Scaling Near-Optimal SFT–RL Annotation Budget Allocation from Small to Large LLMs

Jingtan Wang[1,3], Arun Verma[2,*], Xiaoqiang Lin[1], Zhengyuan Liu[3], Nancy F. Chen[3], Daniela Rus[4,2], Bryan Kian Hsiang Low[1,2]

[1]Department of Computer Science, National University of Singapore, Republic of Singapore

[2]Singapore-MIT Alliance for Research and Technology Centre, Republic of Singapore

[3]Agency for Science, Technology, and Research (A*STAR), Republic of Singapore

[4]CSAIL, Massachusetts Institute of Technology, USA

Abstract

How to divide a fixed annotation budget between supervised fine-tuning (SFT) and reinforcement learning (RL) during LLM post-training remains an open problem. Existing work characterizes only broad trends (e.g., SFT dominates in low-data regimes), lacks a principled allocation framework, and does not examine whether the optimal ratio transfers across model sizes. We frame this problem in terms of near-optimality: rather than seeking a single optimal SFT–RL ratio, we characterize the near-optimal region, the set of allocations within a specified tolerance of peak performance. Empirically, this region is wide even for small tolerances (2–10%), widens with model scale, and transfers reliably from small proxy models to large target models. This yields a practical strategy: small proxy-model experiments suffice to identify a transferable near-optimal region, eliminating the need for exhaustive large-scale search. Our results hold consistently across tasks, model families, and both preference-based off-policy and reward-supervision on-policy RL methods. We further analyze how the asymmetry in annotation costs between SFT and RL data shifts the near-optimal region.

1 Introduction

Post-training has become an essential stage for equipping pretrained models with targeted downstream capabilities (Ivison et al., 2023; Lambert et al., 2025). Modern LLM post-training pipelines are multi-stage, typically consisting of supervised fine-tuning (SFT) followed by reinforcement learning (RL) or preference-based optimization (Guo et al., 2025). This structure raises a natural question: *given a fixed annotation budget for training samples, how should it be divided between SFT and RL?* Allocating too few samples to SFT leaves a weak initial policy that RL cannot effectively improve; over-investing in SFT reduces the budget available for preference and reward supervision data necessary for policy improvement. The right balance depends on model scale, task, and post-training method, yet determining it without exhaustive experimentation remains an open problem.

The most relevant work, Raghavendra et al. (2025), characterizes only a broad trend (SFT dominates in the low-data regime while preference optimization becomes more favorable at scale), and neither identifies the optimal allocation without grid search nor establishes whether it transfers across model scales. Other related work on data mixture optimization addresses only single-stage settings: domain mixtures during pretraining (Xie et al., 2023; Fan et al., 2024) or via scaling-law extrapolation (Liu et al., 2025; Ye et al., 2025); instruction-mixture composition during fine-tuning (Ivison et al., 2023; Lambert et al., 2025; Shi et al., 2026; Corrado et al., 2025); and the pretraining–fine-tuning forgetting trade-off (Béthune et al., 2025). None of these addresses multi-stage SFT–RL budget allocation or how to optimize it at scale.

SFT–RL budget allocation problem presents two fundamental challenges. ① Post-training is sensitive to fine-tuning method and task (Zhang et al., 2024), metric construction (Schaeffer et al., 2025), reward overoptimization (Gao et al., 2023), and inverse scaling (McKenzie et al., 2023); moreover, post-training algorithm rankings can flip across scale (Li, 2026). Although certain aggregate behaviors remain predictable (Gadre et al., 2025; Ruan et al., 2024), the optimal *SFT–RL ratio may not transfer reliably across model scales* (Sec. 3.2). ② Any adjustment to the SFT–RL allocation typically requires retraining from scratch or rerunning subsequent stages, making *exhaustive grid search at full model scale prohibitively expensive*.

To address these challenges, we frame the problem in terms of *near-optimality* rather than exact optimality. Instead of seeking a single optimal ratio, we introduce the *near-optimal region*: the

*Corresponding author: arun.verma@smart.mit.edu.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 55601–55630
October 24-29, 2026 ©2026 Association for Computational Linguistics

set of allocations whose performance lies within a fixed tolerance of the best observed performance (Sec. 3.2). Our experimental results show that this region is wide even for small tolerances (2–10% of peak performance), grows consistently wider with model scale, and transfers reliably from small proxy models to large target models. These results yield a practical strategy: small proxy-model experiments suffice to identify a transferable near-optimal region, eliminating the need for exhaustive large-scale search (Sec. 3.3).

We summarize our key contributions as follows:

- **Near-optimal region analysis.** We show that even a small tolerance (e.g., a few percent of the optimum) yields a wide near-optimal region rather than a single sharp optimum across tasks and model families (Sec. 3.2).
- **Scale-dependent region expansion.** At a fixed tolerance, the near-optimal region generally widens with model scale; regions identified on small proxy models transfer reliably to larger targets, enabling efficient region discovery without exhaustive large-scale search (Sec. 3.3).
- **Generality across settings.** The near-optimal region widens consistently across tasks, model families, and both off-policy and on-policy RL methods. We further show how annotation cost asymmetry between SFT and RL data shifts the optimal allocation (Sec. 3.4): the region widens as SFT becomes relatively more expensive.

2 Preliminaries

2.1 Problem Formulation

We study the allocation of a fixed post-training budget between SFT and subsequent RL-based training. Given a pretrained model $\mathcal{M}_N$ of size N and a total annotation budget B, an allocation ratio $r \in [0, 1]$ assigns rB to SFT and $(1-r)B$ to the RL stage. The two stages are applied sequentially (SFT followed by RL), yielding a post-trained model that is evaluated using a non-negative, task-specific performance metric $\mathcal{P}(N, r, B; \tau)$. We suppress the task τ when clear from context. Following Raghavendra et al. (2025), we measure B in terms of annotated training samples, assuming comparable annotation costs across SFT and RL; we generalize to heterogeneous annotation costs in Sec. 3.4. We assume training prompts are available for both stages and omit infrastructure and engineering costs. We simplify the budget and only consider annotated samples, following standard convention (Raghavendra et al., 2025; Shen et al., 2025) and, crucially, annotation dominates the total cost in post-training. Using our measured runtimes and current cloud GPU prices in L40s[1], per-example training compute ranges from $\$2 \times 10^{-5}$ (1B SFT, the cheapest) to $\$3 \times 10^{-4}$ (8B DPO). Against human annotation ($\$0.5$–$1.0$ per example (Kiela et al., 2021; Lee et al., 2024)), this is 3–4 orders of magnitude smaller. Against cheap synthetic annotation (10^{-3}\$/example), the margin is smaller, but compute remains below annotation, so annotation is the dominant term and our first-order budget axis.

From point optimum to near-optimal region. A natural choice is to optimize the allocation ratio by aiming at the point of optimum, defined as:

$$r^*(N, B) = \arg\max_{r \in [0,1]} \mathcal{P}(N, r, B). \quad (1)$$

In practice, post-training performance is sensitive to the task, model family, and training method (Caballero et al., 2023; McKenzie et al., 2023; Zhang et al., 2024; Schaeffer et al., 2025), making r^* ill-conditioned and a poor target. As we show in Sec. 3.2, $\mathcal{P}(N, \cdot, B)$ often forms a wide plateau near its maximum, so the question of practical interest is not the single best ratio but the set of ratios achieving near-optimal performance. We therefore study the ε-near-optimal region, defined as:

$$\mathcal{R}_\varepsilon(N, B) = \Big\{ r \in [0, 1] : \mathcal{P}(N, r, B) \geq (1 - \varepsilon) \max_{r' \in [0,1]} \mathcal{P}(N, r', B) \Big\}, \quad (2)$$

where $\varepsilon \in [0, 1]$ is a relative performance tolerance (e.g., $\varepsilon = 5\%$ allows any ratio retaining at least 95% of peak performance). Eq. (2) reduces to the point-optimum formulation in Eq. (1) as $\varepsilon \to 0$.

Positioning. Prior work observes broad trends in allocation ratio r^* across SFT and preference optimization (Raghavendra et al., 2025). However, it does not characterize *where* near-optimal allocations lie, *how wide* the near-optimal region is, or *whether* allocations transfer across model scales. To fill this gap, we study how $\mathcal{R}_\varepsilon$ varies with model size, establishing that (i) the region widens at larger scales under a fixed tolerance, reflecting greater flexibility in practice; and (ii) near-optimal regions transfer more reliably across scales than exact optima, offering a robust proxy-to-target allocation mechanism.

[1] https://www.coreweave.com

2.2 Supervised Fine-Tuning (SFT)

SFT adapts a pretrained language model on curated (prompt, response) pairs. Given a dataset $\mathcal{D}_{\text{SFT}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{S}}}$ of N_{S} demonstration samples, where y_i denotes the response for prompt x_i, the model is trained via a next-token prediction loss:

$$\mathcal{L}_{\text{SFT}}(\pi_\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{SFT}}}\left[\log \pi_\theta(y \mid x)\right],$$

where π_θ denotes the policy parameterized by θ, and $\pi_\theta(y \mid x)$ is the likelihood of the target response y for a given prompt x.

2.3 RL-based Post-Training

Beyond supervised imitation, RL fine-tuning equips the model with reward-maximizing or preference-aware behavior. We study both the off-policy and on-policy paradigms via Direct Preference Optimization (DPO) and Group Relative Policy Optimization (GRPO), respectively.

Off-policy RL: DPO. DPO (Rafailov et al., 2023) provides a reward-model-free alternative to PPO-based RLHF (Schulman et al., 2017), trained on a static preference dataset $\mathcal{D}_{\text{DPO}} = \{(x_i, y_{i,w}, y_{i,l})\}_{i=1}^{N_{\text{D}}}$ of N_{D} samples, where $y_{i,w}$ and $y_{i,l}$ denote the preferred and rejected responses for prompt x_i. DPO optimizes the policy relative to a fixed reference model π_{ref} using the objective:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{DPO}}}\left[\log \sigma\left(\beta\Delta\right)\right],$$

where $\Delta = \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$, β controls the strength of the KL penalty, and $\sigma(\cdot)$ is the sigmoid function. Since DPO learns from a static preference dataset without sampling new responses during training, it is an off-policy method.

On-policy RL: GRPO. GRPO (Shao et al., 2024) is an on-policy RL method that updates the model using responses sampled from the current policy. For each prompt x, the model generates a group of N_{G} responses $\{y_i\}_{i=1}^{N_{\text{G}}}$, each assigned reward r_i. GRPO normalizes rewards within the group to compute relative advantages: $A_i = \left(r_i - \text{mean}(\{r_j\}_{j=1}^{N_{\text{G}}})\right) / \text{std}(\{r_j\}_{j=1}^{N_{\text{G}}})$. The policy is then optimized via a PPO-style clipped objective:

$$\mathcal{L}_{\text{GRPO}}(\pi_\theta) = -\mathbb{E}\left[\min\left(\rho_i A_i, \xi_i\right)\right],$$

where $\xi_i = \text{clip}(\rho_i, 1-\varepsilon_{\text{clip}}, 1+\varepsilon_{\text{clip}})A_i$, $\rho_i = \pi_\theta(y_i|x) / \pi_{\theta_{\text{old}}}(y_i|x)$ is the importance sampling ratio, and $\varepsilon_{\text{clip}}$ is the clipping threshold. Unlike DPO, GRPO continuously samples responses from the current policy, making it an on-policy method.

3 Analysis

3.1 Experimental Setup

Models and tasks. We study post-training allocation using the Llama 3 (Grattafiori et al., 2024), Qwen 2.5 (Yang et al., 2024a), and Qwen 3 (Yang et al., 2025) model families across multiple scales up to 14B parameters, evaluating four representative post-training capabilities: math (Cobbe et al., 2021), instruction following (Zhou et al., 2023b), summarization (Stiennon et al., 2020), and helpfulness (Wang et al., 2024b,a). These tasks cover a diverse range of language model behaviors: reasoning, controllable generation, alignment to user intent, and preference-sensitive response quality. An overview is given in Tab. 1; further details are in App. A.1. Scaling results up to 14B parameters are deferred to App. D.

Allocation and budget grids. We instantiate the formulation of Sec. 2.1 on discrete grids of allocation ratios and budget values. Following Raghavendra et al. (2025), we evaluate over the allocation grid $\mathcal{G} = \{0.00, 0.25, 0.50, 0.75, 1.00\}$[2] over budgets $\mathcal{B} \subset [0, 15\text{k}]$ across different model sizes $\mathcal{N}$. For each $(N, r, B) \in \mathcal{N} \times \mathcal{G} \times \mathcal{B}$, we train on $\lceil rB \rceil$ SFT samples followed by $\lfloor (1-r)B \rfloor$ RL-stage samples and evaluate the resulting model on $\mathcal{P}$. Performance curves in the small-budget regime ($B < 5\text{k}$) are noisy and occasionally non-monotonic, reflecting instability in early post-training dynamics before convergence. We therefore restrict our analysis of scaling behavior to $B \geq 5\text{k}$, consistent with the common practice of excluding pre-convergence points from scaling-law analysis (Kaplan et al., 2020; Hoffmann et al., 2022). Full performance curves across all budgets, including the pre-convergence regime, are reported in App. B.

Estimating near-optimal region. Since $\mathcal{P}$ is measured on $\mathcal{G}$, we use the grid-restricted estimate:

$$\widehat{\mathcal{R}}_\varepsilon(N, B) = \Big\{ r \in \mathcal{G} : \mathcal{P}(N, r, B) \geq (1-\varepsilon) \max_{r' \in \mathcal{G}} \mathcal{P}(N, r', B) \Big\}. \quad (3)$$

We report two complementary grid-based estimators of the width of $\widehat{\mathcal{R}}_\varepsilon(N, B)$:

- **Range width:** $w_\varepsilon^{\text{rng}}(N, B) = \max \widehat{\mathcal{R}}_\varepsilon - \min \widehat{\mathcal{R}}_\varepsilon$, the span of near-optimal ratios on $[0, 1]$;

[2] To verify that $\mathcal{G}$ is sufficient, we repeat the Sec. 3.3 analysis on Math and Summarization using a denser 9-point grid formed by inserting midpoints $\{0.125, 0.375, 0.625, 0.875\}$ to $\mathcal{G}$; results are qualitatively consistent (App. E.1).

Task	SFT-stage	RL-stage	Evaluation
Math	GSM8K (Cobbe et al., 2021)	Tülu3 Grade School Math (Lambert et al., 2025)	GSM8K Test Accuracy
Instruction following	Tülu3 Persona IF (Lambert et al., 2025)	Tülu3 Persona IF / Tülu3 RLVR IF (Lambert et al., 2025)	IFEval Accuracy (Zhou et al., 2023b)
Summarization	Reddit TL;DR (Völske et al., 2017)	Reddit Comparison (Stiennon et al., 2020)	ROUGE-L F1
Helpfulness	HelpSteer (Wang et al., 2024b)	HelpSteer2 (Wang et al., 2024a)	Reward Model (Yang et al., 2024b)

Table 1: Overview of task-specific datasets and evaluation settings. For most tasks, DPO and GRPO use the same or closely matched prompt distributions. For instruction following, GRPO uses the Tülu3 RLVR data.

- **Count width:** $w_\varepsilon^{\mathrm{cnt}}(N,B) = |\widehat{\mathcal{R}}_\varepsilon|/|\mathcal{G}|$, the fraction of grid points in $\mathcal{G}$ that are near-optimal.

Both estimators lie in $[0,1]$ and share the same interpretation: the fraction of allocation ratios achieving flexibility under tolerance ε. They differ only when $\widehat{\mathcal{R}}_\varepsilon$ is non-contiguous on the grid: range width captures the convex-hull span (sensitive to boundary ratios), while count width captures cardinality (sensitive to gaps). Empirically, the two yield consistent qualitative trends (Sec. 3.3, App. E.2); per-ratio hit-rate heatmaps (Sec. 3.3) confirm that near-optimal ratios are generally contiguous, indicating $\widehat{\mathcal{R}}_\varepsilon$ is approximately an interval. We report range width $w_\varepsilon^{\mathrm{rng}}$ in Sections 3.2 to 3.4 and count-based analysis in App. E.2. For model-scale analyses in Sections 3.2 to 3.4, statistics are averaged over budgets in the stable regime ($B \geq 5$k).

Training Setup. For RL-based post-training, we primarily analyze off-policy preference optimization (DPO) for training efficiency, and extend to on-policy optimization (GRPO) with verifiable rewards. To ensure fair comparisons across allocation ratios, SFT and RL-stage training data for each task come from a consistent source: either human-annotated or machine-generated. Given the large number of post-training runs required, we use LoRA (Hu et al., 2022) for training efficiency. Detailed hyperparameters are in App. A.2.

3.2 Sensitivity of Budget Allocation

As established in Sec. 2.1, our analysis centers on the ε-near-optimal region $\mathcal{R}_\varepsilon(N,B)$ rather than the point optimum $r^*(N,B)$. Throughout this section and the appendix, $\mathcal{R}_\varepsilon$ refers to the grid-restricted width estimate of $\widehat{\mathcal{R}}_\varepsilon(N,B)$ (Eq. (3)), and we report the range-based width $w_\varepsilon^{\mathrm{rng}}$ as the primary width statistic. We first characterize how wide $\mathcal{R}_\varepsilon$ is across tasks and model families.

Optimal allocation as a near-optimal region rather than a single ratio. As shown in Fig. 1, the near-optimal region width $\mathcal{R}_\varepsilon$ expands consistently with tolerance across tasks and model families. Under a 10% tolerance, retaining at least 90% of optimal performance, most tasks admit near-optimal ratios covering 55%–75% of the feasible allocation space. Post-training performance is therefore broadly insensitive to allocation choice within a wide neighborhood, rather than being concentrated at a single ratio. This empirically supports the plateau structure of $\mathcal{P}(N,\cdot,B)$ posited in Sec. 2.1 and motivates studying $\mathcal{R}_\varepsilon$.

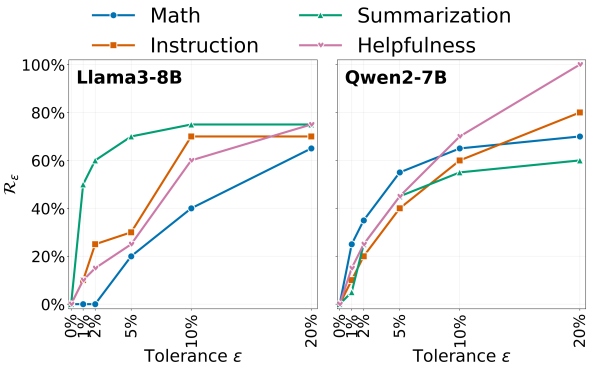


Figure 1: Tolerance ε versus near-optimal-region width $\mathcal{R}_\varepsilon(= w_\varepsilon^{\mathrm{rng}})$ for Llama3-8B and Qwen2.5-7B across four tasks. The near-optimal region expands consistently with tolerance across tasks and model families.

3.3 Scaling Behavior of Budget Allocation

Having established that $\mathcal{R}_\varepsilon$ is substantially wider than the point optimum across all tasks, we now study how it scales with model size N and whether near-optimal allocations transfer from a small proxy model M_{N_s} to a larger target model M_{N_t}. Results are shown in Figs. 2 to 4.

Near-optimal region widens with model scale. The first row of each sub-figure reports the average width $w_\varepsilon^{\mathrm{rng}}(N,B)$ across budgets. Across tasks and model families, the near-optimal region under a fixed tolerance ε generally widens with N: smaller models often admit only a narrow set of near-optimal ratios under strict tolerances, whereas larger models accommodate a broader range. From a practitioner's view, deploying a larger model thus affords greater freedom in choosing r while retaining a target fraction of peak performance.

This widening can be driven by larger models achieving higher absolute peak performance

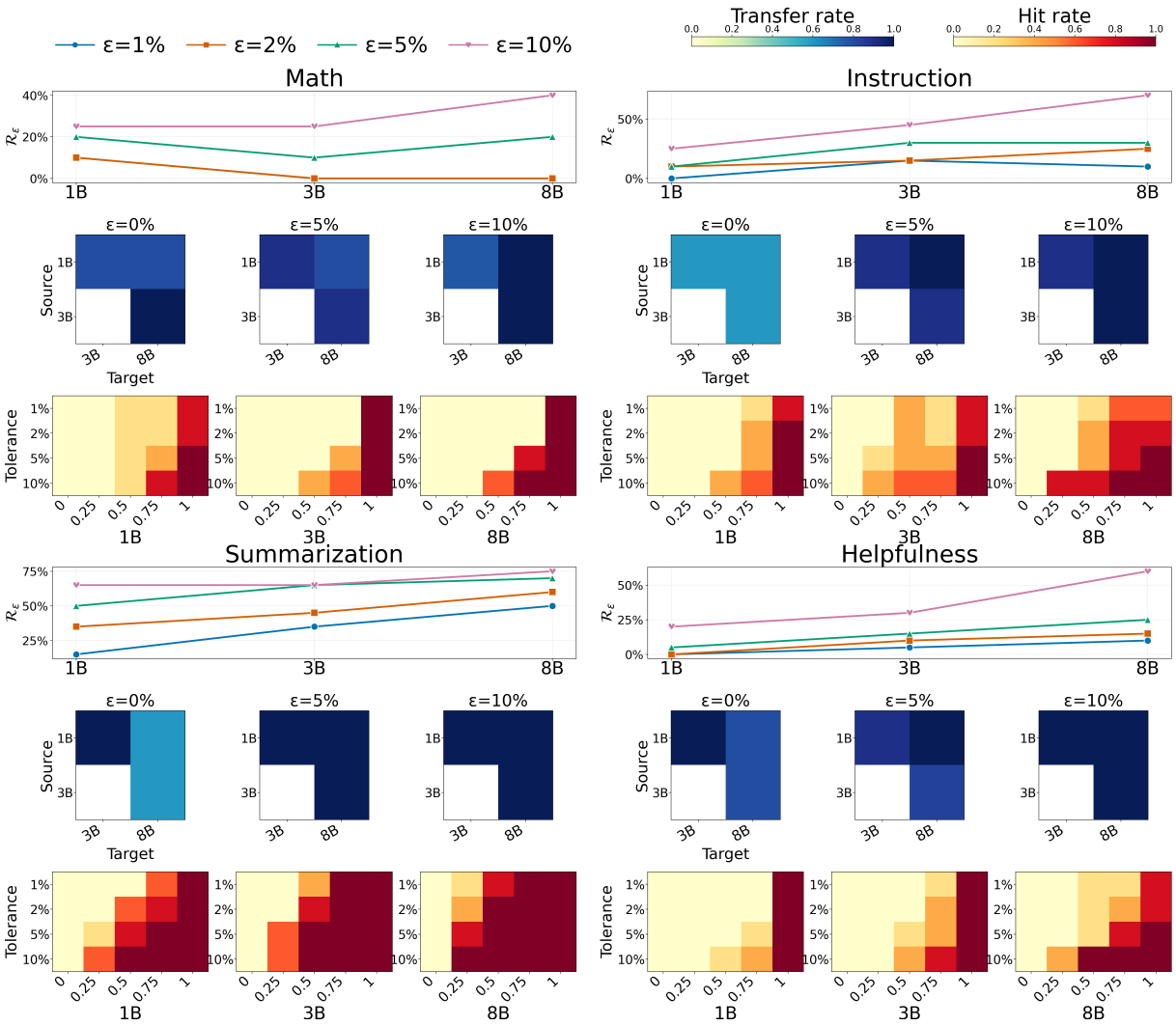


Figure 2: **Main analysis for the Llama family with SFT–DPO training.** Each task panel has three rows. **Row 1 (line plot):** mean near-optimal region width versus model size N per tolerance level ε; width generally increases with model scale. **Row 2 (transfer matrices):** cross-size overlap of near-optimal region grid-restricted estimates (for brevity, we use near-optimal region) $\widehat{\mathcal{R}}_\varepsilon(N, B)$ (small $\rightarrow$ large); these regions transfer more reliably across model scales than point optimum. **Row 3 (heatmaps):** per-ratio hit rate across budgets; each (tolerance, ratio) cell indicates the fraction of budgets B for which that ratio lies within the near-optimal region.

$\mathcal{P}^*(N, B)$: under a fixed tolerance ε, a higher peak implies more absolute slack at the same threshold $(1-\varepsilon)\mathcal{P}^*$, which mechanically admits more ratios into $\mathcal{R}_\varepsilon$. A full decomposition (into peak-performance scaling versus intrinsic changes in allocation sensitivity) is task- and method-dependent, where a detailed analysis in App. E.6 identifies edge cases in which the widening can fail under an alternative definition of tolerance. We therefore interpret the widening as a *fixed-tolerance phenomenon* rather than as evidence that allocation sensitivity itself decreases with scale.

Near-optimal regions transfer more reliably across model scales than point optimum: To quantify cross-scale transferability, we measure the fraction of near-optimal allocations identified on a small proxy model M_{N_s} that remain near-optimal on a larger target model M_{N_t}:

$$T_\varepsilon(N_s \rightarrow N_t;\, B) = \frac{|\widehat{\mathcal{R}}_\varepsilon(N_s, B) \cap \widehat{\mathcal{R}}_\varepsilon(N_t, B)|}{|\widehat{\mathcal{R}}_\varepsilon(N_s, B)|}.$$

The asymmetry is intentional: we care about preserving near-optimal allocations from a cheaper proxy as we scale up. We compute T_ε *per-budget* B, and aggregated across budgets for reporting. The second row of each sub-figure reports the budget-aggregated T_ε for each model-size pair.

Under point optimum ($\varepsilon = 0$), transferability is sometimes high but can be inconsistent across tasks, budgets, and model families, particularly for the Qwen2.5 family (Fig. 3). Allowing a modest

tolerance substantially and consistently improves transfer, indicating that near-optimal regions are a more robust object for cross-scale transfer than the exact point optimum. A wider region on the target model mechanically increases its overlap with the source region, providing a structural explanation for why T_ε improves as ε increases.

The third row of each sub-figure jointly illustrates the expansion and transferability of near-optimal regions. Each heatmap reports, across budgets, how frequently each allocation ratio falls within $\mathcal{R}_\varepsilon$ at a given (N, ε). Darker cells indicate ratios that consistently achieve near-optimal performance. As ε increases, darker regions expand *contiguously*: neighboring allocation ratios generally yield comparable performance, supporting the contiguity assumption underlying our width estimator (Sec. 3.1). Moreover, the high-frequency regions of smaller models typically overlap with, or are contained within, those of larger models, further supporting cross-scale transfer.

We note one apparent exception: for Qwen 2.5 summarization, $\mathcal{R}_\varepsilon$ narrows marginally at larger scales under higher tolerances. This reflects a potential saturation effect (smaller Qwen models for summarization already admit wide, near-optimal regions, leaving little room for further widening) rather than a collapse in allocation behavior. Cross-scale transfer for Qwen 2.5 summarization remains strong (Fig. 3, heatmaps in second and third rows), consistent with this interpretation. Together, these findings indicate that smaller proxy models can reliably guide post-training allocation for larger target models without requiring expensive large-scale allocation sweeps. The near-optimal region $\mathcal{R}_\varepsilon$ is a more robust objective for cross-scale transfer than the point optimum r^*, with practical implications for the cost-efficient post-training pipeline design.

Trends extend to on-policy post-training: The preceding analyses use DPO, an off-policy preference optimization method chosen for training efficiency. We now extend the analysis to on-policy RL with GRPO trained on verifiable rewards or reward models. Fig. 5 reports results for the SFT–GRPO setting on the Llama family across all four tasks. The qualitative pattern matches the SFT–DPO setting: $\mathcal{R}_\varepsilon$ widens with tolerance, and near-optimal regions transfer across model scales more reliably than the point optimum. Compared with SFT–DPO, the SFT–GRPO results exhibit slightly stronger fluctuations, particularly on Helpfulness, consistent with the higher variance of on-policy training, where policy sampling, reward estimation, and iterative updates introduce additional noise. Our conclusion holds across both settings: the near-optimal region $\mathcal{R}_\varepsilon$ is a more reliable object for cross-scale transfer than the point optimum r^*, for both off-policy and on-policy post-training.

3.4 Extension to Cost Asymmetry Setting

Allocation under heterogeneous annotation costs: Previous sections measure post-training budgets by the number of annotated samples, implicitly assuming equal annotation costs for SFT and RL-stage data. In practice, high-quality supervised demonstrations are typically more expensive than preference annotations or verifier-based rewards (Raghavendra et al., 2025); for instance, generating synthetic SFT data via frontier LLMs can cost roughly 1–2$\times$ as much as preference-style annotations under contemporary API pricing (Wang et al., 2023; Lee et al., 2024). We examine if our scaling and transfer trends persist when SFT and RL data incur different annotation costs, focusing on settings where DPO is the RL-stage objective.

Experimental setting: Fix a monetary budget B (in dollars) and let $\rho := c_{\mathrm{SFT}}/c_{\mathrm{DPO}}$ denote the SFT-to-DPO cost ratio. An allocation r assigns rB dollars to SFT and $(1-r)B$ to DPO, yielding $n_{\mathrm{SFT}} = rB/c_{\mathrm{SFT}}$ and $n_{\mathrm{DPO}} = (1-r)B/c_{\mathrm{DPO}}$ samples. We consider $\rho \in \{1, 2, 5, 10\}$, where larger ρ corresponds to progressively more expensive SFT annotations. Following Raghavendra et al. (2025), we set $c_{\mathrm{DPO}} = \$0.001$ and scale c_{SFT} accordingly; $\rho = 1$ recovers the sample-count budget of Sec. 3.1. We focus on $\rho \geq 1$ because SFT annotations (human-written or LLM-distilled responses) are typically more expensive than preference pairs or verifier-based rewards in contemporary post-training pipelines. We exclude $r = 0$ as it consistently underperforms other ratios and requires disproportionately more data when $\rho > 1$. We primarily analyze the Math task; additional results across tasks are deferred to App. F.

Scaling trends persist under heterogeneous annotation costs: Fig. 6 shows that the qualitative scaling behavior remains largely unchanged under heterogeneous annotation costs. Across ρ settings, larger tolerance thresholds continue to produce wider $\mathcal{R}_\varepsilon$ and stronger cross-scale transfer T_ε. Additionally, as ρ increases, $\mathcal{R}_\varepsilon$ generally widens at the same tolerance: a wider range of allocations

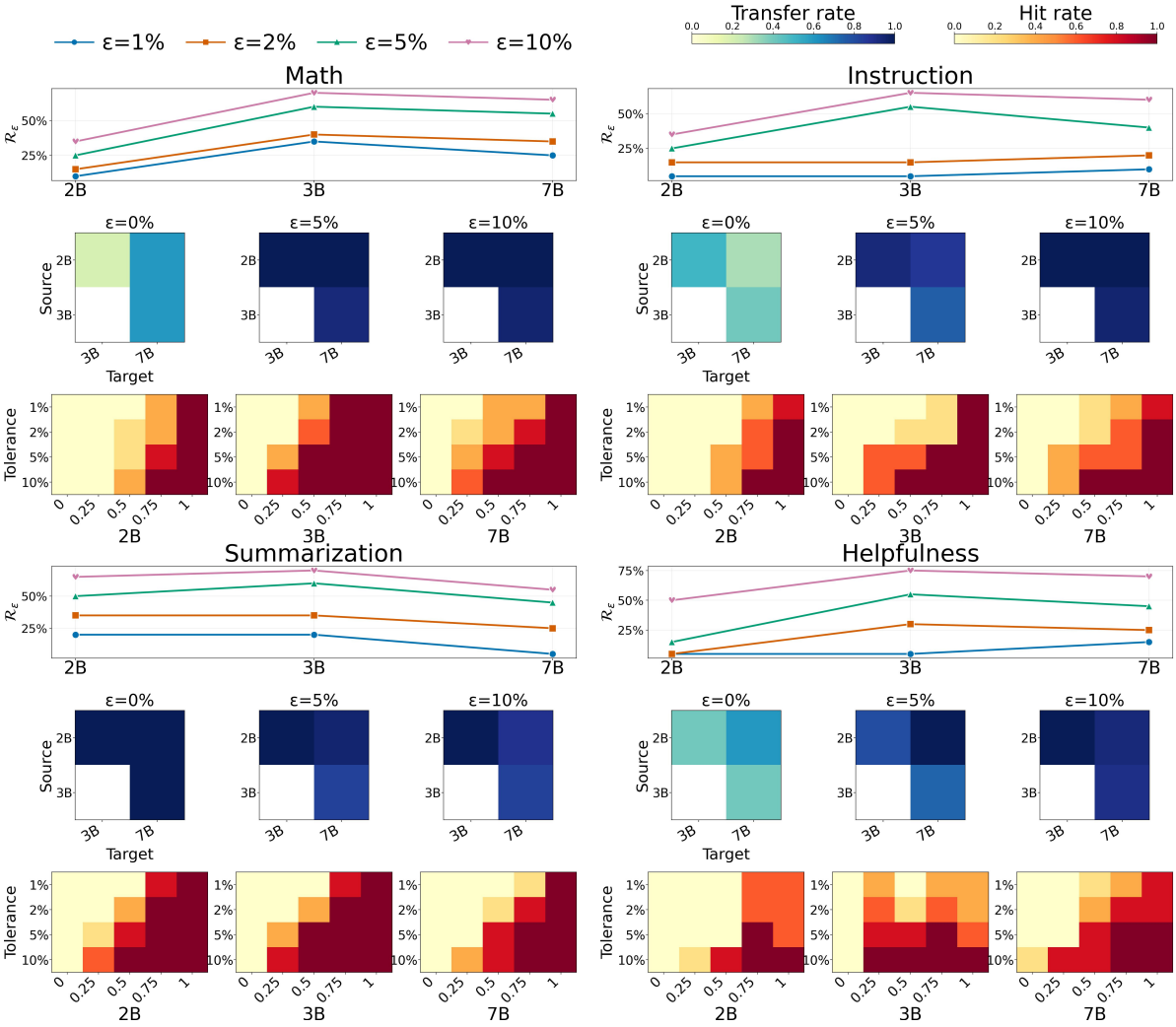


Figure 3: **Main analysis for Qwen 2.5 family with SFT–DPO training stages.** Qualitative trends align with Fig. 2, supporting the same findings. 2B refers to 1.5 Qwen 2.5 1.5B model. For brevity, here and throughout all figures involving the Qwen2.5 family, 2B denotes the Qwen2.5-1.5B model.

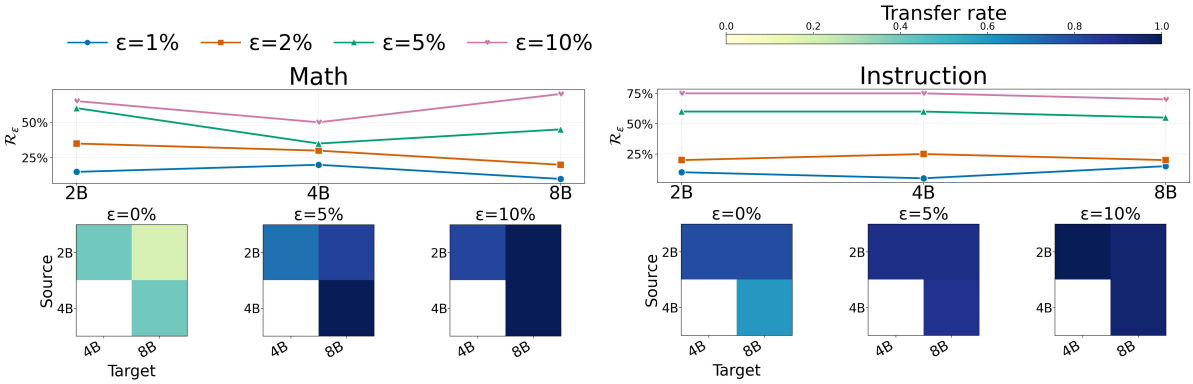


Figure 4: **Main analysis for Qwen 3 family with SFT–DPO training stages.** Qualitative trends align with Fig. 2, supporting the same findings. For a more complete figure with a heat map, see App. C.

achieves comparable performance when SFT annotations are relatively expensive. From a practitioner's perspective, this indicates *increased flexibility* in choosing the SFT–DPO budget split as SFT becomes more costly: the choice of r matters less, and other criteria (annotation logistics, sample reuse) can drive the final allocation. Overall, near-optimal allocation regions remain transferable and stable across a range of plausible cost asymmetries between SFT and DPO annotations, support-

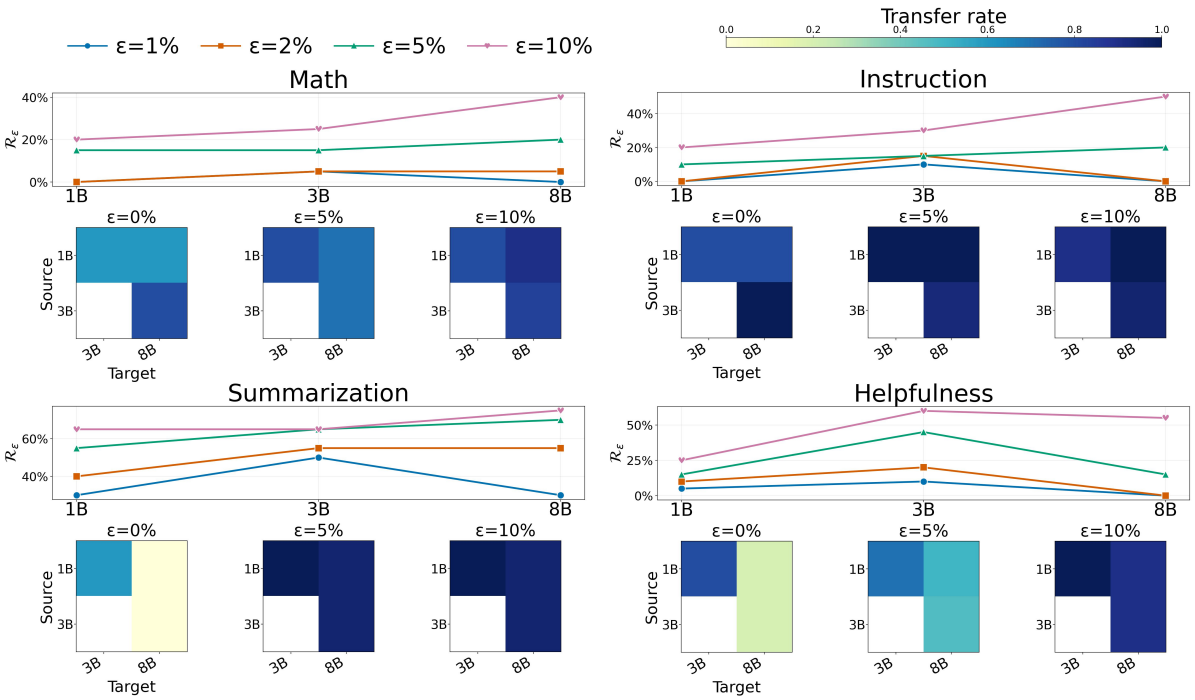


Figure 5: **Analysis for Llama family with SFT–GRPO training stages.** Qualitative trends align with Fig. 2, supporting the same findings. For a more complete figure with a heat map, see App. C.

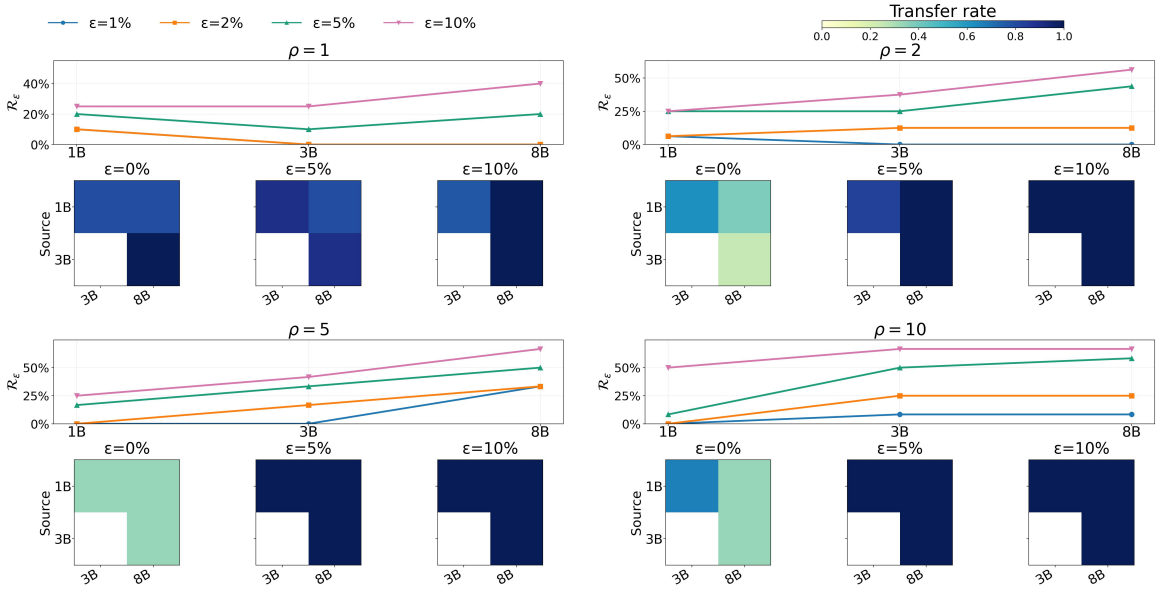


Figure 6: **Effect of heterogeneous annotation costs on transferable allocation regions for the math task.** We simulate varying SFT-to-RL annotation cost ratios across the Llama model family. Qualitative trends remain consistent with the equal-cost setting: larger tolerance thresholds produce broader near-optimal regions and stronger cross-scale transferability. A more complete figure with heatmaps is in App. C.

ing the use of small-proxy allocation guidance in cost-constrained pipelines.

3.5 From Analysis to Practice: A Proxy Allocation Procedure

The preceding results suggest a simple recipe: use a small proxy model to identify a robust allocation region, choose a tolerance that reflects the desired trade-off between performance and allocation flexibility, and then select a ratio from that region. We describe this procedure as follows:

1. **Proxy sweep.** Take the smallest model in the target family as the proxy M_{N_s} and evaluate the 5-point ratio grid $\mathcal{G} =$

$\{0, 0.25, 0.5, 0.75, 1\}$ at the *target* budget B. Five points are sufficient: a denser 9-point grid identifies the same regions (App. E.1).

2. **Choose the tolerance.** Set ε according to how much peak performance the practitioner is willing to trade for allocation flexibility; i.e., retain at least $1 - \varepsilon$ of the proxy's peak performance. We recommend $\varepsilon \in [5\%, 10\%]$, a range in which proxy transfer is reliable across all families, tasks, and budgets we evaluate.

3. **Select the ratio.** Use the proxy sweep to form the ε-optimal region $\widehat{R}_\varepsilon(N_s, B) = [\min \widehat{R}_\varepsilon, \max \widehat{R}_\varepsilon]$, and choose its midpoint. If the final run must use one of the evaluated ratios, use the nearest grid point instead. The midpoint provides a robust choice for transfer: near-optimal ratios form contiguous regions (Sec. 3.3, App. E.2), so the midpoint is less sensitive to small shifts in the optimal region between the proxy and target models than a boundary ratio or the exact proxy optimum.

We test this three-step procedure using our existing 1B–8B runs across all tasks and both the Llama and Qwen 2.5 families. The procedure identifies a target-region ratio that satisfies the chosen tolerance in 94.3% of cases for $\varepsilon = 5\%$ and 97.1% of cases for $\varepsilon = 10\%$. These results support the proxy sweep as a practical approach to selecting robust allocation ratios without exhaustive tuning at the target scale. As post-training scaling can be fragile, exhibiting phenomena such as broken or inverse scaling, metric discontinuities, and cross-scale rank inversions (Caballero et al., 2023; McKenzie et al., 2023; Schaeffer et al., 2025), we focus on understanding the mechanisms underlying this transfer and identifying when the above recipe is reliable, rather than proposing a universal scaling law.

4 Related Work

Budget allocation between training stages. The closest line of work to ours is Raghavendra et al. (2025), who study the trade-off between SFT and preference optimization under fixed annotation budgets and observe a generic trend in the optimal SFT–DPO ratio as a function of total budget: SFT dominates in the low-data regime while preference optimization becomes more favorable at scale. Their analysis identifies this broad trend but does not characterize the optimal-allocation region or its transferability across model scales. Li et al. (2026) study an orthogonal question: given a fixed SFT data budget, when to stop SFT for best downstream RL performance. They propose Adaptive Early-Stop Loss (AESL) and show that diversity-based early stopping (e.g., entropy, self-BLEU) outperforms performance-based stopping criteria due to distributional drift during SFT. Their decision variable is the SFT *stopping step* at a fixed n_{SFT}; ours is the SFT *budget share* relative to RL. The two questions are complementary.

A broader set of works examines budget trade-offs between other training stages: pretraining vs. fine-tuning (Bai et al., 2021), finetuning vs. distillation from a larger teacher (Kang et al., 2023; Busbridge et al., 2025), and the interplay between SFT and RL methods, including DPO–PPO comparisons (Ivison et al., 2024), forgetting (Fernando et al., 2024), generalization (Kirk et al., 2024), and alignment trade-offs (Saeidi et al., 2025). A complementary line of work treats SFT and DPO as connected via an implicit reward formulation (Wang et al., 2025), which motivates the budget-allocation question we study. Our contribution differs from all of the above works in characterizing not a single optimal ratio but the entire ε-near-optimal region and its cross-scale transferability. Additional related work on data mixture optimization and cross-scale extrapolation is deferred to App. G.

5 Conclusion

We studied the problem of dividing a fixed annotation budget between SFT and RL in LLM post-training, framing it in terms of near-optimality rather than exact optimality. Our experiments show that the near-optimal region (i.e., the set of allocations within a fixed performance tolerance of the best observed allocation) is even for small tolerances (2–10%), widens consistently with model scale, and transfers reliably from small proxy models to large target models, yielding a practical strategy that eliminates the need for exhaustive large-scale grid search. These findings hold across tasks, model families, off-policy preference-based methods, and on-policy RL, and extend to settings with cost asymmetry between SFT and RL annotation. A limitation of this work is that our analysis is restricted to two-stage SFT→RL pipelines; extending the near-optimality framework to more complex multi-stage or interleaved schedules and to settings where the annotation cost model is uncertain or adaptive is a promising future direction.

Limitations

Target-task training setting. All experiments train on task-specific data drawn from the same distribution as the evaluation, following the convention of Raghavendra et al. (2025). This in-distribution setting yields comparatively predictable scaling behavior. The out-of-distribution regime, where general-purpose training data (e.g., a broad instruction or preference mix) must transfer to a specific held-out target task, scales less reliably, with larger generalization gaps and frequent non-monotonicity in performance versus budget (Caballero et al., 2023; Kirk et al., 2024). Whether near-optimal allocation regions and their cross-scale transferability extend to OOD evaluation remains an important open question.

Model and data scale ceiling. Our analysis covers model scales up to 14B parameters (leaving Llama 3 70B and Qwen 2.5 32B–72B for future verification) and budgets up to 15k examples per task in the majority setting (cost asymmetry may use more training examples). The data ceiling is at or near the publicly available task-specific training corpus for several of our SFT and RL settings: GSM8K provides ∼7.5K SFT examples and ∼16.6K synthetic preference pairs (Cobbe et al., 2021; Raghavendra et al., 2025); HelpSteer 2 contains ∼20K examples used for both stages (Wang et al., 2024a); and GRPO-IFeval is itself a ∼15K-example training set (Lambert et al., 2025). Within our tested range, the cross-scale transferability trends are generally consistent. Extending verification to frontier scales would entail substantial data collection and computational costs, which may be beyond the scope of academic budgets.

Restricted RL algorithms. Other on-policy algorithms, such as PPO, and other off-policy algorithms, such as SimPO (Meng et al., 2024), are not tested. We mainly compare the most popular and generally used RL policies, namely DPO and GRPO, given computational constraints. However, the consistent results across tasks and settings demonstrate the potential of generalization.

Simplicity of cost model. The notion of budget in our setting is relatively simple; we do not consider training compute, GPU time, or the cost of experimental search. Jointly modeling annotation, compute, and search costs into a single scalar budget is non-trivial and lacks precedent in cross-scale scaling laws. The closest studies treat annotation-vs-compute as a two-way dollar trade-off (Bai et al., 2021; Kang et al., 2023), without the cross-scale transfer that we study.

Ethical Considerations

This work is an empirical study of the allocation of budget between supervised fine-tuning and RL-based post-training. All experiments use publicly released model families (Llama 3, Qwen 2.5, and Qwen 3) and publicly available task-specific datasets under their original licenses. We conduct no human-subjects research and collect or release no new data or models. The pretrained models inherit any biases or representational issues present in their original training data, which we do not separately audit. Our experimental sweep involves on the order of hundreds to thousands of post-training runs across the Cartesian product of model sizes, allocation ratios, budgets, tasks, and algorithms. We use parameter-efficient fine-tuning (LoRA) and cap model sizes at 8B parameters per family (except Qwen2.5 14B model in Fig. 14) to make the sweep tractable on academic compute. Our practical contribution: that near-optimal SFT–RL allocation regions transfer from small proxy models to larger models, is intended to reduce the compute required for principled post-training decisions at scale, broadening access for academics and practitioners. We do not foresee specific dual-use risks beyond those already associated with publicly released LLMs and standard post-training methods.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-001). This research is supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. The Mens, Manus, and Machina (M3S) is an interdisciplinary research group (IRG) of the Singapore MIT Alliance for Research and Technology (SMART) centre. Jingtan Wang is supported by the Institute for Infocomm Research of the Agency for Science, Technology and Research (A*STAR).

References

Fan Bai, Alan Ritter, and Wei Xu. 2021. Pre-train or annotate? domain adaptation with a constrained budget. In *Proc. EMNLP*, pages 5002–5015.

Louis Béthune, David Grangier, Dan Busbridge, Eleonora Gualdoni, marco cuturi, and Pierre Ablin. 2025. Scaling laws for forgetting during finetuning with pretraining data injection. In *Proc. ICML*.

David Brandfonbrener, Nikhil Anand, Nikhil Vyas, Eran Malach, and Sham M. Kakade. 2025. Loss-to-loss prediction: Scaling laws for all datasets. *Transactions on Machine Learning Research*.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russell Webb. 2025. Distillation scaling laws. In *Proc. ICML*.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. 2023. Broken neural scaling laws. In *Proc. ICLR*.

Qi Cao, Ruiyi Wang, Ruiyi Zhang, Sai Ashish Somayajula, and Pengtao Xie. 2025. DreamPRM: Domain-reweighted process reward model for multimodal reasoning. In *Proc. NeurIPS*.

Ernie Chang, Yang Li, Patrick Huber, Vish Vogeti, David Kant, Yangyang Shi, and Vikas Chandra. 2025. AutoMixer: Checkpoint artifacts as automatic data mixers. In *Proc. ACL*, pages 19942–19953.

Yangyi Chen, Binxuan Huang, Yifan Gao, Zhengyang Wang, Jingfeng Yang, and Heng Ji. 2025. Scaling laws for predicting downstream performance in LLMs. *Transactions on Machine Learning Research*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv:2110.14168*.

Nicholas E. Corrado, Julian Katz-Samuels, Adithya M Devraj, Hyokun Yun, Chao Zhang, Yi Xu, Yi Pan, Bing Yin, and Trishul Chilimbi. 2025. AutoMix-Align: Adaptive data mixing for multi-task preference optimization in LLMs. In *Proc. ACL*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Proc. NeurIPS*, pages 10088–10115.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan SU, Markus Kliegl, ZIJIA CHEN, Peter Belcak, Yoshi Suhara, Hongxu Yin, Mostofa Patwary, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. 2025. Nemotron-CLIMB: Clustering-based iterative data mixture bootstrapping for language model pre-training. In *Proc. NeurIPS Datasets and Benchmarks Track*.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2024. DoGE: Domain reweighting with generalization estimation. In *Proc. ICML*, pages 12895–12915. PMLR.

Heshan Fernando, Han Shen, Parikshit Ram, Yi Zhou, Horst Samulowitz, Nathalie Baracaldo, and Tianyi Chen. 2024. Understanding forgetting in LLM supervised fine-tuning and preference learning–a convex optimization perspective. *arXiv:2410.15483*.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Luca Soldaini, Jenia Jitsev, Alex Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, and 6 others. 2025. Language models scale reliably with over-training and on downstream tasks. In *Proc. ICLR*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *Proc. ICML*, pages 10835–10866.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv:2203.15556*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. 2024. Unpacking DPO and PPO: Disentangling best practices for learning from preference feedback. In *Proc. NeurIPS*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, and 1 others. 2023. Camels in a changing climate: Enhancing LM adaptation with Tülu 2. *arXiv:2311.10702*.

Yiding Jiang, Allan Zhou, Zhili Feng, Sadhika Malladi, and J Zico Kolter. 2025. Adaptive data optimization: Dynamic sample selection with scaling laws. In *Proc. ICLR*.

Junmo Kang, Wei Xu, and Alan Ritter. 2023. Distill or annotate? cost-efficient fine-tuning of compact models. In *Proc. ACL*, pages 11100–11119.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv:2001.08361*.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, and 1 others. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proc. NAACL HLT*, pages 4110–4124.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of RLHF on LLM generalisation and diversity. In *Proc. ICLR*.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. 2024. Understanding catastrophic forgetting in language models via implicit inference. In *Proc. ICLR*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025. Tülu 3: Pushing frontiers in open language model post-training. In *Proc. COLM*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and 1 others. 2024. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proc. ICML*, pages 26874–26901.

Xiaoyi Li. 2026. Do post-training algorithms actually differ? a controlled study across model scales uncovers scale-dependent ranking inversions. *arXiv:2603.19335*.

Xinran Li, Guangda Huzhang, Siqi Shen, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Jun Zhang. 2026. Getting your LLMs ready for reinforcement learning with lightweight SFT. In *Proc. ICLR*.

Yuan Li, Zhengzhong Liu, and Eric P. Xing. 2025a. Data mixing optimization for supervised fine-tuning of large language models. In *Proc. ICML*.

Zeman Li, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. 2025b. PiKE: Adaptive data mixing for large-scale multi-task learning under low gradient conflicts. In *Proc. NeurIPS*.

Yiqing Liang, Jielin Qiu, Wenhao Ding, Zuxin Liu, James Tompkin, Mengdi Xu, Mengzhou Xia, Zhengzhong Tu, Laixi Shi, and Jiacheng Zhu. 2025. MoDoMoDo: Multi-domain data mixtures for multimodal LLM reinforcement learning. *arXiv:2505.24871*.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2025. RegMix: Data mixture as regression for language model pre-training. In *Proc. ICLR*.

Ian R. McKenzie, Alexander Lyzhov, Michael Martin Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Xudong Shen, Joe Cavanagh, Andrew George Gritsevskiy, Derik Kauffman, Aaron T. Kirtland, Zhengping Zhou, Yuhui Zhang, Sicong Huang, Daniel Wurgaft, Max Weiss, Alexis Ross, Gabriel Recchia, and 7 others. 2023. Inverse scaling: When bigger isn't better. *Transactions on Machine Learning Research*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. In *Proc. NeurIPS*.

Xu Ouyang, Shengzhuang Chen, Michael Arthur Leopold Pearce, Thomas Hartvigsen, and Jonathan Richard Schwarz. 2025. ADMIRE-BayesOpt: Accelerated data MIxture RE-weighting for language models with Bayesian optimization. *Transactions on Machine Learning Research*.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, W. Bradley Knox, Chelsea Finn, and Scott Niekum. 2024. Scaling laws for reward model overoptimization in direct alignment algorithms. In *Proc. NeurIPS*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proc. NeurIPS*.

Mohit Raghavendra, Junmo Kang, and Alan Ritter. 2025. Balancing the budget: Understanding trade-offs between supervised and preference-based finetuning. In *Proc. ACL*, pages 25702–25720.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. 2024. Observational scaling laws and the predictability of language model performance. In *Proc. NeurIPS*.

Amir Saeidi, Shivanshu Verma, Md Nayem Uddin, and Chitta Baral. 2025. Insights into alignment: Evaluating DPO and its variants across multiple tasks. In *Proc. ACL Student Research Workshop*, pages 409–421.

Rylan Schaeffer, Hailey Schoelkopf, Brando Miranda, Gabriel Mukobi, Varun Madan, Adam Ibrahim, Herbie Bradley, Stella Biderman, and Sanmi Koyejo. 2025. Why has predicting downstream capabilities of frontier AI models with scale remained elusive? In *Proc. ICML*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*.

Wei Shen, Guanlin Liu, YuYue, Ruofei Zhu, Qingping Yang, Chao Xin, and Lin Yan. 2025. Exploring data scaling trends and effects in reinforcement learning from human feedback. In *Proc. NeurIPS*.

Kai Shi, Jun Yang, Ni Yang, Binqiang Pan, Qingsong Xie, Zhangchao, Zhenyu Yang, Tianhuang Su, and Haonan Lu. 2026. DaMo: Data mixing optimizer in fine-tuning multimodal LLMs for mobile phone agents. In *Proc. ACL Findings*.

Mustafa Shukor, Louis Béthune, Dan Busbridge, David Grangier, Enrico Fini, Alaaeldin El-Nouby, and Pierre Ablin. 2025. Scaling laws for optimal data mixtures. In *Proc. NeurIPS*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize from human feedback. In *Proc. NeurIPS*, pages 3008–3021.

Haoru Tan, Sitong Wu, Yanfeng Chen, Jun Xia, Ruobing Xie, Bin Xia, Xingwu Sun, and Xiaojuan Qi. 2026. Fast data mixture optimization via gradient descent. In *Proc. ICLR*.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proc. Workshop on New Frontiers in Summarization*. Association for Computational Linguistics.

Bo Wang, Qinyuan Cheng, Runyu Peng, Rong Bao, Peiji Li, Qipeng Guo, Linyang Li, Zhiyuan Zeng, Yunhua Zhou, and Xipeng Qiu. 2025. Implicit reward as the bridge: A unified view of SFT and DPO connections. In *Proc. NeurIPS*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning language models with self-generated instructions. In *Proc. ACL*, pages 13484–13508.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh N Sreedhar, and Oleksii Kuchaiev. 2024a. HelpSteer2: Open-source dataset for training top-performing reward models. In *Proc. NeurIPS Datasets and Benchmarks Track*, pages 1474–1501.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Scowcroft, Neel Kant, Aidan Swope, and 1 others. 2024b. Helpsteer: Multi-attribute helpfulness dataset for steerlm. In *Proc. of NAACL*, pages 3371–3384.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Proc. NeurIPS*, pages 69798–69818.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024a. Qwen2.5 technical report. *arXiv:2412.15115*.

Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. 2021. Tuning large neural networks via zero-shot hyperparameter transfer. In *Proc. NeurIPS*, pages 17084–17097.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024b. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. In *Proc. ICML*.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. 2025. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *Proc. ICLR*.

Thomson Yen, Andrew Wei Tung Siah, Haozhe Chen, C. Daniel Guetta, Tianyi Peng, and Hongseok Namkoong. 2025. Data mixture optimization: A multi-fidelity multi-scale bayesian framework. In *Proc. NeurIPS*.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. 2024. When scaling meets LLM finetuning: The effect of data, model and finetuning method. In *Proc. ICLR*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023a. LIMA: Less is more for alignment. In *Proc. NeurIPS*, pages 55006–55021.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *arXiv:2311.07911*.

Tong Zhu, Daize Dong, Xiaoye Qu, Jiacheng Ruan, Wenliang Chen, and Yu Cheng. 2025a. Dynamic data mixing maximizes instruction tuning for mixture-of-experts. In *Proc. NAACL*, pages 1663–1677.

Xiaoxuan Zhu, Zhouhong Gu, Baiqian Wu, Suhang Zheng, Tao Wang, Tianyu Li, Hongwei Feng, and Yanghua Xiao. 2025b. ToReMi: Topic-aware data reweighting for dynamic pre-training data selection. *arXiv:2504.00695*.

Table of Contents

Appendix 14

A Experimental Details 14
- A.1 Tasks . . . 14
- A.2 Hyper-parameter settings . . . 14

B Complete Performance Curves with Different Budgets 14

C Complete Analysis Figures with Different Settings 18

D Analysis with Additional Model Family and Model Scale 18

E Additional Ablation Study 21
- E.1 Ablation Study 1: Grid Resolution 21
- E.2 Ablation Study 2: Count-Width Analysis . . . 21
- E.3 Ablation Study 3: Multi-Seed Validation . . . 23
- E.4 Ablation Study 4: Low-Budget Regime . . . 24
- E.5 Ablation Study 5: Robustness to Training Hyperparameters . . . 24
- E.6 Ablation Study 6: Absolute-Tolerance Ratio . . . 25
- E.7 Ablation Study 7: Proxy Versus Fixed Ratio Comparison . . . 26

F Additional Results for Cost Asymmetry Setting 26

G Additional Related Work on Mixture Optimization and Scaling Law 28

H Additional Discussion and Clarifications 29

A Experimental Details

A.1 Tasks

Dataset construction details. Our dataset construction largely follows Raghavendra et al. (2025). For each task, we construct an SFT dataset consisting of prompt-response pairs and an RL-stage dataset consisting of either preference pairs for DPO or prompts with reward signals for GRPO. We use the same dataset sources, pre-processing rules, and evaluation protocols as in the setup of Raghavendra et al. (2025), unless otherwise stated. For DPO, preference data are formatted as triples (x, y_w, y_l), where y_w and y_l denote the preferred and rejected responses. For GRPO, we reuse the same prompts x from the DPO data and task-specific evaluation metrics as rewards. Note that for summarization, we use the preferred column's response as the reference answer when calculating the Rouge-L. The only deviation is the instruction-following task, where we use the original Tülu3 RLVR instruction-following dataset for GRPO, since constructing evaluation instructions that serve as the reward directly from the preference-dataset prompts is less straightforward. For the evaluation datasets, we follow the standard benchmark splits for GSM8K (Cobbe et al., 2021) and IFEval (Zhou et al., 2023b), and use the official test split of the summarization dataset (Völske et al., 2017). For HelpSteer, we construct a local 200-example held-out test split for evaluation. Dataset sources and download links are listed in Tab. 2.

A.2 Hyper-parameter settings

Hyperparameter settings for each stages we used are in Tables 3 to 5. For all experiments, we use LoRA with rank $r = 32$, scaling factor $\alpha = 32$, dropout of 0, no bias terms, and target the attention modules. The main experiments are run with seed 42 on L40s and H200s.

B Complete Performance Curves with Different Budgets

Figs. 7–10 report raw performance curves across budgets for each allocation ratio. Two patterns are visible. First, in the early-stage training ($B < 5$k), especially for small models and, most notably, for instruction-following tasks in the Llama family and math tasks in the Qwen 2.5 family, the curves can be noisy and occasionally non-monotonic, so the relative ordering of allocation ratios fluctuates with budget. Because of the instability of

Task	Setting	Dataset / Download Link
Grade School Math	SFT	GSM8K generations: https://huggingface.co/datasets/mohit-raghavendra/gsm8k-gpt4ogenerations
	RL	GSM8K preferences: https://huggingface.co/datasets/mohit-raghavendra/gsm8k-gpt4opreferences
Instruction Following	SFT	Tülu3 Instruction Following: https://huggingface.co/datasets/allenai/tulu-3-sft-personas-instruction-following
	DPO	Tülu3 Instruction Following preferences: https://huggingface.co/datasets/allenai/tulu-3-pref-personas-instruction-following
	GRPO	GRPO-IFeval: https://huggingface.co/datasets/allenai/RLVR-IFeval
	Evaluation	IFEval: https://github.com/google-research/google-research/tree/master/instruction_following_eval
Summarization	SFT	Reddit TL;DR: https://huggingface.co/datasets/UCL-DARK/openai-tldr-filtered
	RL	Reddit comparison dataset: https://huggingface.co/datasets/UCL-DARK/openai-tldr-summarisation-preferences
Helpfulness	SFT	HelpSteer1: https://huggingface.co/datasets/nvidia/HelpSteer; HelpSteer2: https://huggingface.co/datasets/nvidia/HelpSteer2
	RL	HelpSteer1: https://huggingface.co/datasets/nvidia/HelpSteer; HelpSteer2: https://huggingface.co/datasets/nvidia/HelpSteer2
	Evaluation	Reward model: https://huggingface.co/Ray2333/gpt2-large-helpful-reward_model

Table 2: Detailed dataset and evaluation references for reproducibility. For each task, we report the training and evaluation setting together with the corresponding public dataset or benchmark source.

Parameter	Value
Learning Rate	1×10^{-5}
Effective Batch Size	16
Number of Epochs	1 (2 for Instruction tuning)
Warmup Ratio	0.1
Learning Rate Scheduler	cosine
Optimizer	adamw_8bit
Weight Decay	0.01

Table 3: SFT Hyperparameter Settings

Parameter	Value
Learning Rate	1×10^{-6}
Effective Batch Size	16
Number of Epochs	1 (2 for Instruction tuning)
Warmup Ratio	0.1
Learning Rate Scheduler	cosine
Optimizer	adamw_8bit
Weight Decay	0.01
β	0.1

Table 4: DPO Hyperparameter Settings

early stages, we restrict our scaling and transfer analysis to a more stable training regime, which is consistent with established practice in scaling-behavior analysis work: power-law learning-curve fits are known to break down during early training (Kaplan et al., 2020); low-compute points exhibit systematically larger fitting residuals and are routinely downweighted in scaling-law fits (Hoffmann et al., 2022); and small-scale points can become uninformative before asymptotic regime transitions (Caballero et al., 2023). In the fine-tuning regime specifically, stability has been observed to require a minimum data threshold: e.g., approximately 2K examples in Zhou et al. (2023a). Together, these motivate an analysis of the $B \geq 5$k regime, where post-training curves have stabilized.

Parameter	Value
Learning Rate	1×10^{-6}
Effective Batch Size	16
Number of Epochs	1 (2 for Instruction tuning)
Warmup Ratio	0.1
Learning Rate Scheduler	cosine
Optimizer	adamw_8bit
Weight Decay	0.01
Number of Generations	4
Temperature	0.9
Top-p	1.0
Top-k	-1
Minimum-p	0.0
Repetition Penalty	1.0

Table 5: GRPO Hyperparameter Settings

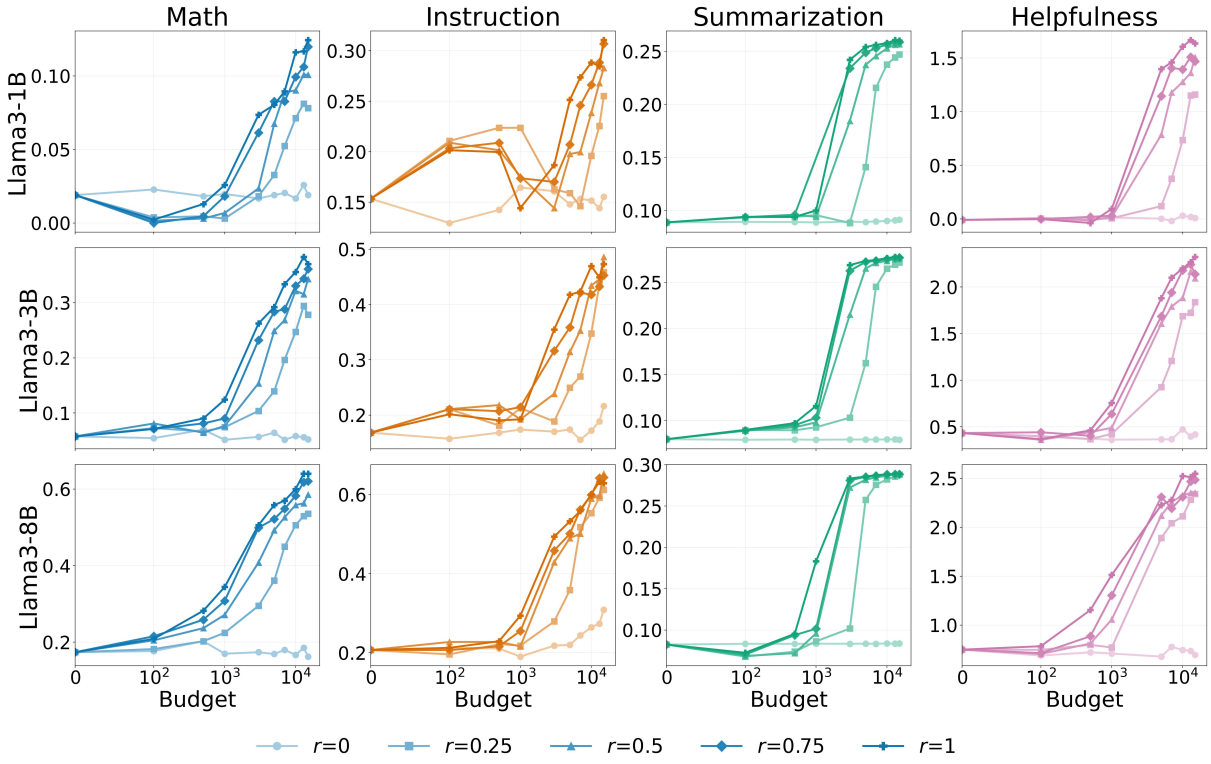


Figure 7: Performance versus training budget across SFT–DPO allocation ratios on the Llama 3 model family, organized by model size (rows) and tasks (columns). Curves can be noisy in the small-budget regime ($B < 5$k) and stabilize for $B \geq 5$k, where ratios converge to comparable downstream performance: consistent with the broad near-optimal regions in Sec. 3.2.

Beyond the general low-budget instability noted above, the Qwen-on-summarization panels exhibit a distinct task-specific pattern: SFT performance is non-monotonic across all model scales, first decreasing before recovering at larger budgets. This contrasts with the other (task, model) combinations, where SFT either improves monotonically or shows only a brief initial dip in the smallest model, which resolves quickly. We hypothesize that the persistent warmup in Qwen summarization may reflect a distribution mismatch. When the fine-tuning distribution differs substantially from the pretraining distribution, early SFT updates can push the model toward sub-optimal regions before sufficient task-specific signal accumulates (Kotha et al., 2024). Qwen's pretraining distribution and

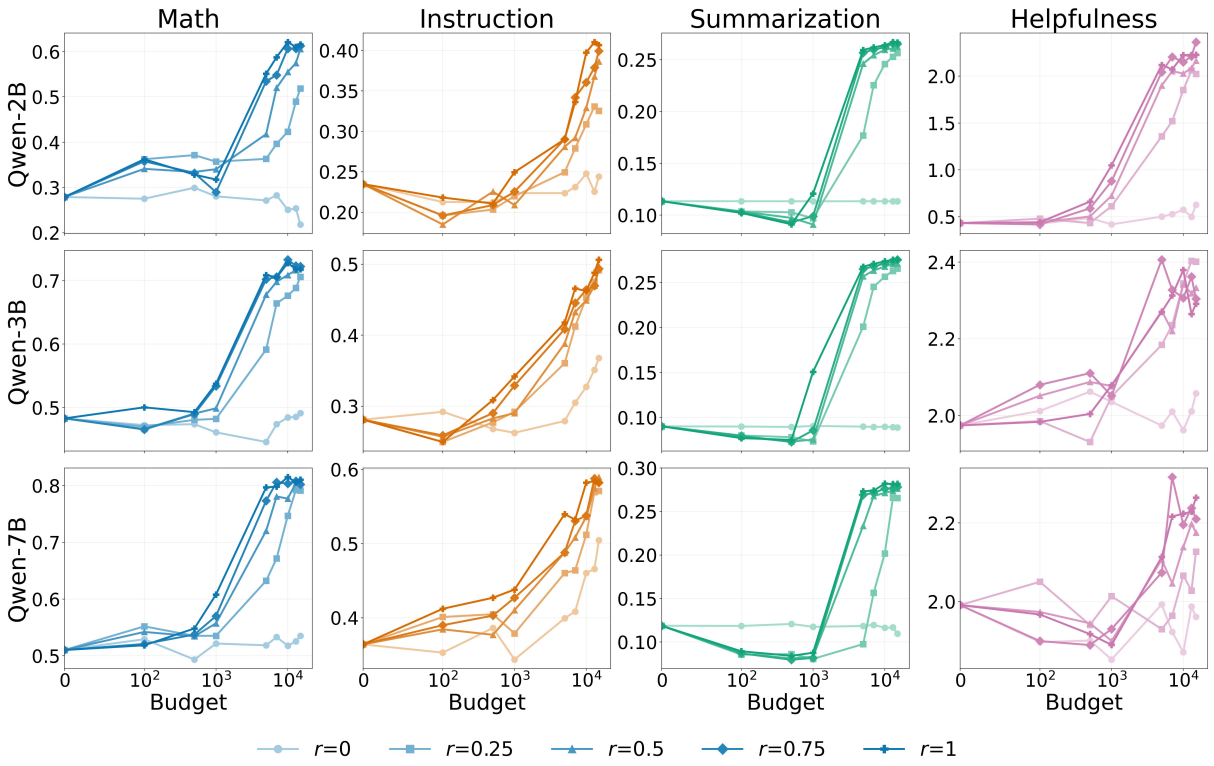


Figure 8: Performance versus training budget across SFT–DPO allocation ratios on the Qwen 2.5 model family. Qualitative trends align with Fig. 7.

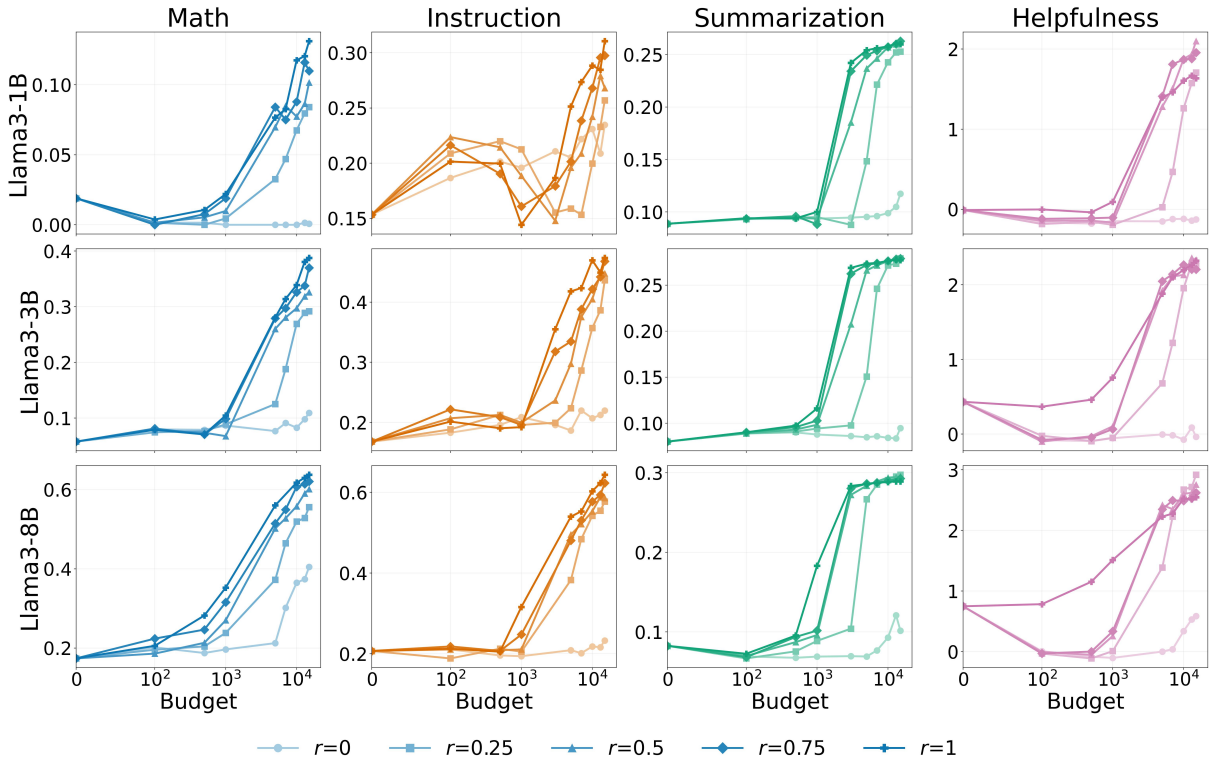


Figure 9: Performance versus training budget across SFT–GRPO allocation ratios on the Llama 3 model family. Qualitative trends align with Fig. 7.

the English-centric Reddit TL;DR distribution (Stiennon et al., 2020) can be far in this sense. Another behavior is for Qwen 7B on helpfulness, zero-shot performance is already strong (∼2.0) and post-

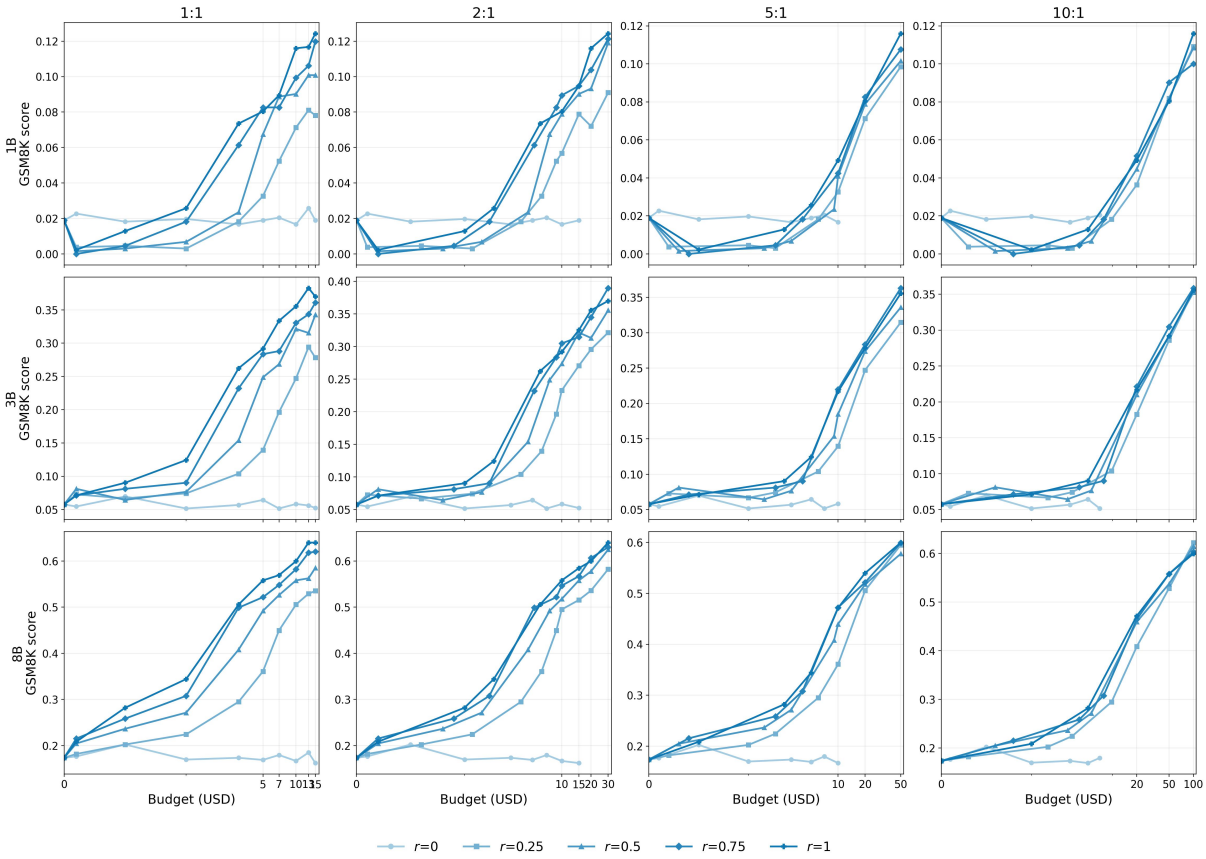


Figure 10: Full performance curves under heterogeneous SFT–DPO annotation costs for the Llama 3 model family on math task, organized by model size (rows) and SFT-to-DPO cost ratio ρ in $\{1, 2, 5, 10\}$ (columns). Within each panel, line correspond to SFT allocation ratios $r \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$; the x-axis shows the dollar-equivalent training budget. As ρ increases (moving right within each row), the spread of curves narrows, consistent with the widening near-optimal allocation region under cost asymmetry reported in Sec. 3.4.

training only marginally improves it ($\sim$2.3 at the larger budget). At the same time, the smaller Qwen 3B reaches $\sim$2.4 on the same task. A broad near-optimal region on this task for 7B may partly reflect saturation: when no allocation meaningfully improves the model, all allocations are trivially near-optimal, rather than genuine allocation invariance. In terms of second patterns, as B increases past 5k, curves stabilize, and the gap between allocation ratios narrows visibly: multiple ratios converge to similar performance. This provides direct visual support for the broad near-optimal regions documented in Sec. 3.2: at larger budget, the choice of SFT–RL allocation has less impact on downstream performance, with the near-optimal region widening as budget grows.

C Complete Analysis Figures with Different Settings

We list the complete analysis figures with mean tolerance-optimal region vs. model size in the first row; cross-size overlap of tolerance-optimal ratio regions in the second row, and per-ratio hit rate across budgets in the third row for Qwen 3 family in Fig. 11, SFT–GRPO setting in Fig. 12, and cost asymmetry setting in Fig. 13.

D Analysis with Additional Model Family and Model Scale

To test whether these trends are tied to the scales and families of our main sweep, we extend the analysis along two axes: a more recent family (Qwen 3, 1.7B/4B/8B), and a larger target within Qwen 2.5 (14B), on math and instruction following, with results in Fig. 11 and Fig. 14.

While the near-optimal region generally trends wider with scale, we observe that this widening is occasionally non-monotonic, with slight contractions at certain scales. However, the 14B scale is far wider than the 2B (Qwen2.5 1.5B model) proxy; on math, the width can dip slightly at 14B while transfer stays strong. Crucially, our transferabil-

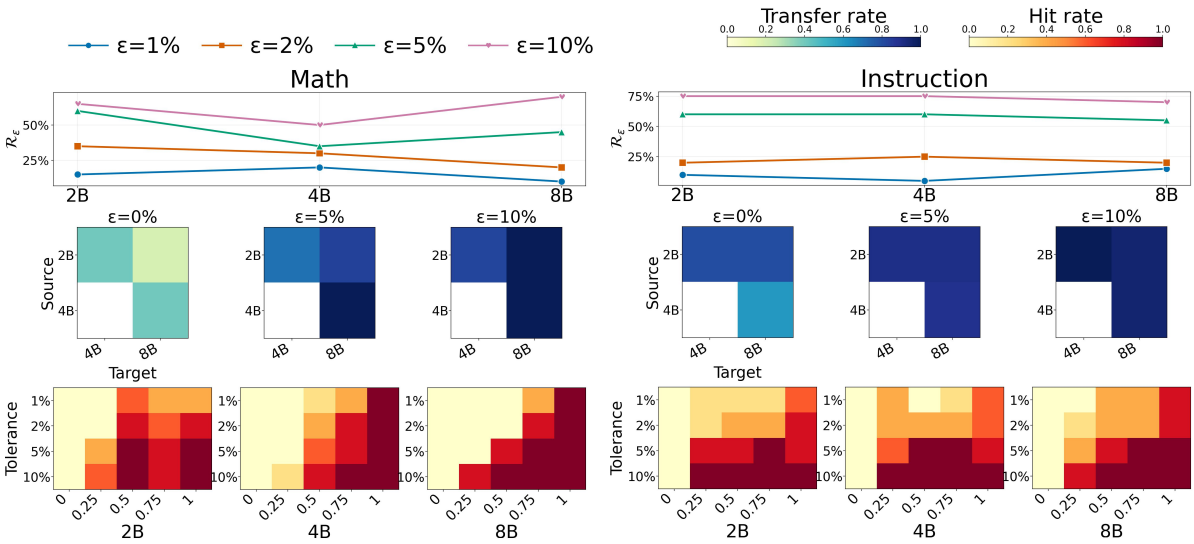


Figure 11: **Main analysis for Qwen3 family with SFT–DPO training stages.** For brevity, 2B in the figures denotes the Qwen3-1.7B model.

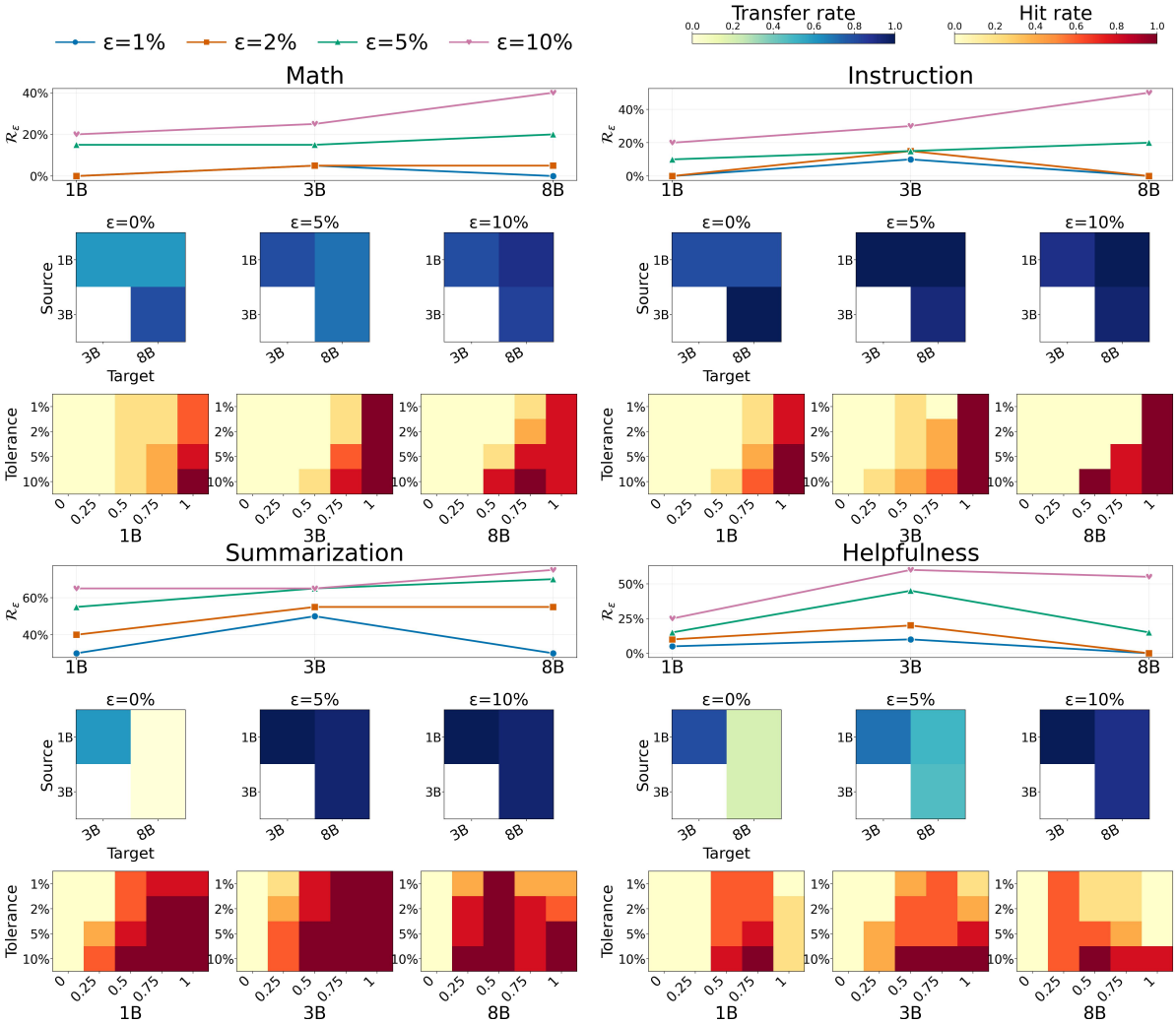


Figure 12: **Main analysis for Llama family with SFT–GRPO training stages.**

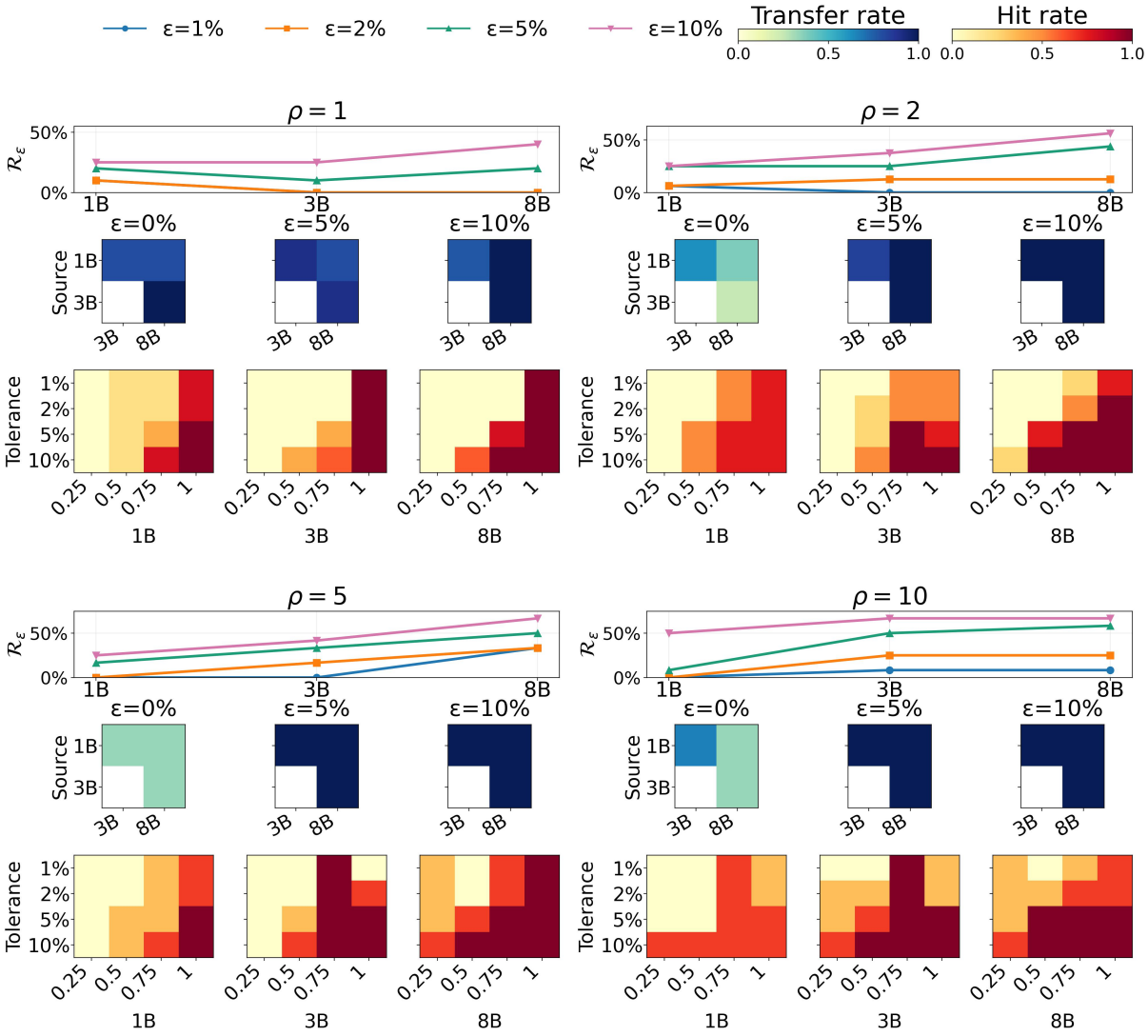


Figure 13: **Effect of heterogeneous annotation costs on transferable allocation regions for Math task.**

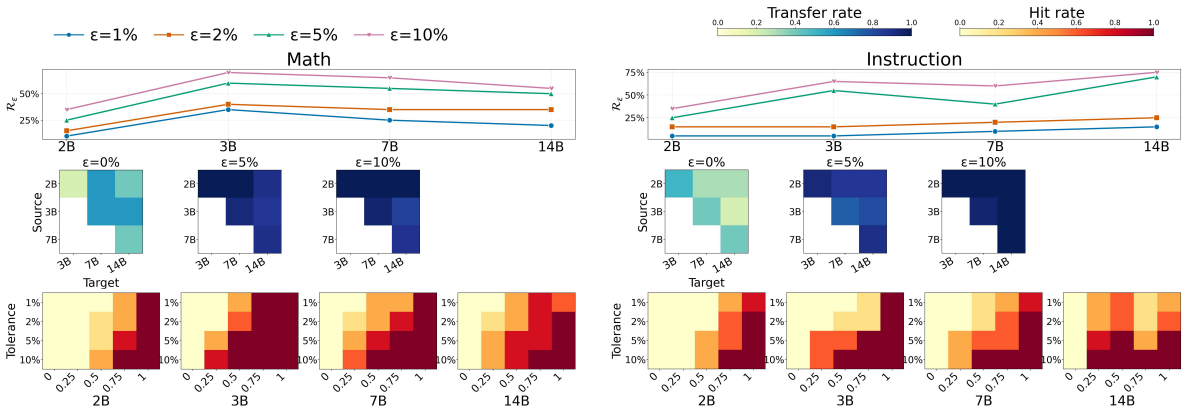


Figure 14: **Main analysis for Qwen2.5 family up to 14B model scale with SFT–DPO training stages.**

ity remains highly robust despite these local width fluctuations, further validating the reliability of the near-optimal region.

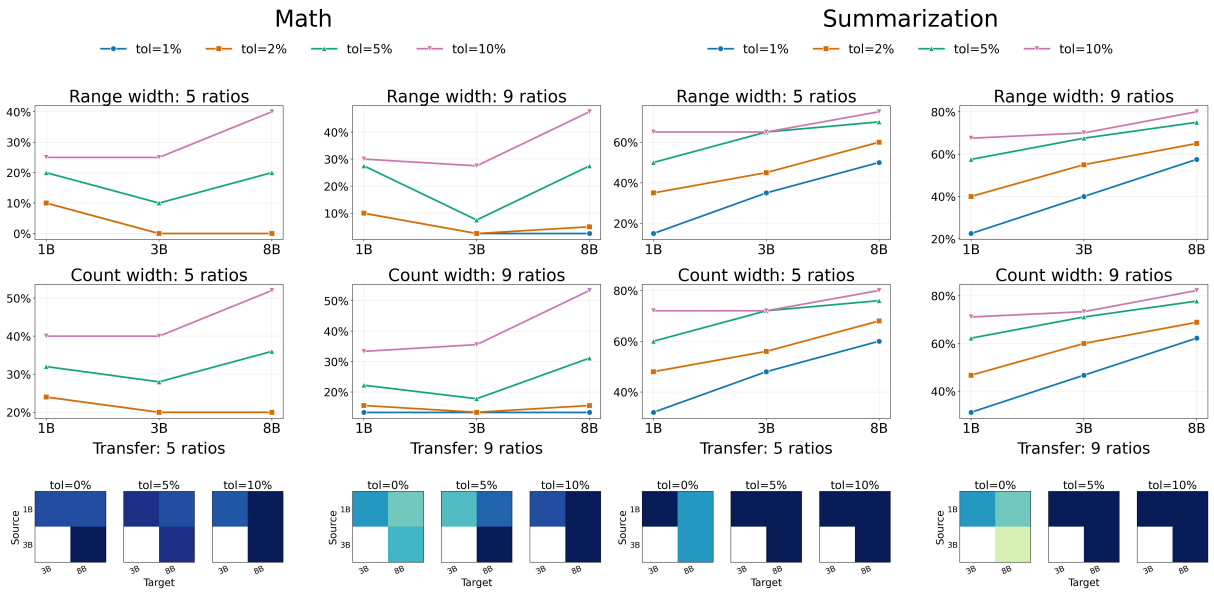


Figure 15: Validation of the 5-point allocation grid on math and summarization task. For each task: Left: range-width $w_\varepsilon^{\mathrm{rng}}$; $w_\varepsilon^{\mathrm{cnt}}$ and transfer hit computed from the 5-point grid $\mathcal{G} = \{0.00, 0.25, 0.50, 0.75, 1.00\}$. Right: range-width computed from a denser 9-point grid adding intermediate ratios $\{0.125, 0.375, 0.625, 0.875\}$. Both grids yield qualitatively similar trends across tolerance and model scale, supporting the use of the 5-point grid as the primary protocol.

E Additional Ablation Study

E.1 Ablation Study 1: Grid Resolution

To assess whether the 5-point allocation grid $\mathcal{G} = \{0.00, 0.25, 0.50, 0.75, 1.00\}$ used in our main experiments is sufficient for the conclusions we draw, we conduct a density-validation study on math (GSM8K) and the summarization task for the Llama 3 family. We compare the range-width estimator $w_\varepsilon^{\mathrm{rng}}$ and count-width estimator $w_\varepsilon^{\mathrm{cnt}}$ computed from the 5-point grid against the same estimator(s) computed from a denser 9-point grid that adds intermediate ratios $\{0.125, 0.375, 0.625, 0.875\}$:

$$\mathcal{G}_9 = \{0.000, 0.125, 0.250, 0.375, 0.500, \\ 0.625, 0.750, 0.875, 1.000\}.$$

Fig. 15 reports the estimates from both grids across tolerance thresholds and model scales. Near-optimal range width and transferability trends are qualitatively preserved. The 9-point estimates agree with the 5-point estimates in direction and approximate magnitude across tested (N, ε) combinations, supporting the use of the 5-point grid as our protocol. The results confirm the 5-point grid resolution is sufficient to capture the qualitative scaling and transferability behavior reported in the main text at a tractable experimental cost.

E.2 Ablation Study 2: Count-Width Analysis

We complement the range-width analysis in the main text with the count-width estimator

$$w_\varepsilon^{\mathrm{cnt}}(N, B) \;=\; |\widehat{\mathcal{R}}_\varepsilon(N, B)| \,/\, |\mathcal{G}|\,,$$

the fraction of evaluated grid candidates within the ε-near-optimal region (Sec. 3.1). Unlike the range estimator, count is sensitive to grid gaps and therefore directly diagnoses whether $\widehat{\mathcal{R}}_\varepsilon$ is non-contiguous on the grid. The choice of range vs. count affects only the scalar width statistic (Row 1 of each main-paper analysis figure); the transfer rate T_ε and per-ratio hit-rate heatmaps (Rows 2 and 3) depend only on the set $\widehat{\mathcal{R}}_\varepsilon$ itself and are therefore identical to those reported in the main paper regardless of width estimator. We accordingly report only the count-width vs. model-size panels here and refer the reader to the main text and App. C for transfer and hit-rate results.

Figs. 16–21 report the count-width analysis across five experimental settings: sensitivity of budget allocation (Fig. 1), Llama SFT–DPO (Fig. 2), Qwen 2.5 SFT–DPO (Fig. 3), Qwen 3 SFT–DPO (Fig. 4), Llama SFT–GRPO (Fig. 5), and the heterogeneous-cost setting under DPO (Fig. 6). Across all settings, the count-based results generally replicate the qualitative trends of the range-width analysis in the main text: (i) the near-optimal

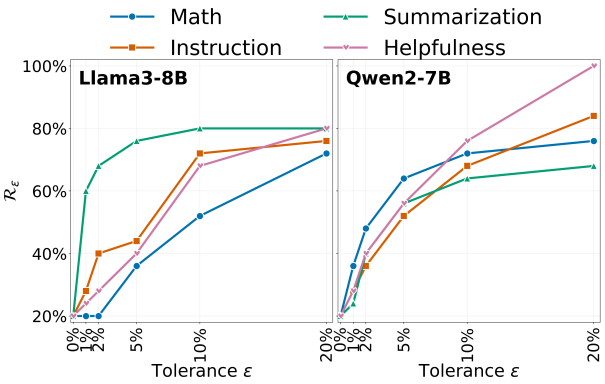


Figure 16: Tolerance ε versus near-optimal-region width $\mathcal{R}_\varepsilon = (w_\varepsilon^{\text{cnt}})$ for Llama3-8B and Qwen2.5-7B across four tasks. The near-optimal region expands consistently with tolerance across tasks and model families.

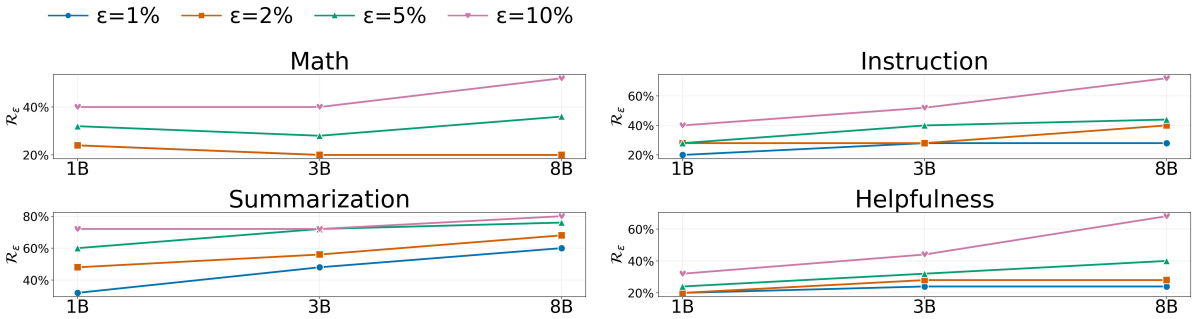


Figure 17: Near-optimal-region width $\mathcal{R}_\varepsilon = (w_\varepsilon^{\text{cnt}})$ vs. model size N for Llama-family. Each panel shows one task; lines correspond to different tolerance thresholds ε. Trends qualitatively replicate the range-width results in Fig. 2. The corresponding transfer rates and per-ratio hit-rate heatmaps are estimator-invariant and reported in the main paper.

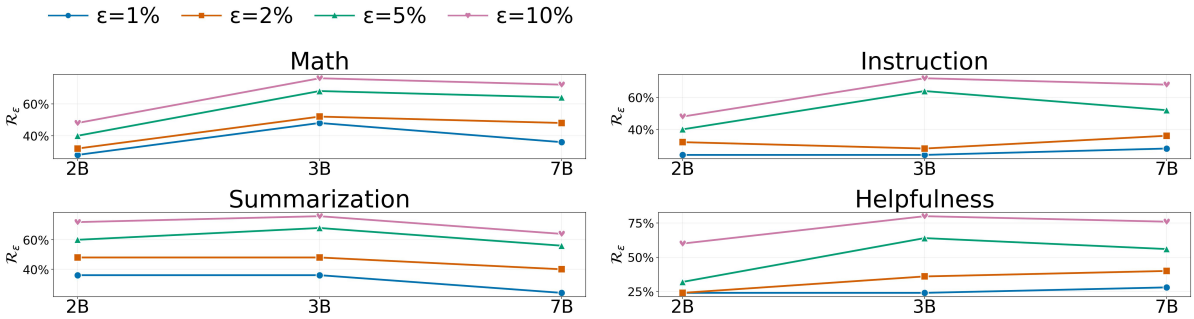


Figure 18: Near-optimal-region width $\mathcal{R}_\varepsilon = (w_\varepsilon^{\text{cnt}})$ vs. model size N for Qwen 2.5-family.

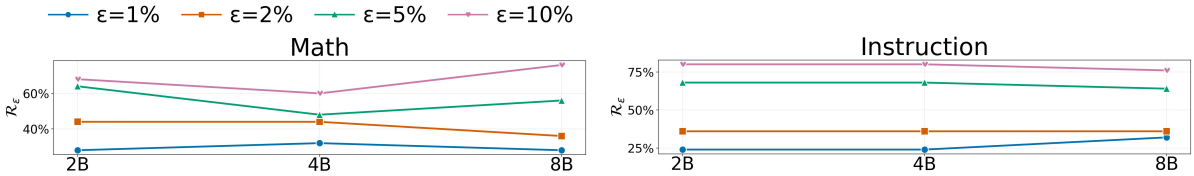


Figure 19: Near-optimal-region width $\mathcal{R}_\varepsilon = (w_\varepsilon^{\text{cnt}})$ vs. model size N for Qwen 3-family.

region widens with tolerance at all model scales; and (ii) width increases with model scale under a fixed tolerance.

The qualitative agreement between range and count estimators supports the contiguity argument from Sec. 3.1. The observed agreement, along with

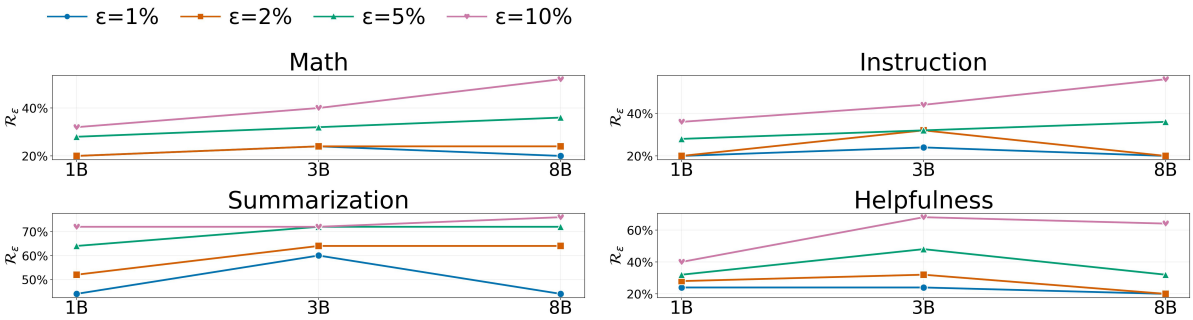


Figure 20: Near-optimal-region width $\mathcal{R}_\varepsilon = (w_\varepsilon^{\text{cnt}})$ vs. model size N for Llama-family for SFT:GRPO Setting.

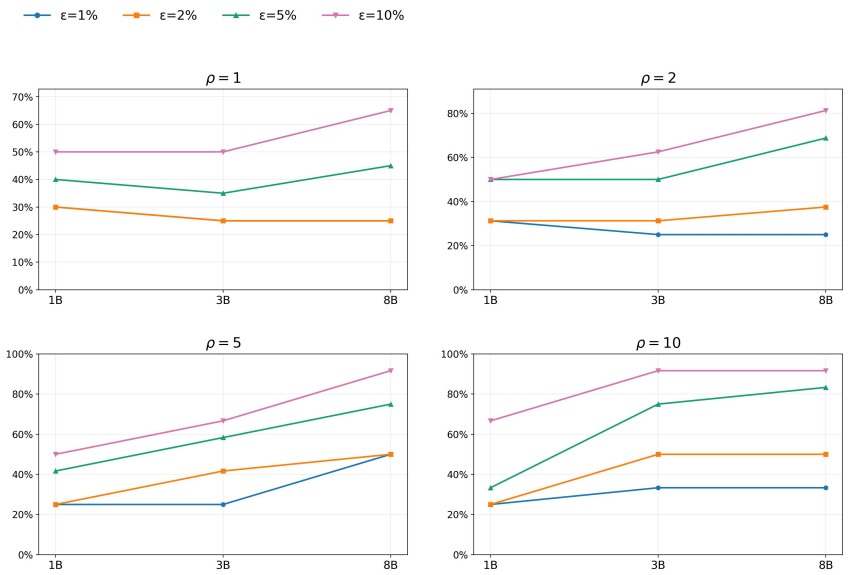


Figure 21: Near-optimal-region width $\mathcal{R}_\varepsilon = (w_\varepsilon^{\text{cnt}})$ vs. model size N for Llama-family for cost-asymmetric setting.

the previous ablation study (App. E.1), empirically validates treating $\widehat{\mathcal{R}}_\varepsilon$ as approximately contiguous on $\mathcal{G}$ and justifies our choice to report range-based width as the primary statistic in the main text.

E.3 Ablation Study 3: Multi-Seed Validation

The full experimental sweep covers the Cartesian product of model sizes, allocation ratios, budgets, tasks, algorithms (DPO, GRPO), and model families (Llama 3, Qwen 2.5). Running each configuration with multiple seeds would multiply this cost by the number of seeds. We therefore report single-seed point estimates in the main text and conduct a 3-seed validation to assess whether the single-seed estimates are sufficient for the qualitative trends we report. We run this validation on math (GSM8K) and summarization for the Llama 3 family, in more compute-intensive settings of our sweep. Seed variation perturbs both the random subset of training samples drawn from the source pool and the training order within each stage.

Fig. 22 reports three views:

- **Row 1 (performance curves):** mean performance versus budget across SFT–DPO allocation ratios is generally stable across seeds.
- **Row 2 (near-optimal region width versus tolerance):** the mean width curve is generally consistent across seeds in shape.
- **Row 3 (cross-scale transfer hit rates at $\varepsilon = 10\%$):** We report per-seed hit-rate matrices rather than averaging, since transfer hit rates are integer-valued and averaging across seeds yields fractional values that are relatively hard to interpret. The qualitative trend, higher hit rates at larger tolerance, similar magnitude across all three seeds, is preserved.

Together, these results support the use of single-seed point estimates for the qualitative phenomena reported in the main text. Obtaining multi-seed variance bars across the full experimental sweep

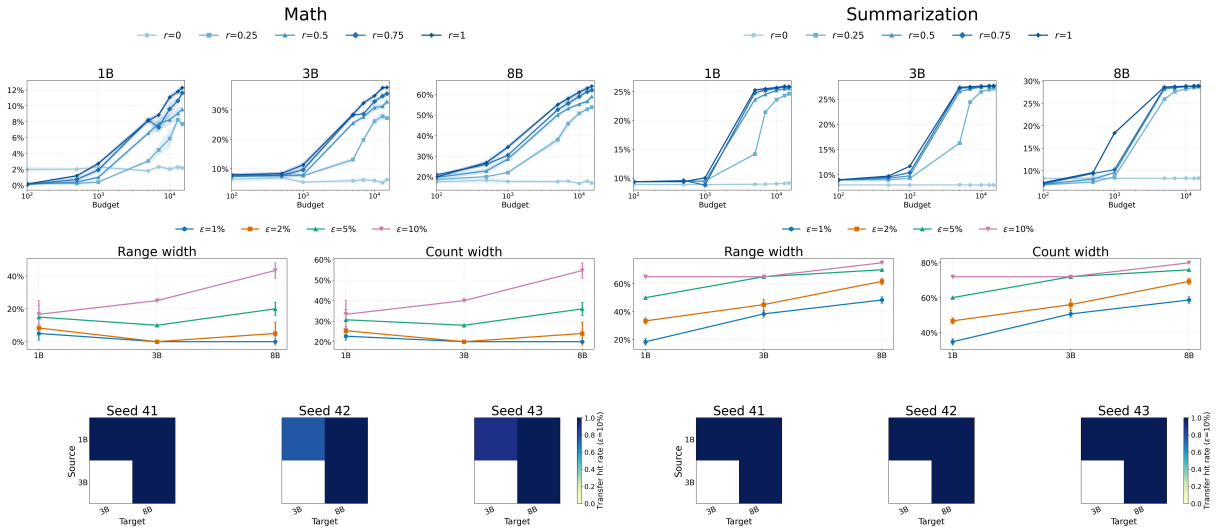


Figure 22: Three-seed validation on math (left) and summarization (right) (Llama 3 family). **Row 1:** Performance curves versus training budget across SFT–DPO allocation ratios; lines show mean and shaded bands show ± 1 standard deviation across 3 seeds. **Row 2:** Mean near-optimal region width versus model size N per tolerance level ε; lines show mean and bands show ± 1 standard deviation across seeds.**Row 3:** Per-seed cross-scale transfer hit-rate matrices at $\varepsilon = 10\%$; each column corresponds to one seed. Seed variation perturbs both the choice of training samples and the training order. Mean trends in Rows 1–2 are stable across seeds, and the qualitative transfer pattern in Row 3 is consistent across all three seeds, supporting the validity of qualitative analyses discovered with single-seed estimates.

would require substantially more compute. However, the consistency observed here suggests that quantitative variance bars would not change the qualitative conclusions.

E.4 Ablation Study 4: Low-Budget Regime

As discussed in Sec. 3.1 and App. B, performance curves in the small-budget regime can be noisy and occasionally non-monotonic. The main analysis, therefore, focuses on the stable $B \geq 5$k regime where cross-model-scale scaling and transferability trends can be cleanly isolated. This ablation extends the analysis to lower budgets to verify that the qualitative findings persist, though somewhat attenuated when the unstable low-data regime is included. Fig. 23 reports the extended-range results. Here are our two key observations:

- **Qualitative trends persist when low-budget points are included.** The widening of $\mathcal{R}_\varepsilon$ with model scale and the cross-scale transferability of near-optimal regions remain observable even when budgets below 5k are included, confirming that the main-text findings are not contingent on the $B \geq 5$k regime.
- **Attenuated effects at the lower budgets, especially for IFEval.** Consistent with the run-to-run instability documented in App. B, the cross-scale trends in $\mathcal{R}_\varepsilon$ and T_ε are weaker when smaller budgets are included, most noticeably for instruction-following (IFEval). This attenuation reflects the fact that noise can dominate the signal at a lower budget, rather than a breakdown of the underlying transferability mechanism.

At $B \sim 1$k, the post-training pipeline can be exhaustively swept directly at lower cost, and the optimal allocation shifts toward SFT-only configurations (Raghavendra et al., 2025). Our transferability framework is not intended to replace direct grid search in this regime; our value emerges when full-scale search becomes computationally prohibitive: the regime where the B passes the early stages phase that we report most directly applies.

E.5 Ablation Study 5: Robustness to Training Hyperparameters

Our main sweep fixes one SFT and one RL configuration per family. To check that the trends are not an artifact of that choice, we re-ran the instruction-following analysis on the Llama family (1B/3B/8B) under two perturbations: with fewer epochs or doubled the learning rate during training, and recomputed width and transfer, with results in Fig. 24. The two qualitative trends persist in both cases:

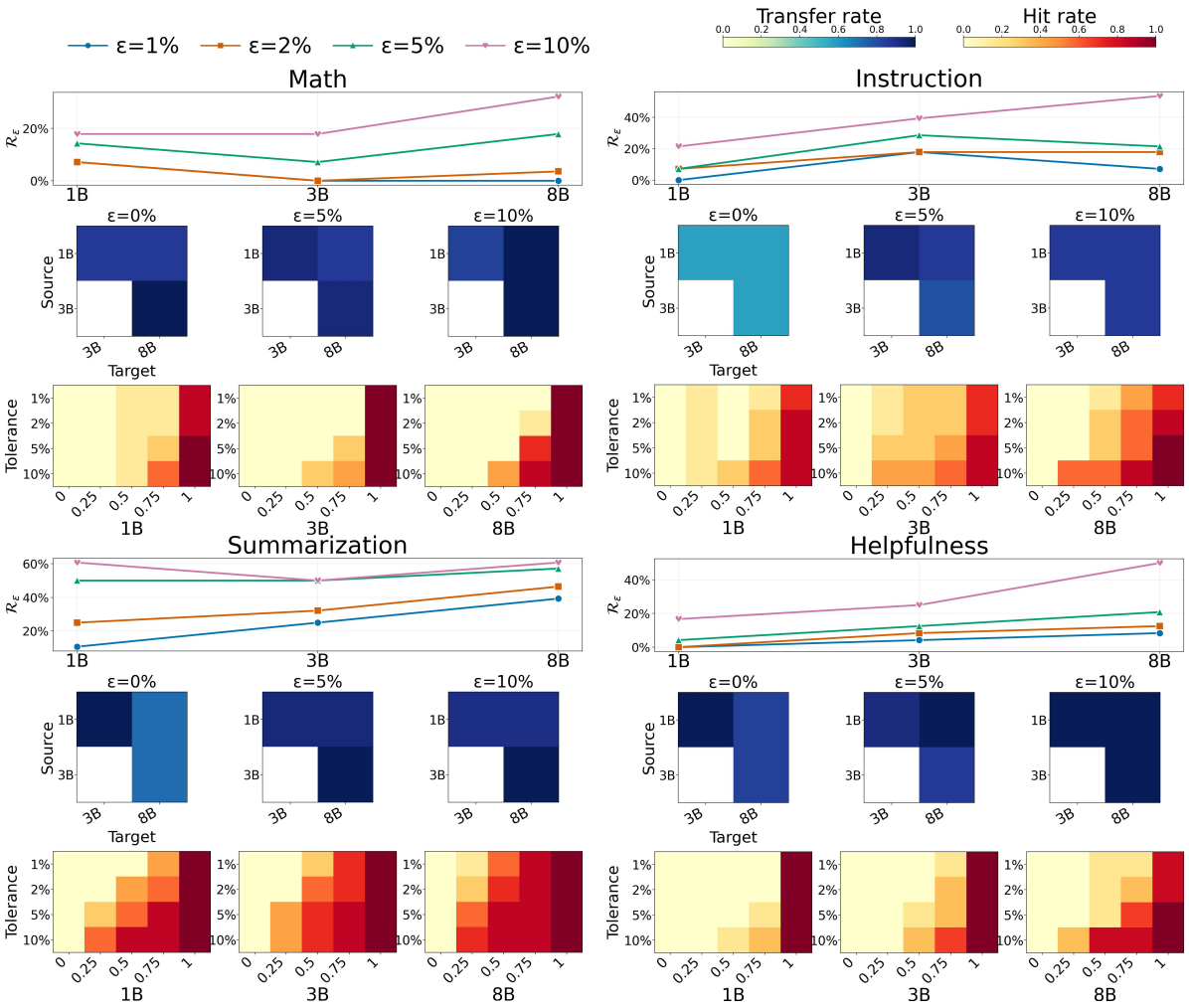


Figure 23: Main analysis for Llama family with SFT–DPO training stages with $B \geq 1k$.

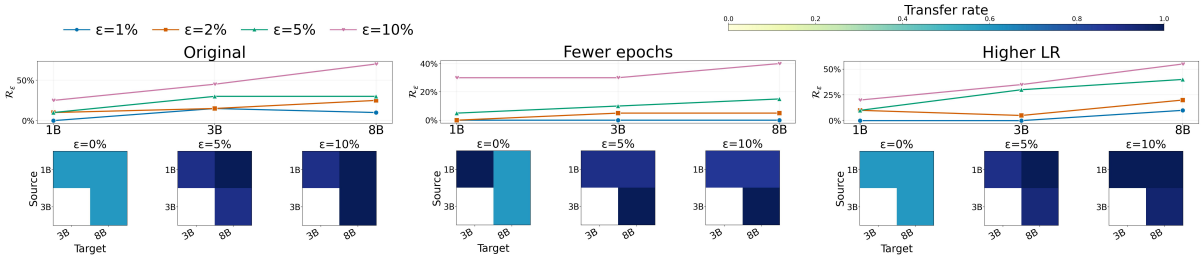


Figure 24: Changing training configurations on the instruction-following task for the Llama family.

width is non-decreasing in scale at every tolerance and strictly increasing at $\varepsilon = 5\%$ and 10% (it is identically 0.00 at $\varepsilon = 1\%$ with fewer epochs), and near-optimal transfer exceeds exact-optimum transfer.

E.6 Ablation Study 6: Absolute-Tolerance Ratio

A relative tolerance can admit more ratios simply because larger models have higher peak performance. To test this directly, we re-ran our analysis under an *absolute* tolerance (a fixed performance margin anchored to the smallest model, as an arbitrary margin range does not make sense). Specifically: Let $N_{\min} = \min \mathcal{N}$ denote the smallest model size (in our setting, 1B for Llama, 1.5B for Qwen2.5 (denoted by 2B)). We define an absolute performance tolerance, anchored to the smallest model, as

$$\Delta_\varepsilon(B) := \varepsilon \max_{r' \in \mathcal{G}} \mathcal{P}(N_{\min}, r', B).$$

We then estimate the near-optimal region by

$$\widehat{\mathcal{R}}_{\varepsilon}^{\mathrm{abs}}(N,B) = \Big\{ r \in \mathcal{G} : \mathcal{P}(N,r,B) \geq \max_{r' \in \mathcal{G}} \mathcal{P}(N,r',B) - \Delta_{\varepsilon}(B) \Big\}.$$

The results comparing $\widehat{\mathcal{R}}_{\varepsilon}(N,B)$ (top row, reports Range Width) and $\widehat{\mathcal{R}}_{\varepsilon}^{\mathrm{abs}}(N,B)$ (bottom row, reports Range Width) are in Fig. 25.

1. Width: the widening is not merely a relative-tolerance artifact. An absolute tolerance largely preserves the scaling behavior seen under a relative tolerance for the Qwen family. For the Llama family, the widening is dampened (e.g., Llama-Inst widens continuously under a 10% relative tolerance, but increases and then decreases under an absolute tolerance). The starkest contrast is Llama-Math, where the absolute slope is negative. This contrast between Qwen-math and Llama-math's absolute tolerance behavior is due to Llama's small model performing far below its 8B family, so a fixed absolute margin spans a very different share of the performance range at each scale. This is precisely why *an absolute tolerance is task- and model-dependent* (Sec. 3.3), and why we adopt the practitioner-relevant relative criterion: it answers "how much allocation freedom retains $X\%$ of *this* model's peak?" without anchoring to a family whose behavior differs sharply across scale.

2. Transfer: Robust under both tolerances. Near-optimal regions transfer far better than the exact optimum in 3 out of 4 settings. This addresses the possibility that transfer is driven by mechanical overlap: under an absolute tolerance, the target region is generally *narrower* than under a relative one (e.g., Qwen-Inst 7B), yet transfer stays high. Mechanical overlap from a wider target would predict the opposite. The exception is Llama-Math, where the 8B absolute region remains a single ratio (width 0.00) even as the absolute tolerance increases because the fixed performance margin anchored to the smallest model is way too small for 8B. Here, the exact optimum transfers unusually well precisely because that single ratio is stable across scale, so "near-optimal vs. exact" is degenerate in this setting rather than a genuine reversal of our claim.

To summarize, peak-performance scaling does marginally contribute to the widening effect (visible in the Llama-math results), but our core practical claim, transferability, remains generally robust under both tolerance definitions. Where absolute transfer drops (Llama-Math), it reflects the task- and model-dependent nature of an absolute margin, which is itself the reason a relative tolerance is a more appropriate metric.

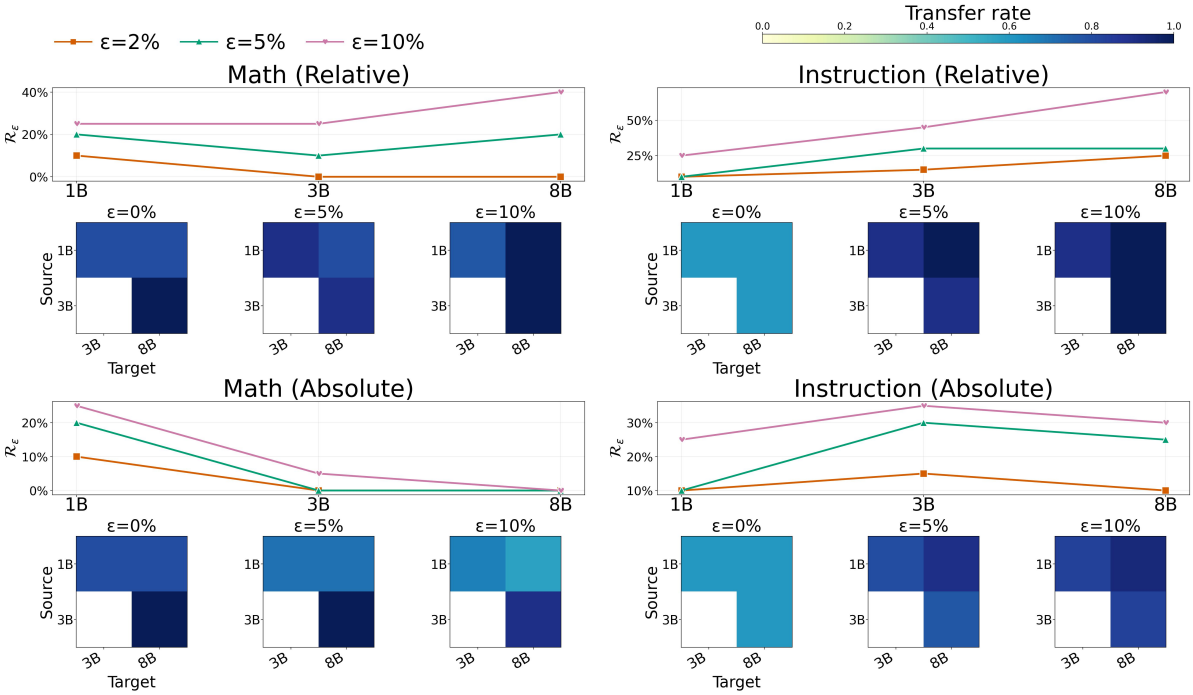
E.7 Ablation Study 7: Proxy Versus Fixed Ratio Comparison

We observe that a 10% tolerance makes the region wide at scale, so intuitively, a centered default often lands within it. However, a fixed $r = 0.5$ breaks in two common cases: (i) a stricter tolerance(e.g., 5%, retaining 95% of peak); and (ii) an off-center region, where the optimum does not sit near 0.5. The proxy handles both, because it transfers the region's *location*, which we show is reliable across scales. We also ran an experiment for a small proxy 5-point search vs. a fixed $r = 0.5$, and report reliability and cost below. We measure how often each recommendation lands within the target model's near-optimal region (proxy: ratios transferred from the 1B/1.5B models), averaged over budgets and target sizes (3B–8B). As shown in Tab. 6, at both tolerances, the proxy generally outperforms or is comparable to a fixed $r = 0.5$, but this is more evident at the smaller tolerance (0.90–0.95 vs. 0.20–0.60). Crucially, whether $r = 0.5$ suffices depends on the task, model family, and the statistics we cannot know in advance.

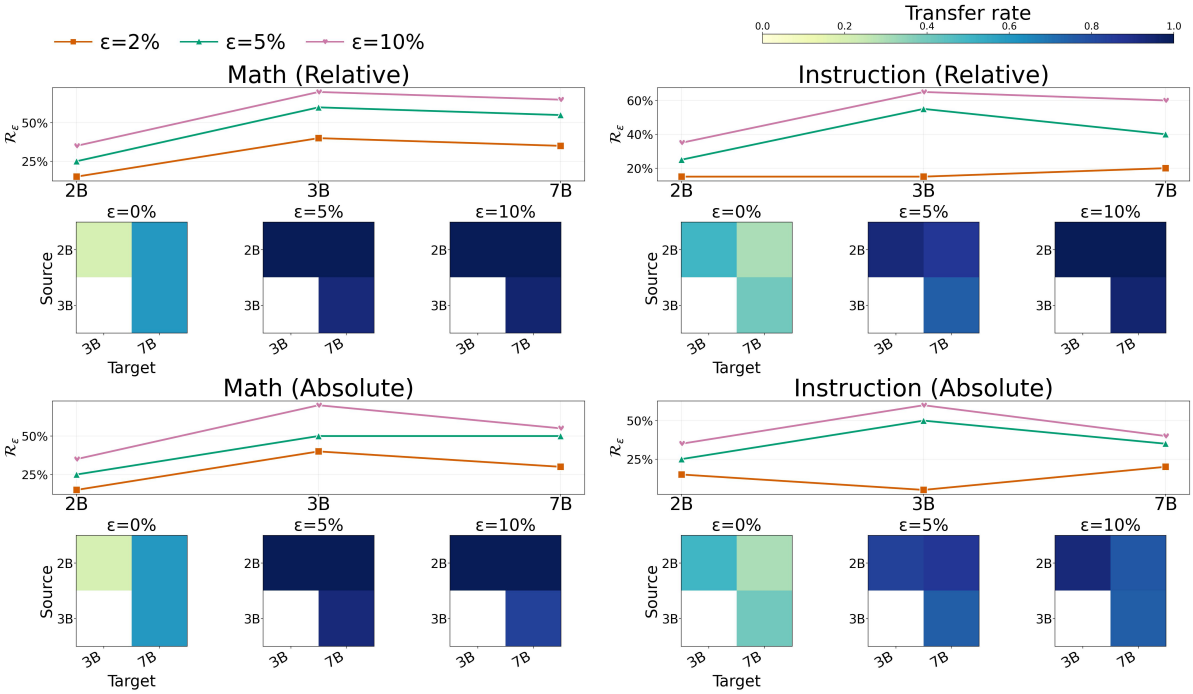
Additionally, using our measured runtimes and CoreWeave L40 pricing ($1.25/GPU-hr), a 5-point ratio search on the 1B proxy (e.g., $10k$/run) and a single 8B run at one ratio ($10k$) take the same $\sim$ 100 GPU-minutes $\sim$ $2.1. A blind guess of $r = 0.5$ costs the same 100 minutes but, per Tab. 6, carries real risk: it misses the 5% near-optimal region up to 80% of the time (e.g., Llama HelpSteer). Spending the extra 100 minutes up front on the 1B proxy serves as an efficient 'insurance policy,' providing a $> 90\%$ guarantee of hitting the target's 5% near-optimal region on the first try. Moreover, the proxy approach (proxy + 8B run $\sim$ 200 GPU-minutes) is $2.5\times$ cheaper than a full 5-point sweep on the 8B target ($\sim$ 500 GPU-minutes). The proxy method thus offers a good trade-off between computational cost and guaranteed performance.

F Additional Results for Cost Asymmetry Setting

We extend the heterogeneous-cost analysis from Sec. 3.4 to the Summarization task. The experimental setup follows the same experimental protocol.



(a) Llama Model Family.



(b) Qwen 2.5 Model Family.

Figure 25: Main analysis results under a relative tolerance (main paper) versus an absolute tolerance.

Fig. 26 reports the results. The qualitative scaling behavior matches the math task results in the main text: across cost settings, larger tolerance thresholds continue to produce broader $\mathcal{R}_\varepsilon$ and stronger cross-scale transfer T_ε; as ρ increases, $\mathcal{R}_\varepsilon$ generally widens at the same tolerance, indicating increased allocation flexibility for practitioners when SFT becomes relatively expensive. These results confirm that the cost-asymmetry findings in Sec. 3.4 are not specific to mathematical reasoning and can generalize across our task suite.

Additionally, we expand the analysis to a GPU-time budget case study. If a practitioner defines the budget purely by GPU compute time (proportional

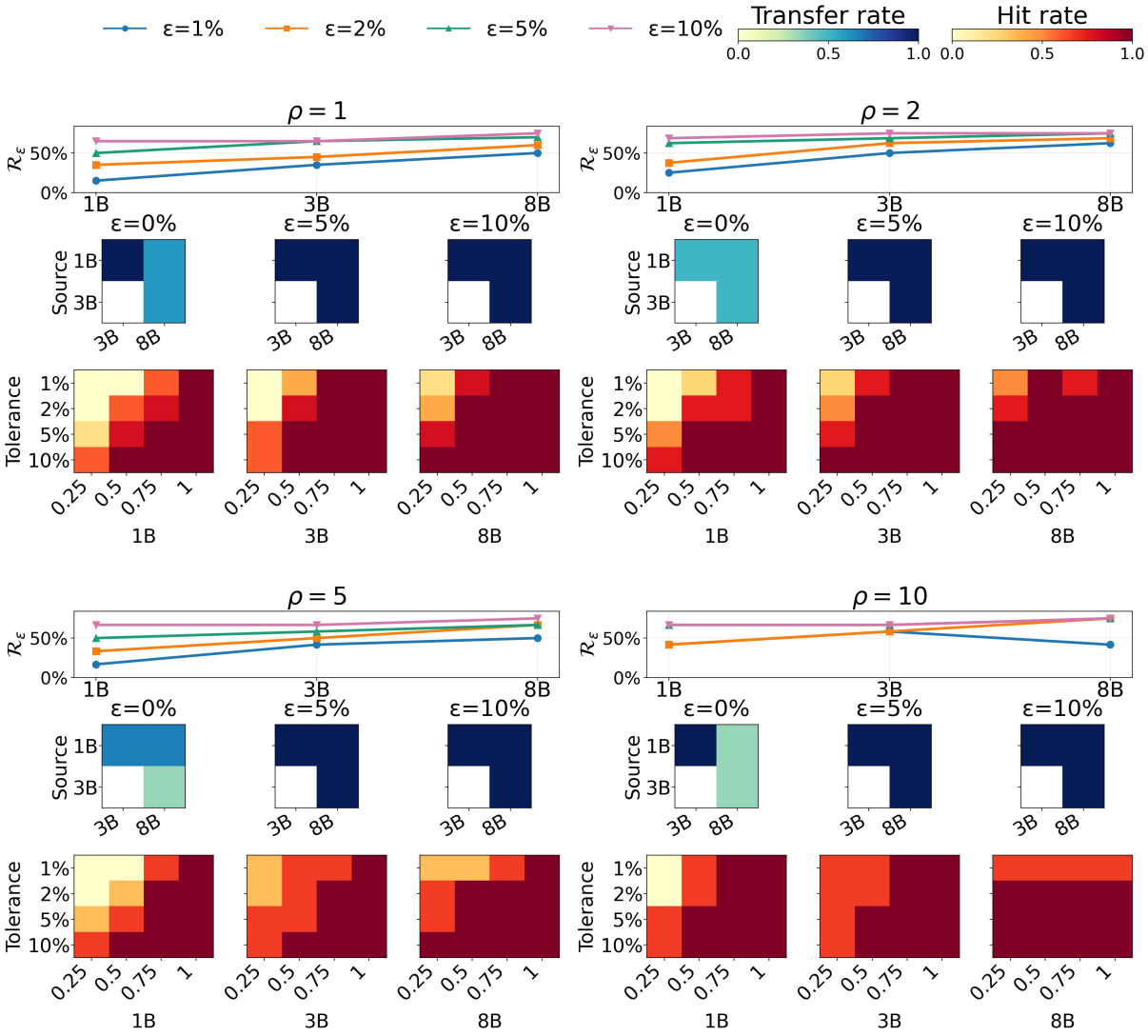


Figure 26: Heterogeneous annotation costs settings for the Summarization task.

Table 6: Transferability comparison: small model proxy vs. fixed $r = 0.5$. We select these settings to vary one axis at a time: instruction-following on both Llama and Qwen 2.5 (same task, different family) and HelpSteer vs. instruction-following on Llama (same family, different task), so the comparison covers both task- and family-level.

Task	Tol.	Proxy	Fixed r=0.5
Llama HelpSteer	5%	0.95	0.20
Llama Inst	5%	0.95	0.40
Qwen 2.5 Inst	5%	0.90	0.60
Llama HelpSteer	10%	1.00	0.70
Llama Inst	10%	0.95	0.70
Qwen 2.5 Inst	10%	1.00	1.00

to training FLOPs (Hoffmann et al., 2022)), our framework still applies. Since DPO takes $\sim$ 2–3$\times$ the GPU time of SFT, a compute-time budget is an instance of our cost-asymmetry analysis(Sec. 3.4) with $\rho = c_{SFT}/c_{DPO} \approx 0.5(1:2)$, the regime where the RL stage is the more expensive one, extending the $\rho \geq 1$ range of the main paper. We ran this setting on math (Llama). At $\rho = 0.5$, the near-optimal region reduces to essentially a single ratio at small scale (Tab. 7, width 0), and that ratio is identical across model sizes. Hence, the proxy's recommendation is always valid on the larger target, giving consistently good transfer (Tab. 8). The region only begins to widen at 8B. Transfer is thus trivially perfect in this concentrated regime. Our practical claim still holds under a compute-time budget.

G Additional Related Work on Mixture Optimization and Scaling Law

Data mixture optimization. A parallel literature optimizes the training data mixture *within* a single

Table 7: Near-optimal region width (Math, Llama; $\rho = 0.5$) when budget is GPU-hour. A width of 0 indicates a single near-optimal ratio.

$\mathcal{R}_\varepsilon$	$\varepsilon = 2\%$	$\varepsilon = 5\%$	$\varepsilon = 10\%$
1B	0.00	0.00	0.00
3B	0.00	0.00	0.06
8B	0.00	0.13	0.25

Table 8: Cross-scale transfer (Math, Llama; $\rho = 0.5$) when budget is GPU-hour.

Transfer	$\varepsilon = 0\%$	$\varepsilon = 5\%$	$\varepsilon = 10\%$
1B $\rightarrow$ 3B	1.00	1.00	1.00
1B $\rightarrow$ 8B	1.00	1.00	1.00
3B $\rightarrow$ 8B	1.00	1.00	1.00

training stage. Pretraining-stage methods include scaling-law-based mixture extrapolation (Jiang et al., 2025; Liu et al., 2025; Ye et al., 2025; Diao et al., 2025; Shukor et al., 2025); group-DRO bilevel domain reweighting (Xie et al., 2023; Fan et al., 2024); online refinement of mixture weights via multi-fidelity Bayesian optimization (Ouyang et al., 2025; Yen et al., 2025), intrinsic-feature signals (Zhu et al., 2025b), influence-function attribution (Chang et al., 2025), and gradient statistics (Li et al., 2025b). Post-training mixture methods study data composition or reweighting within individual post-training stages, including SFT data-mixture optimization (Li et al., 2025a), dynamic instruction-tuning mixtures (Zhu et al., 2025a), Tülu recipes (Ivison et al., 2023; Lambert et al., 2025), DaMo (Shi et al., 2026), multi-task preference optimization with AutoMixAlign (Corrado et al., 2025), process-reward-model reweighting with DreamPRM (Cao et al., 2025), RLVR data-mixture optimization with MoDoMoDo (Liang et al., 2025), and recent gradient-based fast mixture solvers (Tan et al., 2026). Béthune et al. (2025) is the closest in setting, studying the cross-stages pretraining-finetuning forgetting trade-off via controlled pretraining injection. These methods assume mixture optimization in a single stage, in which weights can be refined as training proceeds, and the same loss surface is shared across mixtures. The SFT–RL allocation problem is fundamentally cross-stage: reallocation requires retraining downstream stages from a different SFT checkpoint; the loss surfaces of SFT and RL differ functionally, ruling out direct application of adaptive or extrapolated point performance along the budget axis. Our work studies a complementary problem: identifying transferable allocation *regions* across model scales at matched budgets.

Scaling laws and cross-scale extrapolation. Pretraining scaling laws predict loss from small proxies (Kaplan et al., 2020; Hoffmann et al., 2022; Gadre et al., 2025), with extensions to loss-to-loss prediction across datasets (Brandfonbrener et al., 2025) and to downstream task performance (Ruan et al., 2024; Chen et al., 2025). A related line of work studies whether tuning decisions transfer across scales: Yang et al. (2021) shows that under the maximal-update parameterization, optimal hyperparameters can be transferred zero-shot from small to large models in the pretraining setting. The post-training, cross-stage allocation problem that our study focuses on is transferability, as evidenced by empirical results rather than theoretical analysis. In post-training, however, scaling behavior is documented to be more fragile: scaling laws can exhibit broken or non-monotonic transitions (Caballero et al., 2023), inverse scaling (McKenzie et al., 2023), and metric-induced discontinuities (Schaeffer et al., 2025); reward overoptimization causes performance reversal at high optimization pressure (Gao et al., 2023); and post-training algorithm rankings can flip across scales (Li, 2026). Recent empirical work derives scaling laws specifically for direct alignment (Rafailov et al., 2024) and for RLHF data scaling (Shen et al., 2025), but documents bottlenecks that complicate naive extrapolation from small proxies. Our findings sharpen this picture: exact-optimal-ratio transfer can be inconsistent across model scales, but transferring *near-optimal allocation regions* is substantially more reliable, providing a robust mechanism for guiding allocation from small proxy to large target.

H Additional Discussion and Clarifications

Q.1: Why do all experiments use LoRA instead of full-parameter fine-tuning?

A: Our analysis sweeps over the Cartesian product of model sizes, allocation ratios, budgets, tasks, algorithms, and model families; on the order of hundreds or even thousands of post-training runs. Full-parameter fine-tuning at this volume can be too expensive, a constraint shared by other recent large-scale post-training studies that use LoRA for the

same reason (Raghavendra et al., 2025; Li, 2026). LoRA has been shown to achieve performance comparable to full fine-tuning across instruction tuning and preference optimization in many settings (Hu et al., 2022; Dettmers et al., 2023). We treat LoRA as a reasonable approximation to full fine-tuning within our experimental scope.

Q.2: How robust are the region width and transferability statistics to training stochasticity?
A: Our analysis sweeps post-training runs at the Cartesian product of model sizes, allocation ratios, budgets, tasks, algorithms, and model families: on the order of hundreds or thousands of runs. Multi-seed evaluation across the full sweep would multiply this cost by the number of seeds and would require much more computing resources. To assess whether the qualitative trends we report are robust to seed variance, we conduct a 3-seed validation on math (GSM8K) and summarization for the Llama family (Ablation Study 3 in App. E.3); seed variation perturbs both the random subset of training samples drawn from the source pool and the training order within each stage. The seed-averaged performance curves, the near-optimal region width versus tolerance, and the cross-scale transfer hit rates at $\varepsilon = 10\%$ all preserve their qualitative trends, confirming that the single-seed findings reported are consistent across seeds.

Q.3: Could the broad near-optimal region be an artifact of the coarse 5-point allocation grid?
A: We address this concern in four complementary ways. *First*, our 5-point grid $\mathcal{G} = \{0.00, 0.25, 0.50, 0.75, 1.00\}$ follows the existing protocol of Raghavendra et al. (2025), who study the same SFT-vs-preference-finetuning trade-off and report broad allocation trends consistent with our findings. *Second*, we directly validate that the qualitative width and transferability trends are preserved under a denser 9-point grid (adding intermediate ratios $\{0.125, 0.375, 0.625, 0.875\}$) in Ablation Study 1 (App. E.1); both grids yield the same qualitative scaling and transferability behavior. *Third*, the per-ratio hit-rate heatmaps in our main scaling figures (Row 3 of Fig. 2 and Fig. 3) directly visualize the contiguity of the near-optimal region: ratios admitted into $\mathcal{R}_\varepsilon$ at a given tolerance are generally neighboring on the grid, indicating genuine smoothness of $\mathcal{P}(N, \cdot, B)$ rather than grid-induced artifacts. *Fourth*, the broad-region phenomenon is observed consistently across three model families (Llama 3, Qwen 2.5, and Qwen 3), four tasks, both off-policy (DPO) and on-policy (GRPO) post-training, and across multiple seeds (Ablation Study 3, App. E.3); a grid-induced artifact would not be expected to manifest uniformly across such diverse experimental conditions.

Q.4: Why does the main analysis exclude the low-budget regime ($B < 5$k)?
A: The exclusion is methodologically plausible. *First*, performance curves in the $B < 5$k regime can be noisy, with the relative ordering of allocation ratios fluctuating across budgets (per-budget curves in App. B): most prominently for instruction-following on Llama and math on Qwen. Including this regime would conflate the cross-N widening signal with training-instability noise. Similar model-scale transfer works, such as scaling law, routinely exclude warmup-phase (Kaplan et al., 2020; Hoffmann et al., 2022; Caballero et al., 2023), and instruction-tuning specifically requires a minimum data threshold (Zhou et al., 2023a, ∼2K examples,). *Second*, we explicitly extend the analysis to lower budgets in Ablation Study 4 (App. E.4); the qualitative trends persist, with expected attenuation at the noise floor (most visibly for IFEval). At these low budgets, the optimal allocation also shifts toward SFT-heavy ratios (Raghavendra et al., 2025), and direct exhaustive sweeping itself can be cheap. The proxy-transfer story that our framework provides is less load-bearing precisely where the attenuation appears.

Q.5: Why use a relative percentage tolerance rather than an absolute tolerance?
A: The relative tolerance, $\mathcal{P}(N, r, B) \geq (1 - \varepsilon)\mathcal{P}^*(N, B)$, is the practitioner-relevant criterion: deployers can ask "how much allocation flexibility do I have while retaining $X\%$ of this model's peak?" An absolute tolerance, $\mathcal{P} \geq \mathcal{P}^* - \delta$, would in principle isolate allocation-sensitivity changes from peak-performance scaling, but defining a meaningful scale-invariant δ is non-trivial: anchoring δ to the smallest model collapses to examining relative settings when peak performance is close across scales, and becomes task- and model-dependent (and sometimes ill-defined when the smallest model's performance is at or near trivial baselines). We therefore adopt the relative framing and discuss its consequences. However, the observed widening with scale can be partly driven by peak-performance scaling rather than by changes in allocation sensitivity; a more detailed analysis is provided in App. E.6.