3.3 (± 0)
Kuvasz	100 (± 0)	99.2 (± 0)	99.3 (± 0)	30 (± 0)	2.3 (± 0)	4.3 (± 0)
Labrador retriever	100 (± 0)	86.5 (± 0.2)	94.4 (± 0.5)	30 (± 0)	4.1 (± 0)	6.9 (± 0.1)
Lighter, Light	100 (± 0)	67.0 (± 0.2)	94.2 (± 0.6)	30 (± 0)	3.9 (± 0)	6.7 (± 0)
Lycaenid butterfly	100 (± 0)	88.5 (± 0.1)	94.6 (± 0.4)	30 (± 0)	3.3 (± 0)	4.6 (± 0)
Magpie	100 (± 0)	94.0 (± 0.1)	95.8 (± 0.2)	30 (± 0)	2.5 (± 0)	4.0 (± 0)
Marmot	100 (± 0)	96.9 (± 0.1)	97.7 (± 0.1)	30 (± 0)	3.9 (± 0)	6.2 (± 0.1)
Muzzle	100 (± 0)	94.2 (± 0.1)	96.2 (± 0.2)	30 (± 0)	3.2 (± 0)	5.5 (± 0)
Papillon	100 (± 0)	99.9 (± 0)	99.9 (± 0)	30 (± 0)	2.2 (± 0)	4.3 (± 0)
Rock beauty	100 (± 0)	87.9 (± 0.1)	94.7 (± 0.4)	30 (± 0)	4.4 (± 0)	6.8 (± 0)
Siberian husky	100 (± 0)	61.1 (± 0.2)	94.1 (± 0.6)	30 (± 0)	2.9 (± 0)	5.2 (± 0)
Stinkhorn	100 (± 0)	97.3 (± 0.1)	97.7 (± 0.1)	30 (± 0)	4.4 (± 0)	6.0 (± 0)
Tennis ball	100 (± 0)	84.4 (± 0.2)	93.7 (± 0.5)	30 (± 0)	2.8 (± 0)	4.4 (± 0)
Tinca tinca	100 (± 0)	98.7 (± 0)	98.8 (± 0.1)	30 (± 0)	2.7 (± 0)	4.6 (± 0)
Torch	100 (± 0)	86.9 (± 0.2)	95.2 (± 0.4)	30 (± 0)	5.2 (± 0)	8.0 (± 0)
Unicycle	100 (± 0)	99.8 (± 0)	99.8 (± 0)	30 (± 0)	4.0 (± 0)	6.0 (± 0)
Water ouzel	100 (± 0)	98.9 (± 0)	99.0 (± 0)	30 (± 0)	2.0 (± 0)	3.5 (± 0)
White wolf	100 (± 0)	78.5 (± 0.2)	93.9 (± 0.6)	30 (± 0)	2.8 (± 0)	4.7 (± 0)

Table S7: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials using FLUX-generated synthetic data. The target coverage level is $1 - \alpha = 0.9$. The theoretical coverage guarantees for `SPI` are in the range $[81.2, 93.7]$. Standard errors are shown in parentheses. Other experimental details follow Figure S12.

	Coverage (%)			Size		
Class	`Only Real`	`Only Synth`	`SPI`	`Only Real`	`Only Synth`	`SPI`
Admiral	93.6 (± 0.6)	0.1 (± 0)	81.3 (± 0.9)	5.6 (± 0)	4.1 (± 0)	4.6 (± 0)
American robin	94.5 (± 0.5)	86.2 (± 0.2)	87.5 (± 0.4)	3.2 (± 0)	1.5 (± 0)	2.0 (± 0)
Barracouta	95.3 (± 0.4)	99.9 (± 0)	95.3 (± 0.4)	6.6 (± 0.1)	3.9 (± 0)	5.1 (± 0)
Beaver	94.0 (± 0.6)	74.5 (± 0.2)	82.5 (± 0.7)	3.6 (± 0.1)	2.4 (± 0)	2.5 (± 0)
Bicycle	94.2 (± 0.5)	96.9 (± 0.1)	93.6 (± 0.5)	3.2 (± 0)	2.7 (± 0)	2.5 (± 0)
Bulbul	93.8 (± 0.6)	88.3 (± 0.2)	88.4 (± 0.4)	3.3 (± 0)	1.9 (± 0)	2.2 (± 0)
Coral fungus	93.5 (± 0.6)	98.4 (± 0)	93.4 (± 0.6)	2.9 (± 0)	2.0 (± 0)	2.2 (± 0)
English springer	93.9 (± 0.6)	93.3 (± 0.1)	91.3 (± 0.4)	3.6 (± 0)	2.5 (± 0)	2.5 (± 0)
Garfish	93.5 (± 0.6)	77.2 (± 0.2)	82.7 (± 0.6)	5.4 (± 0.1)	3.3 (± 0)	4.1 (± 0)
Golden retriever	94.1 (± 0.6)	84.0 (± 0.2)	86.5 (± 0.5)	5.0 (± 0.1)	3.0 (± 0)	3.3 (± 0)
Gyromitra	95.0 (± 0.6)	38.8 (± 0.2)	84.8 (± 1.0)	3.3 (± 0)	2.4 (± 0)	2.6 (± 0)
Jay	93.3 (± 0.7)	23.5 (± 0.2)	80.5 (± 1.0)	6.8 (± 0.1)	4.6 (± 0)	5.1 (± 0)
Junco, snowbird	94.2 (± 0.5)	95.3 (± 0.1)	92.9 (± 0.4)	2.7 (± 0)	1.6 (± 0)	1.8 (± 0)
Kuvasz	93.4 (± 0.6)	99.0 (± 0)	93.3 (± 0.6)	3.5 (± 0)	1.8 (± 0)	2.1 (± 0)
Labrador retriever	93.5 (± 0.7)	81.5 (± 0.2)	84.5 (± 0.5)	5.4 (± 0.1)	3.3 (± 0)	3.7 (± 0)
Lighter, Light	94.2 (± 0.6)	57.8 (± 0.2)	81.5 (± 0.9)	5.4 (± 0.1)	3.1 (± 0)	3.8 (± 0)
Lycaenid butterfly	94.0 (± 0.5)	82.2 (± 0.2)	86.2 (± 0.5)	3.9 (± 0.1)	3.0 (± 0)	3.0 (± 0)
Magpie	93.5 (± 0.6)	88.9 (± 0.2)	88.7 (± 0.5)	3.0 (± 0)	1.8 (± 0)	2.0 (± 0)
Marmot	94.0 (± 0.6)	96.0 (± 0.1)	92.9 (± 0.6)	4.9 (± 0.1)	3.1 (± 0)	3.3 (± 0)
Muzzle	93.5 (± 0.6)	90.5 (± 0.1)	90.0 (± 0.5)	4.3 (± 0)	2.6 (± 0)	3.1 (± 0)
Papillon	93.8 (± 0.6)	99.7 (± 0)	93.8 (± 0.6)	3.8 (± 0.1)	1.9 (± 0)	2.5 (± 0)
Rock beauty	94.2 (± 0.5)	80.3 (± 0.2)	84.4 (± 0.5)	5.1 (± 0.1)	3.7 (± 0)	4.1 (± 0)
Siberian husky	94.1 (± 0.6)	53.1 (± 0.2)	80.6 (± 1.0)	4.0 (± 0)	2.2 (± 0)	2.6 (± 0)
Stinkhorn	93.4 (± 0.6)	96.0 (± 0.1)	92.7 (± 0.5)	4.5 (± 0.1)	3.5 (± 0)	3.4 (± 0)
Tennis ball	93.5 (± 0.6)	77.2 (± 0.2)	82.7 (± 0.6)	3.5 (± 0)	2.1 (± 0)	2.5 (± 0)
Tinca tinca	93.2 (± 0.6)	97.5 (± 0.1)	93.0 (± 0.6)	3.8 (± 0)	2.2 (± 0)	2.8 (± 0)
Torch	94.8 (± 0.5)	79.5 (± 0.2)	84.5 (± 0.5)	6.2 (± 0.1)	4.4 (± 0)	4.8 (± 0)
Unicycle	93.3 (± 0.6)	99.7 (± 0)	93.3 (± 0.6)	4.8 (± 0)	3.4 (± 0)	3.5 (± 0)
Water ouzel	94.3 (± 0.5)	98.8 (± 0)	94.2 (± 0.5)	3.0 (± 0)	1.6 (± 0)	2.0 (± 0)
White wolf	93.9 (± 0.6)	72.8 (± 0.2)	82.2 (± 0.8)	3.4 (± 0)	2.2 (± 0)	2.3 (± 0)

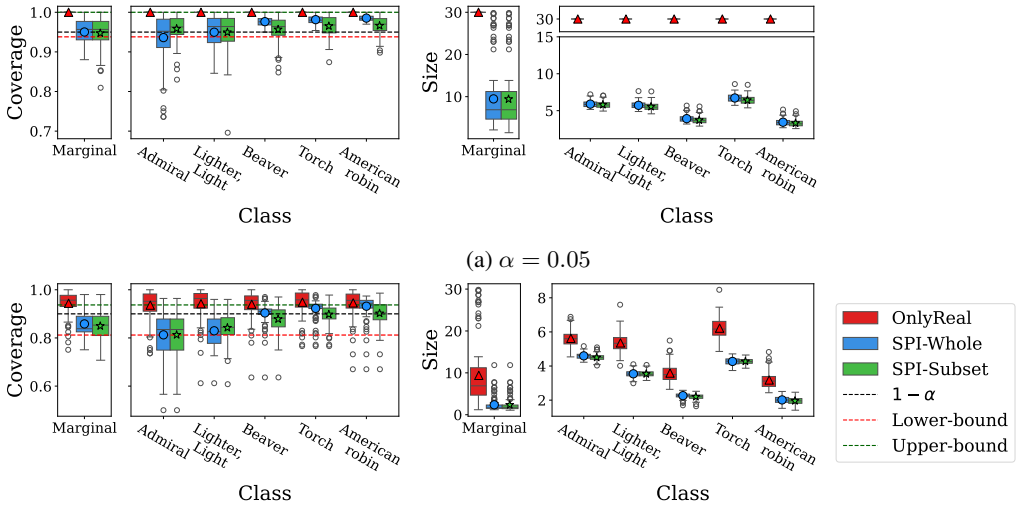
J.2 Experiments with auxiliary labeled data

In this section, we follow the experimental setup described in Section 4.1.2, where the synthetic data comprise 100 classes, none of which are included in the real calibration set.

Figure S13 presents the results for both marginal and label-conditional guarantees at levels $\alpha = 0.05$ and 0.1, demonstrating trends similar to those observed in Figure 4. The standard conformal prediction, `OnlyReal`, conservatively controls coverage at the target level $1 - \alpha$, but results in larger and noisier prediction sets due to the limited sample size. In contrast, both `SPI-Whole` and `SPI-Subset` substantially reduce the size and variance of the prediction sets and, as expected, achieve coverage within the theoretical bounds.

Notably, for the "American robin" and "Torch" classes, the `SPI-Subset` variant achieves coverage more tightly aligned with the target level $1 - \alpha$, outperforming the standard `SPI-Whole` method.

We include the results for all real classes in Tables S8 to S10, corresponding to Figures 4, S13a and S13b, respectively.



(a) $\alpha = 0.05$

(b) $\alpha = 0.1$

Figure S13: Results for the ImageNet data: Coverage rates of `OnlyReal`, `SPI-Whole`, and `SPI-Subset` run at level $\alpha = 0.05$ (a) and 0.1 (b), averaged over 100 trials. Left: Average coverage. Right: Average prediction set size, both under marginal (leftmost box in each group) and label-conditional coverage settings. Label-conditional results are shown for selected classes; see Tables S9 and S10 for results across all classes.

Table S8: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials. Standard errors are shown in parentheses. The target coverage level is $1 - \alpha = 0.98$. The theoretical coverage guarantees for both `SPI-Whole` and `SPI-Subset` are in the range $[93.7, 100]$. Other details are as in Figure 4.

	Coverage (%)			Size		
Class	`Only Real`	`SPI Whole`	`SPI Subset`	`Only Real`	`SPI Whole`	`SPI Subset`
Admiral	100 (± 0)	93.6 (± 0.6)	99.8 (± 0)	30 (± 0)	7.6 (± 0.1)	6.5 (± 0)
American robin	100 (± 0)	99.9 (± 0)	98.7 (± 0.1)	30 (± 0)	5.0 (± 0.1)	4.4 (± 0)
Barracouta	100 (± 0)	98.4 (± 0.1)	98.0 (± 0.2)	30 (± 0)	9.8 (± 0.1)	7.8 (± 0.1)
Beaver	100 (± 0)	99.5 (± 0)	98.1 (± 0.2)	30 (± 0)	6.1 (± 0.1)	4.9 (± 0)
Bicycle	100 (± 0)	100 (± 0)	99.2 (± 0.1)	30 (± 0)	4.7 (± 0)	4.0 (± 0)
Bulbul	100 (± 0)	99.9 (± 0)	99.0 (± 0.1)	30 (± 0)	5.2 (± 0)	4.3 (± 0)
Coral fungus	100 (± 0)	99.7 (± 0)	98.2 (± 0.1)	30 (± 0)	4.1 (± 0)	3.7 (± 0)
English springer	100 (± 0)	99.8 (± 0)	98.8 (± 0.1)	30 (± 0)	5.7 (± 0.1)	4.7 (± 0)
Garfish	100 (± 0)	99.4 (± 0)	97.8 (± 0.2)	30 (± 0)	8.0 (± 0.1)	6.6 (± 0)
Golden retriever	100 (± 0)	99.6 (± 0)	98.0 (± 0.2)	30 (± 0)	7.9 (± 0.1)	6.5 (± 0.1)
Gyromitra	100 (± 0)	95.0 (± 0.6)	95.0 (± 0.6)	30 (± 0)	4.6 (± 0)	4.3 (± 0)
Jay	100 (± 0)	98.3 (± 0.1)	98.1 (± 0.1)	30 (± 0)	11.4 (± 0.1)	8.7 (± 0.1)
Junco, snowbird	100 (± 0)	99.9 (± 0)	98.7 (± 0.1)	30 (± 0)	4.3 (± 0)	3.5 (± 0)
Kuvasz	100 (± 0)	99.5 (± 0)	98.5 (± 0.1)	30 (± 0)	5.7 (± 0)	4.7 (± 0.1)
Labrador retriever	100 (± 0)	99.3 (± 0)	97.6 (± 0.2)	30 (± 0)	8.7 (± 0.1)	7.0 (± 0.1)
Lighter, Light	100 (± 0)	97.8 (± 0.1)	96.9 (± 0.2)	30 (± 0)	8.4 (± 0.1)	6.6 (± 0.1)
Lycaenid butterfly	100 (± 0)	99.8 (± 0)	99.1 (± 0.1)	30 (± 0)	5.2 (± 0)	5.2 (± 0)
Magpie	100 (± 0)	99.7 (± 0)	98.6 (± 0.1)	30 (± 0)	4.7 (± 0)	3.9 (± 0)
Marmot	100 (± 0)	99.7 (± 0)	98.5 (± 0.1)	30 (± 0)	7.5 (± 0.1)	6.4 (± 0.1)
Muzzle	100 (± 0)	98.8 (± 0.1)	96.6 (± 0.3)	30 (± 0)	6.7 (± 0.1)	5.4 (± 0)
Papillon	100 (± 0)	99.2 (± 0)	97.6 (± 0.2)	30 (± 0)	5.7 (± 0.1)	5.1 (± 0)
Rock beauty	100 (± 0)	98.9 (± 0.1)	98.5 (± 0.1)	30 (± 0)	8.3 (± 0.1)	6.2 (± 0.1)
Siberian husky	100 (± 0)	99.3 (± 0)	98.8 (± 0.1)	30 (± 0)	6.3 (± 0.1)	5.2 (± 0)
Stinkhorn	100 (± 0)	99.8 (± 0)	98.3 (± 0.1)	30 (± 0)	7.2 (± 0.1)	6.0 (± 0)
Tennis ball	100 (± 0)	98.6 (± 0.1)	96.6 (± 0.3)	30 (± 0)	5.3 (± 0)	4.5 (± 0)
Tinca tinca	100 (± 0)	99.8 (± 0)	98.8 (± 0.1)	30 (± 0)	5.3 (± 0)	4.7 (± 0)
Torch	100 (± 0)	99.6 (± 0)	98.4 (± 0.1)	30 (± 0)	10.3 (± 0.1)	7.6 (± 0.1)
Unicycle	100 (± 0)	99.7 (± 0)	98.2 (± 0.1)	30 (± 0)	8.1 (± 0.1)	6.1 (± 0.1)
Water ouzel	100 (± 0)	99.2 (± 0)	98.0 (± 0.1)	30 (± 0)	5.0 (± 0)	4.0 (± 0)
White wolf	100 (± 0)	99.1 (± 0)	97.1 (± 0.2)	30 (± 0)	5.5 (± 0)	4.6 (± 0.1)

Table S9: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials. Standard errors are shown in parentheses. The target coverage level is $1 - \alpha = 0.95$. The theoretical coverage guarantees for both `SPI-Whole` and `SPI-Subset` are in the range $[93.7, 100]$. Other details are as in Figure S13.

	Coverage (%)			Size		
Class	`Only Real`	`SPI Whole`	`SPI Subset`	`Only Real`	`SPI Whole`	`SPI Subset`
Admiral	100 (± 0)	93.6 (± 0.6)	95.9 (± 0.3)	30 (± 0)	5.9 (± 0)	5.8 (± 0)
American robin	100 (± 0)	98.5 (± 0.1)	96.6 (± 0.2)	30 (± 0)	3.4 (± 0)	3.3 (± 0)
Barracouta	100 (± 0)	95.3 (± 0.4)	95.8 (± 0.3)	30 (± 0)	7.1 (± 0.1)	6.9 (± 0.1)
Beaver	100 (± 0)	97.6 (± 0.1)	95.7 (± 0.3)	30 (± 0)	3.9 (± 0)	3.7 (± 0)
Bicycle	100 (± 0)	99.3 (± 0)	97.2 (± 0.2)	30 (± 0)	3.4 (± 0)	3.3 (± 0)
Bulbul	100 (± 0)	99.0 (± 0)	96.8 (± 0.2)	30 (± 0)	3.5 (± 0)	3.4 (± 0)
Coral fungus	100 (± 0)	98.1 (± 0.1)	96.0 (± 0.3)	30 (± 0)	3.1 (± 0)	3.0 (± 0)
English springer	100 (± 0)	98.9 (± 0.1)	97.3 (± 0.2)	30 (± 0)	3.9 (± 0)	3.8 (± 0)
Garfish	100 (± 0)	96.9 (± 0.2)	95.2 (± 0.4)	30 (± 0)	5.8 (± 0)	5.6 (± 0)
Golden retriever	100 (± 0)	97.7 (± 0.1)	95.8 (± 0.3)	30 (± 0)	5.5 (± 0.1)	5.3 (± 0.1)
Gyromitra	100 (± 0)	95.0 (± 0.6)	95.0 (± 0.6)	30 (± 0)	3.6 (± 0)	3.5 (± 0)
Jay	100 (± 0)	94.0 (± 0.5)	95.3 (± 0.3)	30 (± 0)	7.9 (± 0.1)	7.2 (± 0.1)
Junco, snowbird	100 (± 0)	98.8 (± 0.1)	96.6 (± 0.2)	30 (± 0)	3.0 (± 0)	2.8 (± 0)
Kuvasz	100 (± 0)	98.5 (± 0.1)	96.5 (± 0.2)	30 (± 0)	3.9 (± 0)	3.7 (± 0)
Labrador retriever	100 (± 0)	97.1 (± 0.1)	95.3 (± 0.4)	30 (± 0)	6.0 (± 0.1)	5.7 (± 0.1)
Lighter, Light	100 (± 0)	95.0 (± 0.4)	95.0 (± 0.5)	30 (± 0)	5.7 (± 0)	5.5 (± 0.1)
Lycaenid butterfly	100 (± 0)	99.2 (± 0)	96.8 (± 0.2)	30 (± 0)	4.2 (± 0)	4.1 (± 0.1)
Magpie	100 (± 0)	98.6 (± 0.1)	96.4 (± 0.2)	30 (± 0)	3.2 (± 0)	3.1 (± 0)
Marmot	100 (± 0)	98.1 (± 0.1)	96.4 (± 0.3)	30 (± 0)	5.1 (± 0.1)	5.1 (± 0.1)
Muzzle	100 (± 0)	95.0 (± 0.3)	94.3 (± 0.5)	30 (± 0)	4.6 (± 0)	4.5 (± 0)
Papillon	100 (± 0)	97.4 (± 0.1)	95.3 (± 0.3)	30 (± 0)	4.0 (± 0)	4.0 (± 0)
Rock beauty	100 (± 0)	94.5 (± 0.4)	95.6 (± 0.3)	30 (± 0)	5.6 (± 0)	5.3 (± 0)
Siberian husky	100 (± 0)	96.5 (± 0.2)	97.0 (± 0.2)	30 (± 0)	4.3 (± 0)	4.1 (± 0)
Stinkhorn	100 (± 0)	98.1 (± 0.1)	95.9 (± 0.2)	30 (± 0)	5.0 (± 0)	4.8 (± 0)
Tennis ball	100 (± 0)	95.7 (± 0.3)	94.6 (± 0.4)	30 (± 0)	3.8 (± 0)	3.6 (± 0)
Tinca tinca	100 (± 0)	98.8 (± 0.1)	96.6 (± 0.2)	30 (± 0)	4.0 (± 0)	3.9 (± 0)
Torch	100 (± 0)	98.1 (± 0.1)	96.5 (± 0.3)	30 (± 0)	6.7 (± 0.1)	6.4 (± 0.1)
Unicycle	100 (± 0)	97.9 (± 0.1)	95.6 (± 0.4)	30 (± 0)	5.3 (± 0)	5.0 (± 0)
Water ouzel	100 (± 0)	98.0 (± 0.1)	96.2 (± 0.3)	30 (± 0)	3.3 (± 0)	3.1 (± 0)
White wolf	100 (± 0)	96.6 (± 0.2)	95.1 (± 0.4)	30 (± 0)	3.7 (± 0)	3.6 (± 0)

Table S10: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials. Standard errors are shown in parentheses. The target coverage level is $1-\alpha = 0.9$. The theoretical coverage guarantee for both `SPI-Whole` and `SPI-Subset` are in the range $[81.2, 93.7]$. Other details are as in Figure S13.

	Coverage (%)			Size		
Class	`Only Real`	`SPI Whole`	`SPI Subset`	`Only Real`	`SPI Whole`	`SPI Subset`
Admiral	93.6 (± 0.6)	81.3 (± 0.9)	81.3 (± 0.9)	5.6 (± 0)	4.6 (± 0)	4.5 (± 0)
American robin	94.5 (± 0.5)	93.2 (± 0.5)	90.2 (± 0.5)	3.2 (± 0)	2.0 (± 0)	2.0 (± 0)
Barracouta	95.3 (± 0.4)	83.4 (± 0.8)	85.1 (± 0.6)	6.6 (± 0.1)	5.0 (± 0)	4.9 (± 0)
Beaver	94.0 (± 0.6)	90.4 (± 0.4)	87.8 (± 0.5)	3.6 (± 0.1)	2.3 (± 0)	2.2 (± 0)
Bicycle	94.2 (± 0.5)	93.5 (± 0.5)	90.9 (± 0.4)	3.2 (± 0)	2.3 (± 0)	2.3 (± 0)
Bulbul	93.8 (± 0.6)	93.0 (± 0.6)	90.0 (± 0.6)	3.3 (± 0)	2.1 (± 0)	2.0 (± 0)
Coral fungus	93.5 (± 0.6)	91.7 (± 0.5)	88.8 (± 0.5)	2.9 (± 0)	2.1 (± 0)	2.0 (± 0)
English springer	93.9 (± 0.6)	93.0 (± 0.5)	90.4 (± 0.5)	3.6 (± 0)	2.2 (± 0)	2.2 (± 0)
Garfish	93.5 (± 0.6)	88.5 (± 0.4)	86.8 (± 0.5)	5.4 (± 0.1)	3.9 (± 0)	3.9 (± 0)
Golden retriever	94.1 (± 0.6)	91.7 (± 0.4)	88.8 (± 0.5)	5.0 (± 0.1)	3.2 (± 0)	3.1 (± 0)
Gyromitra	95.0 (± 0.6)	84.8 (± 1.0)	84.8 (± 1.0)	3.3 (± 0)	2.5 (± 0)	2.4 (± 0)
Jay	93.3 (± 0.7)	80.5 (± 1.0)	83.7 (± 0.6)	6.8 (± 0.1)	4.8 (± 0)	4.6 (± 0)
Junco, snowbird	94.2 (± 0.5)	93.0 (± 0.4)	90.6 (± 0.5)	2.7 (± 0)	1.8 (± 0)	1.7 (± 0)
Kuvasz	93.4 (± 0.6)	92.3 (± 0.5)	90.1 (± 0.5)	3.5 (± 0)	2.1 (± 0)	2.1 (± 0)
Labrador retriever	93.5 (± 0.7)	88.5 (± 0.5)	87.0 (± 0.6)	5.4 (± 0.1)	3.4 (± 0)	3.4 (± 0)
Lighter, Light	94.2 (± 0.6)	83.0 (± 0.7)	84.2 (± 0.6)	5.4 (± 0.1)	3.5 (± 0)	3.5 (± 0)
Lycaenid butterfly	94.0 (± 0.5)	93.3 (± 0.4)	90.7 (± 0.5)	3.9 (± 0.1)	2.7 (± 0)	2.7 (± 0)
Magpie	93.5 (± 0.6)	92.4 (± 0.6)	90.1 (± 0.5)	3.0 (± 0)	1.9 (± 0)	1.9 (± 0)
Marmot	94.0 (± 0.6)	90.9 (± 0.5)	88.3 (± 0.6)	4.9 (± 0.1)	3.1 (± 0)	3.0 (± 0)
Muzzle	93.5 (± 0.6)	86.1 (± 0.6)	85.9 (± 0.6)	4.3 (± 0)	2.8 (± 0)	2.8 (± 0)
Papillon	93.8 (± 0.6)	90.4 (± 0.4)	87.9 (± 0.5)	3.8 (± 0.1)	2.4 (± 0)	2.4 (± 0)
Rock beauty	94.2 (± 0.5)	80.9 (± 0.9)	84.6 (± 0.5)	5.1 (± 0.1)	3.7 (± 0)	3.7 (± 0)
Siberian husky	94.1 (± 0.6)	84.3 (± 0.5)	89.1 (± 0.5)	4.0 (± 0)	2.3 (± 0)	2.4 (± 0)
Stinkhorn	93.4 (± 0.6)	91.9 (± 0.5)	89.3 (± 0.4)	4.5 (± 0.1)	3.0 (± 0)	3.0 (± 0)
Tennis ball	93.5 (± 0.6)	87.7 (± 0.3)	85.8 (± 0.5)	3.5 (± 0)	2.5 (± 0)	2.4 (± 0)
Tinca tinca	93.2 (± 0.6)	92.6 (± 0.5)	90.3 (± 0.5)	3.8 (± 0)	2.6 (± 0)	2.6 (± 0)
Torch	94.8 (± 0.5)	92.3 (± 0.4)	89.8 (± 0.4)	6.2 (± 0.1)	4.3 (± 0)	4.3 (± 0)
Unicycle	93.3 (± 0.6)	90.3 (± 0.5)	88.2 (± 0.6)	4.8 (± 0)	3.3 (± 0)	3.2 (± 0)
Water ouzel	94.3 (± 0.5)	92.4 (± 0.4)	89.9 (± 0.4)	3.0 (± 0)	2.0 (± 0)	1.9 (± 0)
White wolf	93.9 (± 0.6)	89.2 (± 0.4)	87.3 (± 0.5)	3.4 (± 0)	2.1 (± 0)	2.1 (± 0)

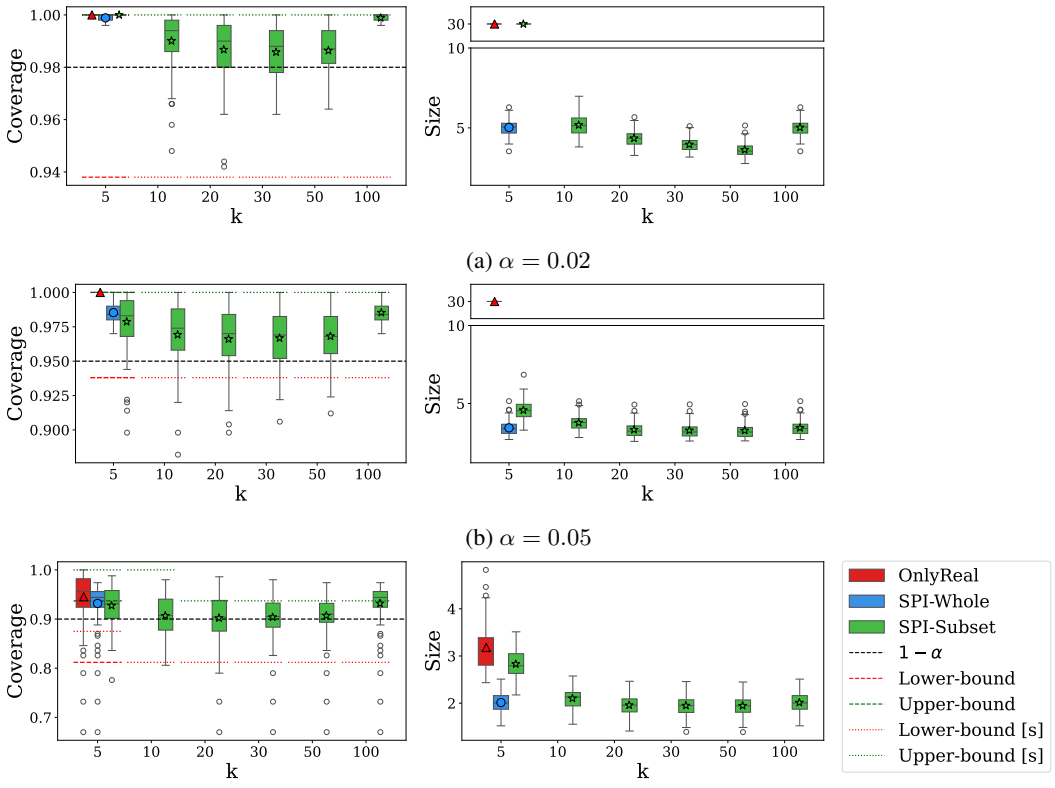
J.2.1 Results for `SPI-Subset` with different hyperparameter values

In this section, we present results for the `SPI-Subset` procedure across different values of k, the number of subsets selected to construct the synthetic calibration set. We compare the performance of `SPI-Subset` with `SPI-Whole`—which uses all 100 synthetic classes—and the standard conformal prediction, `OnlyReal`.

Figure S14 presents the performance of all methods for the "American robin" class as a function of k, at different values of the level α. Notably, for all values of k and α, `SPI-Subset` achieves coverage within the theoretical bounds.

The two methods, `SPI-Subset` and `SPI-Whole`, coincide when $k = 100$, as both use the full synthetic calibration set. However, for smaller values of k, the two methods exhibit significant differences. While `SPI-Whole` tends to produce more conservative prediction sets, the `SPI-Subset` procedure more tightly achieves the target coverage level across different settings.

For the case $\alpha = 0.02$ and $k = 5$, both the theoretical lower and upper bounds on coverage are equal to unity, implying that `SPI-Subset` yields trivial prediction sets that include all possible classes. This outcome is known a priori and can be avoided by selecting a different hyperparameter for window construction.



(a) $\alpha = 0.02$

(b) $\alpha = 0.05$

(c) $\alpha = 0.1$

Figure S14: Results for the ImageNet data: Coverage rate for `OnlyReal`,`SPI-Whole`, and `SPI-Subset` on the American robin class as a function of the number of subsets k, for levels $\alpha = 0.02$ (a), $\alpha = 0.05$ (b), and $\alpha = 0.1$ (c).

Figure S15 shows the results for the "Beaver" class. For $\alpha = 0.02$ and 0.05, we observe the same trend as in Figure S14: `SPI-Whole` yields relatively conservative coverage, while `SPI-Subset` with $k < 100$ achieves coverage closer to the nominal level $1 - \alpha$.

For $\alpha = 0.1$, `SPI-Whole`—which uses the full synthetic set—already achieves coverage close to the target level $1 - \alpha$, suggesting that the empirical $(1 - \alpha)$th quantile of the synthetic data closely

matches that of the real data. Consequently, in this setting, using only a subset of the synthetic data results in an increase in the variance of the coverage rate.

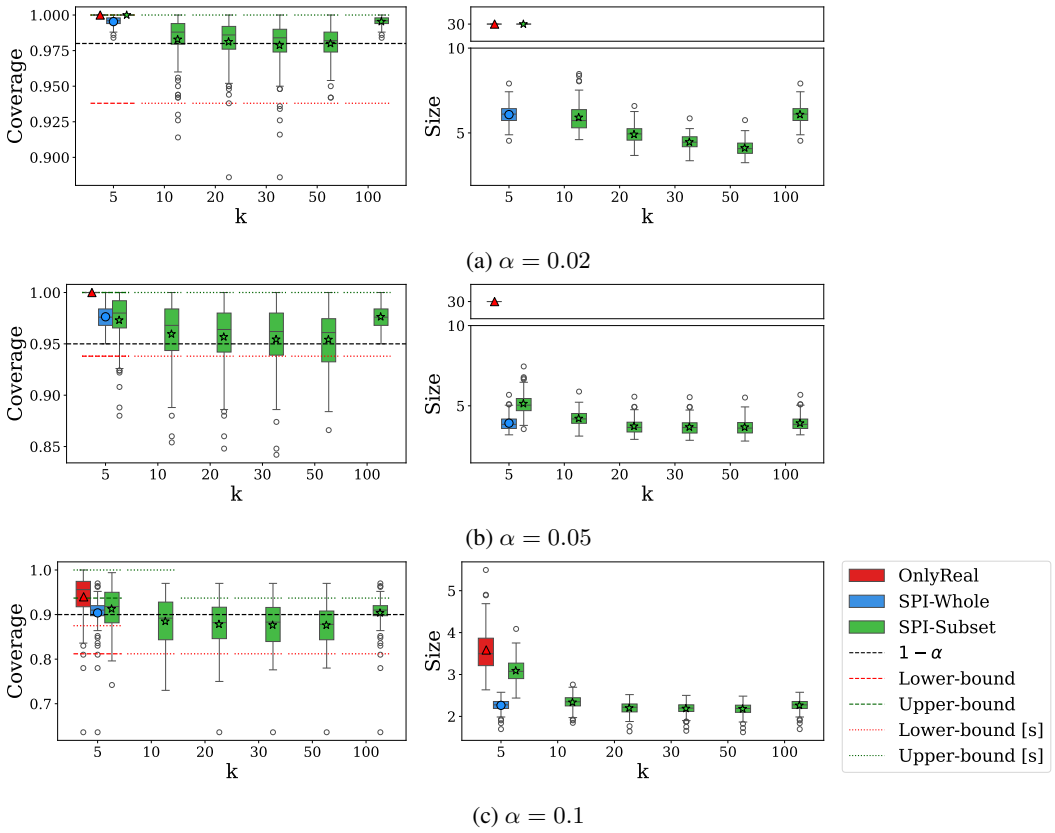


Figure S15: Results for the ImageNet data: Coverage rate for `OnlyReal`, `SPI-Whole`, and `SPI-Subset` on the Beaver class as a function of the number of subsets k, for levels $\alpha = 0.02$ (a), $\alpha = 0.05$ (b), and $\alpha = 0.1$ (c).

J.3 Additional ImageNet experiments with the HPS score function

Here, we replicate the ImageNet experiments using the HPS score function—complementing the APS-based results in the manuscript—across three synthetic datasets: Stable-Diffusion (Figure S16), FLUX (Figure S17), and auxiliary data (Figure S18). Each figure corresponds to the same experimental setup used with the APS score function. Further details on the score function are provided in Appendix C.1.

The score function provides a heuristic measure of the model's uncertainty, and APS and HPS capture this uncertainty in different ways. Recall that we utilize the synthetic data in the score space, where the quality of the `SPI` prediction set depends on how well the distribution of the synthetic score approximates the real score distribution. Consequently, different score functions may induce different alignments between the real and synthetic scores. Therefore, while we generally expect similar trends, the exact per-class coverage is not necessarily preserved across score functions. Indeed, in Figure S16, the per-class results largely follow those of the APS-based experiment (Figure 3). In contrast, in Figure S17, for the class Magpie, `OnlySynth` under-covers substantially, whereas under the APS score in Figure S12b, it achieves coverage closer to the target level $1 - \alpha$.

Given these class-specific differences, the overall trends remain similar to the APS-based results. The `OnlyReal` method yields overly conservative and uninformative prediction sets due to the limited sample size, whereas `OnlySynth` lacks coverage guarantees, leading to under-coverage for some classes. In contrast, `SPI` achieves coverage within the theoretical bounds while producing smaller, more informative prediction sets.

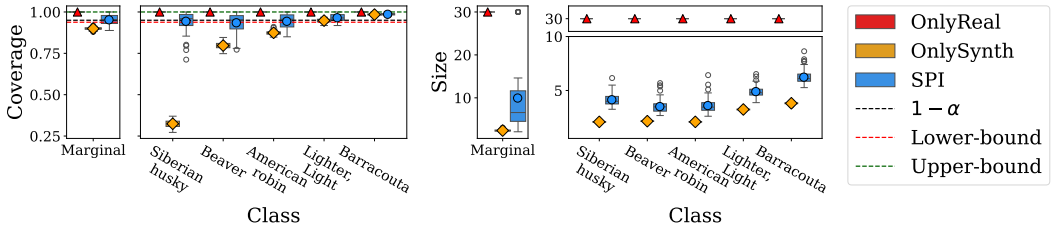


Figure S16: Results for the ImageNet data using the HPS score function: Same experimental setup as in Figure 3, but using the HPS score instead of APS. Coverage rates of `OnlyReal`, `OnlySynth`, and `SPI` at target level $1 - \alpha = 0.95$, averaged over 100 trials. Left: Average coverage. Right: Average prediction set size, both under marginal (leftmost box in each group) and label-conditional coverage settings. Label-conditional results are shown for selected classes; see Table S11 for results across all classes.

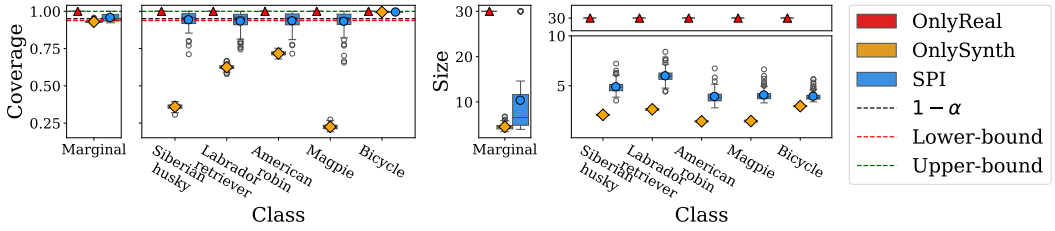


Figure S17: Results for the ImageNet data using FLUX-generated synthetic images and the HPS score function: Same experimental setup as in Figure S12b, but using the HPS score instead of APS. Coverage rates of `OnlyReal`, `OnlySynth`, and `SPI` at target level $1 - \alpha = 0.95$, averaged over 100 trials. Left: Average coverage. Right: Average prediction set size, both under marginal (leftmost box in each group) and label-conditional coverage settings. Label-conditional results are shown for selected classes; see Table S12 for results across all classes.

Finally, Figure S18 presents the performance of the subset-based variant of our approach, `SPI-Subset`, compared to `SPI-Whole`, which uses the entire synthetic dataset. We observe a similar trend to the APS-based results in Figure 4: both SPI variants control the coverage within the theoretical bounds. For some classes, `SPI-Subset` achieves coverage closer to the nominal $1 - \alpha$ level while producing smaller prediction sets. In the marginal setting, however, using only a subset of the synthetic data does not improve the performance.

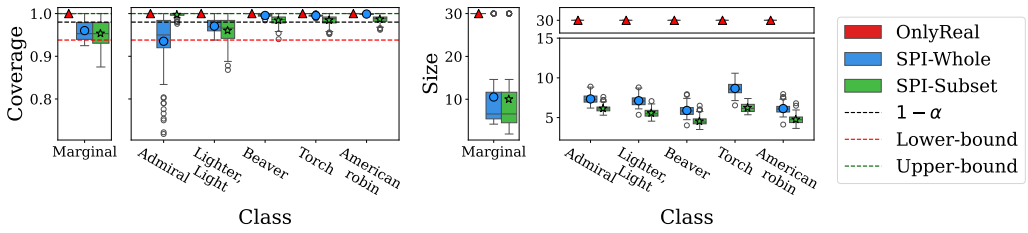


Figure S18: Results for the ImageNet data using the HPS score function: Same experimental setup as in Figure 4, but using the HPS score instead of APS. Coverage rates of `OnlyReal`, `SPI-Whole`, and `SPI-Subset` at target level $1 - \alpha = 0.98$, averaged over 100 trials. Left: Average coverage. Right: Average prediction set size, both under marginal (leftmost box in each group) and label-conditional coverage settings. Label-conditional results are shown for selected classes; see Table S13 for results across all classes.

Table S11: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials using the HPS score function. Standard errors are shown in parentheses. The target coverage level is $1 - \alpha = 0.95$. The theoretical coverage guarantees for `SPI` are in the range $[93.7, 100]$. Other details are as in Figure S16.

	Coverage (%)			Size		
Class	`Only Real`	`Only Synth`	`SPI`	`Only Real`	`Only Synth`	`SPI`
Admiral	100 (± 0)	0.7 (± 0)	93.5 (± 0.7)	30 (± 0)	3.2 (± 0)	5.5 (± 0.1)
American robin	100 (± 0)	87.4 (± 0.1)	94.4 (± 0.4)	30 (± 0)	2.1 (± 0)	3.6 (± 0.1)
Barracouta	100 (± 0)	98.4 (± 0)	98.6 (± 0.1)	30 (± 0)	3.8 (± 0)	6.2 (± 0.1)
Beaver	100 (± 0)	79.7 (± 0.2)	93.5 (± 0.6)	30 (± 0)	2.2 (± 0)	3.5 (± 0.1)
Bicycle	100 (± 0)	91.2 (± 0.1)	95.4 (± 0.3)	30 (± 0)	2.1 (± 0)	3.1 (± 0)
Bulbul	100 (± 0)	98.9 (± 0)	99.0 (± 0)	30 (± 0)	2.2 (± 0)	3.7 (± 0.1)
Coral fungus	100 (± 0)	98.0 (± 0)	98.3 (± 0.1)	30 (± 0)	2.6 (± 0)	3.6 (± 0)
English springer	100 (± 0)	77.0 (± 0.1)	93.7 (± 0.6)	30 (± 0)	2.3 (± 0)	3.8 (± 0.1)
Garfish	100 (± 0)	71.3 (± 0.2)	93.2 (± 0.7)	30 (± 0)	3.3 (± 0)	5.2 (± 0.1)
Golden retriever	100 (± 0)	70.8 (± 0.4)	94.2 (± 0.6)	30 (± 0)	3.0 (± 0)	5.1 (± 0.1)
Gyromitra	100 (± 0)	100 (± 0)	100 (± 0)	30 (± 0)	2.9 (± 0)	3.8 (± 0)
Jay	100 (± 0)	72.3 (± 0.2)	93.1 (± 0.6)	30 (± 0)	3.0 (± 0)	6.0 (± 0.1)
Junco, snowbird	100 (± 0)	94.0 (± 0.1)	95.9 (± 0.2)	30 (± 0)	2.2 (± 0)	3.2 (± 0.1)
Kuvasz	100 (± 0)	81.9 (± 0.1)	93.1 (± 0.6)	30 (± 0)	2.2 (± 0)	3.9 (± 0.1)
Labrador retriever	100 (± 0)	87.1 (± 0.2)	94.5 (± 0.5)	30 (± 0)	2.9 (± 0)	5.3 (± 0.1)
Lighter, Light	100 (± 0)	94.8 (± 0.1)	96.5 (± 0.2)	30 (± 0)	3.2 (± 0)	4.9 (± 0)
Lycaenid butterfly	100 (± 0)	82.0 (± 0.2)	94.6 (± 0.5)	30 (± 0)	2.1 (± 0)	4.3 (± 0.1)
Magpie	100 (± 0)	69.5 (± 0.3)	93.3 (± 0.7)	30 (± 0)	2.0 (± 0)	3.3 (± 0.1)
Marmot	100 (± 0)	88.4 (± 0.1)	95.5 (± 0.4)	30 (± 0)	2.6 (± 0)	4.6 (± 0.1)
Muzzle	100 (± 0)	97.1 (± 0.1)	97.7 (± 0.1)	30 (± 0)	2.3 (± 0)	4.1 (± 0.1)
Papillon	100 (± 0)	73.1 (± 0.2)	93.9 (± 0.6)	30 (± 0)	2.2 (± 0)	3.9 (± 0.1)
Rock beauty	100 (± 0)	51.8 (± 0.4)	94.3 (± 0.5)	30 (± 0)	2.7 (± 0)	5.0 (± 0.1)
Siberian husky	100 (± 0)	32.4 (± 0.2)	94.4 (± 0.5)	30 (± 0)	2.1 (± 0)	4.1 (± 0.1)
Stinkhorn	100 (± 0)	95.1 (± 0.1)	96.5 (± 0.2)	30 (± 0)	3.0 (± 0)	4.5 (± 0.1)
Tennis ball	100 (± 0)	92.4 (± 0.2)	95.4 (± 0.3)	30 (± 0)	2.4 (± 0)	3.4 (± 0)
Tinca tinca	100 (± 0)	97.3 (± 0.1)	97.8 (± 0.1)	30 (± 0)	2.6 (± 0)	3.9 (± 0.1)
Torch	100 (± 0)	95.6 (± 0.1)	97.0 (± 0.2)	30 (± 0)	3.5 (± 0)	5.5 (± 0.1)
Unicycle	100 (± 0)	86.6 (± 0.1)	94.0 (± 0.5)	30 (± 0)	2.4 (± 0)	4.1 (± 0.1)
Water ouzel	100 (± 0)	91.3 (± 0.1)	95.2 (± 0.3)	30 (± 0)	2.1 (± 0)	3.3 (± 0.1)
White wolf	100 (± 0)	60.3 (± 0.2)	93.7 (± 0.6)	30 (± 0)	2.0 (± 0)	3.7 (± 0.1)

Table S12: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials using FLUX-generated synthetic data and the HPS score function. The target coverage level is $1 - \alpha = 0.95$. The theoretical coverage guarantees for `SPI` are in the range $[93.7, 100]$. Standard errors are shown in parentheses. Other experimental details follow Figure S17.

	Coverage (%)			Size		
Class	`Only Real`	`Only Synth`	`SPI`	`Only Real`	`Only Synth`	`SPI`
Admiral	100 (± 0)	0.1 (± 0)	93.5 (± 0.7)	30 (± 0)	3.5 (± 0)	5.6 (± 0.1)
American robin	100 (± 0)	71.7 (± 0.1)	93.7 (± 0.6)	30 (± 0)	1.4 (± 0)	3.9 (± 0.1)
Barracouta	100 (± 0)	99.6 (± 0)	99.6 (± 0)	30 (± 0)	3.7 (± 0)	7.0 (± 0.1)
Beaver	100 (± 0)	35.8 (± 0.2)	93.4 (± 0.6)	30 (± 0)	1.9 (± 0)	4.2 (± 0.1)
Bicycle	100 (± 0)	99.5 (± 0)	99.6 (± 0)	30 (± 0)	3.0 (± 0)	4.0 (± 0)
Bulbul	100 (± 0)	91.3 (± 0.1)	95.2 (± 0.3)	30 (± 0)	2.3 (± 0)	4.5 (± 0.1)
Coral fungus	100 (± 0)	98.6 (± 0)	98.7 (± 0)	30 (± 0)	2.3 (± 0)	4.0 (± 0)
English springer	100 (± 0)	82.4 (± 0.1)	93.9 (± 0.6)	30 (± 0)	2.4 (± 0)	4.6 (± 0.1)
Garfish	100 (± 0)	53.9 (± 0.2)	93.2 (± 0.7)	30 (± 0)	3.1 (± 0)	5.7 (± 0.1)
Golden retriever	100 (± 0)	10.9 (± 0.1)	94.2 (± 0.6)	30 (± 0)	2.1 (± 0)	6.0 (± 0.1)
Gyromitra	100 (± 0)	42.5 (± 0.2)	99.7 (± 0.3)	30 (± 0)	2.8 (± 0)	4.2 (± 0)
Jay	100 (± 0)	28.4 (± 0.2)	93.0 (± 0.7)	30 (± 0)	3.6 (± 0)	7.0 (± 0.1)
Junco, snowbird	100 (± 0)	97.4 (± 0.1)	97.8 (± 0.1)	30 (± 0)	1.7 (± 0)	3.6 (± 0.1)
Kuvasz	100 (± 0)	98.2 (± 0)	98.3 (± 0.1)	30 (± 0)	1.9 (± 0)	4.5 (± 0.1)
Labrador retriever	100 (± 0)	62.5 (± 0.2)	93.5 (± 0.6)	30 (± 0)	2.6 (± 0)	6.0 (± 0.1)
Lighter, Light	100 (± 0)	30.2 (± 0.2)	94.2 (± 0.7)	30 (± 0)	2.6 (± 0)	5.6 (± 0.1)
Lycaenid butterfly	100 (± 0)	80.0 (± 0.2)	94.5 (± 0.5)	30 (± 0)	3.2 (± 0)	4.6 (± 0)
Magpie	100 (± 0)	22.4 (± 0.2)	93.3 (± 0.7)	30 (± 0)	1.5 (± 0)	4.1 (± 0.1)
Marmot	100 (± 0)	94.8 (± 0.1)	96.8 (± 0.2)	30 (± 0)	2.6 (± 0)	5.4 (± 0.1)
Muzzle	100 (± 0)	88.5 (± 0.1)	94.3 (± 0.4)	30 (± 0)	2.6 (± 0)	4.8 (± 0.1)
Papillon	100 (± 0)	99.9 (± 0)	99.9 (± 0)	30 (± 0)	1.6 (± 0)	4.0 (± 0.1)
Rock beauty	100 (± 0)	79.9 (± 0.2)	94.3 (± 0.5)	30 (± 0)	3.4 (± 0)	6.0 (± 0)
Siberian husky	100 (± 0)	36.0 (± 0.2)	94.4 (± 0.5)	30 (± 0)	2.1 (± 0)	4.9 (± 0.1)
Stinkhorn	100 (± 0)	92.7 (± 0.1)	95.4 (± 0.3)	30 (± 0)	3.6 (± 0)	5.4 (± 0.1)
Tennis ball	100 (± 0)	10.3 (± 0.1)	93.3 (± 0.6)	30 (± 0)	1.5 (± 0)	4.0 (± 0)
Tinca tinca	100 (± 0)	97.3 (± 0.1)	97.8 (± 0.1)	30 (± 0)	2.3 (± 0)	4.4 (± 0.1)
Torch	100 (± 0)	40.4 (± 0.2)	94.6 (± 0.5)	30 (± 0)	3.5 (± 0)	6.7 (± 0)
Unicycle	100 (± 0)	99.4 (± 0)	99.4 (± 0)	30 (± 0)	3.4 (± 0)	5.1 (± 0.1)
Water ouzel	100 (± 0)	98.5 (± 0)	98.7 (± 0.1)	30 (± 0)	1.6 (± 0)	3.6 (± 0.1)
White wolf	100 (± 0)	78.5 (± 0.1)	93.7 (± 0.5)	30 (± 0)	2.4 (± 0)	4.7 (± 0.1)

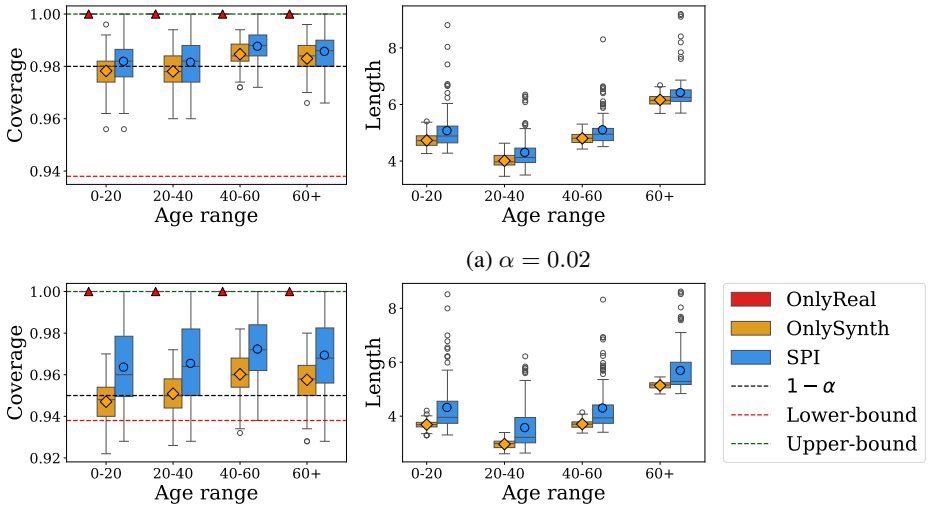
Table S13: Per-class conditional coverage (in %) and prediction set size for each method, computed over 100 trials using the HPS score function. Standard errors are shown in parentheses. The target coverage level is $1 - \alpha = 0.98$. The theoretical coverage guarantees for both `SPI-Whole` and `SPI-Subset` are in the range $[93.7, 100]$. Other details are as in Figure S18.

	Coverage (%)			Size		
Class	`Only Real`	`SPI Whole`	`SPI Subset`	`Only Real`	`SPI Whole`	`SPI Subset`
Admiral	100 (± 0)	93.5 (± 0.7)	95.3 (± 0.4)	30 (± 0)	5.6 (± 0.1)	5.6 (± 0)
American robin	100 (± 0)	98.9 (± 0.1)	96.2 (± 0.2)	30 (± 0)	3.9 (± 0.1)	3.6 (± 0.1)
Barracouta	100 (± 0)	95.1 (± 0.4)	95.6 (± 0.4)	30 (± 0)	6.4 (± 0.1)	6.2 (± 0.1)
Beaver	100 (± 0)	97.4 (± 0.1)	96.0 (± 0.3)	30 (± 0)	3.6 (± 0.1)	3.5 (± 0.1)
Bicycle	100 (± 0)	99.6 (± 0)	97.7 (± 0.1)	30 (± 0)	3.2 (± 0)	3.2 (± 0)
Bulbul	100 (± 0)	99.5 (± 0)	97.2 (± 0.2)	30 (± 0)	3.9 (± 0.1)	3.7 (± 0.1)
Coral fungus	100 (± 0)	98.5 (± 0.1)	96.3 (± 0.2)	30 (± 0)	3.7 (± 0)	3.6 (± 0)
English springer	100 (± 0)	98.0 (± 0.1)	97.2 (± 0.2)	30 (± 0)	3.7 (± 0.1)	3.7 (± 0.1)
Garfish	100 (± 0)	95.7 (± 0.3)	95.0 (± 0.4)	30 (± 0)	5.3 (± 0.1)	5.1 (± 0.1)
Golden retriever	100 (± 0)	96.9 (± 0.2)	95.8 (± 0.3)	30 (± 0)	5.2 (± 0.1)	5.1 (± 0.1)
Gyromitra	100 (± 0)	99.7 (± 0.3)	99.7 (± 0.3)	30 (± 0)	3.8 (± 0)	3.7 (± 0)
Jay	100 (± 0)	93.4 (± 0.6)	94.4 (± 0.4)	30 (± 0)	6.8 (± 0.1)	5.9 (± 0.1)
Junco, snowbird	100 (± 0)	98.8 (± 0.1)	96.9 (± 0.2)	30 (± 0)	3.5 (± 0.1)	3.2 (± 0.1)
Kuvasz	100 (± 0)	98.0 (± 0.1)	96.8 (± 0.2)	30 (± 0)	4.0 (± 0.1)	4.0 (± 0.1)
Labrador retriever	100 (± 0)	95.9 (± 0.3)	95.6 (± 0.3)	30 (± 0)	5.3 (± 0.1)	5.2 (± 0.1)
Lighter, Light	100 (± 0)	94.5 (± 0.5)	94.5 (± 0.5)	30 (± 0)	4.7 (± 0.1)	4.7 (± 0.1)
Lycaenid butterfly	100 (± 0)	99.5 (± 0)	98.9 (± 0.1)	30 (± 0)	4.5 (± 0)	4.4 (± 0)
Magpie	100 (± 0)	98.7 (± 0.1)	96.2 (± 0.2)	30 (± 0)	3.5 (± 0.1)	3.3 (± 0.1)
Marmot	100 (± 0)	98.0 (± 0.1)	97.3 (± 0.2)	30 (± 0)	4.6 (± 0.1)	4.6 (± 0.1)
Muzzle	100 (± 0)	94.1 (± 0.5)	93.9 (± 0.5)	30 (± 0)	4.0 (± 0.1)	4.0 (± 0.1)
Papillon	100 (± 0)	96.5 (± 0.2)	95.6 (± 0.3)	30 (± 0)	3.9 (± 0.1)	3.9 (± 0.1)
Rock beauty	100 (± 0)	94.3 (± 0.5)	94.9 (± 0.4)	30 (± 0)	5.1 (± 0.1)	4.8 (± 0.1)
Siberian husky	100 (± 0)	96.1 (± 0.3)	96.1 (± 0.3)	30 (± 0)	4.0 (± 0.1)	4.0 (± 0.1)
Stinkhorn	100 (± 0)	97.7 (± 0.1)	96.6 (± 0.2)	30 (± 0)	4.5 (± 0.1)	4.3 (± 0.1)
Tennis ball	100 (± 0)	95.2 (± 0.3)	93.9 (± 0.5)	30 (± 0)	3.4 (± 0)	3.3 (± 0)
Tinca tinca	100 (± 0)	99.0 (± 0)	96.6 (± 0.2)	30 (± 0)	3.9 (± 0.1)	3.8 (± 0.1)
Torch	100 (± 0)	96.5 (± 0.2)	96.5 (± 0.2)	30 (± 0)	5.4 (± 0.1)	5.4 (± 0.1)
Unicycle	100 (± 0)	96.6 (± 0.2)	94.7 (± 0.4)	30 (± 0)	4.3 (± 0.1)	4.1 (± 0.1)
Water ouzel	100 (± 0)	97.8 (± 0.1)	95.9 (± 0.3)	30 (± 0)	3.6 (± 0)	3.3 (± 0.1)
White wolf	100 (± 0)	96.6 (± 0.2)	95.5 (± 0.3)	30 (± 0)	3.8 (± 0.1)	3.8 (± 0.1)

K Additional MEPS experiments

In this section, we present additional results for the MEPS regression experiments, complementing those reported in Section 4.2.

Figure S19 reports the coverage rates and prediction interval lengths for all age groups, evaluated at $\alpha = 0.02$, and 0.05. As in the main paper, we observe that `SPI` achieves coverage rates that remain within the theoretical bounds, with lower variance compared to `OnlyReal`.



(a) $\alpha = 0.02$

(b) $\alpha = 0.05$

Figure S19: MEPS dataset results: coverage and interval length for each age group, obtained by `OnlyReal`, `OnlySynth`, and `SPI`, at target coverage levels $1 - \alpha = 0.98$ (a), and 0.95 (b). Experiments are repeated over 100 trials. `OnlyReal` produces trivial (infinite) prediction intervals; thus, its interval length is omitted.

K.1 The effect of the real calibration set size

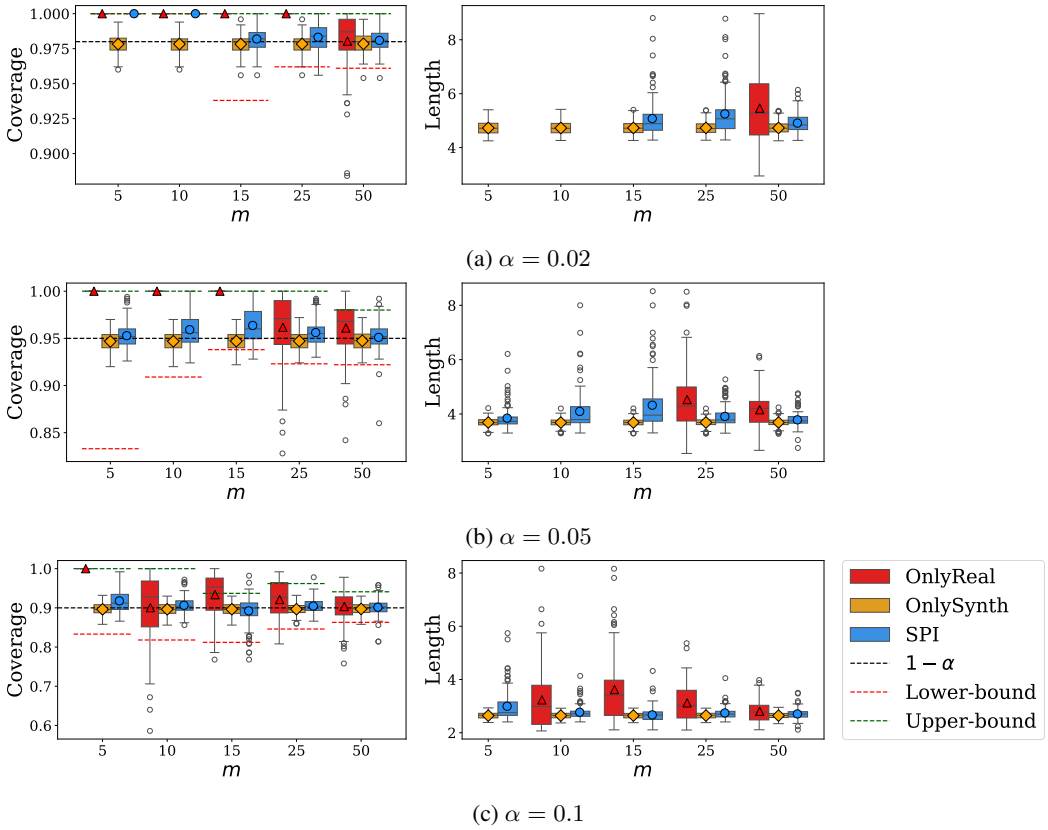
We replicate the experiments from Appendix J.1.1 on the MEPS dataset, evaluating the performance of different methods as a function of the real calibration set size, m.

Figure S20 and Figure S21 present the performance of all methods for age groups 0–20 and 20–40, respectively, across different α levels and values of m. The standard conformal method, `OnlyReal`, conservatively controls the coverage at the target level $1 - \alpha$; however, it results in larger and noisier prediction intervals due to the small sample size.

Similar to the trends observed in the main manuscript, `OnlySynth` achieves coverage close to the nominal $1 - \alpha$ level, indicating that the synthetic data align well with the real one. However, this approach does not have coverage guarantees.

In contrast, the proposed method, `SPI`, achieves coverage within the theoretical bounds, closely matching the target $1 - \alpha$ level while reducing coverage variance and producing smaller, more informative prediction intervals.

For $\alpha = 0.02$ with small calibration sizes ($m = 5$ or 10), the theoretical coverage bounds are equal to one under this window construction. This implies that the proposed method produces trivial prediction intervals. This behavior is known a priori and was also observed in the ImageNet experiment, where we used the same window construction parameters. Nevertheless, it can be avoided by employing a different window construction.

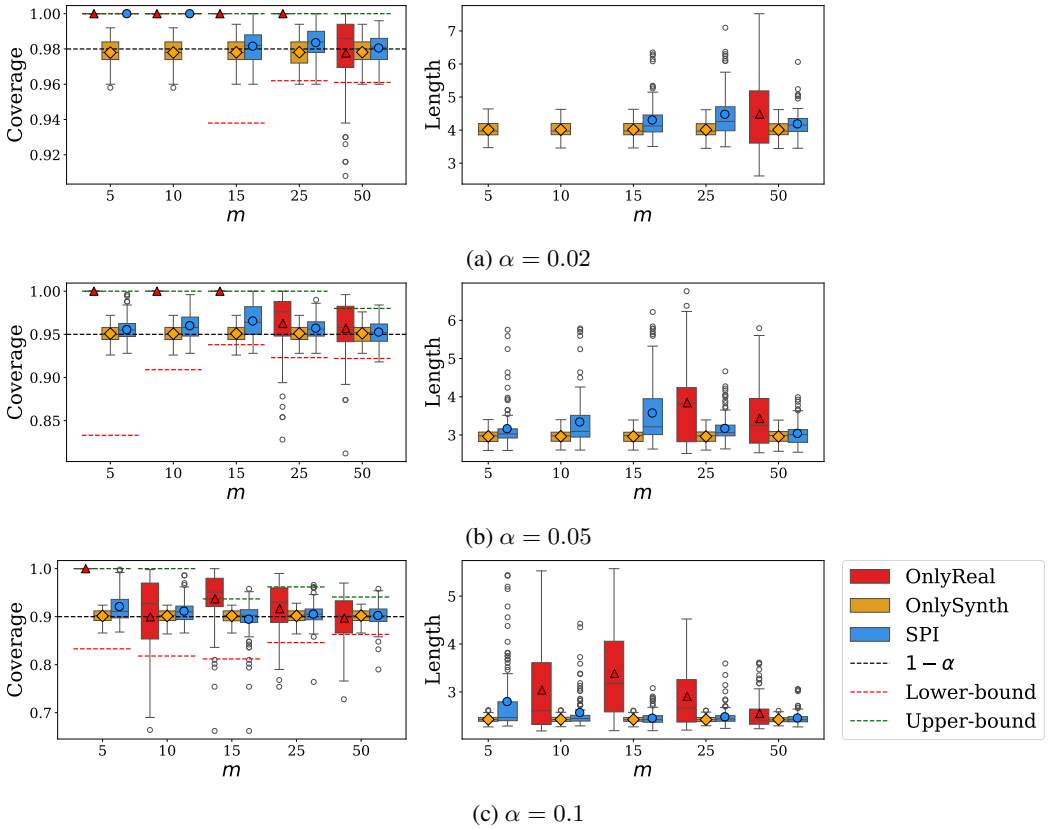


(a) $\alpha = 0.02$

(b) $\alpha = 0.05$

(c) $\alpha = 0.1$

Figure S20: MEPS dataset results: coverage and interval length for the 0–20 age group, obtained by `OnlyReal`, `OnlySynth`, and `SPI`, at target coverage levels $1 - \alpha = 0.98$ (a), 0.95 (b), and 0.9 (c). Experiments are repeated over 100 trials. For $\alpha = 0.02$ and 0.05, methods that produce trivial (infinite) prediction intervals are omitted from the interval length panel.



(a) $\alpha = 0.02$

(b) $\alpha = 0.05$

(c) $\alpha = 0.1$

Figure S21: MEPS dataset results: coverage and interval length for the 20–40 age group, obtained by `OnlyReal`, `OnlySynth`, and `SPI`, at target coverage levels $1 - \alpha = 0.98$ (a), 0.95 (b), and 0.9 (c). Experiments are repeated over 100 trials. For $\alpha = 0.02$ and 0.05, methods that produce trivial (infinite) prediction intervals are omitted from the interval length panel.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are supported by the theoretical results in Section 3.3 and the experimental findings in Section 4 and Appendices I to K.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our method in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The problem setup and all assumptions are detailed in Section 2, and complete proofs for the theoretical results in Section 3.3 are provided in Appendix G.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experimental details are provided in Section 4 and Appendix H, including dataset information. Software for reproducing the experiments is available at https://github.com/Meshiba/spi.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. For closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The software package implementing our method and reproducing the experiments is available at https://github.com/Meshiba/spi, including the code used for image generation with Stable Diffusion via Hugging Face. The ImageNet and MEPS datasets used are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experimental setting is described in Section 4, with additional technical details provided in Appendix H.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All plots are presented as boxplots to reflect variability across runs, and tables report means with standard errors; see Section 4 and Appendices J and K for the corresponding figures and tables.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Appendix H.1, we provide detailed information on the computational resources used in this work.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper follows the NeurIPS Code of Ethics in all aspects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential social impact of our work in Section 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not involve the release of any pretrained models, image generators, or datasets that pose a high risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and sources used in this paper are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets. However, we provide a well-documented software package.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core methodology in this research does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.