Semi-Supervised Conformal Prediction With Unlabeled Nonconformity Score

Supplementary Material

A. Notation and Lemma

We define s as the random variable of true score, $\tilde{s}$ as the random variable of naive unlabeled score. s takes values between $s_{\min}$ and $s_{\max}$. $\tilde{s}$ takes values between $\tilde{s}_{\min}$ and $\tilde{s}_{\max}$. Assume s and $\tilde{s}$ have no ties.

We define NNM as follows. For any data point q, assume that the labeled data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are i.i.d. of $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$, and independent of q. We choose

$$j = \text{argmin}_i |\tilde{S}(\boldsymbol{x}_i) - \tilde{S}(q)|,$$

then $S_{\text{nnm}}(q) = \tilde{S}(q) + S(\boldsymbol{x}_j, y_j) - \tilde{S}(\boldsymbol{x}_j)$. We refer to the random variable of $S(\boldsymbol{x}_j, y_j)$ as s_{n}, and the random variable of $\tilde{S}(\boldsymbol{x}_j)$ as $\tilde{s}_{\text{n}}$. Then we define the random variable of $S_{\text{nnm}}(q)$ as s_{nnm}.

Lemma 1. *Assume that the distribution of the nonconformity score S is continuous, and the labeled data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are i.i.d. samples from the same underlying distribution. $\mathcal{C}_{1-\alpha}(\boldsymbol{x}_{test})$ is given by split conformal prediction.Then the coverage given by the calibration set $\mathcal{D}$ follows a Beta distribution:*

$$\mathbb{P}\big(y_{test} \in \mathcal{C}_{1-\alpha}(\boldsymbol{x}_{test})\big) \mid \mathcal{D}) \sim \text{Beta}(l_\alpha^n, n+1-l_\alpha^n),$$

where $l_\alpha^n = \lceil (n+1)(1-\alpha) \rceil$.

Proof. Let F_s be the CDF of the distribution of s. Define $S_i = S(\boldsymbol{x}_i, y_i)$, which are i.i.d. drawn from the distribution of s. Let $S_{(1)} \leq ... \leq S_{(n)}$ be the order statistics of $S_1, ..., S_n$. In the split conformal prediction, we have $y_{\text{test}} \in \mathcal{C}_{1-\alpha}(\boldsymbol{x}_{\text{test}})$ if and only if $S(\boldsymbol{x}_{\text{test}}, y_{\text{test}}) \leq S_{(l_\alpha^n)}$ for $l_\alpha^n = \lceil (n+1)(1-\alpha) \rceil$, so we have

$$\mathbb{P}\big(y_{\text{test}} \in \mathcal{C}_{1-\alpha}(\boldsymbol{x}_{\text{test}})\big) \mid \mathcal{D}\big) = \mathbb{P}\big(S(\boldsymbol{x}_{\text{test}}, y_{\text{test}}) \leq S_{(l_\alpha^n)} \mid \mathcal{D}\big) = F(S_{(l_\alpha^n)}).$$

Next, let $U_i = F(S_i)$. By Probability Integral Transform, U_i is uniformly distributed in $[0, 1]$. Then, by definition of the Beta distribution, the l_α^n-th order statistic from n i.i.d. uniform random variables has a distribution

$$U_{(l_\alpha^n)} \sim \text{Beta}(l_\alpha^n, n+1-l_\alpha^n).$$

As a result,

$$\mathbb{P}\big(y_{\text{test}} \in \mathcal{C}_{1-\alpha}(\boldsymbol{x}_{\text{test}})\big) \mid \mathcal{D}) \sim \text{Beta}(l_\alpha^n, n+1-l_\alpha^n),$$

where $l_\alpha^n = \lceil (n+1)(1-\alpha) \rceil$. □

B. Proofs

B.1. Proof of Theorem 1

Proof. Assume s follows the distribution D_s, and $\tilde{s}$ follows the distribution $D_{\tilde{s}}$. Let D_{SemiCP} be a mixture distribution of D_s and $D_{\tilde{s}}$. The probability density function of D_{SemiCP} is given by

$$f_{\text{SemiCP}} = \frac{n}{n+N} f_s + \frac{N}{n+N} f_{\tilde{s}},$$

where f_s and $f_{\tilde{s}}$ are the probability density function of D_s and $D_{\tilde{s}}$. We refer to an auxiliary score function as S_{SemiCP}, and its random variable as s_{SemiCP}. Let B be an independent Bernoulli random variable

$$B \sim \text{Bernoulli}\left(\frac{n}{n+N}\right),$$

and define the mixed-score random variable

$$S_{\text{SemiCP}} = \begin{cases} S, & B = 1, \\ \tilde{S}, & B = 0. \end{cases}$$

As a result, we have

$$F_{\text{SemiCP}} = \frac{n}{n+N}F_s + \frac{N}{n+N}F_{\tilde{s}},$$

where F_{SemiCP}, F_s, $F_{\tilde{s}}$ are the CDFs of s_{SemiCP}, s, $\tilde{s}$. As SemiCP uses n labeled samples and N unlabeled samples to calibrate, we can see that D_{SemiCP} is actually the distribution of scores in SemiCP, and $\hat{\tau}_{\text{SemiCP}}$ is the $\frac{\lceil (n+N+1)(1-\alpha) \rceil}{n+N}$-quantile of the distribution D_{SemiCP}. Note that S_{SemiCP} is introduced as an auxiliary quantity for the proof; it is not used in the actual method.

Now we have

$$\begin{aligned}
&\mathbb{P}\big(y_{\text{test}} \in \mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}})\big) = \mathbb{P}\big(S(\boldsymbol{x}_{\text{test}}, y_{\text{test}}) \le \hat{\tau}_{\text{SemiCP}}\big) \\
&= \mathbb{P}\big(S(\boldsymbol{x}_{\text{test}}, y_{\text{test}}) \le \hat{\tau}_{\text{SemiCP}}\big) - \mathbb{P}\big(S_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) \le \hat{\tau}_{\text{SemiCP}}\big) + \mathbb{P}\big(S_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) \le \hat{\tau}_{\text{SemiCP}}\big) \\
&= F_s(\hat{\tau}_{\text{SemiCP}}) - F_{\text{SemiCP}}(\hat{\tau}_{\text{SemiCP}}) + \mathbb{P}\big(S_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) \le \hat{\tau}_{\text{SemiCP}}\big) \\
&= F_s(\hat{\tau}_{\text{SemiCP}}) - \big(\frac{n}{n+N}F_s(\hat{\tau}_{\text{SemiCP}}) + \frac{N}{n+N}F_{\tilde{s}}(\hat{\tau}_{\text{SemiCP}})\big) + \mathbb{P}\big(S_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) \le \hat{\tau}_{\text{SemiCP}}\big) \\
&= \epsilon_{n,N} + \mathbb{P}\big(S_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) \le \hat{\tau}_{\text{SemiCP}}\big) \\
&\ge 1 - \alpha + \epsilon_{n,N}
\end{aligned}$$

where $\epsilon_{n,N} = \frac{N}{N+n}(F_s(\hat{\tau}_{\text{SemiCP}}) - F_{\tilde{s}}(\hat{\tau}_{\text{SemiCP}}))$. □

B.2. Proof of Theorem 2.

Proof. By lemma 1, we have

$$\frac{\lceil (n+N+1)(1-\alpha) \rceil}{n+N+1} - \sqrt{\frac{\log(\frac{2}{\delta})}{2(n+N)}} \le \mathbb{P}\big(S_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) \le \hat{\tau}_{\text{SemiCP}} \mid \mathcal{D}\big) \le \frac{\lceil (n+N+1)(1-\alpha) \rceil}{n+N+1} + \sqrt{\frac{\log(\frac{2}{\delta})}{2(n+N)}},$$

with probability $1-\delta$, here we set S in lemma 1 as S_{SemiCP}, and $\frac{\lceil (n+N+1)(1-\alpha) \rceil}{n+N+1}$ is the expectation of the Beta distribution $\text{Beta}(l_\alpha^{n+N}, n+N+1-l_\alpha^{n+N})$.

As illustrated in the proof of theorem 1, the coverage using SemiCP satisfies the following:

$$\mathbb{P}\big(y_{\text{test}} \in \mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) \mid \mathcal{D}\big) = \epsilon_{n,N} + \mathbb{P}\big(S_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) \le \hat{\tau}_{\text{SemiCP}} \mid \mathcal{D}\big).$$

Then

$$\frac{\lceil (n+N+1)(1-\alpha) \rceil}{n+N+1} + \epsilon_{n,N} - \sqrt{\frac{\log(\frac{2}{\delta})}{2(n+N)}} \le \mathbb{P}(y_{\text{test}} \in \mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) | \mathcal{D}) \le \frac{\lceil (n+N+1)(1-\alpha) \rceil}{n+N+1} + \epsilon_{n,N} + \sqrt{\frac{\log(\frac{2}{\delta})}{2(n+N)}},$$

with probability $1-\delta$, that is,

$$1 - \alpha + \epsilon_{n,N} - \sqrt{\frac{\log(\frac{2}{\delta})}{2(n+N)}} \le \mathbb{P}(y_{\text{test}} \in \mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) | \mathcal{D}) \le 1 - \alpha + \frac{1}{n+N+1} + \epsilon_{n,N} + \sqrt{\frac{\log(\frac{2}{\delta})}{2(n+N)}},$$

with probability at least $1-\delta$. □

B.3. Proof of Theorem 3

Assumption 1. *The probability density function $f_{\tilde{s}}(x)$ is continuous and does not vanish on the interval $[\tilde{s}_{min}, \tilde{s}_{max}]$. $f_s(x)$ is continuous and does not vanish on the interval $[s_{min}, s_{max}]$. The probability density function $f_s(x|\tilde{s}=y)$ is M-Lipschitz with respect to x and y. $f_s(x|\tilde{s}=y)$ is uniformly bounded over (x,y).*

Remark 1. This is a mild assumption. In general, the probability that $\tilde{s}$ takes any value between $\tilde{s}_{\text{min}}$ and $\tilde{s}_{\text{max}}$ is nonzero, as shown in Fig 3. The Lipschitz property of $f_s(x|\tilde{s}=y)$ arises from the fact that its oscillation is not too drastic, which is also consistent with the typical behavior of $f_s(x|\tilde{s}=y)$ we observe in practice.

Assumption 2. *The random variables $\tilde{s}$, $\tilde{s}_n$, and s_n form a Markov chain $\tilde{s} \to \tilde{s}_n \to s_n$, which implies that $\tilde{s}$ is conditionally independent of s_n given $\tilde{s}_n$, i.e.*

$$f_{s_n}(\cdot|\tilde{s}_n = x, \tilde{s} = y) = f_{s_n}(\cdot|\tilde{s}_n = x) \quad \forall x, y \in \mathbb{R}.$$

Remark 2. This assumption is also mild, because we can regard the conditional probability of s_n given $\tilde{s}_n$ as the conditional probability of the true score s of one point given the naive unlabeled score $\tilde{s}$ of this point, independent of the naive unlabeled score of the other points.

Proof. By Assumption 1, $f_{\tilde{s}}(x)$ is continuous and does not vanish on the interval $[\tilde{s}_{\text{min}}, \tilde{s}_{\text{max}}]$, then we define ϵ and E as

$$\begin{aligned}\epsilon &:= \min_t \mathbb{P}(t-\delta \le \tilde{s} \le t+\delta) > 0,\\ E &:= \max_t \mathbb{P}(t-\delta \le \tilde{s} \le t+\delta) > 0, \qquad \forall t \in [\tilde{s}_{\text{min}}, \tilde{s}_{\text{max}}].\end{aligned}$$

then we have

$$\begin{aligned}&\mathbb{P}(|\tilde{s}_{\text{n}} - \tilde{s}| \le \delta \mid \tilde{s} = b) \ge \big(1-(1-\epsilon)^n\big),\\ &\mathbb{P}(|\tilde{s}_{\text{n}} - \tilde{s}| \le \delta \mid \tilde{s} = b) \le \big(1-(1-E)^n\big), \qquad \forall b \in [\tilde{s}_{\text{min}}, \tilde{s}_{\text{max}}].\end{aligned}$$

As $f_s(x|\tilde{s}=y)$ is uniformly bounded over (x,y), there exists B s.t.

$$f_s(x|\tilde{s}=y) \le B.$$

As $s_{\text{nnm}} = \tilde{s} + s_{\text{n}} - \tilde{s}_{\text{n}}$, now we obtain the following

$$\begin{aligned}f_{s_{\text{nnm}}}(a|\tilde{s}=b) &= \int_{-\infty}^{+\infty} f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b) f_{s_{\text{n}}}(a+\alpha|\tilde{s}_{\text{n}}=b+\alpha, \tilde{s}=b)d\alpha\\ &= \int_{-\infty}^{+\infty} f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b) f_{s_{\text{n}}}(a+\alpha|\tilde{s}_{\text{n}}=b+\alpha)d\alpha\\ &= \int_{-\delta}^{\delta} f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b) f_{s_{\text{n}}}(a+\alpha|\tilde{s}_{\text{n}}=b+\alpha)d\alpha\\ &\quad + \Big(\int_{-\infty}^{-\delta} + \int_{\delta}^{+\infty}\Big) f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b) f_{s_{\text{n}}}(a+\alpha|\tilde{s}_{\text{n}}=b+\alpha)d\alpha.\end{aligned}$$

The second equality is because of the Assumption 2. By the Assumption 1, we obtain the following inequality

$$-2M\delta \le f_s(a+\alpha|\tilde{s}=b+\alpha) - f_s(a|\tilde{s}=b) \le 2M\delta \qquad \forall \alpha \in [-\delta,\delta], \forall a,b \in \mathbb{R}.$$

We can see $f_{s_{\text{n}}}(a|\tilde{s}_{\text{n}}=b)$ as the conditional density probability of true score given the pesudo label score. As a result, we have $f_{s_{\text{n}}}(a|\tilde{s}_{\text{n}}=b) = f_s(a|\tilde{s}=b)$.

Then we have the following inequalities

$$\begin{aligned}f_{s_{\text{nnm}}}(a|\tilde{s}=b) - f_s(a|\tilde{s}=b) &= \int_{-\delta}^{\delta} f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b)\big(f_s(a+\alpha|\tilde{s}=b+\alpha) - f_s(a|\tilde{s}=b)\big)d\alpha\\ &\quad + \Big(\int_{-\infty}^{-\delta} + \int_{\delta}^{+\infty}\Big) f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b)\big(f_s(a+\alpha|\tilde{s}=b+\alpha) - f_s(a|\tilde{s}=b)\big)d\alpha\\ &\ge \int_{-\delta}^{\delta} f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b)(-2M\delta)d\alpha\\ &\quad - B\Big(\int_{-\infty}^{-\delta} + \int_{\delta}^{+\infty}\Big) f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b)d\alpha\\ &\ge (-2M\delta)\big(1-(1-E)^n\big) - B(1-\epsilon)^n,\\ f_{s_{\text{nnm}}}(a|\tilde{s}=b) - f_s(a|\tilde{s}=b) &\le \int_{-\delta}^{\delta} f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b)(2M\delta)d\alpha\\ &\quad + B\Big(\int_{-\infty}^{-\delta} + \int_{\delta}^{+\infty}\Big) f_{\tilde{s}_{\text{n}}}(b+\alpha|\tilde{s}=b)d\alpha\\ &\le 2M\delta\big(1-(1-E)^n\big) + B(1-\epsilon)^n, \qquad \forall b \in [\tilde{s}_{\text{min}}, \tilde{s}_{\text{max}}].\end{aligned}$$

As a result, we have

$$\big(1-(1-E)^n\big)\big(-2M\delta\big) - B(1-\epsilon)^n \le f_{s_{\text{nnm}}}(a|\tilde{s}=b) - f_s(a|\tilde{s}=b) \le \big(1-(1-E)^n\big)2M\delta + B(1-\epsilon)^n.$$

Then

$$\big[\big(1-(1-E)^n\big)\big(-2M\delta\big) - B(1-\epsilon)^n\big] \int_{-\infty}^{+\infty} f_{\tilde{s}}(b)db \le f_{s_{\text{nnm}}}(a) - f_s(a) \le \big[\big(1-(1-E)^n\big)2M\delta + B(1-\epsilon)^n\big] \int_{-\infty}^{+\infty} f_{\tilde{s}}(b)db,$$

$$\big(1-(1-E)^n\big)\big(-2M\delta\big) - B(1-\epsilon)^n \le f_{s_{\text{nnm}}}(a) - f_s(a) \le \big(1-(1-E)^n\big)2M\delta + B(1-\epsilon)^n,$$

and

$$F_{s_{\text{nnm}}}(t) - F_s(t) = \int_{\tilde{s}_{\text{min}}+s_{\text{min}}-\tilde{s}_{\text{max}}}^{t} \big(f_{s_{\text{nnm}}}(a) - f_s(a)\big)da.$$

Let $A = 2(\tilde{s}_{\text{max}} - \tilde{s}_{\text{min}}) + (s_{\text{max}} - s_{\text{min}})$, then we obtain the following

$$A\big[\big(1-(1-E)^n\big)\big(-2M\delta\big) - B(1-\epsilon)^n\big] \le F_{s_{\text{nnm}}}(t) - F_s(t) \le A\big[\big(1-(1-E)^n\big)2M\delta + B(1-\epsilon)^n\big].$$

□

C. Detailed analysis of unlabeled nonconformity score function

To understand the effectiveness of the Nearest Neighbor Matching (NNM) approach for estimating the nonconformity scores of unlabeled examples, we compare it against several baseline methods of increasing complexity:

1. **Naive:** Uses the nonconformity score of the pseudo-label as the prediction:

$$\tilde{S}_{\text{naive}}(\tilde{\boldsymbol{x}}_i, S, f) = S(\tilde{\boldsymbol{x}}_i, \hat{y}_i).$$

2. **Debias:** Applies a global correction by adding the average discrepancy between the true and pseudo nonconformity scores computed on labeled data:

$$\tilde{S}_{\text{debias}}(\tilde{\boldsymbol{x}}_i, D_{\text{labeled}}, S, f) = S(\tilde{\boldsymbol{x}}_i, \hat{y}_i) + \frac{1}{n}\sum_{j=1}^{n}\left[S(\boldsymbol{x}_j, y_j) - S(\boldsymbol{x}_j, \hat{y}_j)\right].$$

3. **Random Match (RM):** Adds a bias correction using a randomly selected labeled example from the dataset:

$$\tilde{S}_{\text{rm}}(\tilde{\boldsymbol{x}}_i;\, D_{\text{labeled}}, S, f) = S(\tilde{\boldsymbol{x}}_i, \hat{y}) + S(\boldsymbol{x}_j, y_j) - S(\boldsymbol{x}_j, \hat{y}), \quad j \sim \text{Uniform}\{1, \ldots, n\}.$$

4. **Nearest Neighbor Matching (NNM):** Selects a labeled example whose pseudo nonconformity score is closest to that of the unlabeled input, and uses it to construct a local bias correction:

$$\tilde{S}_{\text{nnm}}(\tilde{\boldsymbol{x}}_i, D_{\text{labeled}}, S, f) = S(\tilde{\boldsymbol{x}}_i, \hat{y})) + S(\boldsymbol{x}_j, y_j) - S(\boldsymbol{x}_j, \hat{y}), \quad \text{where } j = \underset{j\in\{1,\ldots,n\}}{\arg\min} \left|S(\tilde{\boldsymbol{x}}_i, \hat{y}_i) - S(\boldsymbol{x}_j, \hat{y}_j)\right|.$$

Why *naive* estimation is biased. The naive method estimates the nonconformity score of an unlabeled input $\tilde{\boldsymbol{x}}_i$ by evaluating the score at the model's predicted label $\hat{y}_i$, i.e.,$\tilde{S}_{\text{naive}}(\tilde{\boldsymbol{x}}_i) = S(\tilde{\boldsymbol{x}}_i, \hat{y}_i)$. By definition, the pseudo-label $\hat{y}_i$ corresponds to the most confident class under the model, and hence its nonconformity score is typically the smallest among all possible labels.

As a result, the naive method systematically underestimates the true nonconformity score, since $\tilde{y}_i$ may not coincide with the model prediction. This underestimation leads to an underestimated quantile threshold and, consequently, a conformal prediction set that is too narrow. In turn, this violates the target coverage guarantee, as the predicted set is less likely to contain the true label.

Why the *Debias* method is inconsistent. The debias method attempts to correct the naive estimate by adding the global average bias observed on the labeled calibration set:

$$\tilde{S}_{\text{debias}}(\tilde{\boldsymbol{x}}_i) = \tilde{U}_i + \bar{\Delta}_n, \quad \text{where} \quad \bar{\Delta}_n = \frac{1}{n}\sum_{j=1}^{n}(V_j - U_j).$$

By the law of large numbers, $\bar{\Delta}_n \xrightarrow{p} \mathbb{E}[\Delta(X)]$, the expected bias across the entire input space. However, this correction ignores the local context of $\tilde{X}_i$, and generally $\Delta(\tilde{X}_i) \neq \mathbb{E}[\Delta(X)]$, especially when the bias function $\Delta(X)$ varies spatially. As a result, $\tilde{S}_{\text{debias}}(\tilde{\boldsymbol{x}}_i)$ converges to the wrong target and is therefore inconsistent for the true score $\tilde{V}_i$.

Why *random match* is suboptimal. The random match (RM) method improves upon debias by using a labeled instance j drawn uniformly from a local neighborhood $\mathcal{N}_n(\tilde{\boldsymbol{x}}_i, R_n)$, and correcting the pseudo-score using its observed bias:

$$\tilde{S}_{\text{rm}}(\tilde{\boldsymbol{x}}_i) = \tilde{U}_i + \Delta_j, \quad j \sim \text{Uniform}(\mathcal{N}_n).$$

This guarantees that $\|\boldsymbol{x}_j - \tilde{\boldsymbol{x}}_i\| \to 0$, helping control the error $|V_j - \tilde{V}_i|$. However, RM does not consider the pseudo-score $\tilde{U}_i$ when selecting j, meaning the matched point may differ significantly in model confidence. This neglect can result in poor approximation of $\Delta(\tilde{\boldsymbol{x}}_i)$, especially in regions where the model's bias is heterogeneous, thereby limiting RM's accuracy in finite samples.

Why *Nearest Neighbor Matching* is superior. Nearest Neighbor Matching (NNM) enhances RM by selecting, within the same neighborhood, the point j^* whose pseudo-score U_j is closest to $\tilde{U}_i$:

$$\tilde{S}_{\text{nnm}}(\tilde{\boldsymbol{x}}_i) = \tilde{U}_i + \Delta_{j^*}, \quad j^* = \arg\min_{j \in \mathcal{N}_n} |U_j - \tilde{U}_i|.$$

This matching leverages both feature-space proximity and pseudo-score similarity, yielding a local and adaptive bias correction. Under mild continuity assumptions, we can show that $\boldsymbol{x}_{j^*} \xrightarrow{p} \tilde{\boldsymbol{x}}_i$, and thus $\Delta_{j^*} \xrightarrow{p} \Delta(\tilde{\boldsymbol{x}}_i)$, making NNM an asymptotically consistent estimator. Moreover, NNM achieves lower estimation error in finite samples by selecting a neighbor that better matches both the location and the model output, making it theoretically sound and practically more effective.

Finally, we present the experimental results of coverage gap and prediction set size for these four methods in Appendix I.5, demonstrating that our approach achieves the best improvement in the stability and efficiency of CP. Additionally, we provide ablation studies on the selection criteria and number of nearest neighbors in Appendix I.6 and Appendix I.7, respectively.

D. SemiCP vs. Semi-supervised Quantile Estimation

Our approach, SemiCP, addresses the inaccuracy of quantile estimation in conformal prediction (CP) by incorporating nonconformity scores computed for each unlabeled observation into the calibration process. In contrast, semi-supervised quantile estimation methods [3, 6, 56] leverage auxiliary unlabeled data directly to estimate quantiles. Below, we summarize the key innovations and advantages of SemiCP:

1. **Training-free and information-efficient.** Semi-supervised quantile estimation typically requires auxiliary information [56], the training of additional models [6], or iterative optimization such as gradient descent [3]. In contrast, SemiCP requires no extra data beyond the standard CP inputs and is entirely training-free.
2. **Nonconformity-based per-point scores.** Traditional semi-supervised quantile estimation yields a single, global quantile estimate, whereas SemiCP computes a nonconformity score for each unlabeled instance, directly incorporating the core CP concept of nonconformity. This per-point scoring confers several benefits: it naturally extends to conditional conformal prediction, facilitates integration with other CP enhancements, and yields more fine-grained uncertainty measures. We demonstrate these advantages and report significant empirical gains in Section 5.2.
3. **Compatibility with randomized score functions.** Many CP score functions—such as APS [40], RAPS [1], and SAPS [22]—incorporate randomization that cannot be captured by parametric models. Semi-supervised quantile estimation fails to adapt to these randomized scores, whereas SemiCP can easily accommodate them. In Appendix E, we outline the adaptation strategy, and in Appendix I.8 we present corresponding experimental results.

E. Unlabeled nonconformity score with random factor

In the main experiments, we employ the non-random version of the nonconformity score, as the introduction of a random factor can influence the nearest-neighbor matching process. However, some nonconformity scores, such as APS, utilize randomization techniques to get tighter prediction sets. In this section, we describe how to incorporate a random factor into the Nearest-Neighbor Matching (NNM) method.

Let the score with a random factor be denoted as $S(\boldsymbol{x}_i, y_i, u_i)$, where u_i is an independent random variable following a uniform distribution on $[0, 1]$. The randomized version of NNM requires only a slight modification to the original method: the nearest neighbor is still matched using the no-random score, but the same random factor is applied during the computation process. Specifically, it can be expressed as:

$$\tilde{S}_{\text{nnm-r}}(\tilde{\boldsymbol{x}}_i, \mathcal{D}_{\text{labeled}}, S, f, u_i) = S(\tilde{\boldsymbol{x}}_i, \hat{y}, u_i) + S(\boldsymbol{x}_j, y_j, u_i) - S(\boldsymbol{x}_j, \hat{y}, u_i),$$

$$where\ j = \underset{j \in \{1,\ldots,n\}}{\arg\min} \left| S(\tilde{\boldsymbol{x}}_i, \hat{y}) - S(\boldsymbol{x}_j, \hat{y}) \right|.$$

Here, u_i remains independent of $(\boldsymbol{x}_i, y_i)$, but the same u_i in $S(\cdot)$ is used for each $\tilde{\boldsymbol{x}}_i$. The experimental results for NNM-R are presented in Appendix I.8.

F. Supplemental algorithms

The algorithm of SemiCP is Algorithm 1. The algorithm of SemiCP-conditional is Algorithm 2.

Algorithm 1 SemiCP: Semi-supervised Conformal Prediction

Require: Labeled data $\mathcal{D}_{\text{labeled}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, Unlabeled data $\mathcal{D}_{\text{unlabeled}} = \{\tilde{\boldsymbol{x}}_i\}_{i=1}^N$, Significance level $\alpha \in (0,1)$, Score function $S(\cdot,\cdot)$, Prediction model $f(\cdot)$, Test input $\boldsymbol{x}_{\text{test}}$, Label set $\mathcal{Y}$

Ensure: Prediction set $\mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}})$

Compute nonconformity scores for labeled data: $s_i = S(\boldsymbol{x}_i, y_i), \quad i = 1, \ldots, n$
Estimate nonconformity scores $\tilde{s}_i$ for unlabeled data $\tilde{\boldsymbol{x}}_i$:
for $i = 1$ **to** N **do**
    $j = \arg\min_{j \in \{1,\ldots,n\}} |S(\tilde{\boldsymbol{x}}_i, \hat{y})) - S(\boldsymbol{x}_j, \hat{y})|$
    $\tilde{s}_i = S(\tilde{\boldsymbol{x}}_i, \hat{y}) + S(\boldsymbol{x}_j, y_j) - S(\boldsymbol{x}_j, \hat{y})$
end for
Compute threshold: $\hat{\tau}_{\text{SemiCP}} = \text{Quantile}\left(\{s_i\}_{i=1}^n \cup \{\tilde{s}_i\}_{i=1}^N, \frac{\lceil (n+N+1)(1-\alpha) \rceil}{n+N}\right)$
Form prediction set: $\mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}}) = \{y \in \mathcal{Y} : S(\boldsymbol{x}_{\text{test}}, y) \leq \hat{\tau}_{\text{SemiCP}}\}$
return $\mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}})$

Algorithm 2 SemiCP-conditional: Semi-supervised Conformal Prediction on conditional setting

Require: Labeled data $\mathcal{D}_{\text{labeled}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, Unlabeled data $\mathcal{D}_{\text{unlabeled}} = \{\tilde{\boldsymbol{x}}_i\}_{i=1}^N$, Significance level $\alpha \in (0,1)$, Score function $S(\cdot,\cdot)$, Prediction model $f(\cdot)$, Test input $\boldsymbol{x}_{\text{test}}$, Label set $\mathcal{Y}$, Group set $\mathcal{G}$

Ensure: Prediction set $\mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}})$

Compute nonconformity scores for labeled data: $s_i = S(\boldsymbol{x}_i, y_i), \quad i = 1, \ldots, n$
Estimate nonconformity scores $\tilde{s}_i$ for unlabeled data $\tilde{\boldsymbol{x}}_i$:
for $i \in \{1, ..., N\}$ **do**
    $j = \arg\min_{j \in \{1,\ldots,n\}} |S(\tilde{\boldsymbol{x}}_i, \hat{y}) - S(\boldsymbol{x}_j, \hat{y})|$
    $\tilde{s}_i = S(\tilde{\boldsymbol{x}}_i, \hat{y}) + S(\boldsymbol{x}_j, y_j) - S(\boldsymbol{x}_j, \hat{y})$
end for
for $G_i \in \mathcal{G}$ **do**
    Select all nonconformity scores belonging to group G_i: $\{s_i\}_{i=1}^{n_g}$, $\{\tilde{s}_i\}_{i=1}^{N_g}$
    Compute threshold: $\hat{\tau}^g_{\text{SemiCP}} = \text{Quantile}\left(\{s_i\}_{i=1}^{n_g} \cup \{\tilde{s}_i\}_{i=1}^{N_g}, \frac{\lceil (n_g+N_g+1)(1-\alpha) \rceil}{n_g+N_g}\right)$
end for
Form prediction set: $\mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}} \in G_i) = \{y \in \mathcal{Y} : S(\boldsymbol{x}_{\text{test}}, y) \leq \hat{\tau}^g_{\text{SemiCP}}\}$
return $\mathcal{C}_{\text{SemiCP}}(\boldsymbol{x}_{\text{test}})$

G. Computational overhead of algorithm

Theoretical Complexity. Let n denote the number of labeled calibration samples and N the number of unlabeled samples. We assume that computing a nonconformity score takes constant time $\mathcal{O}(1)$.

For the **standard** split conformal predictor, calibration only uses the n labeled samples, leading to a time complexity of $\mathcal{O}(n)$. The **oracle** method performs calibration on all $n + N$ samples and thus requires $\mathcal{O}(n + N)$ time.

For the proposed **SemiCP** method, the NNM step introduces additional computation. The pseudo scores of labeled samples are first sorted in $\mathcal{O}(n \log n)$ time. Then, for each of the N unlabeled samples, the nearest pseudo score is found via binary search in $\mathcal{O}(\log n)$ time, resulting in $\mathcal{O}(N \log n)$ complexity for the matching step. Therefore, the overall complexity of SemiCP is $\mathcal{O}(n + n \log n + N \log n + N) = \mathcal{O}((n + N) \log n)$. In our settings where $N \gg n$, the additional overhead of SemiCP compared to the oracle method is only a logarithmic factor, which is practically negligible.

Empirical Runtime. We empirically evaluate the calibration runtime of the three methods using the TorchCP framework [23], excluding model inference and test-time prediction. Experiments are conducted with a ResNet-50 model on ImageNet using the THR score, on a single RTX 4090 GPU, averaged over 1000 runs. As shown in Table 1, SemiCP incurs slightly higher runtime due to the additional nearest-neighbor matching step, but the absolute overhead remains extremely small.

Table 1. Average calibration runtime (seconds per run).

Method	Standard	Oracle	SemiCP
Runtime (s)	0.000216	0.000273	0.001583

H. Evaluation metrics

AvgSize (Average Prediction Set Size) The average prediction set size quantifies the mean number of labels in the prediction sets:

$$\text{AvgSize} = \frac{1}{|D_{\text{test}}|} \sum_{(\boldsymbol{x},y)\in D_{\text{test}}} |\mathcal{C}(\boldsymbol{x})|.$$

CovGap (Average Coverage Gap) Coverage gap measures how far the marginal coverage deviates from the desired coverage level of $1-\alpha$. The marginal coverage is defined as $\hat{c} = \frac{1}{|D_{\text{test}}|} \sum_{(\boldsymbol{x},y)\in D_{\text{test}}} \mathbb{I}_{y\in\mathcal{C}(\boldsymbol{x})}$, then the average coverage gap is given by:

$$\text{CovGap} = 100 \times |\hat{c} - (1-\alpha)|.$$

For conditional conformal prediction, coverage gap evaluates the deviation of each subgroup's coverage from the target coverage $1-\alpha$. For example, in class-conditional conformal prediction, let $\mathcal{J}^y = \{i \in [N'] : Y_i' = y\}$ denote the index set of samples with label y. The coverage for class y is defined as: $\hat{c}_y = \frac{1}{|\mathcal{J}^y|} \sum_{i\in\mathcal{J}^y} \mathbb{I}_{y_i'\in\mathcal{C}(\boldsymbol{x}_i')}$, then the class-conditional coverage gap is given by:

$$\text{CovGap} = 100 \times \frac{1}{|\mathcal{Y}|} \sum_{y\in\mathcal{Y}} |\hat{c}_y - (1-\alpha)|.$$

Improvement Throughout the paper, "improvement" refers to the relative performance gain of a method over a baseline. Suppose M denotes the evaluation metric of interest, the improvement is computed as:

$$\text{improvement} = \frac{M_{\text{Standard}} - M_{\text{SemiCP}}}{M_{\text{Standard}} - M_{\text{Oracle}}} \times 100\%.$$

OverCovGap and UnderCovGap In section I.1, we also report Over-Coverage Gap and Under-Coverage Gap metrics to avoid the potential masking effect of the overall coverage gap, which may conceal the disadvantage of under-coverage. The Over-Coverage Gap and Under-Coverage Gap are defined as:

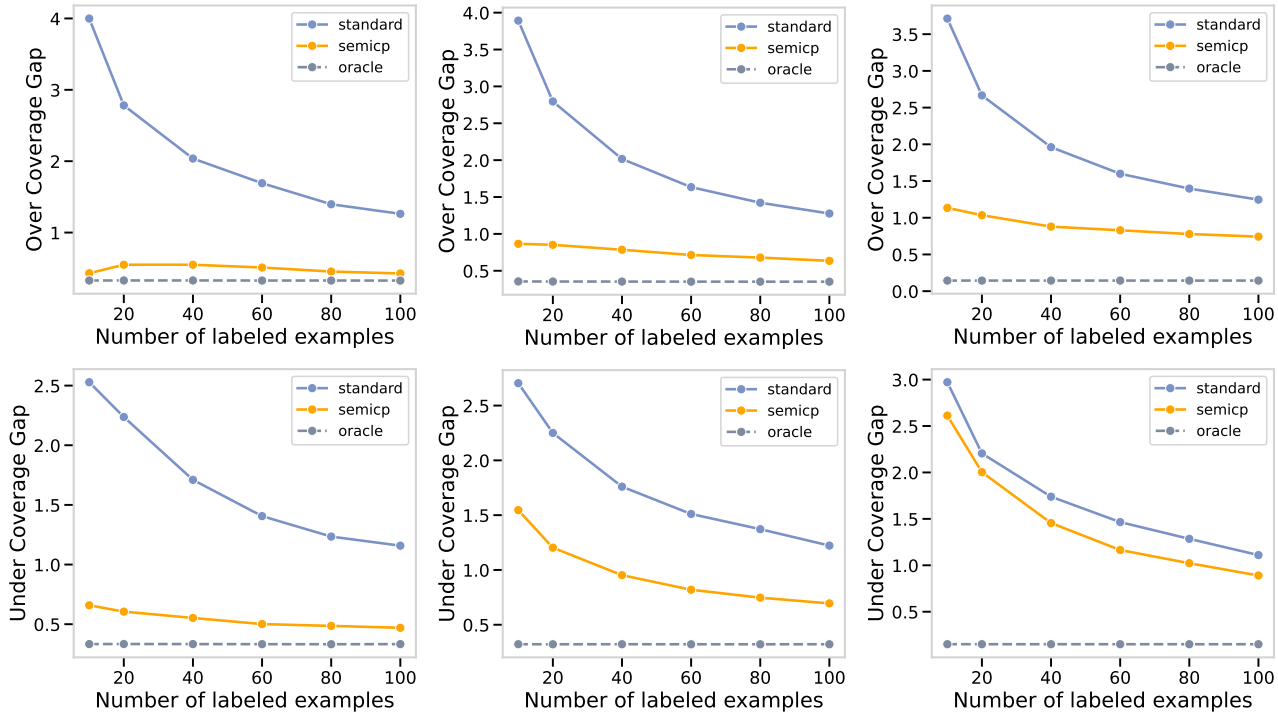
$$\text{OverCovGap} = 100 \times \mathbb{I}\{\hat{c} > (1-\alpha)\} \cdot |\hat{c} - (1-\alpha)|,$$

$$\text{UnderCovGap} = 100 \times \mathbb{I}\{\hat{c} < (1-\alpha)\} \cdot |\hat{c} - (1-\alpha)|.$$

I. Supplemental experiments

I.1. Supplemental metrics with OverCovGap and UnderCovGap metrics

Since the CovGap metric may obscure the separate effects of over-coverage and under-coverage, we additionally measure the over-coverage and under-coverage of our method, SemiCP, denoted as OverCovGap and UnderCovGap (see Appendix H). Fig. 9 illustrates the extent to which SemiCP improves over-coverage and under-coverage compared to the standard method under varying amounts of labeled data. The results demonstrate that our method consistently achieves lower levels of both over-coverage and under-coverage, suggesting a robust improvement over the standard method.



(a) CIFAR-10 (b) CIFAR-100 (c) ImageNet

Figure 9. Comparison of the relationship between Over Coverage Gap (top) and Under Coverage Gap (bottom) with different label data numbers. Experiments conducted on CIFAR-10, CIFAR-100, and ImageNet datasets using ResNet50, average on three score functions with 1000 different trials.

I.2. SemiCP is compatible with various score functions

Table 2 summarizes the performance of our approach under three different score functions—THR, APS, RAPS—across CIFAR10, CIFAR100, and ImageNet. The results reveal that SemiCP consistently narrows the coverage gap and reduces the average set size, regardless of the labeled score employed. For instance, SemiCP reduces 81.89% coverage gap and 97.32% set sizes for APS, while its performance nearly matches that of the oracle method. Similar trends are observed on THR, RAPS, where SemiCP shows substantial improvements. Notably, SemiCP is more effective for those scores that initially yield poorer performance. Specifically, the improvement in APS is greater than that in RAPS, which in turn exceeds the improvement observed with THR. These improvements confirm the compatibility of SemiCP across different labeled score functions.

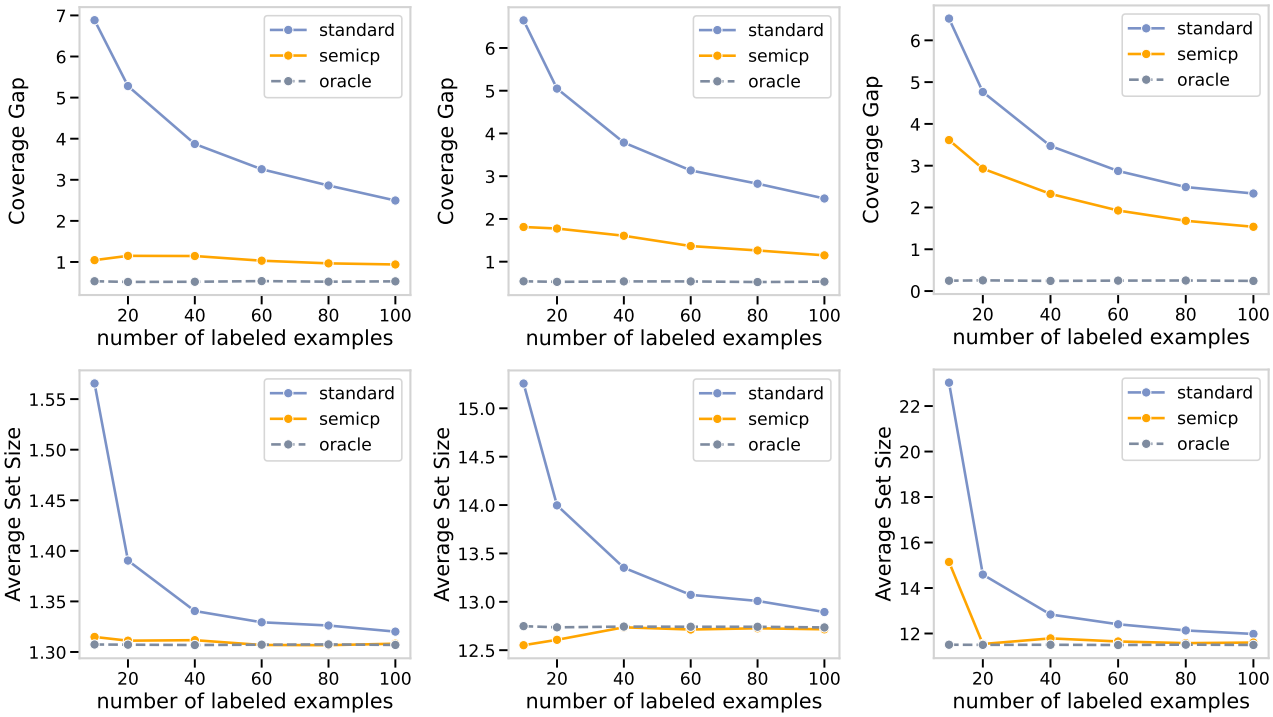
Fig. 10 shows how the coverage gap and prediction set size of SemiCP evolve as more unlabeled data are incorporated when using the SAPS score [22]. Notably, unlike the previous experiments, we adopt the randomized version of SAPS (see E for details on the random factor). This is because the SAPS score function is highly non-smooth, making it difficult to obtain stable coverage guarantees without randomization. The results indicate that SemiCP effectively reduces the coverage gap and shrinks the average prediction set size with SAPS, demonstrating its strong compatibility with different score functions.

I.3. Failure Cases of SemiCP

We now turn to scenarios in which SemiCP may underperform. Since SemiCP selects labeled data based on pseudo-label uncertainty, its effectiveness depends critically on the quality of the pseudo-labels. We quantify this quality by the Top-1 accuracy of the pseudo-labeling model. As shown in Fig. 11, when the pseudo-label accuracy is low, SemiCP comparably to, or even worse than, the standard baseline. In this low-accuracy regime, unreliable pseudo-labels lead to inaccurate uncertainty estimates, which in turn distort the NNM scores and limit the performance improvement. As a result, the coverage gap remains large and the average set size exhibits little reduction. This reveals a failure case of SemiCP: when the model's prediction accuracy is too low, the SemiCP method may fail.

Table 2. Average coverage gap and set size with different score functions on three datasets. Standard errors are reported in parentheses. The number of labeled data is fixed at 80.

Score		**THR**		**APS**		**RAPS**	
Dataset	**Method**	**CovGap**	**AvgSize**	**CovGap**	**AvgSize**	**CovGap**	**AvgSize**
CIFAR10	standard	2.64 (2.0)	0.91 (0.0)	2.62 (1.9)	1.44 (0.1)	2.63 (1.9)	1.40 (0.1)
	semicp	0.88 (0.8)	0.90 (0.0)	0.96 (0.8)	1.42 (0.0)	0.98 (0.8)	1.38 (0.0)
	oracle	0.65 (0.5)	0.90 (0.0)	0.66 (0.5)	1.42 (0.0)	0.66 (0.5)	1.38 (0.0)
CIFAR100	standard	2.68 (2.0)	1.21 (0.2)	2.88 (2.1)	39.40 (5.2)	2.83 (2.1)	11.02 (0.9)
	semicp	2.57 (1.8)	1.19 (0.3)	0.75 (0.6)	38.56 (1.2)	0.95 (0.8)	10.92 (0.3)
	oracle	0.69 (0.5)	1.15 (0.0)	0.67 (0.5)	38.57 (0.9)	0.67 (0.5)	10.93 (0.2)
ImageNet	standard	2.69 (2.0)	1.70 (0.5)	2.66 (2.0)	179.35 (51.2)	2.70 (2.0)	16.57 (3.4)
	semicp	2.63 (1.9)	1.62 (0.5)	1.13 (0.8)	166.98 (22.3)	1.64 (1.2)	15.90 (2.3)
	oracle	0.29 (0.2)	1.52 (0.0)	0.30 (0.2)	166.15 (2.7)	0.29 (0.2)	15.74 (0.2)
improvement		**32.26%**	**55.43%**	**81.89%**	**97.32%**	**71.51%**	**96.11%**



(a) CIFAR-10 (b) CIFAR-100 (c) ImageNet

Figure 10. Average performance comparison of SemiCP with varying numbers of labeled data with SAPS on CIFAR-10, CIFAR-100, and ImageNet. We use 4,000 unlabeled samples for CIFAR-10 and CIFAR-100, and 20,000 unlabeled samples for ImageNet. The weight=0.01.

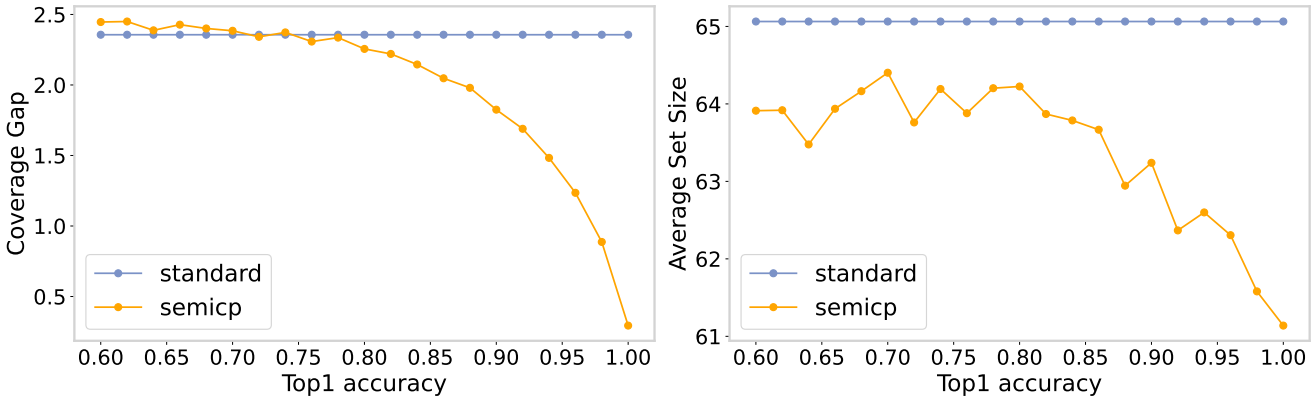


Figure 11. Coverage gap and average set size of SemiCP with different pseudo-label accuracies. Each experiment was conducted on ImageNet and averaged over three score functions with 1000 trials. The number of labeled data and unlabeled data is fixed at 100 and 20000.

I.4. Robustness to labeled data distribution shift

In real-world applications, the exchangeability assumption in conformal prediction often fails when labeled calibration data shift. To assess SemiCP's robustness, we test it with labeled data from a shifted domain (ImageNet-R[26]/ImageNet-S[17]/ImageNet-V2[38]), while the unlabeled and test data follow the target distribution. As shown in Table 3, SemiCP consistently outperforms standard CP in mitigating distribution shift, reducing the coverage gap by 8.35%–49.83% and prediction set size by 3.14%–53.14%, demonstrating its ability to recalibrate decision boundaries with target domain unlabeled data, even when labeled data is biased.

Table 3. Comparison of CovGap and AvgSize across different distribution shift scenarios. Each experiment was conducted on ImageNet and averaged over three score functions with 100 trials. The number of labeled data and unlabeled data is fixed at 20 and 20000.

	ImageNet-R		**ImageNet-S**		**ImageNetV2**	
Method	**CovGap**	**AvgSize**	**CovGap**	**AvgSize**	**CovGap**	**AvgSize**
standard	9.40 (1.1)	546.85 (198.6)	4.85 (3.9)	74.73 (110.4)	5.82 (3.9)	91.18 (129.6)
semicp	8.64 (2.1)	498.33 (358.9)	3.07 (2.7)	74.31 (132.5)	3.06 (2.3)	75.24 (114.9)
oracle	0.29 (0.2)	61.19 (74.6)	0.29 (0.2)	61.19 (74.7)	0.29 (0.2)	61.19 (74.7)
improvement	**8.35%**	**9.99%**	**38.88%**	**3.14%**	**49.83%**	**53.14%**

I.5. Discussion of different bias estimation methods

Fig. 12 compares four bias estimators. The *naive* method ignores the gap between the true nonconformity score $S(\boldsymbol{x}_j, y_j)$ and the pseudo-label score $S(\boldsymbol{x}_j, \hat{y}_j)$. The *debias* estimator corrects this by using the average bias of all labeled data, $\text{Bias} = \frac{1}{n}\sum_{j=1}^{n}\left[S(\boldsymbol{x}_j, y_j) - S(\boldsymbol{x}_j, \hat{y}_j)\right]$. In *random-sample*, we assign each unlabeled example the bias of a randomly chosen labeled point. Our SemiCP estimator is defined in Eq. 3.

The left panel shows that SemiCP steadily reduces the coverage gap as the number of labeled samples n increases, outperforming the standard CP baseline at all n. The other methods yield modest gains at very small n (e.g. $n = 10$) but quickly degrade relative to the baseline. The right panel plots prediction-set size: *random-sample* always produces overly large sets, *debias* performs well initially but worsens with growing n, and *naive* underestimates scores—hence thresholds—and fails to guarantee coverage (see Section 4.2). Detailed theory of four estimators is deferred to Section C. Overall, SemiCP offers the best balance of reliability and efficiency.

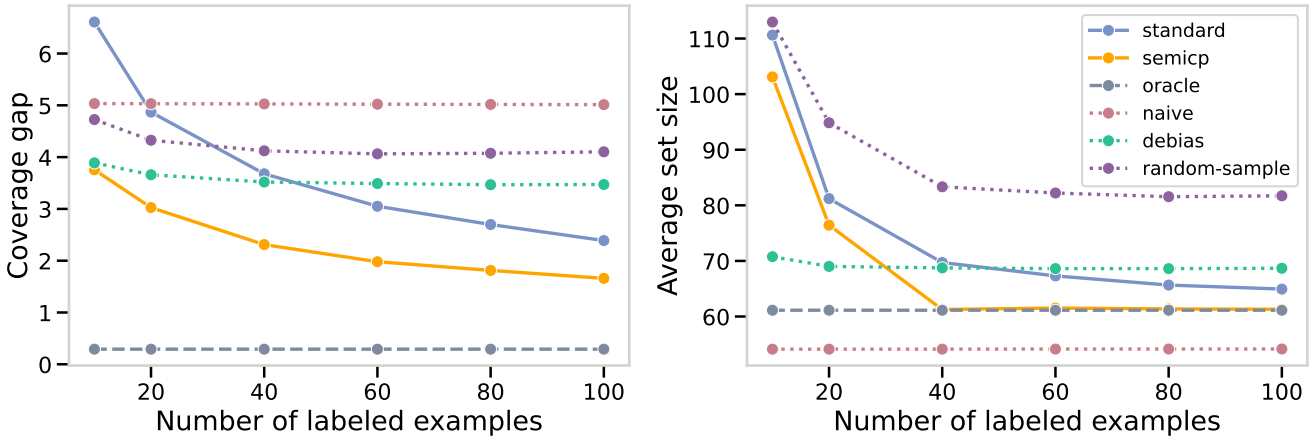


Figure 12. Average Coverage Gap and Set Size of SemiCP with different bias estimation methods. Each experiment was conducted on ImageNet and ResNet50, averaged over three score functions with 1000 trials. The number of unlabeled data is fixed at 20000.

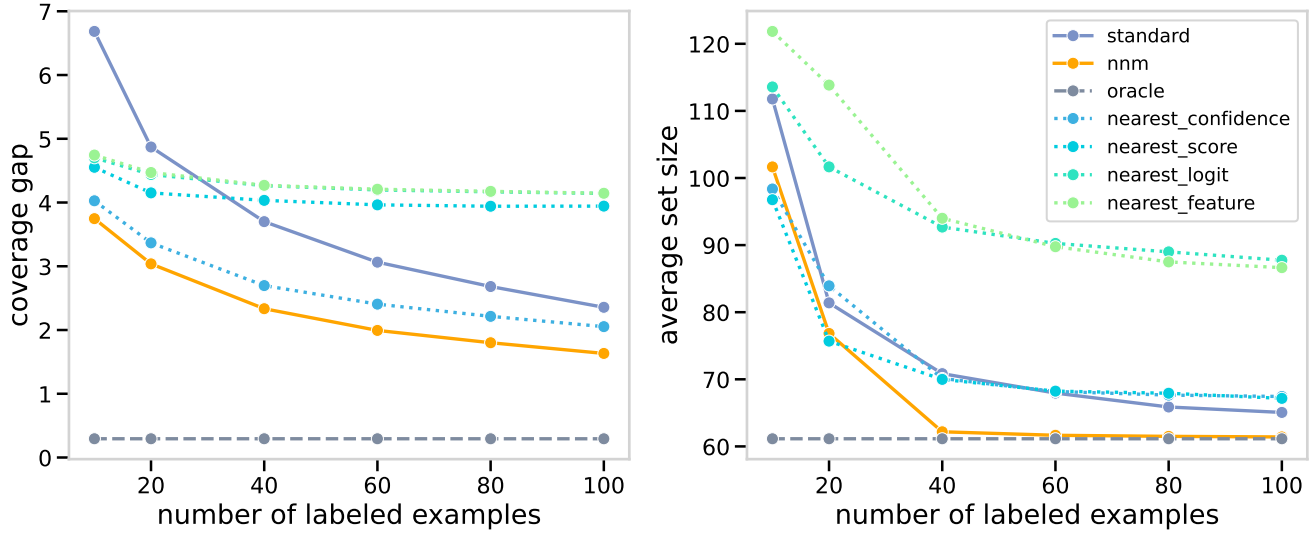


Figure 13. Average Coverage Gap and Set Size of SemiCP with different neighborhood selection methods. Each experiment was conducted on ImageNet and ResNet50, averaged over three score functions with 1000 trials. The number of unlabeled data is fixed at 20000.

I.6. Ablation study of different neighborhood selection methods

In Fig. 13, we compare several neighborhood selection criteria as an ablation study. Our proposed method, *nnm*, matches a labeled example with the most similar nonconformity score with a pseudo-label. In addition, *nearest_confidence* selects labeled examples with the highest model confidence; *nearest_score* finds the neighbor whose full nonconformity-score vector $S(\tilde{\boldsymbol{x}})$ is most similar; *nearest_logit* matches on the raw logit outputs of the model for $\boldsymbol{x}$; and *nearest_feature* uses the input-feature embedding for matching. All high-dimensional distances are measured with the Euclidean metric.

In the coverage gap metric (left), *nnm* uniformly outperforms competing methods. On average set size (right), *nearest_confidence* and *nearest_score* slightly undercut *nnm* at $n = 10$, but as the number of labeled examples increases, our method *nnm* exhibits a clear and sustained advantage. For example, at $n = 60$, *nnm* reduces the average set size from 69 to 61, whereas all alternative methods do not improve the standard baseline. In general, these results confirm that our neighborhood-selection criterion is optimal, providing the most reliable coverage guarantee and the smallest prediction sets.

I.7. Ablation study of different numbers of neighborhood

In Fig. 14, we present the results of SemiCP with different numbers of nearest neighbors. In the Fig. 14, *nnm_k* denotes the number of nearest neighbors chosen, where k indicates the number of neighbors. When multiple neighbors are selected, the bias $\Delta(\tilde{\boldsymbol{x}}_i)$ is estimated as the average of the biases of the k nearest neighbors, i.e., $\frac{1}{k}\sum_{i=1}^{k}\Delta(\boldsymbol{x}_j)$. The left plot shows that with $k = 1$, the coverage gap is minimized, achieving a more stable coverage of $1 - \alpha$. On the right, although the set size for $k = 2, 3, 5$ is even smaller than that of the oracle, this suggests that the coverage may not have been guaranteed, leading to overly small prediction sets. This could be because selecting closer neighbors allows for more accurate distribution estimates, resulting in better performance.

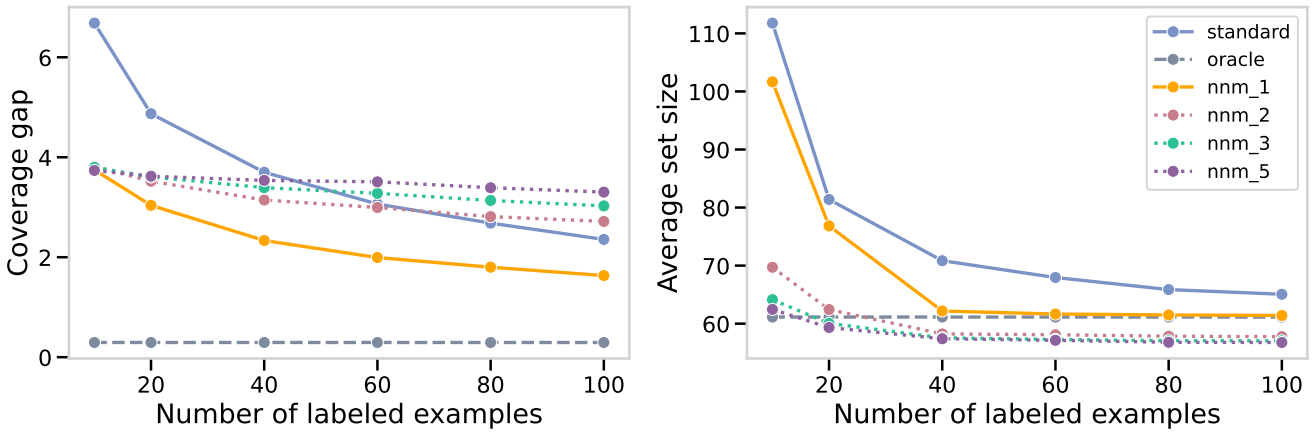


Figure 14. Coverage gap and average set size of SemiCP with different neighborhood numbers. Each experiment was conducted on ImageNet and averaged over three score functions with 1000 trials. The number of unlabeled data is fixed at 20000.

I.8. SemiCP for nonconformity score with random factor

Table 4 compares the performance of SemiCP under nonrandom and randomized matching strategies. Both versions consistently outperform the standard method in terms of reducing coverage gap across datasets and settings, confirming the effectiveness of leveraging unlabeled data. The non-random variant achieves the lowest coverage gap, indicating that deterministic matching yields more accurate bias correction. In contrast, the randomized version produces smaller average set sizes, particularly on high-class datasets such as CIFAR100 and ImageNet. This reflects a trade-off: the random strategy sacrifices a small amount of coverage for improved compactness, which may be desirable in scenarios with large output spaces or limited labeling budgets.

Table 4. Average coverage gap and set size with different score functions, no random and random versions on three datasets. The number of labeled data is fixed at 100.

	Score		APS(no random)		APS(random)		RAPS(no random)		RAPS(random)	
n	**Dataset**	**Method**	**CovGap**	**AvgSize**	**CovGap**	**AvgSize**	**CovGap**	**AvgSize**	**CovGap**	**AvgSize**
10	CIFAR10	standard	6.59	1.94	6.97	1.34	6.59	1.83	6.97	1.33
		semicp	1.23	1.42	1.82	0.97	1.29	1.38	1.83	0.96
		oracle	0.67	1.41	0.67	0.96	0.66	1.38	0.67	0.96
	CIFAR100	standard	6.04	46.91	7.82	15.01	6.13	12.64	7.67	6.71
		semicp	0.77	38.74	2.75	8.26	1.1	11.11	3.24	4.43
		oracle	0.63	38.62	0.69	8.01	0.6	10.94	0.68	4.22
	ImageNet	standard	6.45	273.48	6.23	117.55	6.35	28.9	6.21	21.95
		semicp	1.46	157.77	4.21	40.13	3.04	16.2	4.83	7.8
		oracle	0.3	166.26	0.31	36.89	0.26	15.76	0.32	6.53
20	CIFAR10	standard	5.17	1.53	4.93	1.00	5.17	1.46	4.93	0.99
		semicp	1.12	1.41	1.66	0.96	1.17	1.38	1.67	0.95
		oracle	0.67	1.41	0.66	0.96	0.65	1.38	0.66	0.96
	CIFAR100	standard	4.85	43.48	5.04	11.37	4.80	11.59	4.89	5.04
		semicp	0.70	38.49	2.34	7.82	0.98	10.89	2.62	4.11
		oracle	0.62	38.63	0.67	7.98	0.60	10.94	0.67	4.21
	ImageNet	standard	4.25	219.51	4.88	67.95	4.46	20.03	4.65	11.14
		semicp	1.45	161.03	3.39	39.63	2.46	15.75	3.79	7.16
		oracle	0.30	166.26	0.30	36.82	0.26	15.76	0.31	6.53
50	CIFAR10	standard	3.62	1.49	3.42	0.97	3.60	1.44	3.42	0.97
		semicp	1.17	1.43	1.51	0.96	1.21	1.39	1.52	0.96
		oracle	0.67	1.41	0.65	0.96	0.66	1.38	0.65	0.96
	CIFAR100	standard	3.84	39.96	3.59	9.57	3.81	11.07	3.55	4.67
		semicp	0.71	38.59	1.82	8.31	1.00	10.93	1.99	4.32
		oracle	0.62	38.61	0.66	7.99	0.61	10.94	0.66	4.21
	ImageNet	standard	3.53	186.92	3.49	52.32	3.52	17.13	3.36	8.13
		semicp	1.35	165.10	2.42	43.15	2.03	15.95	2.73	7.36
		oracle	0.30	166.25	0.32	36.68	0.26	15.76	0.30	6.51
100	CIFAR10	standard	2.37	1.44	2.23	0.97	2.36	1.40	2.25	0.97
		semicp	0.99	1.42	1.23	0.96	1.00	1.38	1.23	0.96
		oracle	0.67	1.41	0.65	0.96	0.66	1.38	0.64	0.96
	CIFAR100	standard	2.84	38.96	2.48	9.01	2.74	10.96	2.35	4.51
		semicp	0.76	38.64	1.44	8.20	0.93	10.93	1.49	4.24
		oracle	0.63	38.63	0.66	8.04	0.61	10.94	0.68	4.23
	ImageNet	standard	2.74	172.76	2.41	41.71	2.71	16.24	2.33	7.00
		semicp	1.05	163.38	1.90	39.05	1.50	15.77	2.17	6.86
		oracle	0.30	166.23	0.31	36.73	0.27	15.76	0.31	6.52
	improvement		**85.03%**	**100.55%**	**55.48%**	**88.4%**	**74.26%**	**101.64%**	**48.69%**	**88.69%**