Generative Conformal Prediction with Optimized Coverage Allocation

Minxing Zheng
Carnegie Mellon University

Shixiang Zhu
Carnegie Mellon University

Abstract

Conformal prediction provides model-agnostic uncertainty quantification with guaranteed coverage, but conventional methods often yield overly conservative uncertainty sets, particularly in multimodal or heterogeneous settings. This inefficiency arises from two sources: (i) limited expressiveness of the predictive model and (ii) simple nonconformity score designs. Most existing approaches advance only one of these axes, leaving the other underexplored. We propose generative conformal prediction with Optimized Ranking and Coverage Allocation (`ORCA`), a three-stage framework that advances both aspects jointly. `ORCA` leverages generative models to capture the full conditional distribution and introduces a rank-dependent optimization procedure that adaptively allocates coverage for efficiency while maintaining validity. We cast this coverage allocation as an optimization problem, derive an exact mixed-integer linear programming formulation, and give sufficient conditions for asymptotic convergence to the oracle density-level set. Across synthetic, semi-synthetic, and real datasets, `ORCA` produces substantially more efficient uncertainty sets than state-of-the-art baselines, demonstrating efficiency gains in multimodal and heterogeneous settings.

1 Introduction

Conformal prediction (CP) [1] has gained prominence as a powerful framework for uncertainty quantification, offering uncertainty sets with statistically guaranteed coverage under minimal assumptions. As a model-agnostic approach, CP can enhance any black-box model to provide uncertainty sets that contain the ground truth with a user-specified probability. This versatility has driven its adoption across diverse domains, including healthcare [2], finance [3, 4], and autonomous systems [5, 6, 7], where reliable decision-making under uncertainty is critical.

Despite its strengths in coverage validity and model-agnostic applicability, CP often yields overly conservative uncertainty sets, limiting practicality and interpretability when the conditional distribution $p(Y \mid X)$ exhibits complex structure such as multimodality or heterogeneity. This inefficiency arises from two sources: (i) limited model capacity to capture the geometry of the conditional distribution, and (ii) simplistic nonconformity scores that fail to exploit distributional richness. For instance, linear regression provides only conditional mean estimates, whereas quantile regression could recover multiple conditional quantiles and thus reveals more of the distribution. Likewise, for a regression model, one may choose absolute residual as the score, or a more sophisticated variant such as the locally weighted nonconformity measure that normalizes residuals to improve efficiency [8]. Both directions of using more expressive models and designing more refined scores enhance uncertainty quantification. Recent work has pushed along these axes; however, most approaches treat model expressiveness and score design in isolation, advancing one while neglecting the other. In this work, we unify these perspectives by proposing a framework that couples generative modeling with rank-dependent score constructions optimized for efficient coverage allocation.

In this paper, we propose generative conformal prediction with Optimized Ranking and Coverage Allocation (`ORCA`), a three-stage framework that combines generative modeling with optimization-

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

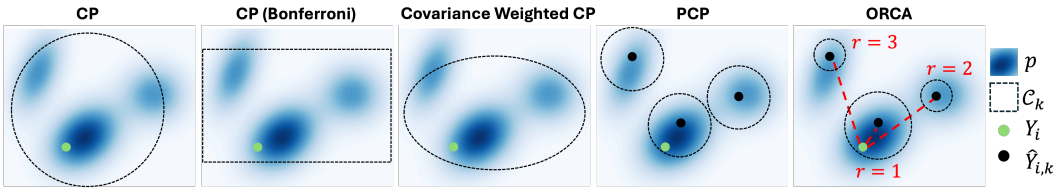


Figure 1: Comparison of conformal prediction methods and their prediction regions. From left to right: standard CP [1], Bonferroni-corrected CP [10], Covariance Weighted CP [11, 12], PCP [9], and ORCA (our method). The background gradient shows the multimodal data distribution. ORCA adjusts circle radii via ranking—larger in dense areas, smaller in sparse ones—yielding more efficient and flexible prediction sets. Green dots denote true responses (Y_i), black dots are generated samples ($\hat{Y}_{i,k}$), and dashed lines mark prediction boundaries ($\mathcal{C}_k$).

based, efficiency-oriented non-conformity score design. ORCA leverages generative samples to capture the full conditional distribution and introduces a rank-dependent optimization procedure that adaptively allocates coverage while maintaining finite-sample marginal validity. This yields prediction sets that adapt to the geometry of $Y \mid X$, improving efficiency. Figure 1 highlights our advances over classical conformal prediction baselines, illustrating how ORCA achieves a more adaptive uncertainty set. In summary, the main contributions are:

1. *Generative conformal prediction.* ORCA employs generative models as predictors to capture distributional complexity beyond mean- or quantile-based approaches. By leveraging generative sampling with local density proxy ranking mechanism, ORCA exploits the fitted conditional distribution, encoding finer distributional structure into the conformal procedure.
2. *Optimized coverage allocation.* Rather than assigning one shared radius to all generated samples as in PCP [9], ORCA learns rank-dependent radii, with PCP as the equal-radius special case. We cast conformal coverage as an explicit coverage allocation optimization problem and reformulate it as a mixed-integer linear program (MILP), yielding an exact formulation for the Stage 2 objective.
3. *Theory with oracle guarantees.* We prove finite-sample marginal validity and establish convergence to the oracle highest-density region under explicit generator, allocation, and calibration conditions.
4. *Empirical evaluation.* Experiments on synthetic and real datasets show that ORCA achieves more efficient uncertainty sets than baselines while maintaining coverage.

Related Work Extended works have been done focusing on CP for deterministic models. Different summary statistics of the conditional distribution have been studied, for example, mean [13, 14] and quantiles [15, 16, 17]. [18] estimate conditional distributions with histograms to obtain efficient intervals with approximate conditional coverage. Though the above methods all provide continuous prediction intervals, they cannot adapt to multimodality well. In [19] and [20], the authors first consider the aspect of density level sets and density-based clustering for providing efficient prediction set. The authors use kernel density estimator to construct the non-conformity scores and with a proper choice of bandwidth, the prediction set could be discontinuous. There are also following works [21, 22, 23] that estimate the density or cumulative distribution function to construct prediction sets. As a reminder, in our setting, we only need access to the random samples generated from the fitted model instead of explicit density estimation. So our setting is more generic and applicable to real applications. Other than density estimation, [24] fits shape functions (convex hull, hyperrectangle, and ellipsoid) on the calibration data to provide a discontinuous prediction set as a union of fitted shapes. In [25], the authors convert regression to a classification problem and can provide discontinuous prediction sets with interpolation of classification scores. Our construction also draws on work on multi-target regression and multivariate quantiles [26, 27].

Efficiency, measured by the size of the prediction set, is a fundamental assessment of the quality of a CP method. Some works explore the effect of different non-conformity score measures such as by scaling and reweighting the non-conformity score of each calibration data [8, 28, 16, 29, 30]. Other approaches improve efficiency during model training by incorporating differentiable measures of prediction-set size into the loss [31, 32]. Bai et al. [33] formulate conformal prediction as a constrained optimization problem. Related optimization-based methods target either calibration thresholds or score parameters. Kiyani et al. [34] optimize covariate-dependent thresholds for a fixed score under marginal or group-conditional coverage constraints. Cleaveland et al. [35] optimize weights in a joint multi-step time-series score through complementarity programming, followed by

independent conformal calibration. ORCA uses this optimization perspective to learn rank-dependent radii around generated samples, shaping the prediction region before final calibration.

Recent advancements, such as probabilistic conformal prediction (PCP) using conditional random samples [9], have paved the way for CP in generative models. PCP leverages generated random samples to construct the uncertainty sets in the form of a union of balls around each generated sample, offering a discontinuous uncertainty set. This is a fundamental step towards better coverage efficiency, as ideally, we want the prediction set to avoid low-density regions. However, PCP faces two major limitations: (i) PCP does not distinguish different generated samples and assigns the same radius to each uncertainty ball regardless of its local density, (ii) when the number of generated samples is relatively small, PCP can yield conservative prediction sets. ORCA keeps PCP's union-of-balls form but learns rank-dependent radii. CASA [36] complements ORCA by adapting the number of generated samples per input under a sampling budget.

2 Preliminaries

Conformal Prediction (CP) is a general framework for uncertainty quantification that provides statistically valid prediction regions for any predictive model under the mild assumption of data exchangeability [13, 37, 38]. For any test input X_{test}, CP aims to construct a prediction set $\hat{\mathcal{C}}(X_{\text{test}})$ that satisfies the coverage guarantee

$$\Pr\big(Y_{\text{test}} \in \hat{\mathcal{C}}(X_{\text{test}})\big) \geq 1 - \alpha, \tag{1}$$

for a user-specified miscoverage rate $\alpha \in (0, 1)$.

Given calibration dataset $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$ with $X_i \in \mathcal{X}$ are features and $Y_i \in \mathcal{Y}$ are responses, CP defines a *non-conformity score* E_i that measures how each sample conforms to fitted model $\hat{f}$. In regression, a common choice is the prediction error $E_i = \|Y_i - \hat{f}(X_i)\|$. The $\lceil (n+1)(1-\alpha) \rceil$-th order statistics of the calibration scores $\{E_i\}$ yields a threshold τ, and the prediction set is

$$\hat{\mathcal{C}}(X_{\text{test}}) = \{y : \|y - \hat{f}(X_{\text{test}})\| \leq \tau\}.$$

The validity of (1) requires only that $(X_{\text{test}}, Y_{\text{test}})$ be exchangeable with $\mathcal{D}$.

Prediction Set Efficiency. Beyond validity, *efficiency*, the size of $\hat{\mathcal{C}}(X_{\text{test}})$, is crucial: trivial sets covering all of $\mathcal{Y}$ achieve coverage but are uninformative. We therefore seek prediction sets that are valid and as small as possible.

3 Proposed Method

In this section, we introduce generative conformal prediction with Optimized Ranking and Coverage Allocation (ORCA), a framework that leverages generative models to construct efficient uncertainty sets with finite-sample coverage guarantees. Unlike classical conformal prediction methods that rely on point estimates or low-dimensional summaries, generative models provide samples from the full conditional distribution $\hat{p}(Y \mid X)$, enabling richer characterization of predictive uncertainty. Existing generative conformal approaches such as PCP [9] assign the same radius to every generated sample, which discards important distributional structure and often leads to inefficient prediction sets.

Our key idea is to preserve and exploit the geometric structure of the generated samples. In particular, we rank generated samples according to their local density and construct uncertainty sets as a union of rank-dependent regions, allowing different parts of the predictive distribution to receive different coverage allocations. Formally, the resulting prediction set takes the form

$$\hat{\mathcal{C}}_{\text{ORCA}}(X_{\text{test}}) = \bigcup_{r=1}^{K} \mathcal{B}_r(X_{\text{test}}; R_r(\alpha)), \tag{2}$$

where $R(\alpha) \in \mathbb{R}^K$ is a vector of rank-dependent radii, and $\mathcal{B}_r(X_{\text{test}}; R_r(\alpha))$ denotes a region centered at the rank r generated sample with radius $R_r(\alpha)$. When $Y \in \mathbb{R}$, these regions reduce to intervals, while for $Y \in \mathbb{R}^d$ they correspond to Euclidean balls; more generally, the regions can be defined using another output-space distance.

The proposed framework combines three ingredients: (i) distribution exploration through generative sampling and density-aware ranking, (ii) optimization-based allocation of coverage across ranked regions to maximize efficiency, and (iii) conformal calibration to guarantee finite-sample marginal coverage. These components combine density-based ranking and optimized coverage allocation with the marginal validity guarantee of conformal prediction.

Algorithm 1 ORCA

Input: generative model $\hat{p}(Y|X)$; splits $\mathcal{D}_{\text{explore}}, \mathcal{D}_{\text{calib}}$; test feature X_{test}; miscoverage α; sample size K
Stage 1: Exploration with Sampling and Ranking
for $(X_i, Y_i) \in \mathcal{D}_{\text{explore}}$ **do**
    Sample $\{\hat{Y}_{i,k}\}_{k=1}^{K} \sim \hat{p}(\cdot|X_i)$
    Rank samples by m-NN density (Eq. (3))
    Construct distance vector E_i via Eq. (4)
end for
Stage 2: Quantile Optimization
Solve optimization problem (6) (via MILP (7)) to obtain threshold $Q_r(\hat{\beta}_r)$ for each r.
Stage 3: Calibration
for $(X_i, Y_i) \in \mathcal{D}_{\text{calib}}$ **do**
    Compute nonconformity scores S_i (Eq. (8))
end for
Calibrate coverage threshold vector $R(\alpha)$ (Eq.(9))
Prediction: Draw $\{\hat{Y}_{\text{test},k}\}_{k=1}^{K} \sim \hat{p}(Y|X_{\text{test}})$ and rank, form $\hat{\mathcal{C}}_{\text{ORCA}}(X_{\text{test}})$ via Eq. (10).

3.1 Stage 1: Efficiency-Oriented Distributional Quantile Exploration

The goal of Stage 1 is to explore the geometry of the predictive distribution. It produces rank-dependent distance vectors by sampling from the fitted model and ranking the generated samples by local density proxies. These vectors retain information about local density and multimodality.

We begin with a holdout dataset $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{n_1+n_2}$, distinct from the training data, and partition it into two disjoint subsets: $\mathcal{D} = \mathcal{D}_{\text{explore}} \cup \mathcal{D}_{\text{calib}}$, with $|\mathcal{D}_{\text{explore}}| = n_1$ and $|\mathcal{D}_{\text{calib}}| = n_2$. We refer to these subsets as the *exploration split* and the *calibration split*, respectively. For each $(X_i, Y_i) \in \mathcal{D}_{\text{explore}}$, we draw K independent samples $\{\hat{Y}_{i,k}\}_{k=1}^{K}$ from $\hat{p}(\cdot \mid X_i)$ and compute each sample's average distance to its m nearest neighbors among the other samples, where $1 \le m < K$,

$$\rho_{i,k} := \frac{1}{m} \sum_{\ell=1}^{m} \big\| \hat{Y}_{i,k} - \hat{Y}_{i,k}^{[\ell]} \big\|,$$

where $\hat{Y}_{i,k}^{[\ell]}$ is the ℓ-th nearest neighbor of $\hat{Y}_{i,k}$ among the other generated samples. Smaller $\rho_{i,k}$ suggests higher local density. We rank the samples using a permutation π_i satisfying

$$\rho_{i,\pi_i(1)} \le \cdots \le \rho_{i,\pi_i(K)}, \tag{3}$$

and form the ranked distance vector $E_i = (E_{i,1}, \ldots, E_{i,K})$, where

$$E_{i,r} = \|Y_i - \hat{Y}_{i,\pi_i(r)}\|, \qquad r \in [K]. \tag{4}$$

The rationale of ranking and constructing the ranked distance vector is two-fold: (i) By retaining distances to all ranked samples rather than only the nearest one (as in PCP), we capture a richer structure of the model predictive distribution. (ii) The ranking stratifies distributional structure: small r emphasizes high-density regions where the model is confident, while large r reflects sparser regions, highlighting uncertainty and tail behavior.

3.2 Stage 2: Optimal Coverage Allocation

The goal of Stage 2 is to identify the optimal coverage structure and allocation across ranks by optimizing thresholds over the distance vectors from Stage 1. This efficiency-oriented optimization produces a rank-dependent threshold vector.

For an input X_i, let $\hat{Y}_{i,\pi_i(1)}, \ldots, \hat{Y}_{i,\pi_i(K)}$ denote the generated samples ordered by the density-aware ranking rule from Stage 1. For any radius $t \ge 0$, define the rank-r region

$$\mathcal{B}_r(X_i; t) := \Big\{ y \in \mathcal{Y} : \|y - \hat{Y}_{i,\pi_i(r)}\| \le t \Big\}. \tag{5}$$

The remaining question is how to assign a radius to each rank to maximize the prediction set efficiency while retaining coverage guarantees. To this end, consider an allocation vector $\beta = (\beta_1, \ldots, \beta_K)^\top$. For each rank r, let $Q_r\big(\{E_{1,r}, \ldots, E_{n_1,r}\}; \beta_r\big)$ denote the empirical $(1-\beta_r)$-quantile of the rank-r distance distribution obtained in Stage 1. For notational simplicity, we write $Q_r(\beta_r)$ when it is clear from context. The allocation vector β induces the rank-dependent prediction set $\bigcup_{r=1}^{K} \mathcal{B}_r\big(X; Q_r(\beta_r)\big)$. We use the convention $Q_r(1) = 0$ to permit zero-radius allocations.

Quantile Optimization. Ideally, coverage allocation would minimize the volume of the resulting prediction region. However, directly optimizing this volume requires accounting for overlaps among the candidate-centered balls, which couples the rank-specific radii and makes the optimization difficult in general. We therefore use the sum of the individual ball volumes as a tractable surrogate that provides an upper bound on the union volume,

$$\left|\bigcup_{r=1}^{K} \mathcal{B}_r\big(X; Q_r(\beta_r)\big)\right| \le v_d \sum_{r=1}^{K} Q_r(\beta_r)^d,$$

where $d = \dim(Y)$ and v_d is the volume of the unit ball in $\mathbb{R}^d$. Omitting the common factor v_d, we minimize $\sum_{r=1}^{K} Q_r(\beta_r)^d$, a separable objective that assigns a cost to each rank. Appendix C.4 analyzes the gap between this surrogate and the union volume. The radii are selected jointly to ensure that at least a $1-\alpha$ fraction of the exploration observations are covered by at least one rank-dependent region, i.e.,

$$\frac{1}{n_1}\sum_{i=1}^{n_1} \hat{S}_i(\beta) \;\ge\; 1-\alpha, \qquad \hat{S}_i(\beta) = \max_{r\in[K]} \mathbb{1}\{E_{i,r} \le Q_r(\beta_r)\}.$$

The resulting optimization problem is

$$\min_{\beta} \sum_{r=1}^{K} (Q_r(\beta_r))^d \quad \text{s.t.} \quad \frac{1}{n_1}\sum_{i=1}^{n_1} \hat{S}_i(\beta) \;\ge\; 1-\alpha. \tag{6}$$

Intuitively, to minimize set size, the optimizer concentrates coverage in high-density regions, where small radii capture substantial probability mass. In contrast, achieving the same coverage in low-density regions requires much larger radii and is therefore inefficient. The vector β thus controls how coverage is allocated across density ranks, providing flexibility beyond a common radius.

Equivalent MILP Reformulation. While the optimization (6) is conceptually simple, solving it directly by grid search requires evaluating up to $(n_1+1)^K$ threshold configurations, which is computationally prohibitive. To address this, we reformulate the problem as a mixed integer linear program (MILP). This reformulation substantially enhances the scalability of our method. Theoretically, it is equivalent to the original problem and admits global optimality certificates from modern MILP solvers. The MILP is substantially faster than the naive search procedure in Table 6, while the LP and greedy variants offer further runtime reductions.

For each rank $r \in \{1,\dots,K\}$, let $E_{(\ell),r}$ denote the ℓ-th order statistic of the set $\{E_{i,r}\}_{i=1}^{n_1}$, so that $E_{(1),r} \le E_{(2),r} \le \cdots \le E_{(n_1),r}$. These serve as the candidate thresholds for $Q_r(\beta_r)$. We define incremental costs

$$c_{r,\ell} := E^d_{(\ell),r} - E^d_{(\ell-1),r}, \qquad \ell = 2,\dots,n_1,$$

with $c_{r,1} := E^d_{(1),r}$, and coverage indicators

$$a_{i,r,\ell} := \mathbb{1}\{E_{i,r} \le E_{(\ell),r}\}, \quad i \in [n_1],\ \ell \in [n_1].$$

Let $\tau = \lceil n_1(1-\alpha)\rceil$ denote the required number of points to cover. We then introduce two binary variables: (i) $z_{r,\ell} = 1$ if $E_{(\ell),r}$ is covered by the chosen threshold and 0 otherwise; by construction, if $z_{r,\ell} = 1$ then all lower levels $\{1,\dots,\ell-1\}$ must also be 1; (ii) y_i may equal 1 only if exploration observation i is covered by at least one rank-dependent region. The MILP problem can then be rewritten as:

$$\min_{z,y} \;\sum_{r=1}^{K}\sum_{\ell=1}^{n_1} c_{r,\ell}\, z_{r,\ell} \tag{7a}$$

$$\text{s.t. } y_i \le \sum_{r=1}^{K}\sum_{\ell=1}^{n_1} a_{i,r,\ell}\, z_{r,\ell}\ (\forall i \in [n_1]),\ \sum_{i=1}^{n_1} y_i \ge \tau \tag{7b}$$

$$z_{r,\ell} \ge z_{r,\ell+1},\ z_{r,\ell'} \in \{0,1\},\ y_i \in \{0,1\} \quad (\forall r \in [K],\ \ell \in [n_1-1],\ \ell' \in [n_1],\ i \in [n_1]). \tag{7c}$$

Remark 3.1 (Exact MILP Reformulation of (6)). Let ℓ_r^* denote the selected threshold index at rank r, with $E_{(0),r} := 0$. Then by summing increments,

$$\sum_{r=1}^{K}\sum_{\ell=1}^{n_1} c_{r,\ell}\, z_{r,\ell} = \sum_{r=1}^{K}\sum_{\ell\le\ell_r^*} \big(E^d_{(\ell),r} - E^d_{(\ell-1),r}\big) = \sum_{r=1}^{K} E^d_{(\ell_r^*),r}.$$

We observe the MILP objective (7a) coincides with $\sum_{r=1}^{K}(Q_r(\beta_r))^d$ in (6). Constraint (7b) encodes the coverage constraint that at least τ points are covered. Constraint (7c) enforces monotonicity across thresholds and integrality of the decision variables. Hence the MILP (7) is an *exact reformulation* of (6) with identical optimal value. This yields Theorem 3.2, with proof deferred to Appendix C.

Theorem 3.2 (Exact MILP Reformulation and Optimality)**.** *The MILP formulation* (7) *is equivalent to optimization* (6) *and has the same optimal objective value. In particular, the feasible radius vectors* $\big(Q_r(\beta_r)\big)_{r=1}^{K}$ *in* (6) *coincide with those induced by feasible binary assignments* (z, y) *in* (7)*.*

Efficient MILP Approximations. The MILP in (7) exactly solves the Stage 2 proxy optimization, but its worst-case runtime is exponential in the number of binary variables, which is on the order of Kn_1. We therefore consider two scalable alternatives. The first, `ORCA`-LP, is a linear-programming (LP) relaxation of the MILP: it relaxes $(z, y) \in \{0, 1\}$ to $(z, y) \in [0, 1]$, yielding an LP of polynomial size with $O(Kn_1)$ variables and $O(Kn_1^2)$ nonzero constraint coefficients, and can be solved in polynomial time. The second, `ORCA`-Greedy, interprets (7) as a weighted partial set-cover problem over rank–level jumps: starting from minimal radii, it repeatedly selects the jump with the largest marginal number of newly covered exploration samples per unit proxy-volume increase, followed by a pruning step that removes redundant expansions. It has worst-case complexity $O(n_1^3K^2)$, but is substantially faster in practice; for thousands of exploration samples, it typically runs within tens of seconds in our experiments. Full algorithmic details and the greedy approximation guarantee are deferred to Appendix D. Importantly, `ORCA`'s finite-sample validity does not depend on solving (7) exactly: Stage 2 only selects a rank-dependent radius direction, while Stage 3 applies split-conformal calibration along that direction. Thus `ORCA`-MILP, `ORCA`-LP, and `ORCA`-Greedy all preserve marginal coverage after calibration, differing only in efficiency and computational cost.

3.3 Stage-3: Conformal Calibration with Coverage Guarantees

Let $\hat{\beta}$ denote the solution of the Stage 2 optimization. In Stage 3, holding $\hat{\beta}$ fixed, we use $\mathcal{D}_{\text{calib}}$ to perform the final conformal calibration. Define the nonconformity scores

$$S_i := \min_{r \in [K]} \frac{E_{i,r}}{Q_r(\hat{\beta}_r)}, \qquad i \in \mathcal{D}_{\text{calib}}. \tag{8}$$

Using $0/0 = 0$ and $a/0 = \infty$ for $a > 0$, the score is the smallest common multiplier of the Stage 2 radii needed to include the calibration response. We then obtain the final coverage radii $R(\alpha) = (R_1(\alpha), \ldots, R_K(\alpha))^\top$, with each $R_r(\alpha) := t_\alpha\, Q_r(\hat{\beta}_r)$, where

$$t_\alpha := \min\left\{ t : \frac{1}{n_2 + 1} \sum_{i=1}^{n_2} \mathbb{1}\{S_i \le t\} \;\ge\; 1 - \alpha \right\}. \tag{9}$$

If no finite threshold satisfies (9), set $t_\alpha = \infty$ and return $\mathcal{Y}$. The resulting prediction set takes the form

$$\hat{\mathcal{C}}_{\text{ORCA}}(X_{\text{test}}) = \bigcup_{r=1}^{K} \mathcal{B}_r(X_{\text{test}}; R_r(\alpha)) = \bigcup_{r=1}^{K} \Big\{ y : \; \|y - \hat{Y}_{\text{test},\hat{\pi}(r)}\| \le R_r(\alpha) \Big\}. \tag{10}$$

where $\hat{\pi}$ denotes the density-based ordering of the generated test samples. If all Stage 2 radii are equal and positive, the score (8) is proportional to the nearest-sample distance and the ranking plays no role, so `ORCA` recovers PCP exactly; see Appendix B.

Theorem 3.3 (Validity of `ORCA`)**.** *Conditional on the fitted generator and exploration data, suppose that the calibration and test observations, augmented with their generated candidates, are exchangeable. Then the prediction set* $\hat{\mathcal{C}}_{ORCA}(X_{test})$ *satisfies*

$$\Pr\big(Y_{test} \in \hat{\mathcal{C}}_{\text{ORCA}}(X_{test})\big) \;\ge\; 1 - \alpha.$$

The proof follows from the standard conformal prediction argument with the exchangeability assumption. The full proof is deferred to Appendix C.

3.4 Asymptotics and Level-Set Limit

We distinguish density ordering under a fitted generator from convergence to the true highest-density region (HDR). Proofs are given in Appendix C. Throughout this subsection, fix an input x.

Assumption 1 (Candidate sampling and density regularity)**.** Conditional on the training data and $X_i = x$, the candidates $\{\hat{Y}_{i,k}\}_{k=1}^{K}$ are i.i.d. from $\hat{p}(\cdot \mid x)$. This density is continuous and bounded away from zero and infinity on a compact support $\mathcal{Y} \subset \mathbb{R}^d$ whose boundary has zero volume. The statistic $\rho_{i,k}$ excludes the candidate itself, and $m \to \infty$ with $m/K \to 0$.

Proposition 3.4 (Density-order consistency)**.** *Under Assumption 1, for a fixed fitted generator and any fixed candidate indices $k \neq \ell$,*

$$\frac{\rho_{i,k}}{\rho_{i,\ell}} = \left(\frac{\hat{p}(\hat{Y}_{i,\ell} \mid x)}{\hat{p}(\hat{Y}_{i,k} \mid x)} \right)^{1/d} \big(1 + o_{\mathbb{P}}(1)\big).$$

Consequently, for every fixed $\delta > 0$,

$$\Pr\Big(\rho_{i,k} \geq \rho_{i,\ell},\ \hat{p}(\hat{Y}_{i,k} \mid x) \geq \hat{p}(\hat{Y}_{i,\ell} \mid x) + \delta \Big) \to 0.$$

Assumption 2 (Generator and allocation consistency)**.** As the training sample size grows, $\sup_{y \in \mathcal{Y}} |\hat{p}(y \mid x) - p(y \mid x)| \xrightarrow{\mathbb{P}} 0$, where $p(\cdot \mid x)$ is continuous on the common compact support $\mathcal{Y}$. The true $(1-\alpha)$ HDR has a unique positive cutoff $\tau^\star(x)$ with $|\{y \in \mathcal{Y} : p(y \mid x) = \tau^\star(x)\}| = 0$. As $K, n_1, n_2 \to \infty$, the calibrated sets have conditional probability at least $1 - \alpha - o_{\mathbb{P}}(1)$ and volume no larger than the fitted-density HDR up to $o_{\mathbb{P}}(1)$, as formalized in Appendix C.5. Here $n_1 = |\mathcal{D}_{\text{explore}}|$ and $n_2 = |\mathcal{D}_{\text{calib}}|$, with $\log K / n_1 \to 0$, so that the rank-wise empirical distribution functions converge uniformly over all K ranks (DKW inequality with a union bound).

Theorem 3.5 (Convergence to the optimal HDR)**.** *Under Assumptions 1 and 2, the calibrated prediction set $\mathcal{C}_{K,n_1,n_2}(x)$ satisfies*

$$|\mathcal{C}_{K,n_1,n_2}(x) \triangle L_{\tau^\star}(x)| \xrightarrow{\mathbb{P}} 0,$$

where $L_{\tau^\star}(x) = \{y \in \mathcal{Y} : p(y \mid x) \geq \tau^\star(x)\}$ and $\int_{L_{\tau^\star}(x)} p(y \mid x)\, dy = 1 - \alpha$.

To isolate generator misspecification, we compare a fitted-density level set with the oracle HDR at the same true probability content.

Corollary 3.6 (Cost of generator error)**.** *Fix x, let $\varepsilon = \sup_{y \in \mathcal{Y}} |\hat{p}(y \mid x) - p(y \mid x)|$, and let $\tilde{L}(x) = \{y : \hat{p}(y \mid x) \geq \tilde{\tau}\}$ with $\int_{\tilde{L}(x)} p(y \mid x)\, dy = 1 - \alpha$. Suppose $t \mapsto \int_{\{p \geq t\}} p(y \mid x)\, dy$ is strictly decreasing near $\tau^\star = \tau^\star(x)$ and $|\{y : |p(y \mid x) - \tau^\star| \leq t\}| \leq C t^\gamma$ for $0 < t \leq t_0$. If $2\varepsilon \leq t_0$, then*

$$\big|\tilde{L}(x) \triangle L_{\tau^\star}(x)\big| \leq C(2\varepsilon)^\gamma, \qquad 0 \leq |\tilde{L}(x)| - |L_{\tau^\star}(x)| \leq \frac{C\,(2\varepsilon)^{1+\gamma}}{\tau^\star}.$$

For smooth p with nonvanishing gradient on $\{p = \tau^\star\}$, $\gamma = 1$, giving $O(\varepsilon^2)$ population excess volume. The proof is in Appendix C.6.

Proposition 3.4 justifies ranking by $\rho_{i,k}$ for pairs with separated fitted-density values. Theorem 3.5 combines this ranking property with generator consistency and with allocation and calibration conditions, stated in Appendix C.5, under which the calibrated union attains the HDR mass and volume. Appendix C.4 relates the additive Stage 2 objective to the union volume.

4 Experiments

In this section, we evaluate the empirical performance of `ORCA` compared with several state-of-the-art baselines on six synthetic datasets, one semi-synthetic dataset, and nine real-world datasets. We demonstrate the superior performance of `ORCA`, especially in handling complex distributions.

Baselines We compare against state-of-the-art conformal prediction methods: Conformalized Quantile Regression (CQR) [15], Conformal Histogram Regression (CHR) [18], Distributional Conformal Prediction (DCP) [22], DistSplit and CDSplit [21], Probabilistic Conformal Prediction (PCP) [9], Conformal Region Designer (CRD) [24], and Regression-to-Classification Conformal Prediction (R2CCP) [25]. CQR and CHR output continuous sets via conditional quantiles and conditional histograms. DCP, DistSplit, and CDSplit incorporate estimated distributions/densities into conformalization. The remaining methods yield discontinuous sets: R2CCP reduces regression to classification and interpolates class scores, while CRD optimizes simple geometric templates (e.g., hyperrectangles, ellipsoids, convex hulls) to reduce volume. We use the authors' public implementations for CQR, CHR, R2CCP, CRD, and PCP; for DCP, DistSplit, and CDSplit we use the public implementation provided in [9].

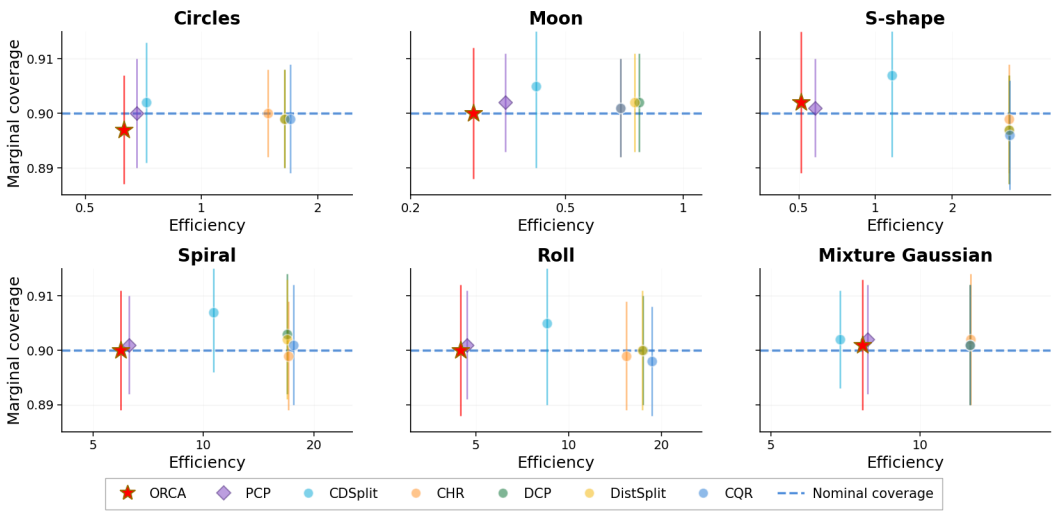


Figure 2: Empirical coverage and uncertainty-set efficiency on synthetic datasets. Each panel reports marginal coverage versus prediction-set efficiency on a log scale; smaller efficiency is better. The dashed horizontal line marks the nominal coverage level 0.90, and vertical bars show the standard error for marginal coverage. ORCA performs best overall.

Implementation Configuration Throughout the experiments, we set $\alpha = 0.1$ and evaluate methods by marginal coverage (Cover), worst-slab conditional coverage (Cond_Cover), and prediction set size (Eff). Unless otherwise specified, we use $K = 20$ ($K = 50$ for Bike) and $m = \lceil \sqrt{K} \rceil$ for ranking. All baselines, including PCP, are calibrated on $\mathcal{D}_{\text{explore}} \cup \mathcal{D}_{\text{calib}}$, so every method uses the same labeled data; ORCA splits these data between Stage 2 and Stage 3. For baseline modeling and conditional sample generation, we use quantile regression forests [39] and mixture density networks [40, 41]. Implementation details are given in Appendix E, and code is available at `https://github.com/Minxing-Zheng/ORCA`.

4.1 Synthetic Data

We benchmark on six synthetic datasets (Circles, Moon, S-shape, Spiral, Roll, Mixture Gaussian) with multimodal distributions. Models are trained on 3,000 points; we use 500 for exploration, 500 for calibration, and 1,000 for testing. Results are averaged over 30 independent runs. Figure 2 summarizes empirical coverage and uncertainty-set efficiency across datasets. All methods achieve near-nominal coverage, while ORCA attains the smallest or near-smallest uncertainty sets on nearly all datasets. Full numerical results are provided in Appendix F.1, Table 3; additional visualizations are provided in Figure 6. The visualization shows that both PCP and ORCA produce discontinuous uncertainty sets adapted to multimodal distributions, while ORCA further improves efficiency through locally adaptive radii from optimized coverage allocation.

4.2 MNIST: Digit Image Generation

We evaluate ORCA on high-dimensional outputs using a semi-synthetic MNIST setup [42]. The input is the digit label $X \in \{0, \ldots, 9\}$ and the response Y is an image of digit X. We train a VAE and construct uncertainty sets in the latent space. Figure 3 shows random samples from PCP and ORCA uncertainty sets with $K = 40$ across digits (rows). Both methods attain nominal coverage, but ORCA's sampled images are visually closer to the target digit, whereas PCP more often includes less similar (and occasionally off-class) samples, indicating tighter uncertainty sets under ORCA. Since set size in pixel space is not directly interpretable, we report the *off-class rate (OCR)*, defined as the probability that a non-X digit appears in the prediction set. Averaged over 30 runs, ORCA achieves a lower OCR than PCP, 0.16 versus 0.22.

4.3 Real Data

We evaluate ORCA on standard benchmarks, including Blog [43], Facebook Comment Volume (Fb1, Fb2) [44], MEPS (2019–2021), Protein Tertiary Structure (Bio), and Temperature Forecasts [45], as well as NYC bike-sharing for 2D destination prediction. For each dataset, we allocate 500 samples for exploration, 500 for calibration, 1,000 for testing, and use the remaining samples for training.

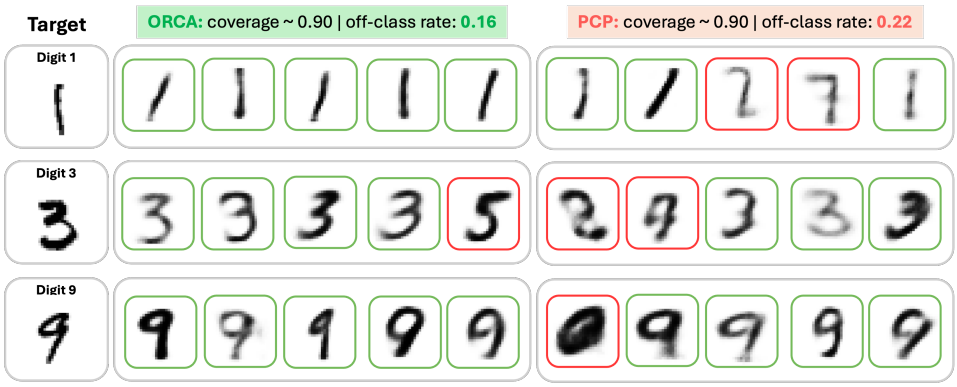


Figure 3: **MNIST digit generation coverage.** Samples from PCP and ORCA uncertainty sets across target digits (rows). ORCA yields more semantically consistent samples with target digit number and a lower off-class rate (OCR) than PCP (0.16 ± 0.02 vs. 0.22 ± 0.03).

Consistent with [18], we find DCP unstable on real data; we therefore use DCP-CQR, a hybrid variant that stabilizes DCP with CQR. Figure 4 reports representative real-data results, and the complete numerical results are deferred to Table 4 in Appendix F.2.

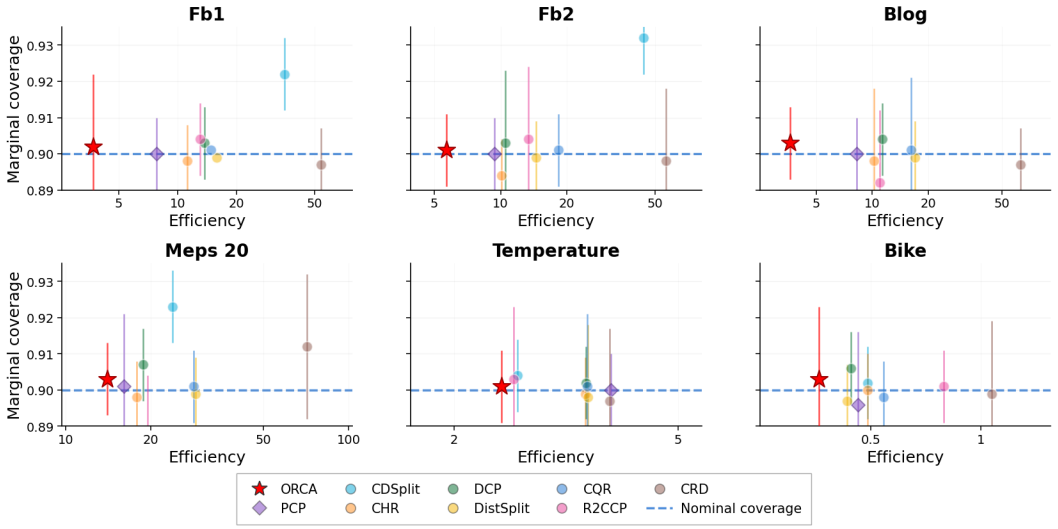
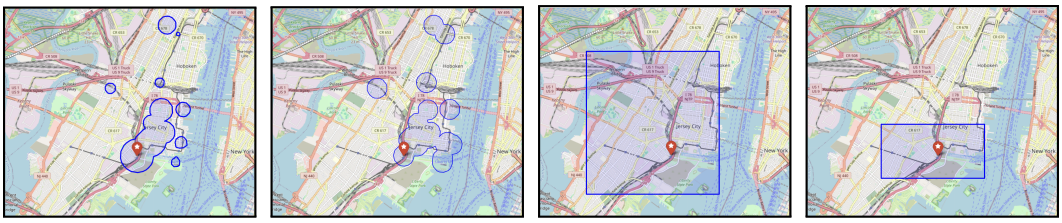


Figure 4: Empirical coverage and uncertainty-set efficiency on real datasets. Each panel reports marginal coverage versus prediction-set efficiency on a log scale; smaller efficiency is better. The dashed horizontal line marks the nominal coverage level 0.90, and vertical bars show the standard error for marginal coverage. ORCA performs best overall.

Figure 4 shows that all methods achieve coverage close to the nominal level, while ORCA yields the smallest prediction sets on most datasets. The gains are especially pronounced on complex datasets such as Blog, where the conditional distribution $P(Y \mid X)$ is highly structured and benefits from rank-dependent coverage allocation. By assigning larger radii to high-density regions and shrinking radii in low-density regions, ORCA avoids unnecessary expansion into sparse areas. In contrast, methods based on a single global scale or simple geometric templates tend to over-cover low-density regions in order to maintain marginal coverage, leading to larger sets. Figure 5 further illustrates this behavior on NYC bike destination prediction. Both PCP and ORCA produce discontinuous prediction regions as unions of local balls, whereas CHR and R2CCP return continuous rectangular regions. Such discontinuity is desirable in this setting because destinations naturally cluster around transit hubs, with low-density regions in between. Compared with PCP, ORCA uses optimized coverage allocation across ranks to suppress unnecessary expansion around low-density generated samples, producing tighter and more targeted uncertainty sets without sacrificing coverage.



(a) ORCA (b) PCP (c) CHR (d) R2CCP

Figure 5: Comparison of prediction sets across methods with the highest efficiency on the Bike dataset. The colored region indicates the set, and the red marker denotes the target location.

4.4 Sensitivity analysis

ORCA relies on two components chosen by the practitioner: the conditional generator, which supplies the candidate samples, and the ranking statistic, which orders them by local density. We examine sensitivity to each, using the S-shape data as a representative example (Table 1). Varying the QRF minimum leaf size from 1 to 200 moves the fitted generator from overfitting to oversmoothing. Marginal coverage remains near the nominal level throughout, while ORCA yields sets 13–16% smaller than PCP at intermediate leaf sizes; the advantage disappears at both extremes. These results complement the population analysis in Corollary 3.6: generator quality influences efficiency, whereas conformal calibration preserves marginal validity. For ranking, small neighborhoods ($m \leq 4$) give noisy local density estimates that order the samples unreliably, and ORCA then produces larger sets than PCP; for $m \geq 7$, the MILP and LP variants reduce set size by 16–32%, and the gains are stable across this range. A reasonably well-fitted generator and a moderate number of neighbors therefore suffice in practice. Both K and m can be tuned on the exploration split alone, which leaves the calibration split untouched and the coverage guarantee intact. For the Stage 2 solver, ORCA-Greedy nearly matches the efficiency of the exact MILP at a small fraction of its runtime (Appendix G), so we recommend it by default and the MILP when a certified optimum is needed.

Table 1: **Sensitivity analysis.** (a) Minimum leaf size of the QRF generator. (b) Number of neighbors m used for density ranking, with Stage 2 solved by MILP or LP. ORCA/PCP denotes the ratio of mean set sizes on the S-shape data; values below 1 favor ORCA.

(a) Leaf size	1	20	50	100	200
Coverage	0.90	0.89	0.90	0.90	0.90
ORCA/PCP	1.00	0.84	0.87	0.85	1.03

(b) m	1	4	7	10	13	16	19
MILP	1.02	1.20	0.74	0.68	0.68	0.71	0.72
LP	1.73	2.13	0.84	0.77	0.73	0.71	0.72

5 Discussion and Future Work

ORCA combines generative modeling with rank-adaptive coverage allocation, solved exactly by a MILP or approximately by scalable LP and greedy variants, to construct flexible prediction regions with finite-sample marginal coverage. The coverage guarantee is marginal and does not ensure $1 - \alpha$ coverage at every input or within every subgroup. Pooling allocation and calibration lets coverage vary across input regions, as the worst-slab results for Fb1 and Bio in Table 4 show.

Several directions remain for future work. First, in high-dimensional output spaces, the choice of metric is important, since Euclidean distance may not reflect semantic similarity; learned metrics may further improve performance. Second, an objective that accounts for overlap among rank-dependent regions, rather than the additive volume surrogate, could further improve efficiency. Third, group-specific coverage constraints and groupwise conformal calibration could address coverage heterogeneity, with additional calibration-data requirements.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science Foundation under Grant No. CAIG-2425888. The authors declare no competing interests.

References

[1] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

[2] Silvia Seoni, Vicnesh Jahmunah, Massimo Salvi, Prabal Datta Barua, Filippo Molinari, and U. Rajendra Acharya. Application of uncertainty quantification to artificial intelligence in healthcare: A review of last decade (2013–2023). *Computers in Biology and Medicine*, 165:107441, 2023.

[3] Yugo Fujimoto, Kei Nakagawa, Kentaro Imajo, and Kentaro Minami. Uncertainty aware trader-company method: Interpretable stock price prediction capturing uncertainty. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1238–1245, 2022.

[4] Txus Blasco, J. Salvador Sánchez, and Vicente García. A survey on uncertainty quantification in deep learning for financial time series prediction. *Neurocomputing*, 576:127339, 2024.

[5] Rhiannon Michelmore, Matthew Wicker, Luca Laurenti, Luca Cardelli, Yarin Gal, and Marta Kwiatkowska. Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7344–7350, 2020.

[6] Sanbao Su, Yiming Li, Sihong He, Songyang Han, Chen Feng, Caiwen Ding, and Fei Miao. Uncertainty quantification of collaborative detection for self-driving. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5588–5594. IEEE, 2023.

[7] Ruben Grewal, Paolo Tonella, and Andrea Stocco. Predicting safety misbehaviours in autonomous driving systems using uncertainty quantification. *arXiv preprint arXiv:2404.18573*, 2024.

[8] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Machine Learning: ECML 2002*, pages 345–356, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[9] Zhendong Wang, Ruijiang Gao, Mingzhang Yin, Mingyuan Zhou, and David Blei. Probabilistic conformal prediction using conditional random samples. volume 206 of *Proceedings of Machine Learning Research*, pages 8814–8836. PMLR, 25–27 Apr 2023.

[10] Jelmer Neeven and Evgueni Smirnov. Conformal stacked weather forecasting. In Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov, and Ralf Peeters, editors, *Proceedings of the Seventh Workshop on Conformal and Probabilistic Prediction and Applications*, volume 91 of *Proceedings of Machine Learning Research*, pages 220–233. PMLR, 11–13 Jun 2018.

[11] Soundouss Messoudi, Sébastien Destercke, and Sylvain Rousseau. Ellipsoidal conformal inference for multi-target regression. In *Conformal and Probabilistic Prediction with Applications*, pages 294–306. PMLR, 2022.

[12] Chen Xu, Hanyang Jiang, and Yao Xie. Conformal prediction for multi-dimensional time series by ellipsoidal sets. In *Forty-first International Conference on Machine Learning*, 2024.

[13] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

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

[15] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[16] Danijel Kivaranovic, Kory D Johnson, and Hannes Leeb. Adaptive, distribution-free prediction intervals for deep networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4346–4356. PMLR, 2020.

[17] Ahmed M. Alaa, Zeshan Hussain, and David Sontag. Conformalized unconditional quantile regression. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 10690–10702. PMLR, 25–27 Apr 2023.

[18] Matteo Sesia and Yaniv Romano. Conformal prediction using conditional histograms. *Advances in Neural Information Processing Systems*, 34:6304–6315, 2021.

[19] Jing Lei, James Robins, and Larry Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108(501):278–287, 2013.

[20] Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):71–96, 2014.

[21] Rafael Izbicki, Gilson Shimizu, and Rafael Stern. Flexible distribution-free conditional predictive bands using density estimators. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3068–3077. PMLR, 26–28 Aug 2020.

[22] Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118, 2021.

[23] Rafael Izbicki, Gilson Shimizu, and Rafael B. Stern. Cd-split and hpd-split: Efficient conformal regions in high dimensions. *Journal of Machine Learning Research*, 23(87):1–32, 2022.

[24] Renukanandan Tumu, Matthew Cleaveland, Rahul Mangharam, George Pappas, and Lars Lindemann. Multi-modal conformal prediction regions by optimizing convex shape templates. In *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, volume 242 of *Proceedings of Machine Learning Research*, pages 1343–1356. PMLR, 15–17 Jul 2024.

[25] Etash Kumar Guha, Shlok Natarajan, Thomas Möllenhoff, Mohammad Emtiyaz Khan, and Eugene Ndiaye. Conformal prediction via regression-as-classification. In *The Twelfth International Conference on Learning Representations*, 2024.

[26] Soundouss Messoudi, Sébastien Destercke, and Sylvain Rousseau. Copula-based conformal prediction for multi-target regression. *Pattern Recognition*, 120:108101, 2021.

[27] Sophia Huiwen Sun and Rose Yu. Copula conformal prediction for multi-step time series prediction. In *The Twelfth International Conference on Learning Representations*, 2024.

[28] Harris Papadopoulos, Vladimir Vovk, and Alex Gammerman. Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research*, 40:815–840, 2011.

[29] Anthony Bellotti. Constructing normalized nonconformity measures based on maximizing predictive efficiency. In Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov, and Giovanni Cherubin, editors, *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 128 of *Proceedings of Machine Learning Research*, pages 41–54. PMLR, 09–11 Sep 2020.

[30] Salim I Amoukou and Nicolas JB Brunel. Adaptive conformal prediction by reweighting nonconformity score. *arXiv preprint arXiv:2303.12695*, 2023.

[31] David Stutz, Ali Taylan Cemgil, Arnaud Doucet, et al. Learning optimal conformal classifiers. *arXiv preprint arXiv:2110.09192*, 2021.

[32] Bat-Sheva Einbinder, Yaniv Romano, Matteo Sesia, and Yanfei Zhou. Training uncertainty-aware classifiers with conformalized deep learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[33] Yu Bai, Song Mei, Huan Wang, Yingbo Zhou, and Caiming Xiong. Efficient and differentiable conformal prediction with general function classes. *arXiv preprint arXiv:2202.11091*, 2022.

[34] Shayan Kiyani, George Pappas, and Hamed Hassani. Length optimization in conformal prediction. In *Advances in Neural Information Processing Systems*, volume 37, pages 99519–99563. Curran Associates, Inc., 2024.

[35] Matthew Cleaveland, Insup Lee, George J. Pappas, and Lars Lindemann. Conformal prediction regions for time series using linear complementarity programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20984–20992, 2024.

[36] Minxing Zheng and Shixiang Zhu. Sample-efficient generative conformal prediction. *arXiv preprint arXiv:2610.11349*, 2026.

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

[38] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

[39] Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(35):983–999, 2006.

[40] Christopher M Bishop. Mixture density networks. 1994.

[41] Jonas Rothfuss, Fabio Ferreira, Simon Walther, and Maxim Ulrich. Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv:1903.00954*, 2019.

[42] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[43] Krisztian Buza. BlogFeedback. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C58S3F.

[44] Kamaljot Singh. Facebook Comment Volume. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C5Q886.

[45] Dongjin Cho, Cheolhee Yoo, Jungho Im, and Dong-Hyun Cha. Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas. *Earth and Space Science*, 7(4):e2019EA000740, 2020.

[46] Don O Loftsgaarden and Charles P Quesenberry. A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics*, 36(3):1049–1051, 1965.

[47] Puning Zhao and Lifeng Lai. Analysis of knn density estimation. *IEEE Transactions on Information Theory*, 68(12):7971–7995, 2022.

[48] Krisztian Buza. BlogFeedback. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C58S3F.

[49] Kamaljot Singh. Facebook Comment Volume. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C5Q886.

[50] Prashant Rana. Physicochemical Properties of Protein Tertiary Structure. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5QW3H.

[51] Bias correction of numerical prediction model temperature forecast. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C59K76.

A Average m-Nearest Neighbor Distance

For each input X_i, the ranking statistic $\rho_{i,k}$ in Stage 1 is computed from the generated samples $\{\hat{Y}_{i,k}\}_{k=1}^{K}$ as follows:

1. Compute the pairwise distances between the generated samples. Exclude the distance from each sample to itself, leaving $K-1$ distances for each sample.
2. For each $\hat{Y}_{i,k}$, order the other samples by their distance to $\hat{Y}_{i,k}$. Write $\hat{Y}_{i,k}^{[\ell]}$ for its ℓ-th nearest neighbor, so that
$$\|\hat{Y}_{i,k} - \hat{Y}_{i,k}^{[1]}\| \le \cdots \le \|\hat{Y}_{i,k} - \hat{Y}_{i,k}^{[K-1]}\|.$$
3. Average the first m distances, with $1 \le m < K$,
$$\rho_{i,k} := \frac{1}{m}\sum_{\ell=1}^{m} \|\hat{Y}_{i,k} - \hat{Y}_{i,k}^{[\ell]}\|.$$

The permutation π_i orders the generated samples by increasing $\rho_{i,k}$, as in (3). The ranked distance vector is $E_i = (E_{i,1}, \ldots, E_{i,K})$, with $E_{i,r} = \|Y_i - \hat{Y}_{i,\pi_i(r)}\|$. Ties are resolved using a fixed rule.

Alternatively, one may draw M independent reference samples from $\hat{p}(\cdot \mid X_i)$ for ranking. In that case, $\hat{Y}_{i,k}^{[\ell]}$ denotes the ℓ-th nearest reference sample to $\hat{Y}_{i,k}$, and the same averaging formula applies for $1 \le m \le M$. The construction in the main text uses the other $K-1$ generated samples.

B Comparison with PCP

Probabilistic Conformal Prediction (PCP) [9] first studied CP for generative models and utilized the information of random samples. In contrast to our method, which considers all the distances between each random sample and the true response, PCP defines its non-conformity score as the minimum distance between random samples and the true response:

$$S_i^{\mathrm{PCP}} = \min_{k=1,\ldots,K} \|Y_i - \hat{Y}_{i,k}\|. \tag{11}$$

for each calibration data (X_i, Y_i). For a new test point $(X_{\text{test}}, Y_{\text{test}})$, the prediction set of PCP is constructed as the union of K regions:

$$\hat{\mathcal{C}}_{\mathrm{PCP}}(X_{\text{test}}) = \cup_{k=1}^{K} \hat{\mathcal{C}}_k, \tag{12}$$

where each region $\hat{\mathcal{C}}_k$ is defined by

$$\hat{\mathcal{C}}_k = \left\{ y : \|y - \hat{Y}_{\text{test},k}\| \le Q(S_{1:n}^{\mathrm{PCP}} \cup \{\infty\}; \alpha) \right\}.$$

Here $Q(\cdot;\alpha)$ is the $1-\alpha$ quantile of the empirical distribution of the non-conformity scores $S_1^{\mathrm{PCP}}, \ldots, S_n^{\mathrm{PCP}}$.

Both PCP and `ORCA` use scalar nonconformity scores. PCP takes the minimum distance to the generated samples, yielding a shared radius across candidates. In `ORCA`, the ranked distance vectors E_i are used to learn rank-dependent radii on the exploration split. These radii define the scalar score S_i in (8), which is calibrated on the independent calibration split. If $Q_r(\hat{\beta}_r) = q > 0$ for every rank, then $S_i = q^{-1}\min_r E_{i,r}$; the common factor cancels during calibration, recovering PCP for the same generated candidates and calibration observations. Unequal radii allow coverage to be allocated differently across density ranks.

C Proof

C.1 Proof of Theorem 3.2

Proof. For each rank r, let $\{E_{(1),r} \le \cdots \le E_{(n_1),r}\}$ denote the order statistics of $\{E_{i,r}\}_{i=1}^{n_1}$, and define incremental costs

$$c_{r,1} = E_{(1),r}^{d}, \qquad c_{r,\ell} = E_{(\ell),r}^{d} - E_{(\ell-1),r}^{d} \ \ (\ell \ge 2).$$

We set $E_{(0),r} = 0$ and use $Q_r(1) = 0$ for the zero-radius choice, corresponding to an empty prefix.

(i) From quantile problem to MILP. Let β be feasible for (6). For each r, let $\ell_r \in \{0, \dots, n_1\}$ be the largest index satisfying $Q_r(\beta_r) = E_{(\ell_r),r}$ and set $z_{r,\ell} := \mathbb{1}\{\ell \le \ell_r\}$. For this construction, set

$$y_i := \mathbb{1}\left\{\sum_{r=1}^{K}\sum_{\ell=1}^{n_1} a_{i,r,\ell}\, z_{r,\ell} \ge 1\right\} = \max_{r\in[K]} \mathbb{1}\{E_{i,r} \le Q_r(\beta_r)\}.$$

Then:

(a) Monotonicity (7c). By choice of ℓ_r and construction $z_{r,\ell}$, $z_{r,\ell} \ge z_{r,\ell+1}$ for all ℓ, and $z_{r,\ell} \in \{0,1\}$. Also $y_i \in \{0,1\}$ by definition.

(b) Coverage constraint (7b). We have $y_i \le \sum_{r,\ell} a_{i,r,\ell} z_{r,\ell}$ and

$$\sum_{i=1}^{n_1} y_i = \sum_{i=1}^{n_1} \max_{r\in[K]} \mathbb{1}\{E_{i,r} \le Q_r(\beta_r)\} \;\ge\; \tau,$$

by the feasibility of β (6).

(c) Objective (7a). By telescoping of increments,

$$\sum_{r=1}^{K}\sum_{\ell=1}^{n_1} c_{r,\ell}\, z_{r,\ell} = \sum_{r=1}^{K}\sum_{\ell=1}^{\ell_r}\big(E^d_{(\ell),r} - E^d_{(\ell-1),r}\big) = \sum_{r=1}^{K} E^d_{(\ell_r),r} = \sum_{r=1}^{K}\big(Q_r(\beta_r)\big)^d.$$

Thus (z, y) is feasible for (7) and preserves the objective.

(ii) From MILP to quantile problem. Let (z, y) be feasible for (7). For each r, set $\ell^\star_r := \max\{\ell : z_{r,\ell} = 1\}$ (define $\ell^\star_r = 0$ if all $z_{r,\ell} = 0$), $Q^\star_r := E_{(\ell^\star_r),r}$, and pick $\beta^\star_r$ with $Q_r(\beta^\star_r) = Q^\star_r$. Then

$$\max_r \mathbb{1}\{E_{i,r} \le Q^\star_r\} \ge \mathbb{1}\left\{\sum_{r,\ell} a_{i,r,\ell} z_{r,\ell} \ge 1\right\} \;\ge\; y_i,$$

so $\sum_i \max_r \mathbb{1}\{E_{i,r} \le Q^\star_r\} \ge \sum_i y_i \ge \tau$, i.e., $\beta^\star$ is feasible for (6). Moreover,

$$\sum_{r=1}^{K}\big(Q_r(\beta^\star_r)\big)^d = \sum_{r=1}^{K} E^d_{(\ell^\star_r),r} = \sum_{r=1}^{K}\sum_{\ell=1}^{n_1} c_{r,\ell}\, z_{r,\ell},$$

so the objective values coincide.

(iii) Equivalence. The constructions above show that the feasible radius vectors $\big(Q_r(\beta_r)\big)_{r=1}^{K}$ in (6) coincide with those induced by feasible binary assignments (z, y) in (7), while preserving objective values in both directions. Hence, the MILP formulation (7) is equivalent to optimization (6) and has the same optimal objective value. □

C.2 Proof of Theorem 3.3

Proof. Condition on the fitted generator and exploration data. The Stage 2 radii $Q_r(\hat\beta_r)$ are fixed before calibration. Generate and rank candidates for calibration and test inputs using the same rule as in Stage 1, and define

$$S_i = \min_{r\in[K]} \frac{\|Y_i - \hat Y_{i,\pi_i(r)}\|}{Q_r(\hat\beta_r)}, \qquad S_{\text{test}} = \min_{r\in[K]} \frac{\|Y_{\text{test}} - \hat Y_{\text{test},\hat\pi(r)}\|}{Q_r(\hat\beta_r)},$$

with the zero-denominator conventions in the main text. By the assumed augmented exchangeability, these scores are exchangeable. The threshold in (9), including its infinite-threshold convention, is the $\lceil (n_2+1)(1-\alpha)\rceil$-th order statistic of $\{S_1, \dots, S_{n_2}, \infty\}$. The standard exchangeable-rank argument gives

$$\Pr(S_{\text{test}} \le t_\alpha) \ge 1 - \alpha,$$

with ties handled conservatively. For finite t_α, the zero-denominator conventions imply

$$\begin{aligned} S_{\text{test}} \le t_\alpha &\iff \exists r \in [K] \quad \|Y_{\text{test}} - \hat{Y}_{\text{test},\hat{\pi}(r)}\| \le t_\alpha Q_r(\hat{\beta}_r) \\ &\iff Y_{\text{test}} \in \hat{\mathcal{C}}_{\text{ORCA}}(X_{\text{test}}). \end{aligned}$$

For $t_\alpha = \infty$, the prediction set is $\mathcal{Y}$, so membership holds automatically. Thus $\Pr\{Y_{\text{test}} \in \hat{\mathcal{C}}_{\text{ORCA}}(X_{\text{test}})\} \ge 1 - \alpha$. Integrating over the conditioning variables proves the marginal guarantee. □

C.3 Proof of Proposition 3.4

Proof. Condition on the fitted generator and $X_i = x$. At any interior point y of $\mathcal{Y}$, continuity gives

$$\int_{\mathcal{B}(y;t)} \hat{p}(u \mid x)\, du = v_d t^d \hat{p}(y \mid x)(1 + o(1)), \qquad t \downarrow 0.$$

Conditional on $\hat{Y}_{i,k} = y$, the other $K - 1$ candidates are i.i.d. from $\hat{p}(\cdot \mid x)$. The nearest-neighbor counting law therefore yields

$$\left\|\hat{Y}_{i,k} - \hat{Y}_{i,k}^{[\lfloor um \rfloor]}\right\| = \left(\frac{um}{(K-1)v_d \hat{p}(y \mid x)}\right)^{1/d} \big(1 + o_{\mathbb{P}}(1)\big)$$

uniformly for $u \in [\varepsilon, 1]$, for every fixed $\varepsilon > 0$; see [46, 47]. Averaging these distances gives a Riemann sum. The terms with index below εm contribute at most ε times the $\lceil \varepsilon m \rceil$-th distance, up to a vanishing rounding term. Taking $\varepsilon \downarrow 0$ and using $\int_0^1 u^{1/d}\, du = d/(d+1)$ gives

$$\rho_{i,k} = \frac{d}{d+1}\left(\frac{m}{(K-1)v_d}\right)^{1/d} \hat{p}(\hat{Y}_{i,k} \mid x)^{-1/d}\big(1 + o_{\mathbb{P}}(1)\big).$$

The support boundary has zero probability, so integrating the conditional statement establishes this expansion for each fixed candidate index. Applying it to k and ℓ and taking their ratio proves the first claim. Since the fitted density is bounded above and below, a fixed density gap δ separates the limiting ratio from one, proving the second claim. □

C.4 Additive Volume Surrogate and Overlap

For a fixed input x and generated candidates, define the union volume and the additive volume surrogate by

$$V(x;\beta) := \left|\bigcup_{r=1}^{K} \mathcal{B}_r\big(x; Q_r(\beta_r)\big)\right|, \qquad A(\beta) := v_d \sum_{r=1}^{K} Q_r(\beta_r)^d.$$

Subadditivity of volume gives $V(x;\beta) \le A(\beta)$. For Euclidean balls in $\mathbb{R}^d$, the gap is the volume counted more than once, and it satisfies

$$0 \le A(\beta) - V(x;\beta) \le \sum_{1 \le r < s \le K} \left|\mathcal{B}_r\big(x; Q_r(\beta_r)\big) \cap \mathcal{B}_s\big(x; Q_s(\beta_s)\big)\right|. \tag{13}$$

Indeed, a point contained in $h \ge 1$ balls contributes $h - 1$ to the excess multiplicity and $\binom{h}{2} \ge h - 1$ to the pairwise intersection count. Integrating gives (13). The surrogate equals the union volume when the balls have pairwise intersections of zero volume. If the balls are restricted to a proper subset $\mathcal{Y}$ of $\mathbb{R}^d$, the upper bound $V(x;\beta) \le A(\beta)$ still holds, but the gap may also include volume outside $\mathcal{Y}$.

Effect on coverage allocation. Overlaps depend jointly on the radii, so minimizing $A(\beta)$ need not minimize $V(x;\beta)$. A sufficient condition for approximate agreement is a uniform bound on the relative gap. Specifically, suppose that, for some $\varepsilon \in [0, 1)$,

$$0 \le A(\beta) - V(x;\beta) \le \varepsilon A(\beta) \tag{14}$$

for every allocation β satisfying the exploration coverage constraint in (6). Let $\hat{\beta}$ be a global minimizer of (6), and let $\beta^{\cup}$ minimize $V(x;\beta)$ subject to the same constraint. Then

$$V(x;\hat{\beta}) \le A(\hat{\beta}) \le A(\beta^{\cup}) \le \frac{V(x;\beta^{\cup})}{1-\varepsilon}.$$

Thus, a small uniform relative gap yields a corresponding bound on the union volume attained by the surrogate optimizer. Condition (14) is an additional geometric condition; shrinking radii alone do not ensure it as the number of balls increases. Without such control, the additive objective remains an upper bound on union volume, and its minimizer may differ from the union-volume minimizer. The finite-sample marginal coverage guarantee is unaffected, since it follows from the independent Stage 3 conformal calibration rather than the choice of Stage 2 objective.

C.5 Conditions and Proof of Theorem 3.5

Allocation and calibration conditions. Fix x and write $\mathcal{C} = \mathcal{C}_{K,n_1,n_2}(x)$ for the final calibrated union. Define the fitted-density region

$$\hat{L}(x) = \{y \in \mathcal{Y} : \hat{p}(y \mid x) \ge \hat{\tau}(x)\},$$
$$\hat{\tau}(x) = \sup\left\{t \ge 0 : \int_{\{y\in\mathcal{Y}:\hat{p}(y|x)\ge t\}} \hat{p}(y \mid x)\, dy \ge 1-\alpha\right\}.$$

The additional conditions in Assumption 2 are

$$\left[1-\alpha-\int_{\mathcal{C}} p(y \mid x)\, dy\right]_+ \xrightarrow{\mathbb{P}} 0, \tag{15}$$
$$\left[|\mathcal{C}| - |\hat{L}(x)|\right]_+ \xrightarrow{\mathbb{P}} 0, \tag{16}$$

where $[a]_+ = \max(a,0)$ and the limits include training, exploration, calibration, and candidate randomness. Condition (15) concerns conditional probability at the fixed input x, given the realized prediction set. Condition (16) requires no asymptotic excess union volume relative to the fitted-density region. These are sufficient conditions on the calibrated sets: the first requires asymptotically correct coverage at x, and the second requires no excess union volume relative to the fitted-density region. Appendix C.4 gives a condition under which optimality for the additive Stage 2 objective transfers to union-volume optimality over the same feasible allocations.

Proof of Theorem 3.5. Fitted-density approximation. Let $L = L_{\tau^\star}(x)$, $\tau = \tau^\star(x)$, and $\epsilon = \sup_{y\in\mathcal{Y}} |\hat{p}(y \mid x) - p(y \mid x)|$. For any threshold t, the fitted-density upper-level mass is bounded below and above by

$$\int_{\{p\ge t+\epsilon\}} p(y \mid x)\, dy - \epsilon|\mathcal{Y}| \quad \text{and} \quad \int_{\{p\ge t-\epsilon\}} p(y \mid x)\, dy + \epsilon|\mathcal{Y}|,$$

respectively. The unique true cutoff, the absence of a positive-volume plateau there, and $\epsilon \xrightarrow{\mathbb{P}} 0$ imply $\hat{\tau}(x) \xrightarrow{\mathbb{P}} \tau$. Moreover,

$$\hat{L}(x) \triangle L \subseteq \{y \in \mathcal{Y} : |p(y \mid x) - \tau| \le \epsilon + |\hat{\tau}(x) - \tau|\}.$$

The volume of this band tends to zero in probability, so $|\hat{L}(x)\triangle L| \xrightarrow{\mathbb{P}} 0$. In particular, (16) gives $[|\mathcal{C}| - |L|]_+ \xrightarrow{\mathbb{P}} 0$.

From mass and volume to set convergence. Extend $p(\cdot \mid x)$ by zero outside $\mathcal{Y}$. The excess-mass identity is

$$\int_{\mathcal{C}\triangle L} |p(y \mid x) - \tau|\, dy = \tau\big(|\mathcal{C}| - |L|\big) - \left(\int_{\mathcal{C}} p(y \mid x)\, dy - (1-\alpha)\right).$$

The left side is nonnegative. Conditions (15)–(16) and the fitted-density approximation show that it converges to zero in probability. For every $0 < \eta < \tau$,

$$|\mathcal{C}\triangle L| \le |\{y \in \mathcal{Y} : |p(y \mid x) - \tau| \le \eta\}| + \frac{1}{\eta}\int_{\mathcal{C}\triangle L} |p(y \mid x) - \tau|\, dy.$$

First take the sample-size limits, then let $\eta \downarrow 0$. The no-plateau condition makes the first term vanish, which proves the result. □

C.6 Proof of Corollary 3.6

Proof. Write $p = p(\cdot \mid x)$, $\hat{p} = \hat{p}(\cdot \mid x)$, $\tau = \tau^\star(x)$, $L = L_{\tau^\star}(x)$, $\tilde{L} = \tilde{L}(x)$, and $G(t) = \int_{\{p \geq t\}} p(y)\, dy$, so that $G(\tau) = 1 - \alpha$. We assume $\tilde{\tau}$ exists, for example when $\{\hat{p} = t\}$ has zero volume for every t.

Step 1: $|\tilde{\tau} - \tau| \leq \varepsilon$. If $\tilde{\tau} > \tau + \varepsilon$, then $\tilde{L} \subseteq \{p \geq \tilde{\tau} - \varepsilon\}$ with $\tilde{\tau} - \varepsilon > \tau$, so $1 - \alpha = \int_{\tilde{L}} p \leq G(\tilde{\tau} - \varepsilon) < G(\tau) = 1 - \alpha$, because G is nonincreasing and strictly decreasing near τ. This is a contradiction. If $\tilde{\tau} < \tau - \varepsilon$, then $\tilde{L} \supseteq \{p \geq \tilde{\tau} + \varepsilon\}$ with $\tilde{\tau} + \varepsilon < \tau$, so $1 - \alpha \geq G(\tilde{\tau} + \varepsilon) > G(\tau)$, again a contradiction.

Step 2: localization. Let $A = \tilde{L} \setminus L$ and $B = L \setminus \tilde{L}$. On A, $p < \tau$ and $p \geq \hat{p} - \varepsilon \geq \tilde{\tau} - \varepsilon \geq \tau - 2\varepsilon$. On B, $p \geq \tau$ and $p < \hat{p} + \varepsilon < \tilde{\tau} + \varepsilon \leq \tau + 2\varepsilon$. Hence $\tilde{L} \triangle L = A \cup B \subseteq \{y : |p(y) - \tau| \leq 2\varepsilon\}$, and the margin condition gives $|\tilde{L} \triangle L| \leq C(2\varepsilon)^\gamma$.

Step 3: volume gap. Both sets have p-mass $1-\alpha$, so $\int_A p = \int_B p$. Writing $|A| = \tau^{-1} \int_A \{p + (\tau - p)\}$ and $|B| = \tau^{-1} \int_B \{p - (p - \tau)\}$ gives

$$|\tilde{L}| - |L| = |A| - |B| = \frac{1}{\tau} \int_{A \cup B} |p(y) - \tau|\, dy.$$

The right side is nonnegative. By Step 2, the integrand is at most 2ε on a set of volume at most $C(2\varepsilon)^\gamma$, so $|\tilde{L}| - |L| \leq C(2\varepsilon)^{1+\gamma}/\tau$.

Finally, if p is continuously differentiable near the compact level set $\{p = \tau\}$ and $\nabla p \neq 0$ there, the coarea formula gives $|\{|p - \tau| \leq t\}| \leq Ct$ for small t, that is, $\gamma = 1$. □

D Optimization Approximations

This section provides the algorithmic details for the scalable Stage 2 optimization variants used in our experiments. Recall that the exact MILP in (7) optimizes over prefix variables $z_{r,\ell}$ and coverage indicators y_i. For each rank $r \in [K]$, let

$$E_{(1),r} \leq \cdots \leq E_{(n_1),r}$$

denote the order statistics of $\{E_{i,r}\}_{i=1}^{n_1}$, and set $E_{(0),r} = 0$. We use the same incremental costs and coverage indicators as in the main text:

$$c_{r,\ell} := E^d_{(\ell),r} - E^d_{(\ell-1),r}, \qquad a_{i,r,\ell} := \mathbb{1}\{E_{i,r} \leq E_{(\ell),r}\}.$$

We also write $a_{i,r,0} = 0$ and let

$$\tau = \lceil n_1(1 - \alpha) \rceil$$

be the required number of exploration samples to cover.

For any integer level vector $\ell = (\ell_1, \ldots, \ell_K) \in \{0, \ldots, n_1\}^K$, define the induced rank-dependent radii

$$q_r(\ell) := E_{(\ell_r),r}, \qquad r \in [K],$$

and the corresponding covered set

$$U(\ell) := \left\{ i \in [n_1] : \exists r \in [K] \text{ such that } E_{i,r} \leq E_{(\ell_r),r} \right\}.$$

The Stage 2 feasibility condition is $|U(\ell)| \geq \tau$, and the proxy objective is

$$\mathrm{Cost}(\ell) := \sum_{r=1}^{K} q_r(\ell)^d = \sum_{r=1}^{K} E^d_{(\ell_r),r} = \sum_{r=1}^{K} \sum_{h=1}^{\ell_r} c_{r,h}.$$

The exact MILP searches over all feasible level vectors. The two approximation methods below return feasible integer level vectors $\hat{\ell}$, and therefore rank-dependent radii

$$\hat{q}_r = E_{(\hat{\ell}_r),r}, \qquad r \in [K].$$

These radii are then passed to Stage 3, where the final split-conformal rescaling is applied.

ORCA-LP: LP relaxation and threshold rounding. The first approximation relaxes the binary constraints in (7) to obtain a linear program:

$$\min_{z,y} \quad \sum_{r=1}^{K}\sum_{\ell=1}^{n_1} c_{r,\ell} z_{r,\ell} \tag{17a}$$

$$\text{s.t.} \quad y_i \le \sum_{r=1}^{K}\sum_{\ell=1}^{n_1} a_{i,r,\ell} z_{r,\ell}, \qquad i \in [n_1], \tag{17b}$$

$$\sum_{i=1}^{n_1} y_i \ge \tau, \tag{17c}$$

$$z_{r,\ell} \ge z_{r,\ell+1}, \qquad r \in [K],\ \ell \in [n_1 - 1], \tag{17d}$$

$$0 \le z_{r,\ell} \le 1, \qquad r \in [K],\ \ell \in [n_1], \tag{17e}$$

$$0 \le y_i \le 1, \qquad i \in [n_1]. \tag{17f}$$

The LP relaxation provides a lower bound on the exact MILP objective. To recover a feasible integer solution, we round the fractional prefix variables by thresholding. Since the LP preserves the prefix constraints $z_{r,\ell} \ge z_{r,\ell+1}$, thresholding the fractional solution at any level γ produces a valid prefix vector. We scan over all distinct fractional values and choose the cheapest thresholded solution that covers at least τ exploration samples.

Algorithm 2 ORCA-LP: LP relaxation with prefix rounding

Input: distance matrix $\{E_{i,r}\}$; costs $\{c_{r,\ell}\}$; indicators $\{a_{i,r,\ell}\}$; coverage quota τ
Solve the LP relaxation (17) and obtain fractional solution $(\tilde{z}, \tilde{y})$
Let $\Gamma = \{0, 1\} \cup \{\tilde{z}_{r,\ell} : r \in [K], \ell \in [n_1]\}$, sorted in decreasing order
for $\gamma \in \Gamma$ **do**
  Set $z^{(\gamma)}_{r,\ell} = \mathbb{1}\{\tilde{z}_{r,\ell} \ge \gamma\}$ for all r, ℓ
  Define $\ell^{(\gamma)}_r = \max\{\ell : z^{(\gamma)}_{r,\ell} = 1\}$, with $\ell^{(\gamma)}_r = 0$ if no such ℓ exists
  Compute the covered set

$$U^{(\gamma)} = \left\{ i \in [n_1] : \exists r \in [K] \text{ s.t. } a_{i,r,\ell^{(\gamma)}_r} = 1 \right\}$$

  if $|U^{(\gamma)}| \ge \tau$ **then**
    Set $\hat{\ell}_r = \ell^{(\gamma)}_r$ for all $r \in [K]$
    break
  end if
end for
Output: rank-dependent radii $\hat{q}_r = E_{(\hat{\ell}_r),r}$, $r \in [K]$

Because $\gamma = 0$ selects all prefix variables, the rounding procedure always returns a feasible solution. In practice, this LP-guided rounding often achieves an objective close to the exact MILP while being substantially faster for moderate or large K and n_1.

ORCA-Greedy: jump-based coverage expansion. The second approximation directly exploits the partial-cover structure of (7). Instead of increasing one prefix level at a time, ORCA-Greedy considers *jumps*: from a current level ℓ_r, it may move rank r directly to any higher level $h > \ell_r$. This allows the method to skip uninformative intermediate thresholds and substantially reduces runtime.

At a current level vector ℓ, define the current covered set $U(\ell)$. A jump (r, h) with $h > \ell_r$ changes only the rank-r level from ℓ_r to h. Its marginal coverage gain is

$$\Delta_{\text{cov}}(r, h; \ell) := \left| \{ i \notin U(\ell) : a_{i,r,h} = 1 \} \right|,$$

and its marginal cost is

$$\Delta_{\text{cost}}(r, h; \ell) := E^d_{(h),r} - E^d_{(\ell_r),r} = \sum_{s=\ell_r+1}^{h} c_{r,s}.$$

The greedy rule selects the jump with the largest gain-to-cost ratio:

$$(r^\star, h^\star) \in \arg\max_{r\in[K],\, h>\ell_r} \frac{\Delta_{\mathrm{cov}}(r,h;\ell)}{\Delta_{\mathrm{cost}}(r,h;\ell)}.$$

After reaching the coverage quota, we apply a backward pruning step that lowers redundant rank levels whenever doing so preserves feasibility.

Algorithm 3 `ORCA`**-Greedy: jump-based partial cover**

Input: distance matrix $\{E_{i,r}\}$; costs $\{c_{r,\ell}\}$; indicators $\{a_{i,r,\ell}\}$; coverage quota τ
Initialize $\ell_r = 0$ for all $r \in [K]$
Initialize $U = \emptyset$
while $|U| < \tau$ **do**
  for $r \in [K]$ and $h \in \{\ell_r + 1, \dots, n_1\}$ **do**
    Compute

$$\Delta_{\mathrm{cov}}(r,h) = |\{i \notin U : a_{i,r,h} = 1\}|$$

    Compute

$$\Delta_{\mathrm{cost}}(r,h) = E^d_{(h),r} - E^d_{(\ell_r),r}$$

  end for
  Select

$$(r^\star, h^\star) \in \arg\max_{r,h:\Delta_{\mathrm{cov}}(r,h)>0} \frac{\Delta_{\mathrm{cov}}(r,h)}{\Delta_{\mathrm{cost}}(r,h)}$$

  Update $\ell_{r^\star} \leftarrow h^\star$
  Update

$$U \leftarrow \{i \in [n_1] : \exists r \in [K] \text{ s.t. } a_{i,r,\ell_r} = 1\}$$

end while
Backward pruning:
Set changed $\leftarrow$ true
while changed **do**
  Set changed $\leftarrow$ false
  for $r \in [K]$ **do**
    Find the smallest $h \in \{0, \dots, \ell_r\}$ such that replacing ℓ_r by h still gives $|U(\ell)| \geq \tau$
    if $h < \ell_r$ **then**
      Set $\ell_r \leftarrow h$
      Set changed $\leftarrow$ true
    end if
  end for
end while
Output: rank-dependent radii $\hat{q}_r = E_{(\ell_r),r}$, $r \in [K]$

The jump-based greedy method is especially fast because it avoids solving a global integer program. Its objective is not guaranteed to match the MILP optimum, but it returns a feasible Stage 2 radius direction whenever the loop terminates. Moreover, the backward pruning step can only reduce the proxy objective while maintaining the required exploration coverage. As with the exact MILP and LP relaxation, the final finite-sample validity of `ORCA`-Greedy is ensured by the independent Stage 3 conformal calibration; the approximation affects only the efficiency of the resulting prediction set, not its marginal coverage guarantee.

Approximation guarantee for the greedy variant. The greedy procedure can be viewed as a cost-sensitive partial set-cover algorithm over the finite ground set $[n_1]$. Each jump increases a monotone coverage function

$$F(\ell) := |U(\ell)|$$

and incurs a nonnegative separable cost. Let $\mathrm{OPT}_{\mathrm{MILP}}$ denote the optimal objective value of the exact MILP, and let $\mathrm{Cost}_{\mathrm{Greedy}}$ denote the cost of the greedy solution before final conformal calibration. By the standard greedy analysis for weighted partial cover,

$$\mathrm{Cost}_{\mathrm{Greedy}} \leq (1 + \log \tau)\, \mathrm{OPT}_{\mathrm{MILP}}.$$

The pruning step can only decrease this cost. This guarantee is for the Stage 2 proxy objective; the Stage 3 conformal rescaling is then applied in the same way as for the exact MILP solution.

E Implementation Details

Source code is publicly available at `https://github.com/Minxing-Zheng/ORCA` under the MIT license (package version 1.0.0). The README provides installation instructions and a runnable synthetic comparison of PCP and the MILP, LP, and greedy variants of `ORCA`. The release covers this synthetic pipeline; its scope is detailed in the paper checklist.

All conformal calibration and evaluation experiments were run locally on a Mac machine with an Apple M4 chip, a 10-core CPU, and 16GB unified memory, without requiring GPU acceleration. Neural-network-based conditional density models were trained on an NVIDIA RTX 5070 Ti GPU with 16GB memory using Python 3.7. Runtime measurements for the Stage 2 optimization procedures, including the exact MILP and approximation variants, are reported in Appendix G. In particular, Table 6 summarizes the wall-clock runtime and efficiency tradeoff for different choices of K.

E.1 Model Training and Hyperparameter Settings

For conditional distribution estimation in the 1-dimensional case, we implemented quantile regression forests (QRF) [39] with 100 trees, a minimum of 50 samples per node split, and a quantile grid ranging from 1% to 99%. The trained model serves as the base model for CQR, CHR, DCP, DistSplit, CRD, PCP, and `ORCA`. The implementation is based on the publicly available repository.[1] For methods requiring explicit density estimation, such as CDsplit, we employed a mixture density network (MDN) [40] with 10 Gaussian components, dropout rate of 0.2, hidden layer size of $(100, 100)$, and 500 training epochs. The implementation is based on the Python package `cde` [41]. For R2CCP, we used the default configuration from the author's public repository and trained for 500 epochs. In the multivariate setting, we applied Bonferroni correction to construct coverage intervals for CHR, CQR, DCP, DistSplit, CDsplit, and R2CCP, using the same base models as in the 1-dimensional case. For CRD, PCP, and `ORCA`, we used MDN to model multivariate responses.

E.2 Real Datasets Details

We evaluate the performance of `ORCA` and baseline methods on a diverse collection of real-world datasets spanning a range of application domains. These datasets vary in input dimensionality, response dimension, and sample size, and include both univariate and multivariate prediction tasks. A summary of the datasets used in our experiments is provided in Table 2.

E.3 Evaluation Metrics

Throughout the experiments, we set $\alpha = 0.1$ and evaluate each method using three metrics: marginal coverage, worst-slab conditional coverage, and prediction set efficiency. For a test set $\{(X_i, Y_i)\}_{i=1}^{n_{\text{test}}}$ and prediction sets $\hat{\mathcal{C}}(X_i)$, the empirical marginal coverage is defined as

$$\text{Cover} := \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \mathbb{1}\{Y_i \in \hat{\mathcal{C}}(X_i)\}.$$

This measures whether the prediction sets attain the target marginal coverage level $1 - \alpha$.

To evaluate conditional coverage behavior, we also report worst-slab conditional coverage. This diagnostic scans geometric slices of the covariate space and reports the lowest empirical coverage over sufficiently large slabs. Specifically, for a direction $v \in \mathbb{R}^{d_x}$ and scalars $a < b$, define the slab

$$S_{v,a,b} := \{x \in \mathbb{R}^{d_x} : a \le v^\top x \le b\}.$$

Given a test set $\{(X_i, Y_i)\}_{i=1}^{n_{\text{test}}}$, let

$$\mathcal{I}_{v,a,b} := \{i : X_i \in S_{v,a,b}\}.$$

For a minimum mass threshold $\delta \in (0, 1]$ and a finite set of candidate directions $\mathcal{V}$, we define the empirical worst-slab conditional coverage as

$$\text{Cond_Cover} := \min_{v \in \mathcal{V}} \min_{a<b:\, |\mathcal{I}_{v,a,b}|/n_{\text{test}} \ge \delta} \frac{1}{|\mathcal{I}_{v,a,b}|} \sum_{i \in \mathcal{I}_{v,a,b}} \mathbb{1}\{Y_i \in \hat{\mathcal{C}}(X_i)\}.$$

[1] `https://github.com/msesia/chr`

Table 2: Summary of real-world datasets.

Dataset	# Features	Response Dimension	# Samples
Blog Feedback (Blog) [48]	280	1	52,397
Facebook Comment Volume (Fb1) [49]	53	1	40,948
Facebook Comment Volume (Fb2)	53	1	81,311
MEPS (Year 19)[2]	139	1	15,785
MEPS (Year 20)	139	1	17,541
MEPS (Year 21)	139	1	15,656
Protein Structure (Bio) [50]	9	1	45,730
Temperature [51]	21	1	7,590
Bike[3]	11	2	735,502

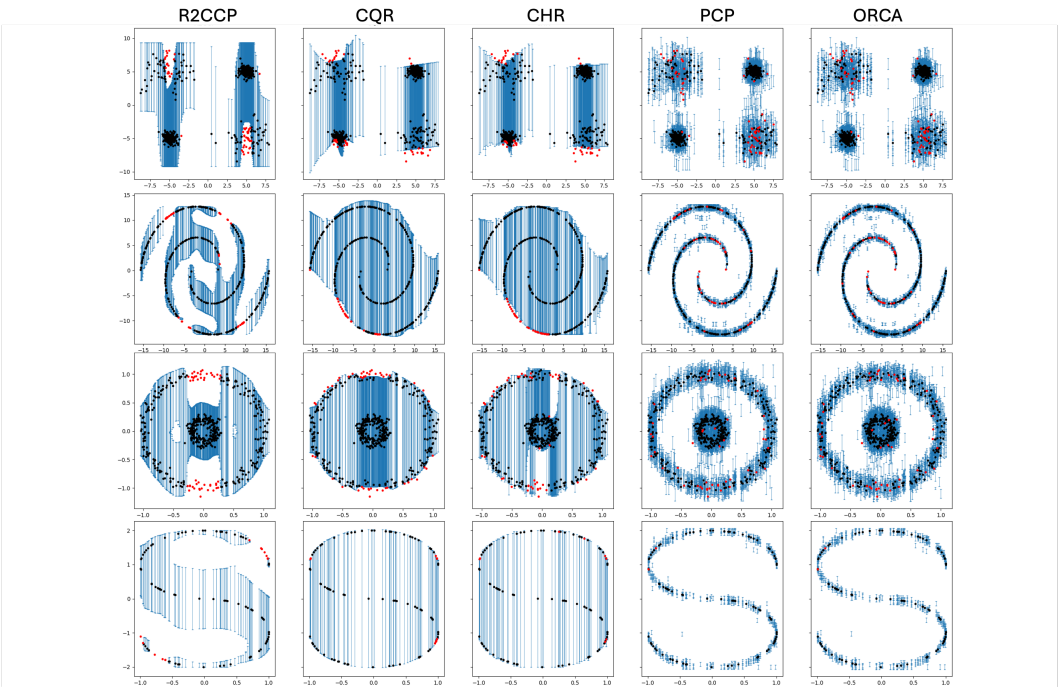


Figure 6: Comparison of prediction set ($\alpha = 0.1$) generated by different methods on 1-dimensional synthetic data: S-shape, Circles, Spirals, and Unbalanced clusters (from top to bottom row). Blues lines: the predictive intervals of each method; Black dots: test points that are covered by the predictive sets; Reds dots: test points not covered. We plot the figure with 100, 500, 400, and 500 random samples for each data, respectively.

Here, δ prevents the diagnostic from selecting slabs containing too few test points. In practice, $\mathcal{V}$ is chosen as a finite collection of random projection directions. This metric captures the worst empirical coverage over localized geometric regions of the covariate space and serves as a diagnostic for conditional coverage heterogeneity.

Finally, we measure prediction set efficiency by the average set size,

$$\text{Eff} := \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \mu\big(\hat{\mathcal{C}}(X_i)\big),$$

where $\mu(\cdot)$ denotes Lebesgue measure for continuous outputs and cardinality for discrete outputs. Smaller values of Eff indicate more informative prediction sets, provided that coverage is maintained.

[2] `https://meps.ahrq.gov/mepsweb/`
[3] `https://citibikenyc.com/system-data`

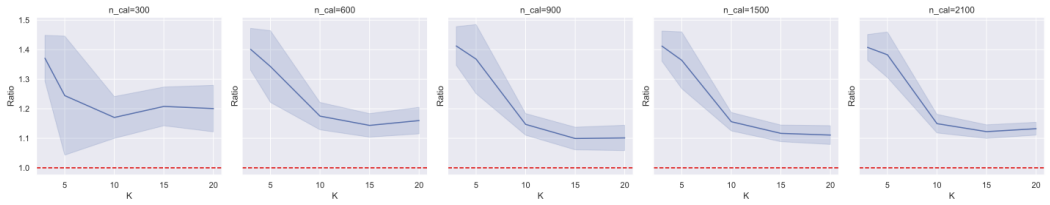


Figure 7: Ratio of PCP prediction set length over ORCA prediction set length under different $K = 3, 5, 10, 15, 20$ and number of calibration data for S-shape data. The blue curve represents the mean, and the shaded area indicates the standard error of the ratio. We observe that the ratio remains consistently above 1, demonstrating that ORCA achieves better efficiency.

F Experiments Results

In this section, we present the additional experimental results for synthetic data, and real data.

F.1 Synthetic Data Results

We report the full synthetic-data results in Table 3, including marginal coverage, worst-slab conditional coverage, and efficiency, with means and standard errors computed over 30 runs.

We further study the effect of K on efficiency in Figure 7, which reports the ratio of PCP prediction-set length to ORCA prediction-set length on the S-shape dataset for $K \in \{3, 5, 10, 15, 20\}$ and varying calibration sample sizes. The curves denote the mean ratio, and the shaded bands indicate standard errors over 100 runs. The ratio remains consistently above 1 across all settings, confirming that ORCA yields more efficient prediction sets than PCP. The ratio gradually decreases as K increases, suggesting that the relative efficiency gain of ORCA becomes smaller for larger K.

Table 3: Summary of synthetic data results.

Dataset	Metric	CQR	CHR	DCP	DistSplit	CDSplit	CRD	R2CCP	PCP	ORCA
Circles	Cover	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01
	Cond_Cover	0.85 ± 0.04	0.84 ± 0.05	0.86 ± 0.03	0.86 ± 0.03	0.87 ± 0.03	0.80 ± 0.03	0.86 ± 0.01	0.85 ± 0.03	0.86 ± 0.03
	Eff	1.70 ± 0.01	1.49 ± 0.06	1.64 ± 0.04	1.64 ± 0.05	0.72 ± 0.04	1.61 ± 0.01	2.77 ± 0.04	0.68 ± 0.31	$\mathbf{0.63 \pm 0.24}$
Moon	Cover	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.92 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.02
	Cond_Cover	0.85 ± 0.05	0.87 ± 0.08	0.87 ± 0.06	0.85 ± 0.06	0.87 ± 0.05	0.78 ± 0.03	0.86 ± 0.02	0.85 ± 0.03	0.85 ± 0.04
	Eff	0.69 ± 0.05	0.69 ± 0.08	0.77 ± 0.06	0.75 ± 0.06	0.42 ± 0.03	0.33 ± 0.03	1.84 ± 0.11	0.35 ± 0.04	$\mathbf{0.29 \pm 0.04}$
S-shape	Cover	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.91 ± 0.02	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.02	0.90 ± 0.01
	Cond_Cover	0.53 ± 0.04	0.80 ± 0.07	0.80 ± 0.06	0.80 ± 0.07	0.88 ± 0.02	0.81 ± 0.04	0.85 ± 0.02	0.84 ± 0.03	0.83 ± 0.04
	Eff	3.36 ± 0.02	3.34 ± 0.02	3.34 ± 0.02	3.34 ± 0.02	1.16 ± 0.11	2.63 ± 0.25	6.94 ± 0.47	0.58 ± 0.01	$\mathbf{0.51 \pm 0.05}$
Spiral	Cover	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.91 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01
	Cond_Cover	0.75 ± 0.06	0.81 ± 0.06	0.82 ± 0.07	0.80 ± 0.06	0.87 ± 0.04	0.81 ± 0.05	0.87 ± 0.02	0.85 ± 0.04	0.85 ± 0.05
	Eff	17.59 ± 0.40	17.02 ± 0.58	16.94 ± 0.55	16.92 ± 0.53	10.67 ± 0.98	12.62 ± 1.27	45.26 ± 2.83	6.27 ± 0.68	$\mathbf{5.96 \pm 0.57}$
Roll	Cover	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.91 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.02
	Cond_Cover	0.65 ± 0.07	0.81 ± 0.05	0.77 ± 0.05	0.76 ± 0.06	0.84 ± 0.04	0.71 ± 0.05	0.86 ± 0.03	0.84 ± 0.04	0.83 ± 0.04
	Eff	18.59 ± 0.17	15.33 ± 0.42	17.39 ± 0.49	17.33 ± 0.47	8.50 ± 0.48	14.64 ± 0.92	36.51 ± 2.56	4.67 ± 0.57	$\mathbf{4.46 \pm 0.58}$
Mixture Gaussian	Cover	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.91 ± 0.01	0.90 ± 0.01	0.90 ± 0.02	0.90 ± 0.01	0.90 ± 0.02
	Cond_Cover	0.89 ± 0.04	0.89 ± 0.04	0.89 ± 0.04	0.89 ± 0.05	0.87 ± 0.03	0.82 ± 0.03	0.85 ± 0.03	0.85 ± 0.03	0.86 ± 0.03
	Eff	12.61 ± 0.11	12.66 ± 0.10	12.62 ± 0.11	12.61 ± 0.10	$\mathbf{6.89 \pm 0.19}$	13.64 ± 0.43	14.49 ± 0.80	7.84 ± 0.53	7.66 ± 0.58

[1] Metrics include marginal coverage (Cover), worst-slab conditional coverage (Cond_Cover), and prediction set efficiency (Eff). Values report mean $\pm$ standard error over 30 runs. The lowest efficiency per dataset is bolded and highlighted along with corresponding coverage metrics.

F.2 Real Data Results

We report the full real datasets results in Table 4, including marginal coverage, worst-slab conditional coverage, and efficiency, with means and standard errors computed over 30 runs.

G Sensitivity Analysis

This section supplements the sensitivity analysis in Section 4.4. We provide the full QRF model-complexity comparison, discuss the nearest-neighbor ranking parameter, and report the runtime–efficiency tradeoff of the Stage 2 solvers and the effect of the output-space metric. The generator-complexity and ranking results are summarized in Table 1.

Table 4: Summary of real data results.

Dataset	Metric	CQR	CHR	DCP-CQR	DistSplit	CDSplit	CRD	R2CCP	PCP	ORCA
Fb1	Cover	0.90 ± 0.00	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.00	0.92 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.02
	Cond_Cover	0.87 ± 0.02	0.86 ± 0.05	0.88 ± 0.00	0.89 ± 0.03	0.87 ± 0.07	0.83 ± 0.03	0.76 ± 0.04	0.82 ± 0.03	0.80 ± 0.05
	Eff	14.78 ± 2.31	11.23 ± 1.34	13.73 ± 1.55	15.88 ± 2.07	35.09 ± 2.69	54.07 ± 40.50	12.97 ± 0.82	7.79 ± 0.28	$\mathbf{3.70 \pm 1.75}$
Fb2	Cover	0.90 ± 0.01	0.89 ± 0.01	0.90 ± 0.02	0.90 ± 0.01	0.93 ± 0.01	0.90 ± 0.02	0.90 ± 0.02	0.90 ± 0.01	0.90 ± 0.01
	Cond_Cover	0.88 ± 0.02	0.84 ± 0.04	0.89 ± 0.05	0.89 ± 0.01	0.87 ± 0.08	0.84 ± 0.03	0.73 ± 0.05	0.82 ± 0.03	0.80 ± 0.04
	Eff	18.31 ± 1.57	10.12 ± 1.04	10.54 ± 1.12	14.48 ± 1.43	44.41 ± 1.61	56.26 ± 2.64	13.39 ± 1.65	9.35 ± 0.21	$\mathbf{5.68 \pm 1.62}$
Blog	Cover	0.90 ± 0.02	0.90 ± 0.02	0.90 ± 0.01	0.90 ± 0.01	0.96 ± 0.01	0.90 ± 0.01	0.89 ± 0.02	0.90 ± 0.01	0.90 ± 0.01
	Cond_Cover	0.87 ± 0.02	0.89 ± 0.03	0.90 ± 0.04	0.88 ± 0.03	0.91 ± 0.02	0.81 ± 0.03	0.74 ± 0.06	0.81 ± 0.04	0.82 ± 0.05
	Eff	16.18 ± 2.61	10.26 ± 1.77	11.37 ± 1.76	16.93 ± 2.13	39.95 ± 3.30	63.07 ± 37.05	10.95 ± 1.82	8.27 ± 0.21	$\mathbf{3.61 \pm 1.27}$
Bio	Cover	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.89 ± 0.01	0.90 ± 0.01	0.90 ± 0.01
	Cond_Cover	0.90 ± 0.03	0.90 ± 0.04	0.89 ± 0.03	0.90 ± 0.04	0.86 ± 0.03	0.84 ± 0.04	0.83 ± 0.03	0.85 ± 0.02	0.82 ± 0.03
	Eff	13.09 ± 0.18	$\mathbf{9.44 \pm 0.33}$	11.58 ± 0.29	11.73 ± 0.30	10.89 ± 0.39	22.10 ± 2.79	9.47 ± 0.48	11.54 ± 0.54	10.88 ± 0.71
Meps 19	Cover	0.90 ± 0.01	0.90 ± 0.01	0.91 ± 0.00	0.90 ± 0.01	0.92 ± 0.01	0.90 ± 0.01	0.89 ± 0.12	0.90 ± 0.01	0.90 ± 0.01
	Cond_Cover	0.89 ± 0.02	0.90 ± 0.03	0.90 ± 0.04	0.86 ± 0.03	0.88 ± 0.02	0.83 ± 0.03	0.81 ± 0.15	0.85 ± 0.03	0.87 ± 0.04
	Eff	30.75 ± 1.42	17.41 ± 1.23	19.35 ± 1.47	29.91 ± 2.11	23.73 ± 1.03	65.66 ± 23.88	22.74 ± 1.24	16.95 ± 0.66	$\mathbf{14.94 \pm 2.96}$
Meps 20	Cover	0.90 ± 0.01	0.90 ± 0.01	0.91 ± 0.01	0.90 ± 0.01	0.92 ± 0.01	0.91 ± 0.02	0.88 ± 0.02	0.90 ± 0.02	0.90 ± 0.01
	Cond_Cover	0.90 ± 0.02	0.90 ± 0.04	0.89 ± 0.00	0.90 ± 0.03	0.85 ± 0.03	0.84 ± 0.04	0.80 ± 0.04	0.85 ± 0.03	0.85 ± 0.03
	Eff	28.27 ± 1.76	17.84 ± 1.41	18.73 ± 1.72	28.71 ± 3.03	23.81 ± 1.22	71.10 ± 26.01	19.46 ± 1.59	16.06 ± 1.87	$\mathbf{14.02 \pm 2.86}$
Meps 21	Cover	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.02	0.90 ± 0.01	0.91 ± 0.01	0.90 ± 0.01	0.88 ± 0.06	0.90 ± 0.01	0.90 ± 0.02
	Cond_Cover	0.90 ± 0.02	0.90 ± 0.03	0.89 ± 0.05	0.89 ± 0.02	0.84 ± 0.02	0.83 ± 0.03	0.80 ± 0.08	0.86 ± 0.03	0.82 ± 0.04
	Eff	30.42 ± 1.33	17.89 ± 1.35	20.23 ± 2.26	29.42 ± 2.13	24.39 ± 3.17	53.65 ± 18.49	19.22 ± 1.60	16.93 ± 1.32	$\mathbf{14.16 \pm 1.97}$
Bike	Cover	0.90 ± 0.01	0.90 ± 0.01	0.91 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.02	0.90 ± 0.01	0.90 ± 0.02	0.90 ± 0.02
	Cond_Cover	0.80 ± 0.02	0.84 ± 0.03	0.81 ± 0.03	0.81 ± 0.03	0.80 ± 0.05	0.82 ± 0.04	0.83 ± 0.05	0.80 ± 0.05	0.81 ± 0.05
	Eff ($\times 10^{-3}$)	0.54 ± 0.02	0.49 ± 0.02	0.44 ± 0.02	0.43 ± 0.02	0.49 ± 0.03	1.07 ± 0.13	0.79 ± 0.09	0.46 ± 0.03	$\mathbf{0.36 \pm 0.00}$
Temperature	Cover	0.90 ± 0.02	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.02	0.90 ± 0.01	0.90 ± 0.02	0.90 ± 0.02	0.90 ± 0.01	0.90 ± 0.01
	Cond_Cover	0.89 ± 0.04	0.89 ± 0.05	0.88 ± 0.03	0.88 ± 0.05	0.85 ± 0.03	0.80 ± 0.04	0.79 ± 0.04	0.85 ± 0.03	0.82 ± 0.03
	Eff	3.45 ± 0.10	3.42 ± 0.12	3.43 ± 0.10	3.46 ± 0.11	2.59 ± 0.07	3.78 ± 0.11	2.55 ± 0.09	3.80 ± 0.01	$\mathbf{2.43 \pm 0.13}$

[1] Metrics include marginal coverage (Cover), worst-slab conditional coverage (Cond_Cover), and prediction set efficiency (Eff). Values report mean $\pm$ standard error over 30 runs. The lowest efficiency per dataset is bolded and highlighted along with corresponding coverage metrics.

Sensitivity to generator complexity. We vary the minimum leaf size of the QRF generator on S-shape. Table 1(a) summarizes marginal coverage and efficiency relative to PCP, and Table 5 additionally reports worst-slab coverage and the comparison with CHR. The gains relative to PCP occur at intermediate leaf sizes, showing that the efficiency gain depends on generator complexity.

Table 5: **Sensitivity to generator complexity (QRF).** We vary the minimum leaf size and report marginal coverage (Cov.), worst-slab conditional coverage (CondCov.), prediction-set size (Eff.), and size ratios relative to PCP and CHR. Smaller sizes and ratios indicate greater efficiency.

#leaf	Cov.	CondCov.	Eff (ORCA)	$\frac{\text{Eff(ORCA)}}{\text{Eff(PCP)}}$	$\frac{\text{Eff(ORCA)}}{\text{Eff(CHR)}}$
1	0.90	0.86	3.99	1.00	1.14
20	0.89	0.84	0.11	0.84	0.03
50	0.90	0.85	0.21	0.87	0.06
100	0.90	0.87	0.52	0.85	0.16
200	0.90	0.85	0.77	1.03	0.22

Sensitivity to m-NN ranking parameter. ORCA ranks candidates using m-nearest-neighbor statistic over the conditional samples. Small m yields high-variance rankings, while larger m smooths local geometry and stabilizes the ordering. We vary m and report the fractional efficiency gain over PCP $\frac{\widehat{\text{Eff}}(\text{PCP}) - \widehat{\text{Eff}}(\texttt{ORCA})}{\widehat{\text{Eff}}(\text{PCP})}$, where positive values indicate smaller sets than PCP. Table 1(b) shows consistent gains for $m \geq 7$, whereas very small m can lead to unstable ranks and weaker improvements. Overall, our method performs well for moderate choices of m.

Runtime–efficiency tradeoff. We compare the runtime and efficiency of PCP, ORCA-MILP, ORCA-LP, and ORCA-Greedy on the S-shape dataset. Reported runtimes cover calibration (the Stage 2 optimization and Stage 3) and test-set evaluation; sample generation and ranking are performed beforehand and excluded. Stage 2 is solved once at calibration time. At test time, ORCA generates and ranks K samples and checks membership in their balls, with no further optimization. Table 6 reports the mean runtime in seconds and prediction-set efficiency over 10 independent runs, where smaller efficiency values indicate tighter prediction sets. Overall, ORCA-Greedy provides the most favorable practical tradeoff: it is orders of magnitude faster than the exact MILP while achieving nearly identical, and in some cases slightly better, efficiency. For example, when $K = 20$, ORCA-Greedy matches the efficiency of ORCA-MILP while reducing runtime from 23.15 seconds to 0.06 seconds. ORCA-LP also substantially reduces runtime relative to the MILP and maintains competitive efficiency, though it is consistently slower than the greedy variant. ORCA-Naive is substantially slower and is

included only as a reference implementation. These results suggest that the greedy approximation is a scalable and effective default choice for practical deployments of ORCA.

Table 6: **Runtime and efficiency vs. K.**

	time (s) / Eff.			
Method	K=10	15	20	25
PCP	0.00/0.75	0.00/0.48	0.00/0.37	0.00/0.34
ORCA-MILP	11.95/0.67	16.97/0.35	23.15/0.32	40.96/0.31
ORCA-LP	1.50/0.67	2.88/0.37	5.22/0.34	8.94/0.36
ORCA-Greedy	0.02/0.66	0.03/0.35	0.06/0.32	0.09/0.32
ORCA-Naive	174.12/0.69	442.16/0.37	548.92/0.35	665.89/0.33

Geometry and metric choice. The conformal coverage argument applies to any nonnegative dissimilarity fixed before calibration and used consistently at calibration and test time. The chosen dissimilarity determines the geometry of the prediction region, while its volume interpretation depends on that geometry.

Table 7: **Metric ablation on MNIST-VAE.** We compare PCP and ORCA under different output-space metrics. Coverage is reported together with the off-class rate (OCR), where smaller OCR indicates fewer semantically incorrect images included in the prediction set.

Metric	PCP Cover	PCP OCR	ORCA Cover	ORCA OCR
Pixel-space ℓ_2	0.899	0.374	0.900	**0.338**
Latent-space ℓ_2	0.909	0.178	0.908	**0.123**
Latent cosine distance	0.906	0.090	0.897	**0.054**
Learned latent metric	0.901	0.062	0.905	**0.019**

To examine this flexibility, we extend the MNIST-VAE experiment using four metrics: pixel-space ℓ_2 distance, latent-space ℓ_2 distance with latent dimension 16, latent-space cosine distance, and a learned latent metric. Since raw prediction-set size is not directly comparable across different metrics, we instead report the *off-class rate* (OCR). For a target input digit class c, OCR measures the probability that images from classes other than c are included in the uncertainty set. A smaller OCR therefore indicates that the prediction set contains fewer semantically incorrect images. As shown in Table 7, both PCP and ORCA achieve coverage close to the nominal level, while ORCA consistently attains lower OCR across all metrics. The improvement is especially pronounced for learned latent metrics, suggesting that incorporating task-relevant geometry can substantially improve the semantic quality of conformal prediction sets in high-dimensional output spaces.

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 abstract and introduction state the main contributions: rank-adaptive generative conformal prediction, finite-sample coverage, and empirical efficiency gains.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The paper discusses limitations related to generative model quality, metric choice, computational cost, and sensitivity to design parameters.

   Guidelines:

   - The answer [N/A] 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: Section 3.4 states Assumptions 1–2 and the scope of the HDR result. Proofs and the additional allocation/calibration conditions are given in Appendix C.

   Guidelines:

   - The answer [N/A] 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: The paper reports datasets, baselines, data splits, evaluation metrics, hyperparameters, and implementation details needed to reproduce the main results.

   Guidelines:

   - The answer [N/A] 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. In the case of 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: Source code is publicly available at `https://github.com/Minxing-Zheng/ORCA` under the MIT license, with installation and run instructions that reproduce the synthetic experiments. The data sources for the remaining experiments are public and cited, and their preprocessing and settings are described in Appendix E.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies model choices, hyperparameters, calibration procedures, evaluation metrics, and dataset details in the main text and appendix.

   Guidelines:

   - The answer [N/A] 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: The paper reports means and standard errors over repeated runs for the main experimental tables and figures.

   Guidelines:

   - The answer [N/A] 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: Appendix E reports the hardware used for model training and for conformal calibration and evaluation, and Appendix G reports the runtimes of the Stage 2 solvers.

   Guidelines:

   - The answer [N/A] 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 work uses public datasets and standard experimental protocols and does not involve deceptive, harmful, or privacy-sensitive data collection.

   Guidelines:

- The answer [N/A] 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: [N/A]

    Justification: This is foundational methodology for uncertainty quantification and is not tied to a specific deployment. Section 5 discusses a practical caveat for users: the guarantee is marginal, so coverage can vary across input regions.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, or systems requiring special safeguards.

    Guidelines:

    - The answer [N/A] 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 are public benchmarks and all baseline implementations are cited to their original sources; the `ORCA` code is released under the MIT license with a third-party attribution notice.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: The released software is documented through its README, package-version metadata, MIT license, and third-party attribution notice at `https://github.com/Minxing-Zheng/ORCA`. Installation and synthetic-experiment commands are provided; no new dataset is introduced.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve human-subject experiments and therefore does not require IRB review.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: No LLM is a component of the proposed prediction or calibration procedure.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.