WHERE DOES HARMFUL REPLACEMENT COME FROM? COVERAGE AND SWITCHING IN FROZEN MODEL ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adding a frozen expert gives a router another possible answer, but also another way to replace a correct incumbent answer with an incorrect one. We introduce a diagnostic protocol for this trade-off when two new candidates are available. Holding a fixed support budget of 64 candidate evaluations per intervention, a *pair-fixed* policy uses one candidate on every eligible request, whereas a *per-request* policy chooses between the two candidates separately for each request. Across two development assets, per-request choice increases net accuracy gain relative to the incumbent by 0.182 percentage points (pp) compared with pair-fixed choice, while increasing harmful-replacement mass by 0.201 pp. An exact request-wise decomposition attributes the difference to *coverage*, where one policy returns a candidate and the other retains the incumbent, and *switching*, where both return candidates but return different ones. A controlled 2×2 experiment shows that these decisions interact: expanding candidate use can help or hurt depending on which candidate is returned on the added requests. A held-out text-model asset and an independently constructed score reproduce the interaction and the sign reversal of the coverage effect, whereas switching-harm reduction is not stable across assets or score constructions.

1 INTRODUCTION

Serving a query is no longer just a lookup. A router may choose among several frozen models, because different models are better on different requests and have different costs (Ong et al., 2025; Shnitzer et al., 2023; Hu et al., 2024). This flexibility creates a practical question when a new model becomes available: should the router start using it?

Adding a candidate can rescue an error. If the incumbent is wrong and the candidate is correct, the replacement is beneficial. The same change can also damage an answer that was already correct: if the incumbent is correct and the candidate is wrong, the replacement is harmful. An onboarding analysis therefore needs to report both events, rather than only the final average accuracy (Yan et al., 2021; Echterhoff et al., 2024).

When two candidates are added, onboarding involves two decisions. The first is *adoption*: on which requests should the system leave the incumbent? The second is *identity*: which candidate should answer after that change? These decisions are often combined into one routing score, but they need not have the same effect. A policy may adopt candidates on a new set of requests, or keep the same adopted set while changing which candidate is returned.

We study this distinction in a frozen setting. The incumbent pool, candidate models, router score, and cached outputs remain fixed. For each candidate pair, a *pair-fixed* policy selects one candidate for the pair, whereas a *per-request* policy chooses between the candidates separately for each request. The two policies use the same support budget and evaluation record; they differ only in the granularity of candidate choice. The ledger shows whether the policies differ because they adopt different requests or choose different candidates on the same requests.

The data show that adoption and identity interact. Expanding candidate use can reduce the coverage contribution when it is tied to one candidate, but increase it when the policy can choose the candidate

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.

per request. On the two development assets, per-request choice raises pooled net gain by 0.182 percentage points, while harmful-replacement mass also rises by 0.201 pp. The gain and the harm come from the same expansion, so reporting accuracy alone hides the relevant trade-off.

The controlled 2×2 analysis shows where this interaction lives: it is concentrated in the requests that the two adoption rules treat differently. The same interaction direction appears on the held-out asset Z and under a second score construction. On Z, the switching-harm estimate is close to zero and its interval includes zero. The result gives a way to diagnose an onboarding change; it does not by itself justify deploying per-request choice.

Contributions.

- We frame frozen-model onboarding as two linked decisions: whether to adopt a candidate and which candidate to use.
- We give an exact request-wise decomposition into *coverage* and *switching*, and report beneficial replacements, harmful replacements, and net gain separately.
- We show with a controlled 2×2 comparison that the effect of adoption depends on candidate identity, and identify the request group that accounts for most of the interaction.
- We examine how these findings change across held-out data, score constructions, support budgets, operating points, and aggregation units.

2 Setting and Evaluation Protocol

Frozen pool and incumbent. Each intervention starts with an old pool E of frozen experts. For request x, the frozen router selects

$$e_R(x) = \arg\max_{e \in E} s_R(x, e),$$

which we call the *incumbent*. The intervention adds two frozen candidates $C = \{c_1, c_2\}$. Candidates do not retrain the router; the same frozen scoring function and support-profile construction are used throughout. This is a frozen-pool, per-instance selection setting related to classical mixtures of experts and modern black-box routing (Jacobs et al., 1991; Jordan & Jacobs, 1994; Shnitzer et al., 2023; Hu et al., 2024). Let $Y_e(x) \in \{0, 1\}$ indicate whether expert e is correct.

Pair-fixed and per-request policies. For policy π, let $A_\pi(x) \in \{0, 1\}$ indicate whether a candidate replaces the incumbent, and let $J_\pi(x) \in C$ be the candidate named when $A_\pi(x) = 1$. The returned model and its correctness are

$$M_\pi(x) = \begin{cases} J_\pi(x), & A_\pi(x) = 1, \\ e_R(x), & A_\pi(x) = 0, \end{cases} \qquad Y_\pi(x) = Y_{M_\pi(x)}(x).$$

The *pair-fixed* policy, denoted π_{pair}, chooses one candidate j_{pair} from the shared support profile and sets $J_{\text{pair}}(x) = j_{\text{pair}}$ for every evaluation request on which it replaces the incumbent. The *per-request* policy, denoted π_{req}, instead uses

$$J_{\text{req}}(x) = \arg\max_{c \in C} s_R(x, c),$$

with deterministic tie breaking. Both policies can return c_1, c_2, or the incumbent; only the granularity of candidate choice differs. Thus "pair-fixed" means fixed across the whole candidate pair, whereas "per-request" means re-evaluated separately for each request.

Evaluation quantities. All probabilities below are empirical proportions over the held-out evaluation requests, pooled as specified below. Define

$$B_\pi = \mathbb{P}(Y_\pi = 1, Y_{e_R} = 0), \quad H_\pi = \mathbb{P}(Y_\pi = 0, Y_{e_R} = 1), \quad G_\pi = B_\pi - H_\pi.$$

Thus B is beneficial-replacement mass, H is harmful-replacement mass, and G is net accuracy gain relative to retaining the incumbent. Adoption is $\mathbb{P}(A_\pi = 1)$. Throughout the main text, Δ means per-request minus pair-fixed; in particular, $\Delta G = \Delta B - \Delta H$. The separation of beneficial and harmful flips follows the negative-flip perspective in model update evaluation (Yan et al., 2021).

Policy	G (pp)	H (pp)	Adoption
pair-fixed	1.008	1.127	19.34%
per-request	1.190	1.328	22.18%
Margin gate ($\tau = 0.01$)	1.190	1.309	21.85%

Table 1: **Core policy levels.** Request-pooled evaluation estimates. G is net accuracy gain relative to the incumbent and H is harmful-replacement mass. The margin-gated row falls back to pair-fixed. The action rules behind each row are in Appendix B; the decomposition checks are in Appendix B, the counterfactual cells and interaction intervals are in Appendix C, and the matched-risk accounting is in Appendix C.

Splits, budget, and uncertainty. The support profile contains 32 requests evaluated on both candidates, hence 64 candidate evaluations per intervention. A separate tuning split selects thresholds; the held-out evaluation split is used only to report utility. Within an intervention, pair-fixed and per-request share the old pool, candidates, support data, tuning data, and evaluation requests. These roles are disjoint throughout: the requests that build the support profile are not used to fit the score, to tune thresholds, or to report utility, and the off-sample analysis of Section 3.3 draws its diagnostic rows from calibration requests that the support profile does not use. The equal budget equalizes *profile construction*, not the online cost of scoring candidates on the evaluation split; uncertainty is summarized with cluster-aware resampling rather than treating requests as independent (Efron, 1979; Davison & Hinkley, 1997; Good, 2000): the per-request rule needs both candidates scored on every request it considers, the pair-fixed rule only its one fixed candidate. Appendix B describes both costs.

We use two retrospective development assets, S and U, and one held-out asset, Z. Each intervention uses six incumbent experts and 32 support requests; registry sizes and evaluation counts are in Appendix B. The protocol for Z was frozen before its outcomes were inspected. Headline intervals quantify variation across intervention draws under the fixed cached assets. A candidate-pair clustered bootstrap quantifies variation across the assets' candidate pairs and is reported as a sensitivity analysis: it widens the intervals but preserves the signs of both interaction terms. Neither projects onto new prompts, new pairs or new model families. Headline estimates pool requests, while Appendix H reports matrix-equal and intervention-equal weighting as sensitivity analyses.

3 Adoption and Candidate Choice Interact

A policy change can alter either whether a candidate is returned or which candidate is returned (Dekoninck et al., 2025; Ding et al., 2024; Chen et al., 2024a). We first compare pair-fixed and per-request, then separate these two sources of change, and finally test their interaction.

3.1 Main comparison and exact decomposition

At equal profile budget, per-request exceeds pair-fixed by $\Delta G = 0.182$ pp (95% CI $[0.163, 0.203]$), while $\Delta H = 0.201$ pp ($[0.183, 0.219]$) and adoption rises by 2.84 pp ($[2.60, 3.09]$). The signs agree on both development assets. Per-request selection therefore improves average net accuracy, but it does not improve the measured harm metric.

Let

$$\mathcal{E}_{\text{cov}} = \{A_{\text{pair}} \neq A_{\text{req}}\}$$

be the *coverage* event: one policy returns a candidate and the other retains the incumbent. Let

$$\mathcal{E}_{\text{sw}} = \{A_{\text{pair}} = A_{\text{req}} = 1,\ J_{\text{pair}} \neq J_{\text{req}}\}$$

be the *switching* event: both policies return candidates, but return different ones. These events are disjoint and contain every request on which pair-fixed and per-request can return different models. This is an accounting identity; the interaction test appears in the next subsection. (Simpson, 1951).

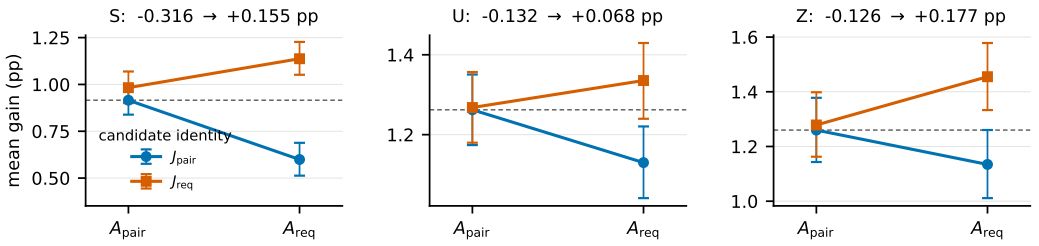

Figure 1: **Coverage depends on candidate identity.** The x-axis changes the adoption rule: A_{pair} is pair-fixed and A_{req} is per-request. Each line holds the candidate identity fixed: J_{pair} is the pair-fixed candidate and J_{req} is the per-request candidate, so points are the crossed cells $(A_{\text{rule}}, J_{\text{candidate}})$. The coverage contrast is negative at J_{pair} and positive at J_{req}. Intervals are 95% clustered intervals; the dashed line is the $(A_{\text{pair}}, J_{\text{pair}})$ baseline.

Proposition 1 (Exact request-wise split)**.** *Define the correctness difference* $D_G(x) = Y_{\text{req}}(x) - Y_{\text{pair}}(x)$ *and the harmful-replacement indicator* $h_\pi(x) = \mathbf{1}\{Y_{e_R}(x) = 1, Y_\pi(x) = 0\}$. *Then*

$$D_G = D_G\mathbf{1}\{\mathcal{E}_{\text{cov}}\} + D_G\mathbf{1}\{\mathcal{E}_{\text{sw}}\}, \qquad h_{\text{req}} - h_{\text{pair}} = (h_{\text{req}} - h_{\text{pair}})\mathbf{1}\{\mathcal{E}_{\text{cov}}\} + (h_{\text{req}} - h_{\text{pair}})\mathbf{1}\{\mathcal{E}_{\text{sw}}\}.$$

Averaging over the evaluation split gives

$$\Delta G = \Delta G_{\text{coverage}} + \Delta G_{\text{switching}}, \qquad \Delta H = \Delta H_{\text{coverage}} + \Delta H_{\text{switching}}.$$

Proof. Outside $\mathcal{E}_{\text{cov}} \cup \mathcal{E}_{\text{sw}}$, pair-fixed and per-request return the same model, so both pointwise differences are zero. The two events are disjoint, hence their indicator-weighted terms sum request by request. The expanded event decomposition is in Appendix B. □

Coverage accounts for $\Delta G_{\text{coverage}} = 0.128$ pp and $\Delta H_{\text{coverage}} = 0.219$ pp. Switching accounts for $\Delta G_{\text{switching}} = 0.054$ pp and $\Delta H_{\text{switching}} = -0.018$ pp. Thus coverage supplies most of the additional net gain and all of the net harm increase in the pooled development result; switching is smaller and offsets part of that harm. The per-asset estimates and the exact $\Delta B/\Delta H/\Delta G$ identities are in Appendix B.

3.2 Controlled 2x2 interaction

The coverage and switching rows are an exact accounting identity. To test whether the two decision axes interact, we evaluate all four combinations of adoption and candidate-choice rules.

The coverage contrast changes sign: it contributes -0.268 pp to G when candidate identity is held to J_{pair}, but $+0.131$ pp when identity is held to J_{req}. Harm changes as well ($+0.478$ versus $+0.216$ pp). Equivalently, the effect of replacing J_{pair} with J_{req} depends on which adoption rule is used. Figure 1 shows that the adoption effect changes with the candidate-choice rule. The four policy definitions, interaction contrasts, and intervals are in Appendix D.2.

3.3 What the interaction is made of

Regrouping the same evaluation record by policy actions shows that the requests newly served by the per-request policy account for 99.2% of the absolute gain and harm interaction contributions. They form 3.103% of pooled evaluation requests, whereas requests served only by pair-fixed form 0.264%. That concentration has an exact form.

Here I_G measures the 2×2 interaction, whereas ΔG is the endpoint difference between the two policies.

Let $J_{\text{pair}}(x)$ and $J_{\text{req}}(x)$ denote the candidate each rule names on request x, defined for every request, including those on which a policy keeps the incumbent, and let $A_{\text{pair}}(x), A_{\text{req}}(x)$ indicate whether a candidate is returned. Write $\delta(x) = Y_{J_{\text{req}}(x)}(x) - Y_{J_{\text{pair}}(x)}(x)$ for the signed correctness difference between the two named candidates on x.

Proposition 2 (Interaction as scale times selection advantage)**.** *With* $C_{\text{add}} = \{A_{\text{pair}} = 0, A_{\text{req}} = 1\}$ *and* $C_{\text{drop}} = \{A_{\text{pair}} = 1, A_{\text{req}} = 0\}$*, writing* $p_g = \mathbb{P}(C_g)$ *and* $\mu_g = \mathbb{E}[\delta \mid C_g]$*,*

$$I_G = \mathbb{E}\big[(A_{\text{req}} - A_{\text{pair}})\,\delta\big] = p_{\text{add}}\mu_{\text{add}} - p_{\text{drop}}\mu_{\text{drop}}, \qquad I_H = -\mathbb{E}\big[(A_{\text{req}} - A_{\text{pair}})\,Y_{e_R}\delta\big].$$

Proof. The coefficient $A_{\text{req}} - A_{\text{pair}}$ is $+1$ on C_{add}, -1 on C_{drop}, and zero on the two groups where the adoption rules agree — the groups that contribute zero to both interactions. Expanding I_G and I_H as the two contrasts of the 2×2 therefore leaves exactly the terms displayed; the expansion is written out in Appendix D.3. □

Corollary 1 (The interaction needs a disagreement set)**.** *If the two adoption rules agree on every request, then* $I_G = I_H = 0$*. More generally the interaction can be non-zero only through the requests where the rules disagree, so it is a property of the two rules jointly rather than of either one.*

Proof. With $A_{\text{pair}} = A_{\text{req}}$ on every request the coefficient $A_{\text{req}} - A_{\text{pair}}$ vanishes, and both expectations in Proposition 2 are zero; the two groups it sums over are subsets of the disagreement set, so outside it the products are zero as well. □

Three components are separable, and they answer different questions. The distinction between opportunity, selection, and cancellation echoes ensemble-diversity analyses, but here it is applied to frozen black-box candidates (Krogh & Vedelsby, 1995; Caruana et al., 2004; Jiang et al., 2023). *Scale*: the two rules change the served set by $p_{\text{add}} = 3.103\%$ and $p_{\text{drop}} = 0.264\%$ of requests, so the added group is an order of magnitude larger. *Selection advantage*: on the added requests the per-request rule names the better candidate, $\mu_{\text{add}} = +12.76$ pp, and on the dropped ones it names the worse one, $\mu_{\text{drop}} = -1.15$ pp. *Cancellation*: the two contributions are subtracted, so the interaction depends on their difference and not on either advantage alone — equal weighted contributions would cancel even with both advantages non-zero, and the interaction is exactly zero whenever the two rules agree on every request. In these data the two reinforce rather than cancel: pooled, $p_{\text{add}}\mu_{\text{add}} = +0.396$ pp and $-p_{\text{drop}}\mu_{\text{drop}} = +0.003$ pp. The harm counterpart applies the same expression to requests on which the incumbent is right and carries an overall minus sign, which is why the harm interaction is negative while the gain interaction is positive. Per-asset values are in Appendix D.3; the exploratory analyses of switching mass and choice skill remain there as well, because those quantities do not transfer reliably across assets.

Is the selection advantage visible before the evaluation outcomes are used? Proposition 2 reads μ_{add} off the evaluation split, so it cannot by itself say whether a pair's advantage is visible off-sample. Two questions must be separated, because they have different answers. *Whether* the mechanism can act at all is itself informative: on the calibration rows the support profile does not use (4,545 for S, 1,618 for U, 4,968 for Z, disjoint from the support profile, the tuning split, the score fit and the evaluation split), S has 125 of 128 pairs with at least one newly adopted request, U 42 of 47, Z 1,173 of 1,723 (68%), and the pairs without one have an evaluation-side interaction of essentially zero (-0.000, $+0.000$, $+0.002$ pp) against $+0.494$, $+0.214$, $+0.432$ pp for the pairs that have one. *How large* it is, among the pairs where it can act, is positively rank-associated with the off-sample selection advantage: Spearman $\rho = +0.70$ $[+0.59, +0.78]$, $+0.69$ $[+0.46, +0.83]$ and $+0.56$ $[+0.51, +0.60]$ on S/U/Z (bootstrap over pairs), computed over exactly the pairs of the first part. The association is descriptive and is computed only for pairs whose calibration data show an adoption change.

Figure 2 shows the pair-level fact the pooled number hides: in most pairs the two terms of Proposition 2 have *opposite* signs — 94 of 128 pairs on S, 40 of 47 on U, 1,596 of 1,723 on Z — so the positive pooled interaction is the net of pair-level terms that largely cancel. The mechanism is a statement about the average pair, not about every one. Pair-level heterogeneity is therefore reported descriptively rather than treated as independent replication (Efron, 1979; Simpson, 1951).

4 Held-out Replication and Score Robustness

Everything in Section 3 is measured under the frozen R0 score construction. We keep the discovery assets, policy construction, and held-out checks separate (Geifman & El-Yaniv, 2017; 2019; Romano

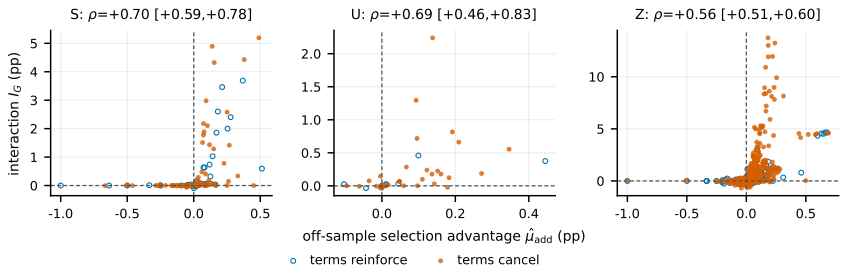

Figure 2: **Pair-level heterogeneity and off-sample selection advantage.** Each panel plots the pair's calibration-side selection advantage against its evaluation-side interaction; the title reports the pair-bootstrap Spearman interval. Filled points have opposite-signed identity terms, whereas open points have reinforcing terms. The association is conditional on an identifiable calibration-side adoption change and is descriptive rather than predictive; pairs without that change are reported in the text and supplementary artifact.

et al., 2020; Angelopoulos et al., 2022). The cache also contains a second score construction, R1, built independently of this project's evaluation outcomes. Here independently constructed refers to the score construction, not to a new dataset: R1 reads the same input field on the same cached assets. This section therefore asks two separate robustness questions: does the interaction survive on a held-out asset, and does it survive a different way of constructing the score?

4.1 A HELD-OUT ASSET AND AN INDEPENDENT SCORE

Everything above is discovery on two retrospective development assets. Asset Z is a cross-dataset, cross-task *text-model* routing asset that was not used in method development; its protocol was frozen and its 2,000 interventions built before any outcomes were inspected (provenance, item key and counts in Appendix G). The assets are reported separately and never pooled; cross-asset and cross-score differences are not tested.

The interaction is reproduced on Z and remains concentrated in the same action group: I_G and I_H are non-zero with the same signs, the coverage leg's gain still reverses with the conditioning, and C_{add} still accounts for at least 98% of the absolute gain and harm interaction contributions. The reduction in switch harm is not consistently supported: on Z, the estimate is close to zero and its interval includes zero. The coverage row accounts for the whole *net* harm increase (108.7% of pooled ΔH, 100.5% on Z), not for all harmful replacement, since switch can introduce and remove harm within itself and only its net is reported.

4.2 WHAT A SECOND SCORE CHANGES

R1 changes only the score construction: it uses a frozen pretrained sentence encoder and a per-model correctness head fit on the same source labels, with the head fixed before replay. The requests, splits, policies, threshold grid, and cached outcomes are unchanged; both scores read the same text field. Thus R1 is an independently constructed score, not an independent dataset or input. Builder details and source-side diagnostics are in Appendix G.1.

The interaction and the coverage sign reversal persist under R1 on all three assets. This comparison tests sensitivity to score construction; it does not add a second dataset (Hanley & McNeil, 1982; DeLong et al., 1988). Switching-harm reduction remains asset-dependent: it is supported on S but its interval covers zero on U and Z. What replicates is the interaction and the request group that carries it, not a general switching-harm reduction.

4.3 ROBUSTNESS UNDER SCORE PERTURBATIONS

The perturbation ladder preserves the interaction direction but not every component sign. We use the ladder as a fixed-protocol sensitivity analysis (Benjamini & Hochberg, 1995). Operating-point, task-group, and leave-one-model/family checks preserve the point-estimate interaction direction,

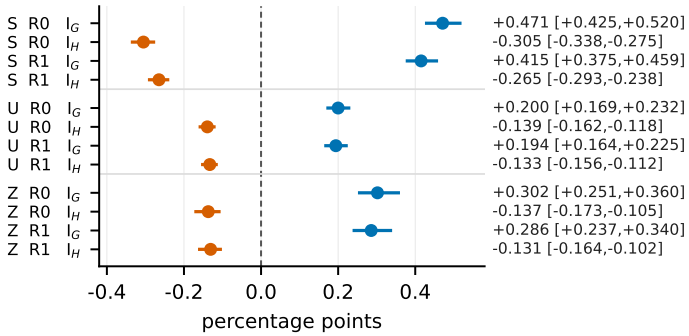

Figure 3: **Interaction estimates under R0 and R1.** Dots show point estimates and whiskers show 95% clustered bootstrap intervals; blue is I_G and orange is I_H. The listed intervals are positive for I_G and negative for I_H. The primary clustering unit is intervention; candidate-pair sensitivity is reported in Appendix D.1.

whereas endpoint gain is less stable. We therefore treat these analyses as sensitivity checks rather than independent replications.

5 Scope and Boundary Conditions

5.1 Does the support-set budget drive the interaction?

Every result above is measured at one profile budget: 32 support requests, or 64 candidate evaluations per intervention. We re-run the comparison at $n = 8, 16, 24, 32, 64$, keeping both policies at exactly $2n$ candidate evaluations and holding the remaining protocol fixed. Each budget uses ten nested support draws.

Across the tested budgets, the gain interaction remains positive, the harm interaction remains negative, and the coverage legs keep opposite signs in all fifteen asset–budget cells. Smaller supports have greater across-draw variability. The interaction magnitude shows an overall downward trend, but it is not strictly monotone: on Z it is $+0.69$ pp at 16 evaluations and $+0.19$ pp at 128. The support-budget record is in Appendix E.2; full pair-level outputs are in the supplementary files.

5.2 What a gate on the frozen score does

The ledger shows where the policy difference occurs. We next ask whether a frozen score can identify a safer subset in advance. At matched harm, the tested margin and support gates trade adoption for nearly the same net gain. The complete threshold grid is reported in Appendix D.

5.3 Cost and external validity

Profile construction uses 64 cached candidate evaluations per intervention; per-request scoring adds online work. S and U are development assets, while Z is the held-out asset. The perturbation ladder and R1 are single score constructions. These choices define the scope of the reported results. Requests within a candidate pair share models and items, and the package reports cluster-aware sensitivity analyses in Appendix I.

6 Discussion and Conclusion

The analysis gives a simple way to read model onboarding. Adding candidates changes both the set of requests that leave the incumbent and the identity of the candidate used after the change. These axes interact: in our controlled comparison, the coverage effect changes when the candidate identity rule changes. The interaction remains visible on the held-out asset and under the second score, while the switching-harm reduction varies across settings.

The practical lesson is to separate these quantities in an onboarding report. Measure beneficial replacements, harmful replacements, and net gain; then split policy differences into coverage and switching before deciding whether a new candidate helped. In this study, per-request choice improves pooled average gain but also increases harmful-replacement mass. The protocol exposes that trade-off; it does not turn the observed interaction into a generally safe deployment rule.

Reproducibility Statement

The paper uses a frozen cached-output protocol. The source, figures, bibliography, styles, analysis scripts, and supplementary run records are included in the released project. The experiments use disjoint support, tuning, and evaluation splits; all reported intervals resample interventions within asset. No live model or API calls are required to reproduce the reported tables and figures. The public source assets are referenced by provenance and hashes rather than redistributed.

AI Use Statement

Generative AI tools assisted with code drafting, figure preparation, and language editing. The authors designed and verified the analyses, checked the reported quantities against the cached records, and take responsibility for the manuscript, code, figures, and released artifacts.

References

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022. URL https://arxiv.org/abs/2208.02814.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *International Conference on Machine Learning*, pp. 18, 2004.

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. URL https://arxiv.org/abs/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. *arXiv preprint arXiv:2409.19886*, 2024b. URL https://arxiv.org/abs/2409.19886.

Mohammed Nowaz Rabbani Chowdhury, Shuai Zhang, Meng Wang, Sijia Liu, and Pin-Yu Chen. Patch-level routing in mixture-of-experts is provably sample-efficient for convolutional neural networks. In *International Conference on Machine Learning*, volume 202, pp. 6074–6114, 2023.

Anthony C. Davison and David V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge University Press, 1997.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for large language models. In *International Conference on Machine Learning*, 2025.

Elizabeth R. DeLong, David M. DeLong, and Daniel L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, 44(3):837–845, 1988. doi: 10.2307/2531595.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pp. 1–15. Springer, 2000. doi: 10.1007/3-540-45014-9_1.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed H. Awadallah. Hybrid-LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024.

Nan Du, Yanping Dai, Yanping Huang, Siyuan Wu, Tal Schuster, Zhifeng Chen, Aakanksha Chowdhery, Sharan Narang, et al. GLaM: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pp. 5542–5568, 2022.

Jessica Maria Echterhoff, Fartash Faghri, Raviteja Vemulapalli, Ting-Yao Hu, Chun-Liang Li, Oncel Tuzel, and Hadi Pouransari. MUSCLE: A model update strategy for compatible LLM evolution. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7320–7332, 2024. URL https://aclanthology.org/2024.findings-emnlp.430/.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. doi: 10.1214/aos/1176344552.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning*, volume 97, pp. 2156–2165, 2019.

Phillip I. Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer, 2000.

J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic curve. *Radiology*, 143(1):29–36, 1982. doi: 10.1148/radiology.143.1.7063747.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

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. URL https://arxiv.org/abs/2403.12031.

Zhehao Huang, Baijiong Lin, Jingyuan Zhang, Jingying Wang, Yuhang Liu, Ning Lu, Tao Li, and Xiaolin Huang. VL-RouterBench: A benchmark for vision-language model routing. *arXiv preprint arXiv:2512.23562*, 2025a. URL https://arxiv.org/abs/2512.23562.

Zhongzhan Huang, Guoming Ling, Vincent S. Liang, 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 *Findings of the Association for Computational Linguistics: EMNLP*, 2025b.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

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*, pp. 14165–14178, 2023. doi: 10.18653/v1/2023.acl-long.792.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient LLM inference. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ka82fvJ5f1.

Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994. doi: 10.1162/neco.1994.6.2.181.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7, 1995.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, Zhen Wang, and Shuyue Hu. LLMRouterBench: A massive benchmark and unified framework for LLM routing. In *Findings of the Association for Computational Linguistics: ACL*, 2026.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2022. URL https://arxiv.org/abs/2211.09110.

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. URL https://openreview.net/forum?id=8sSqNntaMr.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023. URL https://arxiv.org/abs/2309.15789.

Edward H. Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B*, 13(2):238–241, 1951. doi: 10.1111/j.2517-6161.1951.tb00088.x.

Xin Wang, Fisher Yu, Lisa Dunlap, Yi-An Ma, Ruth Wang, Azalia Mirhoseini, Trevor Darrell, and Joseph E. Gonzalez. Deep mixture of experts via shallow embedding. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 552–562, 2020.

Mitchell Wortsman, Gabriel Ilharco, Shuang Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weight-space models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998, 2022.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-congruent training: Towards regression-free model updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14299–14308, 2021.

Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. Capability instruction tuning: A new paradigm for dynamic LLM routing. *arXiv preprint arXiv:2502.17282*, 2025. URL https://arxiv.org/abs/2502.17282.

Yichi Zhang, Fangzheng Xie, Shu Yang, and Chong Wu. Meta-Router: Bridging gold-standard and preference-based evaluations in large language model routing. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=r0BFucF2dH.

Hang Zheng, Hongshen Xu, Yongkai Lin, Shuai Fan, Lu Chen, and Kai Yu. DiSRouter: Distributed self-routing for LLM selections. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KDcwXKr0NU.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7103–7117, 2022.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. *arXiv preprint arXiv:2410.02223*, 2024. URL https://arxiv.org/abs/2410.02223.

A Related Work

External model routing. External routing selects among separately served black-box models. RouteLLM studies a fixed strong/weak pair (Ong et al., 2025); Meta-Router and DiSRouter study learned routing and self-assessing model signals (Zhang et al., 2026; Zheng et al., 2026); UniRoute maps prompts and validation fingerprints into a shared error space for unseen-model routing (Jitkrittum et al., 2026). RouterBench, RouterEval, and LLMRouterBench broaden the benchmark and evaluation view (Hu et al., 2024; Huang et al., 2025b; Li et al., 2026), while Hybrid-LLM and RouterDC study cost-aware or representation-based routing (Ding et al., 2024; Chen et al., 2024b). These works ask which available model should answer a request. We study a related intervention: what changes when candidates are added to a frozen pool, and which decision event carries that change. VL-RouterBench supplies one cached correctness asset (Huang et al., 2025a); EmbedLLM and Model-SAT illustrate complementary approaches to representing model capability and routing context (Zhuang et al., 2024; Zhang et al., 2025).

Degradation under model updates. Negative flips are cases where a new prediction is wrong while the old prediction was correct (Yan et al., 2021). Adapter-based work likewise preserves an old model on instances where a new model would regress (Echterhoff et al., 2024); related model-update and model-combination work studies compatibility or aggregate gains rather than frozen-pool onboarding (Wortsman et al., 2022). Those settings modify or retrain one model. Our experts remain frozen, and the expanded decision can choose between multiple candidates. We analyze the resulting coverage–switching interaction.

Risk control and mixtures of experts. Selective prediction and risk-controlled decision rules report accuracy together with abstention or error risk. Ensemble methods and sparse mixture-of-experts models study aggregate prediction or jointly trained routing (Dietterich, 2000; Krogh & Vedelsby, 1995; Caruana et al., 2004; Jacobs et al., 1991; Jordan & Jacobs, 1994; Eigen et al., 2013; Wang et al., 2020; Lepikhin et al., 2021; Du et al., 2022; Zhou et al., 2022; Chowdhury et al., 2023; Shazeer et al., 2017; Fedus et al., 2022). We instead keep the experts and router frozen during onboarding. The gates in the appendix are diagnostic baselines used to test whether the identified effects support an operational rule, in the spirit of selective prediction and risk-controlled decision rules (Geifman & El-Yaniv, 2017; 2019; Romano et al., 2020; Angelopoulos et al., 2022).

Broader evaluation efforts emphasize reporting model behavior across tasks and evaluators rather than collapsing it to one score (Hendrycks et al., 2021; Liang et al., 2022; Zheng et al., 2023). Existing routing studies mainly evaluate which model should answer a request; we study how onboarding changes the served request set and the candidate identity jointly.

Organization of the Appendices

The appendices provide details of the protocol, decomposition, robustness analyses, and additional results. Appendix A describes the protocol and policies; Appendix B derives the ledger; Appendices C–D report interaction and reliability analyses; Appendices E–H give historical, held-out, mechanism, and aggregation analyses. Complete row-level logs and run outputs are supplied with the supplementary files.

B Full Policies and Budget

We use the following replay protocol. Each intervention keeps the incumbent pool, candidate outputs, score, and evaluation rows fixed; only the adoption and candidate-choice rules vary.

For request x, let E be the intervention's old pool, $C = \{c_1, c_2\}$ the two candidates, $s_R(x, \cdot)$ the frozen router score, and $e_R(x) = \arg\max_{e \in E} s_R(x, e)$ the incumbent with ties broken by the deterministic model order. Correctness is $Y_e(x) \in \{0, 1\}$. What is fixed within a matrix is the expert *registry* the score asset covers, not the pool: each intervention draws its own size-6 pool E from that registry, so E and the incumbent vary across interventions. Both policies compared on one intervention always share that intervention's E and $e_R(\cdot)$, so every Δ in this paper is within-intervention.

Rule	Candidate selected	Gate, and the action when it is not met
Pair-fixed	one candidate per candidate pair, by mean support improvement	task-profile threshold on that candidate; below it, the incumbent
Per-request	the higher frozen score on the request	task-profile threshold on the selected candidate; below it, the incumbent
Full margin gate	the higher frozen score on the request	score margin over the incumbent $> \tau$, read on the adopted-only and switch groups; outside those groups, and where the gate is shut, the pair-fixed action
Restricted two-threshold family	the higher frozen score on the request	one threshold per group on the adopted-only and switch groups; outside those groups, and where a gate is shut, the pair-fixed action
Support gate	the higher frozen score on the request	the intervention is accepted only if the 32-row support signal fires; on rejected interventions, the pair-fixed action
Harm-matched random mix	the per-request or pair-fixed choice, by a per-intervention coin flip	none; inherits the fallback of the endpoint it selects

Table 2: **Unified action rules.** The two endpoints fall back to the incumbent; all three gated families fall back to the pair-fixed action. The full margin gate and the restricted two-threshold family share both their domain (the adopted-only and switch groups) and their signal; they differ only in threshold count, one shared threshold against one per group. The threshold grid is $\{-10^9, 0, 0.005, \ldots, 0.12\}$; the acquisition budget is recorded below.

Field	Value
Matrix S	VL-RouterBench action-dev pool, official correctness matrix
Matrix U	VLM exec-routerbench expanded pool
Old pool	expert registry fixed within a matrix; per-intervention size-6 draw E
Candidates	2 per intervention, drawn from different model families
Interventions	2,000 per matrix, 4,000 total
Candidate pairs	175 after aggregating repeated interventions (S: 128, U: 47)
Router	frozen pool-aware score, same interface in both matrices
Support profile	32 requests on each candidate (64 calls per intervention)
Threshold-tuning split	`target_policy`, disjoint from evaluation; 24,926,000 rows, 3,304–9,159 per intervention
Evaluation split	`target_certificate`, action-disjoint (held out); 24,968,000 rows total
Primary estimand	request-pooled $\Delta B, \Delta H, \Delta G$ on the evaluation split
Interval	95% clustered bootstrap, unit = `matrix:intervention_id`
Intervals cover	stability under the fixed cache; not new requests or families
API calls	0 (cached replay)

Table 3: **Protocol and budget.** Both policies acquire the same 64 candidate-profile calls and differ only in whether the choice is fixed globally or made per request. All reported utility is measured on the held-out evaluation split. U contributes 47 candidate pairs, so U-specific intervals are wider; interval precision is the relevant quantity, not a fixed pair-count threshold.

Pair-fixed policy. A single candidate c^* is chosen for the candidate pair by mean support improvement, computed on the 32-request support profile of the two candidates. Repeated interventions that share a candidate pair therefore share c^*, but c^* is not fixed across different pairs or across matrices. Only c^* is then matched against the incumbent, and a task-profile threshold on c^* is tuned on the threshold-tuning split. If the threshold is met the policy returns c^*, otherwise it returns $e_R(x)$.

Per-request policy. On each request the proposing candidate is $\hat{c}(x) = \arg\max_{c \in C} s_R(x, c)$ with deterministic ties. A task-profile threshold is tuned for the selected candidate on the same split. If the threshold is met the policy returns $\hat{c}(x)$, otherwise it returns $e_R(x)$.

Full margin gate (falls back to the pair-fixed policy). The gate is read on the adopted-only and switch groups of Section 3; on every other request the pair-fixed action is kept, so the fallback is the pair-fixed policy and not the incumbent, exactly as in Table 2. Within those two groups the per-request candidate is adopted only when its frozen score margin over *the incumbent* exceeds τ; otherwise the pair-fixed action is kept. In this cache no adopt or switch request has a non-positive margin against the incumbent, so the bottom of the grid reproduces the per-request action on those two groups; it differs from the per-request policy only on abandoned-only requests (0.26% of requests), where the pair-fixed action is retained. By the fallback rule the $\tau \to +\infty$ member is the pair-fixed policy. The sweep grid used in this paper is $\{-10^9, 0, 0.005, 0.01, 0.015, 0.02, 0.03, 0.04, 0.06, 0.08, 0.12\}$.

Restricted two-threshold family (falls back to the pair-fixed policy). Behaviour is changed only on the adopted-only and switch groups defined in Section 3; on every other request the pair-fixed action is kept, and abandoned-only requests are never gated. The family carries two thresholds, one applied to the adopted-only group and one to the switch group, and reads the same frozen scores. By its fallback rule its $\tau \to +\infty$ member is exactly the pair-fixed policy. It shares the full margin gate's domain (the adopted-only and switch groups), score (the frozen score margin over the incumbent) and fallback (the pair-fixed action); the two are distinguished only by threshold count, the full margin gate being the diagonal of this family with one threshold applied to both groups. The two are therefore never combined in a comparison, and a two-threshold search can only enlarge the reachable set of the margin gate, which is why the gain it reports over the gate is a search gain rather than a new mechanism.

Support gate (falls back to the pair-fixed policy). The per-request policy is used only on interventions accepted by the 32-row support signal (10.325% of interventions, i.e. 413 of $4{,}000$); all other interventions keep the pair-fixed action. It shares the restricted family's fallback convention.

Budget. Both endpoint policies profile both candidates on the same 32 support requests, for 64 candidate calls per intervention. Threshold selection uses the labels of the threshold-tuning split only. All reported utility is computed on the held-out evaluation split. No live API calls are made anywhere in this paper; every number replays cached outputs.

Matrix provenance. Matrix S is the VL-RouterBench action-dev pool with the official per-request correctness matrix and a frozen pool-aware router score. Matrix U is the VLM exec-routerbench expanded pool with the same score interface. Repeated interventions are aggregated before candidate-pair statistics are computed, leaving 175 pairs (S: 128, U: 47).

C Decomposition Proof and Unit Checks

The decomposition can be checked directly from the request-level action record. For each request we record whether each policy adopts a candidate, which candidate it names, and whether the returned answer is correct. The coverage and switching events partition every request on which the policies return different models, and the beneficial-minus-harmful identity reconstructs the net gain.

Request-wise identity. Let A_1, J_1 and A_2, J_2 be the adoption indicator and the adopted candidate of the pair-fixed and the per-request policy, and let Y denote correctness of the returned model. On

Term	Meaning
Coverage	the two policies adopt on different request sets ($A_1 \neq A_2$); the net of adopted-only and abandoned-only requests
Switch	both policies adopt but name a different candidate ($A_1 = A_2 = 1$, $J_1 \neq J_2$)
Pair-fixed	one candidate chosen for the candidate pair by mean support improvement
Per-request	the higher-scoring candidate chosen per request
B, H, G	correction mass, harm mass and net gain; Δ is always per-request minus pair-fixed, so $\Delta G = \Delta B - \Delta H$ row by row (Theorem 1)
a_s, m_s	source-side choice skill (accuracy on support disagreements) and disagreement capacity (rate of informative candidate disagreement), on the 32-row profile
a_d, m_d	deployment-side choice skill $a_d = \mathbb{P}(Y_{J_2} = 1 \mid S \cap D_{\text{corr}})$ and switch mass $m_d = \mathbb{P}(S \cap D_{\text{corr}})$: the part of the raw switch rate $\mathbb{P}(S)$ on which a choice can change the outcome. $D_{\text{corr}} \subseteq S$ is the correctness-disagreement set; D_{pair} is the different, adoption-and-proposal domain the pairing shuffle permutes within
m_H, a_H	harm-changeable mass and choice share of the switch row in Appendix D.4: m_H is the share of evaluation requests on which both policies adopt different candidates, the incumbent is right and the two names disagree in correctness, and a_H the share of those on which the per-request name is the right one. a_H is not a_d: different conditional events
Full margin gate	falls back to the pair-fixed policy; its bottom grid point is the per-request action on the adopted-only and switch groups, and $\tau \to +\infty$ is the pair-fixed policy
Support gate, restricted two-threshold family	both fall back to the pair-fixed policy

Table 4: **Definitions used in the ledger and sensitivity analyses.**

Source of the change	Net correction ΔB	Harm mass ΔH	Net gain ΔG
Coverage (adoption set changes)	$+0.347$	$+0.219$	$+0.128$
Switch (both adopt, different model)	$+0.036$	-0.018	$+0.054$
Total	$+0.383$	$+0.201$	$+0.182$

Table 5: **One ledger, one convention.** Pooled per-request-minus-pair-fixed changes in percentage points, aggregated as $\sum$events/$\sum$requests over the held-out evaluation split. Every row is a $\Delta B/\Delta H/\Delta G$ triple, so the middle column is never a gain-side quantity. In particular the switch row's ΔB is $+0.036$ pp, the net correction change; the switch group's total positive movement ($+0.139$ pp, new corrections plus avoided harm) is a different quantity and is not ΔB.

Event class	Condition	Magnitude (pp)	Sign in ΔG
`mo_benefit`	only per-request adopts, needs a fix	0.3690	+
`mo_harm`	only per-request adopts, breaks it	0.2397	−
`so_lost_benefit`	only pair-fixed adopts, needs a fix	0.0219	−
`so_avoided_harm`	only pair-fixed adopts, breaks it	0.0211	+
`sw_new_correction`	switched, incumbent wrong, new choice right	0.0781	+
`sw_lost_correction`	switched, incumbent wrong, new choice wrong	0.0418	−
`sw_harm_reduction`	switched, incumbent right, pair-fixed wrong	0.0614	+
`sw_new_harm`	switched, incumbent right, per-request wrong	0.0438	−

Table 6: **Eight signed event classes.** Pooled magnitudes on the held-out evaluation split, aggregated as $\sum$events/$\sum$requests. The signed sum is $+0.1823$ pp, the total ΔG.

Weighting	Total ΔG (pp)	Total ΔH (pp)	Per-matrix ΔG (pp)
Request-pooled (main text)	$+0.182$	$+0.201$	S $+0.222$, U $+0.073$
Matrix-equal	$+0.147$	$+0.161$	mean of the two matrices
Intervention-equal	$+0.147$	$+0.161$	mean over 4,000 interventions

Table 7: **Aggregation sensitivity of the core ledger.** Evaluation-split request counts are 18,342,000 in S and 6,626,000 in U. Every column is a per-request-minus-pair-fixed change; the corresponding harm *levels* for the per-request policy are 1.328 pp pooled and 1.191 pp equal-weighted, and are not what the table reports. The three weightings are not interchangeable — they differ by 0.035 pp on total ΔG and 0.040 pp on total ΔH — so the weighting is stated rather than assumed. Request pooling is used for every table and figure in the main text.

every request the signed difference of the returned-model correctness, $Y_2 - Y_1$, decomposes over the two disjoint events $\{A_1 \neq A_2\}$ and $\{A_1 = A_2 = 1,\ J_1 \neq J_2\}$:

$$\underbrace{A_2(1-A_1)(Y_{J_2} - Y_{e_R}) - A_1(1-A_2)(Y_{J_1} - Y_{e_R})}_{\text{coverage}} + \underbrace{A_1 A_2 (Y_{J_2} - Y_{J_1})}_{\text{switch}}. \tag{1}$$

The remaining requests satisfy $A_1 = A_2$ and either $J_1 = J_2$ or no adoption, so they contribute zero. Expanding (1) by incumbent correctness and by which candidate is correct gives the eight signed event classes in Table 6.

Group masses. The three routing groups used throughout the paper are defined by the policies' actions only and never by correctness labels: adopted-only ($A_1 = 0, A_2 = 1$) 3.103%, switched ($A_1 = A_2 = 1, J_1 \neq J_2$) 1.223%, and abandoned-only ($A_1 = 1, A_2 = 0$) 0.264% of pooled requests. Coverage combines the first and third groups.

Unit checks. The request-level identity reconstructs every aggregate in Table 5.

Aggregation sensitivity. The main text pools by request, that is $\sum$(field $\times$ n)/$\sum n$ over the evaluation split of both matrices. Two other weightings are available, and they are reported here so that the choice is explicit rather than implied.

On this cache the matrix-equal and intervention-equal weightings coincide to four decimals, because the two matrices contribute the same number of interventions (2,000 each). The pooled and equal weightings differ measurably, which is why the convention is fixed once in Section 2 and used throughout instead of being left implicit.

D Controlled Interaction, Thresholds, and Mechanism

We evaluate the two adoption rules with the two candidate-choice rules over the pre-specified threshold grid. This section reports the matched-risk comparison, threshold sensitivity, interaction decomposition, pairing test, coverage profile, and routing-quality sweep.

Calibration rule and degeneracy. Under the maximum-fit-gain rule, all three threshold families select the no-gating end of the grid (fit-split gain 1.2815 pp, harm 1.3429 pp, adoption 22.45%). We therefore compare the families at matched harm, as in the main analysis.

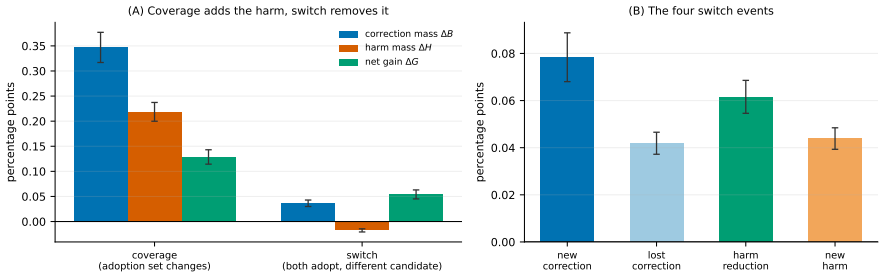

Figure 4: **The risk source of the same change.** (A) Coverage versus switch contributions to net correction ΔB, harm mass ΔH, and net gain ΔG, in percentage points, with 95% clustered bootstrap intervals. Coverage adds harm; switch removes it. (B) The four switch events, signed by their contribution to the ledger. Panels share one denominator and one weighting ($\sum$ events/ $\sum$ requests).

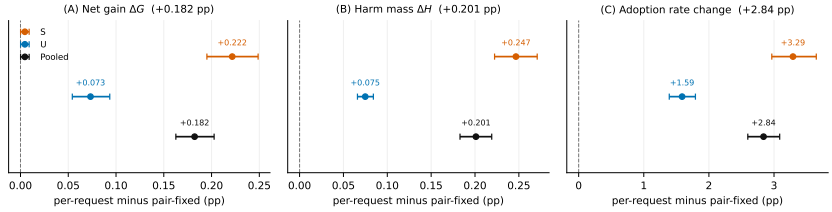

Figure 5: **Paired decision-space expansion at equal profile budget.** Per-request-minus-pair-fixed differences in mean gain, harm mass, and adoption for each matrix and pooled, with 95% clustered bootstrap intervals on the intervention cluster. Gain and harm differences come from the same accounting artifact as Table 5; adoption is bootstrapped on the same cluster unit. Both matrices gain and both matrices add harm.

Reachable frontiers. Over the grid, the reachable harm ranges and maximum gains are: restricted single-threshold margin gate $[1.1537, 1.3488]$ with maximum gain 1.1908 pp; restricted two-threshold same-signal $[1.1477, 1.3545]$ with 1.1910 pp; and restricted decomposed two-threshold $[1.1476, 1.3592]$ with 1.1908 pp. At the support gate's harm level (1.2356 pp), the three families attain 1.1778, 1.1780, and 1.1778 pp, respectively, by interpolation. The frontiers therefore differ by at most 0.0002 pp at this operating point.

Contrasts at matched risk. At the support gate's harm level, the decomposed and single-threshold gates have nearly identical point-estimate gain. The threshold family therefore does not improve the gain–harm frontier at this operating point. The conditional intervals in the last two rows should be interpreted as descriptions of the reachable draws only.

Across the grid, coverage harm remains positive and switching harm remains negative. Coverage ΔH decreases from $+0.240$ to $+0.039$ pp, while switching ΔH changes from -0.0176 to -0.0119 pp. Total ΔG decreases from $+0.183$ to $+0.112$ pp, and the coverage share of the gain decreases from 70.6% to 63.7%.

The matrix-level pattern is similar but less precise on U. On S, coverage ΔH remains within $[+0.052, +0.294]$ pp across the grid. On U, it decreases from $+0.090$ to $+0.003$ pp; at $\tau = 0.12$

Matched target	Contrast	Estimate (pp)	95% interval	Reachable
Support gate harm (1.2356)	decomposed − margin	-0.0001	$[-0.0022, +0.0040]$	99.98%
Support gate harm (1.2356)	same-signal − margin	$+0.0000$	$[-0.0003, +0.0047]$	99.98%
Margin max-gain harm (1.3488)	decomposed − margin	$+0.0000$	$[-0.0000, +0.0000]$	50.2%
Pair-fixed harm (1.1267)	decomposed − margin	undefined	$[+0.0032, +0.0133]$	9.4%

Table 8: **Matched-risk contrasts.** The last column gives the fraction of bootstrap draws for which the target harm lies within the family's reachable range. Intervals for the first two rows are unconditional. Intervals for unreachable targets use only draws that reach the target.

τ	Coverage ΔH	Switch ΔH	Total ΔH	Total ΔG	Δadoption	Coverage share of ΔG
-10^9	$+0.240$	-0.0176	$+0.222$	$+0.183$	$+3.10$	70.6%
0	$+0.240$	-0.0176	$+0.222$	$+0.183$	$+3.10$	70.6%
0.005	$+0.219$	-0.0178	$+0.201$	$+0.183$	$+2.79$	70.5%
0.01	$+0.200$	-0.0176	$+0.182$	$+0.182$	$+2.51$	70.6%
0.015	$+0.184$	-0.0179	$+0.166$	$+0.182$	$+2.27$	70.3%
0.02	$+0.171$	-0.0174	$+0.153$	$+0.178$	$+2.06$	70.0%
0.03	$+0.142$	-0.0173	$+0.125$	$+0.174$	$+1.69$	69.3%
0.04	$+0.122$	-0.0174	$+0.104$	$+0.169$	$+1.43$	68.7%
0.06	$+0.093$	-0.0158	$+0.077$	$+0.153$	$+1.04$	66.7%
0.08	$+0.063$	-0.0143	$+0.048$	$+0.143$	$+0.75$	66.2%
0.12	$+0.039$	-0.0119	$+0.027$	$+0.112$	$+0.46$	63.7%

Table 9: **Ledger across the margin-threshold grid.** Values are request-pooled percentage points. Δadoption is the gate's change relative to the pair-fixed policy. Total ΔH is the sum of the coverage and switching rows. Only τ changes across rows; the candidate pairs, scores, and data remain fixed. The grid ends at $\tau = 0.12$, where adoption is still 0.46 pp above the pair-fixed policy.

Scope	Grid points	Coverage ΔH excludes 0	Switch ΔH excludes 0	Switch ΔH
Pooled	11	11	11	negative at all 11
S	11	11	11	negative at all 11
U	11	11	1	negative at all 11

Table 10: **Sign stability over the threshold grid.** A grid point excludes zero when its clustered interval lies strictly on one side of zero. Coverage excludes zero at all points in the pooled data and on S. On U, the coverage interval narrows as the gate controls fewer requests. The switching estimate remains negative throughout, but its interval excludes zero at only one point on U.

its clustered interval is $[+0.001, +0.005]$. Switching ΔH is negative at every grid point in both matrices, although its interval separates from zero at only one point on U.

D.1 Threshold-grid stability and the switch-mass bound

We compute 95% intervals by resampling candidate pairs within each matrix and re-pooling the resulting draws. Each interval uses $2{,}000$ bootstrap repetitions with seed 20260917.

The three highest U coverage points, where the interval narrows with the controlled mass, are:

Scope	τ	Coverage ΔH	Gate adoption
U	0.06	$+0.0209$ $[+0.0090, +0.0359]$	0.3730
U	0.08	$+0.0106$ $[+0.0042, +0.0182]$	0.2347
U	0.12	$+0.0028$ $[+0.0010, +0.0050]$	0.0890

At $\tau = 0.12$, the gate controls 0.0889 pp of requests on U, compared with 1.7378 pp at the bottom of the grid. The interval then spans only 0.004 pp, so the estimate becomes less informative as the controlled set shrinks.

Per-pair sign test. We recompute the coverage and switching rows separately for each candidate pair. Coverage harm is positive for 137 of the 175 pairs and negative for 27; the negative pairs all occur on S. Switching harm has a roughly balanced pair-level sign pattern, while its pooled estimate is negative because the harm-reducing pairs have larger magnitudes.

The sign test uses the pair's net harm difference, which can be positive or negative.

The switch-mass bound. Every switched request is either *movable*, meaning that the two candidates differ in correctness, or *neutral*, meaning that both candidates produce the same outcome. Let m_d denote the movable mass. We split it according to which policy names the correct candidate: m_d^+ is the mass where the per-request policy is correct, and m_d^- is the mass where the pair-fixed policy is correct.

Scope	Row	+	−	0	Sign test p	Pair-equal mean	Req-pooled
Pooled	Coverage ΔH	137	27	11	6.4×10^{-19}	$+0.216$	$+0.219$
Pooled	Switch ΔH	58	56	61	0.93	-0.017	-0.018
S	Coverage ΔH	96	27	5	2.9×10^{-10}	$+0.266$	$+0.268$
S	Switch ΔH	45	43	40	0.92	-0.021	-0.022
U	Coverage ΔH	41	0	6	9.1×10^{-13}	$+0.081$	$+0.081$
U	Switch ΔH	13	13	21	1.00	-0.006	-0.006

Table 11: **Pair-level signs of the harm differences.** The sign test is two-sided and uses the nonzero pairs. Pair-equal means weight candidate pairs equally; request-pooled values weight requests according to the evaluation data.

Scope	$P(S)$	m_d	m_d interval	m_d^+	m_d^-	Realized ΔG
Pooled	1.223	0.225	$[0.199, 0.252]$	$+0.140$	-0.086	$+0.054$
S	1.491	0.285	$[0.249, 0.322]$	$+0.177$	-0.107	$+0.070$
U	0.478	0.060	$[0.049, 0.072]$	$+0.035$	-0.025	$+0.010$

Table 12: **Movable switching mass and gain ceilings.** Values are request-pooled percentage points. $P(S)$ is the rate at which the two policies name different candidates. m_d is the subset of those requests on which the candidates differ in correctness. m_d^+ and m_d^- give the positive and negative gain ceilings for re-selection within the fixed switch set.

For any rule that changes behavior only on switch requests and chooses one of the two endpoint actions on each such request,

$$\Delta G_{\text{switch}} \le m_d^+, \qquad \Delta G_{\text{switch}} \ge -m_d^-, \qquad \Delta H_{\text{switch}} \le h^+, \qquad \Delta H_{\text{switch}} \ge -h^-. \tag{2}$$

The upper and lower bounds are attained by always naming the correct or incorrect candidate, respectively. The looser bound $|\Delta G_{\text{switch}}| \le m_d$ also holds, but it is not tight because keeping the pair-fixed action on the m_d^- requests loses nothing relative to that endpoint.

The realized switch row captures 39% of the pooled gain ceiling: $+0.054$ of a possible $+0.140$ pp. The neutral part of the switch group is $P(S) - m_d = 0.997$ pp, which cannot be changed by choosing between candidates. The remaining 0.086 pp is m_d^- and could be recovered by choosing the other candidate on those requests. Thus the bound concerns candidate choice within a fixed switch set; it does not apply to a rule that changes which requests enter that set. On U, the entire positive movable mass is only 0.035 pp, compared with the policy gain of 1.190 pp.

D.2 The controlled adoption-set × candidate-choice comparison

We combine the two adoption rules with the two candidate-choice rules to form four counterfactual policies. The adoption rule is taken from one endpoint policy and the named candidate from the other. The incumbent is returned whenever the selected adoption rule does not adopt. The diagonal cells are the two endpoint policies; the off-diagonal cells change only one decision at a time.

The primary intervention-level bootstrap gives

$$I_H \in [-0.286, -0.239],\ [-0.338, -0.275],\ [-0.162, -0.118]$$

and

$$I_G \in [+0.366, +0.436],\ [+0.425, +0.520],\ [+0.169, +0.232]$$

for the pooled data, S, and U, respectively. None of these intervals contains zero.

As a sensitivity analysis, we resample candidate pairs rather than individual interventions. This widens the intervals by about a factor of 3.7, but the interaction signs remain unchanged:

$$I_H \in [-0.352, -0.179],\ [-0.423, -0.198],\ [-0.239, -0.066]$$

and

$$I_G \in [+0.272, +0.536],\ [+0.309, +0.651],\ [+0.091, +0.345].$$

The only unit-sensitive contrast is the gain component of $Choice_{A_1}$ on U: its intervention-level interval is $[+0.0001, +0.012]$, whereas the pair-level interval is $[-0.015, +0.030]$.

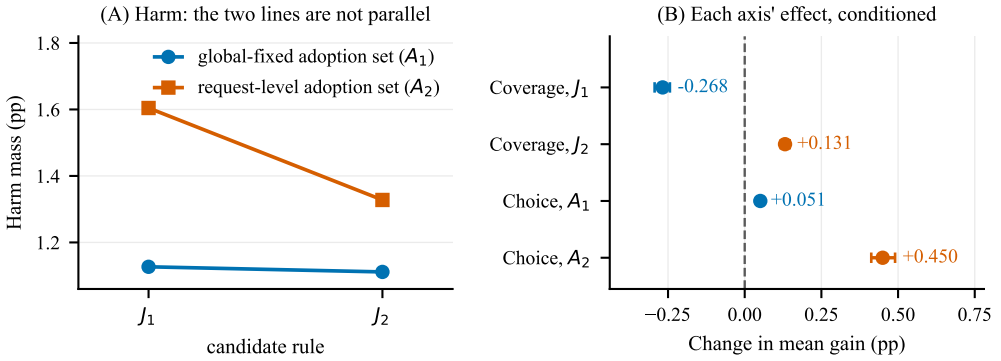

Figure 6: **Controlled 2×2 comparison.** Panel (A) shows the four policy cells as a function of the candidate-choice rule. Panel (B) shows the adoption and candidate-choice contrasts conditional on the other decision. Intervals are 95% clustered bootstrap intervals.

Scope	Cell	Gain (pp)	Harm (pp)	Adoption
Pooled	(A_1, J_1)	1.008	1.127	19.34%
	(A_1, J_2)	1.058	1.111	19.34%
	(A_2, J_1)	0.740	1.604	22.18%
	(A_2, J_2)	1.190	1.328	22.18%
S	(A_1, J_1)	0.916	1.236	18.67%
	(A_1, J_2)	0.983	1.217	18.67%
	(A_2, J_1)	0.599	1.808	21.97%
	(A_2, J_2)	1.137	1.483	21.97%
U	(A_1, J_1)	1.262	0.823	21.19%
	(A_1, J_2)	1.268	0.819	21.19%
	(A_2, J_1)	1.130	1.042	22.77%
	(A_2, J_2)	1.336	0.898	22.77%

Table 13: **The four counterfactual policy cells.** A_1, A_2 denote the pair-fixed and per-request adoption rules; J_1, J_2 denote the candidate named by the policy. Adoption is constant within each pair of rows because it is determined by A.

Scope	Coverage$_{J_1}$	Coverage$_{J_2}$	Choice$_{A_1}$	Choice$_{A_2}$	I_G	I_H
Pooled	−0.268	+0.131	+0.051	+0.450	+0.399	−0.261
S	−0.316	+0.155	+0.067	+0.538	+0.471	−0.305
U	−0.132	+0.068	+0.006	+0.206	+0.200	−0.139
harm contrasts, same columns						
Pooled	+0.478	+0.216	−0.015	−0.277	—	−0.261
S	+0.571	+0.266	−0.019	−0.325	—	−0.305
U	+0.218	+0.079	−0.004	−0.144	—	−0.139

Table 14: **First differences and interaction terms.** Coverage$_J$ is $A_2 - A_1$ with candidate J fixed. Choice$_A$ is $J_2 - J_1$ with adoption rule A fixed. The interaction is $I = x_{22} - x_{21} - x_{12} + x_{11}$.

Set	Share	Inc. acc.	$Y_{J_1} \neq Y_{J_2}$	g_1	g_2	h_1	h_2	to I_G	to I_H
C_{add}	3.103%	72.13%	23.21%	−0.267	+0.129	+0.499	+0.240	+0.3960	−0.2590
C_{drop}	0.264%	76.43%	6.03%	+0.001	−0.002	+0.021	+0.023	+0.0030	−0.0022
C_{common}	19.08%	75.90%	1.18%	+1.007	+1.061	+1.106	+1.088	0	0
C_{none}	77.56%	80.36%	14.72%	−12.627	−17.639	+16.522	+21.184	0	0
All requests	100%	79.24%	12.38%	—	—	—	—	+0.3990	−0.2612

Table 15: **Requests contributing to the interaction.** Groups are defined by the two adoption actions. "Inc. acc." is incumbent accuracy within the group, and $Y_{J_1} \neq Y_{J_2}$ is the rate at which the candidates differ in correctness. The last two columns give each group's contribution to I_G and I_H on the all-request denominator. Groups with $A_1 = A_2$ contribute zero by definition.

Scope	Set	Share	Inc. acc.	$Y_{J_1} \neq Y_{J_2}$	g_1	g_2	h_1	h_2	to I_G / to I_H
S	C_{add}	3.597%	71.50%	24.09%	−0.316	+0.153	+0.597	+0.294	+0.4683 / −0.3030
S	C_{drop}	0.306%	77.51%	6.07%	+0.001	−0.002	+0.026	+0.028	+0.0027 / −0.0022
S	C_{common}	18.37%	74.80%	1.55%	+0.915	+0.985	+1.211	+1.189	0 / 0
S	C_{none}	77.73%	79.34%	18.59%	−12.313	−19.198	+16.713	+23.114	0 / 0
U	C_{add}	1.738%	75.80%	18.15%	−0.131	+0.065	+0.227	+0.090	+0.1958 / −0.1373
U	C_{drop}	0.149%	70.26%	5.79%	+0.001	−0.003	+0.008	+0.011	+0.0041 / −0.0021
U	C_{common}	21.04%	78.55%	0.29%	+1.261	+1.271	+0.815	+0.809	0 / 0
U	C_{none}	77.08%	83.21%	3.93%	−13.499	−13.325	+15.996	+15.841	0 / 0

Table 16: **Interaction contributions by matrix.** The columns match Table 15. The C_{add} group contributes 99.4% of I_G and 99.3% of I_H on S, and 98.0% and 98.5% on U. The only unit-sensitive quantity is the C_{drop} gain contribution on U.

D.3 Which requests carry the interaction

We partition the evaluation rows according to the pair of adoption actions:

$$C_{\text{add}} = \{A_1 = 0, A_2 = 1\}, \quad C_{\text{drop}} = \{A_1 = 1, A_2 = 0\},$$
$$C_{\text{common}} = \{A_1 = A_2 = 1\}, \quad C_{\text{none}} = \{A_1 = A_2 = 0\}.$$

These four sets partition the evaluation split. Correctness labels are used to compute gains and harm, not to define the groups.

Substituting the group contributions into the four first differences recovers the values in Table 14. For example,

$$g_1(C_{\text{add}}) - g_1(C_{\text{drop}}) = -0.2676$$

gives the coverage contrast at J_1, while

$$(g_2 - g_1)(C_{\text{common}}) + (g_2 - g_1)(C_{\text{add}}) = +0.4499$$

gives the choice contrast under A_2. The drop group enters the choice contrasts with a negative sign because naming the per-request candidate on requests that the per-request policy abandons reduces both gain and harm.

Inside the newly adopted requests. Within C_{add}, the candidates differ in correctness on 23.21% of requests: J_2 alone is correct on 17.99%, whereas J_1 alone is correct on 5.23%. Among requests where the incumbent is correct, J_1 is wrong on 22.28% and J_2 on 10.70%. Averaged over the group, J_1 has accuracy 63.54%, compared with 72.14% for the incumbent, while J_2 has accuracy 76.30%. Thus the candidate named by the pair-fixed rule is worse than the incumbent on the newly adopted requests, whereas the per-request candidate is better.

(Y_e, Y_{J_1}, Y_{J_2})	000	001	010	011	100	101	110	111
Share of C_{add}	13.75%	6.64%	2.22%	5.25%	4.72%	11.35%	3.00%	53.06%

Table 17: **Outcome states within C_{add}.** The group contains requests on which the pair-fixed policy does not adopt and the per-request policy does. Y_e is the incumbent outcome, and Y_{J_1}, Y_{J_2} are the candidate outcomes.

Scope	Realized	Shuffle null	Count-pres. oracle	Label oracle	p
gain (pp)					
Pooled	$+1.190$	$+1.149$	$+1.391$	$+1.401$	0.0005
S	$+1.137$	$+1.096$	$+1.372$	$+1.380$	0.0005
U	$+1.336$	$+1.294$	$+1.444$	$+1.461$	0.0005
harm (pp)					
Pooled	$+1.328$	$+1.348$	$+1.226$	$+1.222$	0.0005
S	$+1.483$	$+1.502$	$+1.365$	$+1.362$	0.0005
U	$+0.898$	$+0.921$	$+0.841$	$+0.833$	0.0005

Table 18: **Request-to-candidate pairing test.** The shuffle preserves, within each intervention, the number of requests assigned to each candidate and randomizes their assignment within D_{pair}. The count-preserving oracle keeps these counts fixed and chooses the assignment using the observed outcomes. The label oracle may also change the counts and is therefore a looser upper bound.

The four group contributions and four policy cells are resampled jointly, so the algebraic reconstruction holds within each bootstrap draw. Under intervention-level clustering, C_{add} contributes $+0.3960$ $[+0.3628, +0.4333]$ to I_G and -0.2590 $[-0.2837, -0.2373]$ to I_H. The corresponding C_{drop} contributions are $+0.0030$ $[+0.0024, +0.0037]$ and -0.0022 $[-0.0026, -0.0017]$. Pair-level clustering widens these intervals by about 3.7; the reported signs remain unchanged except for the C_{drop} gain contribution on U.

D.4 Is the request-to-candidate pairing informative?

The switching leg and the off-diagonal cells of the 2×2 depend on whether the frozen score assigns candidates to requests in an informative way. We test this by holding the adoption set, candidate usage counts, scores, and evaluation split fixed while randomizing the request-to-candidate assignment.

Within an intervention, the pair-fixed policy names one candidate throughout. The shuffle therefore acts on

$$D_{\text{pair}} = \{\text{requests adopted by the per-request policy on which both candidates propose}\}.$$

This is the set on which the policy has a genuine candidate choice. The shuffle preserves the number of requests assigned to each candidate and changes only which requests receive each candidate. Requests outside D_{pair} remain unchanged.

The randomization null treats all assignments with the observed candidate counts as equally likely. For each draw, we recompute ΔG and ΔH while keeping the domain, candidate counts, scores, and evaluation split fixed. With 2,000 draws, the one-sided Monte Carlo p-value is

$$p_{\text{MC}} = \frac{1 + \#\{\text{draws at least as extreme as the observed value}\}}{1 + 2{,}000}.$$

No draw reached the observed value, so the smallest resolvable p-value is $1/2{,}001 \approx 0.0005$.

The observed pairing outperforms the shuffle on both axes and on both matrices. Relative to the shuffle, it adds 0.041 pp of gain and removes 0.020 pp of harm in the pooled data (0.041/0.019 on S and 0.041/0.023 on U). The observed value is more extreme than all $2{,}000$ draws, giving $p_{\text{MC}} = 1/2{,}001 \approx 0.0005$.

The oracle columns quantify the remaining value of re-pairing candidates within the realized adoption domain. Pooled, outcome-aware re-pairing would add another 0.211 pp of gain and remove 0.106 pp of harm; the corresponding values are 0.242/0.121 on S and 0.126/0.065 on U. The frozen score therefore captures 16.3% of the available pairing gain and 16.0% of the available harm reduction in the pooled data. Holding candidate usage counts fixed gives 17.0% of the pooled gain ceiling. These oracles concern candidate assignment within the realized domain; they do not evaluate new request features or a different adoption rule.

The mass bound above addresses how much movable switching mass is available. The pairing test addresses whether the score chooses well within the requests where a choice is available. The two quantities describe different limits on the switching contribution.

Group	Requests (%)	Adopted-only harm	Abandoned-only avoided	Coverage harm
MathVerse_MINI	9.47	+0.867	+0.017	+0.851
TextVQA_VAL	12.02	+0.170	+0.030	+0.140
DocVQA_VAL	12.86	+0.138	+0.011	+0.126
InfoVQA_VAL	6.74	+0.204	+0.010	+0.194
AI2D_TEST	7.43	+0.201	+0.026	+0.175
all 20 *datasets*	100	+0.240	+0.021	+0.219

Table 19: **Coverage harm by dataset.** Values are percentage points per request within each group. The five displayed rows are the largest contributors. The top three datasets contain 34.3% of requests and account for 52.0% of coverage harm. Coverage harm is positive in all 20 datasets.

Rule	Gain (pp)	Harm (pp)	Adoption	Fallback
Pair-fixed	1.008	1.127	19.34%	incumbent
Per-request	1.190	1.328	22.18%	incumbent
Full margin gate, $\tau = 0.01$	1.190	1.309	21.85%	pair-fixed
Support gate	1.130	1.236	20.86%	pair-fixed
Harm-matched mix (p^*=0.542)	1.106	1.236	20.90%	pair-fixed

Table 20: **Policy comparison.** Pooled point estimates on the held-out evaluation split. The harm-matched mix is interpolated between the two endpoint policies.

D.5 Where the coverage harm sits

We stratify the two coverage events by dataset, request length, and score margin using the same two endpoint policies. Adopted-only harm and abandoned-only avoided harm are disjoint because the first requires adoption by the per-request policy and the second requires non-adoption.

Group	Requests (%)	Coverage harm	Share of total harm
Length $\leq$ 60 chars	41.04	+0.165	30.9%
Length 61–140 chars	26.15	+0.236	28.2%
Length 141–320 chars	14.96	+0.435	29.7%
Length $>$ 320 chars	17.85	+0.137	11.2%
Margin $\leq$ 0.05	12.94	+1.028	60.8%
Margin 0.05–0.15	7.36	+1.128	38.0%
Margin 0.15–0.35	1.72	+1.280	10.1%
Margin $>$ 0.35	0.16	+1.011	0.7%
Not adopted by the rule	77.82	−0.027	−9.6%

Length is a weak stratifier. The per-request rate reaches +0.435 pp in the 141–320-character band, but the four length bands contribute between 11% and 31% each. Margin is mainly a mass effect: the per-adoption harm rate is between 1.0 and 1.3 pp across the margin bands, while the tightest band contains most adopted requests. The task groups show the largest variation in harm rate, although coverage harm remains positive in every dataset.

D.6 The controlled routing-quality sweep

We perturb the frozen score along a label-free ladder. For $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$,

$$s_\alpha = (1 - \alpha)s_{\text{frozen}} + \alpha u.$$

The candidate pairs, requests, profile budget, data splits, and threshold procedure remain fixed. Thresholds are re-tuned on the tuning split at each configuration, and the four policy cells and their coverage and switching contributions are recomputed.

Perturbation construction. For each (request row, expert) pair, u is drawn independently from $\text{Uniform}[0, 1)$ on the raw score scale. The draw is generated from a SHA-256 hash of (seed, matrix, expert, row index), making the perturbation reproducible. The frozen score lies in $[0, 1]$, with observed range $[0.0000, 0.9832]$ and mean 0.592 across the two matrices. The five seeds are 20260918–20260922. For a given seed, the same noise draw is used at all positive α values, so

α	ρ_R	ΔG	ΔH	coverage G/H	switch G/H	I_G	I_H
0.00	0.636	+0.182	+0.201	+0.128/+0.219	+0.054/−0.018	+0.399	−0.261
0.25	0.569	+0.167	+0.248	+0.146/+0.250	+0.021/−0.003	+0.347	−0.218
0.50	0.423	+0.262	+0.364	+0.270/+0.355	−0.007/+0.009	+0.269	−0.160
0.75	0.224	+0.376	+0.404	+0.401/+0.389	−0.025/+0.015	+0.167	−0.101
1.00	0.001	+0.286	+0.358	+0.323/+0.340	−0.037/+0.018	+0.114	−0.091

Table 21: **Routing-quality perturbation, pooled.** Values are percentage points at each point of the label-free score ladder. The interaction retains the same direction at every scope-and-rung combination: $I_G > 0$ and $I_H < 0$. The switching contribution changes sign as the score becomes less informative: it removes harm at the first two positive rungs but adds harm from $\alpha = 0.5$ onward.

	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
pooled switch G	$0.021[0.017, 0.023]$	$-0.007[-0.010, -0.005]$	$-0.025[-0.028, -0.022]$	$-0.037[-0.040, -0.035]$
pooled switch H	$-0.003[-0.004, -0.001]$	$0.009[0.007, 0.010]$	$0.015[0.013, 0.017]$	$0.018[0.017, 0.020]$
S cov_{J_1} G	$-0.202[-0.217, -0.190]$	$0.105[0.099, 0.109]$	$0.429[0.416, 0.445]$	$0.497[0.470, 0.516]$
U cov_{J_1} G	$-0.143[-0.160, -0.110]$	$-0.132[-0.236, -0.059]$	$0.109[0.077, 0.143]$	$0.248[0.209, 0.299]$

Table 22: **Draw-to-draw variation at the sign-changing points.** Each cell gives the mean over five draws, followed by the smallest and largest draw. The frozen-score signs persist at $\alpha = 0.25$ in all five draws. The S coverage leg changes sign at $\alpha = 0.5$, while the U coverage leg changes sign at $\alpha = 0.75$.

those points form one correlated trajectory. The baseline $\alpha = 0$ is seed-independent; the remaining configurations contain five draws at each positive α.

Definition of ρ_R.

$$\rho_R = \frac{V_R - V_{\mathrm{rand}}}{V_{\mathrm{oracle}} - V_{\mathrm{rand}}}.$$

Here V_R, V_{rand}, and V_{oracle} are the accuracies of the incumbent, a uniformly random old-pool member, and a correctness-oracle old-pool member. They are computed on the development split after removing the intervention's 32 support rows. We first pool the accuracies,

$$V_\bullet = \frac{\sum_{\text{interventions}} \sum_{\text{rows}} \mathbb{1}\{Y_\bullet = 1\}}{\sum_{\text{interventions}} \sum_{\text{rows}} 1},$$

and then compute the ratio. Thus ρ_R is a ratio of pooled accuracies rather than an average of per-intervention ratios.

The per-request policy has higher net gain at every rung and in every draw, but both endpoint accuracies decline as the score is perturbed. At $\alpha = 1$, the larger net-gain difference is measured relative to a substantially less accurate incumbent: pooled incumbent accuracy falls from 79.24% to 59.84%, while per-request accuracy falls from 80.43% to 61.75%. The five ladder points therefore show the persistence of the interaction under this perturbation, not a monotone relationship between overall routing quality and endpoint performance.

	S (%)			U (%)			pooled (%)		
α	inc.	g.-f.	req.	inc.	g.-f.	req.	inc.	g.-f.	req.
0.00	78.22	79.13	79.36	82.09	83.35	83.42	79.24	80.25	80.43
0.25	76.43	77.40	77.62	79.28	80.38	80.41	77.18	78.19	78.36
0.50	71.93	73.47	73.78	75.15	76.51	76.65	72.78	74.28	74.54
0.75	65.26	67.36	67.76	70.50	71.95	72.26	66.65	68.58	68.96
1.00	57.51	59.30	59.58	66.28	67.45	67.73	59.84	61.46	61.75

Table 23: **Absolute accuracy along the score ladder.** Values are five-draw means in percent. "inc." is the incumbent, "g.-f." is the pair-fixed policy, and "req." is the per-request policy. The ordering req. > g.-f. > inc. holds at every rung, while all three levels decline as the score becomes less informative.

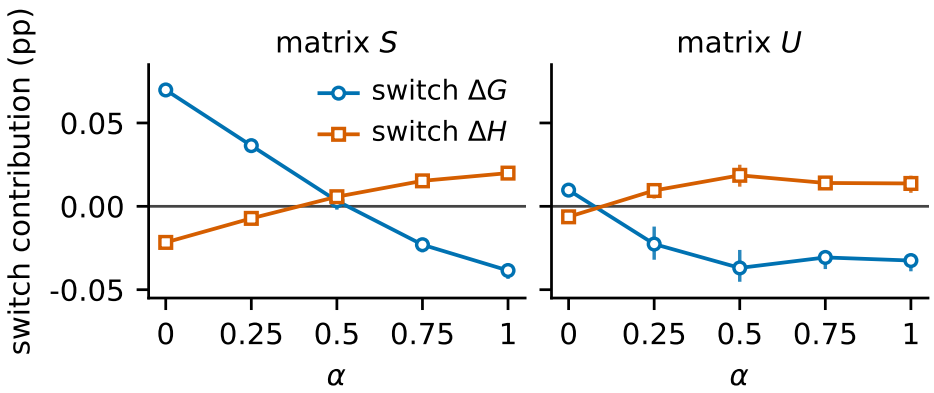

Figure 7: **Switching contribution along the score ladder.** Each point is the mean over five draws, and each bar spans the smallest to largest draw. The bars show draw-to-draw ranges rather than confidence intervals. The switching harm leg changes sign by $\alpha = 0.25$ on U and by $\alpha = 0.5$ on S; the switching gain leg on S changes sign at $\alpha = 0.75$.

E Reliability, Classifiers, and Oracles

The support profile contains 32 rows. We evaluate the stability of its estimates using split-half checks, larger support budgets, request-level prediction, and outcome oracles. We report split-half checks, support-budget sweeps, a request-level classifier probe, and outcome oracles. These analyses describe available signal and headroom; they are not additional policies.

E.1 Support-profile reliability: splits, missingness, imputation

We compute these quantities from the fixed support profile to assess which summaries are stable under split-half and support-budget changes. Undefined entries are reported rather than filled silently, and all quantities use cached assets.

The split is fixed, not resampled. The support profile is a single deterministic 32-row selection per matrix. The two halves used in Figure 8 are the first and the second 16 rows of that one ordering, so there is no random draw for a reader to re-draw and no seed to fix. The same fixed ordering supplies the 64-row prefix used only in the nested sweep that Section E.1 drops.

Undefined entries and their frequency. Source choice skill a_s is undefined whenever a support set contains no informative candidate disagreement, because the denominator of the statistic is then zero. We keep intervention-level and candidate-pair missingness separate. At the *intervention* level, over all 4,000 interventions, this happens for 46.6% of first halves and 52.3% of second halves, and for 39.0% of the full 32-row profiles and 32.5% of the 64-row prefixes. At the *candidate-pair* level, where repeated interventions are aggregated before the statistic is formed, a pair counts as undefined only when every intervention in it is: 139 of the 175 pairs are defined on each 16-row half, 127 on both, 151 on the full 32-row profile (13.7% undefined), and 168 on the 64-row prefix (4.0%). The joint drop from 175 to 127 is 48 pairs, or 27.4%. The 127 pairs defined on both halves are a subset of the 151 defined at the full 32 rows, with no violations, so the effective sample of the split-half check can never exceed the effective sample at the full budget. Capacity m_s has no such failure mode: it is defined on every prefix at every size and on all 175 pairs, with 0% missing.

Two conventions, two answers. Both conventions are applied here to one record, so that the difference below is a difference of rule and not of implementation.

- **Listwise deletion.** Undefined entries are dropped pairwise, so the sample size varies by statistic: 127 pairs for the a_s-versus-a_s split-half check and 175 for the m_s-versus-m_s check. This is the convention behind Figure 8.
- **Chance imputation.** Undefined skill is set to 0.5 and undefined capacity to 0, so all 175 pairs enter every correlation.

Rows per half n	ρ	95% interval	Pairs	$\varepsilon(n)$	Informative rows	Empty blocks
4	-0.127	$[-0.309, +0.105]$	70	0.296	0.29	77.3%
8	-0.025	$[-0.204, +0.174]$	107	0.295	0.58	64.8%
12	$+0.028$	$[-0.155, +0.207]$	119	0.281	0.84	57.9%
16	$+0.015$	$[-0.158, +0.199]$	127	0.273	1.16	53.0%
24	-0.096	$[-0.268, +0.080]$	136	0.230	1.78	44.2%
32	-0.010	$[-0.184, +0.163]$	138	0.227	2.31	41.3%

Table 24: **Split-half correlation of source choice skill a_s by support budget.** Intervals are pair-clustered bootstraps over 2,000 resamples and are marginal (pointwise), so they do not support a joint statement across the six budgets. Every interval covers zero. $\varepsilon(n)$ is the *block-difference variability* of a_s at that budget: the standard deviation of $a_i - a_j$ over disjoint block pairs, divided by $\sqrt{2}$. It is on the probability scale of a_s itself — a rate on $[0, 1]$, so 0.296 means 29.6 pp — and not on the correlation scale of ρ. Reading it as the standard error of a single block's estimate needs the further assumptions that blocks have equal variance, independent errors and no systematic difference between them; the text reports it as a descriptive spread and relies on none of the three. "Informative rows" is the mean number of rows per block on which at least one candidate proposes and the candidates disagree — the only rows a_s averages over.

Sensitivity to the convention. On the same record the split-half correlation of a_s moves from 0.015 $[-0.165, +0.198]$ ($n = 127$, deleted) to 0.144 $[+0.001, +0.290]$ ($n = 175$, chance-filled); both are near zero, so the instability of the 16-row estimate is a property of the statistic and not of the rule. The source-to-evaluation skill correlation moves from -0.220 $[-0.358, -0.073]$ ($n = 151$, deleted) to -0.182 $[-0.313, -0.055]$ ($n = 175$, chance-filled); the sign is stable and both intervals exclude zero. It is not explained away here. The two statistics are formed on different conditional events — the source one on the support rows a profile ordered, the evaluation one on evaluation rows — so a negative value is compatible both with estimation error and with a genuine difference between the two conditions, and this asset separates neither of those from the change of conditional set itself. The capacity correlations are positive under both rules and need no convention. The source-side skill statistic is defined on candidate correctness disagreement throughout.

At this support size, a_s is not stable enough to rank pairs reliably. The budget sweep below shows how that result changes with the measured support range.

E.2 The budget the source estimate would need

Section E.2 repeats the split-half check at other support budgets. The design is the same-size one: from the single deterministic 64-row ordering per matrix, take disjoint blocks of n consecutive rows and correlate a_s between two blocks. At $n = 16$ (block 0 versus block 1, rows 0–15 against rows 16–31) this reproduces the published split-half check exactly ($\rho = +0.0146$, absolute difference 0). The unit is the candidate pair, interventions within a pair are averaged before correlating, and a pair enters a correlation only where a_s is defined on both blocks — the same listwise convention as Appendix E.

The block-difference variability falls from 0.296 to 0.227 over an eightfold increase in profile rows (23%); on the scale of a_s that is 29.6 pp to 22.7 pp. It sits where the last two columns predict. Since a_s is a mean of 0/1 outcomes over a block's informative rows, a block whose rows were independent draws at its own rate would vary by $\sqrt{\bar{p}(1-\bar{p})/n_{\text{inf}}}$ — 0.226 at 32 rows against the measured 0.227, and above the measured value at every smaller budget — so the observed spread across blocks is compatible with sampling variation at one to two informative rows. It does not rule out block differences, heterogeneity or missingness selection; those remain uncontrolled here, as the table note says. The design moves the row count and the expected informative count together (a block of n rows contains nq of them, q the informative rate), so it cannot attribute the flatness to one rather than the other. Fitting $\varepsilon(n)^2 = a/n + b$ to these six points gives $a = 0.153$, $b = 0.057$ and a floor $\varepsilon_\infty = 0.238$ that exceeds the across-pair spread the estimate is meant to resolve (0.167); the curve without a floor would predict $\rho = 0.5$ by about six rows and $\rho(32) = 0.47$, which the measured -0.010 rules out. Both of those readings are *extrapolations* beyond the measured range, which ends at 64 profile rows; the fit is two parameters on six budgets, its floor is not tightly determined, and

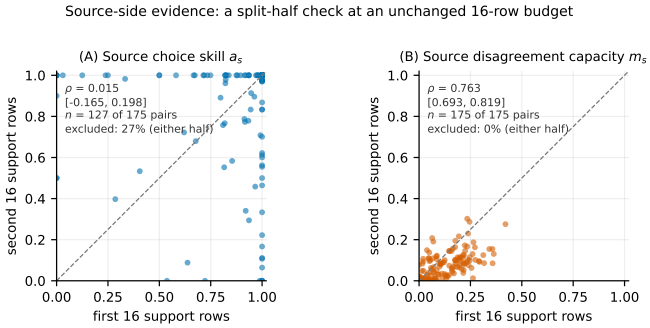

Figure 8: **Source-side evidence under a split-half check.** The fixed 32-row support profile split into its first and second 16 rows, aggregated to candidate pairs. (A) Source choice skill a_s does not repeat across the two halves at this budget ($\rho = 0.015$, $n = 127$; the 175 pairs less the 48 on which either half carries no informative correctness disagreement, 27.4%; a block carries 1.16 informative rows on average, 53% of blocks none). (B) Disagreement capacity m_s is stable at the identical budget ($\rho = 0.763$, $n = 175$, no undefined pairs). Both panels use the same 16 rows per half and the same fixed ordering, so the comparison is between statistics at equal sample size, not between independent draws.

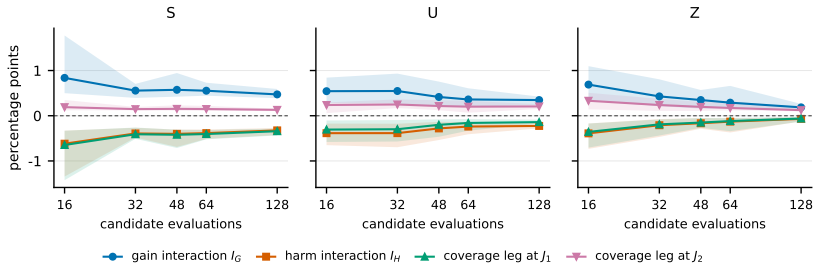

Figure 9: **Support-budget sweep.** The interaction is recomputed over ten nested support draws at each measured budget. Bands show the smallest and largest point estimates across draws, not confidence intervals.

Section E.2 reports the measured points only. The corresponding sample is 127–138 pairs, so the correlation is itself noisy.

Per-request classifiers. The full features, the hold-out grouping by candidate pair, the per-matrix AUROC tables, and the paired intervals behind the 0.768-versus-0.768 comparison quoted in Section E are in the following subsection.

E.3 Per-request classifier probe

The probe trains a logistic classifier on candidate-disagreement requests from the 32-row support split. This is a request-level selection probe, related to learned routing representations but evaluated here as a diagnostic rather than as a new router (Chen et al., 2024b; Zhuang et al., 2024; Zhang et al., 2025). support split. Five-fold GroupKFold holds out complete candidate pairs, so no pair contributes to both fit and evaluation, and evaluation uses at most 512 deterministically selected certificate disagreements per pair. The per-request target is which candidate is correct, so the valid baseline is the raw candidate-score difference on the same request-level target.

Pooled over both matrices, macro AUROC is 0.768 for the learned classifier and 0.768 for the raw score, with a paired interval on the difference of $[-0.0011, +0.0002]$. Per matrix the learned classifier reaches 0.776 against the score's 0.776 in S (difference -0.00001, $[-0.0002, +0.0002]$) and 0.746 against 0.748 in U (difference -0.0017, $[-0.0041, +0.0009]$); every interval covers zero, and the predeclared criterion of exceeding 0.813 in both matrices is not met. The tested classifier does not add a stable advantage over the score the router already uses on this target.

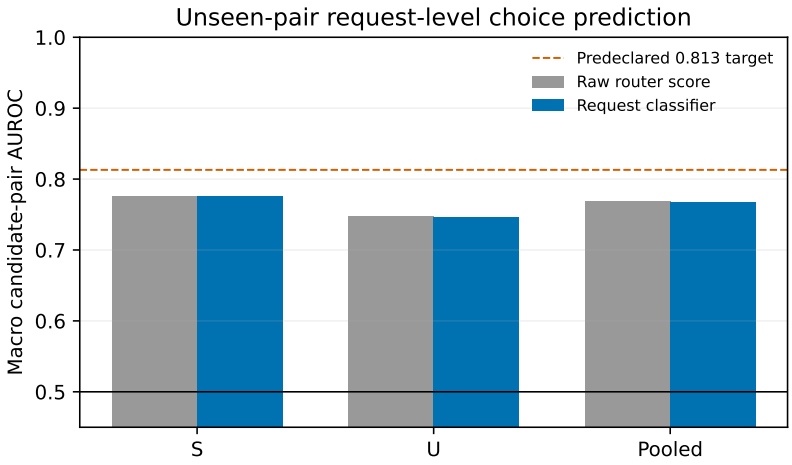

Figure 10: **Per-request choice probe.** The learned classifier does not improve macro candidate-pair AUROC over the raw router score in either matrix and remains below the predeclared 0.813 target.

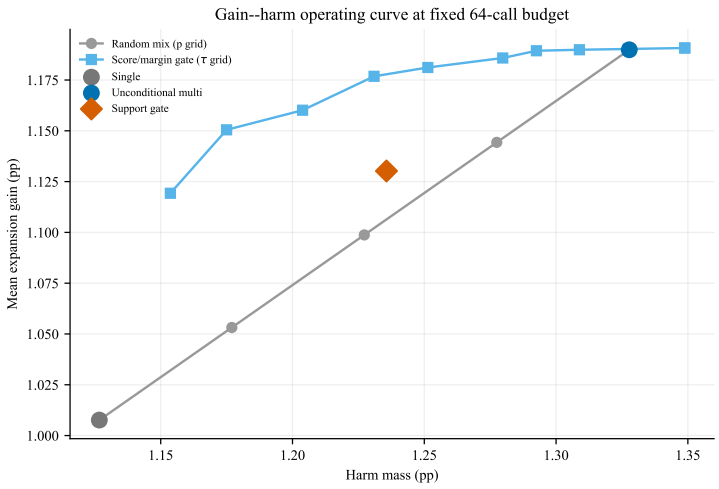

Figure 11: **Risk–return operating curves.** The random-mix line interpolates between the pair-fixed and per-request endpoints. The full margin gate lies above the support gate at the support gate's harm level; this is a point-estimate advantage from a retrospective threshold sweep, not a predeclared dominance claim.

Matrix	Full splits	Non-anchor splits	Top-q Δ	Top-q negative
A	$10/10$	$0/10$	-0.094 pp	48.0%
B	$10/10$	$4/10$	-0.083 pp	47.4%

Table 25: **Repeated-split falsification of the historical band at $K = 6$.** "Full" includes the random and oracle anchors. "Non-anchor" asks whether the bottom and top quintiles of the remaining routers have negative and positive mean gain respectively. The last two columns average over the ten splits.

E.4 Outcome oracles

The outcome oracles use evaluation labels to measure headroom under label-aware selection. The best-fixed oracle chooses one candidate per pair by evaluation accuracy, while the per-request oracle chooses the better candidate separately on each request. Their gap measures the value of request-level flexibility; the gap between the per-request oracle and the realized 1.19 pp is remaining choice headroom.

The *best-fixed* oracle keeps the per-request adoption set and, for each intervention, picks the better of the two candidates, reaching 2.51 pp. The *per-request choice* oracle picks a beneficial candidate on every request, reaching 2.84 pp. The 0.33 pp gap between them is the part of the improvement that requires per-request rather than per-intervention flexibility.

The gate oracle. Under the action set used here, the gate oracle can return the incumbent whenever both candidates are harmful, so its harmful-replacement mass is zero. Write $d = Y_{\text{cand}} - Y_{\text{inc}} \in \{-1, 0, 1\}$; a harmful adoption is exactly $d = -1$, and returning the incumbent repairs every such adoption, including requests on which both candidates are wrong. In the equation below, H_{request} is the per-request policy's harm level, not a Δ.

$$G_{\text{gate}} = G_{\text{request}} + H_{\text{request}} = 1.1899 + 1.3278 = 2.5177 \text{ pp}, \tag{3}$$

within 0.008 pp of the best-fixed oracle above. The gate oracle is a label-using upper bound and is not a deployable policy.

F The Historical Routing-Quality Band (Matrices A, B, C)

Before developing the request-wise ledger, we evaluated a global routing-quality score on Matrices A–C. The apparent zero crossing came from the constructed random and oracle endpoints and disappeared when those anchors were removed. Because these matrices do not contain the request-level action record used in the main analysis, they motivated the move to a coverage/switching decomposition.

The band. On A and B a random-to-oracle ladder of old-pool routing quality had negative mean expansion gain at its low end and positive at its high end in all eight matrix-by-pool-size cells ($K \in \{2, 4, 6, 8\}$), with an estimated zero crossing between 0.563 and 0.859 of the normalised scale. That reading does not survive removing the two anchors. Across ten outcome-independent stratified splits at $K = 6$, the full ladder — which contains a uniform-random and an oracle endpoint by construction — crosses zero in all 20 splits, while the bottom-to-top quintile of the *remaining* routers crosses in 0 of 10 Matrix A splits and 4 of 10 Matrix B splits, and its top non-anchor quintile stays slightly negative on average with 47–48% of interventions negative (Table 25). Under any-correct Top-k fusion ($k \in \{1, 2, 4\}$) the full ladder still crosses in all six matrix-by-budget cells; the non-anchor audit crosses in 1 of 3 Matrix A budgets and 0 of 3 Matrix B budgets.

Why it fails to order routers. On the same 381 Matrix B $K = 6$ interventions a UniRoute-style router attains $+0.308$ pp mean gain with 0.79% negative interventions, while two RouteLLM-style variants attain -0.503 and -0.567 pp with 66.9–69.3% negative ones: the scalar scale does not order these architectures. What is stable is a proposal-level balance — for an ungated router $\Delta = C_{\text{all}} - T_{\text{router miss}} - T_{\text{harm}}$, and the sign of the net effect follows the ratio of beneficial to harmful replacements, 0.123% against 0.112% in Matrix A's high-quality deterministic profiles and 1.353% against 2.403% in Matrix B. Matrix C makes the same point with a strong trained router: RouterDC improves mean accuracy by 1.07 pp while 23.05% of its $2{,}000$ interventions are negative (1.01 pp

and 22.95% on a disjoint replacement manifest). A positive average therefore does not imply safe expansion, which is the motivation this paper inherits, not a result it establishes.

The global score does not provide a portable threshold. Its apparent zero crossing depends on the constructed endpoints and on the particular ladder, which motivates the request-wise coverage/switching analysis.

G Held-Out Asset and Independent Score (Matrix Z)

This appendix separates the two checks performed on Matrix Z. The original Z run was frozen before its outcomes were viewed and serves as the held-out validation. The later R1-on-Z run was conducted after that asset had been inspected and tests score construction rather than a second untouched dataset.

Section 4.1 reports the original controlled 2×2 on Z. This block records its provenance, the full cell and interval results, and the separate R1 follow-up.

The asset. Matrix Z is built from the public EmbedLLM release (Zhuang et al., 2024), which is used nowhere else in this project; no other matrix, score, threshold or intervention of this paper touches it. It is a cross-dataset, cross-task *text-model* routing matrix — not a third multimodal matrix — covering 10 benchmark groups and two task settings (29,359 multiple-choice / QA items and 6,608 math items) over 112 text models carrying 70 family labels. All 112 models enter as candidate slots and all 112 occur in old pools; nothing is banded, filtered or ranked for inclusion, so the different-family constraint on the two candidates is the only restriction, exactly as in S and U.

Sample identity, the item key, and the count. The released question list holds 35,673 distinct `prompt_id` values, but `prompt_id` is *not* a reliable item key: in the validation split 19 of those ids appear under more than one benchmark category, and for 585 of the 2,128 (id, model) pairs the categories carry a different label for the same model, so the rows are different items rather than duplicates. The item key used throughout is the pair (category, `prompt_id`). Under that key 277 `prompt_id` values recur across subsets — one of them under six — producing 294 additional items and a matrix of 35,967 items: 29,917 train, 3,021 validation and 3,029 test, with no duplicate item key and no missing model–item entry. The EmbedLLM paper reports 36,054 questions, which differs from 35,967 by 87. We have not reconciled that gap beyond noting that the count is taken from a different release revision; every number in Section 4.1 and here is computed from the files we hashed, and the reconciliation does not change any reported quantity.

Role assignment is by a stable hash of the item key under a fixed salt, giving 9,000 score-training items, 5,000 calibration items (from which the 32-row support profile is drawn), 18,938 threshold-tuning items and 3,029 final evaluation items. The evaluation split is the only split any reported utility is measured on, and it is disjoint from every other role by construction (score-training/later overlap 0).

The score, and what was deliberately not used. The score is produced by the project's own builder script, unmodified, with a source-only out-of-fold scheme: TF-IDF features on the concatenated dataset tag and question text, a per-model logistic regression, five-fold out-of-fold fitting *within* the score-training split, then a full-fit prediction elsewhere. Matrix Z is used as the held-out asset. R1 is a later replay on the same asset with an independently constructed score.

One upstream artefact deserves an explicit note, because it is a selection channel that would invalidate the word "held-out". The EmbedLLM release also ships a matrix-factorisation embedding checkpoint, `optimal_mf_model_embeddings.pth`, which its README describes as coming from the training run with the *highest test accuracy*. That checkpoint was *not* downloaded and is not used anywhere: only the README, the model and question order files, and the three label CSVs were fetched, and the score is trained from the label matrix alone. No representation selected by test performance enters the score, the support profile or the threshold, and the evaluation labels enter nothing but the reported utility.

The 2×2, the ledger, and the intervals. Table 26 gives the four cells, the two interactions and the decomposition on Z alongside the same quantities on $S + U$. The decomposition identity closes to

	$S+U$	Z
G_{11} / G_{12} / G_{21} / G_{22}	1.008 / 1.058 / 0.740 / 1.190	1.260 / 1.279 / 1.134 / 1.455
H_{11} / H_{12} / H_{21} / H_{22}	1.127 / 1.111 / 1.604 / 1.328	1.293 / 1.294 / 1.725 / 1.589
ΔG / ΔH	$+0.182$ / $+0.201$	$+0.195$ $[+0.158, +0.237]$ / $+0.297$ $[+0.268, +0.326]$
I_G / I_H	$+0.399$ / -0.261	$+0.302$ $[+0.251, +0.360]$ / -0.137 $[-0.173, -0.105]$
coverage gain, J_1 / J_2	-0.268 / $+0.131$	-0.126 $[-0.156, -0.098]$ / $+0.177$ $[+0.146, +0.212]$
coverage harm, J_1 / J_2	$+0.478$ / $+0.216$	$+0.433$ $[+0.379, +0.491]$ / $+0.296$ $[+0.267, +0.325]$
choice gain, A_1 / A_2	$+0.051$ / $+0.450$	$+0.019$ $[+0.008, +0.030]$ / $+0.321$ $[+0.265, +0.382]$
choice harm, A_1 / A_2	-0.015 / -0.277	$+0.001$ $[-0.004, +0.006]$ / -0.136 $[-0.174, -0.102]$
coverage ΔH / share of ΔH	$+0.219$ / 108.7%	$+0.298$ / 100.5%
switch ΔH	-0.018 $[-0.021, -0.014]$	-0.002 $[-0.007, +0.003]$
coverage net / switch net	$+0.128$ / $+0.054$	$+0.172$ $[+0.141, +0.207]$ / $+0.024$ $[+0.014, +0.035]$
incumbent accuracy	79.24%	49.07% [48.87, 49.28]

Table 26: **The controlled 2×2 and its decomposition on both sides.** The $S+U$ column is pooled point estimates from the frozen record and carries no intervals here; the Z column carries the intervals reported in Section 4.1. Both columns use the same definitions, the same policy families and the same threshold grid.

routing quality ρ_R	switch ΔG	switch ΔH	I_G	I_H
0.636	$+0.0538$	-0.0176	$+0.399$	-0.261
0.569	$+0.0207$	-0.0028	$+0.347$	-0.218
0.423	-0.0074	$+0.0092$	$+0.269$	-0.160
0.224	-0.0251	$+0.0149$	$+0.167$	-0.101
0.001	-0.0369	$+0.0183$	$+0.114$	-0.091

Table 27: **The degradation ladder, five-draw means, pooled scope.** Each rung's incumbent, proposal set, candidate choice, support profile and threshold move together, so the table describes a path, not a cross-asset quality axis.

1.1×10^{-16}. Intervals resample the 2,000 interventions of the single matrix, which share questions and models, so the 1,723 distinct candidate pairs among them are not 1,723 independent samples and the two columns of the table are not compared by a test.

Where the interaction sits, on Z. The regrouping of Section 3.3 applies unchanged. C_{add} carries $+0.2970$ pp of I_G (of $+0.3020$, i.e. 98.3%) and -0.1344 pp of I_H (of -0.1370, i.e. 98.1%), while C_{drop} carries $+0.0051$ and -0.0026 pp. The incumbent is right on 35.16% of the newly adopted requests against 49.73% where neither policy adopts, so the same "harder requests" reading holds, at a much lower absolute accuracy. Two magnitudes differ from $S+U$ and are worth recording without interpretation: the coverage share is larger on Z (4.17% against 3.10%, with the abandon rate 0.34% against 0.26%), and the correctness-disagreement rate inside C_{add} is lower (19.83% against 23.21%).

Why the switch leg is smaller, on the same definitions. The perturbation ladder of Section 4.3 runs the whole 2×2 at five levels of a label-free score degradation, and its switch leg changes sign along the path. Table 27 gives the five-draw means.

Matrix Z's own routing quality at the frozen score is $\rho_R = 0.333$. That value lies between the 0.423 and 0.224 rungs, and at both of those the switch ΔH is *positive*, so the ladder does not predict Z's -0.002. Matching Z's value to the 0.569 rung would be post-hoc selection. Two further reasons not to read the table as a cross-asset law: each rung moves five things at once, so no threshold in ρ_R is identified; and ρ_R is normalised by each asset's own random-to-oracle gap, so 0.333 against 0.636 is not "half the selection ability" and does not by itself fix how much harm removal is available. What can be said is the weaker statement: Z did not confirm the switch harm reduction, this is compatible with the situational dependence the ladder exposes, and the difference cannot currently be attributed to routing quality.

An exploratory reading of the same switch set. The analysis below was proposed *after* the Z result was seen. It is exploratory mechanism analysis; it does not revise the confirmation result, and the confirmation of the interaction and the non-confirmation of the switch reduction stand as reported above. It applies the frozen switch-set definition to all three matrices with identical code and definitions, and it reproduces the published pooled $m_d = 0.225\%$ and $a_d = 0.620$ from its own

	S	U	Z
switch rate $\mathbb{P}(S)$	1.491%	0.478%	0.936%
differing mass m_d	0.285%	0.060%	0.190%
differing share of switch	19.09%	12.61%	20.33%
choice skill a_d	0.623	0.581	0.562
incumbent correct, of m_d	46.26%	52.83%	37.81%
incumbent *wrong*, of $\mathbb{P}(S)$	33.03%	23.92%	56.03%
new harm rate	0.0550	0.0128	0.0352
harm reduction rate	0.0767	0.0191	0.0368
reduction : new harm	1.39	1.49	1.05

Table 28: **Decomposing the switch set on all three matrices.** Rates are over the evaluation split; the last two rows are the two directional event rates whose difference is the reported net switch ΔH. Exploratory: proposed after the Z result was seen.

	S	U	Z
harm-changeable mass m_H	0.1317 $[0.1166, 0.1474]$	0.0319 $[0.0266, 0.0372]$	0.0719 $[0.0606, 0.0840]$
choice share a_H	0.582 $[0.572, 0.592]$	0.599 $[0.562, 0.632]$	0.511 $[0.475, 0.544]$
new harm $m_H(1-a_H)$	0.0550	0.0128	0.0352
removed harm $m_H a_H$	0.0767	0.0191	0.0368
switch ΔH (pp)	-0.0216 $[-0.0259, -0.0176]$	-0.0063 $[-0.0093, -0.0035]$	-0.0016 $[-0.0068, +0.0034]$

Table 29: **The two factors of the switch harm row, with intervals.** m_H is the share of evaluation requests where both policies adopt different candidates, the incumbent is right and the two named candidates disagree in correctness; a_H is the share of those requests on which the per-request candidate is the right one. The last row equals $m_H(1-2a_H)$, and its point value reproduces the frozen ledger row. Rows are read as follows: m_H and the two harm rows are percentages of *evaluation requests* (0.1317 means 0.1317% of requests), a_H is a conditional share among the harm-changeable requests (0.582 means 58.2% of them, not 0.582% and not a share of the evaluation split), and switch ΔH is in percentage points. **Unit check:** $m_H \le m_d \le \mathbb{P}(S)$ on every asset — on S those are 0.1317%, 0.2847% and 1.4914% — so m_H is a share of a small subset of requests and is read in hundredths of a percent; a value of 13.17% would exceed the whole switch rate. Intervals are 95% clustered bootstraps resampling interventions within matrix ($5{,}000$ resamples, seed 20260918), all quantities recomputed inside one draw; the other tables in this paper report $2{,}000$ resamples of the same unit. Where the a_H denominator is zero the value is undefined and is never filled with 0 or $1/2$; no asset here is undefined. The neighbouring table's a_d is defined on every informative switch, so a_H and a_d are different conditional events and are never interchanged. *Exploratory*: proposed after the Z result was seen.

benefit/loss split (0.2251% and 0.6196), which is the check that the implementation agrees with the frozen record.

The switch leg is not small on Z for want of switching: Z's switch rate sits between S's and U's, its differing share of the switch set is the highest of the three, and its a_d is only slightly below S's. Two separate factors have to be kept apart here. The *mass* fixes the opportunity — how many requests can produce either harm event at all — and both events require the incumbent to be right, so the share of the differing mass on which it is right is 37.81% on Z against 46.26% on S and 52.83% on U. Z's harm-changeable mass is therefore the smallest fraction of its own differing mass, but not the smallest mass: m_H is 0.0719% of requests on Z against 0.0319% on U and 0.1317% on S, so the opportunity on Z is larger than U's. The *balance* is what makes the net vanish, and it is a property of the choices, not of the pool's competence: new harm and harm reduction come out nearly equal on Z (1.05 against 1.39 and 1.49), which is the arithmetic of $a_H = 0.511$ sitting close to $1/2$ — the incumbent being wrong more often shrinks the opportunity but cannot by itself make the two directions cancel. Effective mass sets the scale, a_H sets the balance, and the first does not imply the second. Offered as a candidate explanation, not a result. Pooled, $m_H = 0.1052\%$ $[0.0943, 0.1169]$ and $a_H = 0.583$ $[0.574, 0.593]$; estimated directly, Z's a_H lies below S's by -0.071 $[-0.108, -0.036]$ and below U's by -0.088 $[-0.136, -0.039]$, while $U - S$ is $+0.017$ $[-0.020, +0.051]$. One asset's interval excluding a value and another's covering it is not a difference and is not read as one.

Asset	Score	G_{11}	G_{12}	G_{21}	G_{22}	H_{11}	H_{12}	H_{21}	H_{22}	adopt (%)
S	R0	0.916	0.983	0.599	1.137	1.236	1.217	1.808	1.483	21.97
	R1†	0.946	1.002	0.686	1.157	1.164	1.149	1.673	1.393	20.59
U	R0	1.262	1.268	1.130	1.336	0.823	0.819	1.042	0.898	22.77
	R1†	1.237	1.225	1.111	1.293	0.717	0.724	0.917	0.791	20.52
Z	R0	1.260	1.279	1.134	1.455	1.293	1.294	1.725	1.589	20.38
	R1	1.247	1.263	1.134	1.436	1.285	1.286	1.668	1.539	19.78

Table 30: **The four cells of the controlled 2×2 under both scores.** Percentage points; G is net gain $B - H$, H harm mass, $A \in \{A_1, A_2\}$ the adoption rule and $J \in \{J_1, J_2\}$ the named candidate. Adoption is the share of evaluation requests on which the policy adopts. R0 reproduces the frozen record; †on the multimodal assets both scores are text-input routers reading the same field, so the R0/R1 contrast there is a contrast of representations. Absolute accuracy on the evaluation split is not part of this record and is therefore not reported here.

G.1 A SECOND, INDEPENDENTLY BUILT SCORE (R1)

This block documents Section 4.2. The intervention tables, item roles, pool size, the two-candidate setting, the $32 \times 2 = 64$ profile calls, the support selection rule that fixes J_1, the threshold-fitting objective with its search budget and fallbacks, and the aggregation and interval conventions are the ones of the frozen record and were not re-tuned. **R0 is the frozen score.** Its eight cells are rebuilt from the cache and reproduce the frozen record to a maximum absolute residual of 2.2×10^{-16} across the reconstruction and identity checks recorded with the artefact, so the two rows of each block are comparable cell for cell.

How R1 was built, and what was held fixed. R1 replaces the ranking signal and nothing else. It is a frozen pretrained sentence encoder (`all-MiniLM-L6-v2`, local snapshot, no fine-tuning of the encoder) producing a mean-pooled, L_2-normalised 384-dimensional representation of the question text, followed by the *same* per-model logistic-regression head as R0, refit on the same source training labels with no scaler and the same folds. Encoder, head and hyperparameters were fixed before the run. R1 was not selected by any test-side quantity — not by the interaction, not by the coverage sign, not by test AUROC and not by any comparison against R0 on the evaluation split — and the encoder is one this project had never used; the source-side metrics below are the only model-selection-relevant numbers, and they were read after the code was frozen. **Input: both scores read the same text field, character for character** — the dataset tag concatenated with the question — because the R1 builder reuses R0's field verbatim; R0 then applies TF-IDF and R1 a sentence encoder to that one field. Neither score can see the image on the multimodal assets, so on S and U both are *text-input routers* over the same input, and the R0/R1 contrast there is a contrast of representations and not of inputs. The multimodal experts of those assets and the text-only input of the routers are separate facts and are never used to explain one another. Extra embedding computation and score-training cost are recorded separately and are not claimed to fit inside the 64 profile calls.

Evidence status. S and U are development assets and every R1 result on them is discovery or robustness. Z has already been inspected, so the original frozen experiment stands as the confirmation of the original hypothesis and the R1 run on Z is *follow-up validation*, not a second confirmation on a never-inspected asset.

Source-side metrics and what the second score is not. Out-of-fold AUROC on the source training labels, mean over models, is $0.765/0.737$ on S (R0/R1), $0.756/0.727$ on U and $0.752/0.733$ on Z, so R1 predicts source-side correctness less accurately out of fold on every asset. Both scores read the same text field — the dataset tag and the question — so neither can see the image on S or U, and this gap is not attributable to one of them reading the image. Two consequences follow. The cross-score difference is *not* the effect of a single quality factor: the two scores differ in representation and in source-side accuracy at once — their input field is the same — so the comparison answers whether the interaction is a property of the two axes or of one way of building a score, and it does not decompose that difference into a quality component. And because R1 was fixed in advance and never chosen on test-side behaviour, its rows are reported as they came out rather than adjusted toward or away from R0. Overall AUROC is reported only as a source-side diagnostic;

Asset	Score	I_G	I_H	cov. J_1	cov. J_2
S	R0	$+0.471$ $[+0.425, +0.520]$	-0.305 $[-0.338, -0.275]$	-0.316 $[-0.351, -0.284]$	$+0.155$ $[+0.137, +0.175]$
	R1	$+0.415$ $[+0.375, +0.459]$	-0.265 $[-0.294, -0.238]$	-0.260 $[-0.290, -0.232]$	$+0.155$ $[+0.138, +0.174]$
U	R0	$+0.200$ $[+0.169, +0.232]$	-0.139 $[-0.162, -0.118]$	-0.132 $[-0.158, -0.109]$	$+0.068$ $[+0.052, +0.085]$
	R1	$+0.194$ $[+0.164, +0.225]$	-0.133 $[-0.156, -0.112]$	-0.126 $[-0.152, -0.104]$	$+0.068$ $[+0.052, +0.086]$
Z	R0	$+0.302$ $[+0.251, +0.360]$	-0.137 $[-0.173, -0.105]$	-0.126 $[-0.156, -0.098]$	$+0.177$ $[+0.146, +0.212]$
	R1	$+0.286$ $[+0.237, +0.340]$	-0.131 $[-0.164, -0.102]$	-0.113 $[-0.141, -0.088]$	$+0.173$ $[+0.142, +0.210]$

Table 31: **The interaction and the coverage legs under both scores, with intervals.** The last two columns are the coverage gain at J_1 and at J_2. This is the citable form of the body's claim that the interaction keeps its sign and excludes zero on all three assets: every interval above excludes zero, $I_G > 0$ and $I_H < 0$, and the coverage leg is negative at J_1 and positive at J_2 in all six rows. Intervals are the same 95% clustered bootstrap resampling interventions within matrix, $2{,}000$ repetitions, all cells recomputed inside one draw.

Contrast (pp)	point	interventions	items	items × interventions
I_G	$+0.3020$	$[+0.252, +0.355]$	$[+0.274, +0.329]$	$[+0.242, +0.362]$
I_H	-0.1370	$[-0.172, -0.106]$	$[-0.150, -0.124]$	$[-0.174, -0.104]$
coverage gain at J_1	-0.1255	$[-0.155, -0.098]$	$[-0.142, -0.109]$	$[-0.161, -0.094]$
coverage gain at J_2	$+0.1765$	$[+0.146, +0.209]$	$[+0.156, +0.196]$	$[+0.139, +0.215]$
choice gain at A_1	$+0.0186$	$[+0.008, +0.030]$	$[+0.012, +0.025]$	$[+0.007, +0.032]$
choice gain at A_2	$+0.3207$	$[+0.266, +0.378]$	$[+0.290, +0.351]$	$[+0.256, +0.385]$

Table 32: **The same contrasts under three resampling units.** The point column is pooled over items and interventions and is identical in all three arms, reproducing the frozen record's pooled cells with no residual in the run's own reconciliation. Intervals are 95% percentile bootstraps, $2{,}000$ repetitions, seed 20260918; the first arm recomputes the published unit inside this code path. Assets, pools, candidate pairs and thresholds are held fixed in every arm.

it is not substituted for the candidate-pair and conditional-event metrics of Table 20, which are the quantities the interaction is about.

G.2 What changes when the requests, not the interventions, are the unit

Every interval in the body resamples interventions within matrix and is therefore conditional on the fixed evaluation-item set; matrix Z's $3{,}029$ items are reused by all of its $2{,}000$ interventions. This sensitivity adds two further units, each conditional on something different. The *items* arm holds the intervention table fixed and resamples the evaluation items, giving each item one weight inside a draw that applies to every intervention (the item draw is common across interventions rather than drawn per intervention). The *crossed* arm draws items and interventions independently in the same draw and so reflects both sources of variation at once. The cells, the thresholds and the intervention table are untouched — only the resampling unit changes — and the run reproduces the frozen per-intervention record to 0 before any draw is taken.

All six contrasts keep their sign and exclude zero under all three units, so the unit changes the width and not the direction of any leg. The three arms answer different questions and are not corrections of one another. The published interval is conditional on the fixed item set and describes the variation of intervention sampling; the items-only arm fixes the interventions instead and lets the item set vary, so its narrower intervals are the width of a different variance component rather than evidence that the published intervals overstate anything. The crossed arm is the widest on all six and is the one to read when both sources of variation are to be reflected; its intervals are one-signed throughout. The first arm is the control: its point estimates match the frozen record exactly and its endpoints differ from the published ones by Monte Carlo noise, because the draw stream differs — the I_G upper bound is 0.355 here against 0.360 in the confirmation record. The published intervals remain the ones the body reports, and like every other interval in this paper all three arms hold the assets, pools and pairs fixed, so none is a projection onto new items, pairs or model families.

H Mechanism checks

We use three checks to ask whether candidate-quality differences or support-side choice skill can explain the interaction and whether either quantity supports a simple operating rule. The first two

Table 33: **Candidate-pair directions.** Percentages give the fraction of pairs with a strictly positive pair-level quantity. Medians are reported in percentage points (pp). These are descriptive summaries rather than pair-level significance tests.

Asset	pairs	$\Pr(\Delta G > 0)$	$\Pr(\Delta H > 0)$	$\Pr(\Delta G_{\mathrm{sw}} > 0)$	median $(\Delta G, \Delta H)$
S	128	51.6%	76.6%	39.8%	$(+0.001, +0.019)$
U	47	40.4%	85.1%	29.8%	$(+0.000, +0.047)$

summarize the frozen ledgers at the candidate-pair level; the third examines whether support-side choice skill is repeatable across splits. These summaries complement the asset-level intervals in Appendix D.

H.1 Pair-level directions and candidate-quality gaps

Table 33 summarizes the directions of the pair-level differences. Pair-level effects vary around the pooled estimates. Positive harm is more common than positive gain on both assets, consistent with the main result that per-request choice increases both opportunity and exposure.

We next stratify candidate pairs by the observed absolute accuracy gap between the two candidates on the certificate ledger. This is a post hoc diagnostic of heterogeneity rather than an online feature available before candidate evaluation. The gain and harm differences are not monotone in the gap. On S, the low-, middle-, and high-gap strata have

$$(\Delta G, \Delta H) = (+0.484, +0.473), \quad (+0.068, +0.088), \quad (+0.112, +0.167) \text{ pp},$$

respectively. On U, the corresponding values are

$$(+0.066, +0.110), \quad (-0.009, +0.063), \quad (+0.149, +0.051) \text{ pp}.$$

Harm is positive in all six strata. Candidate-quality separation therefore does not give a monotone criterion for deciding when candidate adoption is safer.

H.2 A frozen source-only transfer check

We also test whether a simple pair-level screen transfers between assets. On a source asset, we retain a pair when its disagreement mass m is at least the source median and its choice skill is at least one half. We then apply the same rule to the other asset using only the source-side quantities.

Training on S and applying the rule to U selects 14 of 47 pairs. Their mean $(\Delta G, \Delta H)$ is $(+0.127, +0.099)$ pp, compared with $(+0.070, +0.075)$ pp across all U pairs. The reverse transfer selects nearly all S pairs because the median disagreement mass on U is zero, so it provides little filtering. The screen changes the selected pair set but does not reliably reduce harm.

H.3 Support-side choice-skill repeatability

The support analysis separates two questions: whether a choice-skill estimate is defined, and whether it preserves the same pair ordering on another split. For support sizes $4, 8, 12, 16, 24, 32$, the fractions of pairs with a defined estimate are

$$0.560,\ 0.680,\ 0.726,\ 0.794,\ 0.846,\ 0.863.$$

The corresponding split-half sampling standard deviations are

$$0.296,\ 0.295,\ 0.281,\ 0.273,\ 0.230,\ 0.227,$$

and the split-half Spearman correlations are

$$-0.127,\ -0.025,\ +0.028,\ +0.015,\ -0.096,\ -0.010.$$

Every reported correlation interval includes zero. Larger supports therefore reduce the number of undefined estimates, but they do not make the pair ranking a stable predictor in this cache. This analysis differs from the aggregate support-budget sweep in Table 24, which changes the support/profile budget and recomputes the interaction.

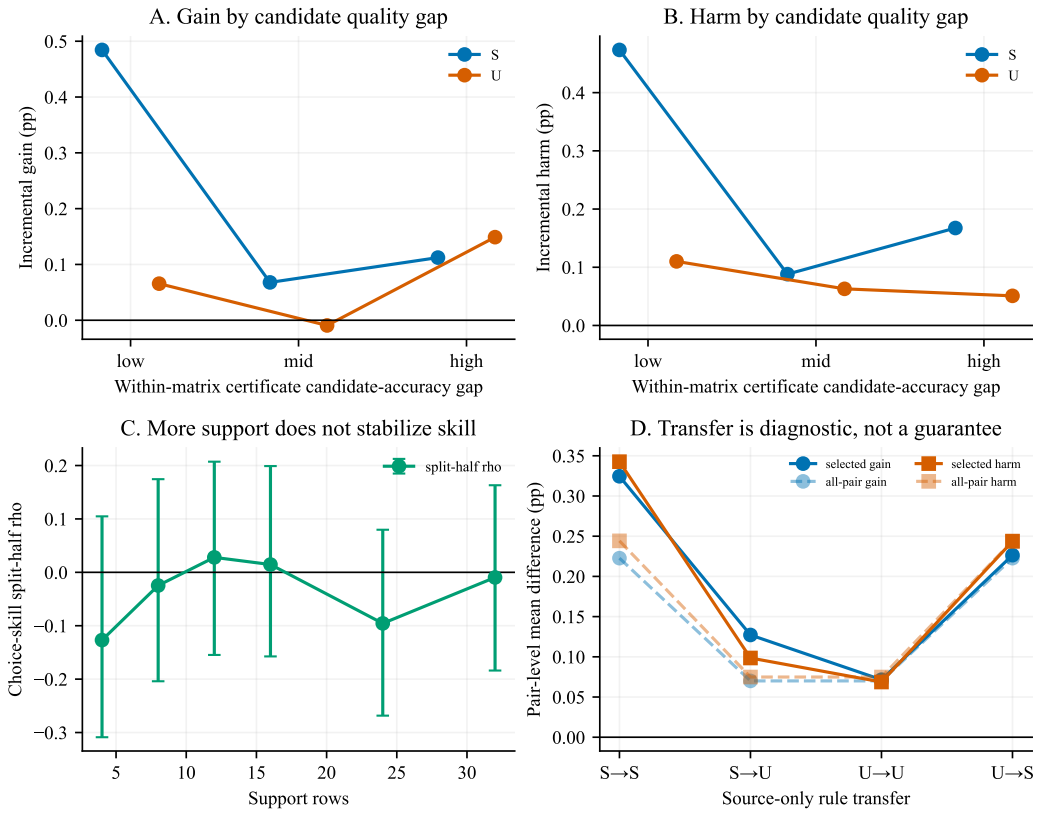

Figure 12: **Mechanism checks across candidate pairs and support sizes.** Panels (A–B) show that candidate-quality-gap strata are non-monotone and have positive harm. Panel (C) shows that larger support budgets make choice-skill estimates available for more pairs, but split-half correlations remain close to zero. Panel (D) shows that a source-only screen changes the selected pair set without reliably reducing harm.

Point	A_{pair}	A_{req}	ΔG	ΔH	I_G	95% CI	I_H	95% CI
Conservative	15.94	18.14	$+0.154$	$+0.157$	$+0.333$	$[+0.226, +0.458]$	-0.211	$[-0.291, -0.142]$
Nominal	19.34	22.18	$+0.182$	$+0.201$	$+0.399$	$[+0.265, +0.541]$	-0.261	$[-0.355, -0.173]$
Aggressive	20.57	23.42	$+0.179$	$+0.207$	$+0.410$	$[+0.283, +0.555]$	-0.272	$[-0.367, -0.186]$

Table 34: **Pooled effects across three operating points.** A_{pair} and A_{req} are the percentages of evaluation requests on which the pair-fixed and per-request policies return a candidate. All other values are percentage points. Confidence intervals are computed by resampling candidate pairs.

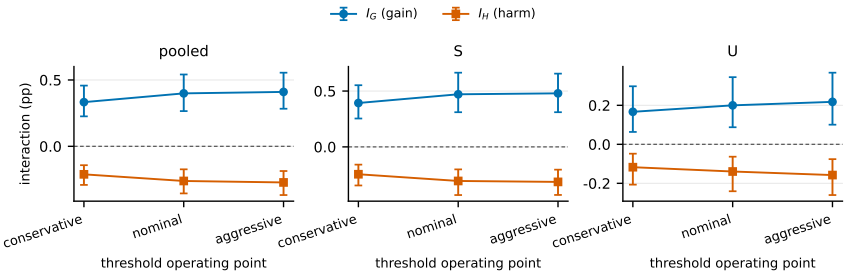

Figure 13: **Interaction estimates across operating points.** The panels show pooled estimates and separate estimates for S and U. Whiskers are candidate-pair bootstrap intervals. The estimates retain $I_G > 0$ and $I_H < 0$ across the conservative, nominal, and aggressive settings.

H.4 Interpretation

Across these checks, the interaction remains visible at the asset level, but neither candidate quality separation nor the source-only screen produces a monotone or reliably harm-reducing rule. Support-side choice skill is also not sufficiently repeatable to support a stable pair ranking. These results support coverage and switching as a diagnostic framework for onboarding analysis; they do not identify a single candidate-admission rule that is reliable across the tested assets.

I Aggregation, Operating Points, and Cluster Sensitivity

We examine how the interaction changes with the adoption thresholds and aggregation weights. We then summarize its variation across candidate pairs, task groups, and model exclusions.

I.1 Operating-point sensitivity

We apply fixed offsets to both candidate-profile thresholds: $+0.05$ for the conservative setting, 0 for the nominal setting, and -0.05 for the aggressive setting. At each setting, we recompute the four adoption-rule and candidate-choice combinations. Candidate pairs, scores, support sets, and the tuning and evaluation splits remain fixed. The nominal setting uses the original thresholds.

The pooled interaction retains the same direction at all three operating points: $I_G > 0$ and $I_H < 0$, with all six intervals excluding zero (Table 34). Per-request choice also increases endpoint gain and harm throughout. Relative to the nominal setting, the conservative setting reduces the gain advantage from $+0.182$ to $+0.154$ pp and the additional harm from $+0.201$ to $+0.157$ pp.

I.2 Aggregation units and pair heterogeneity

The primary estimates pool evaluation requests. We compare them with matrix-equal means, which give equal weight to S and U, and pair-equal means, which give equal weight to each of the 175 candidate pairs. We also report pair-level medians to show how the average effects compare with a typical pair.

Changing the aggregation weights preserves the signs of all four mean quantities (Table 35). Pair-level medians are smaller in magnitude than the corresponding means. Endpoint gain is positive for 48.6% of pairs, while $I_G > 0$ and $I_H < 0$ hold for 66.3% and 68.0%, respectively. Thus the interaction directions are more common across pairs than a positive endpoint gain.

Summary	ΔG	ΔH	I_G	I_H
Request-pooled	$+0.182$	$+0.201$	$+0.399$	-0.261
Matrix-equal mean	$+0.147$	$+0.161$	$+0.335$	-0.222
Pair-equal mean	$+0.182$	$+0.199$	$+0.404$	-0.266
Pair-level median	0.000	$+0.025$	$+0.013$	-0.008
Held-out Z	—	—	$+0.302$	-0.137

Table 35: **Effects under different aggregation weights.** Values are percentage points. The first four rows summarize the development assets. The final row reports the held-out Z interaction estimates; its crossed item–intervention intervals exclude zero in the corresponding directions.

We further summarize the results by task group and repeat the analysis after excluding one model or model family at a time:

Analysis	Comparisons	$I_G > 0$	$I_H < 0$	$\Delta G > 0$
Task groups	20	$20/20$	$20/20$	$18/20$
Leave-one-model/family	45	$45/45$	$45/45$	$44/45$

Both interaction signs hold in all 20 task groups and all 45 model-exclusion analyses. Endpoint gain is positive in 18 task groups and 44 model-exclusion analyses. These comparisons share requests and models, so their sign counts summarize sensitivity within the evaluated assets.