THE SHAPE OF THE DPO HYPERPARAMETER LANDSCAPE: WHAT TO MEASURE, HOW IT MOVES WITH SCALE, AND WHEN NOT TO TRANSFER IT

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct preference optimisation (DPO) is widely used for language model post-training, but its performance depends strongly on the choice of its hyperparameters, such as the learning rate and the reference-policy regularisation parameter β. To better understand the DPO hyperparameter landscape, we conduct extensive experiments using a dense hyperparameter grid across different model scales from the same family. This allows us to analyse how the optimal DPO hyperparameters evolve with model size and the number of update steps, with intermediate checkpoints providing an inexpensive signal to accelerate the hyperparameter optimisation process. Furthermore, by evaluating our models on both static and dynamic benchmarks, we identify which benchmarks provide the strongest signal for hyperparameter selection. Finally, we investigate how different state-of-the-art hyperparameter optimisation strategies perform in this setting. Our experiments show that the optimal learning rate varies with model size and that hyperparameters are not transferable across training datasets. We also find that validation loss is a poor proxy for hyperparameter optimisation, as hyperparameters that minimize validation loss only weakly correlate with optimal hyperparameters for downstream benchmarks. Therefore, DPO hyperparameters need to be tuned specifically for the target model, dataset, and evaluation objective. We also show that short candidate training runs can substantially reduce the cost of this selection. We release all our training configurations and evaluation results to support reproducible analysis of selection quality and training cost.

1 INTRODUCTION

Large language models (LLMs) have rapidly advanced in both capability and scale, but pretrained models are not necessarily well-suited for interaction with users: they may fail to reliably follow instructions, answer questions, or respond appropriately to user intent. Post-training is therefore a critical step in transforming a pretrained model into a useful assistant (Ouyang et al., 2022). Common post-training recipes consist of supervised fine-tuning (SFT), which teaches a model to follow instructions using demonstration data, followed by preference optimisation that adapts model behaviour toward preferred responses.

Direct preference optimisation (DPO) (Rafailov et al., 2023) optimises directly from preference pairs relative to a fixed SFT reference policy, without a separate reward-model training stage. Its learning rate controls the size of the update, while β controls the strength of the reference-adjusted preference signal in the DPO objective and therefore the objective's trade-off between fitting preference pairs and remaining close to the reference policy. Both hyperparameters may therefore have effects that depend on the reference checkpoint and preference data. Prior work has already shown sensitivity to β and the characteristics of preference pairs (Wu et al., 2024). This raises the question: how reliably can a DPO configuration selected in one setting be applied to another?

Post-training recipes commonly report configurations that perform well for a particular model, dataset, and reference checkpoint, often using a fixed suite of static capability benchmarks to guide those choices (Tunstall et al., 2024; Ivison et al., 2023; Lambert et al., 2025; Team Olmo, 2025; Allal et al., 2025). These configurations are then generally reused across related models or experiments,

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

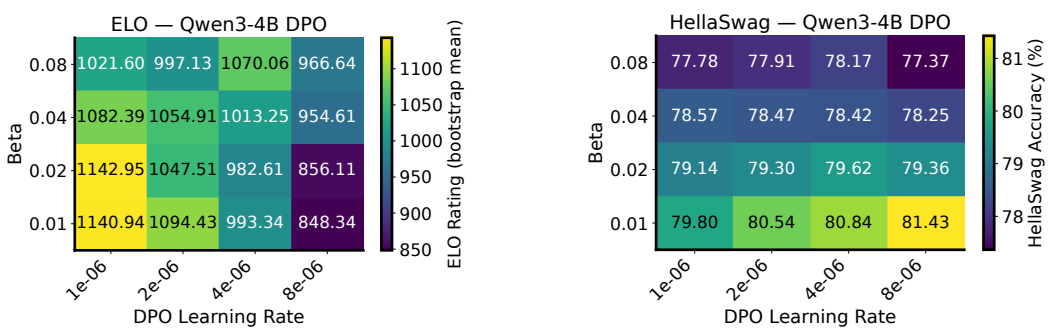

Figure 1: **DPO performance is sensitive to the jointly tuned learning rate and β.** Each cell is one configuration in the displayed 4×4 Qwen3-4B subgrid, scored by Elo (**left**) and HellaSwag (**right**); higher is better. Elo spans roughly 295 points, and the two evaluations select different configurations.

implicitly assuming both that the configuration transfers and that the static evaluation signal used to select it remains informative. Whether either assumption holds across model scale has received substantially less systematic investigation. Getting this wrong is not a minor efficiency loss. In our primary sweep, the gap between the best and the worst configuration within a single model size reaches roughly 295 Elo points at a 4B parameter scale (Figure 1). Under the fitted Bradley–Terry model used to construct these scores, this corresponds to an estimated 84.5% probability that the best configuration wins a pairwise comparison with the worst. By comparison, the DPO improvement over its SFT checkpoint is 176 Elo points, corresponding to a 73.3% win probability. The worst configurations in the grid also land *below* the SFT checkpoint. A poor configuration can therefore erase the entire benefit of the stage and leave a model worse than its SFT checkpoint.

We study DPO selection across five Qwen3 (Qwen Team, 2025) sizes (0.6B–14B), three complementary Llama models (Grattafiori et al., 2024; Meta AI, 2024), eleven downstream benchmarks, and two LLM judges. Fully logged training trajectories allow us to analyse configuration transfer and compare single- and multi-fidelity hyperparameter optimisation (HPO) methods without additional training. We release the resulting checkpoint-level benchmark, representing approximately 11,849 H100 GPU-hours of completed DPO training and evaluation (including the complementary experiments), so future work can reuse these results without having to repeat this compute. Appendix C.1.1 provides the breakdown.

Our contributions are as follows:

- **Evaluation signals for DPO tuning.** We show that validation loss is an unreliable selection signal and that chat benchmarks produce more consistent configuration rankings than static benchmarks, with limited sensitivity to the choice of the judge model used.
- **Transfer across scale and training data.** We show that DPO configurations do not reliably transfer across model sizes or training preference datasets. Preferred learning rates exhibit an overall decrease with Qwen3 model scale, but configuration rankings can vary sharply even between neighbouring models.
- **Compute-efficient hyperparameter selection.** Partial model training runs can provide useful ranking information within roughly 20–30% of training. We also show that under sufficient budgets, adaptive HPO methods can outperform random search.
- **An offline benchmark for DPO hyperparameter selection.** We release checkpoint-level Qwen3 and Llama trajectories with training costs and scores on eleven benchmarks, along with code to reconstruct the surrogate environments and HPO experiments. This enables reproducible research on selection signals, early-selection policies, transfer, and compute allocation without repeating these expensive DPO runs.

The remainder of the paper is organized as follows: Section 2 situates our study within work on DPO, hyperparameter transfer, and evaluation reliability. Section 3 describes the models, data, training grids, evaluation suite, downstream aggregation objectives, and offline HPO simulation framework.

Section 4 presents the empirical results, followed by practical recommendations, limitations, and conclusions. The appendix provides complete experimental settings, supplementary analyses, and detailed HPO simulation results.

2 Related Work

DPO and hyperparameter sensitivity. DPO optimises preferences relative to a fixed reference policy without training a separate reward model (Rafailov et al., 2023). Alternative objectives such as ORPO and SimPO change the treatment of the reference policy, preference margin, or regularisation (Hong et al., 2024; Meng et al., 2024). We hold the DPO objective fixed to isolate hyperparameter transfer within one method and leave an extension of this analysis with alternate preference-optimisation objectives to future work. Prior work shows that DPO is sensitive to training data (Ivison et al., 2024) and the joint choice of learning rate and β (Wu et al., 2024; Meng et al., 2024), while post-training recipes tune these values for their particular models and datasets (Tunstall et al., 2024; Ivison et al., 2023; Lambert et al., 2025). We quantify how a jointly selected $(\text{learning rate}, \beta)$ configuration transfers across model scales and preference data, and whether the target model reveals a suitable configuration before training completes.

Hyperparameter transfer and scaling. Scaling laws characterize how training behaviour changes with model and data size (Kaplan et al., 2020; Hoffmann et al., 2022), while μTransfer and related work study when hyperparameters tuned on smaller models predict larger-model behaviour (Yang et al., 2021; Li et al., 2025a). DPO presents a different setting because it follows pretraining, depends on an SFT reference policy, and introduces preference-specific hyperparameters. We test whether an analogous learning-rate trend appears in DPO and assess whether configurations selected at one scale remain effective at another.

Evaluation signals and benchmark reliability. Post-training studies commonly use both fixed capability benchmarks and LLM-as-judge evaluations such as MT-Bench (Zheng et al., 2023), AlpacaEval (Dubois et al., 2024), and Arena-Hard (Li et al., 2025b). LLM-as-judge evaluations have documented sensitivities including position bias (Wang et al., 2024), length bias (Dubois et al., 2024), and model self-preference (Panickssery et al., 2024). Relatedly, FineTasks selects multilingual evaluation tasks using properties such as non-random performance, low variability, and stable model rankings across training checkpoints (Kydlíček et al., 2024). Our setting is closely related but addresses a distinct post-training question: *whether a metric provides a reliable ranking signal for hyperparameter selection*. We examine whether validation loss, static capability benchmarks, and chat-based evaluations produce consistent rankings over DPO configurations, and whether rankings observed at intermediate checkpoints predict the final ordering of configurations. Evaluation is therefore treated not only as a reporting mechanism but as part of the HPO problem itself.

3 Experimental Setup

Models. We select Qwen3 as the primary model family because it provides competitive, text-only base checkpoints at multiple practical scales (Qwen Team, 2025). Our study uses the 0.6B, 1.7B, 4B, 8B, and 14B models, which span a $23\times$ parameter range within one family. This allows us to study how DPO hyperparameter selection changes with size while holding the model family and modality fixed. Complementary experiments on Llama models are reported in Appendix D.2.

Data. We first perform SFT on Dolci-Instruct-SFT and then train with DPO on Dolci-Instruct-DPO (Team Olmo, 2025). To isolate the effects of training data, we also repeat the shared DPO grid on UltraFeedback Binarized (Cui et al., 2024) dataset for the same Qwen3 models. For this ablation, we retain the SFT checkpoint and evaluation suite while replacing the DPO training dataset.

Training Protocol. We use an SFT learning rate of 3×10^{-5} for all primary runs, while comparing DPO configurations across model sizes. Appendix C.2 reports a five-point SFT learning rate sweep showing that this learning rate choice lies in a stable regime. Rankings vary across the five sizes and evaluation aggregates, and the highest learning rate tested consistently degrades the downstream performance. We leave joint optimisation of the SFT and DPO hyperparameters to future work.

DPO begins from the corresponding SFT checkpoint. Its search space contains a fully observed 3×3 core shared by every model: learning rates $\in \{1 \times 10^{-6}, 2 \times 10^{-6}, 4 \times 10^{-6}\}$ and β values $\in \{0.01, 0.02, 0.04\}$. We extend this core where needed to cover each size's promising learning-rate range. The shared core supports like-for-like transfer and the size-specific extensions inform the local optimum. We save intermediate checkpoints and evaluate them alongside the final checkpoints. All other training settings are held fixed within each sweep. Per-size search spaces, checkpoint steps, training settings, and distributed training setups are reported in Appendix C.3. All post-training runs use `TRL` (von Werra et al., 2022).

Evaluation. We evaluate eleven benchmarks in two groups. The chat group comprises Elo (Chiang et al., 2024), Arena-Hard (Li et al., 2025b), MT-Bench (Zheng et al., 2023), and length-controlled AlpacaEval 2.0 (Dubois et al., 2024). These evaluations measure instruction following and response quality through LLM judgments. We score the chat responses separately with Qwen3-Next-80B-A3B-Instruct and Gemma 4 31B-it (Qwen Team, 2025; Gemma Team, 2026) to assess their sensitivity to judge choice (Wang et al., 2024; Panickssery et al., 2024). The main paper results use the Qwen model as the judge, unless specified otherwise. The static group comprises ARC-C (Clark et al., 2018), GPQA (Rein et al., 2024), GSM8K (Cobbe et al., 2021), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), TruthfulQA (Lin et al., 2022), and IFEval (Zhou et al., 2023). We select these benchmarks to cover distinct capabilities commonly examined in language-model evaluation: scientific and mathematical reasoning, commonsense and physical reasoning, truthfulness, and instruction and format compliance. This diversity allows us to test whether a configuration that performs well on one fixed capability transfers to other capabilities, and to monitor capability-specific regressions during DPO. We also record validation loss as a candidate selection signal. The static benchmarks use `lm-evaluation-harness` (Gao et al., 2023). The chat benchmarks use their corresponding benchmark implementations, including `JudgeArena` for MT-Bench and Elo ratings (Lushtaku et al., 2026), with candidate and judge models served through `vLLM` (Kwon et al., 2023). Complete benchmark settings are reported in Appendix C.4.

Aggregation and Selection Criteria. For configuration i and benchmark b, let $x_{i,b}$ denote the observed benchmark score for a specific configuration. Because the scales across benchmarks differ, we standardize scores as $z_{i,b} = (x_{i,b} - \mu_b)/\sigma_b$, where μ_b and σ_b are the mean and population standard deviation of benchmark b across the relevant configuration pool. For an aggregate G of benchmarks, we average these standardized scores as $Z_{i,G} = \frac{1}{|G|} \sum_{b \in G} z_{i,b}$. We use G for the aggregation of chat benchmarks (Z_{Dynamic}), static benchmarks (Z_{Static}), or all benchmarks (Z_{All}). Trajectory plots use final-grid statistics as a fixed reference, whereas checkpoint-selection analyses recompute them at each step. For configuration i and aggregate G, we define normalised regret as $\text{Regret}_{i,G} = \frac{\max_{j \in \mathcal{C}} Z_{j,G} - Z_{i,G}}{\max_{j \in \mathcal{C}} Z_{j,G} - \min_{j \in \mathcal{C}} Z_{j,G}}$ where $\mathcal{C}$ is the set of jointly scored configurations used for the comparison: normally the full grid at one model size, and the shared 3×3 core at the target size for cross-scale transfer. Thus, regret is zero for the best configuration in $\mathcal{C}$ and one for the worst.

Offline HPO Simulation. Hyperparameter optimisation (HPO) (Franceschi et al., 2025) uses the outcomes and costs of earlier trials to decide which training configurations to evaluate next. Multi-fidelity HPO additionally uses partial training runs as cheaper, lower-fidelity signals when allocating further compute. Training progress is the fidelity variable. Each method optimises Z_{Dynamic} under matched budgets, and we evaluate its recommended configuration at true final fidelity. At each Qwen3 size, we fit four surrogate blackboxes separately: a Gaussian process (Williams & Rasmussen, 2006) and one-, three-, and five-nearest-neighbour regressors (Hastie et al., 2009). This offline-HPO setup enables repeated comparisons under matched budgets and random seeds, although its conclusions are conditional on the fitted interpolation models. We compare three single-fidelity methods, random search (Bergstra & Bengio, 2012), TPE (Bergstra et al., 2011), and CQR (Salinas et al., 2023), which evaluate every proposed configuration to completion, with the multi-fidelity methods ASHA (Li et al., 2020a) and BOHB (Falkner et al., 2018), which allocate compute using intermediate results from step 400 onward. Each setting uses two workers and 25 seeds under an $8\times$ wall-clock budget and a restrictive one-unit budget, corresponding to approximately two full-run equivalents of aggregate compute across both workers. At each cutoff, we evaluate the recommended configuration at final fidelity; before any trial completes, we compare candidates at their highest shared fidelity. The simulations use `syne-tune` (Salinas et al., 2022). Appendices A and C.5 give additional related work and full HPO settings.

4 RESULTS

4.1 IDENTIFYING THE METRIC TO TUNE ON

Validation loss is an unreliable selection criterion. Validation loss is an insufficient proxy signal to select hyperparameters as it poorly correlates with downstream performance (see Figure 2). Selecting the configuration with minimum validation loss yields mean normalised regret of 0.50, 0.62, and 0.53 under Z_{Static}, Z_{Dynamic}, and Z_{All}, respectively. At 0.6B, it selects the worst configuration under Z_{Dynamic}. Pooled across sizes, rank correlations are only 0.10, 0.18, and 0.14, and sometimes change sign across scales. Because DPO loss depends on β, we also rank learning rates separately within each fixed β. Agreement with Z_{All} still varies across Qwen3 sizes ($\rho = -0.67$ to 0.67; Appendix D.1.2). The underlying benchmark-level correlation matrices appear in Appendix D.1.1.

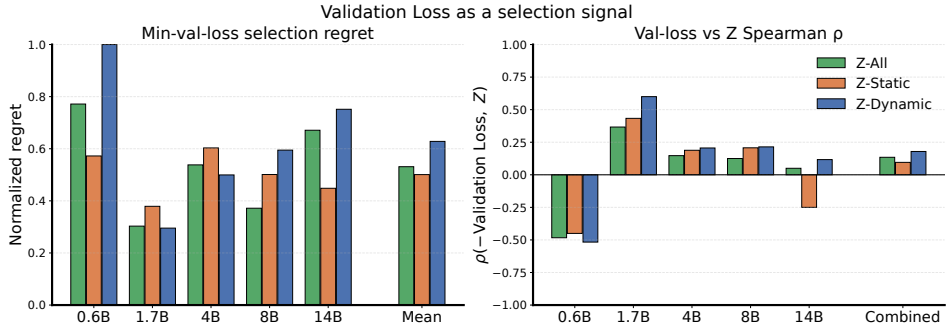

Figure 2: **Validation loss is an unreliable DPO selection signal. Left:** Normalised final-performance regret from selecting the configuration with lowest final validation loss at each Qwen3 size; zero denotes the best and one the worst configuration under the corresponding aggregate. **Right:** Spearman correlation between negative final validation loss and final Z_{Static}, Z_{Dynamic}, and Z_{All} scores. Positive values indicate that lower loss is associated with better evaluation performance. The minimum-loss choice is never near-optimal, and correlations are weak and inconsistent across scales.

Chat benchmarks produce more consistent configuration rankings. We measure the pairwise Spearman correlation between the configuration rankings produced by benchmarks within each evaluation group. Table 1 shows that mean agreement among static benchmarks is consistently weak across the five Qwen3 sizes, whereas mean agreement among chat benchmarks is consistently positive. Agreement also varies among benchmark pairs: the standard deviation ranges from $\sigma = 0.10$ to 0.20 for chat-benchmark pairs, compared with $\sigma = 0.37$ to 0.64 for static-benchmark pairs. Higher agreement does not establish that chat benchmarks measure ground-truth quality, but it indicates that hyperparameter selection based on the chat group is less sensitive to the particular benchmark chosen. Because the static suite intentionally spans heterogeneous capabilities, its lower agreement is partly expected and should not be interpreted as evidence that its individual benchmarks are invalid. Instead, the result shows that the DPO configuration preferred for one capability may not be preferred for another. The corresponding Llama and Qwen3 (UltraFeedback) analyses in Appendices D.2 and D.1.12 show the same pattern.

Static and chat aggregates favour different configurations. At Qwen3-4B, maximizing Z_{Static} selects $(2\times10^{-6}, 0.01)$, whereas maximizing Z_{Dynamic} selects $(4\times10^{-7}, 0.01)$. Figure 3 shows that these configurations occupy different parts of the Pareto frontier. Thus, the appropriate learning rate depends on whether the intended target is static capability, chat quality, or an explicit aggregate of the two. Appendix D.1.5 provides the corresponding Pareto comparisons for the other four Qwen3 sizes.

Independent judges largely agree on chat-performance rankings. To test sensitivity to judge choice, we compare the Z_{Dynamic} configuration rankings produced independently by Qwen3-Next-80B-A3B-Instruct and Gemma 4 31B-it over the nine-configuration learning-rate–β core. Agreement is high for Qwen3-1.7B and larger models, with Spearman correlations of $\rho = 0.90$, 0.98,

Table 1: **Chat benchmarks produce more consistent DPO configuration rankings than static benchmarks on Qwen3.** Mean pairwise Spearman correlation and population standard deviation across within-group benchmark pairs, computed separately at each size over jointly scored configurations. Validation-loss columns are the mean ρ of $(-\text{eval loss})$ vs. each benchmark in that group

Qwen3 size	Chat ($\bar{\rho} \pm \sigma$)	Static ($\bar{\rho} \pm \sigma$)	Val. loss vs chat	Val. loss vs static
0.6B	0.47 ± 0.20	0.03 ± 0.54	-0.19 ± 0.10	-0.18 ± 0.36
1.7B	0.84 ± 0.10	0.27 ± 0.55	0.50 ± 0.17	0.38 ± 0.37
4B	0.78 ± 0.14	0.18 ± 0.37	0.37 ± 0.06	0.13 ± 0.08
8B	0.57 ± 0.17	0.01 ± 0.64	0.16 ± 0.15	0.01 ± 0.34
14B	0.63 ± 0.18	-0.10 ± 0.58	0.47 ± 0.20	-0.08 ± 0.40

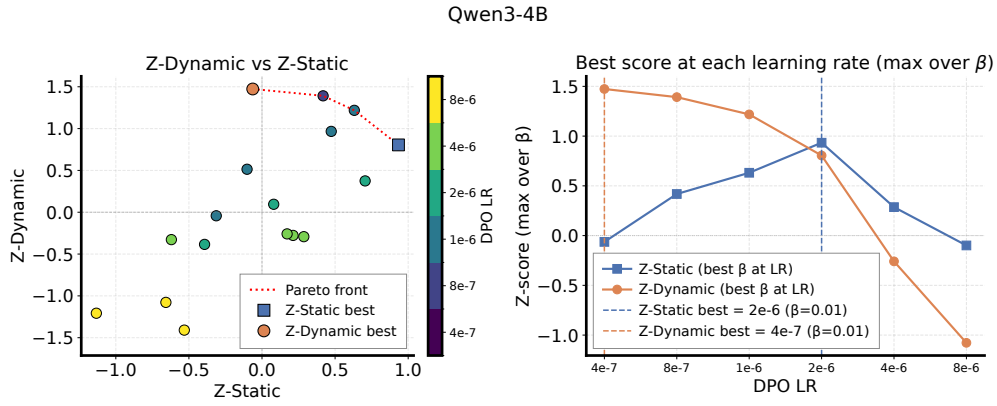

Figure 3: **Dynamic and static evaluations favour different Qwen3-4B DPO learning rates. Left:** Final Z_{Dynamic} and Z_{Static} scores. The red curve marks the Pareto frontier, and squares and circles identify configurations that maximize the Z_{Static} and Z_{Dynamic} scores. **Right:** Best score at each learning rate after maximizing over β.

0.90, and 0.95 at 1.7B, 4B, 8B, and 14B, respectively, but falls to $\rho = -0.27$ at 0.6B. This may reflect insufficiently distinguishable responses at the smallest scale. Within our setting, reliable chat-based model selection therefore emerges only above 0.6B, cautioning against using very small models as proxies for hyperparameter selection. Because the candidate responses and evaluation procedures are held fixed while the judge model changes, this agreement makes it unlikely that the observed coherence is driven solely by idiosyncrasies of a single judge. It does not exclude biases shared across judges or effects of common prompts, aggregation procedures, and comparators. Complete aggregate and benchmark-level correlations appear in Appendix D.3.

4.2 The Hyperparameter Landscape Across Scale

The optimal Qwen3 configuration changes with scale. Under Z_{Dynamic}, Figure 4 shows the learning-rate estimates from two-dimensional quadratic response surfaces over $(\log_{10} \text{LR}, \log_{10} \beta)$ at each Qwen3 size. The estimates rise from 0.6B to 1.7B and then decline through 14B, showing an overall downward trend. We use the fitted line only as a descriptive summary of these five Qwen3 models, not as a monotonic scaling law or a predictor for an unseen size. Appendix D.1.3 compares the corresponding estimates under Z_{All}, Z_{Static}, and validation loss.

The selected configuration changes with the preference dataset. On Dolci-Instruct-DPO, the best observed Z_{All} configuration uses $\beta = 0.01$ at every Qwen3 size. We repeat the shared 3×3 grid with UltraFeedback at all Qwen3 sizes, retaining the corresponding Dolci-trained SFT checkpoint and evaluation suite. The Z_{All} winners agree at 0.6B, differ only in β at 1.7B, and differ in both coordinates at 4B–14B. Dolci-to-UltraFeedback transfer regret rises from 0.00 at 0.6B to 0.25 at 1.7B and 0.67 at 4B, reaching 1.00 at both 8B and 14B, showing that the difference at larger models is consequential. Since $\beta = 0.01$ is the lower Dolci boundary and 4×10^{-6} is the upper Ul-

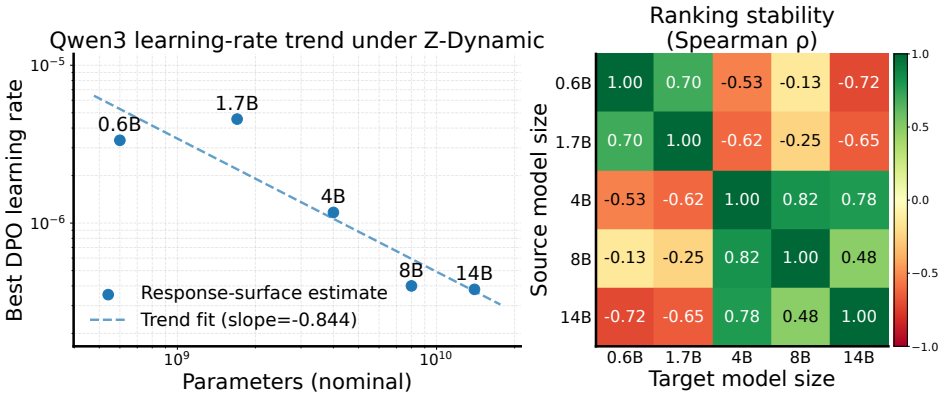

Figure 4: **Learning-rate trend and configuration-ranking stability under Z_{Dynamic} across Qwen3 sizes. Left:** Per-size response-surface estimates of the preferred learning rate and a descriptive fit across all five sizes. **Right:** Spearman correlations between rankings of the nine configurations in the shared (3×3) grid. Positive values indicate similar configuration rankings and hence more plausible transfer, whereas negative values indicate ranking reversals. Rankings agree within the 0.6B–1.7B and 4B–14B groups but reverse across them.

traFeedback boundary, these are preferences within the tested grid rather than located optima. This comparison shows that dataset sensitivity varies across the tested scales; Appendix D.1.12 gives the complete comparison.

4.3 Cheaper Tuning and Transfer

Hyperparameter transfer is unreliable across scales and model families. The right panel of Figure 4 shows pairwise Spearman correlations between the Z_{Dynamic} rankings of the nine configurations shared by every Qwen3 grid. The matrix forms two groups: rankings agree between 0.6B and 1.7B ($\rho = 0.70$) and among the 4B–14B models ($\rho = 0.48$–0.82), but all correlations across these groups are negative ($\rho = -0.72$ to -0.13). Transfer also depends on the metric used. Under Z_{All} (Figure 14), the small-model agreement strengthens ($\rho = 0.93$) and cross-block correlations remain negative ($\rho = -0.78$ to -0.25), but agreement among the larger models largely disappears ($\rho = 0.67$ between 4B and 8B, 0.23 between 4B and 14B, and -0.02 between 8B and 14B). Moreover, positive rank agreement does not guarantee low transfer regret. Model-size distance alone is therefore insufficient to determine whether a configuration will transfer. Additional aggregate-specific, transfer, and cross-family analyses appear in Appendix D.5.1–D.5.3.

Partial runs provide an informative but scale-dependent selection signal. At each Qwen3 size, we compare the intermediate and final Z_{All} rankings over configurations evaluated at both fidelities. After 600 steps ($\approx 30\%$ of training), their Spearman correlations are 0.85, 0.90, 0.85, 0.76, and 0.63 from 0.6B to 14B. At 4B, selecting from the step-600 scores yields zero final Z_{All} regret, while directly transferring the configuration selected at 1.7B incurs maximal regret (1.00). A short target-model run can therefore provide a more useful selection signal than transfer from a neighbouring scale. Under UltraFeedback, dynamic-ranking correlations at step 144, roughly 30% of training, are 0.85, 0.72, 0.75, 0.67, and 0.65 from 0.6B to 14B. Because agreement can fluctuate across checkpoints, we use early evaluations to prioritize promising configurations rather than as a universal stopping rule. The complete Qwen3 trajectories on Dolci-Instruct-DPO, UltraFeedback replications, and Llama trajectories are reported in Appendices D.1, D.1.12, and D.2.

Adaptive search methods outperform random search when given sufficient budget. We compare five search strategies over continuous learning-rate and β domains using four surrogate black-boxes fitted separately to the Qwen3 trajectories: a Gaussian process (GP) and one-, three-, and five-nearest-neighbour (1-NN, 3-NN, and 5-NN) regressors. Each method optimises Z_{Dynamic}, and we evaluate its recommendation at final fidelity. Under the $8\times$ budget, CQR is the only method with a conclusive improvement over random search under all four surrogates (Table 2). BOHB has

Table 2: **Final Z_{Dynamic} improvement over random search under the $8\times$ wall-clock budget.** Differences are paired by seed after averaging the five Qwen3 sizes within each of 25 seeds. Bold marks the largest mean improvement for each surrogate; † denotes a paired 95% interval excluding zero.

Surrogate	TPE	CQR	ASHA	BOHB
Gaussian process	$+0.05^{\dagger}$	$+0.08^{\dagger}$	$+0.08^{\dagger}$	$\mathbf{+0.10}^{\dagger}$
1-NN	$+0.04$	$\mathbf{+0.09}^{\dagger}$	$+0.01$	$+0.03$
3-NN	$+0.06^{\dagger}$	$\mathbf{+0.11}^{\dagger}$	$+0.04^{\dagger}$	$+0.05^{\dagger}$
5-NN	$+0.09^{\dagger}$	$\mathbf{+0.16}^{\dagger}$	$+0.02$	$+0.06^{\dagger}$

the largest GP gain, whereas CQR has the largest gain under each NN surrogate. TPE and BOHB improve conclusively under the GP, 3-NN, and 5-NN surrogates, while ASHA does so under the GP and 3-NN surrogates. Figure 5 shows how GP regret decreases with budget at each model size. The gains are uneven across sizes: separation from random search is largest at 8B and 14B and nearly absent at 1.7B. Thus, the relative advantage of an HPO method depends on the response surface and the model size, although CQR is the most consistent method in these simulations. Under the restrictive budget, no method differs detectably from random search because the budget generally expires before adaptive proposals begin. Complete results and regret curves appear in Appendix D.4.

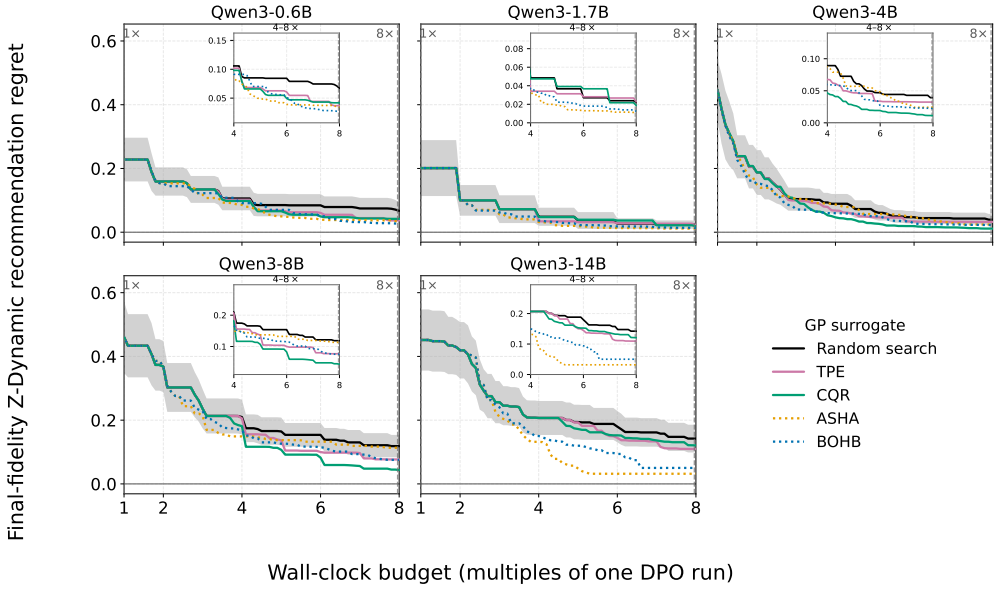

Figure 5: **HPO recommendation regret over time under the GP surrogate.** Each panel shows final-fidelity Z_{Dynamic} recommendation regret for one Qwen3 size as a function of normalised wall-clock budget; lower is better. Lines show the mean and the shaded region shows the 95% t-interval for random search across 25 seeds. Results for other surrogates appear in Appendix D.4.

5 Practical Recommendations

Select configurations using downstream evaluations. Do not use validation loss alone to choose a DPO configuration: its minimum can be far from the best downstream configuration. Nor should a single downstream benchmark determine the choice, since benchmark rankings often disagree. When the goal is conversational response quality, select using multiple chat benchmarks rather than a single metric, and check individual static benchmarks for regressions. Where two judges yield different configuration rankings, treat the selected configuration as evaluation-dependent and include targeted human assessment before making consequential deployment decisions.

Retune for the target setting. Treat a configuration from another model or preference dataset as a starting point for a small target-specific search, not as a default. Search learning rate and β jointly: model-size distance does not reliably predict whether configurations retain their ranking, and changing the preference dataset can alter the selected configuration, with substantial transfer regret at larger model sizes. If the selected value lies at the edge of the tested range, extend the range before interpreting it as a well-localized optimum.

Use partial target-model runs before committing full training. Configuration rankings become informative within 20–30% of training across the model families and preference-data settings studied, making early checkpoints useful for narrowing the candidate set, although they do not guarantee the final winner at every scale. When using adaptive HPO, allocate enough compute for the method to move beyond its initialization phase. Under our larger budget, CQR provides the most consistent improvement, while TPE, ASHA, and BOHB also outperform random search. Because gains vary across model sizes, validate the allocation strategy on the target setting. Our compute budgets include training but exclude evaluations.

6 LIMITATIONS

Limited cross-family and preference-data coverage. The five-size scaling analysis is confined to Qwen3. The three Llama checkpoints span two releases rather than a homogeneous scaling series, so they provide qualitative tests of benchmark agreement and transfer but cannot establish whether the Qwen3 learning-rate trend generalizes across families. We test preference-data sensitivity only on one alternative dataset within one model family, by replacing Dolci-Instruct-DPO with UltraFeedback Binarized while retaining the SFT checkpoint and evaluation suite. One dataset pair cannot establish how common the resulting reversal in preferred learning rate is. The selected Dolci-Instruct learning rates and β values should not be treated as defaults for other datasets.

Finite grids, offline search, and training variance. We select among logged configurations whose coverage differs across sizes; shared-core transfer need not equal full-grid transfer, and boundary selections need not locate unconstrained optima. The Qwen3-only HPO experiments interpolate logged trajectories through four fitted response surfaces rather than adding DPO runs. Each DPO configuration uses one training seed, and benchmark bootstrapping and 25 HPO seeds measure evaluation-composition and search variation, but not training-seed variance.

Evaluation sensitivity. At Qwen3-0.6B, the two judges provide conflicting rankings, possibly because small response differences are especially judge-sensitive at this scale. Arena-Hard and MT-Bench also compare configurations with matched SFT checkpoints, which may introduce comparator-specific judge preferences. This limits conclusions from individual metrics or judges.

7 CONCLUSION

Our findings show that DPO hyperparameter selection depends on the target model, preference data, and evaluation objective. Validation loss can select configurations far from the best-performing downstream configuration, static benchmarks disagree about configuration rankings, and neither parameter-count distance nor a configuration selected on another preference dataset reliably predicts transfer quality. Although the selected Qwen3 learning rate exhibits an overall decline with scale, its small-scale rise and the boundary selection of β argue against a universal configuration or simple transfer rule. Partial target-model runs provide useful ranking information within 20–30% of training across the model families and training data settings studied, offering a lower-cost way to narrow the candidate set before full training. Continuous-surrogate HPO experiments show that informed search can improve final configuration selection when the $8\times$ budget permits adaptive proposals. The released trajectories provide an offline benchmark for studying this trade-off between selection quality and training cost.

AI USE STATEMENT

We used generative AI tools to help polish the writing and presentation of this paper. During experimentation, generative AI was used to help prepare Slurm submission scripts and YAML configuration files; all generated files were manually inspected and verified before use. AI was not used to generate the remaining training and HPO simulation code, which was drawn from their respective mentioned libraries. For evaluation and analysis, generative AI was used to help write plotting code that produced figures from the experimental data. We manually checked the resulting plots and their underlying values for correctness. The authors take full responsibility for the paper and for all artifacts produced with AI assistance.

REPRODUCIBILITY STATEMENT

All training and evaluation settings are documented in Section 3 and the Appendix, including the per-size search spaces, optimisation settings, evaluation protocols, and judge configurations. For review, we release an anonymized checkpoint-level offline benchmark: `https://anonymous.4open.science/r/iclr2027-dpo-hpo-dataset-B911/`. It contains checkpoint-level evaluation data for all Qwen3 (including the UltraFeedback Binarized replications) and Llama DPO experiments, including training fidelity, elapsed training cost, and scores on the eleven downstream benchmarks. Producing each entry requires a full DPO training run and a costly downstream evaluation suite. The completed DPO runs consumed approximately 8,454 H100 GPU-hours and evaluations (including intermediate checkpoints and two judges) consumed 3,395 H100 GPU-hours; Appendix C.1.1 provides the breakdown and accounting scope of the 11,849 total H100 GPU-hours used. The release enables future work to compare selection signals, early-selection policies, transfer rules, and HPO methods without repeating that compute. The blackbox tables additionally provide Z_{Static}, Z_{Dynamic}, Z_{All}, and Z_{Macro} aggregates. The artifact includes manifests, the HPO protocol, and code to reconstruct the Qwen3 blackboxes and surrogate environments. It also provides complete Qwen3 HPO traces across five model sizes, four surrogates, two budgets, three objectives, and five optimisers. The training checkpoints and cluster logs are not included in the review artifact.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, 2019. doi: 10.1145/3292500.3330701.

Loubna Ben Allal, Lewis Tunstall, Nouamane Tazi, Elie Bakouch, Ed Beeching, Carlos Miguel Patiño, Clémentine Fourrier, Thibaud Frere, Anton Lozhkov, Colin Raffel, Leandro von Werra, and Thomas Wolf. The smol training playbook: The secrets to building world-class LLMs. `https://huggingface.co/spaces/HuggingFaceTB/smol-training-playbook`, 2025.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012. URL `http://jmlr.org/papers/v13/bergstra12a.html`.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, volume 24, pp. 2546–2554, 2011.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix

Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chiang24b.html.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 9722–9744, 2024.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. Length-controlled AlpacaEval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=CybBmzWBX0.

Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, Rene Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=1k4rJYEwda-.

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1437–1446, 2018.

L. Franceschi, M. Donini, V. Perrone, A. Klein, C. Archambeau, M. Seeger, M. Pontil, and P. Frasconi. Hyperparameter optimization in machine learning. *Foundations and Trends in Machine Learning*, 2025.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2023. URL https://zenodo.org/records/10256836.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2 edition, 2009.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11170–11189, 2024.

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

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking DPO and PPO: Disentangling best practices for learning from preference feedback. *Advances in Neural Information Processing Systems*, 37:36602–36633, 2024.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Hynek Kydlíček, Guilherme Penedo, Clémentine Fourrier, Nathan Habib, and Thomas Wolf. FineTasks: Finding signal in a haystack of 200+ multilingual tasks. `https://huggingface.co/spaces/HuggingFaceFW/blogpost-fine-tasks`, 2024.

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, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL `https://openreview.net/forum?id=i1uGbfHHpH`.

Houyi Li, Wenzhen Zheng, Qiufeng Wang, Hanshan Zhang, Zili Wang, Shijie Xuyang, Yuantao Fan, Zhenyu Ding, Haoying Wang, Ning Ding, Shuigeng Zhou, Xiangyu Zhang, and Daxin Jiang. Predictable scale: Part I, step law – optimal hyperparameter scaling law in large language model pretraining. *arXiv preprint arXiv:2503.04715*, 2025a. URL `https://arxiv.org/abs/2503.04715`.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, 2:230–246, 2020a.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. PyTorch distributed: Experiences on accelerating data parallel training. *Proceedings of the VLDB Endowment*, 13(12):3005–3018, 2020b.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-Hard and Benchbuilder pipeline. In *Forty-second International Conference on Machine Learning*, 2025b. URL `https://openreview.net/forum?id=KfTf9vFvSn`.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Erlis Lushtaku, Bora Kargi, Ali Elganzory, Fabio Ferreira, Alejandro R. Salamanca, Julia Kreutzer, and David Salinas. JudgeArena: A unified framework for reproducible LLM-judge evaluation. *arXiv preprint arXiv:2608.02620*, 2026.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Meta AI. Llama 3.2 model cards and prompt formats. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Florian Pfisterer, Lennart Schneider, Julia Moosbauer, Martin Binder, and Bernd Bischl. YAHPO Gym – an efficient multi-objective multi-fidelity benchmark for hyperparameter optimization. In *International Conference on Automated Machine Learning*, pp. 3/1–39. PMLR, 2022.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

David Salinas, Matthias Seeger, Aaron Klein, Valerio Perrone, Martin Wistuba, and Cedric Archambeau. Syne Tune: A library for large scale hyperparameter tuning and reproducible research. In *International Conference on Automated Machine Learning*, pp. 16/1–23. PMLR, 2022.

David Salinas, Jacek Golebiowski, Aaron Klein, Matthias Seeger, and Cedric Archambeau. Optimizing hyperparameters with conformal quantile regression. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *PMLR*, pp. 29876–29893, 2023.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aKkAwZB6JV.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning. https://github.com/huggingface/trl, 2022.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450, 2024.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-DPO: Direct preference optimization with dynamic β. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17084–17097. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/8df7c2e3c3c3be098ef7b382bd2c37ba-Paper.pdf.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2023.

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

A Additional Related Work

Hyperparameter optimisation and multi-fidelity search. Hyperparameter optimisation (HPO) seeks a training configuration that performs well while minimizing the cost of evaluating candidate configurations. In single-fidelity HPO, each candidate is trained to completion before the next informed choice is taken. Bayesian optimisation (BO) instead fits a model to completed trials and uses it to propose new configurations (Franceschi et al., 2025). Multi-fidelity HPO additionally treats partial training runs as cheaper, lower-fidelity evaluations and allocates further training only to candidates that seem promising under the chosen evaluation objective. This is the case in Hyperband, ASHA, and BOHB (Li et al., 2018; 2020a; Falkner et al., 2018). We use logged DPO trajectories to compare random search (Bergstra & Bengio, 2012), TPE (Bergstra et al., 2011), CQR (Salinas et al., 2023), ASHA (Li et al., 2020a), and BOHB (Falkner et al., 2018) under matched training-time budgets, treating training progress as fidelity and downstream performance as the objective.

HPO benchmark suites. Benchmark suites such as HPOBench and YAHPO Gym support controlled, reproducible comparisons of HPO methods on expensive multi-fidelity problems (Eggensperger et al., 2021; Pfisterer et al., 2022). HPOBench packages reproducible benchmark problems, whereas YAHPO Gym uses fitted surrogates to make repeated optimiser comparisons inexpensive. Our released benchmark applies this paradigm to DPO. It provides checkpoint-level trajectories, measured training costs, and downstream evaluation outcomes from expensive post-training runs. It therefore enables future work to compare evaluation signals, early-selection policies, transfer rules, and HPO allocation strategies without repeating the underlying DPO training.

B Dataset Details

Dataset versions and splits. Our primary post-training recipe uses the `train` splits of `allenai/Dolci-Instruct-SFT` for supervised fine-tuning and `allenai/Dolci-Instruct-DPO` for preference optimisation (Team Olmo, 2025). The same datasets and preprocessing are used for all Qwen3 and Llama experiments. We split Dolci-Instruct-DPO with `test_size=0.005` and seed 42, yielding 258,622 training pairs and a fixed held-out validation set of 1,300 pairs. The Qwen3 preference-data ablations instead use the `train_prefs` split of `HuggingFaceH4/ultrafeedback_binarized` (Cui et al., 2024),

while retaining the same Dolci-trained SFT checkpoint. Of its 61,135 preference pairs, 60,000 are used for training and 1,135 for validation. Table 3 summarizes the resulting example counts.

Table 3: **Datasets used for post-training.** Hub-split sizes and the numbers retained for training and validation after preprocessing. Repository names and configurations are given in the text.

Dataset	Hub split	Training	Validation
Dolci-Instruct-SFT	2,152,112	1,944,341	—
Dolci-Instruct-DPO	259,922	258,622	1,300
UltraFeedback	61,135	60,000	1,135

SFT formatting and filtering. SFT examples are represented as `messages` and formatted with a custom OLMo-style ChatML template using <|im_start|> and <|im_end|> delimiters. The template includes a default system prompt supporting function calling and is saved with each SFT checkpoint as `chat_template.jinja`. We train with assistant-only loss, masking user and system tokens, and do not use sequence packing. Before training, we discard examples with a missing or invalid `messages` field or with `content=None` in any turn, reducing Dolci-Instruct-SFT from 2,152,112 to 1,944,341 examples. We apply no further deduplication or subsampling. The maximum SFT sequence length is 8,192 tokens for all Qwen3 and all Llama models; longer conversations are truncated by `SFTTrainer`.

DPO formatting and truncation. The DPO runs load the tokenizer and chat template stored with their corresponding SFT checkpoint. For Dolci-Instruct-DPO, the `chosen` and `rejected` conversations are retained while turn-level metadata is removed, leaving only `role` and `content`; any existing `prompt` or `messages` columns are removed so that TRL reconstructs the common prompt from the two conversations. The same processing is applied to UltraFeedback after retaining only its `chosen` and `rejected` columns. Dolci-Instruct-DPO uses a maximum sequence length of 4,096 tokens and a maximum prompt length of 512 tokens. UltraFeedback uses 2,048 and 1,024 tokens, respectively. Both use TRL's default `keep_end` truncation mode, which retains the end of an overlength prompt and truncates the response as needed to satisfy the total sequence-length limit. Beyond constructing the validation splits, we apply no additional filtering, deduplication, or subsampling to either DPO dataset.

C Additional Details On The Experimental Setup

C.1 Training Configurations

Supervised fine-tuning. We sweep SFT learning rates and use 3×10^{-5} for the SFT checkpoint that initializes every primary DPO run. SFT is run for two epochs, corresponding to approximately 30,382 optimiser steps, with a global batch size of 128. The per-device batch size is one. Most runs use gradient accumulation over 16 steps on eight GPUs; Qwen3-14B uses eight accumulation steps on 16 GPUs, preserving the same global batch size. We optimise with `adamw_torch`, Adam coefficients $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, no weight decay, and maximum gradient norm 1.0. The learning rate follows a linear schedule with a 0.03 warmup ratio. Training uses bfloat16 precision and seed 42. Checkpoints are saved every 3,000 steps for Qwen3 and Llama models.

Direct preference optimisation. The primary DPO grids vary learning rate and β while holding the remaining settings fixed. Each configuration is trained for one epoch, with the logged trajectories ending at approximately 2,031 optimiser steps. We use a per-device batch size of one on eight GPUs and 16 gradient-accumulation steps, giving a global batch size of 128. The optimiser is `adamw_torch` with the same Adam coefficients, numerical stability constant, zero weight decay, and maximum gradient norm as SFT. We use a linear learning-rate schedule with a 0.1 warmup ratio, bfloat16 precision, seed 8, the standard sigmoid DPO loss, and zero label smoothing. Checkpoints are saved every 200 steps, and the final checkpoint is also evaluated. Downstream evaluations are run separately on these saved checkpoints.

The Qwen3 UltraFeedback experiments use the same DPO recipe, except for the sequence-length settings reported in Appendix B and 32 gradient-accumulation steps. With four GPUs, this gives

Table 4: **Shared SFT and DPO optimisation settings.** Learning rate and β are the variables swept during DPO; their per-model search spaces are given in Table 7. UltraFeedback uses the primary DPO values except where stated in the text.

Setting	SFT	Primary DPO
Epochs	2	1
Learning rate	3×10^{-5}	Swept
β	—	Swept
Per-device batch size	1	1
Global batch size	128	128
Optimiser	`adamw_torch`	`adamw_torch`
Adam (β_1, β_2); ϵ	$(0.9, 0.999)$; 10^{-8}	$(0.9, 0.999)$; 10^{-8}
Scheduler; warmup ratio	linear; 0.03	linear; 0.10
Weight decay; maximum gradient norm	0; 1.0	0; 1.0
Precision	bfloat16	bfloat16
Seed	42	8
Checkpoint interval	3,000 steps	200 steps

a global batch size of 128. Its one-epoch runs contain approximately 469 optimiser steps; for the trajectory analysis, we evaluate checkpoints every 48 steps and the final checkpoint.

Hardware and distributed training. All training runs use NVIDIA H100 GPUs and are executed through Slurm with four GPUs per node. Table 6 reports the number of GPUs and distributed strategy used for each model group. The primary Dolci-Instruct DPO runs use eight GPUs, the UltraFeedback ablation uses four, and the largest SFT run, Qwen3-14B, uses 16 GPUs.

C.1.1 Compute Accounting

Table 5: **H100 GPU-hours for the reported DPO experiments.** Training and evaluation columns include completed jobs only. 'Dolci' denotes Dolci-Instruct-DPO and 'UF' denotes UltraFeedback Binarized.

Series	Size	Training	Evaluation	Total
Qwen3/Dolci	0.6B	234	410	644
	1.7B	315	356	671
	4B	1,787	723	2,510
	8B	1,578	500	2,078
	14B	2,809	614	3,423
	Subtotal	6,723	2,603	9,326
Llama 3/Dolci	1B	153	49	202
	3B	300	138	438
	8B	771	151	922
	Subtotal	1,224	338	1,562
Qwen3/UF	0.6B	52	66	118
	1.7B	49	68	117
	4B	72	112	184
	8B	169	86	255
	14B	165	122	287
	Subtotal	507	454	961
Reported-workload total		8,454	3,395	11,849

We estimate compute as the number of allocated NVIDIA H100 GPUs multiplied by job wall-clock time. Table 5 reports training and benchmark-evaluation compute for the model and preference-data combinations analysed in this paper. Each distinct evaluation is counted once; when an evaluation was resubmitted, we retain only the latest completed job. Evaluation compute includes LM Evaluation Harness tasks, response generation, AlpacaEval, LLM-judge scoring, and Elo estimation at final

Table 6: **Distributed training configuration.** Each node contributes four GPUs. DDP denotes distributed data parallel training (Li et al., 2020b); ZeRO stages refer to DeepSpeed parameter and optimiser sharding (Rasley et al., 2020).

Models	SFT GPUs	SFT strategy	DPO GPUs	DPO strategy
Qwen3-0.6B and Qwen3-1.7B	8	DDP	8	DDP
Qwen3-4B	8	DDP	8	ZeRO-3
Qwen3-8B	8	ZeRO-2	8	ZeRO-3
Qwen3-14B	16	ZeRO-3	8	ZeRO-3
Llama-3.2-1B and Llama-3.2-3B	8	DDP	8	DDP
Llama-3.1-8B	8	ZeRO-3	8	ZeRO-3
Qwen3-0.6B (UltraFeedback)	—	—	4	DDP
Qwen3-1.7B (UltraFeedback)	—	—	4	DDP
Qwen3-4B (UltraFeedback)	—	—	4	DDP
Qwen3-8B (UltraFeedback)	—	—	4	ZeRO-3
Qwen3-14B (UltraFeedback)	—	—	4	ZeRO-3

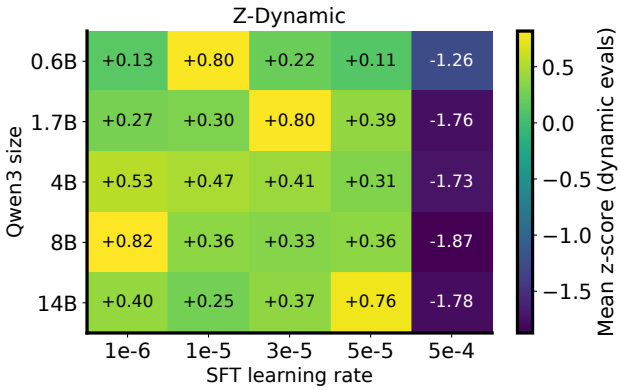

Figure 6: **SFT learning-rate sensitivity on dynamic evaluations.** Mean Z_{Dynamic} score after SFT, standardized separately within each Qwen3 size across the five SFT learning rates. Rankings vary by model size, but 5×10^{-4} is consistently poor.

and intermediate checkpoints. Distinct evaluations account for approximately 29% of the reported training-and-evaluation workload. Jobs that were cancelled/crashed mid-run, SFT runs, and offline HPO simulation are excluded.

C.2 SFT Learning-Rate Sweep

We evaluate SFT learning rates $\in \{10^{-6}, 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}, 5 \times 10^{-4}\}$ for each Qwen3 model size. Scores are standardized separately within each model size and evaluation aggregate. No single moderate learning rate value dominates across all sizes or across our aggregates (Z_{All}, Z_{Static}, Z_{Dynamic}), with the relative rankings varying with both scale and objective. We therefore hold the SFT learning rate fixed at 3×10^{-5} in the primary DPO experiments, rather than selecting a separate SFT optimum per size, which would confound cross-size DPO comparisons. The sweep serves as a sensitivity check that this choice lies in the non-degenerate plateau rather than as a claim that 3×10^{-5} is uniquely optimal. By contrast, 5×10^{-4} consistently yields substantially worse downstream scores, especially on Z_{Dynamic} (Figure 6).

C.3 Search Spaces per Model Scale

Every primary grid contains the nine configurations in the shared core

$$\{1, 2, 4\} \times 10^{-6} \text{ (learning rate)} \quad \times \quad \{0.01, 0.02, 0.04\} \ (\beta).$$

This common core enables like-for-like transfer comparisons across Qwen3 sizes and between Qwen3 and Llama. We extend the grid at individual Qwen3 sizes to cover the locally promis-

Table 7: **DPO search spaces.** "Core" denotes the shared 3×3 Cartesian product defined above. Added configurations extend this support for a particular model or dataset.

Model	DPO search space	# Configurations
Qwen3-0.6B	Core $+ (8 \times 10^{-6}, 0.01)$	10
Qwen3-1.7B	Core $+ (8 \times 10^{-6}, 0.01)$	10
Qwen3-4B	Core + size-specific extensions	18
Qwen3-8B	$\{0.4, 0.8, 1, 2, 4\} \times 10^{-6} \times \{0.01, 0.02, 0.04\}$	15
Qwen3-14B	Core + size-specific extensions	17
Llama-3.2-1B	Core	9
Llama-3.2-3B	Core	9
Llama-3.1-8B	Core	9
Qwen3-0.6B (UltraFeedback)	Core	9
Qwen3-1.7B (UltraFeedback)	Core	9
Qwen3-4B (UltraFeedback)	Core	9
Qwen3-8B (UltraFeedback)	Core	9
Qwen3-14B (UltraFeedback)	Core	9

Table 8: **Static evaluation settings.** Shots are the few-shot examples supplied by the evaluation harness.

Benchmark	Split (n)	Shots	Metric
ARC-Challenge	test (1,172)	25	Normalised accuracy
GPQA-Diamond	train (198)	0	Normalised accuracy
GSM8K	test (1,319)	5	Strict exact match
HellaSwag	validation (10,042)	10	Normalised accuracy
PIQA	validation (1,838)	0	Normalised accuracy
TruthfulQA-MC2	validation (817)	0	Accuracy
IFEval	train (541)	0	Mean strict/loose accuracy

ing learning-rate range and, at 4B, an additional β value. Table 7 gives the exact configurations. Cross-scale and cross-family transfer are evaluated on the shared core, whereas full-grid optima and response-surface analyses use every available configuration at the corresponding size.

Beyond the shared core, Qwen3-4B adds $\beta = 0.08$ at learning rates 1, 2, and 4×10^{-6}; all four tested β values at 8×10^{-6}; and $(4 \times 10^{-7}, 0.01)$ and $(8 \times 10^{-7}, 0.01)$. Qwen3-14B adds $(4 \times 10^{-7}, \beta)$ and $(8 \times 10^{-7}, \beta)$ for $\beta \in \{0.01, 0.02, 0.04\}$, plus $(1 \times 10^{-7}, 0.01)$ and $(8 \times 10^{-8}, 0.01)$.

For the Dolci-Instruct grids, we evaluate checkpoints every 200 steps and at the final training step. For UltraFeedback, we evaluate checkpoints every 48 steps and at the final step. These intervals correspond to evaluations at approximately 10% intervals of training progress for both datasets. These trajectories support the early-selection and offline HPO analyses. The validation loss directed Bayesian-optimisation sweeps use continuous learning-rate and β domains and are described separately in Appendix E.

C.4 Evaluation Benchmark Settings

C.4.1 Static Benchmarks

We evaluate static tasks with `lm-evaluation-harness`. Model weights are loaded in bfloat16 precision, and log-likelihood tasks use scaled dot-product attention. Table 8 reports the splits, sample counts, few-shot settings, and metrics; inference details are given immediately below it. Except for IFEval, these tasks are evaluated as raw-text completions without applying the model's chat template, matching their standard harness protocols and facilitating comparison with previously reported results.

ARC-C, GPQA-Diamond, HellaSwag, PIQA, and TruthfulQA-MC2 are scored by comparing the likelihood of the answer choices; normalised accuracy is used where shown in Table 8. GSM8K uses greedy generation with a limit of 1,024 new tokens and its standard stop strings. IFEval also uses greedy generation, with a limit of 1,280 new tokens, and averages prompt- and instruction-level

Table 9: **Chat-evaluation protocols.** The SFT comparator is the checkpoint used to initialize the corresponding DPO run.

Evaluation	Comparator	Reported score
Arena-Hard v2.0	Matched SFT checkpoint	Bootstrapped win rate
MT-Bench	Matched SFT checkpoint	Pairwise win rate
AlpacaEval 2.0	GPT-4 reference; 805 prompts	Length-controlled win probability
Elo	LMArena pool; 500 prompts/opponent	Bradley–Terry rating

Table 10: **Decoding settings for chat evaluation.** Limits are candidate/judge output tokens.

Evaluation	Candidate temperature	Judge temperature	Token limit
Arena-Hard v2.0	0	0	4,096 / 4,096
MT-Bench	Category-specific	0	4,096 / 2,048
AlpacaEval 2.0	0	1 (one token)	2,048 / 1
Elo	0.6 ($p = 0.95$)	0.6 ($p = 0.95$)	32,768 / 32,768

accuracy under both strict and loose matching. GPQA and IFEval use the only splits supplied by their source datasets. TruthfulQA's nominal zero-shot setting includes the fixed six-example QA preamble embedded in its standard prompt.

IFEval chat-template convention. The default IFEval scores apply the chat template stored with each model. These scores are used in the heatmaps, aggregate trajectories, benchmark correlations, and phase-gain analyses.

C.4.2 Chat-based evaluations

We evaluate every chat benchmark independently with two judges: Qwen3-Next-80B-A3B-Instruct-FP8 and Gemma 4 31B-it. Both judges evaluate the same generated candidate and comparator responses using identical prompts, decoding settings, response-order rules, tie handling, and aggregation procedures; only the judge model changes. This controlled comparison isolates sensitivity to judge choice. Table 9 summarizes the comparator and reported score for each benchmark; response-order and aggregation details follow the table.

Arena-Hard judges every comparison in both response orders; ties contribute 0.5, and strong preferences receive threefold weight in the bootstrap. MT-Bench uses a fixed order with the candidate first and the SFT comparator second; ties again contribute 0.5. Its reference-assisted prompts are used for relevant mathematics, reasoning, and coding questions. AlpacaEval randomizes response order deterministically for each instruction and computes a continuous preference from the two possible one-token judgments. Elo evaluates each comparison in both orders and fits a Bradley–Terry model to the resulting pairwise scores; exact ties and unparseable judgments contribute 0.5 to each response.

Table 10 gives the decoding settings for candidate-response generation and both judges. Arena-Hard and AlpacaEval generate candidate responses greedily. MT-Bench follows the FastChat category-specific temperatures. Elo is the only pipeline that samples both candidate and judge outputs: submitted jobs inherit the JudgeArena `ChatVLLM` defaults of temperature 0.6 and `top_p` $= 0.95$. Consequently, individual Elo evaluations are stochastic across repeated executions, unlike the other three pipelines under the submitted settings.

For MT-Bench, candidate-generation temperatures are kept at their default values: 0.7 for writing and role-playing, 0.1 for STEM and humanities, and 0 for extraction, mathematics, coding, reasoning, and Arena-Hard-200 categories. We use the reference prompt templates supplied by Arena-Hard v2.0, MT-Bench, and AlpacaEval 2.0.

C.4.3 Held-out DPO loss

For the primary Dolci-Instruct grids, we construct the fixed 1,300-pair validation set described in Appendix B. We do not evaluate on this set during DPO training. After training, we evaluate the

latest saved checkpoint, or the final model when no intermediate checkpoint is available, once on the held-out pairs. The resulting loss is the validation loss value used in Section 4.1. The validation loss directed Bayesian-optimisation experiments use the same held-out set but evaluate loss every 100 training steps and save checkpoints every 400 steps.

C.5 Hyperparameter-optimisation Simulation Settings

Compared optimisers. We compare three single-fidelity methods, which train each proposed configuration to completion, with two methods that use intermediate training results to allocate further compute:

- **Random search** samples learning-rate and β configurations independently of previous outcomes (Bergstra & Bengio, 2012).
- **TPE** models regions associated with good and poor completed trials using separate density estimators, then proposes configurations favoured by their density ratio (Bergstra et al., 2011).
- **CQR** fits conformalized quantile-regression models to predict performance and uncertainty from completed configurations, then uses these estimates to propose subsequent configurations (Salinas et al., 2023).
- **ASHA** uses asynchronous successive halving: candidates with strong partial-training results receive more training, while weaker candidates are stopped at intermediate resource levels (Li et al., 2020a).
- **BOHB** combines model-based configuration proposals with Hyperband-style allocation of training resources across candidates (Falkner et al., 2018).

Simulation protocol and compute budgets. Each simulation searches a continuous domain in $(\log_{10} \mathrm{LR}, \log_{10} \beta)$ bounded by the minimum and maximum values measured at that model size. Training progress is the fidelity variable, with checkpoints at steps 200, 400, . . ., 2,000 and the final step 2,031; the surrogate's cumulative training time is the trial cost. Two trials run in parallel, and each optimiser–surrogate–size setting is repeated with 25 random seeds. We define one budget unit as the measured wall-clock duration of one complete reference DPO run. The larger budget is eight units, which allows CQR's five initial random trials to finish and leaves time for adaptive proposals. The restrictive budget is one unit, corresponding to approximately two full-run equivalents of total compute across both workers.

We compare random search, the tree-structured Parzen estimator (TPE), conformalized quantile regression (CQR), asynchronous successive halving (ASHA), and Bayesian optimisation with Hyperband (BOHB). For TPE, we set `top_n_percent`=33 and `num_min_data_points`=3; all other TPE settings use the implementation defaults. Random search, TPE, and CQR evaluate configurations to completion, whereas ASHA and BOHB may terminate candidates at intermediate checkpoints. All methods receive the same time limit, workers, and continuous domain. Z_{Dynamic} is the primary maximization objective; Z_{All} is retained as a cross-objective evaluation.

Successive-halving schedules. ASHA and BOHB use a single successive-halving bracket with a grace period of 400 steps and reduction factor two, yielding comparisons at steps 400, 800, and 1,600 before final fidelity. Approximately half of the active configurations are retained at each comparison. Delaying the first decision until step 400 avoids pruning at step 200, where the 8B and 14B Z_{Dynamic} rankings are poorly aligned with their final orderings.

Continuous surrogate environments. The logged grids answer only at observed configurations, so we construct continuous surrogate versions of every per-size benchmark. Learning rate and β are transformed to base-10 logarithms, and no query extrapolates outside the measured range. Training step remains discrete, and one fitted model predicts the complete trajectory at the recorded fidelities.

We fit four alternatives separately and never pool their predictions: one-nearest-neighbour regression, which is piecewise constant between measured configurations; distance-weighted three- and five-nearest-neighbour regression; and a Gaussian process with a constant component, Matérn-$5/2$ kernel, and white-noise term initialized to 10^{-2}. GP targets are normalised, kernel optimisation uses

three restarts, and the fitting seed is zero. These environments test sensitivity to different interpolation assumptions and do not constitute additional observed DPO runs.

Recommendation and scoring. At a budget cutoff, we recommend the trial with the highest search score at final step 2,031 whenever at least one trial has completed. If no trial has completed, we compare only trials evaluated at the highest fidelity shared within that run. We then evaluate the recommended (LR, β) at true final fidelity on the same surrogate. This prevents an optimiser from benefiting merely because it reports more intermediate checkpoints. For pooled comparisons, we first average the five model-size outcomes within each seed and then form a 95% t-interval across the 25 seed-level averages. Differences from random search are paired by seed.

D Additional Experimental Results

D.1 Qwen3 Experiments

D.1.1 Benchmark-Level Agreement And Validation Loss

Figure 7 gives the complete benchmark-level rank-correlation matrices underlying the aggregate analyses in Section 4.1. Each matrix shows that the relationship between negative validation loss and downstream rankings varies across both benchmarks and model sizes. It can align positively with some evaluations while being weakly or negatively associated with others, explaining why validation loss does not provide a reliable aggregate selection signal.

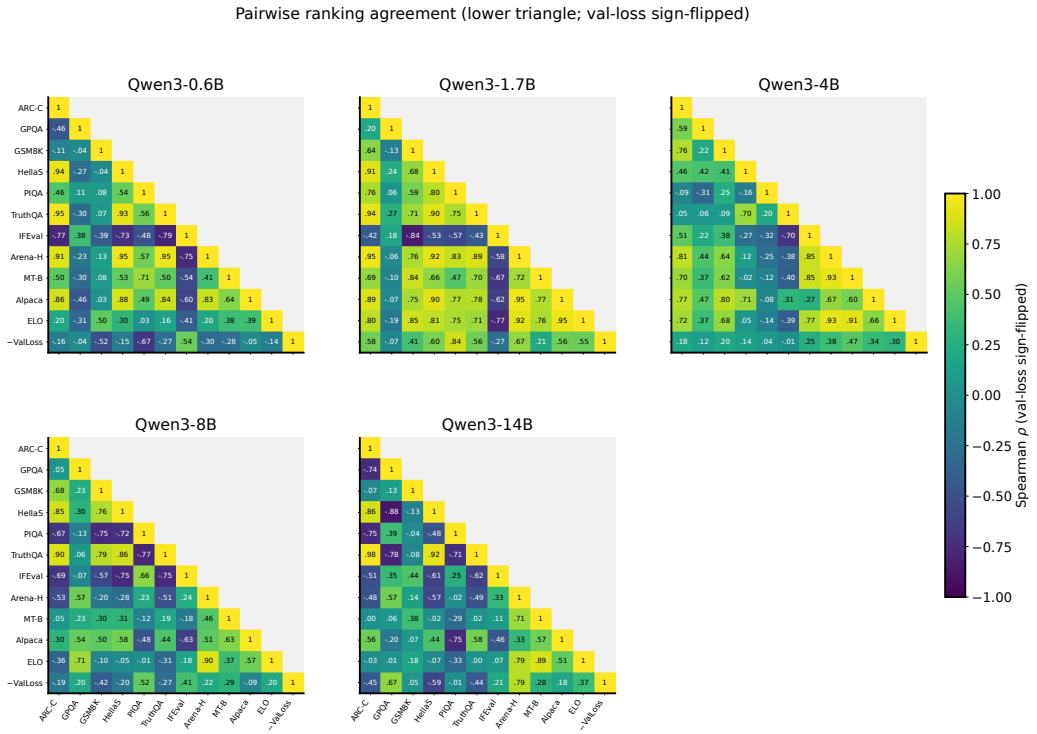

Figure 7: **Benchmark-level configuration-ranking correlations and negative validation loss across Qwen3 sizes.** Each cell gives the Spearman correlation between two final configuration rankings within one model-size grid. The final row and column use negative validation loss, so positive values indicate that lower validation loss is associated with stronger downstream performance. Benchmark agreement and alignment with validation loss both vary substantially by metric and model size.

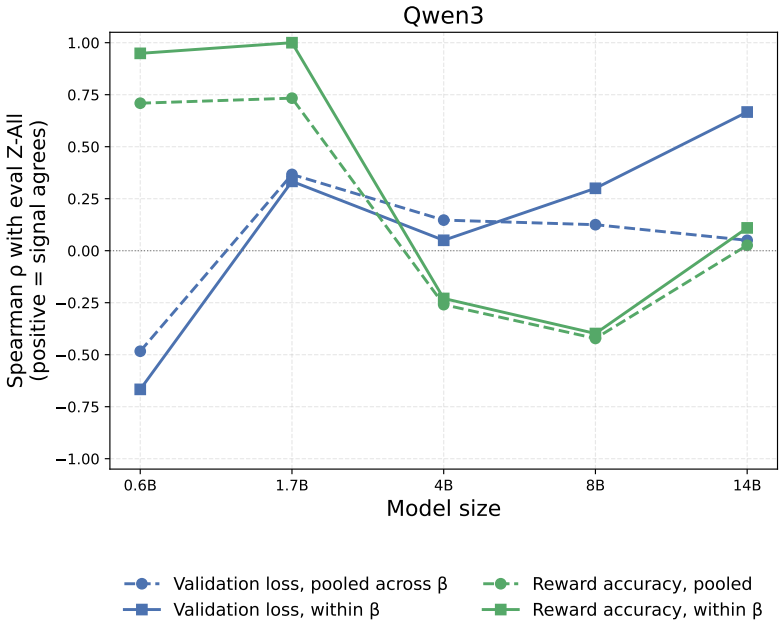

Figure 8: **Agreement of final DPO signals with Z_{All} across Qwen3 sizes.** Dashed lines rank configurations across the full grid; solid lines rank learning rates separately within each β before pooling the ranks. Blue denotes negative validation loss, so positive correlation means lower loss agrees with higher Z_{All}; green denotes reward accuracy.

D.1.2 Validation signals at fixed β

DPO loss changes with β, so its values need not be directly comparable across the full grid. As a sensitivity check, we rank learning rates by lower final validation loss and higher final Z_{All} separately within each β, then pool those within-β ranks to compute Spearman correlation. From Qwen3-0.6B through 14B, the resulting correlations are -0.67, 0.33, 0.05, 0.30, and 0.67. The corresponding full-grid correlations are -0.48, 0.37, 0.15, 0.12, and 0.05. Fixed-β comparisons therefore do not produce consistently strong agreement with downstream performance, although the relationship varies substantially by size.

We also compare TRL training reward accuracy with Z_{All}. Unlike the magnitude of the DPO loss, the chosen-versus-rejected decision is unchanged by multiplying a fixed policy's reward margin by positive β. Its within-β correlations are 0.95, 1.00, -0.23, -0.40, and 0.11 across the same sizes. Reward accuracy is informative at the two smallest Qwen3 sizes but does not consistently rank configurations at larger sizes. Each β slice contains only a few learning rates, so these correlations describe the tested configurations rather than precise population relationships.

D.1.3 Objective-Specific Learning-Rate Trends

The learning-rate trend in Section 4.2 uses Z_{Dynamic}, but the descriptive relationship depends on the evaluation objective. Figure 9 compares the corresponding learning-rate estimates for Z_{All}, Z_{Static}, Z_{Dynamic}, and validation loss. At each Qwen3 size, we fit a two-dimensional quadratic response surface over $(\log_{10}\mathrm{LR}, \log_{10}\beta)$ and extract its preferred learning rate.

The estimates differ across selection signals, particularly at larger sizes: Z_{Static} favours higher learning rates than Z_{Dynamic}, while validation loss follows a different pattern again. These response-surface estimates summarize preferences within the measured grids; they are neither unconstrained optima nor predictions for an unseen model size.

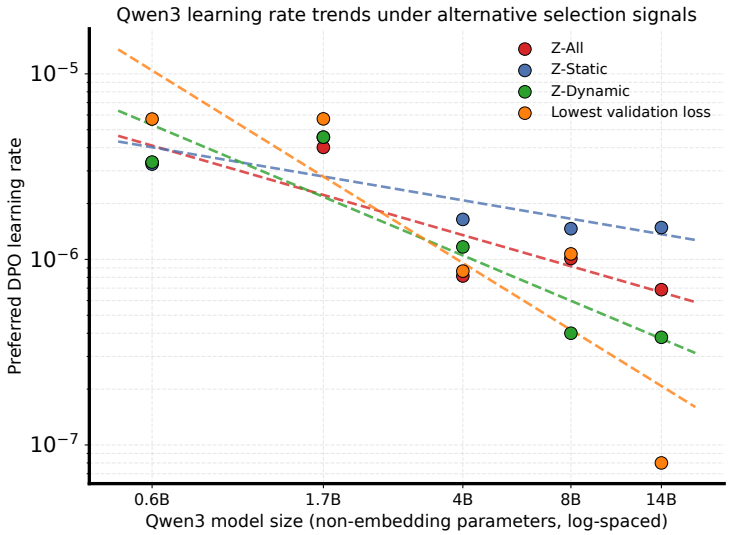

Figure 9: **Preferred Qwen3 DPO learning rates differ by selection signal.** Each point is the learning-rate component of a two-dimensional quadratic response-surface optimum over $(\log_{10} \mathrm{LR}, \log_{10} \beta)$ at one Qwen3 size. The three downstream aggregates are maximized, whereas validation loss is minimized. The estimates describe preferences within the measured grids and are not extrapolated scaling laws.

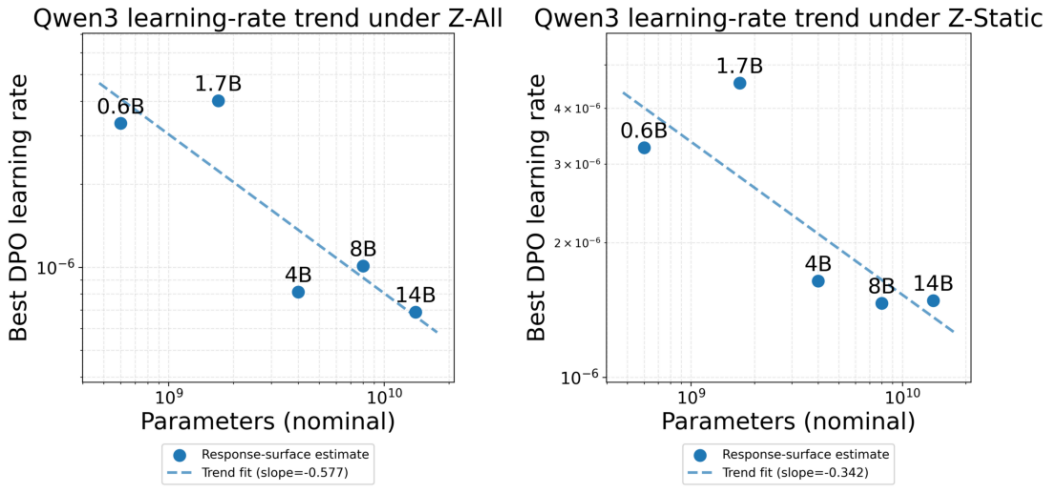

Figure 10: **Objective-specific Qwen3 learning-rate trends. Left:** Z_{All}. **Right:** Z_{Static}. Each point is the learning-rate estimate from a two-dimensional response surface at one Qwen3 size; dashed lines are descriptive log-linear fits across the five estimates. These within-family fits are not held-out predictions.

D.1.4 Final aggregate optima

Table 11 reports the best observed final configuration under each evaluation aggregate. The Z_{All} selections reproduce the non-monotonic pattern discussed in Section 4.2: the learning rate first rises from 0.6B to 1.7B and then decreases through 14B. The objective-specific columns show that this pattern should not be separated from the evaluation criterion. At 8B, for example, Z_{Dynamic} selects a learning rate of 4×10^{-7} whereas Z_{Static} selects 4×10^{-6}. The selected β also differs across objectives at 0.6B, 1.7B, and 14B.

Table 11: **Best observed final Qwen3 configuration under each aggregate.** Each entry is $(\text{learning rate}, \beta)$ and is selected from the full grid available at that model size. $\downarrow$ and $\uparrow$ mark the lowest and highest tested learning rates, respectively. Wherever selected, $\beta = 0.01$ is the lower tested bound; the 0.6B Z_{Static} winner uses the upper bound $\beta = 0.04$. Boundary selections do not locate unconstrained optima.

Model	Z_{All}	Z_{Macro}	Z_{Dynamic}	Z_{Static}
Qwen3-0.6B	$(4 \times 10^{-6}, 0.01)$	$(4 \times 10^{-6}, 0.01)$	$(8 \times 10^{-6}, 0.01)^{\uparrow}$	$(4 \times 10^{-6}, 0.04)$
Qwen3-1.7B	$(8 \times 10^{-6}, 0.01)^{\uparrow}$	$(8 \times 10^{-6}, 0.01)^{\uparrow}$	$(4 \times 10^{-6}, 0.02)$	$(8 \times 10^{-6}, 0.01)^{\uparrow}$
Qwen3-4B	$(2 \times 10^{-6}, 0.01)$	$(1 \times 10^{-6}, 0.01)$	$(4 \times 10^{-7}, 0.01)^{\downarrow}$	$(2 \times 10^{-6}, 0.01)$
Qwen3-8B	$(1 \times 10^{-6}, 0.01)$	$(4 \times 10^{-7}, 0.01)^{\downarrow}$	$(4 \times 10^{-7}, 0.01)^{\downarrow}$	$(4 \times 10^{-6}, 0.01)^{\uparrow}$
Qwen3-14B	$(4 \times 10^{-7}, 0.01)$	$(4 \times 10^{-7}, 0.01)$	$(4 \times 10^{-7}, 0.01)$	$(4 \times 10^{-6}, 0.02)^{\uparrow}$

D.1.5 Dynamic–static trade-offs across Qwen3 sizes

Figure 11 extends the Qwen3-4B comparison in Figure 3 to the remaining model sizes. At each size, the configurations maximizing Z_{Dynamic} and Z_{Static} occupy different locations on, or near, the Pareto frontier. Thus, the objective-specific learning-rate choices in Table 11 reflect a dynamic–static trade-off across the Qwen3 family rather than only the 4B grid.

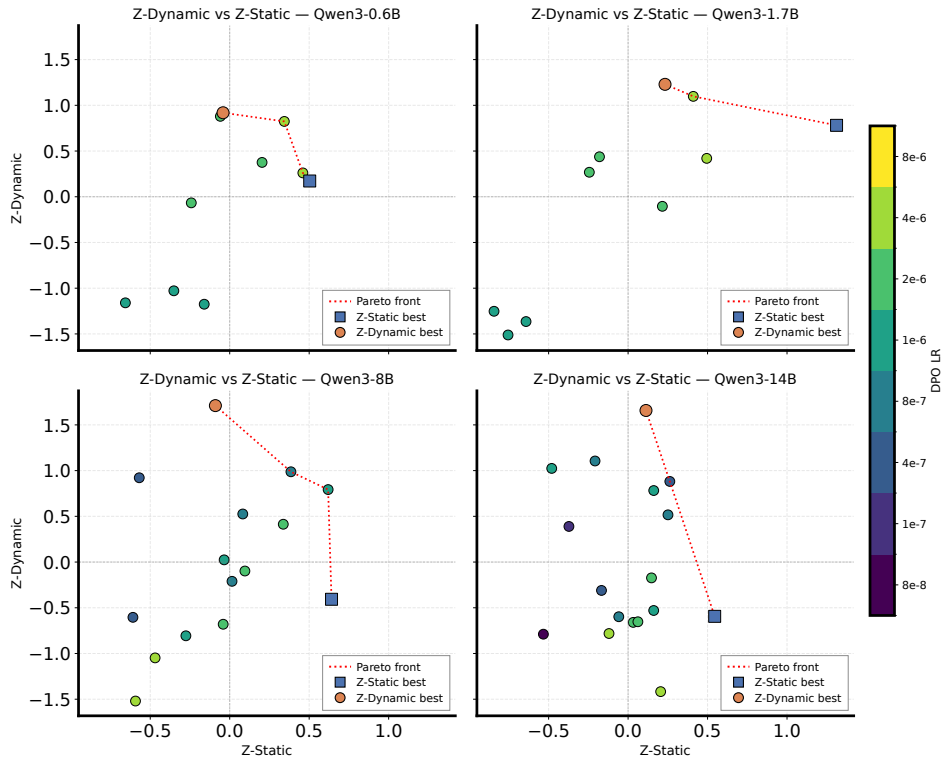

Figure 11: **Dynamic–static trade-offs at the remaining Qwen3 sizes.** Each point is one final DPO configuration, with Z_{Dynamic} and Z_{Static} standardized within its model-size grid. Red segments mark the per-size Pareto frontier; squares and circles identify the configurations maximizing Z_{Static} and Z_{Dynamic}, respectively. Point colour denotes DPO learning rate on a common logarithmic scale.

D.1.6 Z_{All} response surfaces

Figure 12 shows the final Z_{All} response surface over the configurations evaluated at each Qwen3 size. The preferred region shifts toward lower learning rates at larger sizes, while the irregular and

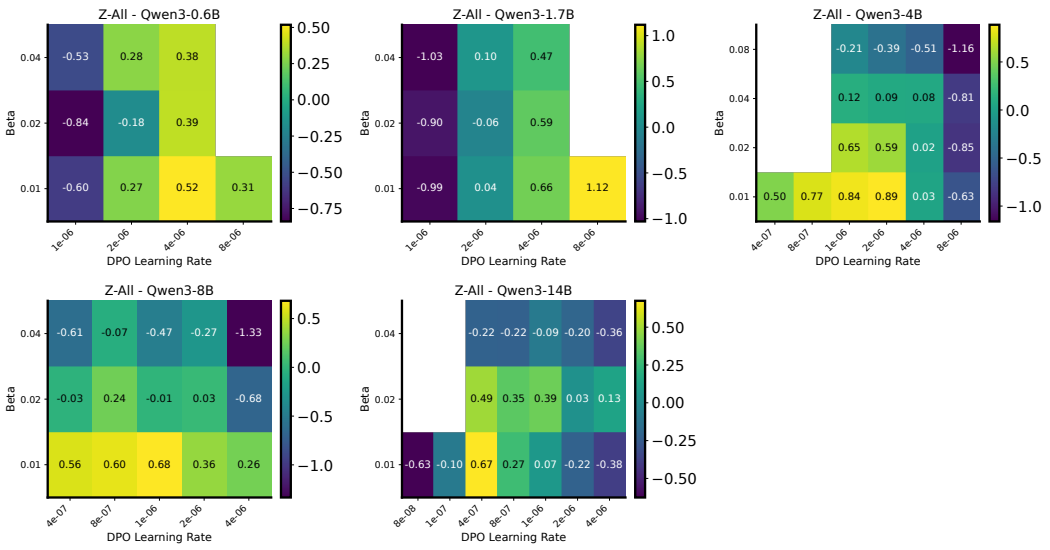

Figure 12: **Final Z_{All} response surfaces across Qwen3 sizes.** Each cell gives the final all-benchmark aggregate score of one evaluated DPO configuration; lighter cells indicate stronger performance. The evaluated grids differ by size, so blank cells denote unobserved configurations and colour scales are normalised separately within each panel. The surfaces show a shift toward lower preferred learning rates at larger scales, but also substantial size-specific variation in the relative ordering of configurations.

size-specific surface shapes reinforce that a configuration selected at one scale need not rank well at another. Blank cells denote configurations that were not evaluated at that size.

D.1.7 Equal-group aggregate robustness

Z_{All} weights each benchmark equally, which assigns seven of eleven total weights to the static group. We therefore define

$$Z_{\mathrm{Macro}} = \tfrac{1}{2}\left(Z_{\mathrm{Dynamic}} + Z_{\mathrm{Static}}\right),$$

giving chat and static benchmarks equal total weight. The Z_{All} and Z_{Macro} configuration rankings remain highly correlated at every Qwen3 size ($\rho = 0.903$–0.994). Their selected configurations match at 0.6B, 1.7B, and 14B, while the preferred learning rate differs at 4B and 8B. Thus, equal group weighting can alter the precise selected configuration, but it preserves the main conclusions about validation loss and cross-scale ranking instability.

Minimum-validation-loss selection remains unreliable under Z_{Macro}: its mean normalised regret is 0.57, and the combined Spearman correlation between negative validation loss and Z_{Macro} is only 0.16. Cross-scale agreement also remains non-monotonic in model-size distance (Figure 14).

D.1.8 Stability of the selected configuration

We ask whether the selected configuration depends heavily on the particular benchmarks included in an aggregate. For each aggregate, we repeatedly resample its constituent benchmarks with replacement, recompute configuration scores on the shared 3×3 core, and record whether the winner from the original aggregate remains selected. Figure 15 reports this recovery probability. All three aggregates recover their original winner more often than the $1/9$ uniform reference, but stability differs by objective. Z_{Dynamic} recovers its winner in 52.0–93.7% of resamples and is more stable than Z_{Static} at every Qwen3 size. Z_{All} recovers its winner in 48.3–70.6% of resamples. This analysis measures sensitivity to benchmark composition only and not the uncertainty within an individual benchmark.

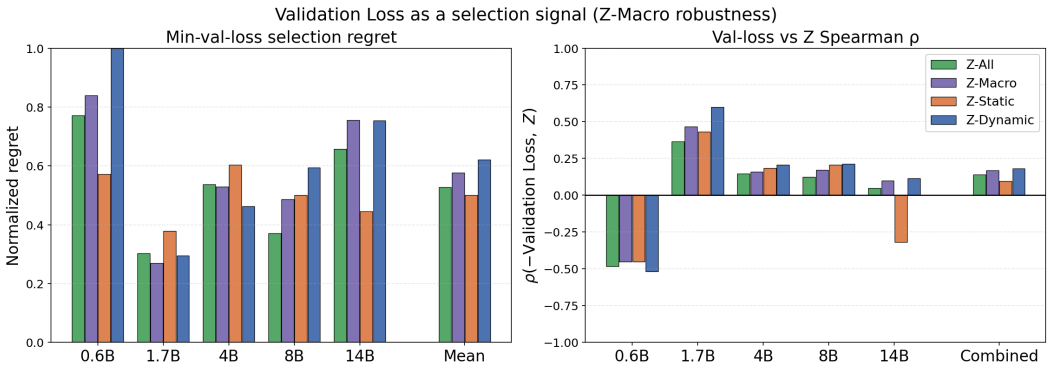

Figure 13: **Validation loss remains an unreliable selection signal under equal group weighting.** The Z_{Macro} aggregate gives chat and static benchmarks equal total weight. Selecting the minimum-validation-loss configuration remains far from the best Z_{Macro} configuration at every Qwen3 size, and validation-loss rankings agree only weakly with Z_{Macro}.

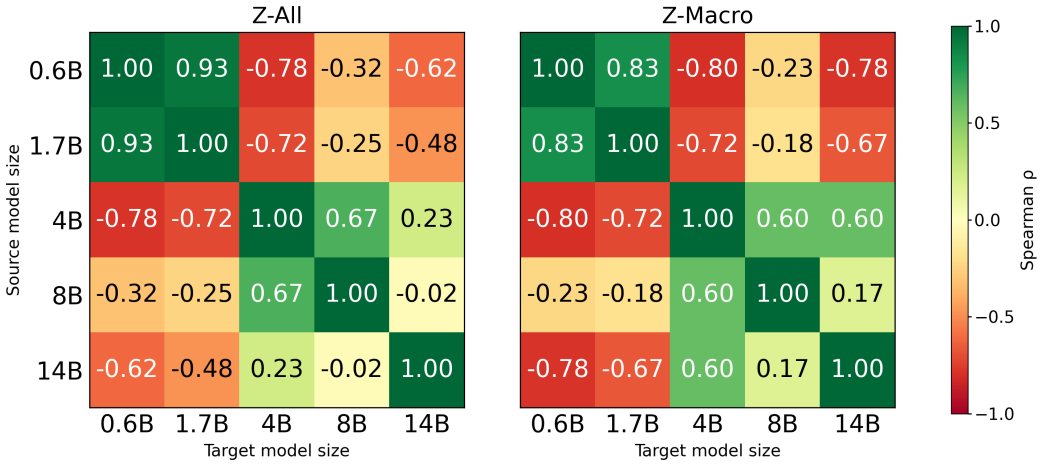

Figure 14: **Cross-scale ranking instability persists under equal group weighting.** Each cell gives Spearman's ρ between rankings of the nine configurations in the shared 3×3 grid. Z_{All} weights all eleven benchmarks equally, whereas Z_{Macro} assigns equal total weight to chat and static benchmarks. Neither aggregate shows a consistent relationship between model-size distance and ranking agreement.

D.1.9 Benchmark-pair agreement

Figure 16 expands the group summaries in Table 1 into their constituent benchmark pairs. Every chat-benchmark pair is positively correlated at every model size, although the strength of agreement varies. Static-benchmark pairs span strong positive and negative correlations at the same size. Their low mean agreement therefore reflects conflicting configuration rankings across capabilities rather than uniformly uninformative benchmarks.

D.1.10 Ranking stability during training

Figure 17 reports the complete ranking-stability trajectories underlying the step-600 values in Section 4.3. For this longitudinal comparison, each benchmark is standardized using the mean and population standard deviation of its final scores across the corresponding model-size grid; the same transformation is then applied to every intermediate checkpoint. This keeps the benchmark scale fixed over training. At each step, we compute the Spearman correlation between the intermediate and final configuration rankings using configurations observed at both points.

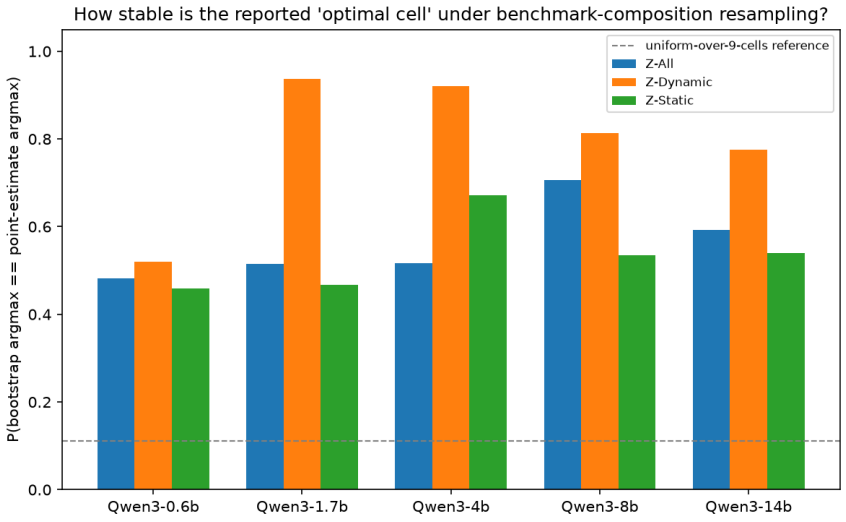

Figure 15: **Winner recovery under benchmark bootstrap resampling.** Bars show how often resampling the benchmarks within an aggregate selects the same configuration as the original aggregate on the shared nine-configuration core. The dotted line gives the $1/9$ uniform reference. Z_{Dynamic} has the most stable winner at every Qwen3 size.

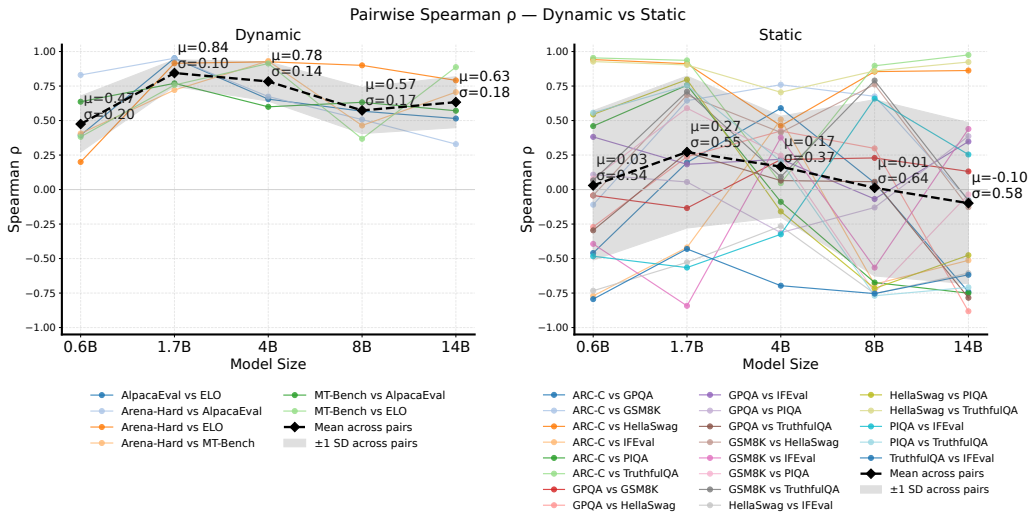

Figure 16: **Pairwise agreement between Qwen3 configuration rankings.** Each coloured line gives the Spearman correlation between one pair of benchmarks at each model size. Black diamonds denote the mean across the six chat pairs (left) or 21 static pairs (right), and the shaded regions show one population standard deviation across pairs. "Dynamic" in the figure denotes the four chat-based evaluations.

The trajectories confirm that useful ranking information appears early but at different rates across scales and objectives. At step 600, Z_{All} correlations are 0.85, 0.90, 0.85, 0.76, and 0.63 from 0.6B through 14B. The 1.7B ranking is already highly stable at the first checkpoint, while 14B improves more gradually. Several trajectories fluctuate after becoming informative, which is why we use early scores to allocate additional training rather than prescribe one universal stopping checkpoint.

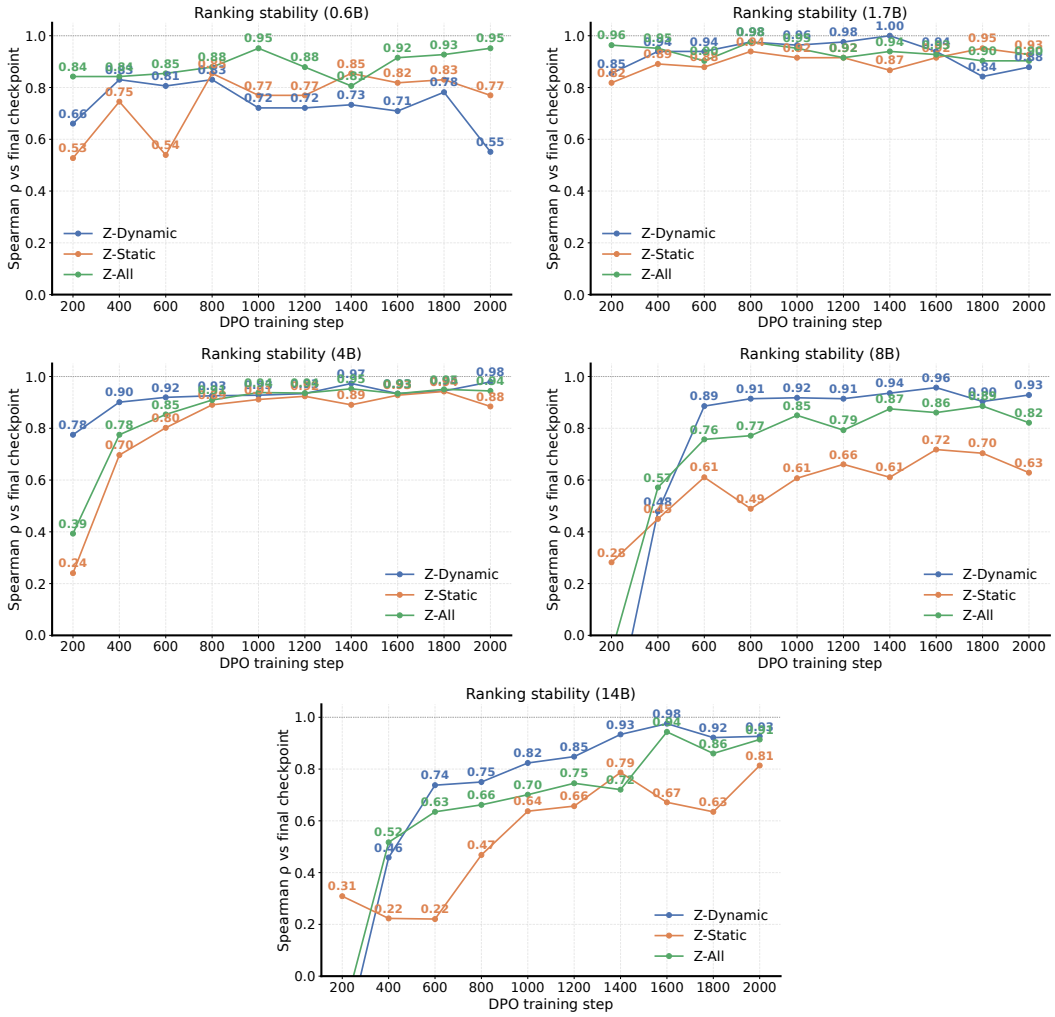

Figure 17: **Intermediate Qwen3 rankings increasingly resemble the final ranking, but convergence is scale- and objective-dependent.** Each panel compares the ranking of all available (learning rate, β) configurations at an intermediate checkpoint with their ranking after the full DPO run. Lines report Spearman correlation for Z_{All}, Z_{Dynamic}, and Z_{Static}.

D.1.11 Early-selection regret

Rank correlation measures the full ordering, whereas configuration selection depends mainly on the top of that ordering. Figure 18 therefore evaluates the final regret of the configuration that would be selected at each checkpoint. For this cross-sectional calculation, benchmark means and standard deviations are recomputed separately at every checkpoint over the configurations available at that exact step. We then select the highest-scoring Z_{All} configuration and measure its final Z_{All} regret relative to the best and worst final configurations in that model-size grid.

Low or zero regret can be reached well before exhaustive training, but the best selection checkpoint is not monotonic or shared across sizes. This variability motivates multi-fidelity allocation rather than a fixed early-stopping rule.

D.1.12 UltraFeedback preference-data ablation

We repeat the shared nine-configuration DPO grid on UltraFeedback at all five Qwen3 model sizes. These experiments retain the corresponding Dolci-trained SFT checkpoint, DPO objective, and evaluation suite, while using UltraFeedback and its dataset-specific sequence-length settings. Table 12

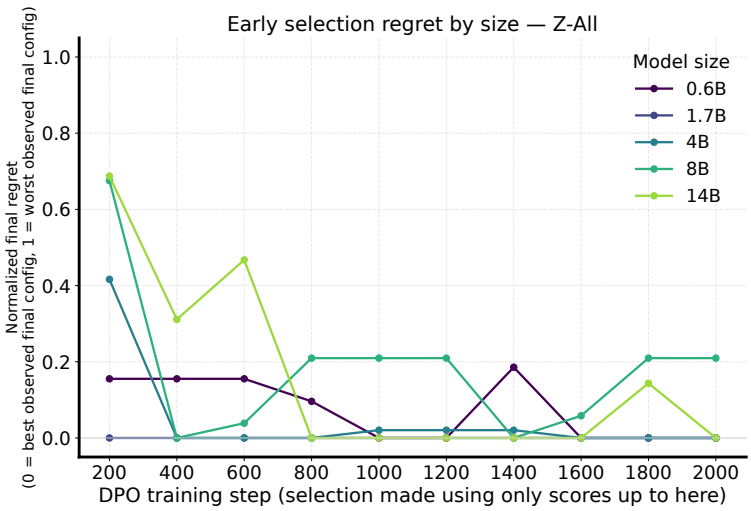

Figure 18: **Early target-model selection can reduce tuning cost, but its reliability varies by scale.** Final normalised Z_{All} regret when the configuration is selected using only the scores at each intermediate checkpoint. Regret is zero for the best final configuration and one for the worst.

Table 12: **Preference-data sensitivity across Qwen3 sizes.** Z_{All} winners are selected from the shared 3×3 grid. Transfer regret evaluates each dataset's winner on the other dataset and normalizes using the target dataset's best and worst shared-grid configurations. All Dolci winners use the lower tested β boundary, while all UltraFeedback winners use the upper tested learning-rate boundary; the latter also select the upper β boundary at 1.7B–14B.

Size	Dolci winner	UltraFeedback winner	Dolci $\rightarrow$ UF	UF $\rightarrow$ Dolci
0.6B	$(4 \times 10^{-6}, 0.01)$	$(4 \times 10^{-6}, 0.01)$	0.00	0.00
1.7B	$(4 \times 10^{-6}, 0.01)$	$(4 \times 10^{-6}, 0.04)$	0.25	0.11
4B	$(2 \times 10^{-6}, 0.01)$	$(4 \times 10^{-6}, 0.04)$	0.67	0.92
8B	$(1 \times 10^{-6}, 0.01)$	$(4 \times 10^{-6}, 0.04)$	1.00	1.00
14B	$(1 \times 10^{-6}, 0.01)$	$(4 \times 10^{-6}, 0.04)$	1.00	1.00

compares the Z_{All} winners on the common grid and measures the regret from transferring each dataset's selected configuration to the other dataset.

The shared-grid winners agree at 0.6B, differ only in β at 1.7B, and differ in both learning rate and β at 4B–14B. Cross-dataset regret follows the same pattern: it is zero at 0.6B, limited at 1.7B, and substantial at 4B–14B. This five-size comparison strengthens the evidence for preference-data sensitivity, but does not establish a general scaling relationship. Moreover, every UltraFeedback Z_{All} winner uses the upper learning-rate boundary, 4×10^{-6}; the 1.7B–14B winners also use the upper tested β. These results therefore identify preferences within the shared grid.

At 0.6B, Z_{Static} varies over a much narrower range than Z_{Dynamic}, making its particular argmax less informative. The stronger and more consistent UltraFeedback result is that Z_{Dynamic} and Z_{All} both favour the upper tested learning rates at all sizes.

Figure 19 compares intermediate and final rankings across the five UltraFeedback sizes. Dynamic-evaluation rankings are generally informative by 144 steps, roughly 30% of training: correlations are at least 0.65 for all sizes. The 1.7B dynamic ranking is a useful counterexample to a fixed earlier cutoff, falling to 0.08 at step 96 before recovering at step 144. These results support using early target-model evaluations to prioritize candidates, while reinforcing that a fixed early-stopping point is not reliable across objectives or model sizes.

Benchmark agreement under UltraFeedback. The qualitative difference between dynamic and static benchmark agreement also holds under UltraFeedback. Across all tested sizes, dynamic

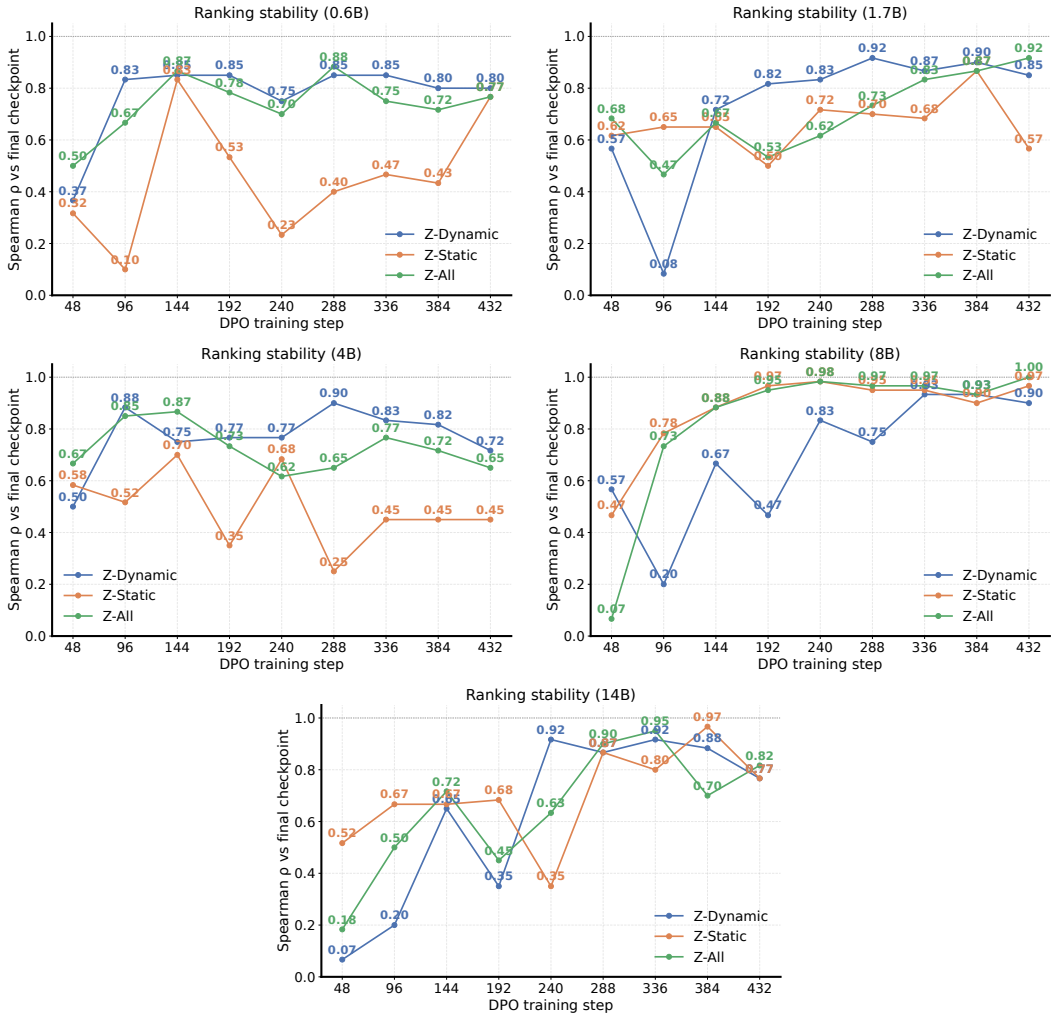

Figure 19: **Intermediate-ranking stability across UltraFeedback model sizes.** Spearman correlation between each intermediate configuration ranking and the final ranking on the shared nine-configuration grid. Dynamic rankings usually become informative by roughly 30% of training, whereas static rankings fluctuate more across checkpoints and sizes.

Table 13: **Benchmark-ranking agreement under UltraFeedback.** Mean pairwise Spearman correlation and population standard deviation across within-group benchmark pairs, computed separately at each size over the shared nine configurations.

Qwen3 size	Dynamic ($\bar{\rho} \pm \sigma$)	Static ($\bar{\rho} \pm \sigma$)
0.6B	0.49 ± 0.21	0.09 ± 0.35
1.7B	0.62 ± 0.14	0.40 ± 0.34
4B	0.39 ± 0.18	0.16 ± 0.48
8B	0.57 ± 0.12	0.53 ± 0.31
14B	0.57 ± 0.19	-0.01 ± 0.57

Table 14: **Best observed final Llama configuration under each aggregate.** Each entry is $(\text{learning rate}, \beta)$, selected from the nine-configuration grid. $\downarrow$ and $\uparrow$ mark the lowest and highest tested learning rates, respectively; $\beta = 0.01$ is the lower tested bound wherever selected. Boundary selections do not locate unconstrained optima.

Model	Z_{All}	Z_{Dynamic}	Z_{Static}
Llama 3.2-1B	$(2 \times 10^{-6}, 0.01)$	$(4 \times 10^{-6}, 0.01)^{\uparrow}$	$(2 \times 10^{-6}, 0.01)$
Llama 3.2-3B	$(4 \times 10^{-6}, 0.01)^{\uparrow}$	$(2 \times 10^{-6}, 0.01)$	$(4 \times 10^{-6}, 0.01)^{\uparrow}$
Llama 3.1-8B	$(4 \times 10^{-6}, 0.02)^{\uparrow}$	$(1 \times 10^{-6}, 0.02)^{\downarrow}$	$(4 \times 10^{-6}, 0.02)^{\uparrow}$

benchmarks produce higher mean pairwise configuration-ranking correlations than static benchmarks (Table 13). Thus, the greater consistency of chat-based evaluations is not specific to the Dolci-Instruct-DPO preference data.

D.2 Llama Experiments

We repeat the principal analyses on Llama 3.2 at 1B and 3B parameters and on Llama 3.1 at 8B, using the same Dolci-Instruct SFT and preference data, training objective, nine-configuration core, and evaluation suite as in the Qwen3 experiments. Because the 8B model belongs to a different Llama release, we use it as a complementary transfer test rather than treating the three sizes as observations from one scaling series.

D.2.1 Final aggregate optima

Table 14 reports the best observed final configuration under each aggregate. Under Z_{All}, the selected learning rate rises from 2×10^{-6} at Llama 3.2-1B to 4×10^{-6} at Llama 3.2-3B, matching the small-scale rise observed for Qwen3. We do not fit a Llama learning-rate scaling law, since the 8B experiment changes both model size and model version. The differences across columns also reinforce that an "optimal" DPO configuration is conditional on the evaluation objective: Z_{Dynamic} and Z_{Static} select different learning rates at every Llama size.

D.2.2 Benchmark agreement and sensitivity to AlpacaEval

Table 15 gives the mean and population standard deviation of pairwise benchmark-ranking correlations. Using all four chat benchmarks, chat agreement exceeds static agreement at 1B and 3B. At 8B, however, AlpacaEval ranks the configurations differently from the other three chat evaluations: the mean chat correlation is 0.32 with a standard deviation of 0.61. Excluding AlpacaEval in a sensitivity analysis raises the 8B chat mean to 0.92 and reduces its standard deviation to 0.04. The same exclusion has much smaller effects at 1B and 3B. This does not determine which evaluation is correct, but it shows that an aggregate can conceal disagreement specific to one benchmark and model setting. The selected 8B Z_{All} and Z_{Dynamic} configurations are unchanged by this exclusion, even though the agreement statistics change substantially. One plausible contributor is the evaluation mechanism: AlpacaEval is the only chat evaluation in our study implemented as a one-token pairwise classifier, with its preference recovered from the relative probabilities of two label tokens and no generated rationale. This protocol difference may make its rankings more sensitive to prompt or

Table 15: **Agreement between Llama configuration rankings.** Entries are the mean pairwise Spearman correlation ρ and population standard deviation σ across benchmark pairs. "Chat without AE" excludes AlpacaEval 2.0 as a sensitivity analysis.

Model	Chat	Chat without AE	Static
Llama 3.2-1B	0.65 ± 0.18	0.54 ± 0.13	0.05 ± 0.62
Llama 3.2-3B	0.84 ± 0.09	0.87 ± 0.07	0.37 ± 0.57
Llama 3.1-8B	0.32 ± 0.61	0.92 ± 0.04	0.01 ± 0.66

Table 16: **Pairwise Spearman correlations between Llama chat-benchmark rankings.** Each correlation is computed over the nine DPO configurations at that model size. The upper block contains the three pairs retained by the analysis without AlpacaEval 2.0.

Benchmark pair	Llama 3.2-1B	Llama 3.2-3B	Llama 3.1-8B
Arena-Hard–MT-Bench	0.63	0.85	0.95
Arena-Hard–Elo	0.63	0.97	0.95
MT-Bench–Elo	0.35	0.80	0.87
Arena-Hard–AlpacaEval	0.95	0.88	−0.28
MT-Bench–AlpacaEval	0.60	0.68	−0.42
AlpacaEval–Elo	0.73	0.88	−0.17

calibration effects. However, our comparison does not isolate this mechanism from other differences in prompts, reference responses, or aggregation, so we treat it solely as our hypothesis.

Table 16 reports the individual chat-benchmark pairs behind these summaries. The three comparisons that do not involve AlpacaEval remain strongly positive at 8B, while all three comparisons involving AlpacaEval are negative. At 1B and 3B, every chat pair is positively correlated.

D.2.3 Ranking stability and early selection

Figure 20 applies the fixed-final normalization described in Appendix D.1: each benchmark's final-grid mean and population standard deviation are used to transform every checkpoint before comparing intermediate and final rankings. Llama 3.2-1B and 3B develop high ranking correlations early in training. The Llama 3.1-8B Z_{All} trajectory is less stable and changes markedly when AlpacaEval is excluded, consistent with the pairwise disagreement above. This sensitivity concerns the ordering of the full grid; the final Z_{All}-selected configuration remains $(4 \times 10^{-6}, 0.02)$ in both variants.

Figure 21 measures the top-of-ranking decision directly, using the per-checkpoint standardization defined for Figure 18. At step 400, approximately 20% of a full run, early Z_{All} selection identifies the eventual best configuration at 3B and 8B and incurs normalised regret of 0.145 at 1B. The 1B optimum is first selected at step 1,400, and its selected configuration fluctuates again at the final intermediate checkpoint. Thus, the Llama results confirm that partial target-model runs can sharply reduce tuning cost, while also showing why a fixed stopping point cannot guarantee the best configuration at every scale.

D.3 Judge Agreement Across Chat Evaluations

We compare configuration rankings rather than raw scores because the two judges need not use the same numerical scale. For each model size and chat benchmark, we restrict the comparison to configurations evaluated by both Qwen3-Next-80B-A3B-Instruct and Gemma 4 31B-it and compute their Spearman rank correlation. To compare the aggregate rankings, we standardize each judge's scores separately within the common configuration pool, average the four resulting benchmark z-scores into Z_{Dynamic}, and correlate the two Z_{Dynamic} rankings.

Table 17 reports the complete results. Aggregate agreement is high from 1.7B onward, with ρ between 0.90 and 0.98, and is 0.79 when all 52 jointly evaluated configurations are pooled. Agreement is not uniformly high at the benchmark level: for example, MT-Bench has $\rho = 0.45$ at 14B and Elo has $\rho = 0.57$ at 8B. Nevertheless, their aggregate rankings remain strongly correlated, suggest-

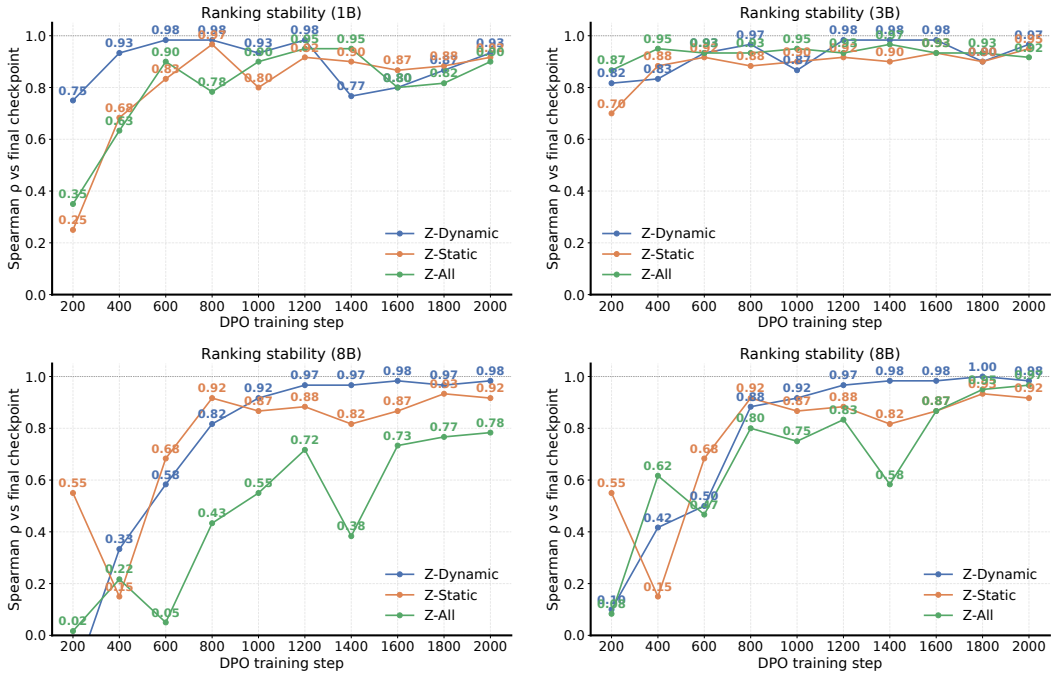

Figure 20: **Intermediate-to-final ranking stability for the Llama grids.** The top row reports Llama 3.2-1B and 3B. The bottom row reports Llama 3.1-8B using the complete evaluation suite (left) and with AlpacaEval 2.0 excluded from Z_{All} and Z_{Dynamic} (right). Lines give the Spearman correlation between intermediate and final configuration rankings.

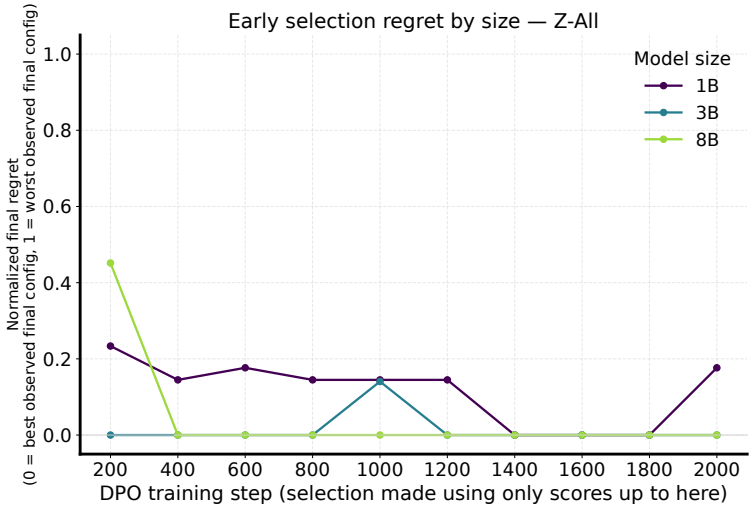

Figure 21: **Early target-model selection on the Llama grids.** Final normalised Z_{All} regret of the configuration selected at each intermediate checkpoint. Regret is zero for the best final configuration and one for the worst.

ing that averaging several chat evaluations reduces sensitivity to judge-specific differences on an individual benchmark.

The 0.6B grid is the clear exception. Its Z_{Dynamic} rankings are negatively correlated ($\rho = -0.27$), driven particularly by Elo ($\rho = -0.73$) and, more weakly, MT-Bench ($\rho = -0.17$). The practical effect is also visible near the top of the ranking: among the five configurations preferred by each judge, the overlap is only one for Elo and two for MT-Bench, compared with three for Arena-Hard

Table 17: **Agreement between configuration rankings produced by the two LLM judges on the shared Qwen3 core.** Entries are Spearman correlations over the nine learning-rate–β configurations shared by every Qwen3 size. Z_{Dynamic} averages separately standardized Arena-Hard, MT-Bench, AlpacaEval 2.0, and Elo scores.

Qwen3 size	Z_{Dynamic}	Arena-Hard	MT-Bench	AlpacaEval	Elo
0.6B	-0.27	0.53	-0.17	0.27	-0.73
1.7B	0.90	0.98	0.87	0.95	0.73
4B	0.98	0.85	0.93	0.97	0.93
8B	0.90	0.78	0.52	0.83	0.57
14B	0.95	0.83	0.45	0.88	0.77

and AlpacaEval. At 4B, by contrast, the judges share four of their top five Arena-Hard configurations and all five for each of the other chat benchmarks. These comparisons support using a second judge for consequential decisions at the smallest scale, while also showing that high aggregate agreement does not imply identical judgments on every constituent benchmark.

D.4 Continuous-Surrogate HPO Results

Table 18 reports the end-of-budget comparison. Every method searches Z_{Dynamic}, recommendations are selected using the common-fidelity rule in Appendix C.5, and each recommendation is scored at final fidelity. Under the $8\times$ wall-clock budget, CQR is the only method with a paired 95% interval excluding zero for every surrogate. BOHB has the largest mean improvement under the Gaussian-process surrogate (0.098), whereas CQR has the largest improvement under the 1-NN, 3-NN, and 5-NN environments (0.09, 0.11, and 0.16). TPE improves conclusively under the GP, 3-NN, and 5-NN surrogates; ASHA does so under the GP and 3-NN environments.

The pooled GP result is heterogeneous across sizes. Table 19 shows that CQR improves conclusively at 0.6B, 4B, and 8B, with its largest gain at 8B (0.20). At 14B, ASHA and BOHB have the largest conclusive gains (0.29 and 0.24), whereas CQR is inconclusive. At 1.7B, none of the informed methods has a conclusive advantage. These differences reinforce the need to validate the search strategy on the target setting rather than infer its value from a single scale.

Under the restrictive one-unit budget, all five methods have statistically indistinguishable pooled Z_{Dynamic} scores for every surrogate. This budget permits only approximately two full-run equivalents across both workers and generally expires before the methods can move beyond their common initialization. We therefore treat it as evidence that adaptive search requires an initialization budget, rather than as a comparison of the optimisers themselves.

Figures 22–25 show recommendation quality throughout the larger-budget runs. The larger budget is labelled "generous" inside these figures. These curves use only the configuration recommended at each cutoff and score it at final fidelity; they do not use the maximum checkpoint score observed so far. The curves show that relative ordering can differ by surrogate and can change when the Z_{Dynamic}-selected recommendation is evaluated on Z_{All}. Per-size recommendation curves for every surrogate are included in the released artifact; Table 19 provides the corresponding compact GP summary.

These simulations interpolate the logged runs rather than adding new DPO observations. Agreement across the four independently fitted response surfaces makes the optimiser comparison less dependent on one interpolation scheme, but it does not remove uncertainty between measured configurations. We therefore interpret the results as evidence about search behaviour on the released offline benchmark, not as additional evidence about the exact location of the DPO optimum.

D.5 Transferring Hyperparameters

D.5.1 Cross-scale ranking agreement by evaluation aggregate

The main paper reports cross-scale ranking agreement under Z_{Dynamic}. Figure 26 compares it with the static-benchmark Z_{Static} aggregates. The same lack of a simple size-based transfer pattern remains under both aggregates. Under Z_{Dynamic}, the two smallest models agree positively, as do the

Table 18: **Final Z_{Dynamic} scores and paired improvements under the $8\times$ budget.** The random-search column gives its mean score and 95% t-interval. Other columns give the paired difference from random search, after averaging the five sizes within each seed. Intervals are over 25 seed-level differences. Higher values are better; bold marks the largest mean improvement in each row. Comparisons use unrounded interval endpoints.

Surrogate	RS score	TPE Δ	CQR Δ	ASHA Δ	BOHB Δ
GP	1.07 [1.04, 1.10]	+0.052 [0.005, 0.100]	+0.077 [0.034, 0.120]	+0.083 [0.050, 0.116]	**+0.098** [0.058, 0.138]
1-NN	1.22 [1.18, 1.27]	+0.04 [−0.02, 0.11]	**+0.09** [0.04, 0.15]	+0.01 [−0.05, 0.07]	+0.03 [−0.02, 0.09]
3-NN	1.07 [1.04, 1.11]	+0.06 [0.01, 0.12]	**+0.11** [0.07, 0.16]	+0.04 [0.01, 0.08]	+0.05 [0.02, 0.09]
5-NN	0.95 [0.93, 0.98]	+0.09 [0.03, 0.14]	**+0.16** [0.11, 0.20]	+0.02 [−0.01, 0.05]	+0.06 [0.03, 0.10]

Table 19: **Gaussian-process HPO improvement over random search by model size.** Entries are paired mean differences in final Z_{Dynamic} over 25 seeds. Bold entries have paired 95% intervals excluding zero.

Qwen3 size	TPE	CQR	ASHA	BOHB
0.6B	**+0.06**	**+0.05**	**+0.06**	**+0.08**
1.7B	0.00	+0.01	+0.02	+0.02
4B	+0.02	**+0.08**	+0.04	+0.05
8B	+0.10	**+0.20**	0.00	+0.10
14B	+0.08	+0.06	**+0.29**	**+0.24**

4B, 8B, and 14B models, but rankings reverse between these groups. Under Z_{Static}, agreement is more heterogeneous and several adjacent model pairs have weak or negative correlation. Thus, the instability of configuration rankings across scale is not specific to the chat-benchmark aggregate.

D.5.2 Directed cross-scale transfer

We evaluate direct transfer on the nine configurations shared by all five Qwen3 grids. For each source size, we select its best final Z_{All} configuration within this common set and evaluate that exact configuration at every target size. Normalised regret is computed relative to the target's best and worst configurations in the same shared set. Restricting both selection and evaluation to this common support avoids treating an unobserved target configuration as transferable. It also means that the 1.7B source uses its core-restricted optimum, $(4 \times 10^{-6}, 0.01)$, rather than its full-grid optimum at learning rate 8×10^{-6}.

Figure 27 gives both the smaller-to-larger transfers and the complete directed matrix. Model-size distance is not predictive: 0.6B→1.7B has zero regret, while 1.7B→4B has regret 0.997 despite being a smaller scale change; 0.6B→8B has regret only 0.287. Transfer is also asymmetric. The configuration selected at 8B incurs regret 0.028 at 4B, whereas the 4B selection incurs regret 0.222 at 8B. Similarly, 4B→14B has regret 0.784, compared with 0.251 in the reverse direction. These results rule out parameter-count distance as a sufficient criterion for deciding whether a configuration should be reused.

D.5.3 Cross-family

We next compare Z_{All} rankings over the same nine configurations between the nearest available Qwen3 and Llama sizes. Scores are standardized separately within each model's grid, so the analysis compares configuration orderings rather than absolute score levels. Figure 28 shows positive agreement for Qwen3-0.6B and Llama 3.2-1B ($\rho = 0.67$), but negative agreement for Qwen3-4B and Llama 3.2-3B ($\rho = -0.67$) and for Qwen3-8B and Llama 3.1-8B ($\rho = -0.40$). Similar parameter counts therefore do not ensure that the two families prefer the same DPO configuration.

This comparison is exploratory. It contains only three model pairs, two pairs are not exactly size-matched, and the 8B comparison uses Llama 3.1 whereas the smaller Llama models use Llama 3.2.

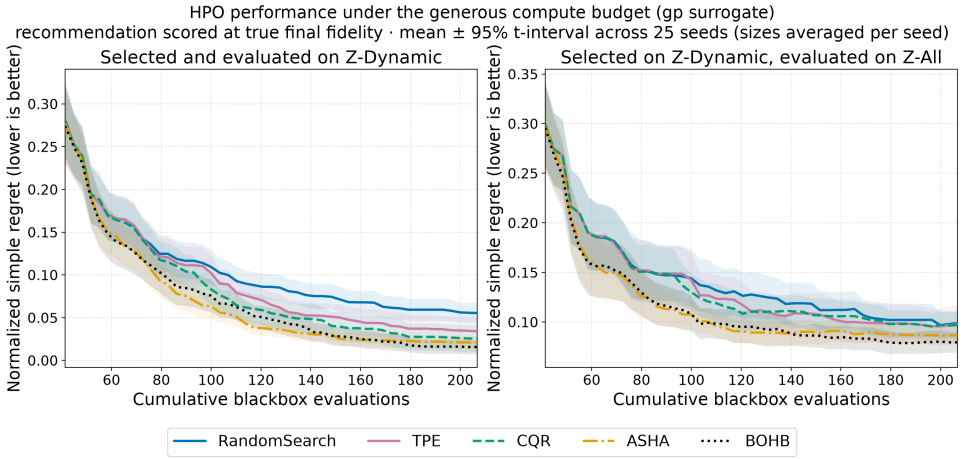

Figure 22: **Recommendation regret under the $8\times$ budget with the Gaussian-process surrogate.** All methods select on Z_{Dynamic}. The left panel evaluates the recommendation on Z_{Dynamic}; the right evaluates the same recommendation on Z_{All}. Curves average the five Qwen3 sizes within each seed and show the mean and 95% t-interval across 25 seeds.

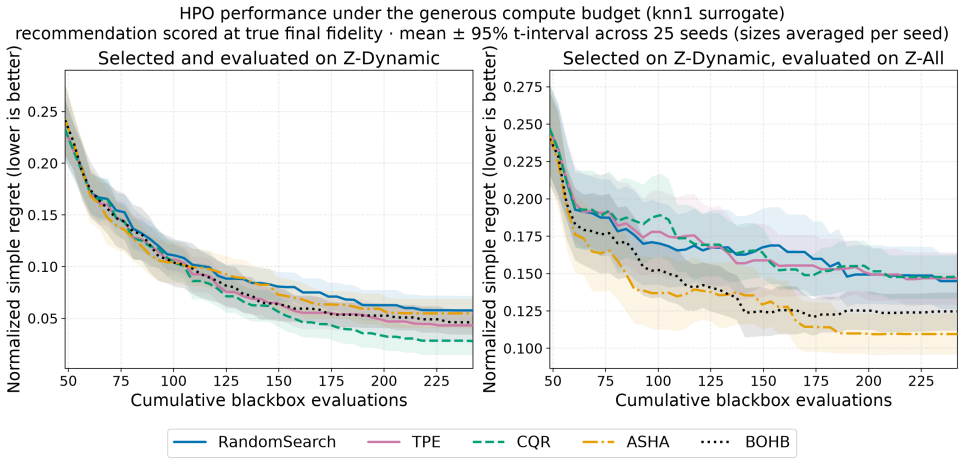

Figure 23: **Recommendation regret under the $8\times$ budget with the one-nearest-neighbour surrogate.** All methods select on Z_{Dynamic}. The left panel evaluates the recommendation on Z_{Dynamic}; the right evaluates the same recommendation on Z_{All}. Curves average the five Qwen3 sizes within each seed and show the mean and 95% t-interval across 25 seeds. The one-nearest-neighbour surface is piecewise constant between measured configurations.

The results establish that cross-family ranking reversal occurs in the tested settings, but they do not estimate how frequently it should occur across model families.

E Preliminary Validation Loss Directed BO Runs

Before constructing the primary grids, we ran a separate sequential search over DPO learning rate and β at each Qwen3 size, using held-out DPO loss as the objective. These experiments train every proposed configuration and are distinct from the offline HPO simulations in Section 4.3: here, BO chooses continuous hyperparameters by minimizing validation loss, whereas the offline experiments compare search algorithms using downstream evaluation aggregates. The runs also start from SFT checkpoints trained with full-sequence loss, rather than the assistant-only-loss checkpoints used by

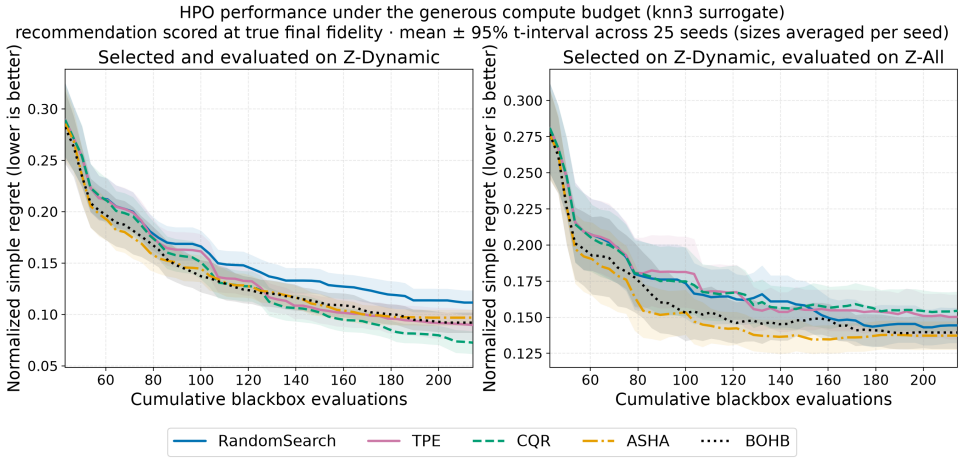

Figure 24: **Recommendation regret under the $8\times$ budget with distance-weighted three-nearest-neighbour interpolation.** All methods select on Z_{Dynamic}. The left panel evaluates the recommendation on Z_{Dynamic}; the right evaluates the same recommendation on Z_{All}. Curves average the five Qwen3 sizes within each seed and show the mean and 95% t-interval across 25 seeds.

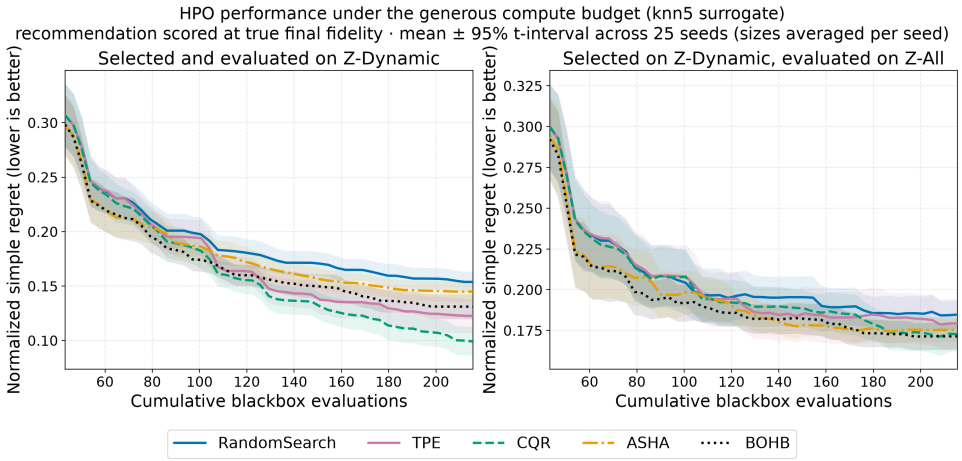

Figure 25: **Recommendation regret under the $8\times$ budget with distance-weighted five-nearest-neighbour interpolation.** All methods select on Z_{Dynamic}. The left panel evaluates the recommendation on Z_{Dynamic}; the right evaluates the same recommendation on Z_{All}. Curves average the five Qwen3 sizes within each seed and show the mean and 95% t-interval across 25 seeds.

the primary grids. We therefore report them as preliminary evidence and do not pool their configurations with the main analyses.

Search protocol. We use Optuna's tree-structured Parzen estimator (TPE) (Akiba et al., 2019; Bergstra et al., 2011). TPE fits separate Parzen density models to configurations with relatively low and high observed loss and proposes points using its expected-improvement density ratio. Each model-size study begins with five startup trials, then continues sequentially to a fixed budget of 20 trials. No pruning or improvement-based stopping rule is used. The objective is the DPO loss on the same fixed 1,300-pair validation set used in Section 4.1; it is evaluated every 100 training steps. Table 20 gives the remaining search settings.

Validation loss remains unreliable for chat-based selection. Table 21 evaluates the completed trial with the lowest validation loss at each size. Of the 100 proposed trials, 93 have both validation

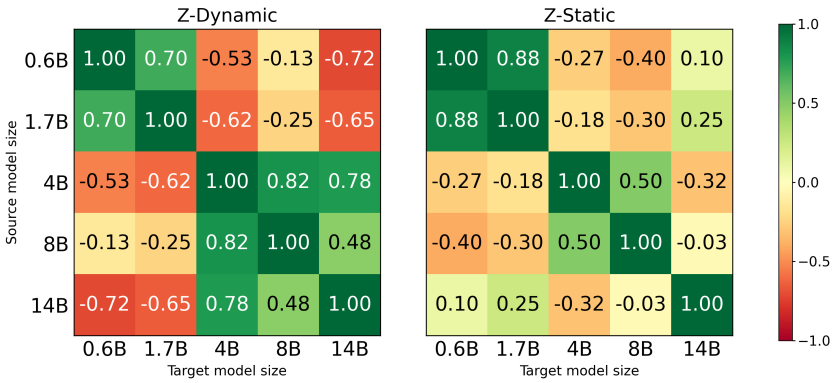

Figure 26: **Cross-scale configuration-ranking agreement by evaluation aggregate.** Each cell reports Spearman's ρ between rankings of the nine configurations in the shared 3×3 learning-rate–β grid. The left panel uses Z_{Dynamic} and the right panel uses Z_{Static}. Neither aggregate exhibits a consistent decline in agreement with model-size distance.

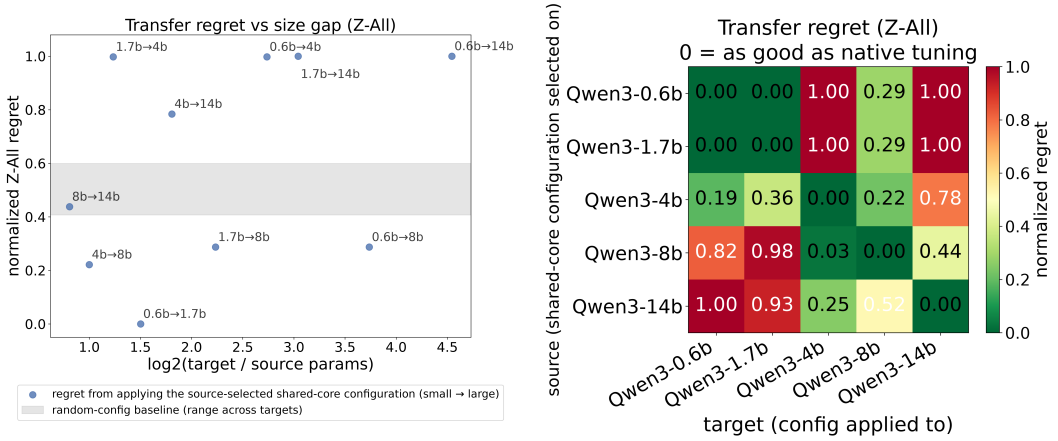

Figure 27: **Cross-scale transfer regret is neither determined by model-size distance nor symmetric. Left:** Normalised Z_{All} regret for smaller-to-larger transfers as a function of the $\log_2$ parameter-count ratio. The shaded region gives the range of expected regret from choosing a shared-core configuration uniformly at random across target sizes. **Right:** The complete matrix of directed transfers. Each cell applies the source model's best Z_{All} configuration from the shared 3×3 grid to the target model. Regret of zero denotes the target's best shared-core configuration and one its worst. Low and near-maximal regret occur at similar scale distances, while reversing the transfer direction can substantially change regret.

loss and all three downstream aggregates in the export; trials missing either quantity are excluded. Regret is normalised within the complete preliminary trials for that model, with zero denoting the best downstream score and one the worst. The selected learning rate rises from 0.6B to 1.7B and then decreases from 4.69×10^{-5} at 1.7B to 4.38×10^{-6} at 14B, a $10.7\times$ reduction that resembles the non-monotonic shape observed in the primary grids. The selected β has no monotonic size trend but remains in the lower part of its search domain: it is 0.011–0.064 from 1.7B to 14B, while the 0.6B selection, 0.053, is close to that run's lower bound of 0.05. We therefore interpret the preliminary runs as additional evidence that larger models above 1.7B favour lower learning rates under this training recipe, while supporting only the weaker conclusion that the preferred β remains low rather than constant across scale.

The downstream quality of the minimum-loss trial depends strongly on the evaluation objective. Mean regret across sizes is 0.21 for Z_{All} and 0.16 for Z_{Static}, but 0.39 for Z_{Dynamic}. At 1.7B, the minimum-loss trial ranks 16th of 18 under Z_{Dynamic} and incurs 0.89 regret; at 4B, it ranks 16th of

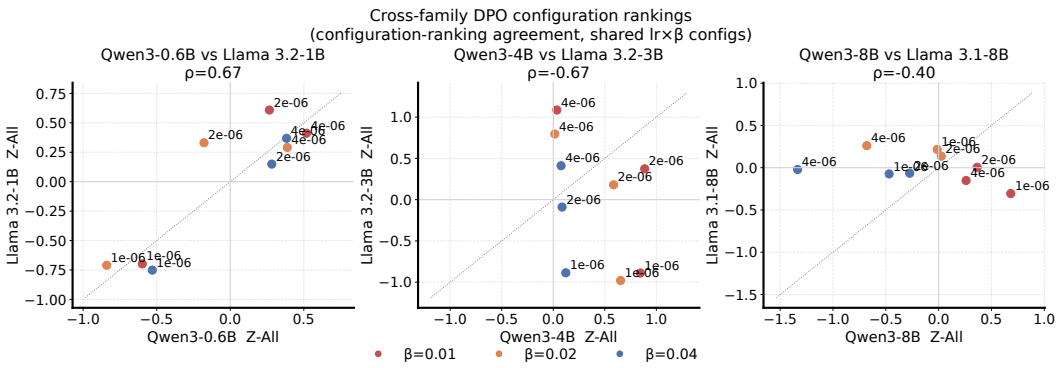

Figure 28: **Configuration rankings do not transfer consistently across model families.** Each panel compares final Z_{All} scores for the nine shared $(\text{learning rate}, \beta)$ configurations. Points are coloured by β and annotated with learning rate; the dotted diagonal is a visual reference. Agreement is positive for the smallest pair but negative for the two larger comparisons.

Table 20: **Settings for the preliminary validation loss directed searches.** The different β domain at 0.6B reflects the bounds used in these preliminary runs.

Setting	Qwen3-0.6B	Qwen3-1.7B–14B
Learning rate	$[10^{-6}, 5\times10^{-5}]$, log-uniform	
β	$[0.05, 1]$, uniform	$[0.01, 1]$, log-uniform
Sampler	Optuna TPE; seed 8	
Objective	minimize held-out DPO loss	
Startup / total trials	5 / 20 per model size	
Stopping and pruning	fixed trial budget; no pruner	
SFT loss	full-sequence loss	

19 with 0.48 regret. Conversely, it is optimal under Z_{Static} at 4B and 14B. Thus, adaptive sampling does not make validation loss a uniformly reliable downstream objective: in these runs it aligns more closely with static performance than with chat-based performance.

Rank correlations give the same qualification. We compute Spearman correlation between negative validation loss, so that larger values are better, and each downstream aggregate. Averaged across sizes, the correlations are 0.54 with Z_{All}, 0.16 with Z_{Dynamic}, and 0.67 with Z_{Static}. For Z_{Dynamic}, the correlation is negative at 1.7B (-0.45) and 4B (-0.20), showing that lower validation loss can favour the opposite configuration ordering from chat-based evaluation. Because these searches use a different SFT loss, search domain, and adaptively sampled set of configurations, their absolute regrets are not directly comparable to those of the primary grids in Section 4.1.

Benchmark agreement persists on the adaptively sampled configurations. Although TPE queries a different, continuous set of configurations at each size, the qualitative agreement pattern

Table 21: **Downstream quality of the minimum validation loss TPE trial.** n is the number of trials with complete loss and downstream results. Aggregate columns report normalised regret followed by the selected trial's downstream rank in parentheses; lower values are better for both quantities.

Model	n	Learning rate	β	Z_{All}	Z_{Dynamic}	Z_{Static}
Qwen3-0.6B	20	1.83×10^{-5}	0.0526	0.20 (4)	0.26 (11)	0.12 (5)
Qwen3-1.7B	18	4.69×10^{-5}	0.0109	0.63 (15)	0.89 (16)	0.36 (10)
Qwen3-4B	19	2.60×10^{-5}	0.0164	0.13 (5)	0.48 (16)	0.00 (1)
Qwen3-8B	16	5.39×10^{-6}	0.0638	0.11 (8)	0.15 (3)	0.29 (9)
Qwen3-14B	20	4.38×10^{-6}	0.0165	0.00 (1)	0.17 (5)	0.00 (1)
Mean regret	—	—	—	0.21	0.39	0.16

Table 22: **Benchmark-ranking agreement within the preliminary TPE runs.** Entries give the mean pairwise Spearman correlation ρ and its population standard deviation across benchmark pairs. P is the number of benchmark pairs contributing to each summary.

	Chat benchmarks		Static benchmarks	
Model	$\rho \pm \sigma$	P	$\rho \pm \sigma$	P
Qwen3-0.6B	0.49 ± 0.19	6	0.15 ± 0.35	15
Qwen3-1.7B	0.52 ± 0.42	6	0.17 ± 0.31	15
Qwen3-4B	0.64 ± 0.33	6	0.30 ± 0.53	15
Qwen3-8B	0.82 ± 0.08	3	0.36 ± 0.34	15
Qwen3-14B	0.85 ± 0.05	6	0.08 ± 0.45	15

from Section 4.1 remains visible. Table 22 reports the mean and population standard deviation of the pairwise Spearman correlations within each evaluation group. Mean chat-benchmark agreement is positive at every size and increases from 0.49 at 0.6B to 0.85 at 14B. Static agreement is lower throughout, ranging from 0.08 to 0.36, with substantially larger variation across benchmark pairs. Because IFEval was not available for enough preliminary trials to estimate its pairwise correlations, the static summary uses the other six static benchmarks. At 8B, the chat summary uses the three available pairs among Arena-Hard, MT-Bench, and Elo; the other sizes use all six chat-benchmark pairs. Pairwise correlations are computed over the configurations for which both constituent evaluations are available, yielding 16–20 configurations depending on model size and benchmark pair.

Together, these results show that the contrast between chat and static rankings is not confined to the rectangular grids. They also explain why a validation loss directed search can appear successful under some static evaluations while selecting a poor chat configuration: the two evaluation groups induce different configuration rankings, and validation loss aligns much more strongly with the static group in this preliminary training recipe.