COPROUTE: DEPENDENCE-AWARE CONFORMAL ROUTING FOR LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Set-based language model routing aims to increase the coverage of the event where at least one model answers a query correctly while controlling the number of inference calls made. Existing risk-controlled routing policies bound the all-miss risk (the probability that every model in the selected set fails) by calibrating a single conformal threshold and applying it to a nested family of model sets. Because each model is scored in isolation, these policies ignore the dependence structure of model failures. Models that share training data or architecture tend to fail on the same queries, so two individually strong models can be jointly redundant while a weaker but complementary model covers the queries that the others miss. To account for this dependence, which existing risk-controlled policies ignore, we propose CopRoute, which orders models by how many new queries each one answers correctly beyond those already selected, decides per query how far down that order to go, and then picks the smallest cutoff that still meets the target risk. We show that CopRoute's ordering attains, on the design data, the best guarantee any efficient fixed-ordering algorithm can give for the average number of calls to the first correct model, and that a separate calibration step certifies the all-miss risk of the deployed sets. On RouterBench and RouterEval, CopRoute attains the fewest certified calls on nine of ten datasets at a target risk of 0.05 and on eight of ten at 0.10, making up to 79 percent fewer calls (median 24 percent) than the state-of-the-art risk-controlled policy at the same certified all-miss risk.

1 INTRODUCTION

A deployed language-model (LM) system typically consists of a pool of models that differ in size, task expertise, architecture, and other attributes (Hu et al., 2024; Huang et al., 2025; Varangot-Reille et al., 2026). A given model in this pool is usually right on some queries and wrong on others, and no single model dominates across all tasks (Srivatsa et al., 2024; Chen et al., 2024b; Zhuang et al., 2025). An LM system aims to obtain the right answer for a given input with the minimum number of inference calls (Chen et al., 2024a; Ong et al., 2025; Ding et al., 2024; 2025). This goal has been posed as a statistical optimization problem based on the risk-controlled routing framework (Hao et al., 2026; Su et al., 2026; Angelopoulos et al., 2024). The risk-controlled routing objective is to minimize the number of inference calls across the pool of models, and the risk is the probability of not obtaining at least one correct answer from the called models.

Existing methods for this problem fall into two categories. The first category does not use the dependence or co-failure information among the candidate models. Single-model routers choose one model from estimates of its quality and cost (Chen et al., 2024a; Ong et al., 2025; Ding et al., 2024; Chen et al., 2024b; Feng et al., 2025; Li, 2025; Wang et al., 2025b), while cascades move through models until a stopping rule is met (Dohan et al., 2022; Gupta et al., 2024; Ding et al., 2025). RACER instead returns a set with a statistical risk guarantee (Hao et al., 2026). It applies one calibrated threshold to per-model scores, and relaxing this threshold produces larger nested model sets. RACER controls the probability that every selected model is wrong, but it scores each candidate in isolation, without accounting for the candidate models already in the selected set. Scoring in isolation ignores that LM failures are correlated; models trained on similar data or built from similar architectures may fail on the same queries (Goel et al., 2025; Chen, 2026). We define two models as complementary when one succeeds on queries that the other fails to answer. A routing policy that ignores this correlation spends calls on models that cover no queries beyond those al-

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

ready selected, while a weaker but complementary model could have correctly answered the queries the set still misses. The second category uses the dependence between model failures or model diversity when forming sets. Classical ensemble selection and diversity measures favor models with different error patterns (Caruana et al., 2004; Giacinto & Roli, 2001; Kuncheva & Whitaker, 2003). Recent methods use the expected quality of the best answer or determinantal point processes to form complementary LM sets (Dekoninck et al., 2025; Wang et al., 2026). These methods improve their coverage, defined as the probability that at least one selected model is correct, but they fall short of risk-controlled routing in three respects. First, they carry no distribution-free guarantee on the all-miss risk. They operate at a fixed call budget or with a fixed preference weight, and can neither accept a target risk level nor signal that one is unattainable. Second, the expected-best-answer objective of Dekoninck et al. (2025) is evaluated from per-model estimates alone, and diversity measures and determinantal point processes score a set through pairwise model similarities, so co-failures among three or more models enter only through their pairwise projections. Third, the learned variants train query encoders and per-model embeddings in place of the router a system already runs (Wang et al., 2026), so changing the model pool requires retraining. Existing work therefore leaves a gap between policies that certify the all-miss risk but ignore dependence and methods that use dependence but carry no certificate. Appendix A has more details about both lines of work.

We improve the risk-controlled routing framework by incorporating a dependence-aware routing rule. We propose `CopRoute`, short for ***COoPerative ROUTEr***, a family of drop-in dependence-aware conformal routing policies that reuse the router a system already runs. Unlike earlier risk-controlled policies that score one candidate model at a time, CopRoute builds the selected set sequentially and tracks the estimated probability that every model in that set fails. CopRoute first orders the candidate models by how many design queries each one answers that the models before it missed; for each new query it then decides, from the router's predicted success probabilities, how many models along this order to call; and finally it calibrates the smallest cutoff on this order that meets the target risk. On ten datasets from RouterBench (Hu et al., 2024) and RouterEval (Huang et al., 2025) with pools of 11 to 500 models, CopRoute attains the fewest certified calls on nine of ten datasets at a target risk of 0.05 and eight of ten at 0.10, calling a median 24 percent and up to 79 percent fewer models than RACER-G, and fewer than FlexRouter and BaRP in 45 of 48 pool comparisons.

We summarize our contributions as follows. (i) We show that the all-miss probability of a model set depends jointly on the individual model accuracies and on the dependence among model failures. Under conditional independence, scoring models in isolation is optimal at every set size, while under dependence the best set need not contain the individually best models. (ii) We introduce CopRoute and establish the three guarantees above for its model ordering, its calibrated cutoff, and its finite-sample correction. The certificate does not depend on the estimator, so any dependence estimator in the family, selected on validation before final calibration, can produce the order without weakening the guarantee. (iii) We evaluate CopRoute against RACER, FlexRouter, BaRP, and independence controls on ten datasets from RouterBench and RouterEval. In addition to the call reductions above, CopRoute lowers end-to-end inference cost by 46.6 percent relative to RACER-G at equal certified risk without lowering final-answer accuracy, and its certified call count changes least among the compared policies when the router is swapped or degraded.

2 Preliminaries on the α-VOR problem and notation

Let $\mathcal{M} = \{1, \dots, m\}$ be a pool of m language models. We write X for a random query drawn from the deployment distribution and x for one realized query. For each model $j \in \mathcal{M}$, let $Y_j = \mathbf{1}\{\text{model } j \text{ answers } X \text{ correctly}\}$. Thus $Y_j = 1$ means that model j is correct, and $Y_j = 0$ means that it is wrong. The set $G(X) = \{j \in \mathcal{M} : Y_j = 1\}$ contains all candidate models that answer X correctly. A routing policy, denoted by C, takes a query as input and returns a set of models to call. Hence $C(x) \subseteq \mathcal{M}$, and $|C(x)|$ is the number of inference calls made for query x.

Hao et al. (2026) pose the α-Valid Optimal Routing (α-VOR) problem as follows. For a user-specified risk target $\alpha \in (0, 1)$, the population problem is

$$\min_{C} \ \mathbb{E}\big[|C(X)|\big] \quad \text{subject to} \quad \mathbb{P}\big(C(X) \cap G(X) = \varnothing\big) \le \alpha. \tag{1}$$

In (1), the subscript C under the minimum states that the optimization ranges over routing policies. This optimization does not choose a model set after the correct models are revealed. The expectation

is the average number of called models for a new query X. The constraint bounds the probability that the called set contains no correct model. Both quantities are taken over the deployment distribution of X and, if the routing policy is randomized, over that randomization. We call the event $C(X) \cap G(X) = \varnothing$ an *all-miss event*. The constraint is attainable only when the full pool itself meets the target, $\mathbb{P}(G(X) = \varnothing) \leq \alpha$; when the data cannot certify a chosen target, the calibration rule below returns `NON-CERTIFIED`.

All-miss probability and coverage for one query. Let $S \subseteq \mathcal{M}$ denote any proposed set of models. After fixing a query value x, define $q_x(S) = \mathbb{P}\big(Y_j = 0 \text{ for every } j \in S \mid X = x\big), \quad c_x(S) = 1 - q_x(S)$. Here $q_x(S)$ is the conditional probability that every model in S is wrong, while $c_x(S)$ is the conditional probability that at least one model in S is correct. We call $c_x(S)$ the *coverage* of S on query x.

Ordering and prefixes. An ordering rule π takes a query x as input and returns a permutation $\pi_x = (\pi_{x,1}, \ldots, \pi_{x,m})$ of the model indices in $\mathcal{M}$. The model $\pi_{x,1}$ is ranked first, $\pi_{x,2}$ second, and so on. A fixed ordering ignores x and returns the same permutation for every query; a query-specific ordering may return a different permutation for each x. For $k \in \{0, 1, \ldots, m\}$, the depth-$k$ prefix is $P_k^\pi(x) = \{\pi_{x,1}, \ldots, \pi_{x,k}\}, \quad P_0^\pi(x) = \varnothing$. Thus k is both the prefix depth and the number of models in that prefix; the prefix is a set, not an ordered list, and the permutation π_x determines which candidate models enter it.

Sizing and entry scores. The ordering rule determines which candidate models are considered first, and a sizing rule determines how many of them to call for a particular query. Formally, it takes a query x and a depth k, then assigns a residual $r_k(x) \in [0, 1]$ to the prefix $P_k^\pi(x)$, with $r_0(x) = 1$. The residual estimates the probability that all k candidate models in the prefix are wrong. We require $r_k(x)$ to be nonincreasing in k, because adding another candidate model cannot increase the estimated all-miss probability. The entry score of the model at position k is

$$s_\pi(x, \pi_{x,k}) = 1 - r_{k-1}(x). \tag{2}$$

The entry score in (2) is the estimated coverage before the k-th candidate model is added. Since the residual is nonincreasing in depth, the entry scores are nondecreasing in depth, and because $r_0(x) = 1$ the first ordered model has entry score zero, so every certified cutoff calls at least one model. Section 3 defines the specific ordering and sizing rules used by CopRoute.

Cutoffs and nested model sets. For a cutoff λ, let $C_\lambda(x) = \{j \in \mathcal{M} : s(x, j) \leq \lambda\}$. The set $C_\lambda(x)$ uses the *sublevel convention*; it includes every candidate model whose score is at most λ, and the nonstrict inequality means that models with the same score enter together. These sets form a nested family. Specifically, if $\lambda \leq \lambda'$, then $C_\lambda(x) \subseteq C_{\lambda'}(x)$ for every query x. The family $\{C_\lambda(x)\}_\lambda$ is therefore a nested family of sets. Increasing the cutoff can add candidate models but cannot remove a candidate model that was already selected. For the entry scores above, each $C_\lambda(x)$ is a prefix of π_x, apart from an entire tied block entering at once.

Calibration loss and empirical risk. Suppose the final-calibration sample contains n labeled queries (X_i, G_i), where $i \in \{1, \ldots, n\}$ and G_i is the set of models correct on X_i. For a candidate cutoff λ, define $L_i(\lambda) = \mathbf{1}\{C_\lambda(X_i) \cap G_i = \varnothing\}$. Thus $L_i(\lambda) = 1$ when the selected set misses every correct model on calibration query i, and it is zero otherwise. The empirical all-miss risk is $\widehat{R}_n(\lambda) = \frac{1}{n} \sum_{i=1}^n L_i(\lambda)$. The hat indicates that $\widehat{R}_n$ is computed from the n calibration queries rather than from the unknown deployment distribution. The calibrated cutoff is

$$\widehat{\lambda} = \min \left\{ \lambda \in \mathbb{R} : \frac{n}{n+1} \widehat{R}_n(\lambda) + \frac{1}{n+1} \leq \alpha \right\}. \tag{3}$$

The expression inside braces is the set of cutoffs that pass the finite-sample calibration rule. The outer minimum chooses the smallest passing cutoff. Because the family is nested, this cutoff selects no more models than any other passing cutoff in the same family. If no cutoff passes, the procedure returns `NON-CERTIFIED` and deploys no routing policy under this certificate. Theorem 2 states the distribution-free guarantee for this construction, and Corollary 1 carries it to the calibrated score cutoff. Hao et al. (2026) additionally use a virtual null model as an abstention mechanism (Appendix G), and using the per-model router score $1 - \widehat{p}_j(x)$ in the construction above reduces it to the inverse-probability (RACER-P) family of Hao et al. (2026) over the real model pool.

3 DEPENDENCE-AWARE ROUTING

CopRoute makes an *ordering* decision, which fixes the order in which models are considered, and a *stopping* decision, which fixes for a given query how far down that order to call. While many orders and stopping rules are possible in theory, we present one default policy and one ablation of its stopping rule, and the last paragraph of this section explains this choice.

Greedy tally ordering. Let $D \in \{0,1\}^{n_d \times m}$ be the design correctness matrix, where $D_{ij} = 1$ when model j answers design query i correctly, and let $\bar{p}_j^D = n_d^{-1} \sum_i D_{ij}$. For a model set S, let $\widehat{q}_D(S) = n_d^{-1} \sum_i \prod_{j \in S} (1 - D_{ij})$ denote its design all-miss frequency. Starting with no selected models, we append one candidate model at a time, each time choosing the model that minimizes this frequency. We denote the resulting *fixed-order tally* by $\pi^T = (\pi_1^T, \ldots, \pi_m^T)$, where T stands for tally, and write $P_k^{\pi^T} = \{\pi_1^T, \ldots, \pi_k^T\}$. Thus

$$\pi_k^T \in \arg\min_{j \notin P_{k-1}^{\pi^T}} \widehat{q}_D\Big(P_{k-1}^{\pi^T} \cup \{j\}\Big). \tag{4}$$

Since both stopping rules use the fixed-tally order, we drop the superscript T below and write π_k and P_k for π_k^T and $P_k^{\pi^T}$, respectively, whenever it is obvious.

Marginal probabilities and joint failure. Let $p_j(x) = \mathbb{P}(Y_j = 1 \mid X = x)$, and let $\widehat{p}_j(x)$ be its router estimate. The true all-miss probability of prefix P_k depends on both the individual miss probabilities and their dependence. Using the correctness subcopula, $q_x(P_k) = C^0_{P_k,x}(1 - p_j(x) : j \in P_k)$. If the model probabilities given a query are conditionally independent, this identity reduces to a product of individual miss probabilities, motivating the first stopping rule below. The second rule adjusts this product using observed co-failure on the design set.

Two query-specific stopping rules Both rules below, $r_k^{\mathrm{ind}}(x)$ and $r_k^{\mathrm{dep}}(x)$, use the fixed order and the router estimates for the current query. The independence-based residual $r_k^{\mathrm{ind}}(x)$ multiplies the estimated miss probabilities. The dependence-based residual $r_k^{\mathrm{dep}}(x)$ additionally uses the design-set dependence ratio $d_k = \widehat{q}_D(P_k) \big/ \prod_{\ell=1}^k (1 - \bar{p}_{\pi_\ell}^D)$, which compares the observed prefix all-miss frequency with its independence prediction. This ratio is noisy when few design queries miss the whole prefix, so the dependence residual uses a shrunk version $\widetilde{d}_k$ controlled by a support threshold h. For $h > 0$ the shrunk ratio keeps the measured value when at least h design queries miss the prefix and otherwise moves it toward the independence value 1, whereas $h = 0$ is defined to apply no correction, so that $\widetilde{d}_k = 1$ at every depth. This gives

$$r_k^{\mathrm{ind}}(x) = \prod_{\ell=1}^{k} \left(1 - \widehat{p}_{\pi_\ell}(x)\right), \qquad r_k^{\mathrm{dep}}(x) = \min_{0 \le t \le k} \mathrm{clip}_{[0,1]} \left(r_t^{\mathrm{ind}}(x)\, \widetilde{d}_t\right). \tag{5}$$

A dependence ratio below one indicates complementary failures; a value above one indicates more co-failure than independence predicts. The minimum over preceding depths prevents the estimated all-miss probability from increasing when another model is added. We choose h from the grid $\{0, 5, 20, 50\}$ on a separate validation split before final calibration, and because the grid includes $h = 0$, the selected value can turn the correction off altogether. For either rule, $s(x, \pi_k) = 1 - r_{k-1}(x)$, and final calibration selects one global cutoff using (3). Appendix E gives the complete construction and validation rule.

The CopRoute family. The default policy, **CopRoute-Ind**, uses the independence-based rule ($h = 0$) and has no hyperparameter. **CopRoute-Dep** uses the dependence-based rule with h chosen on validation; because the grid contains $h = 0$, it can reproduce CopRoute-Ind exactly, and on most redraws it does. We report CopRoute-Dep as an ablation of the stopping rule and not as a second main policy. The dependence ratio is estimated from the design queries that miss an entire prefix, which are few at deep prefixes and in large pools, and the resulting estimation noise can make the corrected rule call more models than the uncorrected one (Section 5).

Beyond the rules described above, CopRoute supports further fixed and query-specific dependence models. Its fixed orders can be learned from empirical co-failure counts, failure correlations, rank correlations, latent-variable models, class mixtures, shrinkage estimators, or model-family information. Its query-specific variants combine the router estimates for the current query with either

Algorithm 1 CopRoute

Require: pool $\mathcal{M}$; disjoint splits $\mathcal{D}_{\mathrm{des}}, \mathcal{D}_{\mathrm{val}}, \mathcal{D}_{\mathrm{cal}} = \{(X_i, G_i)\}_{i=1}^{n}$; router $\widehat{p}_j(\cdot)$; target risk α

1: fit the tally order π^T on $\mathcal{D}_{\mathrm{des}}$ by (4)
2: on $\mathcal{D}_{\mathrm{val}}$, select the stopping rule of (5), r^{ind} for the default CopRoute-Ind or r^{dep} with support threshold h for the CopRoute-Dep ablation, and freeze all choices
3: score the model at position k of π^T by $s(x, \pi_k^T) = 1 - r_{k-1}(x)$ for every query x
4: on $\mathcal{D}_{\mathrm{cal}}$, set $\widehat{\lambda} = \min\{\lambda : \frac{n}{n+1}\widehat{R}_n(\lambda) + \frac{1}{n+1} \le \alpha\}$ as in (3); **if** no λ passes, **return** `NON-CERTIFIED`

Ensure: routing policy $C(x) = \{j \in \mathcal{M} : s(x, j) \le \widehat{\lambda}\}$

design-level dependence statistics or correctness labels from nearby router-training queries. These orderings are paired with independence-based, dependence-based, or local empirical stopping rules. We use the tally order with the two stopping rules above for the policies reported in the main text, because this order requires only design-set counts and has the fixed-order guarantee developed in Section 4.1. Appendix E describes the remaining variants.

4 THEORY

In the previous sections, we discussed the importance of considering the co-occurrence of model failures. In this section, we establish two guarantees. First, the fixed tally order comes within a factor of four of the best possible fixed order on the design set, and no polynomial-time method can guarantee a smaller factor. Second, once the order and the stopping score are frozen, split calibration keeps the all-miss risk below the target level, and the smallest certified cutoff calls the fewest models among all certified cutoffs. Formal proofs are presented in Appendices C and D.

4.1 WHY FIXED TALLY IS A NEAR-OPTIMAL ORDERING

For an ordering π, let $T_i(\pi)$ be the position of the first model that answers design query i correctly, with $T_i(\pi) = m$ when the full pool misses that query. Summing the design all-miss estimate over all prefixes of an order gives the average position at which the first correct model appears, $J_D(\pi) = \sum_{k=0}^{m-1} \widehat{q}_D\big(P_k^{\pi}\big) = \frac{1}{n_d}\sum_{i=1}^{n_d} T_i(\pi)$, and a smaller value means that a correct model appears earlier on average.

Theorem 1 (Fixed tally matches the best achievable approximation factor)**.** *Let $D \in \{0,1\}^{n_d \times m}$ be the realized design correctness matrix. Run the fixed-tally rule of* (4) *through all m positions, using the tie rule in Section 3, and denote the resulting order by π^T. At each step, the fixed-tally rule selects a candidate model that answers the largest number of design queries missed by the current prefix. It is therefore the greedy minimum-sum set-cover ordering, and*

$$J_D(\pi^T) \le 4 \min_{\pi} J_D(\pi). \tag{6}$$

Moreover, for every $\varepsilon > 0$, no polynomial-time algorithm that receives an arbitrary correctness matrix and returns a fixed model ordering can guarantee a factor $4 - \varepsilon$ for this objective unless $\mathrm{P} = \mathrm{NP}$ *(Feige et al., 2004; Bansal et al., 2023).*

Theorem 1 concerns one realized design matrix and the complete fixed ordering. The factor-four guarantee does not transfer to population or test-time calls (Appendix D.2 quantifies this), and a construction stopped after $k_0 < m$ positions receives the guarantee only for that truncated instance. The tally order also inherits the greedy coverage guarantee that at every depth k, its design coverage is at least $1 - (1 - 1/k)^k \ge 1 - 1/e$ times the best coverage attainable by any set of k models, and Appendix C.2 proves both results.

4.2 WHY THE FINAL ROUTING SETS ARE VALID AND EFFICIENT

For real-model sets $S, G \subseteq \mathcal{M}$, define the standard all-miss loss $L^{\mathrm{miss}}(S, G) = \mathbf{1}\{S \cap G = \varnothing\}$. An empty called set counts as a miss, and the loss has no abstain option, so declining to answer, or passing the query to a human reviewer, is not available as a zero-loss outcome.

Theorem 2 (Frozen-family split certification)**.** *Before inspecting any final-calibration label, let design and validation data select a finite nested family $C_0(x) \subseteq C_1(x) \subseteq \cdots \subseteq C_K(x) \subseteq \mathcal{M}$. Condition on those data and every auxiliary random choice used to construct the family. Assume that the same measurable map is applied to every calibration query and one future query. Also assume that these $n+1$ observations are exchangeable. Define $L_i(k) = L^{\mathrm{miss}}\big(C_k(X_i), G_i\big)$, $\widehat{R}_n(k) = \frac{1}{n}\sum_{i=1}^{n} L_i(k)$, and $\mathcal{K}_{\mathrm{cert}} = \left\{k : \frac{n}{n+1}\widehat{R}_n(k) + \frac{1}{n+1} \le \alpha\right\}$. If this set is empty, return* `NON-CERTIFIED`*; otherwise return $\widehat{k} = \min \mathcal{K}_{\mathrm{cert}}$. For $0 \le \alpha < 1$,*

$$\mathbb{P}\big(\mathcal{K}_{\mathrm{cert}} \neq \varnothing \text{ and } C_{\widehat{k}}(X_{n+1}) \cap G_{n+1} = \varnothing\big) \le \alpha. \tag{7}$$

If certification occurs almost surely, then $\mathbb{E}[L_{n+1}(\widehat{k})] \le \alpha$.

The probability in (7) covers the joint event that a certificate is issued and the certified set still misses. It does not bound the miss rate among only those redraws on which a certificate is issued, because conditioning on issuance would select the redraws whose calibration sample was favorable, and no distribution-free bound of this form exists for that conditional quantity. The estimator, hyperparameters, family, tie rules, seeds, and numerical draws must all be fixed before final-calibration labels are inspected. The validation split must therefore select a family before final calibration, and a procedure that calibrates every candidate and then selects among the calibrated policies is not covered. Hao et al. (2026) augment the pool with a null action whose loss is zero by definition, so their construction controls a different quantity from the all-miss risk here, under which declining to call any model counts as a miss.

Corollary 1 (Minimum certified cutoff)**.** *Let a fixed score s generate the frozen sublevel family $C_\lambda(x) = \{j : s(x,j) \le \lambda\}$, and let $\Lambda_{\mathrm{cert}} = \left\{\lambda : \frac{n}{n+1}\widehat{R}_n(\lambda) + \frac{1}{n+1} \le \alpha\right\}$. If $\Lambda_{\mathrm{cert}} \neq \varnothing$, the smallest certified cutoff $\widehat{\lambda} = \min \Lambda_{\mathrm{cert}}$ is attained and satisfies $C_{\widehat{\lambda}}(x) \subseteq C_\lambda(x)$ for every $\lambda \in \Lambda_{\mathrm{cert}}$ and every x. Therefore, among all certified cutoffs in this frozen family, $\widehat{\lambda}$ calls the fewest models. The same holds for any cost that never decreases when a model is added to the called set.*

Essentially this means that once the order and the stopping score are frozen, every certified cutoff produces nested sets, the smallest certified cutoff produces the smallest of them for every query, and calibration therefore never trades calls against risk within the family but simply finds its cheapest certified member. The corollary is exact within one frozen family; it is not an optimality claim across different orderings, stopping rules, or policy classes. Appendix C.3 proves Theorem 2 and Corollary 1, and also shows that the correction $1/(n+1)$ cannot be reduced by any deterministic calibration rule of this form, so no risk target below $1/(n+1)$ is certifiable.

5 Experiments

We answer four questions, namely whether CopRoute calls fewer models than existing routing policies at a matched certified risk, whether dependence-corrected stopping improves on the independence-based default, whether an ordering fit on one dataset remains useful on another, and how much the results depend on the router that provides the scores.

Setup. We evaluate on RouterBench (Hu et al., 2024), whose pool of eleven models answers ARC, HellaSwag, MMLU, and WinoGrande, and we do not use its two remaining tasks, since the GSM8K release contains a single distinct prompt so that no prompt-conditioned router can be fitted, and on MBPP even the full pool misses 13.4 percent of queries so that neither target is certifiable. We also evaluate on RouterEval (Huang et al., 2025), which records per-query correctness for 3,811 models. From RouterEval we use six benchmarks with uniformly subsampled pools of 100, 200, 300, and 500 models; the main table reports the 100-model pool. Throughout, CopRoute without a suffix denotes the default policy CopRoute-Ind, and the four baselines are RACER-G, RACER-P, FlexRouter, and BaRP. A seventh RouterEval benchmark, MATH-intermediate-algebra, certifies for no policy at the targets we study and appears in Appendix H. All policies use the same KNN router scores, except that FlexRouter (Wang et al., 2026) and BaRP (Wang et al., 2025b) train their own routers, which we retrain with five seeds. Every cell uses 20 redraws with disjoint design, router, validation, final-calibration, and test partitions; all fitting and tuning happen before final calibration, after which the policy is frozen. We report the mean number of called models over certified redraws at each target risk α, and we state the certification rate wherever any redraw fails to certify.

Table 1: Certified mean calls at matched target risk, mean (dispersion) over 20 redraws; the dispersion is the standard error, or the standard deviation across five training seeds for FlexRouter and BaRP. CopRoute is the default policy CopRoute-Ind. The best value in each row is shaded, with exact ties shaded together, and the next distinct value is underlined. The last column is the exact-independence control, which orders models by router score alone and is reported but not ranked. RouterBench datasets (first four rows) use the 11-model pool and RouterEval datasets the 100-model pool.

(a) Target risk $\alpha = 0.05$.

		RACER				control
dataset	CopRoute	G	P	Flex.	BaRP	Indep.
ARC	1.25 (0.13)	1.39 (0.17)	2.44 (0.14)	1.33 (0.01)	1.37 (0.02)	1.48 (0.25)
HellaSwag	3.19 (0.09)	3.87 (0.08)	4.25 (0.10)	4.65 (0.13)	5.14 (0.40)	3.95 (0.11)
MMLU	7.75 (0.27)	8.66 (0.10)	8.64 (0.04)	9.67 (0.04)	9.74 (0.08)	9.45 (0.18)
WinoGrande	2.40 (0.03)	5.31 (0.20)	5.10 (0.21)	4.56 (0.15)	4.62 (0.18)	4.81 (0.19)
BBH-boolean	2.63 (0.29)	4.75 (0.58)	5.20 (0.46)	2.70 (0.22)	3.01 (0.31)	4.15 (0.57)
BBH-navigate	6.42 (3.63)	31.3 (5.30)	31.8 (5.15)	24.3 (1.68)	22.0 (2.97)	17.3 (1.63)
GPQA	17.2 (2.14)	21.0 (1.85)	21.0 (1.85)	26.3 (0.67)	25.7 (5.28)	22.6 (2.19)
IFEval	22.1 (5.97)	24.7 (5.04)	26.3 (5.41)	30.3 (3.84)	26.8 (3.29)	24.3 (5.19)
MATH-algebra	35.1 (14.7)	34.7 (18.8)	32.0 (15.5)	45.2 (2.98)	38.1 (13.1)	28.1 (12.3)
MuSR	57.2 (8.76)	75.0 (4.28)	77.0 (4.81)	86.3 (3.15)	87.8 (4.01)	69.3 (7.55)

(b) Target risk $\alpha = 0.10$.

		RACER				control
dataset	CopRoute	G	P	Flex.	BaRP	Indep.
ARC	1.00 (0.00)	1.06 (0.02)	1.71 (0.08)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
HellaSwag	1.80 (0.03)	1.77 (0.03)	2.24 (0.03)	2.51 (0.08)	2.61 (0.24)	1.80 (0.03)
MMLU	2.99 (0.05)	5.64 (0.07)	6.45 (0.04)	3.89 (0.06)	4.06 (0.10)	3.87 (0.07)
WinoGrande	1.80 (0.04)	3.10 (0.15)	3.00 (0.13)	2.38 (0.02)	2.61 (0.20)	2.59 (0.13)
BBH-boolean	1.13 (0.05)	2.25 (0.18)	3.24 (0.35)	1.23 (0.03)	1.34 (0.11)	1.57 (0.18)
BBH-navigate	1.86 (0.07)	7.36 (1.20)	7.61 (1.25)	6.35 (1.24)	6.90 (1.93)	5.95 (1.00)
GPQA	9.65 (0.95)	13.3 (0.93)	13.3 (0.93)	13.2 (0.15)	13.1 (2.46)	14.3 (0.99)
IFEval	2.86 (0.32)	3.76 (0.48)	5.66 (0.53)	7.31 (0.89)	5.16 (1.37)	3.22 (0.44)
MATH-algebra	11.4 (3.74)	11.0 (3.29)	10.4 (3.02)	12.1 (1.51)	17.8 (7.46)	9.79 (2.81)
MuSR	12.7 (1.56)	44.6 (2.81)	53.3 (1.67)	49.7 (5.11)	55.2 (3.95)	32.1 (3.16)

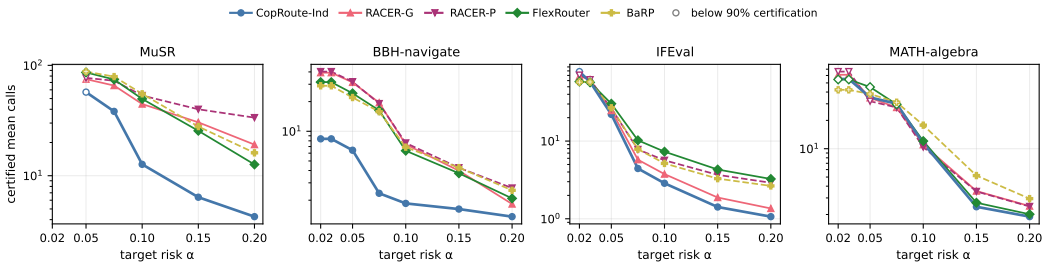
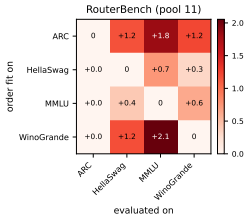


Figure 1: Certified mean calls against the target risk, 100-model pool, logarithmic vertical axis; lower is better. Markers are open where fewer than 90 percent of redraws certify, and a missing point means none certifies.

Certified calls at matched risk. Table 1 gives the main comparison. At $\alpha = 0.05$ every policy certifies all 20 redraws except on IFEval (19), MATH-algebra (3), and MuSR (8), identically across policies, whereas at $\alpha = 0.10$ every cell certifies all 20. CopRoute has the lowest calls on nine of the ten datasets at $\alpha = 0.05$ and on eight at $\alpha = 0.10$. The improvements are largest on the reasoning benchmarks. At $\alpha = 0.10$, MuSR needs 12.7 calls against 44.6 for RACER-G, the closest baseline, and 55.2 for BaRP, and BBH-navigate needs 1.9 against 6.4 for FlexRouter and 7.6 for RACER-P, while MMLU and WinoGrande need roughly half the calls of either RACER variant. The exceptions are MATH-algebra, where RACER-P is better by 3.1 calls at $\alpha = 0.05$ and by 1.0 call at $\alpha = 0.10$, and HellaSwag at $\alpha = 0.10$, where RACER-G is better by 0.03 calls. The exact-independence control orders models by router score alone and stops with the same independence product. It matches CopRoute or calls more models on nine datasets, by up to 19.4 calls more on MuSR, and is better on MATH-algebra by 7.0 calls at $\alpha = 0.05$ and 1.6 calls at $\alpha = 0.10$, so on that benchmark the design co-failure counts order the pool less well than the query-specific router scores do. Figure 3 extends the comparison to the 500-model pool, which the table does not show, and reports relative call efficiency, the per-dataset best calls divided by each policy's calls, so that the winner lies on the rim. Over the 200-, 300-, and 500-model pools (Figure 4 in Appendix H), CopRoute is on the rim on 29 of the 36 axes and FlexRouter on the remaining seven, five of them BBH-boolean or GPQA cells.

Risk sweep. Figure 1 plots certified calls over the full target grid. The CopRoute curve is the lowest across the sweep on MuSR, BBH-navigate, and IFEval, and the advantage widens as the target loosens. The sweep also shows where certification becomes infeasible, since no policy certifies MuSR below $\alpha = 0.05$ at this pool and every policy certifies fewer redraws at the tightest targets. Calibration can certify a target only if the full pool misses few enough calibration queries, namely when $(c_M + 1)/(n + 1) \leq \alpha$ for the full-pool miss count c_M among the n calibration queries. This



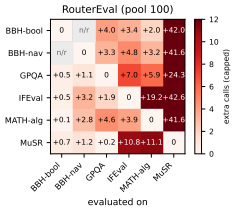


Figure 2: Extra certified calls for CopRoute at $\alpha = 0.10$ when the ordering is fitted on the row dataset and recalibrated on the column dataset. The color scale is capped at 12 and the annotated values are exact. Transfers between RouterBench and RouterEval are not possible because the two benchmarks have disjoint model pools, and the two pairs within the BBH family, marked n/r, were not run so that no transfer is near in-domain. Appendix J reports the same map for every policy.

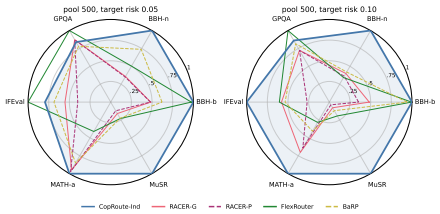


Figure 3: Relative call efficiency at the 500-model pool of the six RouterEval datasets, defined as the best certified calls on each axis divided by the policy's certified calls, so that the per-axis winner lies on the rim at 1. Figure 4 shows the 200- and 300-model pools.

boundary is therefore set jointly by the calibration-set size and by how often even the whole pool fails, and it is shared by all policies because it depends on neither the order nor the stopping rule.

Dependence-corrected stopping. Appendix I (Table 7) isolates the two components of the policy on the cells of Table 1. Replacing the independence-based stopping rule by the dependence-corrected rule of CopRoute-Dep changes nothing on 12 of the 20 cells, because validation selects $h = 0$ (on 10 to 19 of 20 redraws on IFEval and MuSR, the two benchmarks for which the selected values were recorded), and where it does change the result, it lowers calls on MATH-algebra at $\alpha = 0.10$ by 1.7 with an interval that includes zero and raises them on IFEval at $\alpha = 0.05$ by 5.6, so CopRoute-Ind is the default. Replacing the tally order by the design-marginal order, which ranks models by design accuracy alone, costs up to 21 calls on BBH-navigate and 27 on MuSR, so the gain over the baselines comes from the co-failure order rather than from the stopping correction.

Transfer across datasets. Figure 2 reports the extra certified calls when the ordering is fitted on one dataset and deployed on another. On RouterBench every transfer costs at most 2.1 extra calls, and most RouterEval transfers cost fewer than seven; transfers into MuSR are the most expensive at 24 to 43 extra calls. FlexRouter and BaRP degrade least under this shift because their orderings encode little dataset-specific structure (Appendix J); CopRoute degrades the most because its ordering is fitted to one dataset's co-failure structure, the property that makes it efficient in-domain. The ranking narrows, and averaged over all out-of-domain pairs CopRoute remains the lowest-call policy on RouterBench at both targets and on RouterEval at $\alpha = 0.05$ (33.3 calls against 34.6 to 37.6), while at $\alpha = 0.10$ it is 1.1 calls behind FlexRouter. Reusing a source-calibrated cutoff inflates the realized risk for every policy, while recalibrating on the deployment dataset restores the guarantee in every cell. The ordering can therefore be reused across related datasets, but the cutoff must be recalibrated on the deployment distribution.

Changing the router. We repeat the main comparison with RouterDC (Chen et al., 2024b) in place of the KNN router. Swapping the router moves CopRoute by at most 0.6 calls on any dataset, with a median absolute change of 0.2 calls at $\alpha = 0.10$, while RACER-G and RACER-P move by up to 31 calls with median changes of 2 to 3 calls. On RouterDC's own scores, CopRoute calls fewer models than RACER-G on all ten datasets at $\alpha = 0.10$.

Alternative estimators. CopRoute is a family of routing policies built from two interchangeable components. An estimator fitted on the design split produces the model order, and a stopping rule converts that order into a per-query candidate set before the shared calibration step certifies the cutoff. In Algorithm 1, a family variant changes only step 1, where an alternative estimator fits the order in place of the tally count, or step 2, where validation selects the stopping rule; scoring, calibration, and deployment are unchanged. Table 2 compares the alternative orders against the tally default under both stopping rules. No alternative improves on the tally order in aggregate; the closest, a Bernoulli mixture, uses 0.36 to 0.42 more calls, and a seeded random order uses about 7 more. At the 100-model pool the Bernoulli-mixture and Student-t-copula orders use 0.4 to 0.7 fewer

Table 2: Extra certified calls of each alternative order relative to the tally order under the same stopping rule, weighted by certified redraws over the ten datasets; negative (bold) favors the alternative.

	Ind stopping		Dep stopping	
order	all pools	pool 100	all pools	pool 100
Bernoulli mixture	+0.42	**-0.65**	+0.36	**-0.39**
Student-t copula	+1.77	**-0.68**	+1.35	**-0.67**
Gaussian copula	+1.97	**-0.09**	+1.81	**-0.10**
rank correlation	+2.04	+0.26	+1.83	+0.20
double fault	+1.59	+0.13	+1.13	+0.00
error correlation	+6.25	+0.98	+5.50	+1.23
accuracy-weighted random	+4.28	+2.54	+4.04	+2.58
random control	+7.78	+4.19	+7.24	+4.00

Table 3: Cost and final-answer accuracy on RouterBench MMLU at matched certified risk, means over 20 redraws; the best value per column is shaded. Standard errors and uncalibrated references appear in Table 15.

	$\alpha = 0.05$			$\alpha = 0.10$		
policy	calls	acc. (%)	USD/1k	calls	acc. (%)	USD/1k
CopRoute-Ind	7.75	80.44	3.41	2.99	80.97	1.90
RACER-G	8.66	80.19	4.24	5.64	80.18	3.55
RACER-P	8.64	80.19	4.22	6.45	80.13	3.66
exact independence	9.45	80.19	4.28	3.87	80.19	2.83
design-marginal order	9.44	80.20	4.30	3.76	80.66	2.86
call every model	11.00	80.19	4.47	11.00	80.19	4.47
best single model (1 call)	1.00	81.39	1.43	1.00	81.39	1.43

calls than the tally order, which indicates that estimators with more parameters become useful when the design set is large relative to the pool (definitions in Appendix F, complete grids in Appendix I).

Cost and final accuracy. The certified guarantee bounds coverage, and Table 3 therefore also reports final-answer accuracy and cost on RouterBench MMLU, with the answer taken from the called model with the highest router score and each call priced from public per-model prices. At $\alpha = 0.10$, CopRoute answers 80.97 percent of queries at USD 1.90 per thousand queries, against 80.18 percent at USD 3.55 for RACER-G, a 46.6 percent cost reduction with a 0.79-point accuracy gain at the same certified risk. Final-answer accuracy depends on the answer selector and the called set, and the two uncalibrated reference rows show that the certificate does not determine it. Calling every model returns the router's top choice at 80.19 percent, whereas the single model with the best design accuracy is both cheaper and more accurate than every calibrated policy, at USD 1.43 and 81.39 percent, but misses 18.6 percent of queries and therefore cannot certify either target. The calibrated policies add the certified coverage guarantee, and among them CopRoute reaches it at the lowest cost. Appendix L reports the standard errors and a cost-weighted variant for pools with heterogeneous prices.

6 Limitations and Conclusion

Summary. We pair a joint routing objective with a family of policies that learn which models add coverage beyond those already selected. The correctness-subcopula identity defines the objective, the tally order has a tight approximation guarantee for it, and split calibration certifies the deployed cutoff. At matched certified risk, CopRoute is the lowest-call routing policy on nine of ten datasets at $\alpha = 0.05$, on eight of ten at $\alpha = 0.10$, and on 29 of the 36 larger-pool cells, and it keeps that lead across the target sweep, across routers, and at pools of up to 500 models. The gain comes from the co-failure order, since replacing it by the design-marginal order costs up to 27 calls, while swapping the router changes CopRoute by at most 0.6 calls and degrading its scores by at most 0.4 calls.

Limitations. The certificate bounds the average all-miss risk of the candidate set over the deployment distribution, which is the quantity a routing layer controls, and per-query guarantees and certificates for the answer returned by an aggregator are separate objects outside the scope of this framework. The fitted order and the calibrated cutoff are specific to the deployment pool and distribution, both are inexpensive to refit, and Appendices J and L quantify the cost of reusing an order across related datasets or after a pool change. The dependence-corrected stopping rule is estimated from the design queries that miss an entire prefix, which are few at deep prefixes, and it does not improve on the independence-based default, so the open question is a tail estimator that uses more of the design set. Our evaluation uses recorded correctness labels, which permit the 20-redraw protocol across ten datasets and pools of up to 500 models, and it prices calls on one benchmark, while latency and response-dependent aggregation are left to deployment studies. Appendix M discusses each point in more detail.

AI USE STATEMENT

In this work, we used generative AI coding assistants to implement parts of the experiment code and the scripts that regenerate tables and figures from recorded result files; the authors reviewed and tested this code, and every reported number was regenerated from the recorded results. We have not used generative AI tools to develop the theoretical framework, to formulate or prove mathematical claims, to propose hypotheses, to design the experimental methodology, to generate data, or to interpret results. We also used generative AI tools to edit the paper for readability and to format references, and the authors reviewed every edited passage. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly released benchmark artifacts, RouterBench and RouterEval, which record per-query correctness of published language models; no human subjects, personal data, or new model outputs are involved. The intended use is reducing the number of model calls needed to reach a user-specified reliability target, which lowers the cost of serving language models. The certificates studied here bound the average all-miss risk over a deployment distribution; they are not per-query guarantees and are not a substitute for domain-specific safety review in high-stakes applications.

REPRODUCIBILITY STATEMENT

All experiments run on publicly available data. Section 5 and Appendix G specify the split construction, the 20-redraw protocol, the calibration rule, and the certification criterion; Appendix F defines every estimator and its hyperparameter grid; Appendix N lists seeds, split fractions, and the recorded result file behind each reported table. Every table and figure in the paper is generated by a script from a recorded result file. The code, the recorded result files, and one regeneration script per reported artifact are available at `https://anonymous.4open.science/r/dependence-aware-conformal-routing-2CD2/`.

REFERENCES

Raja Hafiz Affandi, Emily Fox, Ryan Adams, and Ben Taskar. Learning the parameters of determinantal point process kernels. In *International Conference on Machine Learning*, pp. 1224–1232, 2014.

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

Anastasios N. Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021. URL `https://arxiv.org/abs/2009.14193`.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024. arXiv:2208.02814.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2), 2025. arXiv:2110.01052.

Nikhil Bansal, Jatin Batra, Majid Farhadi, and Prasad Tetali. On min sum vertex cover and generalized min sum set cover. *SIAM Journal on Computing*, 52(2):327–357, 2023. doi: 10.1137/21M1434052.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning*, pp. 18, 2004. doi: 10.1145/1015330.1015432.

Josef Chen. When does combining language models help? a co-failure ceiling on routing, voting, and mixture-of-agents across 67 frontier models. *arXiv preprint arXiv:2606.27288*, 2026. URL https://arxiv.org/abs/2606.27288.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024a. arXiv:2305.05176.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 66305–66328, 2024b.

Xiangxiang Dai, Jin Li, Xutong Liu, Anqi Yu, and John C. S. Lui. Cost-effective online multi-LLM selection with versatile reward models. *arXiv preprint arXiv:2405.16587*, 2024. URL https://arxiv.org/abs/2405.16587.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 12987–13010, 2025. URL https://proceedings.mlr.press/v267/dekoninck25a.html.

D. Ding, A. Mallick, C. Wang, R. Sim, S. Mukherjee, V. Rühle, L. V. S. Lakshmanan, and A. H. Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024.

D. Ding, A. Mallick, S. Zhang, C. Wang, D. Madrigal, M. D. C. H. Garcia, M. Xia, L. V. S. Lakshmanan, Q. Wu, and V. Rühle. BEST-Route: Adaptive LLM routing with test-time optimal compute. In *International Conference on Machine Learning*, 2025.

David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A. Saurous, Jascha Sohl-Dickstein, Kevin Murphy, and Charles Sutton. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.

Uriel Feige, László Lovász, and Prasad Tetali. Approximating min sum set cover. *Algorithmica*, 40 (4):219–234, 2004. doi: 10.1007/s00453-004-1110-5.

Tao Feng, Yanzhen Shen, and Jiaxuan You. GraphRouter: A graph-based router for LLM selections. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.03834.

Giorgio Giacinto and Fabio Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9-10):699–707, 2001. doi: 10.1016/S0262-8856(01)00045-2.

Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Near-optimal MAP inference for determinantal point processes. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Shashwat Goel, Joschka Struber, Ilze Amanda Auzina, Karuna K Chandra, Ponnurangam Kumaraguru, Douwe Kiela, Ameya Prabhu, Matthias Bethge, and Jonas Geiping. Great models think alike and this undermines AI oversight. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *International Conference on Learning Representations*, 2024. arXiv:2404.10136.

Insu Han, Prabhanjan Kambadur, Kyoungsoo Park, and Jinwoo Shin. Faster greedy MAP inference for determinantal point processes. In *International Conference on Machine Learning*, pp. 1384–1393, 2017.

Sai Hao, Hao Zeng, Hongxin Wei, and Bingyi Jing. RACER: Risk-aware calibrated efficient routing for large language models. *arXiv preprint arXiv:2603.06616*, 2026.

Mahmoud Hegazy, Liviu Aolaritei, Michael I. Jordan, and Aymeric Dieuleveut. Valid selection among conformal sets. In *Advances in Neural Information Processing Systems*, volume 38, pp. 193550–193581, 2025. doi: 10.52202/085713-5812. URL https://arxiv.org/abs/2506.20173.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. RouterBench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. RouterEval: A comprehensive benchmark for routing LLMs to explore model-level scaling up in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023. arXiv:2306.02561.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 2012.

B. Kumar, C. Lu, G. Gupta, A. Palepu, D. Bellamy, R. Raskar, and A. Beam. Conformal prediction with large language models for multi-choice question answering. In *ICML Workshop on Neural Conversational AI*, 2023.

Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003. doi: 10.1023/A:1022859003006.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 661–670, 2010. doi: 10.1145/1772690.1772758.

Yang Li. LLM bandit: Cost-efficient LLM generation via preference-conditioned dynamic routing. *arXiv preprint arXiv:2502.02743*, 2025. URL https://arxiv.org/abs/2502.02743.

K. Lu, H. Yuan, R. Lin, J. Lin, Z. Yuan, C. Zhou, and J. Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1964–1974, 2024.

Roger B. Nelsen. *An Introduction to Copulas*. Springer, 2nd edition, 2006.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14(1):265–294, 1978.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, 2025. arXiv:2406.18665.

Karl Pearson. Mathematical contributions to the theory of evolution. VII. on the correlation of characters not quantitatively measurable. *Philosophical Transactions of the Royal Society of London. Series A*, 195:1–47, 1900.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, 2024. arXiv:2306.10193.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1802.09127.

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3581–3591, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/244edd7e85dc81602b7615cd705545f5-Abstract.html.

T. Shnitzer, A. Ou, M. Silva, K. Soule, Y. Sun, J. Solomon, N. Thompson, and M. Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023. URL https://arxiv.org/abs/2309.15789.

Abe Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, 8:229–231, 1959.

Ajay Narayanan Sridhar, Ronak Singh, Mehrdad Mahdavi, and Vijaykrishnan Narayanan. Correlation-aware contextual bandits with surrogate rewards for LLM routing. *arXiv preprint arXiv:2607.09015*, 2026. URL https://arxiv.org/abs/2607.09015.

K. V. Aditya Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. Harnessing the power of multiple minds: Lessons learned from LLM routing. In *Proceedings of the Fifth Workshop on Insights from Negative Results in NLP*, pp. 124–134, 2024. doi: 10.18653/v1/2024.insights-1.15.

J. Su, F. Lin, Z. Feng, H. Zheng, T. Wang, Z. Xiao, X. Zhao, Z. Liu, L. Cheng, and H. Wang. CP-Router: An uncertainty-aware router between LLM and LRM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 2026. arXiv:2505.19970.

Yigit Turkmen, Baturalp Buyukates, and Melih Bastopcu. Don't always pick the highest-performing model: An information theoretic view of LLM ensemble selection. *arXiv preprint arXiv:2602.08003*, 2026. URL https://arxiv.org/abs/2602.08003.

Clovis Varangot-Reille, Christophe Bouvard, Mathieu Ciancone, Antoine Gourru, Marion Schaeffer, and François Jacquenet. Doing more with less: A survey on routing strategies for resource optimisation in large language model-based systems. *Journal of Artificial Intelligence Research*, 86, 2026.

H. Vishwakarma, A. Mishler, T. Cook, N. Dalmasso, N. Raman, and S. Ganesh. Prune 'n predict: Optimizing LLM decision-making with conformal prediction. In *International Conference on Machine Learning*, 2025.

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

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, 2025a. arXiv:2406.04692.

Wei Wang, Tiankai Yang, Hongjie Chen, Yue Zhao, Franck Dernoncourt, Ryan A. Rossi, and Hoda Eldardiry. Learning to route LLMs from bandit feedback: One policy, many trade-offs. *arXiv preprint arXiv:2510.07429*, 2025b. URL https://arxiv.org/abs/2510.07429.

Wei Wang, Hongzheng Yang, Tiankai Yang, Samyadeep Basu, Hongjie Chen, Yue Zhao, Franck Dernoncourt, Ryan A. Rossi, and Hoda Eldardiry. FlexRouter: Learning complementary model sets for flexible LLM routing. In *Proceedings of the Conference on Language Modeling*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

Laurence A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982. doi: 10.1007/BF02579435.

Nan Xue, Shengkang Chen, Zhiyong Chen, Jiangchao Yao, Yaping Sun, Zixia Hu, and Meixia Tao. CR^2: Cost-aware risk-controlled routing for wireless device-edge LLM inference. *arXiv preprint arXiv:2605.12001*, 2026. URL https://arxiv.org/abs/2605.12001.

Yichi Zhang, Kevin Lu, Yuang Zhang, Jie Gao, Lirong Xia, and Fang-Yi Yu. Mixture of complementary agents for robust LLM ensemble. *arXiv preprint arXiv:2605.24048*, 2026. URL https://arxiv.org/abs/2605.24048.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with UCB-based exploration. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/1911.04462.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Fs9EabmQrJ.

Appendix contents

A Extended related work 16
B Notation reference. 18
C Proofs for the joint target, ordering, and calibration. 19
D Oracle, scope, and structured-estimator theory 24
E Implementation details 27
F Estimator definitions for the CopRoute family 29
G Protocol details and claim boundaries 30
H Detailed calibrated results 31
I Complete estimator grids for the CopRoute family 31
J Transfer across datasets for all policies 34
K Supplementary native learned-baseline metrics 35
L Cost, answer selection, and further analyses. 35
M Limitations. 39
N Reproducibility 41

A Extended related work

Language-model routing and cascades. Most routing work selects a single model per query from candidates that differ in quality and inference cost. FrugalGPT combines a routing stage with a cascade to reduce inference cost (Chen et al., 2024a), RouteLLM learns a preference-based decision between models (Ong et al., 2025), HybridLLM uses a quality threshold to route between a smaller and a larger model (Ding et al., 2024), and BEST-Route allocates a query-specific test-time compute budget (Ding et al., 2025). Learned routers include benchmark-trained predictors (Shnitzer et al., 2023), reward-guided expert routing (Lu et al., 2024), the dual-contrastive query and model representations of RouterDC (Chen et al., 2024b), the graph representation of GraphRouter (Feng et al., 2025), and the compact model representations of EmbedLLM (Zhuang et al., 2025). Language-model cascades instead follow a sequence and stop when an uncertainty or quality rule accepts an answer (Dohan et al., 2022; Gupta et al., 2024). Surveys and benchmark studies organize this literature (Srivatsa et al., 2024; Varangot-Reille et al., 2026; Hu et al., 2024; Huang et al., 2025). Most of this literature either returns a single model or follows a prescribed sequence, and it does not directly optimize the event that an entire selected set fails.

Bandit routers study the setting in which the system observes the outcome only of the candidate model it called, so that feedback on the models not called is never available. Contextual-bandit algorithms learn from this partial feedback (Li et al., 2010; Zhou et al., 2020; Riquelme et al., 2018). C2MAB-V applies an online bandit algorithm to cost-aware multi-LLM selection (Dai et al., 2024). LLM Bandit conditions a dynamic router on the user's quality and cost preference (Li, 2025). BaRP combines a preference-conditioned router with a policy-gradient algorithm that observes only the chosen model's feedback (Wang et al., 2025b). BaRP is designed as a single-model policy, so our learned-baseline experiments use its frozen quality logits only to define a model ordering for the common set-routing evaluation, whereas its original single-model quality and cost objective is reported separately.

Risk-controlled routing and conformal selection. RACER uses a calibration set to convert any base router into a set-valued routing policy (Hao et al., 2026); it chooses a threshold for the score assigned to every candidate language model, and the threshold produces nested model sets and controls the probability that every selected model is wrong. CP-Router uses conformal uncertainty to choose between language and reasoning models (Su et al., 2026), and because it selects between exactly two systems without forming candidate sets over a pool, it has no direct counterpart in the set-routing comparison. CR2 studies a cost-aware conformal loss for a wireless device-edge router (Xue et al., 2026). These methods establish the value of a post-hoc calibration step, but their candidate families are constructed from scores attached to individual models. CopRoute retains the risk target and final calibration step of RACER and changes how the candidate family is ordered and sized.

This calibration step draws on the split-conformal framework, which gives distribution-free uncertainty guarantees (Vovk et al., 2005; Angelopoulos & Bates, 2023). Conformal risk control extends these ideas to bounded monotone losses (Angelopoulos et al., 2024), and Learn then Test provides a multiple-testing procedure for controlling risk (Angelopoulos et al., 2025). Conformal language modeling constructs sets of acceptable generated responses (Quach et al., 2024). Other language-model applications use conformal methods for multiple-choice questions and efficient LM decisions (Kumar et al., 2023; Vishwakarma et al., 2025). Adaptive and regularized adaptive prediction sets use cumulative class probabilities to vary set size across inputs (Romano et al., 2020; Angelopoulos et al., 2021). Their labels are mutually exclusive, whereas several routed models may be correct on the same query. Selecting among conformal constructions after the calibration step can invalidate the original guarantee; and valid selection among conformal sets has been studied as a problem in its own right (Hegazy et al., 2025). CopRoute chooses its estimator and other settings before the final calibration step, then calibrates only the frozen nested family.

Complementary model sets and aggregation. Dekoninck et al. (2025) study the tradeoff between the quality of an answer and its inference cost, and their set objective includes the expected quality of the best answer in a selected set, which for binary outcomes equals the probability that at least one selected model is correct. Their estimate of this joint quantity uses per-model quality estimates and per-model uncertainty variances fitted on validation data; the co-occurrence of failures across models

is not modeled. FlexRouter learns complementary, variable-size model sets with a determinantal point process (Wang et al., 2026); it combines a query-dependent competence score with a learned model-similarity score, then uses greedy determinant gains to select a set. Its original objective raises the probability of selecting at least one correct model and reduces redundancy, but does not provide a distribution-free all-miss certificate. Our FlexRouter reconstruction preserves the DPP ordering mechanism and places the resulting frozen ordering inside the same set-routing evaluation used for CopRoute.

Complementary set routing also connects to the ensemble-selection literature. Greedy selection from a model library can improve aggregate predictions (Caruana et al., 2004), while double-fault and error-correlation measures quantify whether classifiers fail together (Giacinto & Roli, 2001; Kuncheva & Whitaker, 2003). Turkmen et al. (2026) select models with a mutual-information objective, Zhang et al. (2026) train complementary proposers for a summarizer, and Chen (2026) measures how co-failure limits routers, voting rules, and mixtures of agents. LLM-Blender ranks and fuses responses after several models have already been called (Jiang et al., 2023). Self-consistency aggregates multiple reasoning paths from one model (Wang et al., 2023), while Mixture-of-Agents uses outputs from several agents in a layered architecture (Wang et al., 2025a). These aggregation methods address how to combine available responses, whereas CopRoute determines which candidate models to call and controls the all-miss risk.

Dependence models, diversity, and ordering. Copulas separate marginal behavior from dependence (Sklar, 1959; Nelsen, 2006). For binary outcomes, latent Gaussian models trace to the tetrachoric-correlation construction (Pearson, 1900). The correctness-subcopula identity of Section 3 is this separation applied to binary correctness, in which the marginals are the per-model accuracies that the router estimates and the subcopula is the co-failure structure that the design set estimates. The empirical tally estimator works with observed correctness patterns and never fits a parametric copula. Correlation-aware contextual bandits also model statistical relationships among model rewards, but optimize an online surrogate-reward objective rather than calibrated set failure (Sridhar et al., 2026).

Determinantal point processes are a probabilistic model for diverse subsets (Kulesza & Taskar, 2012). Prior work studies learning DPP kernels (Affandi et al., 2014) and greedy MAP inference (Gillenwater et al., 2012; Han et al., 2017). FlexRouter brings this quality-diversity construction to language-model routers. CopRoute orders models with the tally score, which counts how many previously unanswered design queries a candidate model answers.

The coverage of a selected model set is a monotone submodular function, so greedy fixed-cardinality selection inherits the classical approximation guarantee (Nemhauser et al., 1978). Reaching a required coverage level connects to submodular set cover (Wolsey, 1982). Optimizing one fixed ordering for the average position of the first correct model is an instance of minimum-sum set cover (Feige et al., 2004; Bansal et al., 2023), which gives the tight polynomial-time approximation guarantee used by CopRoute. These deterministic ordering results do not themselves provide a statistical certificate; that guarantee comes from the independent conformal calibration step.

B NOTATION REFERENCE

Table 4 collects the symbols used throughout the main text and the appendix. Symbols that appear only inside individual theorems or proofs, and the per-method parameters of the learned baselines, are defined where they first occur.

Table 4: Notation used in the paper.

Symbol	Meaning	Note
Basic routing objects		
$\mathcal{M} = \{1, \dots, m\}$	Pool of candidate language models.	m is the number of models.
i, j, k, ℓ	Indices for a labeled query, a model, a prefix depth, and a position inside a prefix, respectively.	The range follows the equation in which an index appears.
X, x	A random query and one realized query, respectively.	Uppercase denotes a random variable.
$Y_j, Y = (Y_1, \dots, Y_m)$	Binary indicator that model j answers X correctly, and the vector of all model outcomes.	$Y_j \in \{0, 1\}$.
$G(X), G_i$	Models that answer query X, or labeled query i, correctly.	Both are subsets of $\mathcal{M}$.
$S, T \subseteq \mathcal{M}$	Generic sets of models.	Local proofs may use either letter.
$C(X)$	Set of models returned by a routing policy for query X.	The objective minimizes $\mathbb{E}\lvert C(X)\rvert$.
$p_j(x)$	True probability that model j is correct on query x.	$p_j(x) = \mathbb{P}(Y_j = 1 \mid X = x)$.
$\widehat{p}_j(x)$	Router estimate of $p_j(x)$; in the experiments it is a clipped average over 40 nearest router-training queries.	Query-specific marginal accuracy.
$q_x(S)$, $q(S)$	Probability that every model in S is wrong, conditional on x or for the population.	All-miss probability.
$c_x(S)$, $c(S)$	Probability that at least one model in S is correct.	Coverage; $c_x(S) = 1 - q_x(S)$.
$C^0_{S,x}$	Correctness subcopula identified by the Bernoulli joint law on the values its marginal distributions can attain.	$q_x(S) = C^0_{S,x}(1 - p_j(x) : j \in S)$.
Data and marginal estimates		
$D \in \{0, 1\}^{n_d \times m}$	Design correctness table; $D_{ij} = 1$ when model j answers design query i correctly.	n_d is the number of design queries.
$F_{ij} = 1 - D_{ij}$	Failure indicator for model j on design query i.	Input to the dependence estimators.
$\bar{p}^D_j$	Fraction of design queries answered correctly by model j.	$\bar{p}^D_j = n_d^{-1} \sum_i D_{ij}$.
$N(x)$, Z_{rj}	Fixed set of 40 nearest router-training queries for x, and the correctness of model j on neighbor $r \in N(x)$.	No design, calibration, validation, or test label enters this set.
$\mathcal{D}_{\mathrm{des}}, \mathcal{D}_{\mathrm{val}}, \mathcal{D}_{\mathrm{cal}}$	Design, validation, and final-calibration datasets.	Disjoint splits.
U	All auxiliary random choices frozen before final calibration.	Includes seeds, ties, smoothing, and numerical draws.
Ordering and prefixes		
π, π_x	Ordering rule and the resulting permutation of models for query x.	An order may be global or query-specific.
$P^\pi_k(x)$	Set containing the first k models in order π_x.	$P^\pi_0(x) = \varnothing$.
π^D, π^T	Fixed order by decreasing design accuracy, and the fixed tally order of (4).	Superscript T means tally, not transpose.
$\widehat{q}_D(S)$	Fraction of design queries on which all models in S are wrong.	$n_d^{-1} \sum_i \prod_{j \in S}(1 - D_{ij})$.

Continued on the next page

Table 4 continued

Symbol	Meaning	Note
R_{k-1}	Design queries still unanswered after prefix P_{k-1}.	A set of rows, not a correlation matrix.
Stopping scores		
$r_k(x)$	Estimated all-miss residual assigned to the first k ordered models.	Generic sizing quantity; $r_0(x) = 1$.
$r_k^{\mathrm{ind}}(x;\pi)$	Independence-based all-miss estimate for the first k ordered models.	Product of query-specific marginal miss probabilities.
$r_k^{\mathrm{dep}}(x;h,\pi)$	Dependence-corrected all-miss estimate.	Adjusts r_k^{ind} with the design-set dependence ratio; $h = 0$ recovers r_k^{ind}.
$d_k, \widetilde{d}_k(h), h$	Raw dependence ratio, its shrunk value, and the minimum-support parameter.	Defined in Appendix E.
$r_k^{\mathrm{loc}}(x;\pi)$	Fraction of the 40 router-training neighbors missed by every model in the first k positions.	Local empirical stopping.
$s_\pi(x,\pi_{x,k}), s(x,j)$	Entry score of the model at position k, one minus the preceding residual.	Smaller scores enter first.
λ, $C_\lambda(x)$	Cutoff on entry scores, and the model set it selects.	$C_\lambda(x) = \{j : s(x,j) \le \lambda\}$.
Calibration and certification		
α	User-specified target for the all-miss risk.	Reported targets range from 0.02 to 0.20.
n, X_i, G_i	Number of final-calibration queries, calibration query i, and its set of correct models.	Exchangeable with a future query.
$L^{\mathrm{miss}}(S,G)$	Standard all-miss loss, equal to one when $S \cap G = \varnothing$.	An empty called set has loss one.
$\widehat{R}_n(\lambda), \widehat{R}_n(k)$	Average calibration loss at a cutoff or depth.	Empirical all-miss risk.
$\mathcal{K}_{\mathrm{cert}}, \Lambda_{\mathrm{cert}}$	Depths and cutoffs that pass the finite-sample calibration rule.	Empty means no certificate is issued.
$\widehat{k}, \widehat{\lambda}$	Smallest certified depth and smallest certified cutoff.	Call-minimal within the frozen family.
c	Number of calibration queries missed at a given cutoff.	The rule is $(c+1)/(n+1) \le \alpha$.
`NON-CERTIFIED`	Output when no depth or cutoff passes calibration.	No policy is deployed under this certificate.
Success@k	Fraction of queries for which at least one of the first k selected models is correct.	Fixed-depth coverage metric.

C Proofs for the joint target, ordering, and calibration

C.1 Formal joint target and independence benchmark

Let $\mathcal{M} = \{1, \dots, m\}$ be the model pool. For a query X, let $Y_j = \mathbf{1}\{\text{model } j \text{ is correct on } X\}$. For $S \subseteq \mathcal{M}$, recall

$$q_x(S) = \mathbb{P}(Y_j = 0 \text{ for every } j \in S \mid X = x), \qquad c_x(S) = 1 - q_x(S).$$

The notation without subscript x denotes probabilities under one fixed population law.

Proposition 1 (Identified correctness subcopula). *Fix x admitting a regular conditional law of $Y = (Y_1, \dots, Y_m)$, and let $S \neq \varnothing$. Write $p_j(x) = \mathbb{P}(Y_j = 1 \mid X = x)$. If $F_{j,x}$ is the conditional CDF of Y_j and $H_{S,x}$ is the conditional joint CDF of $(Y_j : j \in S)$, then a unique subcopula $C^0_{S,x}$ on the attainable grid satisfies*

$$H_{S,x}(y_S) = C^0_{S,x}\big(F_{j,x}(y_j) : j \in S\big).$$

In particular,

$$q_x(S) = C^0_{S,x}\big(1 - p_j(x) : j \in S\big). \tag{8}$$

The Bernoulli law identifies this subcopula only on $\prod_{j\in S}\{0, 1 - p_j(x), 1\}$, with repeated points removed. A full copula extension away from this grid is generally nonunique.

Proof. For a Bernoulli correctness variable,

$$F_{j,x}(0) = \mathbb{P}(Y_j \le 0 \mid X = x) = 1 - p_j(x).$$

The discrete form of Sklar's theorem gives a unique subcopula on the Cartesian product of the marginal CDF ranges. Evaluating the joint CDF at the all-zero correctness vector gives

$$q_x(S) = H_{S,x}(0_S) = C^0_{S,x}\big(F_{j,x}(0) : j \in S\big) = C^0_{S,x}\big(1 - p_j(x) : j \in S\big).$$

For discrete margins, Sklar's theorem does not identify a unique extension outside the attainable grid. □

Orthant convention. Equation (8) is a lower-orthant probability for the correctness vector. If $B_j = 1 - Y_j$ denotes failure, the all-miss event is an upper-orthant probability for B. It is not the lower-orthant copula value of the failure vector evaluated at its failure marginals.

Lemma 1 (Individual-accuracy ordering under conditional independence)**.** *Fix x and suppose $Y_1, \dots, Y_m$ are mutually independent conditional on $X = x$. Assume equal call costs and one fixed tie rule. If π orders the models so that*

$$p_{\pi_1}(x) \ge p_{\pi_2}(x) \ge \cdots \ge p_{\pi_m}(x),$$

then for every k, the prefix $P_k = \{\pi_1, \dots, \pi_k\}$ minimizes $q_x(S)$ over all model sets S with $|S| = k$.

Proof. Conditional independence gives

$$q_x(S) = \prod_{j\in S}\big(1 - p_j(x)\big).$$

Suppose a size-k set contains model a but excludes model b, with $p_b(x) > p_a(x)$. Replacing a by b replaces the factor $1 - p_a(x)$ by the no-larger factor $1 - p_b(x)$, so the all-miss probability cannot increase. Repeating this exchange until the set consists of the k largest marginals produces P_k. The fixed tie rule makes these prefixes nested. □

Independence is sufficient but not necessary, since dependent outcomes may preserve individual-accuracy ordering or reverse it. Example 1 gives an exact finite law in which the individually best model is not part of the best pair.

C.2 Ordering proof

Throughout this subsection, $D \in \{0,1\}^{n_d \times m}$ is the realized design correctness matrix,

$$G_i^D = \{j : D_{ij} = 1\}, \qquad U_D = \{i : G_i^D = \varnothing\}, \qquad u_D = \frac{|U_D|}{n_d} = \widehat{q}_D(\mathcal{M}).$$

The design tally has the counting form

$$\widehat{q}_D(S) = \frac{1}{n_d}\big|\{i : S \cap G_i^D = \varnothing\}\big|, \qquad \widehat{q}_D(\varnothing) = 1. \tag{9}$$

Prefix-area identity. For an ordering π, define

$$T_i(\pi) = \big|\{k \in \{0, \dots, m-1\} : P_k^\pi \cap G_i^D = \varnothing\}\big|.$$

If design query i is answerable, $T_i(\pi)$ is the position of its first correct model; if $i \in U_D$, then $T_i(\pi) = m$. Exchanging two finite sums yields

$$J_D(\pi) = \sum_{k=0}^{m-1} \widehat{q}_D(P_k^\pi) = \frac{1}{n_d}\sum_{i=1}^{n_d} T_i(\pi) = \frac{1}{n_d}\sum_{i\notin U_D} T_i(\pi) + m\,u_D. \tag{10}$$

The last term is independent of π.

Reduction to greedy minimum-sum set cover. Let

$$R_{k-1} = \{i : P_{k-1} \cap G_i^D = \varnothing\}$$

be the design rows still unanswered after the current prefix. For any candidate model $j \notin P_{k-1}$,

$$n_d \left[\widehat{q}_D(P_{k-1}) - \widehat{q}_D(P_{k-1} \cup \{j\})\right] = \left|\{i \in R_{k-1} : D_{ij} = 1\}\right|. \tag{11}$$

Thus minimizing the next tally value is exactly the same as choosing the model that covers the most still-unanswered rows. Rows in U_D contribute zero to every candidate count. Once only those rows remain, every completion of the ordering has the same value of J_D.

Proof of Theorem 1. Consider the set system whose elements are the answerable design rows $i \notin U_D$ and whose set for model j is

$$S_j = \{i \notin U_D : D_{ij} = 1\}.$$

By (10), the first position containing a set that covers row i is $T_i(\pi)$. Hence

$$A(\pi) = \sum_{i \notin U_D} T_i(\pi)$$

is exactly the minimum-sum set-cover objective. Equation (11) shows that π^T is a greedy execution for this instance. The greedy algorithm is a factor-four approximation for minimum-sum set cover (Feige et al., 2004), so

$$A(\pi^T) \le 4 \min_{\pi} A(\pi).$$

Substituting $J_D(\pi) = A(\pi)/n_d + m\,u_D$ gives

$$J_D(\pi^T) - m\,u_D \le 4\left(\min_{\pi} J_D(\pi) - m\,u_D\right)$$

and therefore $J_D(\pi^T) \le 4 \min_\pi J_D(\pi)$.

For hardness, start from an arbitrary minimum-sum set-cover instance in which every element belongs to at least one set. Define $D_{ij} = 1$ exactly when element i belongs to set j. Then $U_D = \varnothing$, and $A(\pi)$ is the objective of the original instance. A polynomial-time algorithm with ratio $4 - \varepsilon$ for every correctness matrix would therefore give the same ratio for every minimum-sum set-cover instance, which is impossible for any $\varepsilon > 0$ unless P = NP (Feige et al., 2004). □

Theorem 3 (Fixed-budget coverage guarantee)**.** *For any fixed law of the correctness indicators, the coverage*

$$c(S) = \mathbb{P}\left(\bigcup_{j \in S} \{Y_j = 1\}\right)$$

is normalized, monotone, and submodular. Let $G_0 = \varnothing$, and repeatedly add the model with the largest true coverage gain. For every fixed $k \ge 1$, the resulting set G_k satisfies

$$c(G_k) \ge \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \max_{|S|=k} c(S) \ge \left(1 - \frac{1}{e}\right) \max_{|S|=k} c(S). \tag{12}$$

The same statement holds for c_x after fixing x.

Proof. For $A \subseteq B$ and $j \notin B$,

$$c(A \cup \{j\}) - c(A) = \mathbb{P}(Y_j = 1,\ Y_i = 0 \text{ for every } i \in A),$$

while the analogous event with B is a subset of this event. This proves diminishing returns and therefore submodularity. Let O_k maximize coverage among size-k sets and write $\mathrm{OPT} = c(O_k)$. By monotonicity and submodularity,

$$\mathrm{OPT} \le c(G_t) + \sum_{j \in O_k \setminus G_t} \left[c(G_t \cup \{j\}) - c(G_t)\right].$$

There are at most k terms, so the next greedy gain is at least $(\mathrm{OPT} - c(G_t))/k$. Iterating from $c(G_0) = 0$ for k steps gives the first inequality in (12); the second uses $(1 - 1/k)^k \le e^{-1}$. □

Under the uniform law on design rows, the true greedy step in Theorem 3 is exactly the tally step in (11). Thus the same fixed tally ordering has the per-depth guarantee (12) and the whole-order guarantee in Theorem 1. The first compares one depth with the best set at that depth; the second compares the area of one complete ordering with the best complete ordering.

C.3 CALIBRATION PROOF

Let $\mathcal{D}_{\mathrm{des}}$ and $\mathcal{D}_{\mathrm{val}}$ be the design and validation data, and let U contain every auxiliary random choice, including seeds, numerical draws, clustering choices, tie rules, smoothing, and candidate-family selection. Before final-calibration labels are inspected, these quantities determine one measurable nested map

$$(x, k) \longmapsto C_k(x; \mathcal{D}_{\mathrm{des}}, \mathcal{D}_{\mathrm{val}}, U) \subseteq \mathcal{M}, \qquad k = 0, \ldots, K.$$

Conditioning on the frozen information leaves the calibration observations and future observation exchangeable.

Proof of Theorem 2. For observation i, define the first-success depth

$$D_i = \min\{k \in \{0, \ldots, K\} : L_i(k) = 0\},$$

with $D_i = K + 1$ if no set in the family succeeds. Nestedness gives $L_i(k) = \mathbf{1}\{D_i > k\}$, and $D_1, \ldots, D_{n+1}$ are exchangeable conditional on the frozen information.

If $\alpha < 1/(n+1)$, no depth can pass and the certified-miss event is empty. Otherwise, set

$$r = \lfloor \alpha(n+1) \rfloor - 1.$$

A depth k is certified exactly when

$$\sum_{i=1}^{n} \mathbf{1}\{D_i > k\} \leq r.$$

Let $D_{(1)} \leq \cdots \leq D_{(n)}$ be the calibration order statistics. A certificate is issued exactly when $D_{(n-r)} \leq K$, and then $\widehat{k} = D_{(n-r)}$. A future miss implies $D_{n+1} > D_{(n-r)}$, so the future depth occupies one of the largest $r+1$ ranks among the $n+1$ exchangeable depths. Therefore

$$\mathbb{P}\Big(\mathcal{K}_{\mathrm{cert}} \neq \varnothing \text{ and } L_{n+1}(\widehat{k}) = 1 \mid \mathcal{D}_{\mathrm{des}}, \mathcal{D}_{\mathrm{val}}, U\Big) \leq \frac{r+1}{n+1} \leq \alpha.$$

Averaging over the frozen information proves the unconditional claim. If certification occurs almost surely, the certificate event can be removed. □

Proof of Corollary 1. Nestedness makes $\widehat{R}_n(\lambda)$ nonincreasing, so Λ_{cert} is an upper set. The finite calibration sample generates at most nm distinct scores,

$$\mathcal{S}_{\mathrm{cal}} = \{s(X_i, j) : 1 \leq i \leq n,\ j \in \mathcal{M}\}.$$

Hence $\widehat{R}_n$ is a right-continuous step function whose jumps occur only at values in $\mathcal{S}_{\mathrm{cal}}$. The infimum of a nonempty Λ_{cert} is therefore attained. Every certified λ satisfies $\lambda \geq \widehat{\lambda}$, so

$$C_{\widehat{\lambda}}(x) \subseteq C_{\lambda}(x)$$

for every query x. Applying any cost that is nondecreasing under set inclusion preserves this inequality, both pointwise and in expectation. □

Continuous-cutoff form. The score rule (3) also reduces to a one-dimensional conformal statement. For calibration query i, let $S_i^* = \min_{j \in G_i} s(X_i, j)$, with $S_i^* = +\infty$ when $G_i = \varnothing$. The score map is frozen before calibration, so $S_1^*, \ldots, S_{n+1}^*$ are exchangeable, and the all-miss loss of C_λ on query i equals $\mathbf{1}\{S_i^* > \lambda\}$. With $r = \lfloor \alpha(n+1) \rfloor - 1$ as in Lemma 2, the rule (3) returns $\widehat{\lambda} = S^*_{(n-r)}$, the $(n-r)$th order statistic of the calibration scores, and certifies exactly when this value is finite. A certified miss on the next query is the event $\{S^*_{n+1} > S^*_{(n-r)}\}$, whose probability is at most $(r+1)/(n+1) \leq \alpha$ by exchangeability.

C.3.1 EXACT RANK FORM AND SHARPNESS

Lemma 2 (Exact rank form). *Let D_i be the first-success depths defined in the proof of Theorem 2 and set $r = \lfloor \alpha(n+1) \rfloor - 1$. When $1/(n+1) \le \alpha < 1$, depth k is certified if and only if $|\{i \le n : D_i > k\}| \le r$. A certificate is issued if and only if $D_{(n-r)} \le K$, and then $\widehat{k} = D_{(n-r)}$. Moreover,*

$$\mathbb{P}\big(\mathcal{K}_{\mathrm{cert}} \neq \varnothing \text{ and } L_{n+1}(\widehat{k}) = 1\big) \le \frac{\lfloor \alpha(n+1) \rfloor}{n+1} \le \alpha. \tag{13}$$

Proof. Write $c_k = |\{i \le n : L_i(k) = 1\}|$. Since c_k is an integer,

$$\frac{n}{n+1}\frac{c_k}{n} + \frac{1}{n+1} \le \alpha \quad \Longleftrightarrow \quad c_k \le \lfloor \alpha(n+1) \rfloor - 1 = r.$$

Because $L_i(k) = \mathbf{1}\{D_i > k\}$, this is equivalent to $D_{(n-r)} \le k$. The remaining claims follow from the proof of Theorem 2. □

Theorem 4 (Sharp deterministic rank correction). *Fix $n \ge 1$ and $1/(n+1) \le \alpha < 1$, and let $r = \lfloor \alpha(n+1) \rfloor - 1$.*

1. *There is an exchangeable one-model problem for which the probability in* (13) *equals $\lfloor \alpha(n+1) \rfloor / (n+1)$.*
2. *If a deterministic rule accepts one more calibration miss than r, there is an exchangeable one-model problem on which its certified-miss probability exceeds α.*

Thus the additive correction $1/(n+1)$ in (3) *is the smallest valid correction for deterministic distribution-free rank calibration.*

Proof. Take $m = 1$, $K = 1$, $C_0(x) = \varnothing$, and $C_1(x) = \{1\}$. Let $Y^{(1)}, \dots, Y^{(n+1)}$ be a uniformly random permutation of a fixed binary vector with exactly z zeros. This law is exchangeable. The first-success depth is 1 when $Y^{(i)} = 1$ and 2 otherwise.

For the first claim, choose $z = r + 1$. Depth zero calls no model and therefore misses every calibration query, so it never passes calibration, whereas depth one passes calibration exactly when the future observation is one of the z zeros, and on that event the certified set misses. The probability is therefore

$$\frac{r+1}{n+1} = \frac{\lfloor \alpha(n+1) \rfloor}{n+1}.$$

For the second claim, suppose a rule accepts at most $t \ge r+1$ calibration misses. If $t \ge n$, it accepts the empty set and misses with probability one. Otherwise choose $z = t + 1$. The same construction certifies exactly when the future observation is a zero, giving probability

$$\frac{t+1}{n+1} \ge \frac{r+2}{n+1} > \alpha.$$

Therefore no deterministic rule can accept more misses while retaining the guarantee for every exchangeable law. Replacing the additive one by any constant $b < 1$ admits such an extra miss at some attainable target and fails by the same construction. □

Corollary 2 (Certification floor). *If $\alpha < 1/(n+1)$, no depth can pass the calibration rule, even with zero observed calibration misses. For $\alpha \ge 1/(n+1)$, the certified level is $\lfloor \alpha(n+1) \rfloor / (n+1)$, so the attainable deterministic levels are integer multiples of $1/(n+1)$.*

Proof. The corrected empirical risk is always at least $1/(n+1)$, which proves the first claim. Lemma 2 gives the second claim, and Theorem 4 shows that the bound is attained. □

D ORACLE, SCOPE, AND STRUCTURED-ESTIMATOR THEORY

D.1 ORACLE AND SCOPE RESULTS

For one fixed query or one fixed population law, define the best all-miss probability attainable by any size-k model set as

$$q_k^* = \min_{S \subseteq \mathcal{M}:\, |S|=k} q(S). \tag{14}$$

These values are useful lower bounds, but independently optimal sets at different sizes do not automatically form one routing order.

Proposition 2 (Cardinality-oracle sets need not be nested)**.** *For $k = 0, \dots, m-1$, $q_{k+1}^* \le q_k^*$, and every size-k set T satisfies $q(T) \ge q_k^*$. Nevertheless, an optimizer at size $k+1$ need not contain any optimizer at size k.*

Proof. Choose an optimizer S_k^* of size $k < m$ and any model $j \notin S_k^*$. The event that every model in $S_k^* \cup \{j\}$ fails is contained in the event that every model in S_k^* fails. Hence

$$q_{k+1}^* \le q(S_k^* \cup \{j\}) \le q(S_k^*) = q_k^*.$$

The lower bound for T follows from the definition of the minimum. Example 1 proves the last claim. □

Example 1 (The best singleton and pair need not nest). Consider three models and four population atoms whose probabilities and failing models are

atom	probability	failing models
a	0.2	$\{1, 2\}$
b	0.3	$\{2\}$
c	0.2	$\{1, 3\}$
d	0.3	$\{3\}$.

The singleton miss probabilities are $q(\{1\}) = 0.4$ and $q(\{2\}) = q(\{3\}) = 0.5$, so $\{1\}$ is the unique best singleton. The pair miss probabilities are

$$q(\{1, 2\}) = q(\{1, 3\}) = 0.2, \qquad q(\{2, 3\}) = 0,$$

so $\{2, 3\}$ is the unique best pair, and the two optimizers are not nested.

Proposition 3 (Target-specific oracle headroom)**.** *Let $(M_k)_{k=0}^m$ be a fixed marginal-accuracy prefix chain under the same population set function q, and assume that the stopping depths*

$$k_\alpha^* = \min\{k : q_k^* \le \alpha\}, \qquad k_\alpha^{\mathrm{marg}} = \min\{k : q(M_k) \le \alpha\} \tag{15}$$

exist. Then

$$k_\alpha^* < k_\alpha^{\mathrm{marg}} \iff \exists S \subseteq \mathcal{M} : |S| < k_\alpha^{\mathrm{marg}} \text{ and } q(S) \le \alpha. \tag{16}$$

Any optimizer S_α^* of size k_α^* can be placed first in a target-specific nested order whose first successful prefix has size k_α^*.

Proof. If $k_\alpha^* < k_\alpha^{\mathrm{marg}}$, choose an optimizer S_α^* at size k_α^*. It satisfies the right side of (16). Conversely, if a set S of size $\ell < k_\alpha^{\mathrm{marg}}$ has $q(S) \le \alpha$, then

$$q_\ell^* \le q(S) \le \alpha,$$

so $k_\alpha^* \le \ell < k_\alpha^{\mathrm{marg}}$. For the final claim, order the elements of S_α^* first and append all remaining models. No smaller prefix can meet the target, because every set of smaller size has all-miss probability greater than α. □

This proposition is an existence result for one target, and a different target may require a different subset and order. It does not construct an estimated policy or one chain that attains every q_k^*.

Cost-aware oracle and heuristic. If model j has positive call cost a_j, the exact population reference at target α is

$$S^*_{\mathrm{cost}}(\alpha) \in \arg\min_{S\subseteq\mathcal{M}} \sum_{j\in S} a_j \quad \text{subject to} \quad q(S) \le \alpha. \tag{17}$$

Its optimizers are in general not nested, and a fixed order may instead append

$$j^*(S;x) \in \arg\max_{j\notin S} \frac{\widehat{q}_x(S) - \widehat{q}_x(S\cup\{j\})}{a_j}. \tag{18}$$

This is a greedy cost heuristic and does not minimize cost. Once its family is selected without final-calibration labels and then frozen, Theorem 2 can certify its all-miss risk. No cost-approximation ratio follows from that certification.

Matching an ordering does not imply that one policy's sets contain the other's. Consider one model, two query values x_a, x_b, and a score threshold τ such that

$$s_1(x_a) < \tau < s_1(x_b).$$

A threshold policy with a null action may select

$$A_\tau(x_a) = \{1\}, \qquad A_\tau(x_b) = \varnothing.$$

A policy that instead calibrates one common depth calls no model at depth zero and at least one model at every positive depth, so it cannot reproduce both choices. Thus matching a model order under independence does not establish that the sets of a common-depth policy always contain, or are always contained in, those of a query-dependent threshold-and-null policy. Whether one policy's sets contain the other's also depends on the score, the cutoff, the null action, the smoothing, and the tie behavior.

D.2 Population transfer of the ordering guarantee

Theorem 1 is a statement about the realized design matrix, and this subsection records when it transfers to the deployment distribution and why the transferred bound is uninformative at large pools. Write $J_P(\pi) = \mathbb{E}[T_X(\pi)]$ for the population average position of the first correct model under ordering π, with the same truncation convention as J_D.

Proposition 4 (Transfer with an additive sampling term)**.** *Suppose the n_d design queries are drawn independently from the deployment distribution, and let $\delta \in (0,1)$. With probability at least $1-\delta$ over the design draw, the tally order π^T satisfies*

$$J_P(\pi^T) \le 4\min_\pi J_P(\pi) + \varepsilon_u + 4\varepsilon_p, \qquad \varepsilon_u = m\sqrt{\tfrac{m\log m + \log(4/\delta)}{2n_d}}, \quad \varepsilon_p = m\sqrt{\tfrac{\log(4/\delta)}{2n_d}}.$$

Proof. For a fixed ordering π, $J_D(\pi)$ is an average of n_d independent terms $T_i(\pi) \in [1,m]$, so Hoeffding's inequality gives $|J_D(\pi) - J_P(\pi)| \le \varepsilon_p$ with probability at least $1-\delta/2$ for one fixed ordering, and $|J_D(\pi) - J_P(\pi)| \le \varepsilon_u$ simultaneously for all $m!$ orderings with probability at least $1-\delta/2$, using $\log m! \le m\log m$. On the intersection of the two events, with $\pi^\star$ a population minimizer,

$$J_P(\pi^T) \le J_D(\pi^T) + \varepsilon_u \le 4\min_\pi J_D(\pi) + \varepsilon_u \le 4J_D(\pi^\star) + \varepsilon_u \le 4J_P(\pi^\star) + 4\varepsilon_p + \varepsilon_u,$$

where the second step is Theorem 1. The uniform bound is needed for π^T because the tally order is a function of the design draw. □

Two features limit this statement. First, the additive term decides whether the bound is informative. At the eleven-model pools, whose design splits contain between about eight thousand and eighty thousand queries, $\varepsilon_u + 4\varepsilon_p$ stays below one and a half calls at $\delta = 0.05$, so the transferred factor-four bound is informative there. At one hundred models the term already reaches tens of calls, and at five hundred models it is comparable to the pool size and far above the certified call counts, so the bound is vacuous where the pools are largest, and the redraw protocol of Section 5 measures design-to-test generalization directly in that regime. Second, even where the bound transfers, it controls the mean of the first-success depth, while calibration selects, in the depth-indexed view, a quantile of

that depth near the target level. A mean-competitive ordering can still cross the target level late, so no multiplicative guarantee on certified calls at a fixed target follows, beyond Markov-type bounds whose factor inflates by $1/\alpha$. The number of calls that CopRoute makes on test queries is therefore an empirical finding in this paper, and Theorem 1 does not imply it; the guarantee that holds at test time is the certificate of Theorem 2.

D.3 Structured-estimator theory

A parametric dependence model should describe association left after the marginal router has accounted for query difficulty, whereas raw pooled correlation can reflect a mixture of easy and hard queries and may count the same difficulty twice when combined with query-specific marginals.

The rest of this subsection concerns the parametric members of the family, namely the Gaussian and Student-t copula and latent-normal orders of Appendix F, and does not concern the tally default. One structured working model is conditional multivariate probit, in which a latent Gaussian vector with query-dependent correlation is thresholded at the levels that reproduce the router's marginals, namely

$$Z \mid X = x \sim \mathcal{N}(0, R_\theta(x)), \qquad Y_j = \mathbf{1}\{Z_j > t_j(x)\}, \qquad t_j(x) = \Phi^{-1}(1 - p_j(x)). \tag{19}$$

Under correct specification,

$$q_x(S) = \Phi_{R_{\theta,S}(x)}\big(t_S(x)\big). \tag{20}$$

This is a model-based formula and is not an identity for every Bernoulli law, and the Gaussian, Student t, vine, Ising, and other extensions are modeling choices.

Assumption 1 (Conditional residual dependence)**.** *The structured estimator is fitted on design data under the following conditions.*

1. *Each marginal $p_j(x)$ used to fit dependence is estimated out of fold.*
2. *The dependence estimator targets the residual law after accounting for the marginal probabilities and declared router features.*
3. *Using a shared matrix R assumes that residual dependence is constant across queries, and clustered or feature-dependent versions are learned on design data only.*
4. *Marginal probabilities are clipped away from zero and one, and fitted correlation matrices are positive definite with eigenvalues bounded away from zero.*
5. *The dependence family, numerical integration rule, and tuning choices are selected without final-calibration labels.*

Theorem 5 (Structured plug-in consistency for fixed m)**.** *Keep m fixed while the number of design queries n_d grows. Assume the design observations are iid, Assumption 1 holds, and for a fresh query X,*

$$\max_{1 \le j \le m} |\widehat{p}_j(X) - p_j^\dagger(X)| \xrightarrow{p} 0, \qquad \widehat{\theta} \xrightarrow{p} \theta^\dagger. \tag{21}$$

Also assume a compact parameter space, correlation eigenvalues bounded away from zero, finite thresholds, continuity of the orthant map and any correlation projection at the limit, and numerical integration error converging uniformly to zero over subsets. Let $q_X^\dagger(S)$ be the all-miss probability under the limiting structured model. Then

$$\max_{S \subseteq \mathcal{M}} |\widehat{q}_X(S) - q_X^\dagger(S)| \xrightarrow{p} 0. \tag{22}$$

Moreover,

$$\max_{S \subseteq \mathcal{M}} |\widehat{q}_X(S) - q_X(S)| \le \max_{S \subseteq \mathcal{M}} |\widehat{q}_X(S) - q_X^\dagger(S)| + b_m(X), \tag{23}$$

where

$$b_m(X) = \max_{S \subseteq \mathcal{M}} |q_X^\dagger(S) - q_X(S)| \tag{24}$$

is the misspecification error. Under correct specification, $b_m(X) = 0$.

Proof. Fix a model set S. Marginal consistency, parameter consistency, continuity, and vanishing numerical error imply

$$\widehat{q}_X(S) - q_X^\dagger(S) \xrightarrow{p} 0$$

by the continuous mapping theorem. Because m is fixed, there are only 2^m subsets. A union bound over this finite collection gives the uniform convergence in (22). Equation (23) is the triangle inequality. Correct specification identifies $q_X^\dagger$ with the true all-miss law and makes $b_m(X) = 0$. □

This theorem concerns the parametric estimators only, and it states that a correctly specified parametric model of the joint law, fitted consistently, recovers the true all-miss probabilities of every set in the limit. It says nothing about the tally order, which fits no parametric model; it gives no sample-complexity advantage of parametric over empirical estimators, and it gives no guarantee when the parametric model is misspecified. Decision convergence additionally requires a unique limiting winner separated by a positive margin, or a vanishing probability of near ties.

Scope summary. The appendix establishes the formal joint target, the independence benchmark, proofs of the fixed-tally and calibration guarantees, oracle lower bounds, and one structured-model consistency result. It does not establish one universally optimal nested chain, strict containment of the method of Hao et al. (2026), population transfer of the empirical factor-four constant, optimality across different frozen families, conditional risk for every query, final-answer correctness, or a structured small-sample advantage.

E Implementation details

Frozen data and marginal router. The design correctness matrix is $D \in \{0,1\}^{n_d \times m}$, with design marginal $\bar{p}_j^D = n_d^{-1} \sum_i D_{ij}$. The design split is disjoint from router training, validation, final calibration, and test. The frozen prompt representations are the precomputed 768-dimensional mDeBERTa embeddings distributed with the benchmark releases, deduplicated and row-wise L2 normalized. No encoder is trained or loaded in our implementation, and nearest neighbors use cosine similarity. For query x, let $N(x)$ be its 40 nearest router-training queries under this embedding, and let Z_{rj} indicate whether model j answers neighbor r correctly. The marginal router uses $\widehat{p}_j(x) = |N(x)|^{-1} \sum_{r \in N(x)} Z_{rj}$, clipped to $[10^{-6}, 1 - 10^{-6}]$. No design, validation, calibration, or test label enters $N(x)$.

Fixed-tally construction. The main order π^T follows (4). An exact tally tie is resolved by the larger $\bar{p}_j^D$ and then by the smaller model index. If the current prefix covers every design query, the remaining models are appended by descending $\bar{p}_j^D$ and then by model index. This produces one complete order per fitted benchmark and model pool.

Independence-based stopping. For any order π, the independence residual is $r_k^{\mathrm{ind}}(x;\pi) = \prod_{\ell=1}^{k}(1 - \widehat{p}_{\pi_{x,\ell}}(x))$, with $r_0^{\mathrm{ind}} = 1$. The model at position k receives the predecessor-coverage score $1 - r_{k-1}^{\mathrm{ind}}$. We use conditional independence only to estimate prefix failure, and final calibration does not assume that this estimate equals the true all-miss probability.

Dependence-based stopping. For the fixed-tally order, let $a_k = \widehat{q}_D(P_k^{\pi^T})$ be the observed design all-miss fraction and $b_k = \prod_{\ell=1}^{k}(1 - \bar{p}^D_{\pi_\ell^T})$ its independence prediction. Their ratio $d_k = a_k / b_k$ is the design-set dependence ratio. We set $d_0 = 1$, and we set $d_k = 1$ whenever $b_k = 0$, which occurs only if some model in the prefix answers every design query, and then $a_k = 0$ as well, so the independence product already returns zero. Let $c_k = n_d a_k$ be the number of design queries missed by the entire prefix. For $h > 0$, define

$$w_k(h) = \min\{1, c_k/h\}, \qquad \widetilde{d}_k(h) = 1 + w_k(h)(d_k - 1). \tag{25}$$

Prefixes supported by at least h surviving design queries use the full ratio; prefixes with fewer supporting queries move the ratio toward one. Validation considers $h \in \{0, 5, 20, 50\}$. The value $h = 0$ is defined by $\widetilde{d}_k(0) = 1$ at every depth and therefore recovers independence-based stopping exactly. For each query,

$$r_k^{\mathrm{dep}}(x;h) = \min_{0 \le t \le k} \mathrm{clip}_{[0,1]}\left\{ r_t^{\mathrm{ind}}(x;\pi^T)\widetilde{d}_t(h) \right\}. \tag{26}$$

The cumulative minimum keeps the residual from increasing with depth, so thresholding the entry scores always selects an exact prefix. The design-set dependence ratio is an empirical correction along this order and does not estimate the full correctness copula parametrically.

Query-specific and local constructions. An order or selected depth is query-specific when it changes with x, and a dependence estimate is local when it is fitted from $N(x)$. Every local method is query-specific, while pairwise and correlation methods can be query-specific even when they combine $\widehat{p}_j(x)$ with one dependence table fitted on the global design split. The two main policies use a fixed order and a query-specific selected depth.

For a query-specific order π_x, local empirical stopping counts joint misses among the 40 neighbors as

$$r_k^{\mathrm{loc}}(x;\pi) = \frac{1}{40} \sum_{r \in N(x)} \prod_{\ell=1}^{k} (1 - Z_{r,\pi_{x,\ell}}), \qquad r_0^{\mathrm{loc}} = 1. \tag{27}$$

The fully local policy greedily appends the candidate that minimizes this residual and also uses it for stopping. A matched variant keeps the same local order but uses r_k^{ind} for stopping. Residuals within 10^{-12} of one another are treated as ties, which are resolved by larger local marginal correctness and then by smaller model index.

Ties and numerical audit. Before calibration, every query-level score receives the same frozen perturbation used by RACER, namely independent noise drawn uniformly from $[0, 10^{-6}]$, keyed by the experiment and split seeds and added to the raw predecessor-coverage score of each model. The perturbation resolves equal or nearly equal cutoff scores so that the sublevel sets form a nested family in the threshold, and it changes the deployed order only among models whose raw scores are within 10^{-6} of one another. Such near-ties arise when the running product has already reached zero, so that every remaining model in the fixed order carries a score of one. On such a plateau the independent noise can admit a model at a later position ahead of an earlier one, and the certified set is then not a prefix of the fixed order at that depth. Because the plateau begins only after the estimated all-miss probability is zero, calibration treats every model on it as interchangeable, and the guarantee of Theorem 2 covers the perturbed score map as it was run. We record for each cell the depth at which the plateau begins, the fraction of test queries whose calibrated set reaches it, and the corresponding underflow and saturation flags. A prefix-preserving alternative, which adds the offset $(k-1)10^{-6}/m$ at position k of the stably sorted scores, is implemented but was not used for the reported runs. For independence-based stopping, we retain $\log r_k^{\mathrm{ind}} = \sum_{\ell=1}^{k} \log(1 - \widehat{p}_{\pi_{x,\ell}}(x))$ beside the direct product.

Learned baseline reconstructions. FlexRouter uses a trainable determinantal point process with a 128-dimensional query projection, trainable model embeddings, a cosine-similarity kernel, a coverage objective with auxiliary binary cross-entropy, and greedy determinant-gain ordering (Wang et al., 2026). Because its exact encoder checkpoint and pooling implementation are not released with executable code, our reconstruction uses the same audited frozen 768-dimensional prompt embeddings that the other learned baselines receive and sets the unreported pre-screen size to $K_0 = m$.

BaRP uses its preference-conditioned categorical policy and paper-specified REINFORCE training with chosen-action feedback, a mini-batch reward baseline, entropy regularization, and uniformly sampled quality-cost preferences (Wang et al., 2025b). Its unreported preference width, decision-head width, and cost cap are frozen at 32, 256, and the 95th percentile of finite positive training-family call costs, respectively. For common set evaluation, its quality-only logits induce a stable top-k order. Both learned baselines use five training seeds, and in the common set-routing evaluation their orders extend over the full model pool.

Computational cost. All policies read the same router scores, so the shared per-query cost is one embedding lookup and an average over the 40 stored neighbors, and the table below lists what each policy adds. Fitting the tally order is a counting computation on the design matrix and involves no gradient training.

policy	fitting, offline	added per-query cost	tuned
CopRoute-Ind	greedy tally on the design matrix, $O(n_d m)$ per selected position	running product along the fixed order, $O(m)$	none
CopRoute-Dep	tally plus prefix co-failure counts, $O(n_d m)$	product times stored ratios and a running minimum, $O(m)$	h over four values
RACER-G / RACER-P	none beyond the router	per-model score transform, $O(m)$	none
Exact independence	none	per-query sort of the marginals, $O(m \log m)$	none
FlexRouter	gradient training of the query projection and model embeddings, per pool and seed	encoder projection and greedy determinant-gain selection over the pool	training settings
BaRP	policy-gradient training, per pool and seed	network forward pass and top-k sort, $O(m \log m)$	training settings

The proposed policies fit a single greedy pass over the design matrix. This is more than RACER fits, since RACER fits nothing beyond the router, and less than the learned baselines fit, since they require gradient training for every pool and seed before any routing can occur. At deployment, every single-router policy adds one linear pass over the ordered pool per query.

Selection before final calibration. Each repetition fits the router, order, and dependence objects on their designated training splits. We tune each hyperparameter on the validation split and select a family there when required. We then freeze the score map, real-model sets, tie rules, seeds, and numerical draws. Held-out final-calibration labels select only the global cutoff through (3), after which the test split is evaluated once. Appendix G gives the split proportions and feasibility rule.

F Estimator definitions for the CopRoute family

The query-level router provides $\widehat{p}_j(x)$, and the design matrix provides the failure indicators $F_{ij} = 1 - D_{ij}$. Every fixed order in the family is produced by the same greedy step, which repeatedly adds the model that is wrong least often on the design queries where every already-chosen model is wrong. The estimators differ only in the failure matrix fed to that step, either the observed design failures or synthetic failures drawn from a joint model fitted to them. The names below match Tables 2, 8, 9, and 20.

Tally (the default). The tally order applies the greedy step directly to the observed design failure matrix and is the fixed tally order of Section 3.

Double fault. This pairwise variant of the same idea adds the model whose joint-failure rate with the already-chosen set is smallest, using a weight that trades this diversity term against individual accuracy.

Pairwise count. This order estimates pairwise joint failures with additive smoothing, averages the implied conditional-miss surrogates over the chosen set, and multiplies the running residual by that average. It is a bounded ranking heuristic rather than a coherent joint probability, and it appears in the synthetic mechanism map (Table 20).

Error correlation. This order adds the model whose failures are least Pearson-correlated with the chosen set, with the correlation magnitude clipped so that a near-perfect estimate from a small design split is not taken as exact.

Rank correlation. This order uses the latent-threshold construction below, with the correlation estimated from failure concordance through the sine transform used for rank correlations, so that the estimate is unaffected by miscalibration of the marginals.

Latent threshold models (Gaussian copula, Student-t copula, low-rank factor copula). Each model fails when a latent variable falls below a threshold matched to its design failure rate. Dependence is carried by a low-rank factor approximation of the failure correlation, computed from an economy SVD of the standardized failures, since a full correlation matrix over 3,811 models can neither be formed nor supported by a few hundred design labels. The Gaussian variant draws the latent variables as normals (the mechanism map calls this construction latent normal), the Student-t variant gives them heavier tails, and the low-rank factor variant fixes the rank. Synthetic failures drawn from the fitted model replace the observed matrix in the greedy step.

Bernoulli mixture. This order fits, by expectation-maximization, a small number of latent query classes, each with its own per-model failure rates, and synthetic failures drawn from the fitted mixture feed the greedy step.

Shrinkage to independence and shrinkage to one factor. Blend the observed failure matrix with synthetic failures from a target model, independence at the observed marginals or a low-rank factor model, at a tuned weight, and apply the greedy step to the blend.

Controls. Model-family lineage is a zero-parameter control that picks the best model from each name-prefix family in round robin, so that the name prefix is a proxy for shared failure structure in a pool built from fine-tunes of a few dozen base models. Accuracy-weighted random samples the order without replacement with design-accuracy weights, and the random control uses a seeded uniform order and no model information.

Matched non-family controls. The exact-independence control orders models by descending router score and stops with the independence product $\widehat{q}_x^{\mathrm{ind}}(S) = \prod_{j \in S}(1 - \widehat{p}_j(x))$; it fits no dependence statistic. The fixed design-marginal control orders by decreasing design accuracy $\bar{p}_j^D$ with the same stopping rule, which isolates the value of co-failure information in the order. Query-local empirical coverage uses the 40 frozen router-training neighbors to define a local all-miss frequency and greedily minimizes it per query. FlexRouter and BaRP are reconstructed as described in Appendix E, and their calibration uses the same final-calibration rule as every other policy.

F.1 Summary of the family

Twelve alternative orders and the tally default each map the design matrix to one pool ordering, and the hyperparameter grids contain three to five values per estimator.

family	order	failure matrix fed to the greedy step
counting	tally (default)	observed design failures
counting	double fault	observed failures, pairwise joint-failure objective
counting	pairwise count	smoothed pairwise surrogate (ranking heuristic)
correlation	error correlation	observed failures, correlation-penalized objective
correlation	rank correlation	latent draws, concordance-based correlation
latent threshold	Gaussian copula	latent-normal draws, low-rank correlation
latent threshold	Student-t copula	heavy-tailed latent draws
latent threshold	low-rank factor copula	latent draws at fixed rank
mixture	Bernoulli mixture	draws from fitted latent query classes
shrinkage	to independence	observed failures blended with independence draws
shrinkage	to one factor	observed failures blended with factor-model draws
control	lineage, accuracy-weighted random, random	no fitted dependence

G Protocol details and claim boundaries

Splits and selection order. Query-level cells use 20 redraws with design 30 percent, router training 20 percent, final calibration 20 percent, validation 15 percent, and test 15 percent. The model pool is resampled per redraw. Validation selects estimator hyperparameters or a unified candidate using provisional operating points. The provisional cutoff is then discarded, the complete winning map is frozen, and final calibration selects only its threshold, after which the test split is evaluated once.

Standard loss and certification. The primary loss is the standard all-miss loss over real-model sets, under which an empty set has loss one. Calibration uses the finite-sample bound $(c+1)/(n+1) \le \alpha$. If no threshold passes, the draw is `NON-CERTIFIED`; its calls and risk are not pooled with certified draws, so every efficiency table reports the certification rate. Structural infeasibility of the complete model pool and finite-calibration non-certification are reported separately.

The per-model construction and diagnostics of Hao et al. (2026). Hao et al. (2026) train a base router outside the final-calibration sample; for a query x it estimates each model's correctness probability $\widehat{p}_j(x)$ and scores model j either by the inverse probability $1 - \widehat{p}_j(x)$ (RACER-P) or by the gap between its score and that of the highest-ranked model (RACER-G). One calibrated cutoff on these per-model scores yields a query-dependent nested family whose model ranking varies with x while the cutoff is shared across queries. Substituting the per-model value $s(x, j) = 1 - \widehat{p}_j(x)$ for one minus the cumulative-coverage residual $r_{k-1}(x)$ in the construction of Section 2 gives $C_\lambda(x) = \{j : 1 - \widehat{p}_j(x) \le \lambda\}$, which is exactly the RACER-P family over the real model pool.

RACER-P is thus the special case of our construction whose residual ignores the models already selected, whereas CopRoute lets $r_{k-1}(x)$ depend on the current prefix so that a candidate's score reflects the coverage already obtained. Hao et al. (2026) additionally augment the pool with a virtual null model $m_\emptyset$, writing $\mathcal{M}' = \mathcal{M} \cup \{m_\emptyset\}$ and placing $m_\emptyset$ in the augmented ground-truth set whenever no real model is correct, so that selecting only $m_\emptyset$ is read as abstention. Because this null action always covers the augmented target, their calibration always admits a feasible member above the finite-sample floor, and infeasibility is absorbed into abstention rather than reported. In our comparisons RACER-G and RACER-P use the same splits, the same final-calibration split, and the same standard all-miss loss as the proposed policies. We certify the stricter real-model loss with no always-safe action, so that an empty real-model set incurs loss one, and a draw whose full pool cannot meet the target returns `NON-CERTIFIED`. The augmented-null loss of Hao et al. (2026) is retained only as a separately named diagnostic, and its null action is not interpreted as human review, zero cost, or guaranteed safety.

Realized-risk classification. A certified cell is marked above target only when its mean standard all-miss risk exceeds α by more than two standard errors. This classification summarizes the finite test sample and does not restate the theorem.

Synthetic conditional-independence diagnostic. A separate synthetic experiment generates heterogeneous query difficulty with model correctness independent conditional on that difficulty; the fully specified five-regime successor of this diagnostic is the mechanism map of Appendix L. Each of 128 seeds draws a scalar difficulty $x \sim \mathcal{N}(0,1)$ per query and eight models whose conditional correctness probabilities are $\sigma(\beta_j - \gamma_j x + \kappa_j(x^2-1))$ clipped to $[0.03, 0.97]$, with intercepts β from 1.55 down to 0.05, loadings γ between 0.64 and 1.48, and curvatures κ between -0.12 and 0.12; correctness is drawn independently across models given x. Splits of 1,500 design, 1,000 validation, 1,500 final-calibration, and 4,000 test queries are used at $\alpha = 0.10$, with a correlation-penalty grid $\{0.25, 0.5, 1, 2\}$ selected on validation. Pooled off-diagonal failure correlation is 0.152 with 95 percent interval $[0.151, 0.153]$, while residual correlation, computed from the residuals $Y - p(x)$, is 0.0008 with interval $[0.0001, 0.0016]$. Exact factorization, pooled correlation, and residual correlation each use 2.0 calls, with risks 0.0656, 0.0657, and 0.0657. Because the three estimators use the same 2.0 calls at risks within 0.0001 of one another while pooled correlation is 0.152, pooled association alone does not indicate exploitable residual dependence.

H Detailed calibrated results

This section completes the main comparison of Table 1. Table 5 reports the RouterEval benchmarks at the 200-, 300-, and 500-model pools, including the CopRoute-Dep ablation. Table 6 documents the excluded benchmark, MATH-intermediate-algebra, which certifies for no policy at any pool at the main targets and only occasionally at looser targets, so no policy comparison is possible there. The reason is the feasibility condition of Section 5. Even calling every model in the pool misses 40.4, 34.6, 25.4, and 22.1 percent of the calibration queries at pools 100, 200, 300, and 500, so the condition $(c_M + 1)/(n + 1) \leq \alpha$ fails on every redraw at the main targets and holds on some redraws only where the pool is large enough and the target loose enough for the full-pool miss rate to fall below the target, at pool 500 for $\alpha \geq 0.15$ and at pool 300 for $\alpha = 0.20$. At the tightest targets on the hardest pool-100 benchmarks, certification rates also fall for every policy (at $\alpha = 0.02$, 3 of 20 redraws on IFEval, 1 on MATH-algebra, none on MuSR); on the redraws that do certify there, dependence-corrected stopping reduces calls on MATH-algebra from 55.1 to 39.2 and on IFEval from 78.4 to 61.4, although with one and three redraws no interval can be formed for these differences.

I Complete estimator grids for the CopRoute family

Table 8 expands the main-text family comparison (Table 2) to every pool. First, no alternative order improves on the tally order in aggregate under either stopping rule. The gap to the seeded random control (about 7 calls) bounds the value of any fitted order, but it does not isolate the co-failure information, because a random order also discards marginal quality; that contribution is measured by the design-marginal contrast of Table 7, and on GPQA and BBH-navigate the random order beats tally in some aggregates (Table 9), so the random gap is not uniformly positive. Second, the alter-

Table 5: Certified mean calls on the RouterEval benchmarks at the 200-, 300-, and 500-model pools, completing Table 1. A dagger marks cells whose certification rate is below 0.9; a dash means no redraw certifies.

		$\alpha = 0.05$						$\alpha = 0.10$					
dataset	pool	Ind	Dep	G	P	Flex.	BaRP	Ind	Dep	G	P	Flex.	BaRP
BBH-boolean	200	**2.34**	**2.34**	4.26	4.72	4.67	2.70	1.12	1.12	2.19	3.09	**1.11**	1.28
	300	2.45	2.45	4.61	5.69	**2.11**	3.03	1.27	1.27	2.16	3.33	**1.23**	1.34
	500	**2.79**	**2.79**	5.78	5.81	2.86	4.50	**1.40**	**1.40**	2.89	3.95	**1.40**	1.49
BBH-navigate	200	**5.02**	**5.02**	20.7	20.1	6.34	9.14	**2.39**	**2.39**	4.14	4.30	4.94	5.39
	300	6.82	6.82	11.7	11.0	**5.77**	8.94	**2.09**	**2.09**	5.22	5.86	2.45	6.09
	500	**5.72**	**5.72**	16.0	16.3	11.4	7.76	**2.00**	**2.00**	4.99	5.53	5.82	4.75
GPQA	200	**30.2**	**30.2**	38.4	38.4	43.7	41.0	**11.1**	**11.1**	15.7	15.7	21.3	20.7
	300	**23.8**	**23.8**	34.0	34.0	35.0	33.5	**12.8**	**12.8**	17.2	17.2	13.7	16.0
	500	30.6	30.6	29.4	29.4	**25.7**	32.9	14.7	14.7	17.5	17.5	**12.6**	15.5
IFEval	200	**13.7**	20.2	17.5	20.7	17.9	15.1	**3.87**	4.87	5.86	9.67	5.38	3.95
	300	**11.8**	16.3	21.5	25.6	12.1	13.3	**2.12**	2.23	2.96	7.31	4.31	4.97
	500	14.6	30.3	21.0	29.1	**11.6**	16.9	**2.50**	3.07	4.32	9.30	4.13	4.67
MATH-algebra	200	**27.5**	33.1	28.9	28.8	55.9	58.6	**6.98**	7.15	8.17	8.50	9.81	10.0
	300	**17.7**	25.4	19.8	20.1	53.3	61.1	**3.30**	**3.30**	3.73	3.64	7.89	6.49
	500	**22.8**	23.6	24.1	23.5	55.7	26.8	4.55	**4.53**	6.48	7.10	16.1	11.2
MuSR	200	**74.7**†	78.7†	124.1†	154.2†	145.4†	145.5†	**13.9**	17.9	57.5	88.2	59.1	71.6
	300	**88.7**	107.0	166.1	212.4	165.1	164.9	**7.58**	7.59	60.4	113.5	60.3	73.6
	500	**30.5**	48.7	190.9	251.5	132.7	135.1	**5.80**	11.4	73.3	152.7	30.0	48.4

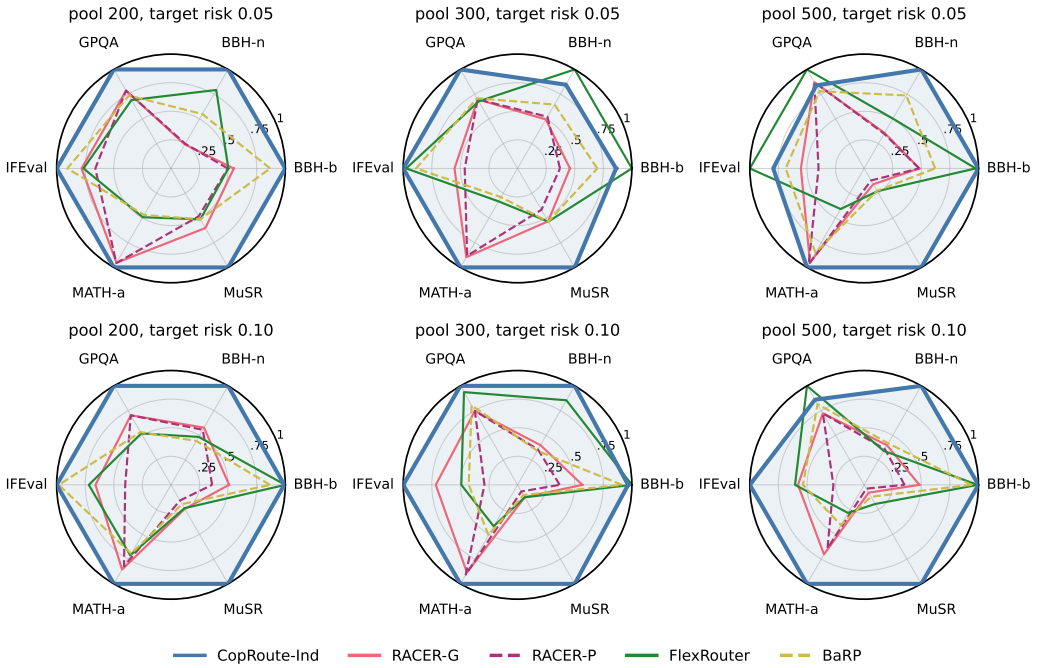


Figure 4: Relative call efficiency at the 200-, 300-, and 500-model pools of the six RouterEval datasets, one panel per pool size and target risk. Each value is the best certified calls on the axis divided by the policy's certified calls, so the per-axis winner lies on the rim at 1; FlexRouter and BaRP are averaged over five training seeds. CopRoute is on the rim on 29 of the 36 axes and FlexRouter on the remaining seven. The 500-model panels are repeated in Figure 3.

natives that model dependence with more structure, the Bernoulli mixture and the Student-t copula, beat tally at the 100-model pool and lose at larger pools. With 100 models the design set supports a richer estimate, whereas at 500 models the same design budget yields far fewer observations per prefix and the simple tally count is the more reliable statistic. Third, the two orders built from correlation matrices (error correlation, rank correlation) win individual cells but lose by tens of calls

Table 6: MATH-intermediate-algebra certification rates for CopRoute-Ind (identical for every policy in each cell). No policy certifies at the targets of Table 1, so the dataset is reported here rather than in the main comparison.

pool	$\alpha = .02$	.03	.05	.075	.10	.15	.20
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00
200	0.00	0.00	0.00	0.00	0.00	0.00	0.00
300	0.00	0.00	0.00	0.00	0.00	0.00	0.15
500	0.00	0.00	0.00	0.00	0.00	0.05	0.30

Table 7: Ablation of the two components on the cells of Table 1. Dep − Ind keeps the tally order and replaces the independence-based stopping rule by the dependence-corrected rule of CopRoute-Dep; marginal − Ind keeps the stopping rule and replaces the tally order by the design-marginal order, which ranks models by design accuracy alone. Entries are paired differences in certified mean calls with bootstrap 95 percent intervals over redraws; bold intervals exclude zero, and a superscript gives the number of jointly certified redraws when it is below 20.

	$\alpha = 0.05$		$\alpha = 0.10$	
dataset	Dep − Ind	marginal − Ind	Dep − Ind	marginal − Ind
ARC	0.00 (tie)	+0.12 [-0.06, +0.42]	0.00 (tie)	0.00 (tie)
HellaSwag	0.00 (tie)	**+0.61 [+0.49, +0.73]**	0.00 (tie)	+0.02 [0.00, +0.04]
MMLU	0.00 (tie)	**+1.69 [+1.41, +1.93]**	0.00 (tie)	**+0.78 [+0.65, +0.89]**
WinoGrande	+0.01 [0.00, +0.02]	**+2.57 [+2.29, +2.85]**	0.00 [-0.01, +0.02]	**+0.95 [+0.70, +1.22]**
BBH-boolean	0.00 (tie)	+0.16 [-0.30, +0.77]	0.00 (tie)	+0.13 [0.00, +0.28]
BBH-navigate	0.00 (tie)	**+21.40 [+13.05, +28.19]**	-0.01 [-0.02, 0.00]	**+5.32 [+3.50, +7.47]**
GPQA	0.00 (tie)	**+4.40 [+1.15, +8.05]**	0.00 (tie)	**+4.20 [+2.40, +6.05]**
IFEval	**+5.61 [+1.76, +10.19]**[19]	**+9.38 [+3.10, +17.00]**[19]	+0.04 [0.00, +0.11]	0.00 [-0.49, +0.47]
MATH-algebra	+0.84 [0.00, +2.52][3]	-0.99 [-2.00, 0.00][3]	-1.67 [-5.80, +0.69]	-0.39 [-1.09, +0.36]
MuSR	0.00 (tie)[8]	**+19.39 [+7.46, +32.41]**[8]	+1.19 [0.00, +3.58]	**+27.25 [+21.31, +33.27]**

elsewhere, so their aggregate differences against tally are positive, and an order that wins only occasionally cannot replace the tally order by default. The same tables report the dependence-corrected stopping rule for every order.

Dependence-corrected stopping. Table 7 isolates the two components of the policy on the cells of Table 1. The first contrast keeps the tally order and replaces the independence-based stopping rule by the dependence-corrected rule of CopRoute-Dep. Validation selects the uncorrected rule ($h = 0$) on most redraws, in 10 to 19 of 20 at every pool on IFEval and MuSR, the two benchmarks for which the selected values were recorded, so the two policies coincide on 12 of the 20 cells and differ by less than 0.05 calls on four more. Where they differ, the correction lowers calls on MATH-algebra at $\alpha = 0.10$ by 1.7 with an interval that includes zero, and raises them on IFEval at $\alpha = 0.05$ by 5.6 calls and on MuSR at $\alpha = 0.10$ by 1.2. Across all 187 certified cells of the four pools, the two rules tie on 104, the correction lowers calls on 21 and raises them on 62, and the six differences whose intervals exclude zero in its favor are at most 0.03 calls or rest on a single redraw, whereas the six against it reach 17 calls (Table 8). The dependence ratio is estimated from the design queries that miss an entire prefix, which become rare at exactly the depths where the correction would matter, and the resulting noise costs more than the correction recovers, so CopRoute-Ind is the default. The second contrast keeps the stopping rule and replaces the tally order by the design-marginal order. Replacing the tally order costs up to 21 calls on BBH-navigate and 27 on MuSR, and its interval excludes zero against the marginal order in 13 of the 20 cells and in none in its favor, so the gain of CopRoute over the baselines comes from the co-failure order rather than from the stopping correction.

Table 9 gives the same comparison per dataset. The alternatives' wins concentrate on GPQA and BBH-navigate, where even the random control improves on tally, and more weakly on MATH-algebra; tally is strongest or tied everywhere else. The error-correlation order shows the heavy-tailed pattern behind its aggregate loss, gaining 20 calls on MuSR and 11 on GPQA but losing 97 on MATH-algebra.

Table 8: Complete order-by-pool grid of extra certified calls relative to the tally order under the same stopping rule, weighted by certified redraws over the ten datasets and seven targets; negative favors the alternative.

	independence-based stopping						dependence-corrected stopping					
order	all	p11	p100	p200	p300	p500	all	p11	p100	p200	p300	p500
Bernoulli mixture	+0.42	+0.10	-0.65	-0.32	+0.67	+2.01	+0.36	+0.08	-0.39	+0.49	+0.47	+0.94
Student-t copula	+1.77	+0.42	-0.68	-0.35	+3.94	+4.63	+1.35	+0.39	-0.67	-0.48	+3.25	+3.55
Gaussian copula	+1.97	+0.54	-0.09	+1.08	+3.35	+4.13	+1.81	+0.55	-0.10	+1.59	+2.97	+3.33
rank correlation	+2.04	+0.54	+0.26	+0.71	+3.10	+4.76	+1.83	+0.54	+0.20	+1.31	+2.96	+3.44
double fault	+1.59	+0.03	+0.13	+0.38	+3.22	+3.40	+1.13	+0.02	+0.00	+0.04	+2.79	+2.21
low-rank factor copula	+2.44	+0.22	+0.17	+0.64	+3.46	+6.51	+2.26	+0.22	+0.17	+1.21	+3.61	+5.06
shrinkage to one factor	+2.94	+0.14	+0.39	+1.65	+4.54	+6.60	+2.70	+0.13	+0.52	+1.85	+4.24	+5.52
shrinkage to independence	+2.97	+0.09	+1.20	+1.44	+5.48	+5.33	+3.10	+0.09	+1.55	+1.78	+5.70	+5.06
error correlation	+6.25	+0.38	+0.98	+7.60	+5.27	+14.32	+5.50	+0.40	+1.23	+6.91	+4.11	+12.53
model-family lineage	+4.83	+0.55	+5.84	+6.05	+2.60	+7.76	+4.24	+0.56	+5.70	+6.32	+1.54	+6.07
accuracy-weighted random	+4.28	+1.41	+2.54	+2.97	+4.05	+9.07	+4.04	+1.40	+2.58	+2.90	+3.88	+8.20
random control	+7.78	+1.87	+4.19	+7.53	+9.08	+13.65	+7.24	+1.86	+4.00	+7.16	+8.33	+12.51

Table 9: Dataset-wise extra certified calls relative to the tally order under the same stopping rule, weighted by certified redraws over each dataset's pools and the seven targets; negative (bold) favors the alternative. RouterBench columns use the 11-model pool and RouterEval columns aggregate pools 100 to 500.

order	ARC	HSwag	MMLU	WinoG	BBH-b	BBH-n	GPQA	IFEval	MATH-a	MuSR
independence-based stopping										
Bernoulli mixture	+0.16	+0.15	+0.09	+0.00	**-0.04**	**-2.51**	**-0.40**	+4.70	**-0.16**	+1.72
Student-t copula	+0.11	+0.32	+0.37	+0.82	**-0.46**	**-1.78**	**-1.26**	+4.94	**-0.21**	+12.65
Gaussian copula	+0.14	+0.34	+0.47	+1.18	**-0.03**	**-1.17**	**-0.79**	+5.26	**-0.28**	+11.83
rank correlation	+0.15	+0.33	+0.45	+1.16	**-0.03**	**-1.59**	**-1.09**	+5.52	**-0.23**	+13.01
double fault	+0.08	+0.05	**-0.01**	+0.01	+1.06	+3.67	**-7.31**	+5.71	+2.61	+5.57
low-rank factor copula	+0.17	+0.21	+0.42	+0.12	+0.21	**-1.83**	+0.50	+5.43	**-0.27**	+14.96
shrinkage to one factor	+0.16	+0.15	+0.11	+0.15	+0.21	+0.20	+1.52	+4.54	**-0.10**	+16.19
shrinkage to independence	+0.08	+0.14	+0.10	+0.04	+0.18	+0.07	+1.64	+5.15	**-0.26**	+16.00
error correlation	**-0.06**	+0.83	+0.93	+0.02	**-1.02**	**-4.34**	**-10.89**	**-5.12**	+97.05	**-20.40**
model-family lineage	**-0.06**	+0.41	+0.79	+1.11	**-0.15**	+9.51	+0.47	**-0.25**	+2.04	+24.01
accuracy-weighted random	+0.86	+1.74	+1.24	+1.76	+0.44	**-1.33**	**-4.84**	**-3.22**	+11.67	+31.97
random control	+1.40	+2.19	+2.13	+1.85	+0.18	**-2.18**	**-6.72**	+0.77	+43.48	+27.21
dependence-corrected stopping										
Bernoulli mixture	+0.10	+0.14	+0.09	+0.00	**-0.04**	**-2.51**	**-0.40**	+5.90	+0.31	**-0.71**
Student-t copula	+0.06	+0.33	+0.38	+0.77	**-0.46**	**-1.73**	**-1.26**	+4.58	**-0.60**	+9.96
Gaussian copula	+0.16	+0.34	+0.47	+1.16	**-0.02**	**-1.09**	**-0.79**	+6.55	**-0.70**	+9.29
rank correlation	+0.15	+0.33	+0.45	+1.16	**-0.04**	**-1.38**	**-1.09**	+6.88	**-0.80**	+9.92
double fault	+0.05	+0.04	**-0.01**	+0.01	+1.06	+3.46	**-7.31**	+4.41	+2.61	+3.66
low-rank factor copula	+0.19	+0.20	+0.42	+0.12	+0.22	**-1.67**	+0.50	+7.03	**-0.79**	+11.93
shrinkage to one factor	+0.14	+0.14	+0.11	+0.14	+0.20	+0.30	+1.52	+4.19	**-0.11**	+14.53
shrinkage to independence	+0.09	+0.13	+0.10	+0.04	+0.18	+0.27	+1.64	+7.91	**-0.42**	+13.67
error correlation	**-0.03**	+0.86	+0.97	+0.02	**-1.02**	**-4.34**	**-10.89**	**-6.33**	+95.63	**-23.72**
model-family lineage	**-0.08**	+0.43	+0.80	+1.13	**-0.13**	+9.98	+0.47	**-2.74**	+1.54	+22.03
accuracy-weighted random	+0.84	+1.75	+1.24	+1.74	+0.50	**-1.44**	**-4.84**	**-5.65**	+12.62	+32.05
random control	+1.41	+2.17	+2.14	+1.81	+0.19	**-2.27**	**-6.72**	**-1.29**	+43.55	+25.30

J TRANSFER ACROSS DATASETS FOR ALL POLICIES

Figure 5 repeats the main-text transfer analysis (Figure 2) for every policy on RouterEval, and Table 10 summarizes both suites. Source and target must share a model pool, so transfers run within RouterBench and within RouterEval only. Within RouterEval we exclude source-target pairs from the same benchmark family, which affects only BBH-boolean and BBH-navigate. An earlier variant that excluded only the target itself made transfer into a BBH task look successful because the source pool contained its sibling, which is near in-domain rather than out-of-domain transfer. The learned baselines used in this suite were likewise retrained with the target's whole family held out. The seventh RouterEval benchmark, MATH-intermediate-algebra, is absent because it certifies for no policy (Table 6). The premiums differ by construction, since CopRoute fits the most target-specific structure and so loses the most when that structure comes from the wrong dataset, while FlexRouter and BaRP change little because their calls are set mainly by their learned routers. The levels still favor CopRoute, because averaged over every directed out-of-domain pair, CopRoute-Ind has the

Table 10: Mean certified calls in domain (ID) and out of domain (OOD, averaged over every directed source-target pair in the suite), recalibrated on the target.

	RouterBench (pool 11)				RouterEval (pool 100)			
	$\alpha = 0.05$		$\alpha = 0.10$		$\alpha = 0.05$		$\alpha = 0.10$	
policy	ID	OOD	ID	OOD	ID	OOD	ID	OOD
CopRoute-Ind	3.65	5.29	1.90	2.69	23.44	33.26	6.60	17.23
CopRoute-Dep	3.65	5.30	1.90	2.69	24.52	35.28	6.53	17.96
RACER-G	4.81	5.44	2.89	2.91	31.91	34.81	13.73	18.50
RACER-P	5.11	5.86	3.35	3.36	32.20	34.62	15.59	18.46
FlexRouter	5.05	5.32	2.45	2.74	35.70	37.59	15.03	16.08
BaRP	5.22	5.52	2.57	2.89	33.37	35.07	16.76	17.87

Table 11: Cross-dataset transfer summary of the mean extra certified calls after out-of-domain fitting with target recalibration (fit premium), and the mean excess of realized risk over the target when a source-calibrated cutoff is reused (risk inflation).

	fit call premium		source-calibrated risk inflation	
policy	RouterBench	RouterEval	RouterBench	RouterEval
CopRoute-Ind	+0.89	+16.53	+0.010	+0.222
CopRoute-Dep	+0.90	+18.63	+0.010	+0.222
RACER-G	+0.13	+5.38	+0.016	+0.169
RACER-P	+0.06	-2.04	+0.015	+0.165
FlexRouter	+0.19	+0.09	+0.002	+0.140
BaRP	+0.18	+0.05	+0.002	+0.121

lowest mean calls on RouterBench at both targets and on RouterEval at $\alpha = 0.05$, and sits within 1.1 calls of FlexRouter at RouterEval $\alpha = 0.10$. Table 11 reports realized risk, where reusing a source-calibrated cutoff inflates it for every policy, on RouterEval by 0.12 to 0.22, so recalibration on the deployment distribution is required for every policy.

K Supplementary native learned-baseline metrics

The common comparison evaluates the learned baselines under the same calibrated standard-loss rule as every other policy, and this section reports each source method's native metrics only as supplementary characterization.

We omit GSM8K from the native BaRP training tasks because BaRP conditions on prompt features, and in the RouterBench release the GSM8K prompt field contains at most one distinct prompt, which makes the prompt-conditioned formulation trivial on that task. We therefore train the native run on MMLU, ARC-Challenge, and Winogrande, substituting no other task, and keep HellaSwag and MBPP as held-out target tasks.

L Cost, answer selection, and further analyses

Cost per query and end-to-end accuracy. The certified guarantee bounds the probability that every called model is wrong; a deployed system must additionally return one answer and pay a price per call. Table 15 reports both on RouterBench MMLU, with the answer taken from the called model with the highest router score and each call priced by the public per-model price sheet. Fewer certified calls lower the realized cost. At $\alpha = 0.10$, CopRoute-Ind answers 80.97 percent of queries at USD 1.90 per thousand, against 80.18 percent at USD 3.55 for RACER-G, a 46.6 percent reduction at the same certified risk. Final-answer accuracy depends on the answer selector and the called set, and the answer selector adds 2.3 microseconds per query. The strongest single model is cheaper than any calibrated policy but misses 18.6 percent of queries, far above both targets, so no certificate is attainable without routing.

Cost-weighted stopping. Model prices on RouterBench differ by two orders of magnitude, so the cheapest certified set is not always the smallest. The cost-weighted variant of Table 16 replaces the tally objective by newly covered design failures per unit of frozen mean price, so that it fits a

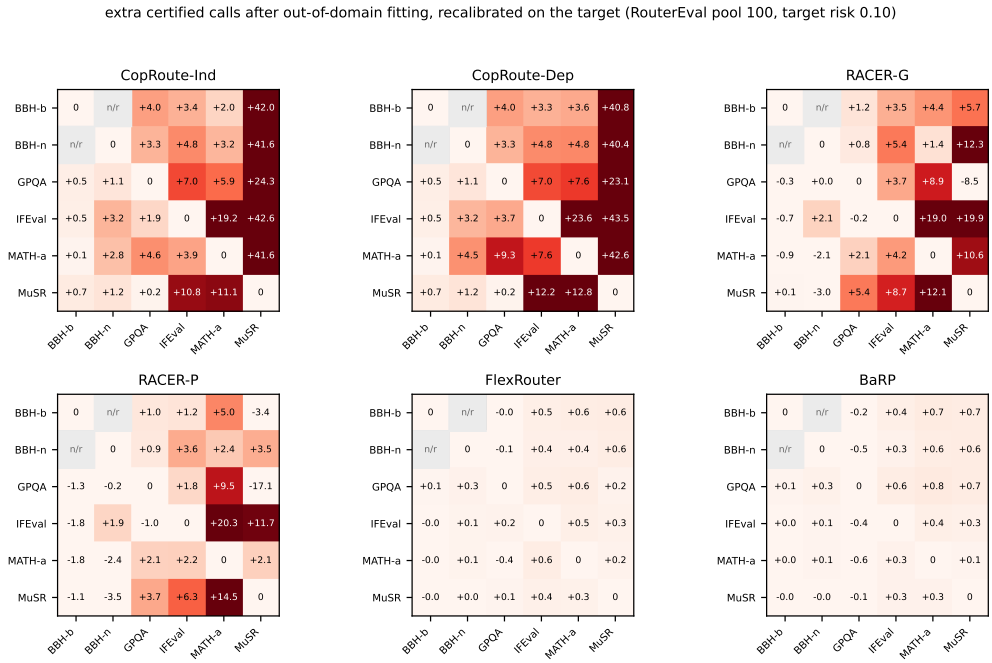


Figure 5: Extra certified calls after out-of-domain fitting for every policy, recalibrated on the target, RouterEval pool 100 at target risk 0.10. Cell values are exact; the color scale is shared and capped at 12. The two pairs within the BBH family were deliberately not run (n/r), so that no transfer is near in-domain.

Table 12: Native FlexRouter adaptive stopping on IFEVAL. Entries are mean calls / Success, averaged over five training seeds and 40 redraws per seed. No risk certificate is claimed for this native rule.

τ	pool 100	pool 200	pool 300	pool 500
0.0	10.00 / 0.934	10.00 / 0.935	10.00 / 0.952	10.00 / 0.960
0.01	10.00 / 0.934	10.00 / 0.935	10.00 / 0.952	10.00 / 0.960
0.05	10.00 / 0.934	10.00 / 0.935	10.00 / 0.952	10.00 / 0.960
0.1	10.00 / 0.934	10.00 / 0.935	10.00 / 0.952	10.00 / 0.960
0.2	10.00 / 0.934	10.00 / 0.935	10.00 / 0.952	9.88 / 0.958
0.3	9.99 / 0.934	9.99 / 0.935	9.76 / 0.951	8.70 / 0.947
0.5	5.83 / 0.882	6.43 / 0.910	4.41 / 0.905	3.09 / 0.872

different order (identical to the tally order only when all prices are equal), and then applies the same independence-based stopping rule and calibration; it is therefore a cost-aware member of the family whose certificate follows from Theorem 2 and whose ordering guarantee is the price-weighted analogue of Theorem 1 rather than that theorem itself. It lowers realized cost in every cell, with all bootstrap intervals excluding zero, at a price of 0.4 to 2.7 extra calls; at $\alpha = 0.10$ its certified cost is 2 to 11 times lower than count-optimized CopRoute-Ind. Under heterogeneous prices the deployment recommendation is therefore to calibrate the cost-weighted objective.

Cascade routing without response signals. Cascade routing decides whether to escalate after reading each model's response, using quality signals computed from that response. Our benchmarks record only per-query correctness labels; the responses, and therefore the quality signals, are unavailable, and using the correctness label in their place would give the cascade access to the ground truth it is meant to predict. We therefore evaluate the cascade with its escalation decision restricted to pre-response information, in which form it reduces to a router. Table 17 reports the consequence on the same splits as CopRoute-Ind, where the restricted cascade exceeds the risk target in seven of

Table 13: Native FlexRouter adaptive stopping on MUSR. Entries are mean calls / Success, averaged over five training seeds and 40 redraws per seed. No risk certificate is claimed for this native rule.

τ	pool 100	pool 200	pool 300	pool 500
0.0	10.00 / 0.796	10.00 / 0.767	10.00 / 0.783	10.00 / 0.859
0.01	10.00 / 0.796	10.00 / 0.767	10.00 / 0.783	10.00 / 0.859
0.05	10.00 / 0.796	10.00 / 0.767	10.00 / 0.783	10.00 / 0.859
0.1	10.00 / 0.796	10.00 / 0.767	10.00 / 0.783	10.00 / 0.859
0.2	10.00 / 0.796	10.00 / 0.767	10.00 / 0.783	9.80 / 0.857
0.3	9.99 / 0.796	10.00 / 0.767	9.66 / 0.782	8.90 / 0.851
0.5	5.31 / 0.721	5.63 / 0.696	4.30 / 0.684	3.89 / 0.767

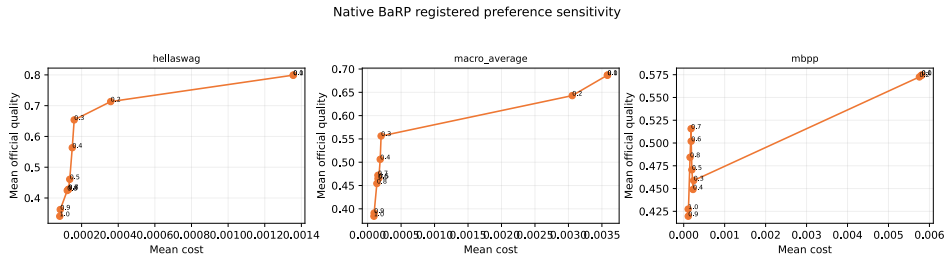


Figure 6: Native BaRP quality-cost sensitivity after training on the three nondegenerate RouterBench task families. These single-model source metrics are supplementary to the common set-routing evaluation.

eight cells, by up to 3.5 times the target on MATH-algebra at $\alpha = 0.10$ (0.350 against a matched 0.110), and provides no distribution-free certificate. Cascades optimize an expected cost-quality tradeoff, while CopRoute certifies a miss rate at minimum calls, so the two frameworks answer different questions.

Distance of the tally order from the best fixed order. Exhaustive dynamic programming over all 2^{11} subsets of the 11-model pool, on all 80 RouterBench redraws, gives the exact optimum of the design objective. The calibrated difference, exact minus tally, is -0.019 calls with bootstrap 95 percent interval $[-0.033, -0.004]$, with 5 cell-target wins, 16 ties, and 5 losses; a beam-search relaxation ($[-0.118, +0.017]$) and a local-swap refinement ($[-0.015, +0.035]$) both include zero and select the plain tally order in the large majority of redraws. The greedy order is within 0.02 calls of the exact optimum, which is consistent with the factor-four guarantee of Theorem 1 being loose on these instances.

Pool composition shift. Table 18 complements the dataset transfer of Appendix J by shifting the model pool itself on every cell of the main comparison. Removing one model from the pool invalidates frozen certificates almost entirely, since only 3.4 percent of them (25 of 729) still certify on the reduced pool, whereas refitting the order and recalibrating the cutoff on the reduced pool certifies 63.1 percent of redraws at a mean premium of 1.07 calls. Adding one model leaves every frozen certificate valid, and refitting changes the certified calls by 0.07 on average. The operational rule matches the transfer analysis, in that the fitted structure can be refit at a mean premium of about one call whereas the cutoff must be recalibrated whenever the deployment pool or distribution changes.

Sensitivity to router degradation. The router-swap result in Section 5 holds between two routers of comparable accuracy. Table 19 degrades the router directly, by temperature scaling, by systematic over- and underconfidence, and by adding Gaussian noise with standard deviation up to 0.40 to its scores, and refits nothing else. On RouterBench MMLU at $\alpha = 0.10$, CopRoute-Ind's certified calls move by at most $+0.38$ under any transformation (systematic overconfidence) and by $+0.05$ under the strongest noise, and on IFEval by at most $+0.03$, whereas under the strongest noise on IFEval exact independence moves by $+0.59$ and RACER-G by $+1.13$ with an interval of $[+0.38, +1.84]$. The realized risk stays within sampling error of the target in every cell, since calibration absorbs score quality into calls rather than risk. Together with the RouterDC swap, these transformations cover

Table 14: Native BaRP preference sensitivity. Entries are official quality / mean cost multiplied by 10^3. The macro column averages HellaSwag and MBPP. Values aggregate five training seeds.

quality wt.	cost wt.	HellaSwag	MBPP	macro
1.0	0.0	0.799 / 1.356	0.575 / 5.810	0.687 / 3.583
0.9	0.1	0.799 / 1.356	0.575 / 5.810	0.687 / 3.583
0.8	0.2	0.713 / 0.359	0.573 / 5.755	0.643 / 3.057
0.7	0.3	0.654 / 0.159	0.459 / 0.243	0.556 / 0.201
0.6	0.4	0.563 / 0.149	0.449 / 0.227	0.506 / 0.188
0.5	0.5	0.461 / 0.135	0.471 / 0.200	0.466 / 0.167
0.4	0.6	0.428 / 0.128	0.502 / 0.183	0.465 / 0.155
0.3	0.7	0.428 / 0.127	0.516 / 0.181	0.472 / 0.154
0.2	0.8	0.424 / 0.122	0.484 / 0.155	0.454 / 0.138
0.1	0.9	0.362 / 0.083	0.420 / 0.111	0.391 / 0.097
0.0	1.0	0.341 / 0.081	0.427 / 0.109	0.384 / 0.095

Table 15: Cost and final-answer accuracy on RouterBench MMLU at matched certified risk; entries are means over 20 redraws with standard errors in brackets, and the best calibrated value per column is shaded. The last two rows are fixed references reported without calibration.

	$\alpha = 0.05$			$\alpha = 0.10$		
policy	calls	accuracy (%)	cost (USD/1k)	calls	accuracy (%)	cost (USD/1k)
CopRoute-Ind	7.75 (0.27)	80.44 (0.19)	3.41 (0.10)	2.99 (0.05)	80.97 (0.18)	1.90 (0.05)
CopRoute-Dep	7.75 (0.27)	80.44 (0.19)	3.41 (0.10)	2.99 (0.05)	80.97 (0.18)	1.90 (0.05)
RACER-G	8.66 (0.10)	80.19 (0.21)	4.24 (0.01)	5.64 (0.07)	80.18 (0.21)	3.55 (0.03)
RACER-P	8.64 (0.04)	80.19 (0.21)	4.22 (0.01)	6.45 (0.04)	80.13 (0.22)	3.66 (0.02)
exact independence	9.45 (0.18)	80.19 (0.21)	4.28 (0.03)	3.87 (0.07)	80.19 (0.21)	2.83 (0.03)
design-marginal order	9.44 (0.18)	80.20 (0.21)	4.30 (0.03)	3.76 (0.08)	80.66 (0.19)	2.86 (0.06)
call every model	11.00 (0.00)	80.19 (0.21)	4.47 (0.01)	11.00 (0.00)	80.19 (0.21)	4.47 (0.01)
best single model (1 call)	1.00 (0.00)	81.39 (0.19)	1.43 (0.00)	1.00 (0.00)	81.39 (0.19)	1.43 (0.00)

the router-sensitivity range from degraded to strong, and the co-failure ordering is the component least affected by the router.

Synthetic mechanism map. Table 20 tests the estimators on five synthetic regimes with known ground-truth dependence. The two regimes without residual dependence are negative controls in which no dependence-aware ordering wins a single cell, including the regime where pooled query difficulty creates large raw correlation with no exploitable structure. Where residual dependence is present, the ranking follows its form, so that with redundant model families the pairwise count wins 14 of 15 cells, with complementary specialists the query-local ordering wins from the smallest budget, and in the higher-order regime, whose dependence is invisible to every pairwise statistic, only the two estimators that use the full joint pattern win any cells, the tally on 3 of 15 and the latent normal on 2 of 15, while the pairwise count and the correlation order win none. The pattern of wins therefore matches the form of dependence built into each regime.

Generative specification. A synthetic query is a record with a scalar difficulty and a binary correctness vector over m simulated models; no text and no language model is involved, and the routing code consumes these records exactly as it consumes benchmark correctness labels. The difficulty is standard normal in four regimes and a fair binary value in the pooled-correlation regime. Each model's conditional correctness probability is a logistic function of $(1, x, x^2 - 1)$ whose coefficients vary deterministically with the model index, clipped away from zero and one; the higher-order regime instead fixes every marginal at one half. Dependence enters only through how correctness is drawn given these marginals.

- *Conditional independence with mixed difficulty.* Independent draws given x, with per-model intercepts $1.25 - j/(m-1)$, difficulty loadings $0.65 + 0.55\,((j \bmod 5)/4)$, and curvature $0.10\sin(j + 0.7)$.

Table 16: Cost-weighted stopping on RouterBench, reporting the certified mean cost per 1,000 queries in USD, and paired differences against count-optimized CopRoute-Ind with bootstrap 95 percent intervals; every cost interval excludes zero.

		cost, USD per 1k queries			cost-weighted minus CopRoute-Ind	
dataset	α	cost-weighted	CopRoute-Ind	RACER-G	Δcost [CI95]	Δcalls [CI95]
ARC	0.05	0.183	0.996	1.049	-0.813 [-0.891, -0.756]	+2.73 [+2.53, +2.92]
	0.10	0.104	0.924	0.893	-0.820 [-0.829, -0.812]	+1.82 [+1.62, +2.00]
HellaSwag	0.05	1.089	2.623	3.684	-1.533 [-1.663, -1.409]	+2.66 [+2.46, +2.87]
	0.10	0.431	2.277	2.176	-1.845 [-1.861, -1.830]	+1.44 [+1.35, +1.53]
MMLU	0.05	3.046	3.405	4.238	-0.359 [-0.469, -0.247]	+0.71 [+0.45, +0.96]
	0.10	0.680	1.895	3.552	-1.216 [-1.310, -1.136]	+1.87 [+1.79, +1.94]
WinoGrande	0.05	0.062	0.567	1.317	-0.505 [-0.508, -0.502]	+0.40 [+0.34, +0.47]
	0.10	0.050	0.547	1.004	-0.496 [-0.498, -0.494]	+0.74 [+0.67, +0.82]

Table 17: Realized all-miss risk of the pre-response cascade against matched CopRoute-Ind on identical splits; a star marks values above the target. The cascade carries no distribution-free certificate in any cell.

	$\alpha = 0.05$		$\alpha = 0.10$	
dataset	cascade	CopRoute-Ind	cascade	CopRoute-Ind
ARC (pool 11)	0.093*	0.032	0.099	0.033
GPQA (pool 100)	0.076*	0.066*	0.139*	0.121*
IFEval (pool 100)	0.100*	0.049	0.315*	0.095
MATH-algebra (pool 100)	0.139*	0.065*	0.350*	0.110*

- *Complementary specialists.* Models form pairs with opposite-sign difficulty loadings $\pm 1.25x$, and each pair shares one uniform draw. The left member is correct when the draw falls below its marginal and the right member when it falls above one minus its marginal, so within-pair failures are as negatively dependent as the marginals permit.
- *Redundant families.* Blocks of four models share one uniform draw, so family members succeed or fail together as far as their marginals allow.
- *Higher-order and pairwise-invisible.* Models form triples in which the first two are independent fair bits and the third is their exclusive-or; every pairwise correlation is exactly zero while the triple is deterministic given any two members. Models left over from an incomplete triple are independent fair bits.
- *Pooled correlation without residual dependence.* The marginals shift by 0.55 between the easy and hard difficulty values and draws are independent given difficulty, so pooling over queries creates large raw correlation with no conditional dependence.

Each cell uses pool sizes $m \in \{8, 16, 32\}$, design budgets $n_d \in \{50, 100, 250, 500, 1000\}$, and twenty seeds, with independent streams of 600 validation, 1,000 final-calibration, and 2,000 test queries per seed at targets $\alpha \in \{0.05, 0.10\}$. The remaining phase constants are fixed in the released generator.

M LIMITATIONS

This section expands the limitations summarized in Section 6, stating for each what the present work covers and what would extend it.

What the certificate covers. Theorem 2 bounds the probability that a certificate is issued and the certified set misses, averaged over the deployment distribution. This is the quantity a routing layer must control, and it is the same quantity that RACER certifies. It is not a per-query guarantee, and it does not by itself certify the answer that a downstream aggregator returns from the called set; the cost case study of Appendix L evaluates one such selector, and certificates for aggregators are a separate line of work. At the tightest targets on the hardest benchmarks, certification is infeasible for every policy alike because the full pool misses too many calibration queries (Section 5); a larger calibration split or a larger pool, rather than a different policy, moves that boundary.

Table 18: Pool composition shift over 51 dataset, pool, and target cells of RouterBench and RouterEval (1,020 redraws, of which 729 certify on the unshifted pool, a rate of 71.5 percent). A frozen certificate survives when the unshifted order and cutoff still certify on the shifted pool without any refitting. The last two columns refit the order and recalibrate the cutoff on the shifted pool and report the fraction of all 1,020 redraws that then certify and the mean change in certified calls relative to the unshifted pool.

policy	intervention	frozen certificates surviving	certified after refit	mean call change
CopRoute-Ind	remove one model	3.4% (25 of 729)	63.1%	+1.07
CopRoute-Ind	add one model	100.0% (729 of 729)	73.6%	+0.07
CopRoute-Dep	remove one model	2.2% (16 of 729)	63.1%	+0.98
CopRoute-Dep	add one model	100.0% (729 of 729)	73.6%	-0.02

Table 19: Router degradation at $\alpha = 0.10$, measured as the change in certified mean calls relative to the unmodified KNN router when its scores are transformed and nothing else is refitted, paired over 20 redraws with bootstrap 95 percent intervals; bold intervals exclude zero. The realized test all-miss risk stays within sampling error of the target in every cell (0.085 to 0.103 across the three policies), so degradation is paid in calls rather than in risk.

score transformation	CopRoute-Ind	exact independence	RACER-G
RouterBench MMLU, 11 models			
temperature scaling	-0.01 [-0.03, +0.01]	**+0.03 [+0.01, +0.05]**	**-0.11 [-0.14, -0.08]**
systematic overconfidence	**+0.38 [+0.32, +0.44]**	**+0.15 [+0.08, +0.21]**	**+0.46 [+0.41, +0.52]**
systematic underconfidence	**+0.05 [+0.02, +0.09]**	**+0.17 [+0.10, +0.23]**	**-0.51 [-0.57, -0.45]**
Gaussian noise, $\sigma = 0.05$	-0.01 [-0.02, +0.01]	-0.01 [-0.05, +0.04]	+0.01 [-0.02, +0.05]
Gaussian noise, $\sigma = 0.10$	0.00 [-0.01, +0.02]	-0.03 [-0.08, +0.02]	-0.01 [-0.04, +0.02]
Gaussian noise, $\sigma = 0.20$	+0.01 [-0.01, +0.03]	-0.05 [-0.11, 0.00]	**-0.11 [-0.15, -0.06]**
Gaussian noise, $\sigma = 0.40$	**+0.04 [+0.02, +0.07]**	+0.05 [-0.02, +0.13]	**-0.23 [-0.30, -0.16]**
RouterEval IFEval, 100 models			
temperature scaling	-0.01 [-0.04, +0.03]	0.00 [-0.01, +0.02]	**+0.23 [+0.10, +0.36]**
systematic overconfidence	0.00 [-0.13, +0.12]	+0.01 [-0.02, +0.05]	-0.15 [-0.50, +0.11]
systematic underconfidence	+0.02 [-0.06, +0.12]	-0.01 [-0.08, +0.04]	**+0.17 [+0.05, +0.29]**
Gaussian noise, $\sigma = 0.05$	-0.05 [-0.10, 0.00]	-0.06 [-0.23, +0.10]	-0.07 [-0.43, +0.22]
Gaussian noise, $\sigma = 0.10$	-0.06 [-0.13, +0.02]	-0.10 [-0.45, +0.18]	+0.01 [-0.44, +0.40]
Gaussian noise, $\sigma = 0.20$	-0.04 [-0.15, +0.08]	-0.02 [-0.62, +0.48]	-0.02 [-0.75, +0.62]
Gaussian noise, $\sigma = 0.40$	+0.03 [-0.14, +0.24]	+0.59 [0.00, +1.14]	**+1.13 [+0.38, +1.84]**

Dependence on the deployment pool and distribution. The tally order is fitted per benchmark and pool, and the cutoff is calibrated per deployment. Appendix J shows that an order fitted on one dataset remains useful on a related one at a bounded call premium, and Appendix L shows that adding a model to the pool leaves every certificate intact while removing one requires refitting and recalibrating, at a mean premium of about one call. The operational rule is therefore to refit the order and recalibrate the cutoff whenever the pool or the query distribution changes; both steps run on labels only and add about one call on average after a pool change.

Dependence-corrected stopping. The correction of CopRoute-Dep is estimated from the design queries that the entire prefix misses. These design queries become rare at the depths where a correction would matter, so the estimation noise exceeds what the correction recovers (Table 7) and CopRoute-Ind is the default. The default reflects the present estimator of the tail, since the order already exploits the dependence information, and a structured estimator of the tail, or a design set several times larger, are the two routes we see to a stopping rule that improves on the independence product.

Scope of the theory. Theorem 1 is a guarantee on the design objective, the mean position of the first correct model under a fixed order, and Appendix D shows when it transfers to fresh queries. It does not bound the number of certified calls at a fixed target, which depends on a quantile of the first-success depth rather than on its mean; the number of calls that CopRoute makes at deployment is therefore an empirical finding, supported across ten datasets, four pool sizes, seven targets, and two routers, while the certificate is the guarantee that holds at test time.

Evaluation on recorded labels. All experiments consume per-query correctness labels released with RouterBench and RouterEval, and no language model is queried. Recorded labels permit 20 redraws across ten datasets and pools of up to 500 models, and they are the protocol under which

Table 20: Synthetic mechanism map, giving the cells out of 15 (three pool sizes, five design budgets) where the ordering beats the design-marginal reference with a bootstrap interval excluding zero, per ground-truth regime.

ground-truth regime	tally	pairwise count	error corr.	latent normal	query-local
conditional independence, mixed difficulty	0/15	0/15	0/15	0/15	0/15
pooled correlation, no residual dependence	0/15	0/15	0/15	0/15	0/15
redundant families (positive residual)	**10/15**	**14/15**	**12/15**	**10/15**	**13/15**
complementary specialists (negative residual)	8/15	6/15	6/15	0/15	**15/15**
higher-order, pairwise-invisible	3/15	0/15	0/15	2/15	0/15

RACER, FlexRouter, and BaRP are evaluated. It also means that latency, response-dependent cascades, and answer aggregation beyond the router-top selector of Appendix L are outside the evidence presented here, and that the cost case study prices calls on one benchmark with a public price sheet. FlexRouter and BaRP are reconstructions from their papers' specifications, evaluated under the common set-routing protocol rather than their native ones; Appendix E records every reconstruction choice and Appendix K reports their native metrics.

N Reproducibility

The routing evaluations are labels-only computations on public RouterBench and RouterEval releases. Learned baseline fitting uses frozen public query embeddings, while the common evaluation still uses cached correctness matrices and requires no new language-model inference. The anonymized release contains estimator and learned-baseline implementations, split and selection logic, fixed hyperparameter grids, five-seed manifests, per-redraw certification status, standard and augmented diagnostic losses under distinct names, and permutation records. The release includes the aggregated results behind the main comparison (Tables 1, 2, and 3, Figure 1, and their appendix expansions) together with the scripts that regenerate those tables and figures from those results; the per-redraw artifacts behind the transfer, pool-shift, router-degradation, stopping-rule ablation, and synthetic mechanism-map tables are not included in the release because of their size, and those tables are reproduced by rerunning the corresponding scripts on the public benchmark data.