WHEN DOES TASK AFFINITY JUSTIFY PAIRWISE MULTI-TASK CONFIGURATION SELECTION?

Anonymous authors
Paper under double-blind review

ABSTRACT

Task-affinity scores in multi-task learning are typically validated by their correlation with multi-task gain and then used to decide which tasks share what. We argue that these are different claims: a score (*diagnostic*) can track a real *mechanism*, such as representation reuse or output coupling, without determining the right configuration decision. We decompose this inference into three links: score $\rightarrow$ intervention effect $\rightarrow$ preferred configuration $\rightarrow$ out-of-sample selection gain, and test them on 1914 ordered pairs from 172 tasks spanning CelebA, QM9, and 16 multi-output regression families. The first link holds selectively: a direct reuse probe tracks frozen representation reuse and residual dependence tracks output coupling, but no diagnostic reliably predicts the value of adaptation. The second is context-dependent: with the measured task relation nearly unchanged, replacing a small CNN with ResNet-18 reduces adapted-transfer selection from 30% to 7%, while preferences also shift with the candidate set, metric, and target-data budget. A fixed portfolio of two or three configurations reduces oracle headroom below replicate noise on CelebA and QM9. Finally, learned selectors that beat the best fixed configuration on held-out pairs generalize to held-out tasks only on CelebA under the log score, while family extrapolation remains unsupported. We distill these results into principles for more reliable task-affinity evaluation.

1 INTRODUCTION

Multi-task learning (MTL) can improve statistical efficiency and reduce computational cost by sharing representations or parameters across tasks (Caruana, 1997; Vandenhende et al., 2021; Abdelsamie et al., 2026), but its success depends on both *which* tasks are trained together and *how* information is shared. Joint training can enable useful transfer but can also introduce interference (Standley et al., 2020; Guo et al., 2020). Thus, selecting a multi-task configuration is a model-selection problem: with T tasks, pairwise choices already scale as $O(T^2)$, before considering alternative forms of representation sharing or output dependence. This motivates inexpensive *task-affinity* scores, which estimate task relatedness from transfer performance, task embeddings, pairwise multi-task gains, or gradient interactions (Zamir et al., 2018; Achille et al., 2019; Standley et al., 2020; Fifty et al., 2021), as well as learned grouping policies fitted from a limited set of evaluated combinations (Song et al., 2022; Li et al., 2024; Ayman et al., 2026).

These methods are usually validated by correlating affinity scores with observed multi-task gain. Such a correlation, however, neither identifies which mechanism a score measures nor shows that the score supports a useful configuration decision. For example, on the same CelebA task pairs, a probe of representation reuse correlates with the measured reuse effect about equally well under a small CNN and ResNet-18 ($\rho = +0.53$ vs. $+0.43$). Yet the share of pairs for which pretrained-then-adapted transfer is the preferred configuration falls from 30% to 7%. The task relation stayed similar; the decision it supports did not.

This reflects a broader inferential problem: conclusions depend on how quantities are operationalized (Jacobs & Wallach, 2021), how precisely they are estimated (Dodge et al., 2019; Bouthillier et al., 2021), and whether evaluation matches deployment (Lipton & Steinhardt, 2019; Raji et al., 2021). Our central claim is not that task-affinity scores fail, but that a measured task relation is evidence about a mechanism, not a context-free prescription for an action. A score may identify a mechanism without determining the preferred action, and a useful pair-specific decision may fail

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.

to reproduce or generalize. We separate the inference from task relation to decision into the chain: score $\rightarrow$ intervention effect $\rightarrow$ preferred configuration $\rightarrow$ routing gain, formalized in Eq. (1).

To test each link in this chain, we evaluate eight diagnostics across 1914 ordered pairs from 172 tasks spanning CelebA, QM9, and 16 regression families. Our controlled configuration space separates frozen representation reuse, adaptation of the same transferred initialization, and residual output coupling. We then examine whether evidence about these mechanisms determines the preferred configuration and whether pair-specific preferences translate into out-of-sample routing value under pair-, task-, and family-disjoint evaluation.

Contributions. These experiments address three increasingly demanding questions.

- *What does an affinity score measure?* Different diagnostics track different mechanisms: a direct probe tracks frozen representation reuse and residual dependence tracks output coupling, whereas we find no robust cross-domain diagnostic for adaptation of the same transferred initialization (§4).
- *When does a measured relation justify an action?* The preferred configuration depends on the action space, evaluation loss, decision precision, competitor optimization, backbone, and target-data regime; moreover, a small task-agnostic candidate portfolio exhausts most oracle headroom on CelebA and QM9 (§5).
- *Does the resulting decision survive?* Pair-specific selection yields positive cross-replicate value on CelebA and QM9 but not on the regression families, while task-extrapolation routing is supported only on CelebA and only under the log score (§6).

Scope of the contribution. We provide an evaluation framework for establishing what affinity evidence justifies, which yields two reusable evaluation principles. First, near-zero score–effect associations are interpretable only when both quantities exhibit reproducible between-pair variation (Appendix B.4). Second, selector value should be assessed against residual headroom after a small fixed portfolio, H_k in Eq. (6), because comparison with a single fixed configuration can overstate the opportunity for pair-specific selection. These checks separate informative null associations from unreliable measurements, and genuine selector value from gains absorbed by a small fixed portfolio.

2 Related work

Task affinity and grouping. Task-affinity methods estimate relations from transfer performance, task embeddings, and pairwise gain (Zamir et al., 2018; Achille et al., 2019; Standley et al., 2020). These relations are used to group tasks (Li et al., 2024; 2025), integrate grouping into optimization (Liu et al., 2022; Gao et al., 2024; Wang et al., 2024; Jeong & Yoon, 2025), or select sources for unseen targets (Kim et al., 2023). Several methods go beyond score–gain correlation by using estimated affinities or predicted gains to construct and evaluate downstream task groupings (Fifty et al., 2021; Song et al., 2022; Ayman et al., 2026). When the task set is fixed, interference can instead be mitigated through loss weighting or gradient manipulation (Chen et al., 2018; Sener & Koltun, 2018; Yu et al., 2020), within the broader architectural space surveyed by Vandenhende et al. (2021). Our focus is complementary: we ask when evidence about a task relation is sufficient to justify a particular configuration decision.

Evaluating affinity, transferability, and measurement. A complementary literature studies whether diagnostic scores support the effects or selection decisions they are intended to predict. Examples include representation similarity (Kornblith et al., 2019) and transferability estimators (Bao et al., 2019; Nguyen et al., 2020; You et al., 2021). Prior work finds uneven agreement between affinity scores and observed joint-training gains (Azorin et al., 2023), characterizes when gradient-based estimates are informative (Zhang & Cheng, 2026), and shows that conclusions about transferability metrics can change with the evaluation setup (Agostinelli et al., 2022) or benchmark and candidate-model construction (Singh et al., 2026). More broadly, evaluation quantities measure specific constructs (Jacobs & Wallach, 2021; Raji et al., 2021), results vary across runs (Bouthillier et al., 2021), and dyadic task data require resampling that respects shared endpoints (Snijders & Borgatti, 1999). We separate whether a diagnostic predicts an intervention effect, whether that effect determines the preferred configuration, and whether the resulting decision generalizes.

Table 1: Configuration set. Rows vary the representation/training strategy, and columns vary output structure; $_c$ denotes residual output coupling. The four unmarked strategies form the controlled 4×2 design. The two † strategies are added in §5, expanding the candidate set to $|\mathcal{A}| = 12$.

Representation / Training strategy	Independent	Coupled
Separate encoders	sep	sep_c
Shared, frozen at initialization	fr	fr_c
Shared, frozen after auxiliary pretraining	fpt	fpt_c
Frozen pretrained + per-task adapter†	adp	adp_c
Pretrained, then adapted†	ptft	ptft_c
Shared, trained jointly	joint	joint_c

3 SETUP

Notation. Let p denote an ordered task pair with a designated target, $a \in \mathcal{A}$ a candidate configuration, and $L(p, a)$ its target loss on data disjoint from model fitting. Pairs are ordered because loss is defined with respect to the target. A score $s(p)$ is computed without training a candidate configuration, while a routing rule π maps scores to configurations using outcomes from other pairs. We distinguish four quantities that a single affinity–gain correlation can conflate:

$$\underbrace{s(p)}_{\text{score}} \longrightarrow \underbrace{\Delta(p)}_{\text{intervention effect}} \longrightarrow \underbrace{a^\star(p)}_{\text{preferred configuration}} \longrightarrow \underbrace{G(\pi)}_{\text{routing gain}} . \tag{1}$$

Each arrow is a separate inferential step: a score may track an effect that no available action exploits, an effect may not determine the preferred configuration once alternatives compete, and a pair-specific preference may yield no out-of-sample routing gain.

Configurations and interventions. We cross four representation states with independent or coupled outputs, giving eight configurations; §5 adds adapter-based and pretrained-then-adapted states, yielding twelve in total (Table 1). The suffix $_c$ denotes residual output coupling. We define

$$\begin{aligned} \Delta_{\text{reuse}}(p) &= L(p, \mathsf{fr}) - L(p, \mathsf{fpt}), & \Delta_{\text{adapt|transfer}}(p) &= L(p, \mathsf{fpt}) - L(p, \mathsf{ptft}), \\ \Delta_{\text{couple}}(p \mid r) &= L(p, a_r) - L(p, a_{r,c}), & \Delta_{\text{adapt}}(p) &= L(p, \mathsf{fpt}) - L(p, \mathsf{joint}). \end{aligned} \tag{2}$$

Δ_{reuse} measures the value of reusing an auxiliary-pretrained frozen representation; $\Delta_{\text{adapt|transfer}}$ isolates the incremental value of adapting that same transferred initialization; and $\Delta_{\text{couple}}(p \mid r)$ measures the value of residual output coupling at representation state r. By contrast, Δ_{adapt} is compound because it also changes initialization: $\Delta_{\text{adapt}} = \Delta_{\text{adapt|transfer}} + [L(p, \mathsf{ptft}) - L(p, \mathsf{joint})]$. Appendix A provides full experimental details.

Selection and headroom. Because a coupled configuration nests its independent counterpart, minimum-loss selection can favor additional complexity for a negligible empirical gain. Therefore, we define the preferred configuration as the simplest candidate within ϵ of the observed optimum:

$$a^\star(p) = \arg\min_{a \in \mathcal{A}_\epsilon(p)} \text{complexity}(a), \qquad \mathcal{A}_\epsilon(p) = \{a : L(p, a) \le \min_{a'} L(p, a') + \epsilon\}, \tag{3}$$

Here ϵ is the median pair/configuration standard error across replicate cells, and $a^\star(p)$ is used to characterize pair-specific preference. Let $L_{\min}(p) = \min_{a \in \mathcal{A}} L(p, a)$, and $a^\dagger$ the best fixed configuration for the relevant training split. The gain $G(\pi)$ of a routing rule π then decomposes as

$$G(\pi) \;=\; \mathbb{E}_p\big[L(p, a^\dagger) - L(p, \pi(p))\big] = \underbrace{\mathbb{E}_p\big[L(p, a^\dagger) - L_{\min}(p)\big]}_{\text{headroom } H} - \underbrace{\mathbb{E}_p[L(p, \pi(p)) - L_{\min}(p)]}_{\text{regret } R(\pi)} . \tag{4}$$

H is the empirical opportunity for pair-specific selection and $R(\pi) \ge 0$ is the oracle headroom not captured by the router. Because $L_{\min}(p)$ takes the minimum over observed evaluation outcomes, H is an optimistic in-sample quantity rather than an achievable out-of-sample gain. The distinction matters: on the regression families, the in-sample gain is $+0.0376$, whereas selection evaluated on independent replicate cells yields -0.0021. We therefore treat H and H/ϵ as descriptive measures of available headroom. Moreover, H compares against a single fixed configuration; §5 measures the residual headroom after allowing a small fixed portfolio.

Proposition 1. *Even if a score s perfectly tracks its named intervention effect Δ, with $\mathrm{corr}(s, \Delta) = 1$, this implies neither positive selection headroom nor positive achievable routing gain. Let Π_s denote the set of s-measurable routing rules. There exist instances $(\mathcal{P}, \mathcal{A}, L)$ with $H = 0$; instances with arbitrarily large H for which $\min_{\pi \in \Pi_s} R(\pi) = H$, equivalently $\max_{\pi \in \Pi_s} G(\pi) = 0$; and instances differing from the first only in $\mathcal{A}$ for which achievable routing gain is positive.*

Accordingly, the score–effect association alone is insufficient to establish decision value, so we evaluate selection and routing directly. The proof and constructions are given in Appendix D.1.

Domains and evaluation. We evaluate 1190 ordered pairs from 35 CelebA attributes, 90 pairs from 10 QM9 properties, and 634 pairs from 127 tasks across 16 regression families, supporting pair-, task-, and family-level evaluation. A synthetic construction with independently controlled representation sharing and residual dependence provides a ground-truth check. Each real-data domain is split into model-fitting (A), nuisance-estimation (B), and evaluation (C) subsets. We use target negative log-likelihood (NLL), estimating calibration scales and output covariances only on B. Pairwise intervals use vertex resampling to account for dependence between pairs sharing a task.

We match uncertainty to the claim: task or family resampling for population inference, replicate cells for reproducibility, and repeated partitions and initializations for optimization variability. In a dedicated 20 partitions $\times$ 4 initializations variance decomposition for each domain, partition variation exceeds initialization variation for headroom and adaptation, showing that seed-only replication can understate uncertainty in these quantities. Appendices A–F give the full construction, replication protocol, and variance decomposition.

The six independent-output configurations predict the target from x alone and define the x-only space $\mathcal{A}_6^{\mathrm{ind}}$. Coupled configurations additionally condition on the auxiliary residual and require y_a at evaluation. We retain them as mechanism controls for residual dependence (no Δ_{couple} for D_{logdet} to predict without this axis), and report preferred-action and routing results over both spaces.

Diagnostics. We evaluate eight scores spanning six notions of task relation: absolute target correlation; residual dependence, $D_{\mathrm{logdet}}(R_{\mathrm{res}}) = -\frac{1}{2} \log\det R_{\mathrm{res}}$ (Zhou & Vanschoren, 2026); representation similarity (CKA) (Kornblith et al., 2019); transferability (LogME, H-score, and LEEP) (You et al., 2021; Bao et al., 2019; Nguyen et al., 2020); gradient compatibility (gradient cosine); and a direct representation-reuse probe g_{probe}, which compares a linear target head on an auxiliary-pretrained frozen encoder with the same probe on an untrained encoder. H-score and LEEP are classification-specific and evaluated only on CelebA; Appendix A gives full definitions.

4 What does an affinity score measure?

Among the contrasts in Eq. (2), Δ_{reuse} and Δ_{couple} isolate specific interventions, whereas Δ_{adapt} is compound because it also changes initialization. We pre-specify two primary diagnostic–intervention hypotheses: g_{probe}–Δ_{reuse} and D_{logdet}–Δ_{couple}; all other comparisons are exploratory (B.6). We test whether these diagnostics track their hypothesized intervention effects and how robust the associations are to measurement choices. Appendix B provides the full associations, reliability analyses, robustness checks, and synthetic validation.

Mechanism specificity. *Different diagnostics track different mechanisms.* On CelebA, g_{probe} is the most selective reuse diagnostic, correlating $+0.564$ with Δ_{reuse} but only $+0.196$ with Δ_{couple} (Table 2). Its reuse association also resolves on the regression families ($+0.520$), although weak reliability of Δ_{reuse} limits interpretation there; CelebA thus provides the clearest real-data reuse test. QM9 is directionally consistent but imprecise at the task level. By contrast, D_{logdet} is most strongly associated with coupling on QM9 ($+0.657$). Absolute target correlation predicts both reuse and coupling on CelebA, making it predictive but less mechanism-specific. No diagnostic resolves coupling on the regression families. However, for the pre-specified coupling hypothesis, both quantities are sufficiently reliable (0.86 and 0.82), so this null is interpretable (B.4). We do not interpret null associations for exploratory scores unless their reliability has also been established.

The reuse signal is reproducible and mechanism-selective. Across independent runs, the g_{probe}–Δ_{reuse} association differs by at most 0.010 on CelebA and the regression families; on QM9 it

Table 2: Diagnostic–intervention associations. Entries are within-target rank correlations under NLL; * denotes a vertex-bootstrap interval excluding zero. Reuse is $L(\mathsf{fr}) - L(\mathsf{fpt})$, adaptation is the compound contrast $L(\mathsf{fpt}) - L(\mathsf{joint})$, and coupling is $L(\mathsf{joint}) - L(\mathsf{joint_}c)$, i.e. Δ_{couple} at the jointly trained representation state; Table 15 in B.5 gives the controlled adaptation contrast. QM9 results are reported in B.2. The confirmatory hypotheses are g_{probe}–reuse and D_{logdet}–coupling; all other associations are exploratory.

		CelebA			Regression		
Construct	Score	reuse	adapt.	couple	reuse	adapt.	couple
Label assoc.	$\lvert\rho_y\rvert$	$+0.534^*$	-0.079	$+0.456^*$	$+0.164$	-0.226	$+0.248$
Resid. corr.	D_{logdet}	$+0.392^*$	$+0.035$	$+0.487^*$	$+0.135$	-0.129	$+0.134$
Repr. similarity	CKA	-0.080	-0.150	-0.156	$+0.246$	-0.247	$+0.163$
Transferability	ΔLogME	$+0.076$	-0.144	-0.033	$+0.314$	-0.330	$+0.218$
	H-score	$+0.146$	-0.115	$+0.306^*$	–	–	–
	LEEP	$+0.431^*$	-0.022	$+0.393^*$	–	–	–
Gradient compat.	grad. cosine	$+0.106$	-0.029	$+0.108$	–	–	–
Direct probe	g_{probe}	$+0.564^*$	-0.143	$+0.196$	$+0.520^*$	-0.516^*	$+0.140$

remains positive ($+0.625$ versus $+0.558$) but becomes less precise. In a synthetic grid where mechanism strengths are independently controlled, g_{probe} tracks the generative sharing parameter at $+0.874$, while its association with the coupling parameter is unresolved (B.1). Thus, the probe selectively tracks representation sharing rather than merely reproducing a related empirical contrast.

We find no confirmatory evidence that a diagnostic predicts adaptation after transfer. Holding the transferred initialization fixed, g_{probe} does not reliably predict $\Delta_{\text{adapt}|\text{transfer}}$ on CelebA or QM9, with associations of -0.157 and -0.440 and intervals including zero (B.5). The effect itself is reliable in both domains ($R = 0.89$ and 1.00), so these nulls are interpretable (B.4). An exploratory QM9 CKA association resolves, but is not part of the pre-specified inference. On the regression families, effect reliability is only moderate ($R = 0.41$), so the observed g_{probe} association of -0.342 is not evidence of absence. Therefore, the compound Δ_{adapt} provides no reliable evidence that transfer and adaptation are substitutes.

Measurement dependence. Before asking whether a score supports a configuration decision, the intervention effect itself must be well defined. We test two often-implicit choices: how likelihood calibration is estimated and the representation state at which the intervention is evaluated.

Nuisance estimation can reverse the sign of the effect. On the same CelebA models, uncalibrated held-out likelihood suggests that encoder sharing hurts on 90% of pairs (median -0.385), whereas calibration on split B shifts the median to $+0.053$ and changes the preferred configuration on 69% of pairs. Temperature scaling is monotone in the logits, so ROC-AUC and prediction ranks are unchanged; the reversal therefore reflects calibration rather than discrimination. Thus, the sign of a likelihood-based sharing effect depends on how its nuisance parameters are estimated.

Representation state changes the magnitude of the effect. For the same task pairs, mean Δ_{couple} varies across representation states by 0.0124 on CelebA, 0.0897 on QM9, and 0.0210 on the regression families (Table 16). The frozen-random state gives the largest coupling gain in all three domains and remains maximal under a second proper scoring rule; the full ordering agrees on QM9 and the regression families but differs on CelebA (Table 17). Therefore, we write the coupling gain as $\Delta_{\text{couple}}(p \mid r)$, emphasizing that it depends on both the task pair p and the representation state r, rather than on the pair alone. Because coupled configurations condition on the observed auxiliary outcome, this conclusion is restricted to the conditional multi-output setting. Appendix C reports the full calibration, state-wise coupling, and scoring-rule analyses.

5 When does a measured relation justify an action?

A pair-specific effect can be measurable and predictable without determining a useful pair-specific action. We first test how the preferred configuration changes with the decision context, and then ask how much pair-specific selection value remains once strong fixed alternatives are available.

Table 3: Candidate-set dependence of the preferred action. $\mathcal{A}_8$ is the original 4×2 design, $\mathcal{A}_{12}$ adds adapter and pretrained-then-adapted configurations, and $\mathcal{A}_6^{\text{ind}}$ is the x-only subspace. We report preferred-action selection rates by transfer type, together with headroom H and H/ϵ. Within each domain, ϵ is fixed to its $\mathcal{A}_{12}$ value, so rows differ only in the available candidates. Dashes denote unavailable configuration types. Appendix D reports the rerun control and full configuration partition.

Domain	Set	Frozen transfer	Adapted transfer	Any transfer	Adapter	H	H/ϵ
CelebA	$\mathcal{A}_8$	1.8%	–	1.8%	–	0.0176	2.7
	$\mathcal{A}_{12}$	1.2%	24.7%	25.9%	0.0%	0.0206	3.1
	$\mathcal{A}_6^{\text{ind}}$	3.6%	21.3%	25.0%	0.0%	0.0229	3.4
QM9	$\mathcal{A}_8$	0.0%	–	0.0%	–	0.0253	4.6
	$\mathcal{A}_{12}$	0.0%	21.1%	21.1%	0.0%	0.0302	5.2
	$\mathcal{A}_6^{\text{ind}}$	0.0%	15.6%	15.6%	0.0%	0.0214	3.7
Regression	$\mathcal{A}_8$	9.0%	–	9.0%	–	0.0728	1.0
	$\mathcal{A}_{12}$	6.3%	23.7%	30.0%	1.4%	0.0699	1.0
	$\mathcal{A}_6^{\text{ind}}$	7.9%	17.5%	25.4%	0.0%	0.0543	0.8

5.1 Which action is preferred?

The preferred configuration is not a property of the task pair alone:

$$a^\star = a^\star\left(p;\ \underbrace{\mathcal{A}}_{\text{config. set}},\ \underbrace{\ell}_{\text{evaluation loss}},\ \underbrace{\epsilon}_{\text{decision precision}},\ \underbrace{\text{opt}}_{\text{competitor optimization}},\ \underbrace{\text{backbone}}_{\text{model regime}},\ \underbrace{n_{\text{target}}}_{\text{target data}}\right), \tag{5}$$

rather than $a^\star(p)$. We vary these components while matching the task pool, pair set, and remaining design choices as closely as possible. Re-scoring the CelebA configurations under a task-native rank loss changes the best fixed configuration from pretrained-then-adapted to joint training (Appendix E.4). Expanding the action space has a similar effect: frozen transfer is rarely selected in the original 4×2 design, whereas adding pretrained-then-adapted states makes transfer competitive (Table 3). Under minimum-loss selection ($\epsilon = 0$), a transfer-based configuration is selected on 24.9%, 16.7%, and 44.3% of CelebA, QM9, and regression pairs, respectively, almost entirely through adapted transfer (23.8%, 16.7%, and 40.9%). In the noise-aware x-only space $\mathcal{A}_6^{\text{ind}}$, adapted transfer is still selected on 21.3%, 15.6%, and 17.5% of pairs. Selection also varies with ϵ, particularly on the regression families, showing that decision precision is itself part of the decision context (Tables 3 and 4; Appendix D).

Table 4 summarizes all six controls; two warrant further discussion. Competitor optimization has different effects across domains. On QM9, additional joint optimization is negatively associated with the residual advantage of adapted transfer within target ($\rho = -0.568$). CelebA shows no resolved within-target association (-0.063) but a negative between-target association (-0.516), suggesting that the variation appears primarily across targets rather than across pairs within a target.

More importantly, the measured relation can persist while the preferred action changes. Across backbones, the g_{probe}–reuse association retains its direction and similar magnitude ($+0.530$ versus $+0.425$), while adapted-transfer selection falls from 30.3% to 6.8% and $\Delta_{\text{adapt}|\text{transfer}}$ falls from $+0.0226$ to approximately zero. Likewise, on CelebA the reliable g_{probe}–reuse association remains nearly unchanged as n_{target} increases from 200 to 1000 ($+0.564$ versus $+0.539$), while separate-encoder selection rises from 7.9% to 40.3%. These controls show that a measured task relation can remain similar while the preferred configuration changes.

Scope of these comparisons. These are matched decision-context comparisons, not isolated causal interventions. The backbone runs differ in architecture, scale, and device, so we treat them only as model-regime comparisons. In the target-data control, optimizer updates are fixed but target-example exposure increases with n_{target}; the shift persists at $\epsilon = 0$, with separate-encoder selection rising from 3.9% to 25.8% on CelebA and from 18.1% to 40.7% on VOC. The update-matched regression control covers 470 of 634 pairs. Candidate-set size alone does not explain the action-space

Table 4: Decision-context dependence of the preferred configuration. Each row varies one component of Eq. (5) while holding the remaining design and task pool fixed as closely as possible. Unless otherwise stated, percentages report adapted-transfer selection rates.

Context component	Held fixed	Observed change
Configuration set $\mathcal{A}$	pair, diagnostics	Adding pretrained-then-adapted states makes transfer competitive; frozen transfer is rarely selected under the original 4×2 design (Table 3)
Evaluation loss ℓ	trained models	Best fixed configuration changes from pretrained-then-adapted under the target-conditional log score to joint training under $1 - \mathrm{AUC}$
Decision precision ϵ	pair, config. set	$\epsilon = 0$ to $4\times$ the replicate-scale reference: $23.8\% \to 24.7\%$ (CelebA), $16.7\% \to 32.2\%$ (QM9), $40.9\% \to 0.8\%$ (regression)
Competitor optimization	tasks, model family	Update-matched joint model: $24.4\% \to 21.2\%$ (CelebA), $31.1\% \to 6.7\%$ (QM9), $31.9\% \to 32.6\%$ (regression)
Backbone	matched CelebA pairs	Small CNN $\to$ ResNet-18: $30.3\% \to 6.8\%$; $29.5\% \to 7.6\%$ at $\epsilon = 0$
Target data n_{target}	task pool, configurations	$200 \to 1000$ reverses mean frozen reuse ($-0.0097 \to +0.0288$ CelebA; $-0.0231 \to +0.0060$ VOC) and raises separate-encoder selection ($7.9\% \to 40.3\%$; $24.2\% \to 62.6\%$) (Table 24)

effect: adapters are selected on only 0–1.4% of pairs, and subtractive candidate sweeps support the same conclusion. Appendix D reports the full robustness analyses.

5.2 How much pair-specific value remains?

Conventional headroom compares pair-specific selection with the best single fixed configuration. A stronger task-agnostic baseline is a small portfolio whose best member is retained for each pair. For a portfolio S_k, define the residual headroom

$$H_k = \mathbb{E}_p\big[\min_{a \in S_k} L(p, a)\big] - \mathbb{E}_p\big[\min_{a \in \mathcal{A}} L(p, a)\big] \tag{6}$$

H_k is the oracle loss reduction remaining beyond the portfolio; choosing S_1 as the best fixed configuration recovers conventional headroom. It is optimistic because the second term minimizes observed evaluation loss over all twelve configurations. It is also a headroom diagnostic, not a compute-matched routing baseline: evaluating S_k requires training all k configurations for a new pair, whereas a selector aims to avoid doing so (Appendix E.4).

We choose S_k using training tasks only and evaluate it on held-out tasks (Figure 1). Performance-selected portfolios rapidly exhaust the available headroom: on CelebA, H_k falls from 0.0206 to 0.0017 at $k = 2$; on QM9, it falls from 0.0302 to 0.0003 at $k = 3$. In both domains, H_k is below the replicate noise floor ϵ from $k = 2$ onward. The regression families decay more slowly but begin below their larger ϵ. The best-k portfolio substantially outperforms ranking configurations independently by mean loss (by $5.5\times$ on QM9 at $k = 2$), showing that the gain comes from complementarity rather than simply including the strongest configurations on average. Thus, variation in pair-specific oracle identity need not leave substantial value for a learned pair-specific selector.

6 Does the decision survive?

Even when headroom exists, a pair-specific decision is useful only if it persists across held-out replicates and generalizes beyond the pairs used to fit it.

6.1 Is the decision stable across replicates?

We select configurations on one subset of replicate cells and evaluate them on the other, estimating ϵ from the selecting half only. Transfer decisions agree across halves for 87%, 89%, and 74% of CelebA, QM9, and regression pairs, with Cohen's $\kappa = 0.67$, 0.67, and 0.34 (Appendix E).

A stable pair-specific preference does not necessarily yield positive cross-replicate value. Pair-specific selection improves over the best fixed configuration on all evaluated splits for CelebA (me-

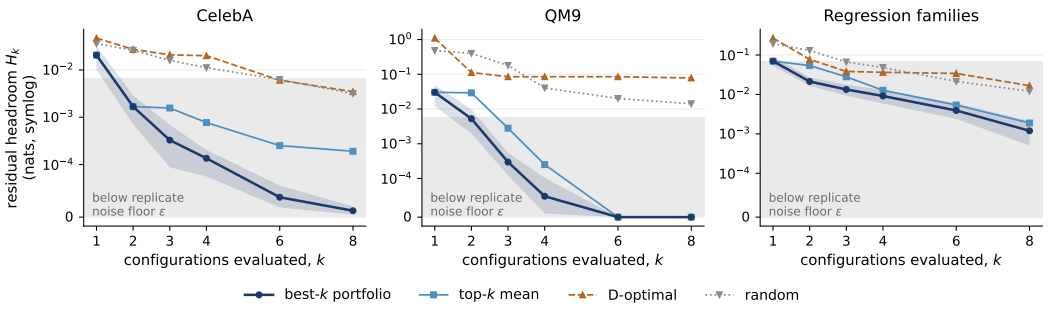

Figure 1: Residual configuration-selection headroom H_k after evaluating k configurations. Portfolios are selected on training tasks and evaluated on held-out tasks. *Best-k* greedily minimizes training best-of-k loss; *top-k mean* ranks configurations by mean training loss; and *D-optimal* uses only factorial configuration descriptors (Appendix D.7). The gray region marks the replicate noise floor ϵ, and shading gives task-level bootstrap intervals. Performance-selected portfolios rapidly exhaust oracle headroom on CelebA and QM9.

dian $+0.0158$) and QM9 (median $+0.0265$), but only 5 of 10 regression splits are positive, with median gain -0.0021. Thus, cross-replicate value is consistently positive for CelebA and QM9 but not for the regression families.

This pattern is consistent with the amount of oracle headroom relative to replicate variability: H/ϵ is 3.1, 5.2, and 1.0 for CelebA, QM9, and the regression families, respectively. Because H uses the minimum over observed configuration losses (§3), H/ϵ is a descriptive, noise-normalized measure of oracle headroom rather than achievable gain; the cross-replicate analysis provides the corresponding empirical selection value. With only three domains, we do not treat H/ϵ as a universal threshold.

6.2 Does it generalize to new tasks and families?

Even if the diagnostics are informative, deployment requires a mapping from diagnostics to configurations that generalizes beyond the pairs used to fit it. We evaluate four routing rules: ridge effect prediction, ridge and gradient-boosted variants with abstention, and a two-stage affinity-plus-correction rule. The last borrows the structure, but not the action space, of a published grouping pipeline (Ayman et al., 2026). Routers are fitted on precomputed diagnostics and configuration outcomes from training pairs; Appendix E gives full details.

Task extrapolation depends on the candidate set. A withheld task is unseen only to *router fitting*: its target labels remain available for the single-task model and diagnostics, while all of its pairs are excluded from router fitting. This is not a zero-shot setting. Under pair interpolation, all four routers resolve positively under $\mathcal{A}_8$ and $\mathcal{A}_{12}$ on both CelebA and the regression families. However, with an entire task withheld, no CelebA router resolves positively under $\mathcal{A}_8$. Task-extrapolation gain resolves positively only in the expanded action spaces: the default router gives $+0.0099$ under $\mathcal{A}_6^{\text{ind}}$ and $+0.0095$ under $\mathcal{A}_{12}$ (Table 27). Task extrapolation on the regression families remains unsupported under every candidate set.

This does not imply that larger candidate sets generally improve generalization: expansion changes both the available interventions and the best fixed alternative. Rather, task extrapolation depends on the action space in which the decision is defined. In $\mathcal{A}_6^{\text{ind}}$, which requires no auxiliary outcome at inference, all four routers resolve positively on CelebA, with gains ranging from $+0.0090$ to $+0.0112$ (Table 5). Thus, the positive routing result does not require access to the auxiliary outcome at inference.

Family extrapolation remains unsupported. Family-disjoint evaluation is available only for the regression benchmark, where no router establishes positive gain under any candidate set. Across increasingly demanding shifts, pair interpolation succeeds, task extrapolation is supported only on CelebA, and family extrapolation remains unsupported (Table 27).

Table 5: Routing gain over the *training-fold* best fixed configuration. Router and baseline are recomputed within each regime, withholding the same unit: a pair, task, or dataset family. Gain range is the minimum–maximum point estimate across four routing rules; "Resolved $+/-$" counts rules whose resampled interval excludes zero in the positive or negative direction. $\mathcal{A}_6^{\mathrm{ind}}$ requires no auxiliary label at inference. Appendix E reports per-router estimates and intervals.

Domain	Regime	Config. set	Gain range	Resolved $+/-$
CelebA	pair interp.	$\mathcal{A}_8$	$+.0094$ to $+.0104$	$4/0$
		$\mathcal{A}_{12}$	$+.0131$ to $+.0140$	$4/0$
		$\mathcal{A}_6^{\mathrm{ind}}$	$+.0144$ to $+.0164$	$4/0$
	task extrap.	$\mathcal{A}_8$	$+.0026$ to $+.0046$	$0/0$
		$\mathcal{A}_{12}$	$+.0085$ to $+.0111$	$3/0$
		$\mathcal{A}_6^{\mathrm{ind}}$	$+.0090$ to $+.0112$	$4/0$
Regression	pair interp.	$\mathcal{A}_8$	$+.0238$ to $+.0308$	$4/0$
		$\mathcal{A}_{12}$	$+.0123$ to $+.0173$	$4/0$
		$\mathcal{A}_6^{\mathrm{ind}}$	$+.0029$ to $+.0078$	$0/0$
	task extrap.	$\mathcal{A}_8$	$-.0141$ to $+.0016$	$0/3$
		$\mathcal{A}_{12}$	$-.0158$ to $-.0034$	$0/3$
		$\mathcal{A}_6^{\mathrm{ind}}$	$-.0133$ to $-.0006$	$0/2$
	family extrap.	$\mathcal{A}_8$	$-.0067$ to $+.0007$	$0/0$
		$\mathcal{A}_{12}$	$-.0179$ to $-.0004$	$0/1$
		$\mathcal{A}_6^{\mathrm{ind}}$	$-.0142$ to $+.0022$	$0/2$

Task extrapolation is metric-dependent. Re-scoring the CelebA sweep with the task-native loss $1 - \mathrm{AUC}$ gives a task-extrapolation gain of $+0.0014\,[-0.0001, +0.0034]$, versus $+0.0099\,[+0.0050, +0.0145]$ under the log score. The sign agrees, but the AUC interval includes zero; the gain is only 0.41% of the loss and $0.13\times$ the replicate noise floor, versus 2.14% and $1.5\times$ under the log score. We therefore treat the routing result as specific to the log score rather than supported under the task-native metric (Appendix E.4).

7 Scope, limitations, and conclusion

Our analysis focuses on ordered target–auxiliary pairs, whereas practical task grouping involves larger sets whose higher-order interactions need not decompose pairwise (Gao et al., 2024; Wang et al., 2024). In a simplified three-task CelebA experiment, pairwise composition matches the group oracle on 40% of triplets, versus 38% for a fixed choice, partly because the oracle often selects an encoder pretrained on both auxiliaries, which has no pairwise analogue (Appendix G).

Residual coupling is evaluated conditionally, using the observed auxiliary outcome y_a, so conclusions about Δ_{couple} are restricted to this setting. Coupled configurations serve as mechanism controls, not necessary deployment options; removing them leaves the preferred-action conclusions of §5 intact across all three domains and the CelebA task-extrapolation routing gain of §6 unchanged. More broadly, conclusions are conditional on the configurations, diagnostics, and domains studied: family extrapolation is available only across the 16 regression families, while QM9's 10 tasks limit precision beyond the pre-specified hypotheses. Appendix H discusses additional limitations.

Task-affinity scores are validated as predictors and then used as prescriptions. Our results show why these claims should be evaluated separately. Diagnostics can be mechanism-specific and reproducible, yet the configuration they support changes with the candidate set, metric, backbone, and target-data budget. On CelebA and QM9, two or three fixed configurations capture most of the apparent pair-specific oracle headroom, while out-of-sample routing generalizes to held-out tasks only narrowly and family-disjoint generalization remains unsupported. For practitioners, this suggests first establishing how much headroom remains beyond a small complementary portfolio before investing in pair-specific selection. For researchers proposing new affinity scores, we recommend explicitly testing mechanism validity, measurement reliability, residual headroom, and deployment generalization. A task relation can remain stable while the action it supports changes, so evaluations should say which of the two they are measuring.

REPRODUCIBILITY STATEMENT

All experiments are run through scripts. The supplementary material contains the analysis and training code, the job specifications, and the per-pair losses and diagnostics as CSV files; the datasets are public and are not included. The released per-pair losses and diagnostics can be used to regenerate every table and figure without retraining the models. Appendix A documents the experimental protocol, diagnostics, and routing implementation; Appendices B.1 and B.6 document the synthetic construction and treatment of confirmatory versus exploratory comparisons.

The release includes a registry that maps the configuration names used in the paper to the abbreviations used in the stored results. The regeneration scripts check this mapping against both the paper and the released data.

Because absolute per-pair likelihoods can vary across hardware, we base inferential comparisons on within-run contrasts, with configurations trained and evaluated on the same device. The backbone control in Appendix D.5 uses separate, non-device-matched runs, so we interpret only within-run contrasts and selection rates, not absolute loss differences across backbones.

AI USE STATEMENT

The authors developed the main idea and overall structure, then used generative AI tools to suggest refinements to claims and hypotheses, propose additional robustness and sensitivity analyses, assist with code and experimental-script development, implement the generative construction in Appendix B.1, and polish the manuscript language. The authors reviewed the edits and take responsibility for the final content of this work.

REFERENCES

Mahmoud Mohamed Abdelsamie, Shahira Shaaban Azab, and Hesham A Hefny. Deep multi-task learning: a review of concepts, methods, and cross-domain applications. *International Journal of Data Science and Analytics*, 21(1):77, 2026.

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *IEEE/CVF International Conference on Computer Vision*, 2019.

Andrea Agostinelli, Michal Pándy, Jasper Uijlings, Thomas Mensink, and Vittorio Ferrari. How stable are transferability metrics evaluations? In *European conference on computer vision*, pp. 303–321. Springer, 2022.

Afiya Ayman, Ayan Mukhopadhyay, and Aron Laszka. Ensemble prediction of task affinity for efficient multi-task learning. In *International Conference on Learning Representations*, 2026.

Raphaël Azorin, Massimo Gallo, Alessandro Finamore, Dario Rossi, and Pietro Michiardi. "It's a match!"—A benchmark of task affinity scores for joint learning. *arXiv preprint arXiv:2301.02873*, 2023.

Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning. In *IEEE International Conference on Image Processing (ICIP)*, 2019.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, et al. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems (MLSys)*, 2021.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A Smith. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2185–2194, 2019.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Christopher Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems*, 2021.

Yuan Gao, Shuguo Jiang, Moran Li, Jin-Gang Yu, and Gui-Song Xia. DMTG: One-shot differentiable multi-task grouping. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *International conference on machine learning*, pp. 3854–3863. PMLR, 2020.

Neil Houlsby, Andrei Giurgiu, Stanisław Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pp. 2790–2799, 2019.

Abigail Z. Jacobs and Hanna Wallach. Measurement and fairness. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021.

Wooseong Jeong and Kuk-Jin Yoon. Selective task group updates for multi-task optimization. In *International Conference on Learning Representations*, 2025.

Joongwon Kim, Akari Asai, Gabriel Ilharco, and Hannaneh Hajishirzi. TaskWeb: Selecting better source tasks for multi-task NLP. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 11032–11052, 2023.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.

Dongyue Li, Aneesh Sharma, and Hongyang R. Zhang. Scalable multitask learning using gradient-based estimation of task affinity. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

Yingya Li, Timothy Miller, Steven Bethard, and Guergana Savova. Identifying task groupings for multi-task learning using pointwise V-usable information. *Journal of Biomedical Informatics*, 2025.

Zachary C Lipton and Jacob Steinhardt. Research for practice: troubling trends in machine-learning scholarship. *Communications of the ACM*, 62(6):45–53, 2019.

Shikun Liu, Stephen James, Andrew J Davison, and Edward Johns. Auto-lambda: Disentangling dynamic task relationships. *arXiv preprint arXiv:2202.03091*, 2022.

Cuong V. Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. LEEP: A new measure to evaluate transferability of learned representations. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

Inioluwa Deborah Raji, Emily M. Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. AI and the everything in the whole wide world benchmark. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, 2018.

Prabhant Singh, Sibylle Hess, and Joaquin Vanschoren. How NOT to benchmark your SITE metric: Beyond static leaderboards and towards realistic evaluation. In *International Conference on Learning Representations*, 2026.

Tom A. B. Snijders and Stephen P. Borgatti. Non-parametric standard errors and tests for network statistics. *Connections*, 22(2):61–70, 1999.

Xiaozhuang Song, Shun Zheng, Wei Cao, James Yu, and Jiang Bian. Efficient and effective multi-task grouping via meta learning on task combinations. *Advances in Neural Information Processing Systems*, 35:37647–37659, 2022.

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, 2020.

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Chenguang Wang, Xuanhao Pan, and Tianshu Yu. Towards principled task grouping for multi-task learning. *arXiv preprint arXiv:2402.15328*, 2024.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical assessment of pre-trained models for transfer learning. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Jasper Zhang and Bryan Cheng. Information-theoretic requirements for gradient-based task affinity estimation in multi-task learning. *arXiv preprint arXiv:2604.07848*, 2026.

Fangqin Zhou and Joaquin Vanschoren. Residual correlation as a diagnostic for joint-uncertainty gains from GP coregionalisation. *arXiv preprint arXiv:2609.30085*, 2026.

A EXPERIMENTAL PROTOCOL AND REPRODUCIBILITY

A.1 EXPERIMENTAL DOMAINS

Table 6 summarizes the role of each domain. For CelebA, we pair a data-poor target with a data-rich auxiliary and retain the 35 attributes whose base rates yield at least ten expected positives at the target budget. We retain the full base-rate range rather than restricting the analysis to approximately balanced attributes. QM9 excludes the three rotational constants, whose heavy-tailed distributions are dominated by a small number of near-linear molecules.

The regression datasets are public multi-output datasets in which tasks within each dataset share the same inputs. They provide a narrower setting for representation transfer than CelebA or QM9. Table 7 lists the 16 families. Each is subsampled to at most 400 rows and contributes both directions of at most 40 unordered pairs so that no family dominates the pool.

Table 6: Experimental domains. Pair counts are ordered pairs. VOC is used only for the target-budget control in §5, not for diagnostic–mechanism tests, for the reason given in Appendix B.4.

Domain	Tasks	Pairs	Families	Primary role	Splits
CelebA	35 attributes	1190	1	representation reuse; residual coupling	pair, task
QM9	10 properties	90	1	residual coupling	pair, task
Regression family	127 tasks	634	16	residual coupling; transfer robustness	pair, task, family
VOC (Everingham et al., 2010)	14 classes	182	1	target-budget control	pair
Synthetic	generated	36 cells	—	construct validation	—

Table 7: The 16 regression families, totaling 127 tasks and 634 ordered pairs. Each family is subsampled to at most 400 rows and contributes at most 40 unordered task pairs, evaluated in both directions. Tasks within a family share inputs, making representation transfer a narrower test than in CelebA or QM9; the 16-family structure enables family-disjoint evaluation. n, d, and T denote the number of samples, input features, and tasks, respectively.

Dataset	n	d	T	Pairs	Dataset	n	d	T	Pairs
atp1d	337	411	6	30	scm1d	400	280	16	80
atp7d	296	411	6	30	scm20d	400	61	16	80
rf1	400	64	8	56	edm	154	16	2	2
rf2	320	576	8	56	oes10	400	298	16	80
jura	359	15	3	6	oes97	334	263	16	80
wq	400	16	14	80	sarcos	400	21	7	42
energy	400	8	2	2	rice	315	204	2	2
tecator	240	100	3	6	wheat	346	204	2	2

A.2 MODELS AND EVALUATION

Splits. Each dataset is partitioned into three disjoint subsets: predictors are fitted on A, nuisance parameters on B, and final losses are evaluated on C. Splits B and C are equal halves of the held-out data. Information from an evaluation unit's C split is not used to fit its predictor, nuisance parameters, diagnostics, or routing decision.

Models and training. Within each domain, configurations share the same base architecture; comparisons vary only the configuration interventions defined in §3.

CelebA: a convolutional encoder with three 3×3, stride-2 convolutions with 32, 64, and 128 channels, each followed by ReLU and batch normalization, then global average pooling; a shared 128-unit linear layer with ReLU; and one linear head per task. Training uses Adam with learning rate 10^{-3}, batch size 256, and 20 epochs. The target receives 200 training examples, the auxiliary 4000, and 3000 examples are held out.

QM9: molecules are represented as SMILES (Simplified Molecular Input Line Entry System) sequences and encoded using 32-dimensional token embeddings, two width-5 convolutions with 64 channels and ReLU, and global average pooling, followed by a shared 64-unit linear layer and the same task-head structure. Training uses Adam with learning rate 2×10^{-3}, batch size 512, and 40 epochs. We use 8000 molecules for fitting and hold out 5000.

Regression families: a two-layer, 64-unit ReLU encoder followed by a shared 64-unit ReLU layer and one linear head per task. Training uses Adam with learning rate 3×10^{-3} for 300 epochs, with 60 rows available to each data-poor target.

VOC and CelebA target-budget control: VOC uses the CelebA architecture and training pipeline unchanged. PASCAL VOC 2007 trainval and test are pooled (9963 images), and images are resized to 64×64. The dataset provides 20 binary class-presence indicators. We retain those with at least ten expected positives in the smaller target budget, $\min(\pi_t, 1 - \pi_t)\, n_{\text{target}} \geq 10$ at $n_{\text{target}} = 200$, the same rule used for CelebA; every VOC class has $\pi_t < 0.5$, so this is simply $\pi_t \geq 0.05$. Fourteen of the twenty classes qualify (Table 6), and the list is fixed across both budgets so that the task pool does not change with n_{target}. Within each replicate cell, one random permutation assigns n_{target} rows to the target, 4000 to the auxiliary, and 3000 to the held-out set, split equally between nuisance estimation and evaluation. The CelebA control uses the same 35 attributes and training specification at $n_{\text{target}} \in \{200, 1000\}$. Optimizer-update count is matched across budgets because it is determined by the fixed 4000-example auxiliary set, although target-example exposure increases with n_{target}. Thus, this is a target-data intervention at matched updates, not matched example passes.

Representation-state controls. An encoder frozen at initialization provides a fixed-random representation baseline but contains no auxiliary-specific information. An *auxiliary-pretrained* encoder instead tests whether features learned from a particular auxiliary benefit the target. We define Δ_{reuse} by contrasting these two states, holding the frozen-encoder constraint and downstream optimization fixed. For auxiliary-pretrained configurations, the encoder is first trained on the data-rich auxiliary split; in fpt, it is then frozen, whereas in ptft, it is adapted on the target. Frozen-random and frozen-pretrained configurations are otherwise trained identically. In both frozen configurations, only the encoder is frozen; the subsequent shared layer and task heads remain trainable.

Adapter configurations. For adp and adp_c, the shared encoder is frozen at the auxiliary-pretrained weights, as in fpt, and each task receives a private residual bottleneck $z \mapsto z + W_2\,\mathrm{ReLU}(W_1 z)$, with $W_1 \in \mathbb{R}^{r \times h}$ and $W_2 \in \mathbb{R}^{h \times r}$, following Houlsby et al. (2019). We use $r = 16$, $h = 128$ on CelebA and $r = 8$, $h = 64$ on QM9 and the regression families. This gives 4240 task-specific parameters versus 93,696 in the small convolutional encoder and 11.2M in ResNet-18, and 1096 versus 4700–41,000 in the tabular encoders. Only the adapters, shared layer, and task heads receive gradients, adding task-specific capacity while keeping the shared encoder fixed. Epoch counts are fixed in advance and held constant across configurations. We do not use early stopping because configuration-specific stopping rules would introduce an additional nuisance parameter.

Loss and nuisance parameters. $L(p, a)$ is the mean per-example target negative log-likelihood (NLL) on C. For binary targets, probabilities are temperature-scaled as $p = \sigma(z/T)$, with T estimated on B; for continuous targets, residual variance is estimated on B.

Coupled configurations additionally condition the target residual on the auxiliary residual. The regression coefficient and conditional variance are estimated on B, with pair covariance shrunk toward its diagonal by 0.1. Hence, evaluation requires the observed auxiliary outcome y_a on C. Independent-output configurations and all representation-side interventions require no auxiliary outcome at evaluation. The same shrinkage is used across configurations and domains and is not tuned per candidate.

A.3 Affinity diagnostics

Definitions. Absolute target correlation is the absolute correlation between the two task outcomes. Residual dependence is measured by $D_{\text{logdet}}(R_{\text{res}}) = -\frac{1}{2} \log\det R_{\text{res}}$ (Zhou & Vanschoren, 2026), where R_{res} is the correlation matrix of held-out single-task residuals. The direct representation-reuse probe g_{probe} pretrains an encoder on the auxiliary, freezes it, and fits a logistic or ridge head

on the target training rows. Its score is the improvement over the same probe using features from an untrained encoder, so larger values indicate greater target benefit from the auxiliary-trained representation.

ΔLogME is defined analogously as LogME for the auxiliary-pretrained encoder minus LogME for an untrained encoder of the same architecture. LogME uses its original fixed-point evidence-maximization procedure, averaging the two one-vs-rest scores for binary targets. We use the difference rather than the raw score because LogME is designed to rank source representations for a fixed target, while its absolute level can contain a target-specific offset. Subtracting the untrained baseline preserves within-target auxiliary rankings while reducing this offset when pairs are compared across targets, as in routing.

LEEP evaluates the source model's predicted label distribution on target inputs. H-score uses class-conditional feature statistics with shrinkage 0.1; without shrinkage, it saturates at 128 feature dimensions with only 200 target examples. LEEP and H-score are therefore used only for classification. CKA is the linear CKA between auxiliary-pretrained and target single-task representations on the same target inputs. Gradient affinity is the cosine similarity between task gradients on the shared encoder, averaged over the second half of a short joint-training trajectory.

Diagnostic computation. On cached features with $n = 3000$ and $d = 128$, absolute target correlation and D_{logdet} each take 0.07 ms per pair, compared with 1.6 ms for H-score, 5.9 ms for g_{probe}, 6.8 ms for LogME, and 14.2 ms for CKA. These timings exclude the dominant cost of obtaining representations. Target correlation and D_{logdet} require only single-task outcomes or residuals, whereas representation-based diagnostics require trained representations, including an auxiliary-pretrained encoder for source-based diagnostics, unless representations can be reused across pairs. We therefore make no end-to-end efficiency claim. The routing experiments instead evaluate *decision validity*: whether diagnostic-based choices outperform the best fixed configuration out of sample and generalize under task- and family-disjoint evaluation.

A.4 Routing implementation

Given routing-training pairs with observed configuration losses, the default router π fits one ridge regressor per candidate, with regularization strength $\alpha = 1$, to predict improvement over the training-fold best fixed configuration. Inputs are five diagnostics—D_{logdet}, g_{probe}, absolute target correlation, ΔLogME, and CKA—plus the $D_{\mathrm{logdet}} \times g_{\mathrm{probe}}$ interaction. Diagnostic standardization is fitted using routing-training pairs only and then applied to the held-out pairs. The router predicts candidate-specific improvements rather than configuration labels.

The router receives no task identity, task embedding, or derived task identifier. "Unseen" therefore means unseen to *router fitting*, not zero-shot: labeled training rows for the held-out task remain available for fitting its single-task model and computing diagnostics, as they would for a new target with labeled training data. We do not use within-target centering for routing because it is undefined when the held-out target has no other pair in the routing-training set.

The training-fold best fixed configuration is the candidate with the lowest mean loss over the available routing-training pairs. It is recomputed after withholding the evaluation pair, task, or family and therefore never uses losses from evaluated pairs. The router selects the candidate with the greatest predicted improvement only when that improvement exceeds τ; otherwise, it returns the training-fold best fixed configuration. The threshold τ is selected within the routing-training data from a grid on $[0, 0.06]$ and is never tuned on the evaluation fold.

Appendix E evaluates the alternative routing mappings used in the main text, including non-abstaining ridge prediction, gradient-boosted prediction with abstention, and the two-stage affinity-plus-correction rule. Gradient boosting uses 120 trees, maximum depth 2, learning rate 0.05, and no subsampling, fixed across folds and domains. Abstaining variants select τ using the same inner-fold procedure. Task- and family-extrapolation folds withhold the corresponding task or family rather than treating dependent dyads as independent.

Table 8: Replication of the principal real-data configuration runs. A replicate cell is one data partition crossed with one weight initialization. Because ϵ is estimated from the across-cell standard error, unequal replication can affect selection rates; the common-budget analysis in Appendix D controls for this difference. Experiments not listed here are described where used.

Experiment	Domain	Partitions	Initializations	Replicate cells
Main configuration sweep	CelebA-35 ($n_{\text{target}} = 200$)	3	2	6
	QM9-10	20	4	80
	Regression families	3	2	6
Variance decomposition	CelebA-35	20	4	80
	QM9-10	20	4	80
	Regression families	20	4	80
Update-matched control	CelebA-35	3	2	6
	QM9-10	5	2	10
	Regression families	3	2	6
Backbone control	CelebA-12	2	2	4
Target-budget control	CelebA-35 ($n_{\text{target}} = 1000$)	3	2	6
	VOC-14 ($n_{\text{target}} = 200$)	3	2	6
	VOC-14 ($n_{\text{target}} = 1000$)	3	2	6

A.5 Replication and uncertainty estimation

A replicate cell is one data partition crossed with one weight initialization. Most configuration experiments use either 3×2 or 20×4 replicate cells; Table 8 gives the exact replication for each experiment. QM9's main configuration sweep uses 20×4 because its 90 pairs are relatively cheap; the update-matched QM9 control uses 5×2, and the corresponding CelebA and regression controls use 3×2.

The near-tie tolerance ϵ is the median, over pairs and configurations, of the standard error of pair-level loss across replicate cells. It therefore depends on both domain variability and replication count; Appendix D examines this dependence using a common-budget analysis.

Intervals for pair-level quantities use a vertex bootstrap that resamples tasks and retains dyads between distinct sampled tasks, accounting for dependence among pairs that share a task endpoint. Routing intervals instead resample the unit withheld by the deployment regime: pairs, tasks, or families as appropriate. Unless stated otherwise, intervals use 2000 bootstrap resamples.

All variance decompositions use the same eight-configuration candidate set as the corresponding headroom analysis; the analysis code verifies the candidate set before recomputation.

Hardware scope. Absolute pairwise likelihoods can vary across devices: in one repeated bake-off, identical code, seeds, and data differed by up to 0.064 nats across hardware classes (Appendix H). However, within a run, all configurations for a pair and replicate cell are trained and evaluated in the same job and therefore on the same device. Contrasts in Eq. (2), configuration selection, ϵ, headroom, tolerance sweeps, and candidate-space comparisons are therefore computed within runs rather than from cross-device absolute losses.

The backbone and target-budget controls compare separately executed runs. Their conclusions use within-run contrasts, selection rates, or correlations and are therefore invariant to a uniform additive device shift; absolute losses are not compared across budgets or backbones. Configuration-dependent hardware effects could nevertheless remain and are not identified by this control. The raw mean NLL in Table 23 is the only cross-run absolute loss reported; it is included for completeness and is not used inferentially.

B Score → construct: validation and robustness

This appendix further tests whether the diagnostics track the constructs they are intended to measure. We first use a synthetic construction in which representation sharing and residual coupling are independently controlled, then report real-data construct results and robustness across runs, encoder initializations, and data partitions.

Table 9: Construct validation against generative parameters known by construction. Spearman correlations are computed across the 36 cell means of the $\alpha \times \beta$ grid, averaging 20 seeds per cell; intervals bootstrap grid cells. A quantity is *selective* when its interval excludes zero on the intended axis and includes zero on the orthogonal axis. Figure 2 shows the corresponding surfaces.

	Quantity	Intended axis	vs α (sharing)	vs β (coupling)	Selective
Diagnostic	g_{probe}	α	$+0.874\,[+0.765, +0.924]$	$+0.172\,[-0.187, +0.489]$	yes
	D_{logdet}	β	$+0.164\,[-0.176, +0.481]$	$+0.986\,[+0.972, +0.987]$	yes
	$\lvert\rho_y\rvert$	both	$+0.637\,[+0.393, +0.794]$	$+0.755\,[+0.577, +0.859]$	no
Effect	realized sharing gain	α	$+0.407\,[+0.001, +0.697]$	$-0.178\,[-0.496, +0.159]$	yes
	realized coupling gain	β	$+0.161\,[-0.177, +0.476]$	$+0.986\,[+0.972, +0.987]$	yes

B.1 Synthetic construct validation

Real-data construct validation defines mechanism strength through observed model outcomes: Δ_{reuse} and Δ_{couple} are target-loss contrasts whose values depend on the intervention and loss specification. The synthetic construction complements this analysis by controlling representation sharing and residual coupling generatively, allowing the diagnostics to be evaluated against known mechanism strengths rather than effects inferred from fitted configurations.

Construction. Each synthetic pair contains two independently varied mechanisms. The *representation* parameter $\alpha \in [0, 1]$ controls the contribution of a shared nonlinear feature map relative to task-specific maps: at $\alpha = 0$ the tasks use independent maps and share no latent representation, whereas at $\alpha = 1$ they use the same feature map and differ only through linear output weights with fixed cosine alignment 0.75. The *coupling* parameter $\beta \in [0, 1]$ controls the correlation between the two additive noise terms, inducing residual dependence that remains after conditioning on the input. The task signals are nonlinear functions of the raw input, generated through two-layer $\tanh$ networks of width 64 mapping $\mathbb{R}^d$ to $\mathbb{R}^k$. Recovering the shared component requires learning a representation rather than relying on a fixed random feature map.

Grid. We sweep $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ and $\beta \in \{0, 0.15, 0.3, 0.45, 0.6, 0.75\}$, yielding 36 grid cells. For each cell, we generate 20 independent datasets of $n = 6000$ examples in $d = 128$ dimensions. We evaluate six configurations: separate, frozen-random, and jointly trained representation states, each with independent or coupled outputs. All configurations use the same target loss as in the real-data experiments. Auxiliary-pretrained configurations are omitted because representation sharing is controlled directly by the generative parameter α rather than induced through auxiliary pretraining. Thus, unlike in the real-data experiments, both mechanism strengths are known by construction.

Diagnostics selectively recover the generative mechanisms. Because α and β vary independently, the grid tests whether each diagnostic responds selectively to its intended mechanism; Table 9 reports rank correlations across the 36 cells and Figure 2 shows the resulting surfaces. The separation is clear in both directions: g_{probe} tracks representation sharing at $+0.874$, with its coupling interval including zero, whereas D_{logdet} tracks residual coupling at $+0.986$, with its sharing interval including zero. The intervention-effect surfaces vary primarily along their intended axes, although the realized sharing effect is small relative to seed variation. This provides a ground-truth analogue of the construct test in §4, with mechanism strength fixed by construction rather than defined through a particular loss contrast. Absolute target correlation behaves differently, associating with both parameters ($+0.637$ with sharing and $+0.755$ with coupling) and varying diagonally across the grid. It is therefore predictive but not mechanism-selective, illustrating why predictive association alone does not determine which intervention is responsible.

As a hardware robustness check, we repeated the synthetic analysis on a second device. The D_{logdet}–β and realized-coupling–β correlations were unchanged to three decimals, whereas the g_{probe}–α

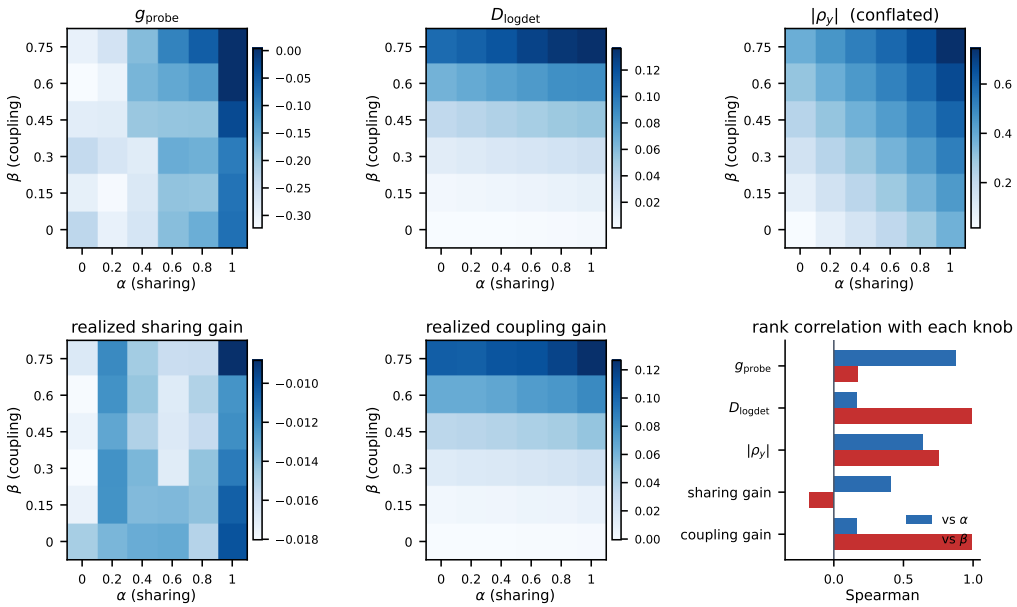

Figure 2: Construct validation with generative ground truth. Cell means over 20 seeds on the 6×6 (α, β) grid, with representation sharing on the horizontal axis and residual coupling on the vertical. g_{probe} varies primarily with α, whereas D_{logdet} varies primarily with β. The observed intervention effects vary along their corresponding axes, although the sharing effect is small relative to seed variation. Absolute target correlation responds to both axes and is not mechanism-selective. Sequential single-hue scales preserve within-sign variation; signed ranges are shown on the colourbars. Numerical correlations and intervals are reported in Table 9.

correlation decreased from $+0.874$ to $+0.680$. This is consistent with the greater hardware sensitivity of learned representation quantities discussed in §7.

What the construction does not establish. Although g_{probe} strongly tracks the generative sharing parameter ($+0.874$), the observed sharing gain changes by only $+0.005$ across the full α sweep, smaller than the typical across-seed standard error of 0.011. Sharing strength is rank-recoverable even though its realized intervention effect is small relative to experimental variation. By contrast, the observed coupling effect spans approximately 26 such standard errors, so both its dependence on β and its magnitude are clearly resolved. Thus, we use the synthetic grid to validate separation of the two generative mechanisms, not to estimate how often transfer should be selected. That question is addressed on the real domains in §5, where adapted-transfer configurations are available and have no direct analogue in this construction.

B.2 Full QM9 construct results and task-level precision

Table 10 reports the full QM9 construct analysis omitted from the main table. With 10 properties and 90 ordered pairs, five of the 18 associations have intervals excluding zero, including both pre-specified hypotheses: g_{probe}–Δ_{reuse} at $+0.625$ $[+0.045, +0.954]$ and D_{logdet}–Δ_{couple} at $+0.657$ $[+0.038, +0.932]$. The remaining associations are estimated too imprecisely to distinguish the diagnostics reliably.

The effective resampling unit is the task rather than the pair. Because pairs sharing a task endpoint are dependent, intervals are obtained by resampling task vertices and retaining dyads between distinct sampled tasks. Thus, 90 pairs formed from only 10 properties contain substantially less independent information than the pair count suggests. A bootstrap draw contains 6.5 distinct QM9 properties on average, compared with 22.3 CelebA attributes, and the median interval width is 1.05 on QM9 versus 0.46 on CelebA.

Table 10: QM9 diagnostic–intervention associations. Entries are within-target rank correlations; * denotes a vertex-bootstrap interval excluding zero. Five of the 18 associations resolve with only 10 task vertices, including both pre-specified hypotheses, g_{probe}–Δ_{reuse} and D_{logdet}–Δ_{couple}. All other comparisons are exploratory.

Construct	Score	Δ_{reuse}	Δ_{adapt} (compound)	Δ_{couple}
Label association	$\lvert\rho_y\rvert$	$+0.637^*$	-0.460	$+0.028$
Residual dependence	D_{logdet}	$+0.287$	-0.250	$+0.657^*$
Representation similarity	CKA	$+0.733^*$	-0.648^*	$+0.000$
Transfer evidence	ΔLogME	$+0.540$	-0.458	$+0.270$
Gradient compatibility	grad. cosine	$+0.332$	-0.335	$+0.157$
Direct representation-reuse probe	g_{probe}	$+0.625^*$	-0.518	$+0.287$

Table 11: Replication of the pre-specified g_{probe}–Δ_{reuse} association across the original eight-configuration and expanded twelve-configuration runs. * denotes a task-level vertex-bootstrap interval excluding zero. CelebA and regression-family estimates differ by at most 0.010; QM9 remains directionally consistent ($+0.625$ versus $+0.558$), although the expanded-run interval includes zero.

Domain	Original 8-configuration run	Expanded 12-configuration run
CelebA-35	$+0.564^*$ $[+0.294, +0.759]$	$+0.564^*$ $[+0.287, +0.751]$
QM9-10	$+0.625^*$ $[+0.045, +0.954]$	$+0.558$ $[-0.063, +0.943]$
Regression families	$+0.520^*$ $[+0.176, +0.744]$	$+0.530^*$ $[+0.187, +0.755]$

A subset analysis supports this interpretation. Across all $\binom{10}{8} = 45$ eight-property subsets, both primary associations remain positive. Their median point estimates, $+0.603$ for g_{probe}–Δ_{reuse} and $+0.643$ for D_{logdet}–Δ_{couple}, remain close to the full-pool estimates of $+0.625$ and $+0.657$. Yet their intervals exclude zero in only 9 and 7 of the 45 subsets, respectively. Removing task vertices affects inferential precision much more than the central association estimates. We accordingly treat QM9 as supporting evidence for the two pre-specified hypotheses rather than as a basis for comparing the full diagnostic set. Appendix F further examines sensitivity to the composition of the QM9 task pool.

B.3 Robustness of the diagnostic–intervention associations

We test whether the primary associations persist under three perturbations: an independent rerun, independently initialized encoders, and a 20×4 partition-by-initialization grid.

Independent replication and encoder initialization. Table 11 compares the pre-specified g_{probe}–Δ_{reuse} association across two independent reruns. The CelebA and regression estimates differ by at most 0.010. QM9 remains directionally consistent ($+0.625$ versus $+0.558$), although the second interval includes zero, consistent with the task-level precision limitation discussed above.

One possible concern is that g_{probe} and Δ_{reuse} are computed using the same auxiliary-pretrained encoder. To test whether their association depends on encoder-specific variation, Table 12 evaluates every ordered pairing of independently initialized encoders after averaging over data partitions. The off-diagonal associations retain 87%, 91%, and 85% of the same-encoder association on CelebA, QM9, and the regression families, respectively. The seed-averaged association also excludes zero in all three domains. Therefore, the g_{probe}–Δ_{reuse} association is not specific to a particular encoder initialization, although the regression result remains limited by the weak pair-level reliability of Δ_{reuse} discussed below.

Partition-by-initialization stability. Because the construct analysis also depends on the sampled data partition, Table 13 recomputes the primary associations separately in every cell of the 20×4 partition-by-initialization grid.

The primary associations remain stable under these perturbations. On CelebA, g_{probe}–Δ_{reuse} and D_{logdet}–Δ_{couple} have cell-level standard deviations of 0.051 and 0.050, respectively, and retain their signs in all 80 cells. The weaker associations in Table 13 show greater cell-to-cell variation. Thus,

Table 12: Cross-initialization robustness of the g_{probe}–Δ_{reuse} association. "Same encoder" pairs probes and effects computed from the same auxiliary-pretrained encoder initialization; "different encoder" averages the association over off-diagonal pairings of independently initialized encoders after averaging over data partitions. "Retained" is the different-encoder association as a percentage of the same-encoder association. "Initialization-averaged" correlates probes and effects after averaging each quantity over encoder initializations, removing initialization-specific variation. * denotes a task-level vertex-bootstrap interval excluding zero.

Domain	Same encoder	Different encoder	Retained	Initialization-averaged
CelebA	$+0.589$	$+0.513$	87%	$+0.581^{*}$ $[+0.088, +0.819]$
QM9	$+0.595$	$+0.539$	91%	$+0.625^{*}$ $[+0.045, +0.952]$
Regression families	$+0.529$	$+0.449$	85%	$+0.662^{*}$ $[+0.211, +0.890]$

Table 13: Stability of diagnostic–intervention associations across the 20×4 partition-by-initialization grid. We report g_{probe}, D_{logdet}, and $|\rho_y|$ against each intervention. Each entry summarizes the 80 correlations obtained by recomputing the association separately in every replicate cell. $[p_5, p_{95}]$ is the 5th–95th percentile range across cells, not a confidence interval. Mean and SD are the mean and standard deviation of the 80 per-cell correlations, and "Sign stable" is the percentage with the same sign as the across-cell mean. The primary g_{probe}–Δ_{reuse} and D_{logdet}–Δ_{couple} associations are sign-stable across all cells on CelebA and QM9, whereas several weaker associations show greater cell-to-cell variation.

Domain	Score	Effect	Mean	SD	$[p_5, p_{95}]$	Sign stable
CelebA	$\lvert\rho_y\rvert$	Δ_{reuse}	$+0.380$	0.049	$[+0.306, +0.459]$	100%
CelebA	$\lvert\rho_y\rvert$	Δ_{adapt} (compound)	-0.007	0.064	$[-0.098, +0.079]$	56%
CelebA	$\lvert\rho_y\rvert$	Δ_{couple}	$+0.371$	0.048	$[+0.288, +0.440]$	100%
CelebA	D_{logdet}	Δ_{reuse}	$+0.305$	0.050	$[+0.226, +0.387]$	100%
CelebA	D_{logdet}	Δ_{adapt} (compound)	$+0.031$	0.067	$[-0.094, +0.135]$	71%
CelebA	D_{logdet}	Δ_{couple}	$+0.375$	0.050	$[+0.286, +0.457]$	100%
CelebA	g_{probe}	Δ_{reuse}	$+0.518$	0.051	$[+0.426, +0.598]$	100%
CelebA	g_{probe}	Δ_{adapt} (compound)	-0.179	0.067	$[-0.296, -0.071]$	100%
CelebA	g_{probe}	Δ_{couple}	$+0.141$	0.061	$[+0.048, +0.246]$	100%
QM9	$\lvert\rho_y\rvert$	Δ_{reuse}	$+0.593$	0.054	$[+0.477, +0.654]$	100%
QM9	$\lvert\rho_y\rvert$	Δ_{adapt} (compound)	-0.394	0.079	$[-0.512, -0.251]$	100%
QM9	$\lvert\rho_y\rvert$	Δ_{couple}	$+0.086$	0.079	$[-0.041, +0.220]$	88%
QM9	D_{logdet}	Δ_{reuse}	$+0.235$	0.091	$[+0.096, +0.384]$	100%
QM9	D_{logdet}	Δ_{adapt} (compound)	-0.215	0.102	$[-0.372, -0.053]$	99%
QM9	D_{logdet}	Δ_{couple}	$+0.517$	0.088	$[+0.390, +0.665]$	100%
QM9	g_{probe}	Δ_{reuse}	$+0.588$	0.063	$[+0.500, +0.679]$	100%
QM9	g_{probe}	Δ_{adapt} (compound)	-0.468	0.096	$[-0.612, -0.306]$	100%
QM9	g_{probe}	Δ_{couple}	$+0.246$	0.072	$[+0.120, +0.349]$	100%
Regression families	$\lvert\rho_y\rvert$	Δ_{reuse}	$+0.156$	0.096	$[-0.006, +0.321]$	94%
Regression families	$\lvert\rho_y\rvert$	Δ_{adapt} (compound)	-0.163	0.086	$[-0.302, -0.037]$	99%
Regression families	$\lvert\rho_y\rvert$	Δ_{couple}	$+0.220$	0.092	$[+0.051, +0.349]$	100%
Regression families	D_{logdet}	Δ_{reuse}	$+0.051$	0.082	$[-0.070, +0.209]$	72%
Regression families	D_{logdet}	Δ_{adapt} (compound)	-0.066	0.081	$[-0.175, +0.064]$	79%
Regression families	D_{logdet}	Δ_{couple}	$+0.143$	0.102	$[-0.043, +0.303]$	94%
Regression families	g_{probe}	Δ_{reuse}	$+0.467$	0.082	$[+0.308, +0.571]$	100%
Regression families	g_{probe}	Δ_{adapt} (compound)	-0.426	0.091	$[-0.560, -0.281]$	100%
Regression families	g_{probe}	Δ_{couple}	$+0.145$	0.092	$[-0.003, +0.305]$	94%

the distinction between the primary and weaker exploratory associations persists after propagating partition and initialization variation in addition to task-level resampling.

B.4 Measurement reliability and interpretability

Testing whether a score predicts an intervention effect presupposes reproducible *between-pair* variation in both quantities. When either quantity is poorly distinguished across pairs, a near-zero as-

Table 14: Reliability of quantities used in the diagnostic–intervention and target-budget analyses. The variance components are reported as standard deviations, in each quantity's own units, and are therefore not comparable across rows or to the across-cell variation of an association in Table 13. R is the reliability of the replicate-cell mean under Eq. (7), with $m = 6$ cells except for QM9, where $m = 80$; intervals use task-level resampling. For diagnostics not independently re-estimated in every cell, such as D_{logdet} and $|\rho_y|$, $R \approx 1$ partly reflects the absence of cell-level re-estimation variation and should not be interpreted as perfect reliability under repeated sampling.

Domain	Budget	Quantity	$\hat{\sigma}_\theta$	$\hat{\sigma}_\varepsilon$	R	95% CI
CelebA	$n = 200$	g_{probe}	0.0218	0.0173	0.90	[0.77, 0.94]
		D_{logdet}	0.0199	0.0000	1.00	[1.00, 1.00]
		$\lvert\rho_y\rvert$	0.1093	0.0116	1.00	[1.00, 1.00]
		Δ_{reuse}	0.0463	0.0174	0.98	[0.96, 0.99]
		$\Delta_{\mathrm{adapt\lvert transfer}}$	0.0252	0.0219	0.89	[0.85, 0.91]
		Δ_{couple}	0.0099	0.0041	0.97	[0.91, 0.98]
CelebA	$n = 1000$	g_{probe}	0.0154	0.0100	0.93	[0.80, 0.96]
		D_{logdet}	0.0199	0.0000	1.00	[1.00, 1.00]
		$\lvert\rho_y\rvert$	0.1092	0.0114	1.00	[1.00, 1.00]
		Δ_{reuse}	0.0493	0.0116	0.99	[0.98, 0.99]
		$\Delta_{\mathrm{adapt\lvert transfer}}$	0.0489	0.0164	0.98	[0.97, 0.99]
		Δ_{couple}	0.0135	0.0041	0.98	[0.95, 0.99]
QM9	default	g_{probe}	0.0993	0.0485	1.00	[0.99, 1.00]
		D_{logdet}	0.0731	0.0093	1.00	[0.99, 1.00]
		$\lvert\rho_y\rvert$	0.2521	0.0106	1.00	[1.00, 1.00]
		Δ_{reuse}	0.2073	0.0591	1.00	[1.00, 1.00]
		$\Delta_{\mathrm{adapt\lvert transfer}}$	0.5618	0.2866	1.00	[0.99, 1.00]
		Δ_{couple}	0.0858	0.1022	0.98	[0.95, 1.00]
Regression families	default	g_{probe}	0.0987	0.2017	0.59	[0.31, 0.84]
		D_{logdet}	0.0562	0.0552	0.86	[0.75, 0.93]
		$\lvert\rho_y\rvert$	0.2987	0.0310	1.00	[1.00, 1.00]
		Δ_{reuse}	0.1180	0.6503	0.16	[0.00, 0.44]
		$\Delta_{\mathrm{adapt\lvert transfer}}$	0.1361	0.3972	0.41	[0.26, 0.56]
		Δ_{couple}	0.0930	0.1057	0.82	[0.70, 0.91]
VOC	$n = 200$	g_{probe}	0.0015	0.0123	0.09	[0.00, 0.31]
		D_{logdet}	0.0107	0.0000	1.00	[1.00, 1.00]
		$\lvert\rho_y\rvert$	0.0561	0.0086	1.00	[0.98, 1.00]
		Δ_{reuse}	0.0097	0.0172	0.65	[0.34, 0.78]
		$\Delta_{\mathrm{adapt\lvert transfer}}$	0.0130	0.0169	0.78	[0.65, 0.86]
		Δ_{couple}	0.0061	0.0041	0.93	[0.43, 0.94]
VOC	$n = 1000$	g_{probe}	0.0011	0.0073	0.13	[0.00, 0.39]
		D_{logdet}	0.0107	0.0000	1.00	[1.00, 1.00]
		$\lvert\rho_y\rvert$	0.0541	0.0087	1.00	[0.98, 1.00]
		Δ_{reuse}	0.0120	0.0083	0.93	[0.85, 0.95]
		$\Delta_{\mathrm{adapt\lvert transfer}}$	0.0146	0.0121	0.90	[0.70, 0.94]
		Δ_{couple}	0.0087	0.0034	0.98	[0.85, 0.98]

sociation is uninformative rather than evidence of absence. For a pairwise quantity measured over m replicate cells, we write $X_{pi} = \theta_p + \varepsilon_{pi}$, where p indexes task pairs and i replicate cells, and estimate the reliability of the replicate-cell mean as

$$R = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_\varepsilon^2/m}, \tag{7}$$

where σ_θ captures latent between-pair variation and σ_ε captures replicate-cell variation. Under classical additive measurement error, imperfect reliability attenuates Pearson correlations. We use R only to assess whether a near-zero *rank* association is meaningfully interpretable, rather than as an exact attenuation correction for the reported Spearman correlations. Table 14 reports R by domain and target-data budget.

The scores separate sharply. g_{probe} is reliable on CelebA at both target budgets (0.90 and 0.93) and on QM9 (1.00), but not on VOC (0.09 and 0.13, with both intervals including zero), where even at the larger budget its between-pair standard deviation is 0.0011 against 0.0073 of replicate

variation. Failure can also occur on the effect side: on the regression families, g_{probe} is moderately reliable (0.59), whereas Δ_{reuse} is not (0.16). This is not an artefact of pooling families with different loss scales: median within-family reliability is 0.19. Associations involving Δ_{reuse} on this domain therefore require the same caution as those involving g_{probe} on VOC.

Applying this criterion to our null results changes one conclusion and supports another. $\Delta_{\text{adapt}|\text{transfer}}$ is reliably measurable on CelebA (0.89) and QM9 (1.00), so null diagnostic associations in these domains cannot be attributed simply to an unreliable adaptation effect. However, the evidence is not uniformly null: an exploratory CKA association resolves on QM9, while no corresponding association resolves on CelebA. Thus, we treat the broader absence of a robust cross-domain diagnostic for adaptation after transfer as interpretable, rather than claiming that every evaluated diagnostic is null. On the regression families, $\Delta_{\text{adapt}|\text{transfer}}$ has reliability only $0.41\ [0.26, 0.56]$, so we do not interpret the corresponding null as evidence of absence.

The regression coupling null survives the same audit: $R(\Delta_{\text{couple}}) = 0.82$, while the pre-specified D_{logdet} diagnostic has $R = 0.86$. Here both quantities exhibit reproducible between-pair variation, so the unresolved association is not readily explained by attenuation from replicate-level measurement noise.

For diagnostics computed once per run and reused across replicate cells, such as D_{logdet} and $|\rho_y|$, $R \approx 1$ partly reflects the absence of cell-level re-estimation noise rather than independently demonstrated sampling reliability. These values rule out attenuation from the replicate variation represented here, but should not be interpreted as universal reliability estimates.

Measurement reliability is thus a precondition for the chain in §1, not an additional link. A weak end-to-end affinity–gain association can otherwise conflate distinct failures: the score may not vary reproducibly across pairs; a reliable score may fail to track its intended effect; or a measured effect may fail to determine the preferred configuration. Only the latter two support substantive conclusions about the score or the resulting decision.

B.5 Controlled adaptation

The original eight-configuration design uses the compound contrast $\Delta_{\text{adapt}} = L(\mathsf{fpt}) - L(\mathsf{joint})$, which changes both adaptation and initialization. The expanded design permits the controlled contrast $\Delta_{\text{adapt}|\text{transfer}} = L(\mathsf{fpt}) - L(\mathsf{ptft})$, which holds the transferred initialization fixed and changes only whether it is adapted.

Table 15 shows no resolved diagnostic association on CelebA. On QM9, one exploratory association resolves: CKA correlates with $\Delta_{\text{adapt}|\text{transfer}}$ at -0.633, whereas the pre-specified g_{probe} association is -0.440 with an interval including zero. For g_{probe}, the corresponding CelebA association is -0.157, also unresolved. Because both g_{probe} and $\Delta_{\text{adapt}|\text{transfer}}$ are reliably measurable on CelebA and QM9 (Appendix B.4), these unresolved g_{probe} associations cannot be attributed simply to limited measurement reliability. On the regression families, the g_{probe} association is -0.342, but the controlled adaptation effect has only moderate reliability ($R = 0.41$), so we do not interpret that null.

For g_{probe}, replacing the compound contrast with the controlled one changes the association little on CelebA (-0.143 versus -0.157) or QM9 (-0.435 versus -0.440). Thus, for this diagnostic, the initialization change in Δ_{adapt} does not explain the observed association pattern in these two domains. On the regression families, the component associations are descriptive only because the controlled adaptation effect is not sufficiently reliable. We therefore retain Δ_{adapt} as a compound contrast and find no robust cross-domain evidence that diagnostics of representation reuse predict the incremental value of adapting the same transferred initialization.

B.6 Pre-specified hypotheses and exploratory comparisons

Although the construct analysis evaluates many diagnostic–intervention associations, two hypotheses were pre-specified as primary: g_{probe}–Δ_{reuse} and D_{logdet}–Δ_{couple}. Each pairs a diagnostic with the intervention it was intended to track. On CelebA, the corresponding associations are $+0.564$ and $+0.487$, and both intervals exclude zero. Both are also positive and resolved on QM9, at $+0.625$ and $+0.657$. On the regression families, the g_{probe}–Δ_{reuse} association is $+0.520$ with an interval

Table 15: Controlled adaptation, $\Delta_{\text{adapt}|\text{transfer}} = L(\mathsf{fpt}) - L(\mathsf{ptft})$, holds the transferred initialization fixed and varies only whether it is adapted. Entries are within-target Spearman correlations from the expanded runs; * denotes a task-level interval excluding zero. No association resolves on CelebA. On QM9, CKA is the only resolved association (-0.633); this comparison is exploratory. The controlled adaptation effect is reliably measurable on CelebA and QM9 ($R = 0.89$ and 1.00), whereas its reliability is only 0.41 on the regression families, so unresolved regression associations are not interpreted as evidence of absence (Table 14).

Score	CelebA-35	QM9-10	Regression families
$\lvert\rho_y\rvert$	-0.083	-0.537	-0.159
D_{logdet}	-0.043	-0.188	-0.137
CKA	$+0.224$	-0.633^*	-0.157
ΔLogME	$+0.288$	-0.378	-0.207
g_{probe}	-0.157	-0.440	-0.342

excluding zero, whereas the D_{logdet}–Δ_{couple} association does not resolve. However, Δ_{reuse} has low reliability on the regression families ($R = 0.16$; Appendix B.4), so we do not treat the former as equally strong evidence. CelebA therefore provides the clearest real-data test of the g_{probe}–Δ_{reuse} hypothesis.

Across the full diagnostic–intervention analysis, 15 associations have vertex-bootstrap intervals excluding zero. These are not treated as a family of confirmatory discoveries: comparisons other than the two primary hypotheses are reported with intervals and interpreted as exploratory. Our primary construct conclusions rest on the two pre-specified diagnostic–intervention hypotheses above, together with their robustness and measurement analyses, rather than on the number of nominally resolved exploratory associations.

C Operationalization of measured intervention effects

§4 examines two sources of operationalization dependence: estimation of likelihood nuisance parameters and the representation state at which coupling is evaluated. Here we report the full analyses and, where possible, compare alternative evaluation metrics.

C.1 Calibration and likelihood operationalization

Because the contrasts in Eq. (2) are defined using likelihood loss, nuisance parameters must be estimated independently of evaluation. Throughout the paper, we estimate them on split B and evaluate on disjoint split C.

For the *same* trained CelebA models, uncalibrated likelihood indicates that encoder sharing hurts on 90% of pairs, with median effect -0.385. Estimating the temperature on split B shifts the median to $+0.053$ and changes the preferred configuration on 69% of pairs. The trained models, initializations, and data partitions are unchanged.

Temperature scaling (Guo et al., 2017) divides logits by a positive scalar and is therefore monotone, leaving ROC-AUC and prediction ranks unchanged. The reversal thus reflects a change in probabilistic calibration rather than discrimination. Because temperature scaling changes the predictive distribution itself, we interpret this as dependence on nuisance estimation within the likelihood evaluation pipeline, rather than disagreement between two metrics applied to the same predictive distribution.

C.2 Sensitivity to task-native metrics

The calibration result raises a separate question: do intervention conclusions persist under task-native metrics that do not depend on the likelihood nuisance parameter? Adaptation permits such a comparison because its independent-output endpoints can be evaluated directly with task-native metrics. On CelebA, the NLL-based adaptation contrast correlates with the corresponding AUC

Table 16: Mean Δ_{couple} by representation state under NLL. The final row reports the absolute range across states.

Representation state	CelebA	QM9	Regression families
Separate	$+0.0007$	$+0.0187$	$+0.0011$
Frozen random	$+0.0131$	$+0.0982$	$+0.0221$
Frozen pretrained	$+0.0014$	$+0.0085$	$+0.0147$
Jointly trained	$+0.0029$	$+0.0336$	$+0.0131$
Range (max − min)	0.0124	0.0897	0.0210

Table 17: Mean coupling gain under NLL and a second proper scoring rule: Brier score for CelebA and CRPS for the continuous domains. NLL is recomputed on the same reduced pair pool used for the alternative score.

	CelebA		QM9		Regression families	
Representation state	NLL	Brier	NLL	CRPS	NLL	CRPS
Separate	$+0.0005$	$+0.0002$	$+0.0211$	$+0.0030$	$+0.0142$	$+0.0050$
Frozen random	$+0.0177$	$+0.0069$	$+0.1019$	$+0.0300$	$+0.0371$	$+0.0130$
Frozen pretrained	$+0.0012$	-0.0021	$+0.0085$	$+0.0024$	$+0.0351$	$+0.0110$
Jointly trained	$+0.0031$	-0.0057	$+0.0395$	$+0.0034$	$+0.0310$	$+0.0083$
Spearman ρ	$+0.20$		$+1.00$		$+1.00$	

contrast at $\rho = +0.87$ and agrees in sign on 83% of pairs. On QM9, the R^2- and NLL-based contrasts both indicate beneficial adaptation for every pair.

The same comparison does not apply directly to residual coupling, which changes the conditional predictive distribution rather than only a point prediction or ranking. We assess coupling below using a second proper scoring rule defined on the full predictive distribution.

Metric robustness is level-dependent. Recording task-native loss for every configuration (Appendix E.4) allows metric robustness to be examined at three levels. At the level of the *intervention effect*, the NLL and AUC adaptation contrasts agree strongly ($\rho = +0.87$). At the level of the *preferred action*, the best fixed configuration changes from ptft under NLL to joint under $1 - \text{AUC}$. At the level of *deployment value*, the task-extrapolation routing gain that resolves under the log score is unresolved under $1 - \text{AUC}$ ($+0.0014\,[-0.0001, +0.0034]$).

Thus, metric robustness at the intervention-effect level does not imply robustness at the action or deployment levels: configuration selection compares all available actions simultaneously, rather than one intervention contrast in isolation.

Together with the calibration result above, evaluation enters the decision in two distinct ways. First, *nuisance estimation within a metric* can change a likelihood-based effect while leaving discrimination unchanged. Second, the *choice of metric* can change the preferred configuration and deployment conclusion even when the corresponding intervention contrast remains stable. These are distinct forms of operationalization dependence rather than alternative explanations of the same sensitivity.

C.3 Representation-state dependence of coupling

We evaluate $\Delta_{\text{couple}}(p \mid r)$ at each representation state (Table 16) and repeat the comparison under a second proper scoring rule: Brier score for CelebA and CRPS for the continuous domains (Table 17).

Coupling gain depends on representation state. Under NLL, frozen-random yields the largest mean gain in all three domains, with ranges across states of 0.0124 on CelebA, 0.0897 on QM9, and 0.0210 on the regression families. The diagnostic–effect association also changes with state: on CelebA, mean coupling gain rises from $+0.0007$ with separate encoders to $+0.0131$ under frozen-random representations, where D_{logdet} correlates $+0.706$ with Δ_{couple}.

The state dependence is largely preserved under the second proper scoring rule (Table 17). For QM9 and the regression families, the representation-state ordering is identical to that under NLL (Spearman $\rho = +1.00$ in both). On CelebA, frozen-random remains the state with the largest coupling gain, but the remaining ordering changes: under Brier score, coupling is harmful for the frozen-pretrained and jointly trained states despite improving NLL.

D MEASURED EFFECT → PREFERRED ACTION: DECISION-CONTEXT SENSITIVITY

§5 varies six aspects of the decision problem: the available actions, evaluation loss, treatment of near-ties, competitor strength, backbone, and target-data budget. We first give the construction underlying Proposition 1, then report these six controls in that order.

What is held fixed differs by control. Within each matched comparison, the task pool and ordered pair set are held fixed. Candidate-set, near-tie, and competitor-strength controls reuse the same diagnostics, whereas the backbone and target-data controls require recomputing representation-dependent quantities. For example, CelebA g_{probe} values across the two target budgets correlate at 0.855. We treat these as matched comparisons over the same task pairs, not as isolated interventions in which all other measured quantities are held numerically fixed.

D.1 WHY PREDICTIVE VALIDITY DOES NOT IMPLY DECISION VALUE

Each case is an explicit population construction. Let s denote the score, Δ its intended intervention effect, and $L(p, a)$ the loss of configuration a on pair p. Headroom and regret are defined as in Eq. (4), using the pointwise oracle loss $L_{\min}(p) = \min_{a \in \mathcal{A}} L(p, a)$ rather than the noise-aware preferred action $a^*(p)$. This keeps the construction independent of the parsimony rule. The pair-level empirical analyses estimate the best fixed configuration by leave-one-pair-out evaluation; the constructions below use the corresponding population quantities.

(i) Nothing to win. Let $\mathcal{A} = \{a_1, a_2\}$, $u \sim \text{Unif}[0, 1]$, and $L(p, a_1) = 1 + u_p, \qquad L(p, a_2) = 1.$ Set $s(p) = u_p$ and $\Delta(p) = L(p, a_1) - L(p, a_2) = u_p$, so $\text{corr}(s, \Delta) = 1$. Since $\Delta \geq 0$ everywhere and $\Delta > 0$ almost surely, a_2 is always optimal, and uniquely so almost surely, giving $H = 0$. By Eq. (4), every router has $G(\pi) = -R(\pi) \leq 0$. Perfect prediction of an intervention effect therefore need not create any pair-specific selection opportunity.

(ii) Something to win that this score cannot win. Let $u, v \in \{-1, +1\}$ be independent Rademacher variables and $\mathcal{A} = \{a_1, \ldots, a_4\}$, with $L(p, a_1) = 10, \qquad L(p, a_2) = 10 - u_p, \qquad L(p, a_3) = 1 + v_p, \qquad L(p, a_4) = 1.02 - v_p$. Again set $s = u$ and $\Delta = L(\cdot, a_1) - L(\cdot, a_2) = u$, so $\text{corr}(s, \Delta) = 1$. The oracle is determined entirely by v, which is independent of s. Hence s provides no information about whether a_3 or a_4 is oracle-optimal. Every rule in Π_s is a function of u alone, and conditionally on u the four expected losses are 10, $10 - u$, 1, and 1.02, so a_3 is the conditionally best choice for either value of u, and every such rule has expected loss at least $\mathbb{E}[L(a_3)] = 1$. Hence $R(\pi) \geq H$ for every $\pi \in \Pi_s$, with equality for the rule that always chooses a_3: $\min_{\pi \in \Pi_s} R(\pi) = H = 0.990$, and hence $\max_{\pi \in \Pi_s} G(\pi) = 0$ despite positive headroom.

Scaling all losses by any $c > 0$ makes $H = 0.990c$ without changing the oracle, the best s-measurable rule, or $\text{corr}(s, \Delta)$. Thus, arbitrarily large headroom can coexist with zero achievable gain from a score that perfectly predicts its named effect.

(iii) The same score, one configuration more. Return to case (i) and add $L(p, a_3) = 1.4 - u_p$, leaving the original losses, score, and intervention unchanged. Now a_2 is optimal for $u_p \leq 0.4$ and a_3 for $u_p > 0.4$. The s-measurable rule that makes exactly this choice matches the oracle, so $R = 0$ and the routing gain equals the full headroom, $G = H = 0.080$. Comparing (i) and (iii) isolates augmentation of the candidate set: the score, its named intervention effect, and their perfect association are unchanged, yet decision value becomes positive after adding one configuration. Table 18 summarizes the three instances.

Table 18: Predictive validity does not imply decision value. In all three population constructions, the score is perfectly correlated with its named intervention effect, and π_s^* is the best s-measurable rule. Case (i) has no pair-specific headroom; case (ii) has positive headroom that is unrelated to s. Adding one configuration to case (i) yields case (iii), where the same score and intervention effect support positive routing gain.

Case	$\lvert\mathcal{A}\rvert$	$\mathrm{corr}(s, \Delta)$	H	$R(\pi_s^*)$	$\max_{\pi \in \Pi_s} G(\pi)$	Reason
(i) Nothing to win	2	$+1.000$	0.000	0.000	$+0.000$	Δ never changes sign, so one configuration is optimal for every pair
(ii) Headroom unrelated to s	4	$+1.000$	0.990	0.990	$+0.000$	the oracle variation is determined by v, which is independent of s
(iii) One configuration added	3	$+1.000$	0.080	0.000	$+0.080$	case (i) plus one configuration whose optimality is determined by s

Avoiding exact-tie artefacts. The constructions avoid exact ties between fixed configurations. Under leave-one-pair-out estimation, a tie in the sample can induce anti-selection: removing one pair may favor the configuration that is worse on that same pair, inflating estimated headroom. Exact population ties do not occur in the measured losses used elsewhere, so we exclude them here to keep the construction about decision geometry rather than estimator behaviour.

D.2 Candidate-set dependence and headroom

The constructive comparison of adding adapter and pretrained-then-adapted configurations to the original eight-configuration design is reported in §5 and Table 3. It is computed within the expanded run by restricting the candidate set to the original eight configurations. A separately executed eight-configuration run gives essentially the same result: frozen-transfer selection is identical on CelebA and QM9 and differs by less than half a percentage point on the regression families. Therefore, the candidate-set effect is not explained by rerun variation.

x-only versus coupled action spaces. Because coupled configurations additionally require y_a at evaluation, we recompute preferred-action selection and headroom over the six x-only configurations $\mathcal{A}_6^{\mathrm{ind}}$, the six coupled configurations $\mathcal{A}_6^{c}$, and all twelve. We hold ϵ fixed at its $\mathcal{A}_{12}$ value so that only candidate availability changes.

Transfer is selected in $\mathcal{A}_6^{\mathrm{ind}}$ on 25.0%, 15.6%, and 25.4% of CelebA, QM9, and regression pairs, versus 25.9%, 21.1%, and 30.0% under $\mathcal{A}_{12}$. The coupled-only space gives similar rates of 26.1%, 21.1%, and 31.7%. CelebA headroom also remains comparable under the x-only restriction (0.0229 versus 0.0206).

Because restricting the candidate set changes both the pointwise oracle loss $L_{\min}(p)$ and the best fixed configuration, these headroom values show only that removing coupled actions does not eliminate pair-specific opportunity; they do not imply that restriction improves selection.

Subtractive candidate-subset sweep. Table 19 examines eight pre-specified subsets of the original eight-configuration run. This differs from the $\mathcal{A}_8$ row of Table 3, which restricts the expanded twelve-configuration run to the same eight candidate types. Seven subsets contain frozen transfer; the remaining subset excludes both frozen representation states and is retained only for its headroom.

Subsets in which frozen transfer is selected frequently generally provide less pair-specific opportunity. Three of the four domain–subset combinations with frozen-transfer selection above 20% have $H < \epsilon$, and CelebA's highest-headroom subset selects frozen transfer on only 5% of pairs. Descriptively, frozen-transfer selection and normalized headroom are negatively associated across candidate subsets ($\rho = -0.77$ on CelebA and -0.43 on the regression families).

Thus, restricting the candidate set can increase how often frozen transfer is preferred without increasing the value of pair-specific selection. This contrasts with the constructive expansion in §5, which adds interventions absent from the original action space.

Table 19: Candidate-set sensitivity within the original eight-configuration run. Each row is a pre-specified subset of the same trained configurations. Values may differ from the $\mathcal{A}_8$ row of Table 3, which restricts the separately trained twelve-configuration run to the same candidate types. "sel." is the percentage of pairs whose noise-aware preferred action uses frozen transfer, and H/ϵ is the corresponding noise-normalized oracle headroom. Frozen transfer is never preferred on QM9; on CelebA and the regression families, higher frozen-transfer selection generally coincides with lower pair-specific headroom.

Candidate subset	$\lvert\mathcal{A}\rvert$	CelebA		QM9		Regression families	
		sel.	H/ϵ	sel.	H/ϵ	sel.	H/ϵ
Full set	8	2%	2.7	0%	4.8	9%	1.0
Without frozen-random	6	16%	2.1	0%	4.8	10%	1.0
Shared-representation only	6	2%	2.6	0%	0.7	25%	0.6
Diagonal subset	4	5%	7.0	0%	3.6	8%	0.6
Coupled-output only	4	2%	2.7	0%	4.8	10%	0.8
Without frozen representations	4	–	1.8	–	4.8	–	0.8
Frozen transfer vs. separate	4	43%	1.9	0%	0.1	11%	0.8
Frozen transfer vs. joint	4	49%	0.7	0%	0.7	32%	0.6

Spread is not headroom. Candidate losses may be widely dispersed without creating pair-specific selection opportunity. Define

$$S = \mathbb{E}_p\left[\max_{a\in\mathcal{A}} L(p,a) - \min_{a\in\mathcal{A}} L(p,a)\right].$$

Unlike H, S measures overall dispersion among candidate losses. For example, expanding one QM9 candidate set from four to six configurations increases S by $6.73\times$ while leaving H at 0.0674, because the added configurations are never oracle-preferred. Candidate dispersion therefore does not itself imply pair-specific selection value.

D.3 Near-tie and precision sensitivity

Near-tie ordering. The noise-aware preferred action in Eq. (3) selects the simplest configuration within ϵ of the minimum observed loss. Here, complexity denotes an ordinal parsimony rule used only to resolve near-ties, rather than parameter count or computational cost. Our default ordering is

$$\mathsf{sep} \prec \mathsf{sep_}c \prec \mathsf{fr} \prec \mathsf{fr_}c \prec \mathsf{fpt} \prec \mathsf{fpt_}c \prec \mathsf{adp} \prec \mathsf{adp_}c \prec \mathsf{ptft} \prec \mathsf{ptft_}c \prec \mathsf{joint} \prec \mathsf{joint_}c.$$

Representation states are ordered by increasing intervention on the shared representation, and each coupled variant is placed immediately above its independent counterpart. In particular, adp lies between fpt and ptft because it adds task-specific adaptive capacity while keeping the shared encoder frozen.

Table 20 evaluates five alternative orderings, including minimum-loss selection ($\epsilon = 0$). Frozen-transfer selection changes by at most two percentage points on CelebA and three on the regression families and remains zero on QM9. The frozen-transfer conclusion is therefore insensitive to the near-tie ordering.

Tolerance ϵ. The default tolerance is the median, over pairs and configurations, of the standard error of pair-level loss across replicate cells. Sweeping from $\epsilon = 0$ to four times this replicate-scale reference moves frozen-transfer selection from 1.6% to 5.3% on CelebA and from 3.2% to 9.8% on the regression families, while QM9 remains at 0%.

This wide sweep intentionally varies the precision requirement; it does not test whether using one global tolerance obscures pair-level heterogeneity. To test that possibility, we replace ϵ with each pair's median standard error, ϵ_p. Frozen-transfer selection becomes 1.9%, 0%, and 7.7%, while adapted-transfer selection changes only from 24.7% to 23.9% on CelebA, 21.1% to 20.0% on QM9, and 23.7% to 24.1% on the regression families.

This stability holds despite substantial heterogeneity in ϵ_p: the 5th–95th percentile spans differ by factors of 3.8, 10.1, and 12.8 across the three domains, with the regression values ranging from

Table 20: Sensitivity of frozen-transfer selection to the near-tie rule. For the first four rows, only the complexity ordering among configurations within ϵ of the minimum loss is varied; losses, tolerance, and candidate set remain fixed. The final row removes the parsimony rule and selects the minimum-loss configuration directly. Frozen-transfer selection changes by at most two percentage points on CelebA and three on the regression families, and remains zero on QM9.

Tie-breaking rule	CelebA	QM9	Regression families
Default ordering	2%	0%	9%
Additive complexity	3%	0%	10%
Higher pretraining cost	2%	0%	8%
Lower coupling cost	3%	0%	10%
Minimum loss ($\epsilon = 0$)	2%	0%	10%

Table 21: Sensitivity of adapted-transfer selection to the near-tie tolerance ϵ. Columns sweep from minimum-loss selection ($\epsilon = 0$) to four times the default replicate-scale tolerance; "Span" is the maximum minus minimum selection rate in percentage points. Adapted-transfer preference is strongly precision-dependent: it is relatively stable on CelebA, increases on QM9, and falls from 40.9% to 0.8% on the regression families. Increasing ϵ can either admit ptft relative to a more complex winner or displace it when a simpler configuration becomes near-optimal. In the matched CelebA-12 backbone control, the small CNN has higher adapted-transfer selection than ResNet-18 at every tolerance, including $\epsilon = 0$.

Domain	$0\times$	$0.5\times$	$1\times$	$2\times$	$3\times$	$4\times$	Span
CelebA	23.8%	24.2%	24.7%	24.1%	23.1%	20.2%	4.5
QM9	16.7%	18.9%	21.1%	27.8%	31.1%	32.2%	15.5
Regression families	40.9%	35.6%	23.7%	6.6%	1.4%	0.8%	40.1
CelebA-12, small CNN	29.5%	28.8%	30.3%	35.6%	38.6%	36.4%	9.8
CelebA-12, ResNet-18	7.6%	7.6%	6.8%	10.6%	14.4%	16.7%	9.9

0.033 to 0.355 around a global ϵ of 0.070. Thus, the sensitivity observed in the broad sweep reflects changing the imposed precision requirement rather than heterogeneity hidden by the global tolerance. Table 21 reports the corresponding sweep for adapted transfer, whose selection rate changes more substantially and in opposite directions across domains.

Why the slopes differ. Increasing ϵ enlarges $\mathcal{A}_\epsilon(p)$. Adapted transfer can therefore gain a selection when it becomes near-optimal relative to a more complex winner, or lose one when a simpler configuration becomes eligible.

At 2ϵ, these flows nearly cancel on CelebA (77 enter, 73 leave), are entirely inward on QM9 (10, 0), and almost entirely outward on the regression families (2, 219), where 161 pairs revert to separate encoders. Every gain comes from a more complex configuration and every loss moves to a simpler one. The opposite slopes therefore arise from the same parsimony rule acting on different local candidate geometries.

A common replication budget. Tolerance depends on both domain variability and replicate count: six cells for CelebA and the regression families, but eighty for QM9. This difference in replication accounts for much of the gap between the ten-cell QM9 rerun (31.1% adapted-transfer selection) and the published eighty-cell result (21.1%).

As a comparability check, we estimate the standard deviation $\hat{\sigma}_{p,a}$ of each pair–configuration loss across all available cells and recompute the tolerance at a common six-cell budget,

$$\epsilon^{(6)} = \text{median}_{p,a} \frac{\hat{\sigma}_{p,a}}{\sqrt{6}}. \tag{8}$$

Each domain therefore uses all available replicates to estimate $\hat{\sigma}_{p,a}$, while the implied tolerance is evaluated at the same nominal replication budget. This moves QM9 from 21.1% to 32.2% adapted-transfer selection and widens the cross-domain range from 21.1–24.7% to 23.7–32.2% (Table 22).

Table 22: Adapted-transfer selection under a common nominal replication budget. Because ϵ depends on both variability and replicate count, we estimate pair-level variability from all available cells and recompute the same tolerance statistic assuming six replicate cells, denoted $\epsilon^{(6)}$. This comparability check moves QM9 from 21.1% to 32.2%, widening rather than eliminating the cross-domain difference. Thus, some of the apparent agreement under the default domain-specific tolerances reflects unequal replication. This is a comparability check, not a correction; minimum-loss selection and the full tolerance sweep in Table 21 remain the primary sensitivity analyses.

Domain	Cells	Default ϵ	Minimum loss	At default ϵ	$\epsilon^{(6)}$	At $\epsilon^{(6)}$
CelebA	6	0.00664	23.8%	24.7%	0.00664	24.7%
QM9	80	0.00582	16.7%	21.1%	0.02125	32.2%
Regression families	6	0.07034	40.9%	23.7%	0.07034	23.7%

Some apparent cross-domain agreement under the default tolerances therefore reflects unequal replication. This is a comparability check, not a correction; minimum-loss selection and the full tolerance sweep remain the primary sensitivity analyses.

D.4 Competitor strength and optimization budget

Pretrained-then-adapted configurations receive auxiliary pretraining before target adaptation and therefore more optimizer updates than the original jointly trained competitor. To test whether this advantage affects configuration selection, we add an update-matched joint configuration, joint2x, trained from random initialization for twice as long. The control covers 1190 CelebA pairs in six replicate cells, 90 QM9 pairs in ten, and 470 regression pairs in six. All comparisons below are computed within these matched control runs and therefore need not equal rates from the main sweeps.

Strengthening the joint competitor has markedly different effects across domains. Adapted-transfer selection changes from 24.4% to 21.2% on CelebA, from 31.1% to 6.7% on QM9, and from 31.9% to 32.6% on the regression families. Although joint2x improves over the original joint model on 88.4%, 87.8%, and 66.8% of pairs, respectively, the improvement exceeds the replicate-scale tolerance on only 60.7%, 80.0%, and 0.6%. Correspondingly, ptft still outperforms joint2x on 38.7%, 12.2%, and 56.6% of pairs. Thus, additional joint optimization removes most adapted-transfer preference on QM9, but changes the CelebA and regression conclusions little. On the twelve-attribute CelebA subset used for the backbone control, the same intervention changes adapted-transfer selection from 33.3% to 27.3%, indicating that the CelebA result is not specific to the full attribute pool.

The control matches optimizer updates rather than total computation or example exposure. ptft and joint2x receive the same number of updates, but on CelebA and the regression families the two strategies process different mixtures and numbers of training examples; for example, joint2x receives about 28% more example presentations on CelebA. We therefore describe the comparison as update-matched, not compute-matched. Where exposure differs, the strengthened joint competitor receives at least as much exposure as adapted transfer.

To examine where stronger joint optimization changes the decision, define $\Delta_{\text{opt}} = L(\text{joint}) - L(\text{joint2x})$, and compare it with the residual advantage $L(\text{joint2x}) - L(\text{ptft})$. On QM9, these quantities are negatively associated within target, $\rho = -0.568\,[-0.790, -0.335]$: pairs benefiting more from additional joint optimization tend to retain less advantage for adapted transfer. CelebA shows no resolved within-target association ($-0.063\,[-0.175, +0.053]$), but a negative between-target association ($-0.516\,[-0.711, -0.242]$), suggesting that the variation occurs primarily across targets. On the regression families, additional joint optimization exceeds the tolerance on only 3 of 470 pairs, leaving little variation to resolve ($\rho = -0.101\,[-0.455, +0.304]$).

Overall, competitor optimization is an important part of the decision context on QM9, where strengthening joint training removes most of the apparent preference for adapted transfer. The same control has much smaller effects on CelebA and the regression families, so optimization budget alone does not explain the cross-domain pattern.

Table 23: Backbone comparison on twelve aligned CelebA attributes. Frozen- and adapted-transfer columns report noise-aware preferred-action selection rates; $\Delta_{\text{adapt}|\text{transfer}}$ measures adaptation from the same transferred initialization. Absolute NLL is descriptive because the two runs are not hardware-matched.

Backbone	Best-fixed NLL	g_{probe}–Δ_{reuse}	Frozen transfer	Adapted transfer	H	H/ϵ	Best fixed	$\Delta_{\text{adapt}\|\text{transfer}}$
Small CNN	0.4647	$+0.530\,[-0.09, +0.87]$	4.5%	30.3%	0.0171	2.4	ptft_c	$+0.0226$
ResNet-18	0.4503	$+0.425\,[-0.18, +0.87]$	7.6%	6.8%	0.0134	1.2	joint_c	-0.0005

Table 24: Target-data budget control on fixed CelebA and VOC task pools. Selection rates pool independent and residual-coupled variants by representation state. The first six selection columns use the run-specific ϵ; the final two use minimum-loss selection ($\epsilon = 0$). "Any" denotes either frozen or adapted transfer.

Domain	n_{target}	Mean Δ_{reuse}	Separate	Frozen-rnd	Frozen-pre	Adapted	Joint	Any	Separate	Any
					run-specific ϵ				$\epsilon = 0$	
CelebA	200	-0.0097	7.9%	59.7%	1.2%	24.7%	6.6%	25.9%	3.9%	24.9%
	1000	$+0.0288$	40.3%	24.7%	6.0%	22.5%	6.5%	28.5%	25.8%	34.9%
VOC	200	-0.0231	24.2%	74.7%	0.0%	1.1%	0.0%	1.1%	18.1%	4.9%
	1000	$+0.0060$	62.6%	25.3%	8.8%	3.3%	0.0%	12.1%	40.7%	25.8%

D.5 Backbone dependence

§5 reports the headline result. We repeat the analysis with ResNet-18 on the same twelve CelebA attributes, splits, and initializations as the small-CNN experiment, so task pairs align exactly. The runs differ intentionally in backbone but were executed separately and are not hardware-matched.

Table 23 shows that the measured reuse relation is relatively stable across backbones: the g_{probe}–Δ_{reuse} association remains positive and of similar magnitude ($+0.530$ for the small CNN and $+0.425$ for ResNet-18). The preferred action changes more substantially. Adapted-transfer selection falls from 30.3% to 6.8%, while $\Delta_{\text{adapt}|\text{transfer}}$ falls from $+0.0226$ to approximately zero. Oracle headroom remains above the replicate-scale reference in both runs, although H/ϵ decreases from 2.4 to 1.2.

The reduced task pool does not explain the selection shift: adapted-transfer selection for the small CNN is 30.3% on the twelve-attribute subset, close to 24.7% on the full CelebA task pool. The shift also persists across the full tolerance sweep. Pair-interpolation routing improves over the best fixed configuration for both backbones, whereas task extrapolation is unresolved on this subset.

We do not interpret the lower ResNet-18 mean NLL (0.4503 versus 0.4647) as evidence of superior absolute performance because the runs are not device-matched and the 0.0144-nat difference lies within the documented 0.064-nat hardware variation. The within-run contrasts are more informative: the reuse association retains its direction, whereas the benefit and selection frequency of adapted transfer largely disappear under ResNet-18.

We therefore treat this result as a model-regime qualification rather than evidence that greater capacity generally removes transfer value. The comparison covers one domain, twelve attributes, and two backbones, and does not isolate capacity, optimization, or architectural inductive bias.

D.6 Target-data budget

Table 24 tests whether the preferred configuration changes with target-data availability. We compare $n_{\text{target}} \in \{200, 1000\}$ on fixed task pools: the same 35 CelebA attributes and 14 VOC classes are retained at both budgets, so the ordered task pairs do not change.

Increasing n_{target} from 200 to 1000 changes both the realized reuse effect and the preferred action. Mean Δ_{reuse} changes sign in both domains, from -0.0097 to $+0.0288$ on CelebA and from -0.0231 to $+0.0060$ on VOC. At the same time, separate-encoder selection rises from 7.9% to 40.3% on CelebA and from 24.2% to 62.6% on VOC.

This is not a simple shift away from transfer. Any-transfer selection also increases, from 25.9% to 28.5% on CelebA and from 1.1% to 12.1% on VOC; the configuration displaced most strongly is frozen-random sharing. Thus, stronger realized reuse can coexist with a substantial change in which configuration is preferred.

The comparison matches optimizer-update counts rather than target-example exposure. The auxiliary set remains fixed at 4000 examples and determines the number of optimizer steps, whereas increasing n_{target} necessarily increases target-example exposure. Diagnostics and ϵ are recomputed at each budget. The selection shift is not explained by the tolerance: ϵ changes only from 0.0066 to 0.0060 on CelebA and from 0.0060 to 0.0051 on VOC, and the increase in separate-encoder selection persists under minimum-loss selection ($\epsilon = 0$).

VOC is used only as a target-budget control. Because g_{probe} has little reproducible between-pair variation on VOC (Table 14), its observed g_{probe}–Δ_{reuse} associations ($+0.148$ at $n_{\text{target}} = 200$ and $+0.145$ at 1000) are not interpreted as evidence for or against the reuse relation.

D.7 Residual headroom after fixed portfolios

Figure 1 compares four ways of choosing which k configurations to evaluate. All choices are made within each fold using training tasks only and evaluated on that fold's held-out tasks, so no portfolio is selected using the rows on which it is scored.

Best-k portfolio is constructed greedily to minimize the portfolio oracle loss, $\mathbb{E}_p[\min_{a \in S} L(p, a)]$, on the training rows: at each step, we add the configuration that produces the largest reduction in this objective. This differs from ranking configurations individually because it favors complementary configurations that are best on different pairs. *Top-k mean* instead selects the k configurations with the lowest individual mean training loss. The two coincide at $k = 1$ by construction and may diverge thereafter.

D-optimal ignores predictive performance and greedily maximizes $\log\det(Z^\top Z + \lambda I)$, where the rows of Z are intervention codes $z_a \in \{0, 1\}^5$ indicating whether the encoder is shared, initialized from auxiliary pretraining, allowed to adapt, given a per-task adapter, and whether outputs are coupled. We use $\lambda = 10^{-6}$ to regularize singular designs. At $k = 1$, this criterion selects the configuration with the most active operations, adp_c, in every domain. It serves as a control for whether spanning the intervention space is itself a useful portfolio criterion.

Random samples k configurations uniformly without replacement and averages results over three draws. It is an illustrative no-information baseline rather than a precisely estimated random-portfolio expectation, so we do not attach a resampling interval.

These portfolios diagnose how much pair-specific oracle headroom remains after evaluating a small fixed set of configurations; they are not compute-matched alternatives to a router, since evaluating a portfolio of size k still requires evaluating k configurations per pair. Axes in Figure 1 use a symmetric-log scale so that exact zeros are retained; QM9 reaches $H_k = 0$ at $k = 6$ under the best-k portfolio. Resampling intervals are shown only for that portfolio to avoid four overlapping bands.

E Preferred action → out-of-sample value and deployment

§5 asks whether pair-specific preferences reproduce on independent experimental replicates, while §6 asks whether a rule learned from other pairs can generalize to new pairs, tasks, or dataset families. We examine these questions in increasing order of distribution shift: first cross-replicate selection, then routing under pair-, task-, and family-level withholding.

E.1 Cross-replicate selection value

We first separate stability of the preferred action from its value on independent replicates. Replicate cells are divided into balanced halves: one half is used to estimate ϵ and select the noise-aware preferred configuration, and that choice is evaluated on the other. Table 25 reports the results. "Same verdict" records pair-level agreement on whether any transfer configuration is selected, while the in-sample column repeats the value calculation without cross-replicate separation. We aggregate

Table 25: Cross-replicate stability and selection value under $\mathcal{A}_{12}$. Preferred configurations and ϵ are estimated on one half of the replicate cells and evaluated on the other. Results aggregate all balanced half-splits.

Domain	Same verdict per pair	Cohen's κ median	Cross-replicate value median [min, max]	Positive splits	In-sample value
CelebA-35	87%	+0.67	+0.0158 [+0.0142, +0.0182]	10/10	+0.0181
QM9-10	89%	+0.67	+0.0265 [+0.0181, +0.0306]	20/20	+0.0319
Regression	74%	+0.34	−0.0021 [−0.0230, +0.0158]	5/10	+0.0376

Table 26: Routing gain over the training-fold best fixed configuration for four routing rules. The baseline and uncertainty resampling match the withheld unit of each evaluation regime. * denotes an interval excluding zero; $\mathcal{A}_6^{\text{ind}}$ requires no auxiliary outcome at evaluation.

Domain	Regime	Set	Ridge	Ridge+abs	Boost+abs	Lead+corr
CelebA	pair interpolation	$\mathcal{A}_8$	$+0.0102^*$	$+0.0103^*$	$+0.0104^*$	$+0.0094^*$
		$\mathcal{A}_{12}$	$+0.0131^*$	$+0.0140^*$	$+0.0139^*$	$+0.0132^*$
		$\mathcal{A}_6^{\text{ind}}$	$+0.0146^*$	$+0.0146^*$	$+0.0164^*$	$+0.0144^*$
	task extrapolation	$\mathcal{A}_8$	+0.0026	+0.0034	+0.0046	+0.0033
		$\mathcal{A}_{12}$	$+0.0095^*$	$+0.0095^*$	$+0.0111^*$	+0.0085
		$\mathcal{A}_6^{\text{ind}}$	$+0.0090^*$	$+0.0099^*$	$+0.0112^*$	$+0.0099^*$
Regression families	pair interpolation	$\mathcal{A}_8$	$+0.0265^*$	$+0.0273^*$	$+0.0308^*$	$+0.0238^*$
		$\mathcal{A}_{12}$	$+0.0132^*$	$+0.0135^*$	$+0.0173^*$	$+0.0123^*$
		$\mathcal{A}_6^{\text{ind}}$	+0.0040	+0.0059	+0.0078	+0.0029
	task extrapolation	$\mathcal{A}_8$	-0.0141^*	-0.0103^*	+0.0016	-0.0078^*
		$\mathcal{A}_{12}$	-0.0158^*	-0.0070^*	−0.0034	-0.0049^*
		$\mathcal{A}_6^{\text{ind}}$	-0.0133^*	−0.0023	−0.0006	-0.0022^*
	family extrapolation	$\mathcal{A}_8$	−0.0067	−0.0032	+0.0007	−0.0048
		$\mathcal{A}_{12}$	-0.0179^*	−0.0031	−0.0004	−0.0034
		$\mathcal{A}_6^{\text{ind}}$	-0.0142^*	−0.0019	+0.0022	-0.0039^*

all balanced half-splits (10 each for CelebA and the regression families and 20 sampled splits for QM9).

CelebA and QM9 show both high preference stability and consistently positive cross-replicate selection value. The regression families differ: the transfer/non-transfer verdict remains moderately stable (74% agreement, $\kappa = 0.34$), but cross-replicate value is positive in only $5/10$ splits, with median -0.0021, despite an in-sample value of $+0.0376$. Thus, reproducibility of the preferred action does not by itself imply positive out-of-sample value from pair-specific selection.

E.2 Routing across deployment regimes

Cross-replicate evaluation asks whether a realized pair-specific choice reproduces. We next examine whether a routing rule learned from other pairs can predict useful configurations under increasingly stronger distribution shifts. Appendix A specifies the routing features, abstention threshold τ, fixed baseline construction, and evaluation folds.

We evaluate four routing rules under the original $\mathcal{A}_8$, the expanded $\mathcal{A}_{12}$, and the independent-output $\mathcal{A}_6^{\text{ind}}$ candidate sets (Table 26). For every evaluation regime, routing gain is measured relative to the best fixed configuration estimated on the corresponding training fold, so router and baseline withhold the same unit. Pair interpolation withholds a pair, task extrapolation withholds an entire task, and family extrapolation withholds an entire regression dataset family. Family extrapolation is unavailable for CelebA because it contains only one family.

Here "abs" denotes abstention. The fourth router uses a two-stage affinity-plus-correction rule: a white-box affinity score (gradient affinity where available and D_{logdet} otherwise) is mapped with gradient boosting, followed by ridge correction using the remaining diagnostics. Its two-stage structure follows Ayman et al. (2026), although the features, action space, and evaluation problem differ from ours; we use it as a robustness check on the routing mapping.

Table 27: Routing gain and cluster-level intervals for the default ridge-with-abstention router. Intervals resample the withheld unit of each deployment regime; * denotes an interval excluding zero.

Regime	Set	CelebA	Regression families
pair interpolation	$\mathcal{A}_8$	$+0.0103^*$ $[+0.0085, +0.0122]$	$+0.0273^*$ $[+0.0180, +0.0365]$
	$\mathcal{A}_{12}$	$+0.0140^*$ $[+0.0121, +0.0160]$	$+0.0135^*$ $[+0.0069, +0.0207]$
	$\mathcal{A}_6^{\mathrm{ind}}$	$+0.0146^*$ $[+0.0132, +0.0161]$	$+0.0059$ $[-0.0002, +0.0120]$
task extrapolation	$\mathcal{A}_8$	$+0.0034$ $[-0.0004, +0.0078]$	-0.0103^* $[-0.0165, -0.0038]$
	$\mathcal{A}_{12}$	$+0.0095^*$ $[+0.0003, +0.0195]$	-0.0070^* $[-0.0131, -0.0017]$
	$\mathcal{A}_6^{\mathrm{ind}}$	$+0.0099^*$ $[+0.0050, +0.0145]$	-0.0023 $[-0.0060, +0.0008]$
family extrapolation	$\mathcal{A}_8$	not defined	-0.0032 $[-0.0079, +0.0012]$
	$\mathcal{A}_{12}$	not defined	-0.0031 $[-0.0090, +0.0020]$
	$\mathcal{A}_6^{\mathrm{ind}}$	not defined	-0.0019 $[-0.0064, +0.0014]$

The qualitative pattern is stable across routing rules. Pair interpolation resolves positively for CelebA under all three candidate sets and for the regression families under $\mathcal{A}_8$ and $\mathcal{A}_{12}$. Stronger extrapolation separates the domains. On CelebA, no router resolves positively under $\mathcal{A}_8$ task extrapolation, three of four do under $\mathcal{A}_{12}$, and all four do under $\mathcal{A}_6^{\mathrm{ind}}$. On the regression families, no router establishes positive task- or family-extrapolation gain. Restricting to independent outputs also makes regression pair interpolation unresolved.

Expanding the candidate set changes the decision problem as well as the available actions. On CelebA, headroom increases from 0.0176 under $\mathcal{A}_8$ to 0.0206 under $\mathcal{A}_{12}$, and the training-fold best fixed configuration changes. The stronger $\mathcal{A}_{12}$ task-extrapolation result should therefore not be interpreted as a generic benefit of adding candidates.

For the default ridge-with-abstention router, Table 27 reports the full cluster-level intervals. Resampling matches the withheld unit of each regime. For CelebA, the clearest task-extrapolation result occurs in $\mathcal{A}_6^{\mathrm{ind}}$, with gain $+0.0099$ $[+0.0050, +0.0145]$. Because every configuration in this action space predicts from x alone, the result does not depend on observing an auxiliary outcome at evaluation. The $\mathcal{A}_{12}$ result is also positive, but its lower interval bound is only $+0.0003$. On the regression families, pair interpolation can be positive, but no candidate set establishes positive task- or family-extrapolation gain.

E.3 What drives task-extrapolation routing on CelebA

Under $\mathcal{A}_{12}$, adapted-transfer configurations are selected on 19.8% of held-out CelebA pairs and account for most of the observed routing gain. g_{probe} is associated with this choice, with $r = +0.354$ $[+0.202, +0.478]$ under task-level resampling. Within the full candidate set, gain is concentrated in ptft_c, whose selected pairs have mean gain $+0.0420$.

The result persists when conditional output coupling is removed. Under $\mathcal{A}_6^{\mathrm{ind}}$, task-extrapolation gain ranges from $+0.0090$ to $+0.0112$ across the four routers, with every interval excluding zero. Here ptft becomes the best fixed configuration, and routing gains arise by avoiding adapted transfer on pairs for which another representation strategy performs better. Thus, the CelebA task-extrapolation result reflects pair-specific variation in when adapted transfer is preferable to the available alternatives rather than access to auxiliary outcomes at evaluation.

E.4 Routing utility, cost, and metric dependence

A statistically resolved routing gain need not be large or economical to obtain. For CelebA under $\mathcal{A}_6^{\mathrm{ind}}$, the default task-extrapolation gain of $+0.0099$ is 2.1% of the loss of the best fixed configuration (ptft, mean NLL 0.4601), $1.5\times$ the replicate-scale reference ϵ, and 43% of oracle headroom. The gain is therefore resolved but modest in absolute scale.

The computational interpretation depends on deployment scale. Exhaustively evaluating the six independent-output configurations for one new target against all 34 auxiliaries requires 170 model trainings: five fits per auxiliary because the separate-encoder configuration reuses the target's cached

single-task networks. Routing requires only the two target-side single-task fits needed for its diagnostics once auxiliary-pretrained encoders are available, giving roughly an $85\times$ reduction in marginal training.

This comparison excludes construction of the router itself. In our experimental harness, meta-training uses candidate outcomes from the 1122 training pairs remaining after the new target's 68 pairs are withheld and requires 5715 trainings with the available caches. Against a marginal saving of 168 trainings per new target, this cost is recovered after roughly 35 new targets. Routing is therefore unattractive as a one-off selector but can become useful when its construction cost is amortized over a stream of targets. These counts are specific to our experimental harness and are intended as order-of-magnitude comparisons rather than throughput benchmarks.

The deployment conclusion is also metric-dependent. Re-scoring CelebA using the task-native loss $1 - \mathrm{AUC}$ gives task-extrapolation gain $+0.0014\,[-0.0001, +0.0034]$, compared with $+0.0099\,[+0.0050, +0.0145]$ under the log score. The sign agrees, but the task-native interval includes zero; the gain is only 0.41% of the loss and $0.13\times$ the replicate-scale reference, versus 2.14% and $1.5\times$ under the log score. We therefore treat the routing result as specific to the log score.

This differs from the adaptation intervention itself, whose NLL- and AUC-based contrasts correlate strongly ($\rho = +0.87$; Appendix C.2). The best fixed configuration also changes from ptft under the log score to joint under $1 - \mathrm{AUC}$. Thus, robustness of an intervention effect across metrics need not propagate to the preferred configuration or deployment value.

E.5 Intervention factorization

A natural alternative is to predict each controlled intervention effect separately and reconstruct a configuration's loss by summing predicted effects along a path through the configuration graph, rather than predicting each configuration's total contrast relative to the default.

For the ridge router used here, however, the two parameterizations are algebraically equivalent when they use the same design matrix and regularization parameter. Ridge prediction is linear in the response, so $S_\lambda(y_1 + y_2) = S_\lambda y_1 + S_\lambda y_2$. Because edge effects telescope to the corresponding total loss contrast, summing their ridge predictions is identical to fitting that contrast directly. We confirm this equivalence to machine precision over $\mathcal{A}_6^{\mathrm{ind}}$.

Within $\mathcal{A}_{12}$, the parameterizations can differ because the factorized model shares the coupling-effect predictor across representation states, introducing parameter tying across edges. Relative to the per-configuration router, this changes gain by $+0.0019\,[+0.0005, +0.0037]$ on the regression families and $-0.0011\,[-0.0028, +0.0003]$ on CelebA, without changing the qualitative conclusions. Thus, decomposing configuration loss into intervention effects does not by itself improve routing; it changes the selector only when the factorized parameterization also imposes additional structural sharing.

F Uncertainty and resampling

§3 distinguishes uncertainty arising from task sampling, data partitioning, and weight initialization. Here we decompose the latter two using the crossed 20×4 partition-by-initialization experiment with the candidate set fixed at $\mathcal{A}_8$, then test whether their ordering is robust to task-pool composition. Robustness of the diagnostic–intervention associations to partition variation is examined separately in Appendix B.3.

F.1 Partition versus initialization variation

Figure 3 reports estimated standard-deviation components in each quantity's native units rather than emphasizing their ratio. Even with 80 crossed cells, the ordering of partition and initialization variation can often be resolved while the exact ratio remains imprecise; under the smaller 3×2 replication used for the full-pair experiments, the ratios are too unstable to interpret.

Partition variation exceeds initialization variation for headroom H and Δ_{adapt} in all three domains, and for Δ_{couple} on CelebA and the regression families. The coupling ordering is unresolved on

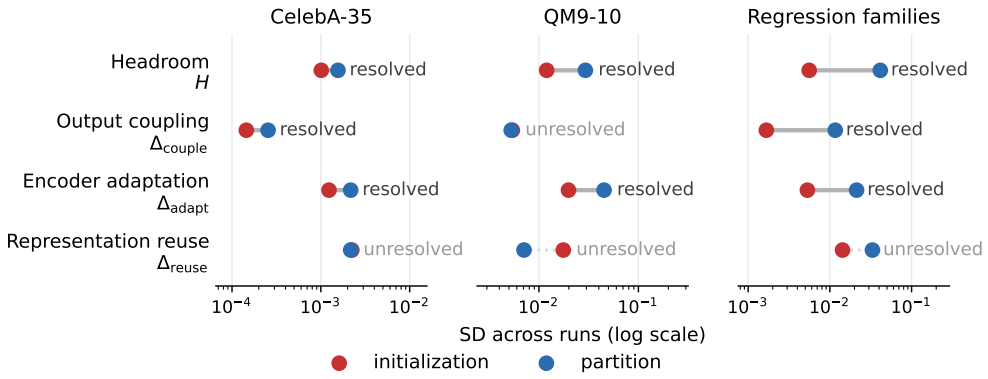

Figure 3: Partition and initialization components of run-to-run variation from 20 data partitions $\times$ 4 weight initializations per domain. Points show estimated standard deviations in each quantity's native units on a log scale. Solid connectors indicate that the partition-to-initialization ratio has an interval excluding one; dotted connectors indicate an unresolved ordering. Exact ratios and task-composition sensitivity are reported in Table 28.

Table 28: Sensitivity of the partition-versus-initialization ordering to task composition. "Full ratio" is the full-data ratio of partition to initialization standard deviations; the leave-one-out range removes one task at a time, or one dataset family for regression. "80% agreement" is the fraction of 40 random 80% task subsets whose ratio remains on the same side of one. "Ordering" is resolved only when the full-data interval in Figure 3 excludes one.

Domain	Quantity	Full ratio	Leave-one-out range	80% agreement	Ordering
CelebA-35	H	1.54	1.38–1.74	100%	resolved
	Δ_{couple}	1.76	1.47–2.40	98%	resolved
	Δ_{adapt}	1.76	1.62–1.98	100%	resolved
	Δ_{reuse}	0.97	0.91–1.05	32%	unresolved
QM9-10	H	2.45	1.60–3.54	100%	resolved
	Δ_{couple}	0.97	0.89–5.80	45%	unresolved
	Δ_{adapt}	2.28	1.68–2.92	100%	resolved
	Δ_{reuse}	0.40	0.33–0.50	100%	unresolved
Regression	H	7.41	5.22–10.06	100%	resolved
	Δ_{couple}	6.98	5.14–12.77	100%	resolved
	Δ_{adapt}	4.00	1.53–5.36	100%	resolved
	Δ_{reuse}	2.32	1.65–8.85	100%	unresolved

QM9, while Δ_{reuse} is unresolved in all three domains. Thus, initialization seeds alone can understate run-level uncertainty for several quantities.

F.2 Sensitivity to task composition

The variance decomposition is itself estimated from a finite task pool. We test whether the partition-versus-initialization ordering changes when that pool is perturbed. We refit the decomposition after removing one task at a time—or one dataset family for regression—and over 40 random subsets retaining 80% of the available tasks (Table 28). Here Δ_{adapt} is the compound contrast, which changes initialization as well as adaptation.

All orderings classified as resolved in the full data are preserved in every leave-one-out refit and in 98–100% of the random subsets. The two composition-sensitive cases are CelebA Δ_{reuse} and QM9 Δ_{couple}, both with full-data ratios of 0.97: their random-subset agreement falls to 32% and 45%, respectively. Thus, composition sensitivity is concentrated where partition and initialization variation are nearly tied. By contrast, QM9 and regression Δ_{reuse} remain classified as unresolved

Table 29: Three-task composition on CelebA. The observed-outcome pairwise rule shares with each auxiliary whose corresponding pairwise shared model outperforms target-only training. We compare this rule with the oracle over six group configurations and with the best fixed configuration. The group oracle most often selects an encoder pretrained on both auxiliaries, an action with no pairwise counterpart.

Quantity	Value
Triplets (distinct targets)	100 (32)
Selection headroom H	0.0183 ($2.2 \times \epsilon$)
Best fixed configuration	pretrained on both, frozen
Best-fixed oracle agreement	38%
Pairwise-rule oracle agreement	40%
Pairwise-rule gain over best fixed	$+0.0078\,[-0.0016, +0.0185]$
Both auxiliaries individually beneficial	28%
Among these, joint sharing beats best single auxiliary	65%

despite stable ratio point estimates; this reflects uncertainty in the full-data intervals rather than instability under task resampling.

QM9's coupling result is particularly sensitive to task composition. Its ten properties are not equally distinct: across the 45 unordered pairs, the median absolute label correlation is 0.309, with u_0^{atom} and ZPVE reaching 0.967. Removing u_0^{atom} raises the coupling variance ratio to 5.80 and removing ZPVE raises it to 2.17, whereas removing any other task leaves it between 0.89 and 1.07. This explains why the QM9 coupling ordering is unresolved and unusually dependent on the particular task pool.

G Beyond pairs: three-task composition

Our analysis maps ordered target–auxiliary pairs to configurations, whereas practical multi-task settings may involve multiple auxiliaries. Therefore, we test pairwise composition in a minimal three-task setting with one data-poor target and two data-rich auxiliaries. The candidate set contains six configurations: target only; sharing with either auxiliary; sharing with both; and an encoder pretrained on both auxiliaries, either frozen or adapted. We evaluate 100 triplets over three data partitions and two initializations using the same 35 CelebA attributes as in the main experiments. Output coupling is omitted to isolate the representation axis. The two single-auxiliary configurations exactly match their corresponding pairwise shared configurations on the same split.

We consider an *observed-outcome pairwise composition rule*: share with each auxiliary whose corresponding pairwise shared model outperforms target-only training. This is deliberately stronger than a diagnostic-based rule because it uses realized pairwise outcomes. Nevertheless, it matches the group oracle among the six configurations on only 40% of triplets, compared with 38% for the best fixed configuration; its gain over that baseline is $+0.0078$, with a target-level bootstrap interval including zero (Table 29).

The mismatch is not simply due to negligible group-level opportunity: $H = 0.0183$, or $2.2\times$ the replicate-scale reference ϵ. A structural limitation is that the most frequent group-oracle configuration, selected on 38% of triplets, uses an encoder pretrained on *both* auxiliaries and therefore has no pairwise counterpart. Pairwise benefits also do not compose deterministically: among the 28% of triplets for which both auxiliaries are individually beneficial, sharing with both outperforms the better single-auxiliary configuration on 65%. Similarly, $\max g_{\mathrm{probe}}$ is nearly unrelated to the incremental benefit of adding the second auxiliary (Spearman $\rho = +0.050$).

This is a boundary experiment rather than a test of whether the twelve-configuration router in §6 generalizes to arbitrary task groups. It covers one domain, 100 triplets, the minimal non-pairwise case $T = 3$, and a simplified action space in which no diagnostic directly selects the group configuration. Within this scope, even observed pairwise outcomes do not determine the group-level choice, providing an empirical limitation of pairwise analysis without implying that pairwise affinity generally fails for multi-task grouping.

H SCOPE AND LIMITATIONS

Several limitations are examined empirically elsewhere: pairwise composition in Appendix G, QM9's limited task-level precision in Appendix B.2, and sensitivity of the variance decomposition to task composition in Appendix F. Here we summarize the remaining scope limitations.

Candidate and diagnostic coverage. Our conclusions are conditional on the configurations and diagnostics studied. The original eight-configuration design includes frozen auxiliary-pretrained transfer but not adaptation from the same transferred initialization; adding that option changes the preferred-action result in §5. Other optimization procedures or parameter-efficient adaptation methods could likewise change the available headroom. Diagnostic coverage also varies by domain. LEEP and H-score are classification-specific and are evaluated only on CelebA, so cross-domain comparisons involve six of the eight diagnostics. We interpret exploratory diagnostic comparisons cautiously and base the primary diagnostic–intervention conclusions on the pre-specified tests.

Conditional output coupling. The coupling intervention is a conditional multi-output predictor: at evaluation time it uses the observed auxiliary outcome to condition the target predictive distribution. The evaluated coupling configuration is therefore unavailable when auxiliary outcomes are not observed at evaluation; this restriction does not apply to the representation-side interventions. For routing, the CelebA task-extrapolation gain persists when the action space is restricted to the six independent-output configurations. Thus, the positive CelebA task-extrapolation result is not driven solely by access to auxiliary outcomes at evaluation.

Deployment scope and untested distribution shifts. Task extrapolation is evaluated on CelebA and the regression collections, whereas family extrapolation is available only for the regression domain because CelebA contains a single family. The resolved CelebA task-extrapolation result therefore does not establish generalization to unseen dataset families. It is also metric-dependent: the gain resolves under the log score but not under the task-native $1-\mathrm{AUC}$ metric (Appendix E.4). More broadly, all routing experiments use the present diagnostic set and task distributions. Performance under stronger distribution shifts remains untested.

Remaining uncertainty, including hardware variation. Task sampling, data partitioning, weight initialization, and hardware are distinct sources of uncertainty. The crossed 20×4 experiment isolates partition and initialization variation, while task- or family-level resampling accounts for finite task pools in the corresponding population-level analyses; neither captures all experimental variability. In one repeated bake-off, otherwise identical code, seeds, and data produced pair-level likelihood differences of up to 0.064 nats across devices. Hardware variation is not included as a formal variance component. Very small absolute likelihood differences should not be interpreted as robust across hardware.