EFFICIENT UNTYING OF WEIGHT-TIED LANGUAGE MODELS VIA ASYMMETRIC GEOMETRIC DECOUPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Weight tying reduces vocabulary-layer parameters, but forces input embeddings and output unembeddings to share one vocabulary geometry. Modern language models increasingly untie these matrices at larger scales, raising the question of whether parameter-constrained models can relax tying without paying the full cost of an independent vocabulary matrix. We diagnose this trade-off through training dynamics and embedding geometry across public model families. Tied shared matrices are consistently closer to untied output unembeddings than to input embeddings; output spaces also show stronger cross-model cosine alignment and stronger mean/rank-1 spectral structure. These patterns match dense softmax gradients on the output side versus sparse lookup gradients on the input side, suggesting that hard tying makes the shared matrix output-dominated and constrains independent input-side geometry. We propose **Asymmetric Geometric Decoupling (AGD)**, which keeps a shared vocabulary matrix on the output side while adding small structured freedom to the input side through bias correction and hidden-dimensional low-rank deformation. Input-side AGD improves tied baselines in MiniMind and FineWeb-Edu pretraining, including a 0.6B configuration trained from scratch on 20B tokens, with minimal parameter overhead. Its gains persist in held-out assistant-token likelihood after full-parameter SmolTalk fine-tuning.

1 INTRODUCTION

Weight Tying (WT), sharing the input embedding matrix and the output unembedding matrix, is a classic parameter-efficiency design in neural language models. Press & Wolf (2017) and Inan et al. (2017) show that output classifier weights themselves can serve as effective word representations, allowing input embeddings and output unembeddings to share the same vocabulary matrix while reducing parameters and preserving or even improving language modeling performance. This design also embeds a strong structural premise: token semantic encoding on the input side and vocabulary prediction on the output side can be served by the same lexical geometry.

This classic design was inherited in the modern LLM era, but it did not become a universal default. Recent public models show a scale-dependent split: smaller models often keep tied designs to save

Table 1: **Parameter share of a single embedding matrix in Qwen3 dense models.** A single Emb denotes $V \times d$. For untied models, input embeddings and output unembeddings each have a matrix of the same size, so the "actual total vocabulary-matrix share" is twice the single-Emb share.

Model	tied?	d	Single Emb Params	Total Params	Single Emb / Total	Actual Vocab Matrices / Total
Qwen3-0.6B	tied	1024	155.6M	596.0M	26.10%	26.10%
Qwen3-1.7B	tied	2048	311.2M	1.721B	18.08%	18.08%
Qwen3-4B	tied	2560	389.0M	4.022B	9.67%	9.67%
Qwen3-8B	untied	4096	622.3M	8.191B	7.60%	15.20%
Qwen3-14B	untied	5120	777.9M	14.768B	5.27%	10.53%
Qwen3-32B	untied	5120	777.9M	32.762B	2.37%	4.75%

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.

vocabulary-layer parameters, whereas larger or MoE models more often adopt untied designs to gain greater representational freedom (full inventory in Appendix B.1). Table 1 illustrates why this split is natural: in the Qwen3 dense series, a single vocabulary matrix accounts for 26.10% of Qwen3-0.6B but only 2.37% of Qwen3-32B. Full untying is therefore expensive for small models but increasingly affordable at larger scales.

This trend raises the central question of this paper: **can parameter-constrained language models relax the geometry imposed by weight tying without paying the full cost of an independent vocabulary matrix?** We argue that the cost of hard tying is not merely one fewer matrix, but the assumption that input embeddings and output unembeddings should share the same lexical geometry.

Our diagnostics show that this assumption is asymmetric in practice. Across public model families, tied shared matrices are more output-like than input-like: they are closer to untied output unembeddings, output spaces show stronger cross-model cosine alignment, and output/tied matrices have stronger mean/rank-1 spectral structure. This geometry is consistent with the underlying training dynamics. Input embeddings receive sparse lookup-path gradients, whereas output unembeddings receive dense full-vocabulary softmax gradients; after tying, both signals update the same matrix, making the shared vocabulary space more easily dominated by output-side objectives.

Based on this diagnosis, we propose **Asymmetric Geometric Decoupling (AGD)**, a family of lightweight embedding-head untying operations between hard tying and full untying. AGD keeps a shared vocabulary matrix for the output side while adding small structured freedom to the input side through bias correction and hidden-dimensional low-rank deformation. Input-side AGD improves tied baselines in MiniMind and FineWeb-Edu pretraining, including a 0.6B configuration trained from scratch on 20B tokens; both bias and affine variants retain their gains after full-parameter SmolTalk SFT, measured by held-out assistant-token likelihood.

Our main contributions are summarized as follows:

(1) Diagnosis of hard tying. We provide cross-model geometric and spectral evidence that tied vocabulary matrices are output-like rather than neutral compromises between input and output roles.

(2) Mechanistic explanation. We connect this output-like geometry to dense-output versus sparse-input gradient asymmetry, motivating independent geometric freedom on the input side.

(3) Efficient untying. We introduce input-side-prioritized AGD and validate that bias correction and an affine low-rank AGD variant improve tied small-model pretraining with minimal parameter overhead.

2 Related Work

Weight tying and its implicit assumption Weight tying was originally introduced to reduce vocabulary-layer redundancy between input embeddings and output classifiers in neural language models. Press & Wolf (2017) show that output unembedding weights themselves can serve as effective word representations and observe that tied shared matrices evolve closer to output-side word representations in untied models; Inan et al. (2017) explain the rationale for sharing input/output vocabulary matrices from the loss-function and word-embedding-space classification perspectives.

Table 2: Models used in the empirical analysis.

Model Family	Coverage (Params)	Architecture	Count
Qwen-2.5	0.5–72B	Dense	14
Qwen-3/3.5/3.6	0.6B–397B	Dense / MoE	28
Llama-3.x	1–70B	Dense	8
Gemma-2/3	1–27B	Dense	12
DeepSeek	V2/V3/R1	Dense / MoE	16
GLM	GLM-5/5.1	MoE	2
MiniMax	M2.5/M2.7	MoE	2
Total	0.5B–754B	Dense / MoE	82

We revisit this assumption: in modern language models, is a shared matrix still suitable for both input semantic encoding and output vocabulary prediction?

Geometric constraints on input/output vocabulary spaces Bertolotti & Cazzola (2024) relate tying to compatibility between semantic and conditional equivalence. Gao et al. (2019) study narrow-cone degeneration in weight-tied generative models; Biś et al. (2021) instead explain anisotropy through common-direction drift under softmax updates and show that mean subtraction restores isotropy. Machina & Mercer (2024) associate untied designs with higher isotropy in some large models. We compare untied input E, output U, and tied shared W directly: GCorr and spectral diagnostics place W closer to U and reveal stronger mean/rank-1 structure in U than in E. This asymmetry motivates lightweight input-side decoupling during pretraining.

Relaxing hard tying and embedding/head decoupling Prior work has attempted to relax hard tying through structured output layers, intermediate prediction spaces, or capacity reallocation. Pappas et al. (2018) model structured sharing between input and output embeddings in neural machine translation; Gulordava et al. (2018) decouple hidden-state space from word-embedding prediction space while preserving tying; REMBERT (Chung et al., 2021) reallocates capacity between input embeddings and output vocabulary representations through re-embedding. These methods show that vocabulary entry and exit need not be strictly identical, but they usually require additional mappings, output-layer changes, or capacity redistribution. AGD has a different goal: it preserves the tied vocabulary matrix as a parameter-efficient base and adds only a small amount of structured freedom on the input side.

Training dynamics and cross-model geometry Training dynamics also support input/output asymmetry. Press & Wolf (2017) already note that input embeddings receive sparse updates only when tokens appear, whereas output unembeddings receive dense full-vocabulary updates at every step through softmax. We connect this dense-output/sparse-input imbalance to geometric diagnostics in modern models, explaining why tied matrices exhibit output-like structure. On the other hand, cross-model studies such as Lee et al. (2025) find shared geometric structure in unembedding spaces, supporting our use of embedding/unembedding geometry as the central object for revisiting weight tying.

3 DIAGNOSTIC ANALYSIS: THE GEOMETRY OF WEIGHT TYING

3.1 EXPERIMENTAL SETUP

We construct a diagnostic set from major public language-model families, covering dense and MoE architectures, tied and untied designs, and model sizes from 0.5B to 754B parameters (full inventory in Table 2). The analysis consists of 35 Base–Instruct comparisons (30 in the comparable subset), 110 within-family model-series comparisons comprising 37 tied–untied, 48 untied–untied, and 25 tied–tied pairs, 26 additional cross-series untied–untied comparisons, and static vocabulary-matrix spectral analysis over 82 comparable public models.

Diagnostic protocol Let the input embedding matrix be $E \in \mathbb{R}^{V\times d}$ and the output unembedding matrix be $U \in \mathbb{R}^{V\times d}$; under tied parameterization, both share the same matrix W. **GCorr** (Global Correlation) measures agreement between token–token geometries induced by two aligned vocabulary matrices: for $X, Y \in \mathbb{R}^{n\times d}$ and a pairwise metric μ, it is the Pearson correlation between $[\mu(X_{i_k}, X_{j_k})]_k$ and $[\mu(Y_{i_k}, Y_{j_k})]_k$ on shared token pairs (formal definition and sampling protocol in Appendix A.3). We report cosine, Euclidean, and squared-Euclidean variants in the main diagnostic comparisons, and complement GCorr with static spectral analysis of the mean axis, the leading singular component, and raw rank-1 energy. Token alignment, bootstrap, and spectral subset construction are given in Appendices A.1 and B.2.

3.2 GEOMETRIC ASYMMETRY UNDER WEIGHT TYING

We summarize the diagnostics into three points: tied shared matrices are closer to output unembeddings; untied output spaces have stronger cross-model cosine agreement than input spaces; and both output unembeddings and tied shared matrices exhibit a stronger mean/rank-1 spectral axis.

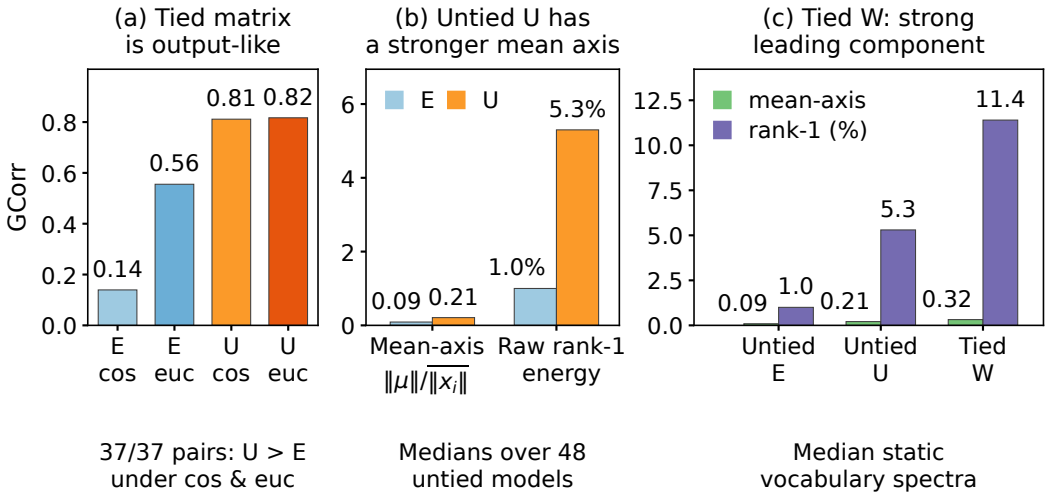


Figure 1: **Three core pieces of evidence from the weight-tying geometry diagnostic. (a)** In 37 within-family tied–untied GCorr comparisons, the tied shared matrix is more similar to the untied output unembedding than to the untied input embedding in all 37/37 pairs under both cosine and Euclidean metrics. **(b)** Among 48 comparable untied models, the U-side static spectrum has a much stronger mean/rank-1 axis than the E side: the median $\|\mu\|/\operatorname{mean}\|x_i\|$ is about 2.3× larger, and raw rank-1 energy is about 5.2× larger. **(c)** For 34 comparable tied models, the shared matrices exhibit strong leading-component concentration: median raw rank-1 energy is 11.4%, compared with 5.3% for untied U and 1.0% for untied E.

Insight 1: tied shared matrices are closer to output unembeddings In within-family tied–untied comparisons, geometric similarity on the input and output sides shows strong asymmetry. Averaged over 37 tied–untied pairs, output-side similarity is substantially higher than input-side similarity: $U_{\cos} = 0.8114$, $U_{\text{euc}} = 0.8169$, versus $E_{\cos} = 0.1395$, $E_{\text{euc}} = 0.5553$. Pairwise inspection shows that all 37/37 pairs satisfy both $U_{\cos} > E_{\cos}$ and $U_{\text{euc}} > E_{\text{euc}}$. This indicates that when a tied model is compared with an untied model from the same family, the tied shared matrix is closer to the output unembedding in the untied model than to its input embedding.

The result also holds under squared-Euclidean GCorr (37/37), with mean U-side similarity 0.7982 versus 0.5124 on the E side. The smallest pairwise U-minus-E gaps are 0.5167, 0.1221, and 0.1268 for cosine, Euclidean, and squared-Euclidean GCorr, respectively (Table 17). Supplementary Base–Instruct comparisons describe within-model geometric stability in Appendix B.3.1.

Insight 2: output spaces share stronger cross-model cosine geometry To examine output-side commonality independently of hard sharing, we first compare the 48 within-series pairs in which both models are untied. Mean cosine GCorr is higher for U–U than for E–E, with U-side gains in 35/48 pairs (Table 3). We then extend the comparison to 26 cross-series untied pairs spanning Qwen2.5, Qwen3, Qwen3.5 dense, and Llama, with tokens aligned by string. Output-side cosine GCorr is higher in all 26/26 pairs, averaging 0.6690 versus 0.3095 on the input side. This extends the evidence for shared output-side angular structure beyond comparisons within a single model series.

Table 3: Cosine GCorr between untied models. Values are means over model pairs; the final column counts pairs with higher output-side similarity.

Comparison	Pairs	E–E	U–U	U > E
Within series	48	0.6095	0.8907	35/48
Across series	26	0.3095	0.6690	26/26

Euclidean GCorr captures a different relation: within-series mean E–E and U–U values are 0.9446 and 0.9208, respectively. The grouping and all three metrics are reported in Appendix B.3.2; the 25 tied–tied pairs form an $E = U$ reference group.

Insight 3: output unembeddings have a stronger mean/rank-1 spectral axis The static spectral analysis further clarifies what distinguishes U from E. Among 48 comparable untied models, the U-side raw spectrum is more concentrated almost entirely in the leading component: the median $\|\mu\|/\operatorname{mean}\|x_i\|$ is 0.21 for U versus 0.09 for E (about 2.3×), and the median raw rank-1 energy fraction is 5.3% for U versus 1.0% for E (about 5.2×), with U rank-1 energy exceeding E in all 48/48 cases.

Crucially, this is not broad low-rank concentration across the first several principal directions. Incremental rank2–5 and rank6–10 energy are lower on U than on E in all 48/48 comparable untied models. The U-side difference is therefore best described as a *mean-axis / rank-1 effect*: the vocabulary classifier matrix carries a stronger global mean component, and its leading singular direction is almost aligned with this mean direction. Consistent with the centering analysis of Biś et al. (2021), removing the mean reduces median rank-1 energy from 5.264% to 1.111% for U and from 1.007% to 0.616% for E. The ratio of these medians falls from 5.23 to 1.80; U remains higher in 43/48 models. Normalized participation ratios further show that, after centering, median U-side spectral energy is more broadly distributed than E-side energy (Table 14), distinguishing the strong mean-axis effect from the shape of the remaining spectrum.

Tied matrices exhibit the same leading-component pattern. Across the 34-model tied subset, median raw rank-1 energy is 11.4% and median $\|\mu\|/\operatorname{mean}\|x_i\|$ is 0.32. All 34 exceed the untied E median in raw rank-1 energy, and 26 exceed the untied U median. These mean-axis and leading-component features reinforce the output-like geometry identified by GCorr.

Training dynamics and design implication Gradient-level observations provide a mechanistic explanation for the geometric results above. Output unembeddings receive dense full-vocabulary updates at every position through the softmax classification objective, whereas input embeddings receive sparse updates only through the lookup path of currently observed tokens; under tying, the two gradient paths are merged onto the same W, so the shared matrix serves the input role while evolving long-term toward output unembedding objectives. The full dense–sparse derivation is given in Appendix A.2.

These observations motivate an asymmetric relaxation: retain the shared output matrix and give the input mapping its own structured freedom. A learned translation directly addresses the mean-axis distinction; low-rank and affine variants test the value of additional input-side flexibility. This leads to AGD in the next section.

4 Method: Asymmetric Geometric Decoupling

The diagnosis in Section 3.2 shows that standard weight tying is not neutral sharing between the input and output sides: the shared vocabulary matrix is geometrically closer to output unembeddings and inherits their stronger mean/rank-1 structure. Based on this observation, we study **Asymmetric Geometric Decoupling (AGD)**: preserving a shared vocabulary matrix as a parameter-efficient base while adding a small amount of structured freedom only on the input side, allowing input embeddings to deviate from the output-dominated shared space.

4.1 From Hard Tying to Input-Side AGD

Let the vocabulary size be V and the hidden dimension be d. A standard tied language model uses the same matrix $W \in \mathbb{R}^{V\times d}$ for both input embedding and output unembedding:

$$e_t = W[x_t], \qquad z_t = Wh_t + b, \tag{1}$$

where h_t is the final-layer hidden state and b is the output bias. Equivalently, standard tying imposes the hard constraint

$$E_{\text{eff}} = U_{\text{eff}} = W. \tag{2}$$

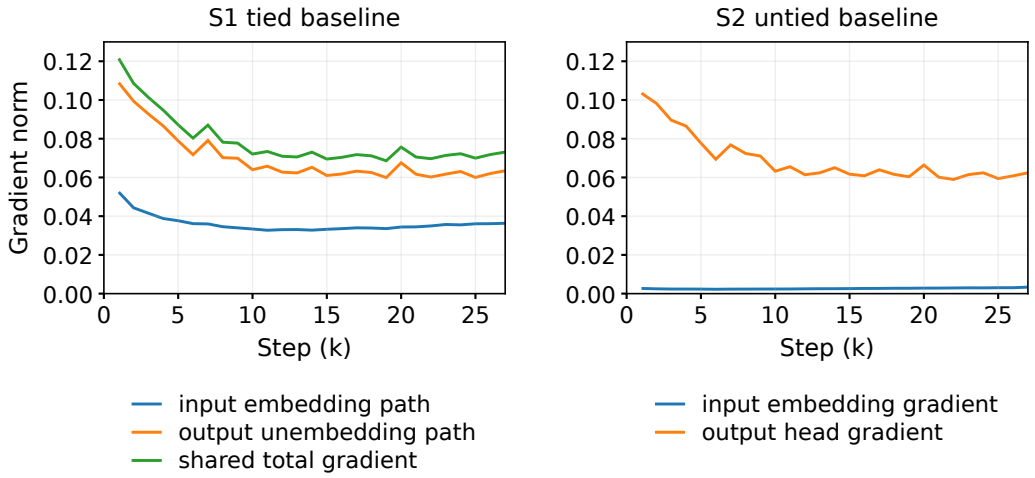


Figure 2: **Input-side and output-side gradient norms in FineWeb-Edu/GPT-2 pretraining.** Curves are averaged over three seeds. In the S1 tied baseline, gradients on the shared matrix are largely elevated by the output-side path; in the S2 untied baseline, input embedding gradients are much smaller than output unembedding gradients. This shows that dense-output/sparse-input update asymmetry pushes tied matrices toward output-side geometry.

AGD relaxes this hard equality rather than introducing a full new vocabulary matrix. We keep

$$U_{\text{eff}} = W, \tag{3}$$

so the output side continues to use the shared matrix and dense full-vocabulary classification gradients; and we set

$$E_{\text{eff}} = \phi_{\text{in}}(W; \theta), \tag{4}$$

where ϕ_{in} is an input-side transformation with only a small number of parameters θ. Since the tied matrix already behaves output-like, limited extra parameters should be allocated to the constrained input side rather than further strengthening the output side. The full dense-output/sparse-input gradient derivation is given in Appendix A.2.

The shared matrix W remains trainable through both the input and output paths. Asymmetry refers to where the additional parameters act: they transform input representations while the output projection continues to use W.

4.2 Four Input-Side Variants in the Main Text

We study four input-side variants: Input-Bias (S3), Input-Add-LoRA (S4, additive LoRA), Input-Mult-LR (S5, multiplicative low-rank), and Input-Affine (S6). The S identifiers index the full S1–S12 ablation grid in Appendix B.4, including Tied (S1), Untied (S2), and output-side controls. All four keep $U_{\text{eff}} = W$ and differ only in ϕ_{in}. Multiplicative variants use $W(I_d + sPQ)$ with $s = 1$ in the main experiments.

During pretraining, the backbone, shared vocabulary matrix, and added modules are trained jointly. Bias vectors and one factor of each low-rank branch are initialized to zero, so each variant starts with $E_{\text{eff}} = U_{\text{eff}} = W$; the other factor is sampled from $\mathcal{N}(0, 0.02^2)$.

Input-Bias (S3) A learnable vector $\beta \in \mathbb{R}^d$ translates all input embeddings by the same amount. The translation is learned from the language-modeling objective rather than fixed to the vocabulary mean. This is the minimal response to the mean/rank-1 diagnosis: its vocabulary-side coefficients are fixed to one, so it adds no token-specific memory.

Table 4: Four input-side AGD variants used in the main text. All variants keep $U_{\text{eff}} = W$.

Variant	Effective Input Matrix E_{eff}	Added
Input-Bias (S3)	$W - \mathbf{1}\beta^\top$	d
Input-Add-LoRA (S4)	$W + A_{\text{in}}B_{\text{in}}$	$r(V+d)$
Input-Mult-LR (S5)	$W(I_d + sP_{\text{in}}Q_{\text{in}})$	$2dr$
Input-Affine (S6)	$W(I_d + sP_{\text{in}}Q_{\text{in}}) - \mathbf{1}\beta^\top$	$d + 2dr$

Input-Add-LoRA (S4) With $A_{\text{in}} \in \mathbb{R}^{V\times r}$ and $B_{\text{in}} \in \mathbb{R}^{r\times d}$, the additive residual assigns different low-rank coefficients to each token. Its cost grows with vocabulary size, testing whether token-specific freedom is worth the additional parameters.

Input-Mult-LR (S5) With $P_{\text{in}} \in \mathbb{R}^{d\times r}$ and $Q_{\text{in}} \in \mathbb{R}^{r\times d}$, the multiplicative variant applies a shared low-rank transformation in hidden coordinates. Its cost is independent of vocabulary size. The fixed scalar s controls its strength; a related multiplicative form appears in LoRMA (Bihany et al., 2025).

Input-Affine (S6) Input-Affine combines the global translation of Input-Bias with the hidden-dimensional transformation of Input-Mult-LR. The linear part is $I_d + sP_{\text{in}}Q_{\text{in}}$, and the translation is $-\beta$; both act on input lookup vectors, while the output projection uses W. Its $d + 2dr$ parameters jointly provide translation and low-rank coordinate deformation without growing with vocabulary size.

5 Experimental Evaluation

We evaluate AGD against tied baselines in three pretraining configurations, including a fixed-random split replication, and test whether its gains persist after full-parameter supervised fine-tuning (SFT).

5.1 Experimental Setup

The two smaller-model settings use the same lightweight dense decoder-only Transformer architecture with hidden dimension $d = 768$ and output bias enabled (`lm_head_bias`), aligned with the Qwen3 dense model design (Yang et al., 2025). Low-rank variants uniformly use rank $r = 32$. We set up two pretraining data and tokenizer configurations: the first uses MiniMind (Gong, 2025) data and the MiniMind tokenizer, training all twelve embedding-head variants; the second uses FineWeb-Edu (Penedo et al., 2024) data and the GPT-2 tokenizer (Radford et al., 2019), training key input-side and output-side lightweight untying variants on roughly 6 billion tokens of packed corpus. All runs use three random seeds (42, 123, 2026) and report mean validation cross-entropy loss and PPL; detailed data splits, training hyperparameters, parameter overhead, gradient logging, and log paths are given in Appendix B.4.

We use two data partitioning schemes: a sequential split (the first 99% for training and the last 1% for validation) for the MiniMind and first FineWeb-Edu/GPT-2 groups, and a fixed-random 1% validation split for the GPT-2 replication and Qwen3-0.6B group. The Untied control in Table 5 uses independent input/output matrices initialized to identical values.

We further train S1, Input-Bias, and Input-Affine from random initialization on approximately 20B packed FineWeb-Edu tokens using a Qwen3-0.6B-aligned configuration and tokenizer: 28 layers, $d = 1024$, 16 attention heads, 8 KV heads, head dimension 128, and intermediate size 3072. The tied baseline has approximately 595.8M parameters. All three seeds use the same fixed-random 1% validation split. Paired effects are computed against same-seed S1 and reported as mean $\pm$ sample SD; additional setup and per-seed results are in Appendix C.

5.2 Main Results

Parameter efficiency relative to full untying Table 5 compares AGD with Untied under the same FineWeb-Edu/GPT-2 data split. Input-Bias and Input-Affine recover 35.8% and 61.5% of its mean

Table 5: Three-seed validation results. Seed Δ PPL reports paired differences from S1 (mean ± sample SD); PPL is $\exp(\text{mean loss})$; PPL Δ is the mean percentage change relative to S1; PPL Δ/Param Δ follows the "reduction-is-positive" convention. Wins counts seeds with lower loss than S1. Full ablations are in the appendix.

Variant	Param Δ	Mean Loss	Mean PPL	Seed Δ PPL vs S1	PPL Δ	PPL Δ/ Param Δ	Wins vs S1
FineWeb-Edu/GPT-2 (with Untied control)							
Tied (S1)	0.0000%	3.237896	25.480064	0.000 ± 0.000	0.00%	–	–
Untied (S2)	39.5284%	3.225541	25.167186	-0.313 ± 0.047	-1.23%	0.03	3/3
Input-Affine (S6)	0.0511%	3.230316	25.287638	-0.192 ± 0.071	-0.76%	14.78	3/3
Input-Bias (S3)	0.0008%	3.233494	25.368130	-0.112 ± 0.082	-0.44%	549.13	3/3
Input-Add-LoRA (S4)	1.6722%	3.235201	25.411471	-0.069 ± 0.069	-0.27%	0.16	3/3
Input-Mult-LR (S5)	0.0503%	3.235579	25.421078	-0.059 ± 0.080	-0.23%	4.60	2/3
MiniMind							
Tied (S1)	0.0000%	1.914962	6.786681	0.000 ± 0.000	0.00%	–	–
Input-Bias (S3)	0.0012%	**1.903291**	**6.707934**	$\mathbf{-0.079 \pm 0.040}$	**-1.16%**	**966.93**	3/3
Input-Affine (S6)	0.0781%	1.906849	6.731843	-0.055 ± 0.036	-0.81%	10.35	3/3
Input-Add-LoRA (S4)	0.3589%	1.912134	6.767515	-0.019 ± 0.044	-0.28%	0.79	2/3
Input-Mult-LR (S5)	0.0769%	1.912824	6.772186	-0.014 ± 0.031	-0.21%	2.78	1/3
FineWeb-Edu/GPT-2 (random split, 6B tokens)							
Tied (S1)	0.0000%	3.210006	24.779243	0.000 ± 0.000	0.00%	–	–
Input-Bias (S3)	0.0008%	3.204392	24.640522	-0.139 ± 0.101	-0.56%	699.09	3/3
Input-Affine (S6)	0.0511%	**3.202202**	**24.586610**	$\mathbf{-0.193 \pm 0.076}$	**-0.78%**	15.21	3/3
FineWeb-Edu/Qwen3-0.6B (20B tokens)							
Tied (S1)	0.00000%	2.758717	15.779585	0.000 ± 0.000	0.00%	–	–
Input-Bias (S3)	0.00017%	2.750357	15.648217	-0.131 ± 0.083	-0.83%	4843.89	3/3
Input-Affine (S6)	0.01117%	**2.747262**	**15.599856**	$\mathbf{-0.180 \pm 0.025}$	**-1.14%**	101.96	3/3

PPL reduction, using 0.00199% and 0.129% of its added parameters, respectively; Figure 3 visualizes the comparison. Subsequent comparisons use Tied as the common baseline.

Input-side relaxation consistently outperforms the tied baseline Input-Bias offers the smallest parameter overhead and the largest PPL reduction per added parameter in both smaller-model settings. Input-Affine achieves lower mean loss on both FineWeb-Edu/GPT-2 splits and the 0.6B configuration, while Input-Bias performs better on MiniMind. The two variants thus offer different trade-offs between parameter cost and validation loss. Input-Add-LoRA and Input-Mult-LR provide smaller mean gains in the two smaller-model settings (Table 5).

Fixed-random split replication On FineWeb-Edu/GPT-2 with a fixed-random 1% validation split, Input-Bias and Input-Affine improve over their same-seed S1 baselines in all three seeds, reducing PPL by 0.139 ± 0.101 and 0.193 ± 0.076, respectively (Table 5). This preserves the mean ordering observed under the sequential split; protocol and per-seed losses are in Appendix B.5.

Scaling to 0.6B parameters and 20B tokens The Qwen3-0.6B-aligned experiment extends the positive result to a larger pretraining configuration (Table 5). Input-Bias and Input-Affine improve over S1 in all three seeds, with paired PPL reductions of 0.131 ± 0.083 and 0.180 ± 0.025, respectively. Input-Bias adds 1,024 parameters (0.00017%), while Input-Affine adds 66,560 (0.01117%). Input-Affine achieves the largest mean reduction with less variation across seeds.

Input-side versus output-side allocation At the same rank and parameter budget ($2dr = 49{,}152$), Input-Mult-LR (S5) achieves lower loss than Output-Mult-LR (S8) in all three seeds on both MiniMind and FineWeb-Edu/GPT-2, with mean reductions of 0.003526 and 0.003277, respectively (Tables 23 and 24). This controlled placement comparison supports allocating limited extra freedom to the input side, as motivated by the output-like geometry of the tied matrix.

Learned translation versus dynamic centering We compare Input-Bias with a zero-parameter alternative that subtracts the current vocabulary mean $\mu(W) = V^{-1}\sum_i W_i$ from input embeddings at every forward pass, retaining $U_{\text{eff}} = W$. Under a fixed-random MiniMind split, dynamic centering improves over S1 in 3/3 seeds, but Input-Bias has lower mean loss (Table 6). On Qwen3-0.6B/FineWeb-Edu, centering worsens loss in 3/3 seeds, whereas Input-Bias improves in 3/3. These results support learning the input translation from the language-modeling objective rather than prescribing complete mean removal. Per-seed results and settings are in Appendix B.6.

Table 6: Dynamic centering versus learned input translation: mean validation cross-entropy over three seeds. Both configurations use fixed-random validation splits.

Configuration	Tied	Centering	Input-Bias
MiniMind	1.738252	1.733425	**1.730690**
Qwen3-0.6B/FineWeb-Edu	2.758717	2.790172	**2.750357**

5.3 Preserving Gains after Supervised Fine-Tuning

We fine-tune all nine checkpoints from the 0.6B/FineWeb-Edu experiment—S1, Input-Bias, and Input-Affine at each of three seeds—for one epoch on HuggingFaceTB/smol-smoltalk. All runs update the full model under the same recipe and Qwen3 ChatML template. We reload the final weights and evaluate all 24,229 held-out test conversations, computing cross-entropy only on assistant content and its terminating token (Appendix C).

Table 7: SmolTalk SFT: held-out assistant-token cross-entropy. Paired Δ is variant minus same-seed S1, reported as mean $\pm$ sample SD.

Variant	Mean CE	Paired Δ CE	Improved seeds
Tied (S1)	1.353064	–	–
Input-Bias (S3)	1.344412	-0.008652 ± 0.007520	3/3
Input-Affine (S6)	**1.338463**	-0.014602 ± 0.004075	3/3

Both AGD variants retain their advantage over S1 in all three seeds after SFT (Table 7). The mean ordering is Input-Affine, Input-Bias, S1, showing that the pretraining benefit persists in held-out assistant-token likelihood after instruction tuning.

6 Conclusion

Weight tying couples input representation and output prediction through a shared vocabulary matrix. Our analysis shows that this sharing is geometrically asymmetric: the shared space is shaped more strongly by the output role, constraining the input side to an output-oriented geometry. AGD turns this insight into an architectural principle: preserve the parameter efficiency of vocabulary sharing while giving the input side its own structured freedom. Lightweight input-side corrections improve pretraining and retain their benefits after supervised fine-tuning. These findings support asymmetric decoupling as an effective middle ground between hard tying and full untying for parameter-constrained language models.

Limitations

Our geometric diagnostics cover public models up to roughly 754B parameters, while controlled AGD training reaches a 595.8M-parameter configuration on approximately 20B FineWeb-Edu tokens. Post-training evidence consists of full-parameter SmolTalk SFT, evaluated by held-out assistant-token likelihood. Generalization to multi-billion-parameter training, broader downstream task metrics, and RL/DPO remains to be tested. Additional low-rank settings, longer training budgets, and finer analysis of gradient allocation between input-side bias and multiplicative modules are directions for further study.

AI USE STATEMENT

Generative AI tools assisted with implementation and debugging of experimental code, discussion of experimental design and interpretation of results, and translation and editing of manuscript material. They also assisted with organizing research materials and adapting the manuscript's LaTeX formatting. The authors are responsible for the final text, claims, experimental results, and accompanying artifacts.

ETHICS STATEMENT

This work performs diagnostic analysis and pretraining experiments on publicly available pretrained language models and standard pretraining corpora. We use public models, tokenizers, and corpora according to their released licenses and terms of use; we do not redistribute the original corpora or pretrained model weights. It does not involve human subjects or personally sensitive data. To the authors' knowledge, there are no additional ethical risks requiring disclosure.

REPRODUCIBILITY STATEMENT

The model parameterizations and assumptions are specified in Section 4 and Appendix A. The appendix provides the diagnostic model inventory, data-processing and sampling protocols, training hyperparameters, variant definitions, and per-seed evaluation results in Appendices B, B.2, B.4 and C. These details describe the settings underlying the results reported in this manuscript.

REFERENCES

Francesco Bertolotti and Walter Cazzola. By tying embeddings you are assuming the Distributional Hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3584–3610. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bertolotti24a.html.

Harsh Bihany, Shubham Patel, and Ashutosh Modi. LoRMA: Low-rank multiplicative adaptation for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10113–10133, Vienna, Austria, Jul 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.527. URL https://aclanthology.org/2025.findings-acl.527/.

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5117–5130, Online, Jun 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.403. URL https://aclanthology.org/2021.naacl-main.403/.

Hyung Won Chung, Thibault Févry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xpFFI_NtgpW.

DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model, 2024a. URL https://arxiv.org/abs/2405.04434.

DeepSeek-AI. DeepSeek-V3 technical report, 2024b. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI. DeepSeek V4 preview release. DeepSeek API documentation, with link to official technical report, Apr 2026. URL https://api-docs.deepseek.com/news/news260424.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkEYojRqtm.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Gemma Team. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

GLM-5 Team. GLM-5: From vibe coding to agentic engineering. arXiv preprint arXiv:2602.15763, 2026. URL https://arxiv.org/abs/2602.15763.

Jingyao Gong. MiniMind: Train a small language model from scratch. GitHub repository, 2025. URL https://github.com/jingyaogong/minimind.

Google DeepMind. Gemma 4. Google DeepMind model page, 2026. URL https://deepmind.google/models/gemma/gemma-4.

Kristina Gulordava, Laura Aina, and Gemma Boleda. How to represent a word and predict it, too: Improving tied architectures for language modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2936–2941, Brussels, Belgium, Oct–Nov 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1323. URL https://aclanthology.org/D18-1323/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=r1aPbsFle.

Andrew Lee, Melanie Weber, Fernanda Viégas, and Martin Wattenberg. Shared global and local geometry of language model embeddings. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=aJDykpJAYF.

Llama Team, AI at Meta. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Anemily Machina and Robert Mercer. Anisotropy is not inherent to transformers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4892–4907, Mexico City, Mexico, Jun 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.274. URL https://aclanthology.org/2024.naacl-long.274/.

MiniMax. MiniMax M2.5: Built for real-world productivity. MiniMax news post, Feb 2026a. URL https://minimax.io/news/minimax-m25.

MiniMax. MiniMax M2.7: Early echoes of self-evolution. MiniMax news post, Mar 2026b. URL https://www.minimax.io/news/minimax-m27-en.

Nikolaos Pappas, Lesly Miculicich, and James Henderson. Beyond weight tying: Learning joint input-output embeddings for neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 73–83, Brussels, Belgium, Oct 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6308. URL https://aclanthology.org/W18-6308/.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.17557.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, Valencia, Spain, Apr 2017. Association for Computational Linguistics. doi: 10.18653/v1/E17-2025. URL https://aclanthology.org/E17-2025/.

Qwen Team. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3.5: Towards native multimodal agents. Qwen Team blog post, Feb 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all. Qwen Team blog post, Apr 2026b. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Z.AI. GLM-5.1. Z.AI documentation, 2026. URL https://docs.z.ai/guides/llm/glm-5.1.

A THEORETICAL AND NOTATION DEFINITIONS

This section supplements technical details compressed in Sections 3 and 4. Appendix A.1 summarizes the main notation in the order it appears in the main text; Appendix A.2 gives the full gradient derivation of dense–sparse update asymmetry; Appendix A.3 explains the metrics used in later geometric diagnostics; and Appendix A.4 expands the low-rank relaxation assumptions in AGD.

A.1 NOTATION DEFINITIONS IN MAIN-TEXT ORDER

Table 8: Main notation used in this paper. Symbols are arranged roughly in the order they first appear in the main technical narrative; when a symbol is overloaded across contexts, the description notes this explicitly.

Symbol	Meaning	Description
V	Vocabulary size	Number of rows in vocabulary matrices; V may differ across model families.
d	Hidden dimension	Number of columns in vocabulary matrices and Transformer hidden-state dimension.
$\mathbf{x} = (x_1, \ldots, x_T)$	Token sequence	$x_t \in \{1, \ldots, V\}$ denotes the t-th input token.
$E \in \mathbb{R}^{V \times d}$	Input embedding matrix	Maps discrete tokens to hidden-space vectors.
$e_t = E[x_t]$	Input embedding vector	Input-side representation obtained by lookup for token t.
$U \in \mathbb{R}^{V \times d}$	Output unembedding	Projects final hidden states back to vocabulary logits; `lm_head` in code corresponds to this object.
W	Shared or output vocabulary matrix	In tied settings, usually the shared matrix serving both input and output roles; in some formulas also denotes the output unembedding matrix.
$tied$ / $untied$	Tied / untied parameterization	Tied means $E = U = W$; untied means input embeddings E and output unembeddings U are independent.
$h_t \in \mathbb{R}^d$	Final hidden state	Hidden vector output by the Transformer backbone at position t.
z_t	Output logits	Usually written as $z_t = Uh_t + b$; if output bias is omitted, $z_t = Uh_t$.
$b \in \mathbb{R}^V$	Output bias	Can absorb unigram frequency or global calibration terms; our geometric analysis focuses mainly on row-vector structure of E and U.
p_t	Next-token distribution	$p_t = \mathrm{softmax}(z_t)$.
y_t	Target token	In autoregressive language modeling, usually $y_t = x_{t+1}$.
$\ell_t, \mathcal{L}$	Token loss and average loss	$\ell_t = -\log p_t[y_t]$, and $\mathcal{L}$ is mean cross-entropy over sequence positions.
∇W_{out}	Output-side gradient	From the full-vocabulary dense classification objective of output unembeddings.
∇W_{in}	Input-side gradient	From the row-sparse input objective of embedding lookup.
$X, Y \in \mathbb{R}^{n \times d}$	Aligned vocabulary matrices	Two matrices used in geometric diagnostics; may correspond to E–E or U–U comparisons.
μ	Token-pair relation metric	May be cosine similarity, Euclidean distance, or squared Euclidean distance.
$\mathrm{GCorr}_\mu(X, Y)$	Global geometric correlation	Compares pairwise geometric relations induced by X and Y on the same token pairs.
$N_{tok}, N_{pair}, N_{fit}$	Sampling and fitting scale	Token sampling cap, pairwise estimate count, and number of tokens used in affine-mapping fitting.
A, c	Affine-mapping parameters	Used in geometric diagnostics to fit $\hat{Y} = XA + \mathbf{1}c^\top$; distinct from A in LoRA.
$M \in \mathbb{R}^{V \times d}$	Generic vocabulary matrix	In the method section, the base input embedding or output unembedding matrix being relaxed.
ΔM	Vocabulary-matrix increment	Low-rank correction matrix learned by LoRA-style relaxation.

Table 8 (continued)

Symbol	Meaning	Description
r	LoRA rank	Controls low-rank correction freedom; usually $r \ll d$ and $r \ll V$.
A, B	Additive LoRA factors	In $\Delta M = AB$, $A \in \mathbb{R}^{V\times r}$ and $B \in \mathbb{R}^{r\times d}$.
P, Q	Multiplicative LoRA factors	In $M_{\text{eff}} = M(I_d + sPQ)$, $P \in \mathbb{R}^{d\times r}$ and $Q \in \mathbb{R}^{r\times d}$.
s	Multiplicative LoRA scale	Scalar $s > 0$ controlling low-rank deformation strength; adds no matrix parameters. Main pretraining experiments use $s = 1$.
β	Affine correction vector	Learnable vector for global translation correction on input or output side in AGD; can also be viewed as a rank-1 additive LoRA special case with vocabulary-side coefficients fixed to 1.
$E_{\text{eff}}, U_{\text{eff}}$	Effective input / output matrices	Matrices actually used for lookup and output unembedding projection in S1–S12.

A.2 Full Derivation of Dense–Sparse Update Asymmetry

For notational simplicity, let $\mathbf{y}_t \in \{0,1\}^V$ denote the one-hot vector of target token y_t. From softmax cross-entropy,

$$\frac{\partial \ell_t}{\partial z_t} = p_t - \mathbf{y}_t. \tag{5}$$

Therefore, the gradient of the output unembedding matrix is

$$\frac{\partial \mathcal{L}}{\partial U} = \frac{1}{T-1}\sum_{t=1}^{T-1}(p_t - \mathbf{y}_t)h_t^\top. \tag{6}$$

For the output row vector U_i corresponding to token i, its gradient is

$$\frac{\partial \mathcal{L}}{\partial U_i} = \frac{1}{T-1}\sum_{t=1}^{T-1}\big(p_t[i] - \mathbb{I}[y_t = i]\big)h_t. \tag{7}$$

Because softmax probabilities $p_t[i]$ are usually non-zero across all vocabulary positions, the output side produces task gradients for classifier vectors of nearly all V tokens at every training position. This is the full-vocabulary dense update described in the main text.

By contrast, input embeddings are a lookup operation. Let $g_t = \partial \mathcal{L}/\partial e_t$ denote the gradient back-propagated from the Transformer backbone to the input embedding vector; then the gradient of input embedding row E_i is

$$\frac{\partial \mathcal{L}}{\partial E_i} = \sum_{t:x_t=i} g_t. \tag{8}$$

Only when token i appears in the current input sequence does that row receive input-side task gradients; therefore, the input side is row-sparse. Optimizer terms such as weight decay, momentum, or Adam second moments may add scaling or state updates for unseen tokens, but do not change the sparse nature of task gradients themselves.

Under tying, the same matrix W serves both input and output roles, so its task gradient can be written as

$$\nabla W_{\text{tied}} = \nabla W_{\text{out}} + \nabla W_{\text{in}}. \tag{9}$$

Because ∇W_{out} covers the full vocabulary at every step whereas ∇W_{in} covers only rows of currently observed tokens, the long-term optimization dynamics of the shared matrix are naturally more easily dominated by output-side classification objectives. This asymmetry is the training-dynamics basis for the "output-manifold attractor" and input geometric degradation analyses in the main text.

A.3 Geometric Metrics and Diagnostic Indicators

Geometric diagnostics in the main text do not directly compare element-wise values of two token vectors, but compare token–token relations induced by vocabulary matrices. Given two aligned

matrices $X, Y \in \mathbb{R}^{n\times d}$, we compute similarity or distance on the same token pairs (i_k, j_k) and compare Pearson correlation of the two relation vectors:

$$\begin{aligned} \mathrm{GCorr}_\mu(X,Y) &= \mathrm{PearsonCorr}\big(\mathbf{v}_\mu^X, \mathbf{v}_\mu^Y\big), \\ \mathbf{v}_\mu^X &:= [\mu(X_{i_k}, X_{j_k})]_{k=1}^m, \\ \mathbf{v}_\mu^Y &:= [\mu(Y_{i_k}, Y_{j_k})]_{k=1}^m. \end{aligned} \tag{10}$$

Here μ includes cosine similarity, Euclidean distance, and squared Euclidean distance. The three metrics answer slightly different questions: cosine focuses more on directional structure and is easily affected by global center offset; Euclidean distance preserves relative distance structure of translated point clouds and is better suited for observing whether input embeddings maintain stable semantic constellations; squared Euclidean serves as a robustness check, amplifying larger distance differences and helping exclude accidental conclusions from a single distance form.

Euclidean and squared-Euclidean GCorr are invariant to translation and uniform positive rescaling of either matrix, since Pearson correlation is unchanged by positive scaling of a distance vector. A decrease in these correlations therefore reflects changes in relative distance relations rather than uniform scale alone.

GCorr near 1 indicates that two matrices induce highly consistent token–token geometric structure; near 0 indicates weak geometric correlation; below 0 indicates anti-correlated relations on sampled token pairs. Because computing the full $V \times V$ distance matrix is too expensive, the main text uses Monte Carlo token-pair estimation with bootstrap confidence intervals.

Spectral indicators For a vocabulary matrix X with rows x_i, define $m = n^{-1}\sum_i x_i$ and $X_c = X - \mathbf{1}m^\top$. Mean-axis strength is $\|m\|_2/(n^{-1}\sum_i \|x_i\|_2)$. Raw rank-1 energy is $\sigma_1(X)^2/\|X\|_F^2$; centered rank-1 energy is $\sigma_1(X_c)^2/\|X_c\|_F^2$. Centering removes the shared mean component, while the centered statistic measures concentration in the remaining variation. Incremental rank a–b energy is $\sum_{j=a}^b \sigma_j(X)^2/\|X\|_F^2$. The normalized participation ratio is $\mathrm{PR}/d = (d\sum_j p_j^2)^{-1}$, where $p_j = \sigma_j(X)^2/\|X\|_F^2$; the centered version uses X_c. Larger values indicate energy spread across more spectral directions relative to hidden dimension d.

A.4 Low-Rank Parameterization and the Low-Rank Subspace Hypothesis

The core motivation of original LoRA is low intrinsic rank during model adaptation: when a pre-trained model is adapted to a specific task or domain, full weight updates ΔW often need not occupy the full rank of the original matrix and can be approximated with low-rank decomposition (Hu et al., 2022). Therefore, original LoRA usually freezes base weights and trains only additional low-rank adaptation matrices; these matrices serve parameter-efficient fine-tuning in post-training and can optionally remain mounted or be merged with base weights at deployment.

Our use of LoRA-style low-rank matrices has a different purpose. We do not add temporary task adapters to an existing model, but use low-rank freedom as part of embedding-head architecture itself to alleviate geometric conflict between tied input embeddings and output unembeddings. In other words, our low-rank hypothesis is that the main differences required on the input and output sides relative to the shared vocabulary matrix need not be a full independent $V \times d$ matrix, but can be expressed by "mobility" along a few shared directions. Therefore, these low-rank matrices are not optional post-training attachments functionally, but permanent components of effective input matrix E_{eff} or effective output matrix U_{eff}.

Given base vocabulary matrix $M \in \mathbb{R}^{V\times d}$, standard additive LoRA restricts the full increment matrix ΔM to low-rank decomposition:

$$\begin{aligned} M_{\mathrm{eff}} &= M + \Delta M, \\ \Delta M &= AB, \\ A \in \mathbb{R}^{V\times r}, &\quad B \in \mathbb{R}^{r\times d}. \end{aligned} \tag{11}$$

where $r \ll \min(V, d)$ and the added parameter count is $r(V + d)$, far smaller than a full additional $V \times d$ matrix introduced by complete untying. The actual assumption is not that correction magnitude is small for each token, but that correction vectors for all tokens jointly lie in the same

low-dimensional subspace. Let the low-rank coefficient for token i be $a_i \in \mathbb{R}^r$; then its row correction can be written as

$$\begin{aligned} \Delta m_i &= a_i^\top B, \\ \Delta m_i &\in \mathrm{span}(B_1, \ldots, B_r) \subseteq \mathbb{R}^d. \end{aligned} \tag{12}$$

Different tokens may have different coefficients a_i, but the directions they can move in are determined by the same set of r global basis directions; equivalently, the full correction matrix satisfies $\mathrm{rank}(\Delta M) \leq r$.

Therefore, the low-rank subspace hypothesis contains two constraints: first, main geometric differences on the input or output side relative to the shared vocabulary matrix can be explained by a few global directions; second, differences among tokens appear mainly as combination weights on these directions, rather than requiring many independent high-rank token-specific corrections. Full untying allows each token to adjust its input representation or output unembedding weight independently in $\mathbb{R}^d$; LoRA allows all tokens to adjust only within a shared low-rank correction space. Our weight-space LoRA ablations test whether the freedom required for vocabulary-layer input/output decoupling can be replaced by this low-rank subspace hypothesis.

Affine correction vector β can be viewed as an extreme special case of this additive low-rank form. For input-side translation,

$$E_{\text{eff}} = W - \mathbf{1}\beta^\top \tag{13}$$

is equivalent to letting $r = 1$, $A = \mathbf{1} \in \mathbb{R}^{V \times 1}$, and $B = -\beta^\top \in \mathbb{R}^{1 \times d}$ in $\Delta M = AB$. That is, β does not assign different coefficients to different tokens, but forces all vocabulary rows to translate along the same global direction. Its key advantage is fixing the large vocabulary-side coefficient vector to all ones and learning no V-dependent parameters on the vocabulary side, achieving maximal vocabulary-side parameter savings while still capturing the global origin/centroid offset observed in the main text.

Compared with additive LoRA, hidden-dimensional multiplicative LoRA places low-rank freedom in hidden coordinate transforms:

$$\begin{aligned} M_{\text{eff}} &= M(I_d + sPQ), \\ P \in \mathbb{R}^{d \times r}, &\quad Q \in \mathbb{R}^{r \times d}, \quad s > 0. \end{aligned} \tag{14}$$

It introduces only $2dr$ parameters and does not grow with vocabulary size V; s is an additional scalar hyperparameter that damps or amplifies the low-rank correction, and all main pretraining experiments use $s = 1$. Geometrically, it does not add independent corrections for each token, but learns a low-rank deformation in hidden dimensions, allowing the full input embedding or output unembedding matrix to relax controllably along shared low-dimensional directions. This form directly supports the AGD design in the main text of "repairing shared geometry with a small amount of structured freedom."

B Detailed Experimental Setup

This appendix supplements the experimental protocol, hyperparameter configuration, and the complete model comparison list used in the diagnostic analysis.

B.1 Analysis Model Inventory and Vocabulary-Matrix Configuration

The model inventory used for tied/untied status analysis comes from a centrally maintained configuration with 94 public models: 44 tied and 50 untied, all with determinable vocabulary-matrix sharing status. We fetch public source files for these models and extract input embedding and output unembedding matrices for analysis; because matrix shapes match in this inventory, we report vocabulary-matrix dimensions uniformly. Total parameters mainly follow nominal counts from public names or model cards; for R1-Distill models, because DeepSeek release versions untie vocabulary matrices, we report parameter counts converted from actual safetensors weights via Hugging Face metadata. For MoE models, this column reports total parameters rather than per-token activated parameters. The main-text empirical analysis (Table 2; Section 3) uses a comparable subset of 82 models—48 untied and 34 tied—obtained by excluding 12 inventory entries: the Gemma-4 family (8 models), Gemma-3-1B / Gemma-3-1B-Instruct (2 models), and DeepSeek-V4-Flash / Pro (2 models). These

models are retained in the inventory below to document public tied/untied configurations, but their static-spectrum or output-head conventions are not directly comparable to the other models; the exclusion rationale is documented in the notes below and in Appendix B.3.

For readability, the inventory is split into multiple tables by model ecosystem. We analyze only text token embeddings and output unembeddings, and do not include vision encoders, cross-modal projectors, or other non-vocabulary modules in multimodal models.

Table 9: Qwen model inventory and vocabulary-matrix configuration. The Qwen family (Qwen Team, 2025; Yang et al., 2025; Qwen Team, 2026a;b) covers dense, MoE, and some natively multimodal variants, and is one of the main samples for scale-stratified tied/untied phenomena in the main text.

Model Group	Model	Total Params	Status	Vocab Matrix Dim
Qwen-2.5	Qwen2.5-0.5B	0.5B	tied	151936×896
	Qwen2.5-0.5B-Instruct	0.5B	tied	151936×896
	Qwen2.5-1.5B	1.5B	tied	151936×1536
	Qwen2.5-1.5B-Instruct	1.5B	tied	151936×1536
	Qwen2.5-3B	3B	tied	151936×2048
	Qwen2.5-3B-Instruct	3B	tied	151936×2048
	Qwen2.5-7B	7B	untied	152064×3584
	Qwen2.5-7B-Instruct	7B	untied	152064×3584
	Qwen2.5-14B	14B	untied	152064×5120
	Qwen2.5-14B-Instruct	14B	untied	152064×5120
	Qwen2.5-32B	32B	untied	152064×5120
	Qwen2.5-32B-Instruct	32B	untied	152064×5120
	Qwen2.5-72B-Base	72B	untied	152064×8192
	Qwen2.5-72B-Instruct	72B	untied	152064×8192
Qwen-3	Qwen3-0.6B-Base	0.6B	tied	151936×1024
	Qwen3-0.6B	0.6B	tied	151936×1024
	Qwen3-1.7B-Base	1.7B	tied	151936×2048
	Qwen3-1.7B	1.7B	tied	151936×2048
	Qwen3-4B-Base	4B	tied	151936×2560
	Qwen3-4B	4B	tied	151936×2560
	Qwen3-8B-Base	8B	untied	151936×4096
	Qwen3-8B	8B	untied	151936×4096
	Qwen3-14B-Base	14B	untied	151936×5120
	Qwen3-14B	14B	untied	151936×5120
	Qwen3-32B	32B	untied	151936×5120
	Qwen3-30B-A3B-Base	30B	untied	151936×2048
	Qwen3-30B-A3B	30B	untied	151936×2048
	Qwen3-235B-A22B	235B	untied	151936×4096
Qwen-3.6	Qwen3.6-35B-A3B	35B	untied	248320×2048
Qwen-3.5 Dense	Qwen3.5-0.8B-Base	0.8B	tied	248320×1024
	Qwen3.5-0.8B-Instruct	0.8B	tied	248320×1024
	Qwen3.5-2B-Base	2B	tied	248320×2048
	Qwen3.5-2B-Instruct	2B	tied	248320×2048
	Qwen3.5-4B-Base	4B	tied	248320×2560
	Qwen3.5-4B-Instruct	4B	tied	248320×2560
	Qwen3.5-9B-Base	9B	untied	248320×4096
	Qwen3.5-9B-Instruct	9B	untied	248320×4096
	Qwen3.5-27B-Instruct	27B	untied	248320×5120
Qwen-3.5 MoE	Qwen3.5-35B-A3B-Base	35B	untied	248320×2048
	Qwen3.5-35B-A3B-Instruct	35B	untied	248320×2048

Continued on next page...

Continued: Qwen model inventory and vocabulary-matrix configuration.

Model Group	Model	Total Params	Status	Vocab Matrix Dim
	Qwen3.5-122B-A10B-Instruct	122B	untied	248320×3072
	Qwen3.5-397B-A17B-Instruct	397B	untied	248320×4096

Table 10: Llama model inventory and vocabulary-matrix configuration. The Llama-3.x (Llama Team, AI at Meta, 2024) group contrasts small-scale tied with larger-scale untied design switches.

Model Group	Model	Total Params	Status	Vocab Matrix Dim
Llama-3.x	Llama-3.1-70B-Base	70B	untied	128256×8192
	Llama-3.1-70B-Instruct	70B	untied	128256×8192
	Llama-3.1-8B	8B	untied	128256×4096
	Llama-3.1-8B-Instruct	8B	untied	128256×4096
	Llama-3.2-1B	1B	tied	128256×2048
	Llama-3.2-1B-Instruct	1B	tied	128256×2048
	Llama-3.2-3B	3B	tied	128256×3072
	Llama-3.2-3B-Instruct	3B	tied	128256×3072

Table 11: Gemma model inventory and vocabulary-matrix configuration. The Gemma family (Gemma Team, 2024; 2025; Google DeepMind, 2026) overall favors tied weights and is an important contrast to Llama/Qwen.

Model Group	Model	Total Params	Status	Vocab Matrix Dim
Gemma-2	Gemma-2-2B	2B	tied	256000×2304
	Gemma-2-2B-Instruct	2B	tied	256000×2304
	Gemma-2-9B	9B	tied	256000×3584
	Gemma-2-9B-Instruct	9B	tied	256000×3584
	Gemma-2-27B	27B	tied	256000×4608
	Gemma-2-27B-Instruct	27B	tied	256000×4608
Gemma-3	Gemma-3-1B	1B	tied	262144×1152
	Gemma-3-1B-Instruct	1B	tied	262144×1152
	Gemma-3-4B	4B	tied	262208×2560
	Gemma-3-4B-Instruct	4B	tied	262208×2560
	Gemma-3-12B	12B	tied	262208×3840
	Gemma-3-12B-Instruct	12B	tied	262208×3840
	Gemma-3-27B	27B	tied	262208×5376
	Gemma-3-27B-Instruct	27B	tied	262208×5376
Gemma-4	Gemma-4-E2B	2B	tied	262144×1536
	Gemma-4-E4B	4B	tied	262144×2560
	Gemma-4-26B-A4B	26B	tied	262144×2816
	Gemma-4-26B-A4B-Instruct	26B	tied	262144×2816
	Gemma-4-31B	31B	tied	262144×5376
	Gemma-4-E2B-Instruct	2B	tied	262144×1536
	Gemma-4-E4B-Instruct	4B	tied	262144×2560
	Gemma-4-31B-Instruct	31B	tied	262144×5376

Gemma exclusions for static spectral analysis The Gemma-4 family and Gemma-3-1B / Gemma-3-1B-Instruct are retained in the inventory to document tied/untied configurations among public models, but they are excluded from the main static SVD/spectral analysis. In the public Gemma-4 weights, input embeddings exhibit a clear row-norm gauge, so raw Euclidean and uncentered spectral statistics are affected by radial scale. Gemma-3-1B shows a similar small-model

normalization exception in the Base–Instruct and static-spectrum checks. To keep the spectral comparison on a consistent convention, the main analysis uses only comparable Gemma-2 and Gemma-3 4B/12B/27B models.

Table 12: DeepSeek model inventory and vocabulary-matrix configuration. The DeepSeek group (DeepSeek-AI, 2024a;b; 2025; 2026) includes MoE, reasoning, and distilled models; R1-Distill denotes distilling R1 reasoning capability onto Qwen/Llama target backbones, not DeepSeek-owned small base models. Total parameters for R1-Distill rows are converted from actual safetensors weights.

Model Group	Model	Total Params	Status	Vocab Matrix Dim
DeepSeek-V2 MoE	DeepSeek-V2-Lite	16B	untied	102400×2048
	DeepSeek-V2-Lite-Chat	16B	untied	102400×2048
DeepSeek-V3 MoE	DeepSeek-V3-Base	671B	untied	129280×7168
	DeepSeek-V3	671B	untied	129280×7168
	DeepSeek-V3.1-Base	671B	untied	129280×7168
	DeepSeek-V3.1	671B	untied	129280×7168
	DeepSeek-V3.2	671B	untied	129280×7168
	DeepSeek-V3.2-Speciale	671B	untied	129280×7168
DeepSeek-V4 MoE	DeepSeek-V4-Flash	284B	untied	129280×4096
	DeepSeek-V4-Pro	1.6T	untied	129280×7168
DeepSeek-R1	DeepSeek-R1	671B	untied	129280×7168
	DeepSeek-R1-0528	671B	untied	129280×7168
R1-Distill-Qwen (R1→Qwen)	DeepSeek-R1-Distill-Qwen-1.5B	1.8B	untied	151936×1536
	DeepSeek-R1-Distill-Qwen-7B	7.6B	untied	152064×3584
	DeepSeek-R1-Distill-Qwen-14B	15B	untied	152064×5120
	DeepSeek-R1-Distill-Qwen-32B	33B	untied	152064×5120
R1-Distill-Llama (R1→Llama)	DeepSeek-R1-Distill-Llama-8B	8.0B	untied	128256×4096
	DeepSeek-R1-Distill-Llama-70B	71B	untied	128256×8192

DeepSeek-V4 exclusions for static spectral analysis DeepSeek-V4-Flash / Pro are also retained in the inventory, but are excluded from the main untied static-spectrum analysis. Their output path combines HC state mixing, `hc_head`, RMSNorm, and `head.weight`; the bare `head.weight` is therefore not equivalent to the conventional complete LM head used in the other models. Consequently, DeepSeek-V4's bare U spectrum is not directly comparable to standard untied output-unembedding spectra, and the main analysis treats it as an architectural exception.

Table 13: GLM and MiniMax model inventory and vocabulary-matrix configuration. The GLM (GLM-5 Team, 2026; Z.AI, 2026) and MiniMax (MiniMax, 2026a;b) groups include additional large-scale MoE families to supplement tied/untied choices across vendors.

Model Group	Model	Total Params	Status	Vocab Matrix Dim
MiniMax-M2.5	MiniMax-M2.5-Chat	230B	untied	200064×3072

Continued on next page...

Continued: GLM and MiniMax model inventory and vocabulary-matrix configuration.

Model Group	Model	Total Params	Status	Vocab Matrix Dim
MiniMax-M2.7	MiniMax-M2.7	229B	untied	200064×3072
GLM-5	GLM-5-744B	744B	untied	154880×6144
GLM-5.1	GLM-5.1	754B	untied	154880×6144

B.2 Hyperparameter Configuration

For global geometric analysis (Protocol I), we use a stratified sampling strategy. Detailed settings are as follows:

- **Matrix objects:** Each model analyzes input embedding matrix E and output unembedding matrix U; Base–Instruct comparisons compute E_{base} vs. E_{instruct} and U_{base} vs. U_{instruct} separately.
- **Token alignment:** Same-tokenizer/same-vocab models align by token ID; cross-tokenizer or different-vocab models align by token-string intersection; exclude special tokens such as bos/eos/pad/unk; comparisons with fewer than 1000 shared tokens are skipped.
- **Token sample size (N_{tok}):** 20,000 (upper bound). For cross-tokenizer comparisons, we sample up to 20,000 tokens from the string intersection.
- **Pairwise estimate count (N_{pair}):** 5,000,000.
- **GCorr metrics:** For the same batch of token pairs, compute cosine similarity, Euclidean distance, and squared Euclidean distance vectors simultaneously and obtain their Pearson correlations separately.
- **Bootstrap iterations:** $N = 100$ for both Base–Instruct and model-series comparisons.
- **Confidence intervals:** 95% CI based on bootstrap percentiles.

Static spectral subsets. The static vocabulary-matrix spectral analysis applies economy SVD to vocabulary matrices in public model weights and reports mean-axis strength, raw rank-1 energy, centered rank-1 energy, and related participation-ratio statistics. To keep spectra comparable, the main untied spectral subset contains 48 public models and excludes DeepSeek-V4-Flash/Pro because their output head is not a conventional LM-head matrix in the same sense as the other models. The main tied spectral subset contains 34 public models and excludes Gemma-4 variants and Gemma-3-1B because their row-norm/gauge behavior produces a different normalization regime for the static-spectrum comparison. These exclusions affect only the static spectral panels and do not affect the 37 tied–untied GCorr pairs used as the main evidence that tied matrices are closer to untied U than to untied E.

Table 14 shows that centering reverses the ordering of median PR/d between untied E and U: U is lower before centering (0.100 vs. 0.309) and higher afterward (0.466 vs. 0.368), with centered U exceeding E in 34/48 models. A stronger leading component can therefore coexist with broadly distributed residual energy.

B.3 Complete Model Comparison List

The Model Series block contains all 110 comparisons used in the main GCorr analysis. Base/Instruct contains the 30-pair comparable subset of the 35-pair archive, excluding Gemma-3-1B and the four Gemma-4 pairs. The Cross-Scale block (120 pairs) includes all 26 cross-series untied–untied pairs summarized in Table 3. This block and MoE Cross-Family (15 pairs) are reported separately from the 110 within-series comparisons. Static-spectrum exclusions are applied separately from the GCorr model-series analysis.

Table 14: Static spectral indicators: medians over 48 untied and 34 tied models. Raw and centered energies use their respective Frobenius norms.

Indicator	Untied E	Untied U	Tied W
Mean-axis strength	0.094	0.213	0.324
Raw rank-1 energy	1.007%	5.264%	11.434%
Centered rank-1 energy	0.616%	1.111%	2.477%
$\lvert\cos(v_1, m)\rvert$	0.972	0.996	0.999
Raw PR/d	0.309	0.100	0.048
Centered PR/d	0.368	0.466	0.490

Table 15: Supplementary GCorr results. We report Pearson correlation (ρ) for Cosine and Euclidean GCorr; squared Euclidean GCorr is computed with the same protocol but omitted from this compact table.

Task	**Model Pair (A vs. B)**	$E_{\cos}$	E_{euc}	$U_{\cos}$	U_{euc}
Base/Instruct	Qwen3-0.6B-Base vs. Qwen3-0.6B	0.990	0.994	0.990	0.994
	Qwen3-1.7B-Base vs. Qwen3-1.7B	0.995	0.998	0.995	0.998
	Qwen3-4B-Base vs. Qwen3-4B	0.997	0.997	0.997	0.997
	Qwen3-8B-Base vs. Qwen3-8B	0.994	1.000	0.998	0.999
	Qwen3-14B-Base vs. Qwen3-14B	0.993	1.000	0.997	0.999
	Qwen3-30B-A3B-Base vs. Qwen3-30B-A3B	0.987	1.000	0.998	0.998
	Qwen3.5-0.8B-Base vs. Qwen3.5-0.8B-Instruct	0.999	0.999	0.999	0.999
	Qwen3.5-2B-Base vs. Qwen3.5-2B-Instruct	1.000	0.999	1.000	0.999
	Qwen3.5-4B-Base vs. Qwen3.5-4B-Instruct	1.000	0.999	1.000	0.999
	Qwen3.5-9B-Base vs. Qwen3.5-9B-Instruct	0.998	1.000	1.000	0.999
	Qwen3.5-35B-A3B-Base vs. Qwen3.5-35B-A3B-Instruct	0.999	1.000	1.000	0.999
	Qwen2.5-0.5B vs. Qwen2.5-0.5B-Instruct	0.998	0.996	0.998	0.996
	Qwen2.5-1.5B vs. Qwen2.5-1.5B-Instruct	1.000	1.000	1.000	1.000
	Qwen2.5-3B vs. Qwen2.5-3B-Instruct	1.000	1.000	1.000	1.000
	Qwen2.5-7B vs. Qwen2.5-7B-Instruct	1.000	1.000	1.000	1.000
	Qwen2.5-14B vs. Qwen2.5-14B-Instruct	1.000	1.000	0.999	1.000
	Qwen2.5-32B vs. Qwen2.5-32B-Instruct	1.000	1.000	1.000	1.000
	Qwen2.5-72B-Base vs. Qwen2.5-72B-Instruct	1.000	1.000	1.000	1.000
	Llama-3.2-1B vs. Llama-3.2-1B-Instruct	0.981	0.970	0.981	0.970
	Llama-3.2-3B vs. Llama-3.2-3B-Instruct	0.986	0.985	0.986	0.985
	Llama-3.1-8B vs. Llama-3.1-8B-Instruct	0.998	1.000	0.999	0.999
	Llama-3.1-70B-Base vs. Llama-3.1-70B-Instruct	0.999	1.000	0.998	0.999
	Gemma-2-2B vs. Gemma-2-2B-Instruct	1.000	0.999	1.000	0.999
	Gemma-2-9B vs. Gemma-2-9B-Instruct	0.999	0.996	0.999	0.996
	Gemma-2-27B vs. Gemma-2-27B-Instruct	0.998	0.982	0.998	0.982
	Gemma-3-4B vs. Gemma-3-4B-Instruct	0.979	0.894	0.979	0.894
	Gemma-3-12B vs. Gemma-3-12B-Instruct	0.988	0.946	0.988	0.946
	Gemma-3-27B vs. Gemma-3-27B-Instruct	0.990	0.953	0.990	0.953
	DeepSeek-V3-Base vs. DeepSeek-V3	1.000	1.000	1.000	1.000
	DeepSeek-V3.1-Base vs. DeepSeek-V3.1	1.000	1.000	1.000	1.000
Model Series	Llama-3.1-70B-Instruct vs. Llama-3.1-8B-Instruct	0.410	0.885	0.843	0.897
	Llama-3.1-70B-Instruct vs. Llama-3.2-1B-Instruct	0.178	0.354	0.752	0.645
	Llama-3.1-70B-Instruct vs. Llama-3.2-3B-Instruct	0.159	0.434	0.750	0.726

Continued on next page...

Table 15 (continued)

Task	Model Pair (A vs. B)	$E_{\cos}$	E_{euc}	$U_{\cos}$	U_{euc}
	Llama-3.1-8B-Instruct vs. Llama-3.2-1B-Instruct	0.074	0.359	0.875	0.672
	Llama-3.1-8B-Instruct vs. Llama-3.2-3B-Instruct	0.057	0.424	0.877	0.722
	Llama-3.2-1B-Instruct vs. Llama-3.2-3B-Instruct	0.949	0.959	0.949	0.959
	Qwen2.5-0.5B-Instruct vs. Qwen2.5-1.5B-Instruct	0.851	0.889	0.851	0.889
	Qwen2.5-0.5B-Instruct vs. Qwen2.5-3B-Instruct	0.848	0.862	0.848	0.862
	Qwen2.5-0.5B-Instruct vs. Qwen2.5-7B-Instruct	0.059	0.624	0.849	0.803
	Qwen2.5-0.5B-Instruct vs. Qwen2.5-14B-Instruct	0.150	0.587	0.666	0.753
	Qwen2.5-0.5B-Instruct vs. Qwen2.5-32B-Instruct	0.152	0.560	0.742	0.767
	Qwen2.5-0.5B-Instruct vs. Qwen2.5-72B-Instruct	0.118	0.516	0.825	0.753
	Qwen2.5-1.5B-Instruct vs. Qwen2.5-3B-Instruct	0.915	0.972	0.915	0.972
	Qwen2.5-1.5B-Instruct vs. Qwen2.5-7B-Instruct	0.090	0.620	0.793	0.796
	Qwen2.5-1.5B-Instruct vs. Qwen2.5-14B-Instruct	0.158	0.604	0.736	0.876
	Qwen2.5-1.5B-Instruct vs. Qwen2.5-32B-Instruct	0.166	0.575	0.769	0.852
	Qwen2.5-1.5B-Instruct vs. Qwen2.5-72B-Instruct	0.143	0.514	0.781	0.760
	Qwen2.5-3B-Instruct vs. Qwen2.5-7B-Instruct	0.092	0.665	0.822	0.787
	Qwen2.5-3B-Instruct vs. Qwen2.5-14B-Instruct	0.165	0.653	0.786	0.915
	Qwen2.5-3B-Instruct vs. Qwen2.5-32B-Instruct	0.171	0.623	0.814	0.892
	Qwen2.5-3B-Instruct vs. Qwen2.5-72B-Instruct	0.144	0.556	0.808	0.781
	Qwen2.5-7B-Instruct vs. Qwen2.5-14B-Instruct	0.306	0.971	0.755	0.807
	Qwen2.5-7B-Instruct vs. Qwen2.5-32B-Instruct	0.313	0.962	0.844	0.822
	Qwen2.5-7B-Instruct vs. Qwen2.5-72B-Instruct	0.278	0.938	0.928	0.918
	Qwen2.5-14B-Instruct vs. Qwen2.5-32B-Instruct	0.502	0.993	0.893	0.979
	Qwen2.5-14B-Instruct vs. Qwen2.5-72B-Instruct	0.277	0.950	0.763	0.815
	Qwen2.5-32B-Instruct vs. Qwen2.5-72B-Instruct	0.301	0.955	0.870	0.845
	Qwen3-0.6B vs. Qwen3-1.7B	0.857	0.932	0.857	0.932
	Qwen3-0.6B vs. Qwen3-4B	0.884	0.900	0.884	0.900
	Qwen3-0.6B vs. Qwen3-8B	0.147	0.572	0.846	0.818
	Qwen3-0.6B vs. Qwen3-14B	0.098	0.527	0.733	0.764
	Qwen3-0.6B vs. Qwen3-32B	0.121	0.514	0.741	0.718
	Qwen3-0.6B vs. Qwen3-30B-A3B	0.159	0.594	0.828	0.823
	Qwen3-0.6B vs. Qwen3-235B-A22B	0.149	0.522	0.683	0.745
	Qwen3-1.7B vs. Qwen3-4B	0.874	0.967	0.874	0.967
	Qwen3-1.7B vs. Qwen3-8B	0.140	0.659	0.884	0.943
	Qwen3-1.7B vs. Qwen3-14B	0.099	0.612	0.775	0.896
	Qwen3-1.7B vs. Qwen3-32B	0.120	0.597	0.760	0.845

Continued on next page...

Table 15 (continued)

Task	Model Pair (A vs. B)	$E_{\cos}$	E_{euc}	$U_{\cos}$	U_{euc}
	Qwen3-1.7B vs. Qwen3-30B-A3B	0.158	0.720	0.849	0.915
	Qwen3-1.7B vs. Qwen3-235B-A22B	0.151	0.626	0.766	0.877
	Qwen3-4B vs. Qwen3-8B	0.130	0.697	0.919	0.966
	Qwen3-4B vs. Qwen3-14B	0.073	0.652	0.811	0.923
	Qwen3-4B vs. Qwen3-32B	0.096	0.631	0.819	0.878
	Qwen3-4B vs. Qwen3-30B-A3B	0.125	0.744	0.875	0.940
	Qwen3-4B vs. Qwen3-235B-A22B	0.121	0.658	0.745	0.914
	Qwen3-8B vs. Qwen3-14B	0.347	0.988	0.832	0.954
	Qwen3-8B vs. Qwen3-32B	0.353	0.979	0.819	0.903
	Qwen3-8B vs. Qwen3-30B-A3B	0.291	0.800	0.884	0.950
	Qwen3-8B vs. Qwen3-235B-A22B	0.267	0.911	0.793	0.955
	Qwen3-14B vs. Qwen3-32B	0.538	0.994	0.919	0.982
	Qwen3-14B vs. Qwen3-30B-A3B	0.309	0.747	0.768	0.888
	Qwen3-14B vs. Qwen3-235B-A22B	0.290	0.890	0.823	0.933
	Qwen3-32B vs. Qwen3-30B-A3B	0.321	0.725	0.794	0.838
	Qwen3-32B vs. Qwen3-235B-A22B	0.305	0.882	0.791	0.896
	Qwen3-30B-A3B vs. Qwen3-235B-A22B	0.381	0.879	0.751	0.946
	Qwen3.5-0.8B-Instruct vs. Qwen3.5-2B-Instruct	0.932	0.916	0.932	0.916
	Qwen3.5-0.8B-Instruct vs. Qwen3.5-4B-Instruct	0.915	0.874	0.915	0.874
	Qwen3.5-0.8B-Instruct vs. Qwen3.5-9B-Instruct	0.180	0.427	0.874	0.722
	Qwen3.5-0.8B-Instruct vs. Qwen3.5-27B-Instruct	0.243	0.468	0.859	0.672
	Qwen3.5-2B-Instruct vs. Qwen3.5-4B-Instruct	0.949	0.959	0.949	0.959
	Qwen3.5-2B-Instruct vs. Qwen3.5-9B-Instruct	0.157	0.377	0.895	0.828
	Qwen3.5-2B-Instruct vs. Qwen3.5-27B-Instruct	0.227	0.414	0.880	0.789
	Qwen3.5-4B-Instruct vs. Qwen3.5-9B-Instruct	0.162	0.423	0.923	0.890
	Qwen3.5-4B-Instruct vs. Qwen3.5-27B-Instruct	0.231	0.444	0.912	0.862
	Qwen3.5-9B-Instruct vs. Qwen3.5-27B-Instruct	0.408	0.962	0.946	0.985
	Qwen3.5-35B-A3B-Instruct vs. Qwen3.5-122B-A10B-Instruct	0.412	0.936	0.904	0.963
	Qwen3.5-35B-A3B-Instruct vs. Qwen3.5-397B-A17B-Instruct	0.414	0.931	0.831	0.914
	Qwen3.5-122B-A10B-Instruct vs. Qwen3.5-397B-A17B-Instruct	0.499	0.967	0.818	0.942
	Gemma-2-2B-Instruct vs. Gemma-2-9B-Instruct	0.697	0.762	0.697	0.762
	Gemma-2-2B-Instruct vs. Gemma-2-27B-Instruct	0.392	0.105	0.392	0.104
	Gemma-2-9B-Instruct vs. Gemma-2-27B-Instruct	0.368	0.091	0.368	0.091
	Gemma-3-1B-Instruct vs. Gemma-3-4B-Instruct	0.668	0.759	0.668	0.759
	Gemma-3-1B-Instruct vs. Gemma-3-12B-Instruct	0.514	0.671	0.514	0.671
	Gemma-3-1B-Instruct vs. Gemma-3-27B-Instruct	0.451	0.629	0.451	0.629
	Gemma-3-4B-Instruct vs. Gemma-3-12B-Instruct	0.756	0.864	0.756	0.864
	Gemma-3-4B-Instruct vs. Gemma-3-27B-Instruct	0.711	0.831	0.712	0.831

Continued on next page...

Table 15 (continued)

Task	Model Pair (A vs. B)	$E_{\cos}$	E_{euc}	$U_{\cos}$	U_{euc}
	Gemma-3-12B-Instruct vs. Gemma-3-27B-Instruct	0.834	0.916	0.835	0.916
	Gemma-4-E2B-Instruct vs. Gemma-4-E4B-Instruct	0.768	0.918	0.768	0.918
	Gemma-4-E2B-Instruct vs. Gemma-4-26B-A4B-Instruct	0.736	0.795	0.736	0.795
	Gemma-4-E2B-Instruct vs. Gemma-4-31B-Instruct	0.737	0.689	0.737	0.689
	Gemma-4-E4B-Instruct vs. Gemma-4-26B-A4B-Instruct	0.760	0.858	0.760	0.858
	Gemma-4-E4B-Instruct vs. Gemma-4-31B-Instruct	0.743	0.737	0.743	0.737
	Gemma-4-26B-A4B-Instruct vs. Gemma-4-31B-Instruct	0.760	0.760	0.760	0.759
	MiniMax-M2.5-Chat vs. MiniMax-M2.7	0.992	0.997	0.976	0.985
	GLM-5-744B vs. GLM-5.1	0.996	1.000	1.000	1.000
	DeepSeek-V2-Lite vs. DeepSeek-V2-Lite-Chat	1.000	1.000	1.000	1.000
	DeepSeek-V3-Base vs. DeepSeek-V3	1.000	1.000	1.000	1.000
	DeepSeek-V3-Base vs. DeepSeek-V3.1-Base	1.000	1.000	1.000	1.000
	DeepSeek-V3-Base vs. DeepSeek-V3.1	1.000	1.000	1.000	1.000
	DeepSeek-V3-Base vs. DeepSeek-V3.2	1.000	1.000	1.000	1.000
	DeepSeek-V3-Base vs. DeepSeek-V3.2-Speciale	1.000	1.000	1.000	1.000
	DeepSeek-V3 vs. DeepSeek-V3.1-Base	1.000	1.000	1.000	1.000
	DeepSeek-V3 vs. DeepSeek-V3.1	1.000	1.000	1.000	1.000
	DeepSeek-V3 vs. DeepSeek-V3.2	1.000	1.000	1.000	1.000
	DeepSeek-V3 vs. DeepSeek-V3.2-Speciale	1.000	1.000	1.000	1.000
	DeepSeek-V3.1-Base vs. DeepSeek-V3.1	1.000	1.000	1.000	1.000
	DeepSeek-V3.1-Base vs. DeepSeek-V3.2	1.000	1.000	1.000	1.000
	DeepSeek-V3.1-Base vs. DeepSeek-V3.2-Speciale	1.000	1.000	1.000	1.000
	DeepSeek-V3.1 vs. DeepSeek-V3.2	1.000	1.000	1.000	1.000
	DeepSeek-V3.1 vs. DeepSeek-V3.2-Speciale	1.000	1.000	1.000	1.000
	DeepSeek-V3.2 vs. DeepSeek-V3.2-Speciale	1.000	1.000	1.000	1.000
	DeepSeek-V4-Flash vs. DeepSeek-V4-Pro	0.783	0.964	0.834	0.941
	DeepSeek-R1 vs. DeepSeek-R1-0528	1.000	1.000	1.000	1.000
	DeepSeek-R1-Distill-Qwen-1.5B vs. DeepSeek-R1-Distill-Qwen-7B	0.223	0.897	0.953	0.930
	DeepSeek-R1-Distill-Qwen-1.5B vs. DeepSeek-R1-Distill-Qwen-14B	0.154	0.818	0.624	0.519
	DeepSeek-R1-Distill-Qwen-1.5B vs. DeepSeek-R1-Distill-Qwen-32B	0.161	0.809	0.661	0.538
	DeepSeek-R1-Distill-Qwen-7B vs. DeepSeek-R1-Distill-Qwen-14B	0.253	0.867	0.674	0.638
	DeepSeek-R1-Distill-Qwen-7B vs. DeepSeek-R1-Distill-Qwen-32B	0.264	0.872	0.718	0.653
	DeepSeek-R1-Distill-Qwen-14B vs. DeepSeek-R1-Distill-Qwen-32B	0.501	0.993	0.903	0.979
	DeepSeek-R1-Distill-Llama-8B vs. DeepSeek-R1-Distill-Llama-70B	0.405	0.878	0.843	0.888
Cross-Scale	Qwen3-0.6B vs. Qwen2.5-0.5B-Instruct	0.783	0.876	0.783	0.876
	Qwen3-0.6B vs. Qwen2.5-1.5B-Instruct	0.861	0.915	0.861	0.915
	Qwen3-0.6B vs. Qwen2.5-3B-Instruct	0.850	0.892	0.850	0.892
	Qwen3-0.6B vs. Llama-3.2-1B-Instruct	0.623	0.528	0.623	0.528
	Qwen3-0.6B vs. Llama-3.2-3B-Instruct	0.619	0.468	0.619	0.468
	Qwen3-0.6B vs. Gemma-2-2B-Instruct	0.595	0.291	0.595	0.291
	Qwen3-0.6B vs. Gemma-3-4B-Instruct	0.312	0.177	0.312	0.177

Continued on next page...

Table 15 (continued)

Task	Model Pair (A vs. B)	$E_{\cos}$	E_{euc}	$U_{\cos}$	U_{euc}
	Qwen3-0.6B vs. Qwen3.5-0.8B-Instruct	0.634	0.604	0.634	0.604
	Qwen3-0.6B vs. Qwen3.5-2B-Instruct	0.628	0.599	0.628	0.599
	Qwen3-0.6B vs. Qwen3.5-4B-Instruct	0.589	0.583	0.589	0.583
	Qwen3-1.7B vs. Qwen2.5-0.5B-Instruct	0.723	0.850	0.723	0.850
	Qwen3-1.7B vs. Qwen2.5-1.5B-Instruct	0.848	0.933	0.848	0.933
	Qwen3-1.7B vs. Qwen2.5-3B-Instruct	0.848	0.948	0.848	0.948
	Qwen3-1.7B vs. Llama-3.2-1B-Instruct	0.641	0.617	0.641	0.617
	Qwen3-1.7B vs. Llama-3.2-3B-Instruct	0.628	0.600	0.628	0.600
	Qwen3-1.7B vs. Gemma-2-2B-Instruct	0.619	0.124	0.619	0.124
	Qwen3-1.7B vs. Gemma-3-4B-Instruct	0.364	0.025	0.364	0.025
	Qwen3-1.7B vs. Qwen3.5-0.8B-Instruct	0.623	0.568	0.623	0.568
	Qwen3-1.7B vs. Qwen3.5-2B-Instruct	0.625	0.611	0.625	0.611
	Qwen3-1.7B vs. Qwen3.5-4B-Instruct	0.599	0.628	0.599	0.628
	Qwen3-4B vs. Qwen2.5-0.5B-Instruct	0.820	0.848	0.820	0.848
	Qwen3-4B vs. Qwen2.5-1.5B-Instruct	0.895	0.916	0.895	0.916
	Qwen3-4B vs. Qwen2.5-3B-Instruct	0.898	0.940	0.898	0.940
	Qwen3-4B vs. Llama-3.2-1B-Instruct	0.709	0.714	0.709	0.714
	Qwen3-4B vs. Llama-3.2-3B-Instruct	0.710	0.724	0.710	0.724
	Qwen3-4B vs. Gemma-2-2B-Instruct	0.653	0.133	0.653	0.133
	Qwen3-4B vs. Gemma-3-4B-Instruct	0.283	0.040	0.283	0.040
	Qwen3-4B vs. Qwen3.5-0.8B-Instruct	0.651	0.607	0.651	0.607
	Qwen3-4B vs. Qwen3.5-2B-Instruct	0.665	0.649	0.665	0.649
	Qwen3-4B vs. Qwen3.5-4B-Instruct	0.637	0.677	0.637	0.677
	Qwen2.5-0.5B-Instruct vs. Llama-3.2-1B-Instruct	0.736	0.456	0.736	0.456
	Qwen2.5-0.5B-Instruct vs. Llama-3.2-3B-Instruct	0.726	0.400	0.726	0.400
	Qwen2.5-0.5B-Instruct vs. Gemma-2-2B-Instruct	0.680	0.333	0.680	0.333
	Qwen2.5-0.5B-Instruct vs. Gemma-3-4B-Instruct	0.160	0.190	0.160	0.190
	Qwen2.5-0.5B-Instruct vs. Qwen3.5-0.8B-Instruct	0.722	0.612	0.722	0.612
	Qwen2.5-0.5B-Instruct vs. Qwen3.5-2B-Instruct	0.736	0.644	0.736	0.644
	Qwen2.5-0.5B-Instruct vs. Qwen3.5-4B-Instruct	0.721	0.628	0.721	0.628
	Qwen2.5-1.5B-Instruct vs. Llama-3.2-1B-Instruct	0.752	0.608	0.752	0.608
	Qwen2.5-1.5B-Instruct vs. Llama-3.2-3B-Instruct	0.744	0.567	0.744	0.567
	Qwen2.5-1.5B-Instruct vs. Gemma-2-2B-Instruct	0.704	0.227	0.704	0.227
	Qwen2.5-1.5B-Instruct vs. Gemma-3-4B-Instruct	0.229	0.095	0.229	0.095
	Qwen2.5-1.5B-Instruct vs. Qwen3.5-0.8B-Instruct	0.717	0.585	0.717	0.585
	Qwen2.5-1.5B-Instruct vs. Qwen3.5-2B-Instruct	0.728	0.629	0.728	0.629
	Qwen2.5-1.5B-Instruct vs. Qwen3.5-4B-Instruct	0.707	0.629	0.707	0.629
	Qwen2.5-3B-Instruct vs. Llama-3.2-1B-Instruct	0.760	0.654	0.760	0.654
	Qwen2.5-3B-Instruct vs. Llama-3.2-3B-Instruct	0.751	0.637	0.751	0.637
	Qwen2.5-3B-Instruct vs. Gemma-2-2B-Instruct	0.696	0.151	0.696	0.151
	Qwen2.5-3B-Instruct vs. Gemma-3-4B-Instruct	0.260	0.038	0.260	0.038

Continued on next page...

Table 15 (continued)

Task	Model Pair (A vs. B)	$E_{\cos}$	E_{euc}	$U_{\cos}$	U_{euc}
	Qwen2.5-3B-Instruct vs. Qwen3.5-0.8B-Instruct	0.699	0.560	0.699	0.560
	Qwen2.5-3B-Instruct vs. Qwen3.5-2B-Instruct	0.711	0.617	0.711	0.617
	Qwen2.5-3B-Instruct vs. Qwen3.5-4B-Instruct	0.702	0.631	0.702	0.631
	Llama-3.2-1B-Instruct vs. Gemma-2-2B-Instruct	0.675	-0.004	0.675	-0.004
	Llama-3.2-1B-Instruct vs. Gemma-3-4B-Instruct	0.181	0.082	0.181	0.082
	Llama-3.2-1B-Instruct vs. Qwen3.5-0.8B-Instruct	0.796	0.716	0.796	0.716
	Llama-3.2-1B-Instruct vs. Qwen3.5-2B-Instruct	0.804	0.733	0.804	0.733
	Llama-3.2-1B-Instruct vs. Qwen3.5-4B-Instruct	0.823	0.745	0.823	0.745
	Llama-3.2-3B-Instruct vs. Gemma-2-2B-Instruct	0.654	-0.028	0.654	-0.027
	Llama-3.2-3B-Instruct vs. Gemma-3-4B-Instruct	0.140	0.029	0.140	0.029
	Llama-3.2-3B-Instruct vs. Qwen3.5-0.8B-Instruct	0.794	0.657	0.794	0.657
	Llama-3.2-3B-Instruct vs. Qwen3.5-2B-Instruct	0.809	0.701	0.809	0.701
	Llama-3.2-3B-Instruct vs. Qwen3.5-4B-Instruct	0.826	0.736	0.826	0.736
	Gemma-2-2B-Instruct vs. Gemma-3-4B-Instruct	0.312	0.220	0.312	0.220
	Gemma-2-2B-Instruct vs. Qwen3.5-0.8B-Instruct	0.704	0.125	0.704	0.125
	Gemma-2-2B-Instruct vs. Qwen3.5-2B-Instruct	0.725	0.108	0.725	0.108
	Gemma-2-2B-Instruct vs. Qwen3.5-4B-Instruct	0.731	0.090	0.731	0.090
	Gemma-3-4B-Instruct vs. Qwen3.5-0.8B-Instruct	0.223	0.187	0.223	0.187
	Gemma-3-4B-Instruct vs. Qwen3.5-2B-Instruct	0.157	0.101	0.157	0.101
	Gemma-3-4B-Instruct vs. Qwen3.5-4B-Instruct	0.168	0.065	0.168	0.065
	Qwen3-8B vs. Qwen2.5-7B-Instruct	0.286	0.941	0.749	0.696
	Qwen3-8B vs. Qwen2.5-14B-Instruct	0.299	0.961	0.780	0.925
	Qwen3-8B vs. Llama-3.1-8B-Instruct	0.285	0.117	0.613	0.713
	Qwen3-8B vs. Gemma-2-9B-Instruct	0.285	-0.280	0.496	-0.115
	Qwen3-8B vs. Gemma-3-12B-Instruct	0.414	-0.078	0.104	-0.151
	Qwen3-8B vs. Qwen3.5-9B-Instruct	0.268	0.733	0.630	0.718
	Qwen3-14B vs. Qwen2.5-7B-Instruct	0.311	0.922	0.763	0.718
	Qwen3-14B vs. Qwen2.5-14B-Instruct	0.322	0.958	0.870	0.952
	Qwen3-14B vs. Llama-3.1-8B-Instruct	0.313	0.059	0.620	0.782
	Qwen3-14B vs. Gemma-2-9B-Instruct	0.278	-0.266	0.450	-0.077
	Qwen3-14B vs. Gemma-3-12B-Instruct	0.449	-0.061	0.227	-0.168
	Qwen3-14B vs. Qwen3.5-9B-Instruct	0.290	0.699	0.541	0.727
	Qwen2.5-7B-Instruct vs. Llama-3.1-8B-Instruct	0.307	0.249	0.716	0.707
	Qwen2.5-7B-Instruct vs. Gemma-2-9B-Instruct	0.286	-0.272	0.520	0.285
	Qwen2.5-7B-Instruct vs. Gemma-3-12B-Instruct	0.439	0.010	-0.016	0.047

Continued on next page...

Table 15 (continued)

Task	Model Pair (A vs. B)	$E_{\cos}$	E_{euc}	$U_{\cos}$	U_{euc}
	Qwen2.5-7B-Instruct vs. Qwen3.5-9B-Instruct	0.285	0.688	0.632	0.583
	Qwen2.5-14B-Instruct vs. Llama-3.1-8B-Instruct	0.274	0.191	0.612	0.816
	Qwen2.5-14B-Instruct vs. Gemma-2-9B-Instruct	0.241	-0.276	0.421	-0.006
	Qwen2.5-14B-Instruct vs. Gemma-3-12B-Instruct	0.372	-0.074	0.222	-0.140
	Qwen2.5-14B-Instruct vs. Qwen3.5-9B-Instruct	0.260	0.688	0.496	0.741
	Llama-3.1-8B-Instruct vs. Gemma-2-9B-Instruct	0.311	0.056	0.459	0.073
	Llama-3.1-8B-Instruct vs. Gemma-3-12B-Instruct	0.548	0.176	-0.007	-0.077
	Llama-3.1-8B-Instruct vs. Qwen3.5-9B-Instruct	0.322	0.527	0.784	0.806
	Gemma-2-9B-Instruct vs. Gemma-3-12B-Instruct	0.261	0.244	0.261	0.244
	Gemma-2-9B-Instruct vs. Qwen3.5-9B-Instruct	0.288	-0.323	0.435	-0.096
	Gemma-3-12B-Instruct vs. Qwen3.5-9B-Instruct	0.474	-0.085	0.002	-0.122
	Qwen3-32B vs. Qwen2.5-32B-Instruct	0.337	0.964	0.894	0.943
	Qwen3-32B vs. Qwen2.5-72B-Instruct	0.289	0.921	0.838	0.723
	Qwen3-32B vs. Llama-3.1-70B-Instruct	0.371	0.059	0.717	0.777
	Qwen3-32B vs. Gemma-2-27B-Instruct	0.282	-0.092	0.220	0.068
	Qwen3-32B vs. Gemma-3-27B-Instruct	0.423	-0.086	0.094	-0.118
	Qwen3-32B vs. Qwen3.5-27B-Instruct	0.300	0.853	0.620	0.744
	Qwen3-235B-A22B vs. Qwen2.5-32B-Instruct	0.285	0.877	0.759	0.909
	Qwen3-235B-A22B vs. Qwen2.5-72B-Instruct	0.311	0.901	0.645	0.732
	Qwen3-235B-A22B vs. Llama-3.1-70B-Instruct	0.423	0.396	0.519	0.716
	Qwen3-235B-A22B vs. Gemma-2-27B-Instruct	0.231	-0.081	0.324	0.036
	Qwen3-235B-A22B vs. Gemma-3-27B-Instruct	0.393	-0.078	0.368	-0.138
	Qwen3-235B-A22B vs. Qwen3.5-27B-Instruct	0.282	0.816	0.496	0.676
	Qwen2.5-32B-Instruct vs. Llama-3.1-70B-Instruct	0.341	0.159	0.671	0.843
	Qwen2.5-32B-Instruct vs. Gemma-2-27B-Instruct	0.241	-0.100	0.214	0.061
	Qwen2.5-32B-Instruct vs. Gemma-3-27B-Instruct	0.350	-0.105	0.083	-0.119
	Qwen2.5-32B-Instruct vs. Qwen3.5-27B-Instruct	0.271	0.831	0.543	0.769
	Qwen2.5-72B-Instruct vs. Llama-3.1-70B-Instruct	0.393	0.212	0.616	0.699
	Qwen2.5-72B-Instruct vs. Gemma-2-27B-Instruct	0.224	-0.107	0.142	0.178
	Qwen2.5-72B-Instruct vs. Gemma-3-27B-Instruct	0.385	-0.067	-0.059	0.041
	Qwen2.5-72B-Instruct vs. Qwen3.5-27B-Instruct	0.258	0.819	0.569	0.579
	Llama-3.1-70B-Instruct vs. Gemma-2-27B-Instruct	0.339	0.109	0.241	0.095

Continued on next page...

Table 15 (continued)

Task	Model Pair (A vs. B)	$E_{\cos}$	E_{euc}	$U_{\cos}$	U_{euc}
	Llama-3.1-70B-Instruct vs. Gemma-3-27B-Instruct	0.543	0.151	0.045	-0.113
	Llama-3.1-70B-Instruct vs. Qwen3.5-27B-Instruct	0.364	0.325	0.700	0.757
	Gemma-2-27B-Instruct vs. Gemma-3-27B-Instruct	0.209	0.243	0.209	0.243
	Gemma-2-27B-Instruct vs. Qwen3.5-27B-Instruct	0.298	-0.058	0.176	0.086
	Gemma-3-27B-Instruct vs. Qwen3.5-27B-Instruct	0.425	-0.116	-0.033	-0.133
MoE Cross-Family	Qwen3-30B-A3B vs. Qwen3.5-35B-A3B-Instruct	0.379	0.707	0.733	0.766
	Qwen3-30B-A3B vs. Qwen3.6-35B-A3B	0.379	0.710	0.746	0.758
	Qwen3-30B-A3B vs. DeepSeek-V2-Lite-Chat	0.464	0.649	0.572	0.728
	Qwen3-30B-A3B vs. DeepSeek-R1-0528	0.449	0.404	0.382	0.793
	Qwen3-30B-A3B vs. MiniMax-M2.7	0.413	0.425	0.661	0.702
	Qwen3.5-35B-A3B-Instruct vs. Qwen3.6-35B-A3B	0.953	0.997	0.994	0.990
	Qwen3.5-35B-A3B-Instruct vs. DeepSeek-V2-Lite-Chat	0.458	0.527	0.546	0.641
	Qwen3.5-35B-A3B-Instruct vs. DeepSeek-R1-0528	0.462	0.321	0.471	0.695
	Qwen3.5-35B-A3B-Instruct vs. MiniMax-M2.7	0.410	0.199	0.763	0.745
	Qwen3.6-35B-A3B vs. DeepSeek-V2-Lite-Chat	0.458	0.531	0.544	0.614
	Qwen3.6-35B-A3B vs. DeepSeek-R1-0528	0.462	0.320	0.464	0.690
	Qwen3.6-35B-A3B vs. MiniMax-M2.7	0.410	0.204	0.769	0.716
	DeepSeek-V2-Lite-Chat vs. DeepSeek-R1-0528	0.568	0.119	0.449	0.623
	DeepSeek-V2-Lite-Chat vs. MiniMax-M2.7	0.505	0.564	0.481	0.793
	DeepSeek-R1-0528 vs. MiniMax-M2.7	0.549	-0.184	0.484	0.560

B.3.1 Base–Instruct input-side GCorr summary (Insight 1 supplement)

The Base–Instruct archive contains 35 pairs. Excluding Gemma-3-1B and four Gemma-4 pairs yields the 30-pair comparable subset: 13 untied and 17 tied pairs.

On the Base/Instruct rows of Table 15, we summarize input-side Base–Instruct GCorr under the same convention as the static-spectrum analysis: **13 untied pairs** and **17 tied pairs** (excluding the Gemma-4 family and Gemma-3-1B; see the model-inventory notes above). The summary is in Table 16.

Table 16: Base–Instruct input-side GCorr summary (13 untied / 17 tied).

Group	n	Metric	mean	range [min, max]
untied	13	E_{euc}	1.000	[0.9998, 1.0000]
untied	13	$E_{\cos}$	0.997	[0.987, 1.000]
tied	17	$E_{\cos}$	0.994	[0.979, 1.000]
tied	17	E_{euc}	0.983	[0.894, 1.000]

Reading guide. Among the 13 untied pairs, E_{euc} is essentially unity ($13/13 \geq 0.999$), indicating that independent input embeddings preserve stable Euclidean token–token geometry across Base–Instruct evolution. Among the 17 tied pairs, $E_{\cos}$ remains very high (mean 0.994, same order as untied), but E_{euc} shows clear spread (range 0.106; mean 0.017 lower than untied); Euclidean distance relations change despite high cosine agreement, and 12/17 pairs satisfy $E_{\cos} > E_{\text{euc}}$. The

distance to 1 ($\delta = 1 - \text{mean}$) differs by about $480\times$ between untied and tied E_{euc} (3.6×10^{-5} vs. 1.7×10^{-2}). This supplements Insight 1: tying makes it harder to preserve the stable E-side Euclidean structure of untied input embeddings while keeping cosine directions high; it does not replace the primary 37/37 and static-spectrum evidence.

B.3.2 CROSS-MODEL U–U VS. E–E GCORR SUMMARY (INSIGHT 2)

We separate the 110 Model Series pairs by the tying status of their two models (Table 17). The 37 tied–untied comparisons support Insight 1 under all three metrics; the 48 untied–untied comparisons isolate the cross-model cosine commonality of output spaces. Tied–tied pairs have identical E/U matrices, so their small differences between separately estimated GCorr values are not counted as output-side gains. For untied–untied pairs, cosine favors U on average, while Euclidean and squared-Euclidean GCorr favor E. The 13 pairs with lower U-side cosine GCorr are listed in Table 18.

For the cross-series extension, we select all 26 pairs with two untied models from a 176-pair comparison set constructed across model series within size groups ($\leq$4B, 4B–27B exclusive, and $\geq$27B). These pairs span six combinations of Qwen2.5, Qwen3, Qwen3.5 dense, and Llama, with no pair overlap with the 48 within-series comparisons. All use token-string alignment. Cosine GCorr favors U in 26/26 pairs (Table 3); Euclidean means are 0.6103 for E and 0.7596 for U (13/26 favor U), and squared-Euclidean means are 0.5955 and 0.7533 (14/26 favor U).

Table 17: Within-series GCorr by pairing type. Values are group means; counts use per-pair U-minus-E means. Tied–tied direction counts are omitted because E=U.

Group	Metric	E–E	U–U	U > E
Tied–Untied (37)	Cosine	0.1395	0.8114	37/37
	Euclidean	0.5553	0.8169	37/37
	Squared Euclidean	0.5124	0.7982	37/37
Untied–Untied (48)	Cosine	0.6095	0.8907	35/48
	Euclidean	0.9446	0.9208	17/48
	Squared Euclidean	0.9369	0.9122	17/48
Tied–Tied (25)	Cosine	0.7548	0.7549	–
	Euclidean	0.7845	0.7845	–
	Squared Euclidean	0.7365	0.7365	–

Table 18: Untied–untied pairs with lower U-side cosine GCorr. The signed gap uses unrounded values.

Model pair	$E_{\cos}$	$U_{\cos}$	$U - E$
MiniMax-M2.5-Chat vs. MiniMax-M2.7	0.992481	0.975786	-1.67×10^{-2}
DeepSeek-V3-Base vs. DeepSeek-V3	0.999998	0.999997	-1.04×10^{-6}
DeepSeek-V3-Base vs. DeepSeek-V3.1-Base	0.999933	0.999747	-1.86×10^{-4}
DeepSeek-V3-Base vs. DeepSeek-V3.1	0.999929	0.999716	-2.13×10^{-4}
DeepSeek-V3-Base vs. DeepSeek-V3.2	0.999851	0.999713	-1.39×10^{-4}
DeepSeek-V3-Base vs. DeepSeek-V3.2-Speciale	0.999852	0.999707	-1.45×10^{-4}
DeepSeek-V3 vs. DeepSeek-V3.1-Base	0.999931	0.999753	-1.78×10^{-4}
DeepSeek-V3 vs. DeepSeek-V3.1	0.999927	0.999727	-2.00×10^{-4}
DeepSeek-V3 vs. DeepSeek-V3.2	0.999849	0.999723	-1.26×10^{-4}
DeepSeek-V3 vs. DeepSeek-V3.2-Speciale	0.999849	0.999717	-1.33×10^{-4}
DeepSeek-V3.1-Base vs. DeepSeek-V3.1	0.999994	0.999986	-8.71×10^{-6}
DeepSeek-V3.2 vs. DeepSeek-V3.2-Speciale	0.999998	0.999993	-4.48×10^{-6}
DeepSeek-R1 vs. DeepSeek-R1-0528	0.999995	0.999988	-6.99×10^{-6}

B.4 PRETRAINING EXPERIMENT DETAILS

Paper IDs group the four input-side variants as S3–S6. For existing code and logs, paper S1–S12 correspond respectively to historical IDs `s1, s2, s3, s4, s6, s12, s5, s7, s8, s9, s10, s11`; experimental records retain those historical names.

This section supplements pretraining experiment settings compressed in Section 5. Larger-scale pretraining and SFT settings are specified in Appendix C. All variants use the same lightweight dense decoder-only Transformer backbone with hidden size $d = 768$; low-rank variants uniformly use rank $r = 32$ with `lm_head_bias` enabled. Model architecture follows the Qwen3 small dense line (Yang et al., 2025); experiment implementation and the first pretraining dataset come from the MiniMind (Gong, 2025) open-source project, which offers a mature small-model from-scratch training pipeline, low reproduction cost, and high-quality pretraining data.

B.4.1 VARIANTS AND PARAMETER OVERHEAD

Table 19: Overview of S1–S12 embedding-head variants.

Variant	tied?	Description
S1	yes	Standard tied baseline; input embedding shared with `lm_head.weight`
S2	no	Standard untied baseline; input embedding independent of output head
S3	yes	Input-side hidden-dimensional bias correction, $e - \beta$
S4	yes	Input-side vocabulary-dimensional additive LoRA
S5	yes	Input-side hidden-dimensional multiplicative LoRA
S6	yes	A-LoRA-style input-side affine correction
S7	yes	Output-side vocabulary-dimensional additive LoRA
S8	yes	Output-side hidden-dimensional multiplicative LoRA
S9	no	untied + input-side hidden-dimensional bias correction
S10	no	untied + input-side vocabulary-dimensional additive LoRA
S11	no	untied + input-side hidden-dimensional multiplicative LoRA
S12	yes	Output-side hidden-dimensional bias correction

To define these variants precisely, let vocabulary size be V, hidden dimension be d, and the shared tied input embedding/output unembedding matrix be $W \in \mathbb{R}^{V\times d}$. For input token x_t and final Transformer hidden state h_t, we denote the effective input matrix by E_{eff} and the effective output matrix by U_{eff}:

$$e_t = E_{\text{eff}}[x_t], \qquad z_t = U_{\text{eff}} h_t + b. \tag{15}$$

where $b \in \mathbb{R}^V$ is the output bias enabled uniformly across all variants.

Table 20: Effective input and output matrices for S1–S12 model variants.

Model	Relaxation	E_{eff}	U_{eff}
S1	Standard tied (no relaxation)	W	W
S2	Full untied (complete relaxation)	E	U
S3	Input-side bias correction	$W - \mathbf{1}\beta^\top$	W
S4	Input-side standard additive LoRA relaxation	$W + A_{\text{in}}B_{\text{in}}$	W
S5	Input-side multiplicative LoRA	$W(I_d + sP_{\text{in}}Q_{\text{in}})$	W
S6	A-LoRA-style input-side affine correction	$W(I_d + sP_{\text{in}}Q_{\text{in}}) - \mathbf{1}\beta^\top$	W
S7	Output-side standard additive LoRA relaxation	W	$W + A_{\text{out}}B_{\text{out}}$
S8	Output-side multiplicative LoRA	W	$W(I_d + sP_{\text{out}}Q_{\text{out}})$
S9	untied input-side bias correction	$E - \mathbf{1}\beta_E^\top$	U
S10	untied input-side standard additive LoRA relaxation	$E + A_E B_E$	U
S11	untied input-side multiplicative LoRA	$E(I_d + P_E Q_E)$	U
S12	Output-side bias correction	W	W; logits use $W(h - \beta_U) + b$

where $\mathbf{1} \in \mathbb{R}^V$; $\beta, \beta_E, \beta_U \in \mathbb{R}^d$; additive LoRA parameters satisfy $A_* \in \mathbb{R}^{V\times r}$ and $B_* \in \mathbb{R}^{r\times d}$; multiplicative LoRA parameters satisfy $P_* \in \mathbb{R}^{d\times r}$ and $Q_* \in \mathbb{R}^{r\times d}$, together with a scalar scale $s > 0$. S3/S12 compare input-side versus output-side bias correction; S4/S7 and S5/S8 compare vocabulary-dimensional additive LoRA versus hidden-dimensional multiplicative LoRA on input/output sides; S9–S11 distinguish whether gains come from alleviating tied constraints or from additional reparameterization capacity of input embeddings themselves.

Let vocabulary size be V, hidden size be $H = 768$, and low-rank rank be $r = 32$. MiniMind and FineWeb-Edu/GPT-2 main pretraining experiments use $s = 1$ for all multiplicative LoRA variants (S5/S8/S6); s adds no matrix parameters and only modulates correction strength. In this experiment, hidden-dimensional bias correction adds only H parameters; vocabulary-dimensional additive LoRA adds $Vr + Hr$ parameters; hidden-dimensional multiplicative LoRA adds $2Hr$ parameters; S6 adds $H + 2Hr$ parameters. Table 21 gives parameter-overhead ratios relative to the S1 baseline. Vocabulary-dimensional LoRA cost grows linearly with V, so under the GPT-2 vocabulary S4/S7 overhead is much higher than with the MiniMind tokenizer; hidden-dimensional multiplicative LoRA and S6 depend only on H and r and do not grow with vocabulary size. In implementation, S6 adds both hidden-dimensional bias correction and multiplicative LoRA residual on input lookup vectors (equivalent to the matrix-level affine form $W(I_d + sP_{\text{in}}Q_{\text{in}}) - \mathbf{1}\beta^\top$ with $s = 1$ in the main text); S12 output-side bias correction acts on hidden dimensions, distinct from independent `lm_head_bias`, which is enabled in all variants.

Table 21: Added parameter counts relative to the S1 baseline. S1 parameter counts are 63.9M (MiniMind tokenizer) and 97.6M (GPT-2 tokenizer).

Module/Variant	MiniMind Overhead	MiniMind Ratio	GPT-2 Overhead	GPT-2 Ratio
S3/S12 hidden β	768	0.0012%	768	0.0008%
S4/S7 vocab-dimensional additive LoRA	229,376	0.3589%	1,632,800	1.6722%
S5/S8 hidden multiplicative LoRA	49,152	0.0769%	49,152	0.0503%
S6 A-LoRA-style affine	49,920	0.0781%	49,920	0.0511%
S2 untied head weights	4,915,200	7.6898%	38,597,376	39.5284%
S10 untied + vocab-dimensional LoRA	5,144,576	8.0486%	40,230,176	41.2006%
S11 untied + hidden multiplicative LoRA	4,964,352	7.7667%	38,646,528	39.5788%

Untied reference For FineWeb-Edu/GPT-2, Untied yields validation losses 3.223342, 3.228236, and 3.225045 for seeds 42, 123, and 2026, respectively (mean 3.225541). Its mean paired PPL reduction from the same-seed sequential-split Tied baseline is 0.312882. We compute the recovered fraction as $\overline{\Delta\text{PPL}}_{\text{AGD}}/\overline{\Delta\text{PPL}}_{\text{Untied}}$, with reductions positive. Input-Bias recovers 35.8% and Input-Affine 61.5%, using 768 and 49,920 added parameters versus 38,597,376 for Untied (0.00199% and 0.129% of its added parameters). These are fractions of improvement, not fractions of absolute performance. The untied-variant gradient traces use independently initialized input/output matrices; the S2 validation results use identical initial values for the two matrices.

B.4.2 Data and Training Configuration

All pretraining configurations use AdamW over all model parameters, with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01, and gradient-norm clipping at 1.0. The learning rate follows $\eta(t) = \eta_0[0.1 + 0.45(1 + \cos(\pi t/T))]$, with $\eta_0 = 5 \times 10^{-4}$ and T the total scheduled training steps, without warmup. Gradient accumulation is one step. The fixed-random GPT-2 replication uses the same training recipe as its sequential-split counterpart.

The FineWeb-Edu/GPT-2 experiment additionally records input embedding/output unembedding gradients for analyzing gradient paths in tied and untied architectures. Recorded quantities include input-embedding-path gradients, output-unembedding-path gradients, total gradients on shared weights in tied variants, and gradients of added modules such as S3/S4/S5/S8/S12/S6. The gradient figures in the main text and appendix are aggregated from the three random seeds.

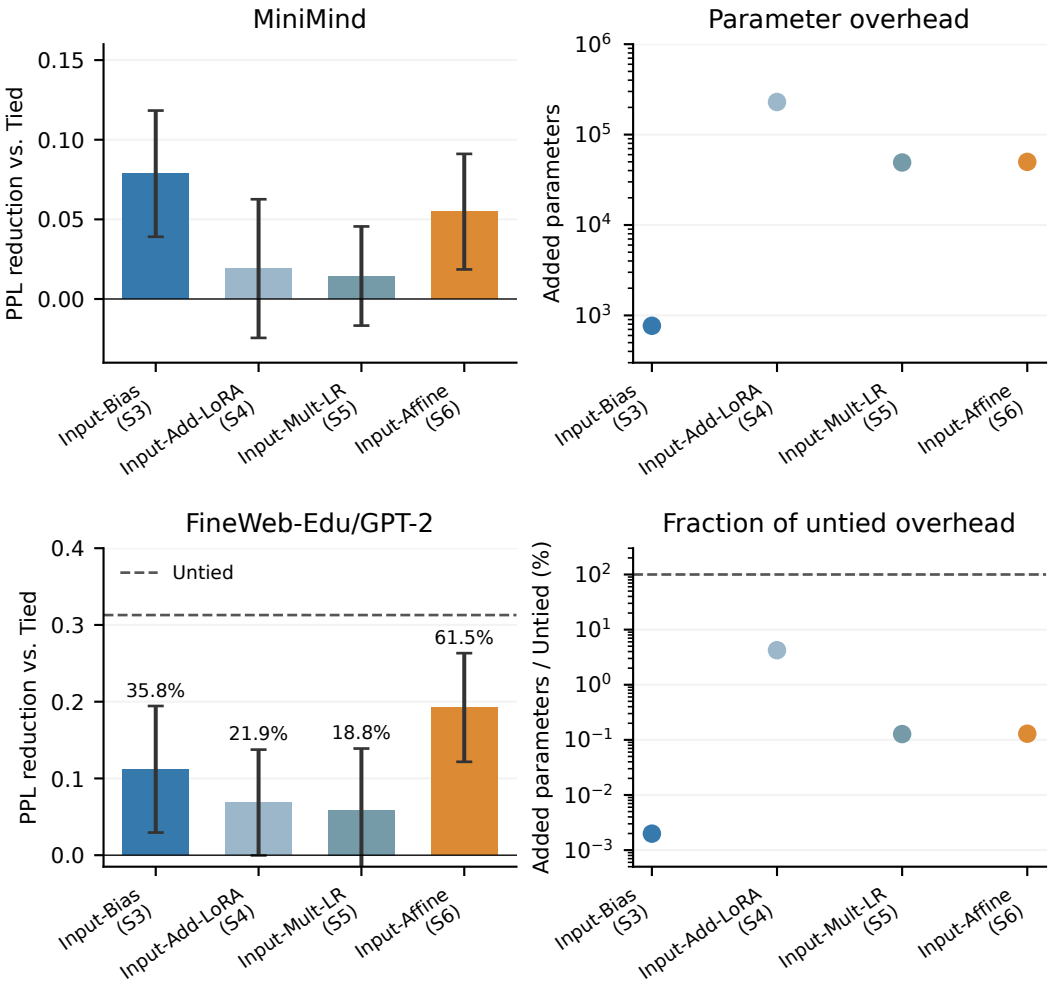


Figure 3: **Pretraining gains and parameter overhead.** Left: mean paired PPL reduction from Tied; error bars are sample SD over three seeds. Right: added parameter counts (MiniMind) or their fraction of the full-untied overhead (FineWeb-Edu/GPT-2). Both panels use the sequential splits of Table 5. The dashed FineWeb-Edu reference is Untied. Percentages above bars give the fraction of its mean PPL reduction.

Table 22: Pretraining experiment configuration for MiniMind original data and FineWeb-Edu/GPT-2.

Setting	MiniMind Original Data	FineWeb-Edu/GPT-2
Training data	MiniMind original pretraining data (Gong, 2025)	FineWeb-Edu pretraining corpus ($\approx$ 6B tokens) (Penedo et al., 2024)
Tokenizer	MiniMind tokenizer	GPT-2 tokenizer (Radford et al., 2019)
Sequence length	340	340
Train/val split	First 99% / last 1% (val start `start_index`=8,384,138)	First 99% / last 1% (val start `start_index`=17,470,587)
Validation sequences	84,689	176,471
Validation tokens	27,069,543	59,766,669
Training variants	S1–S12	S1,S2,S3,S4,S5,S8,S12,S6
Model	Qwen3-aligned dense decoder, 8 layers, $d = 768$	Qwen3-aligned dense decoder, 8 layers, $d = 768$
Low-rank rank	32	32
`lm_head_bias`	on	on
GPU/DDP	8 GPU, `torchrun`	8 GPU, `torchrun`
Batch size	224 per step	80 per step
Effective batch	1,792 sequences	640 sequences
Epochs	2	1
Learning rate	5×10^{-4}	5×10^{-4}
Precision	bfloat16	bfloat16
Seeds	42, 123, 2026	42, 123, 2026

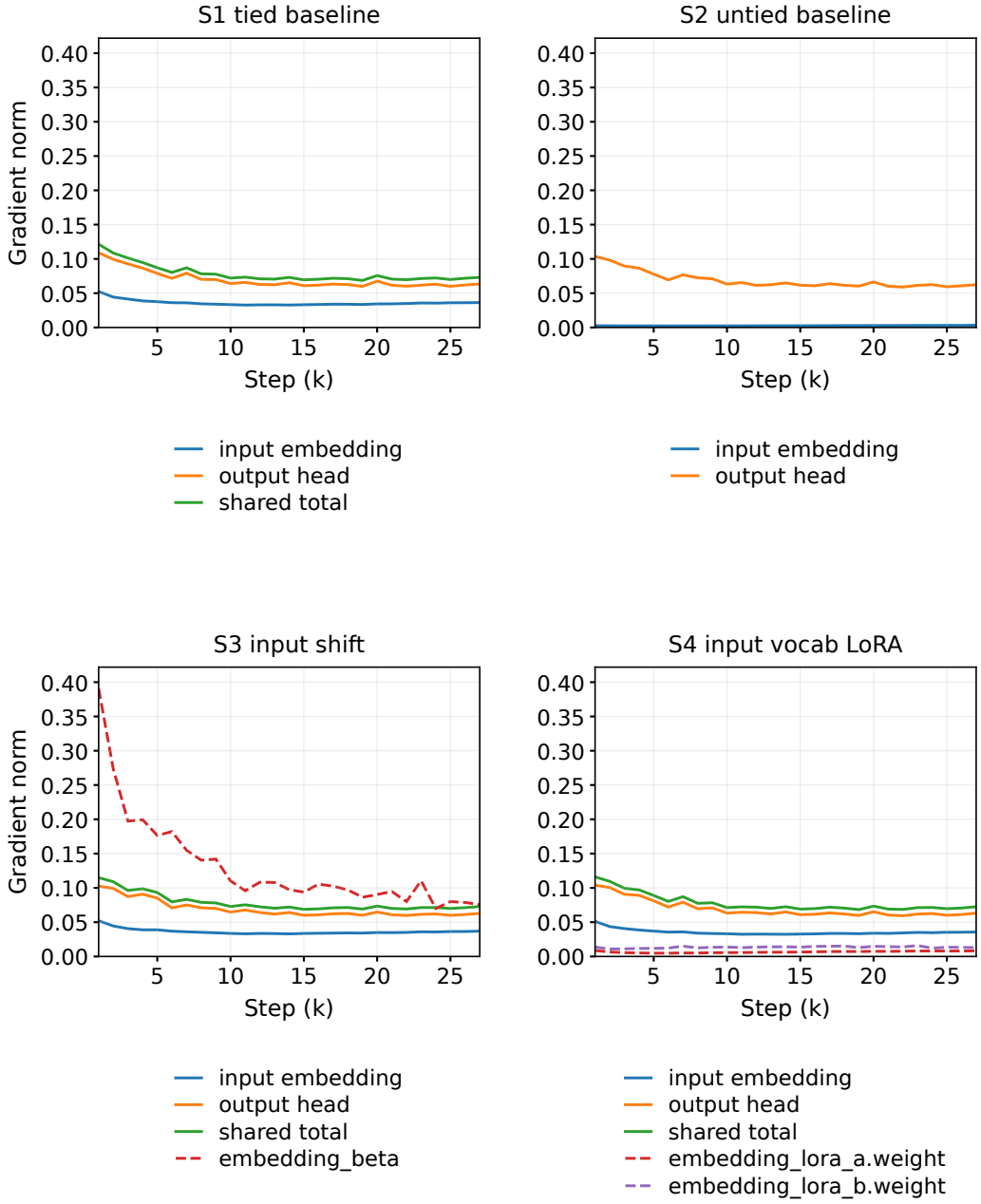


Figure 4: **Gradient-norm overview of baselines and structural variants during FineWeb-Edu/GPT-2 pretraining.** Curves are averaged over seeds 42, 123, and 2026; all subplots share the same y-axis range. Input embedding denotes gradients on input embedding weights from the input path; output head denotes output unembedding-path gradients; shared total denotes total gradients received by the shared vocabulary matrix in tied variants; dashed lines denote gradients of added modules. In the S2 untied baseline, input embedding gradients are much weaker than output unembedding gradients, whereas S3/S6 `embedding_beta` gradients are strong, indicating that the input-side bias module carries a major update signal.

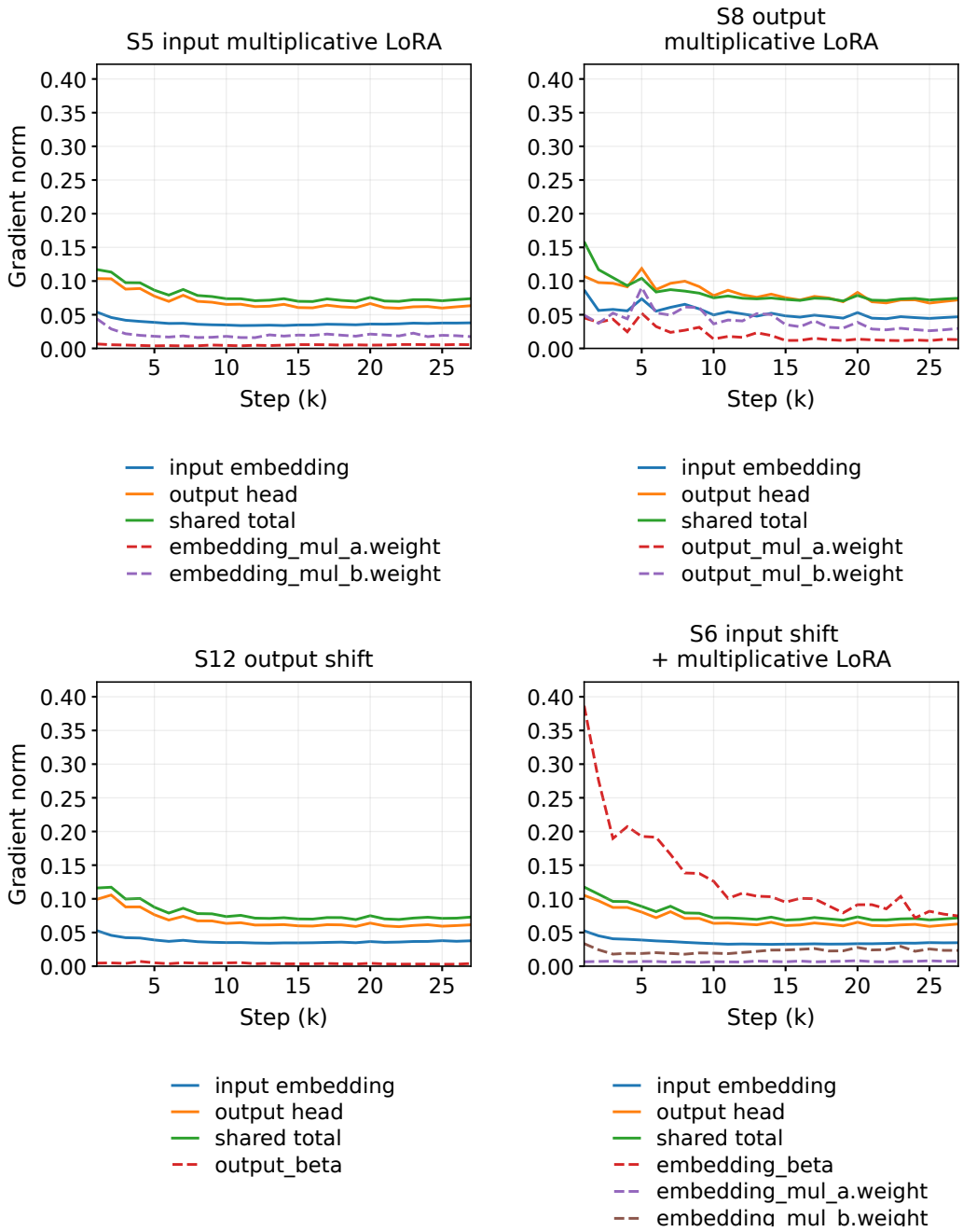


Figure 4: Continued gradient-norm panels: S5/S8/S12/S6. Axes and curve definitions are unchanged from the preceding panels.

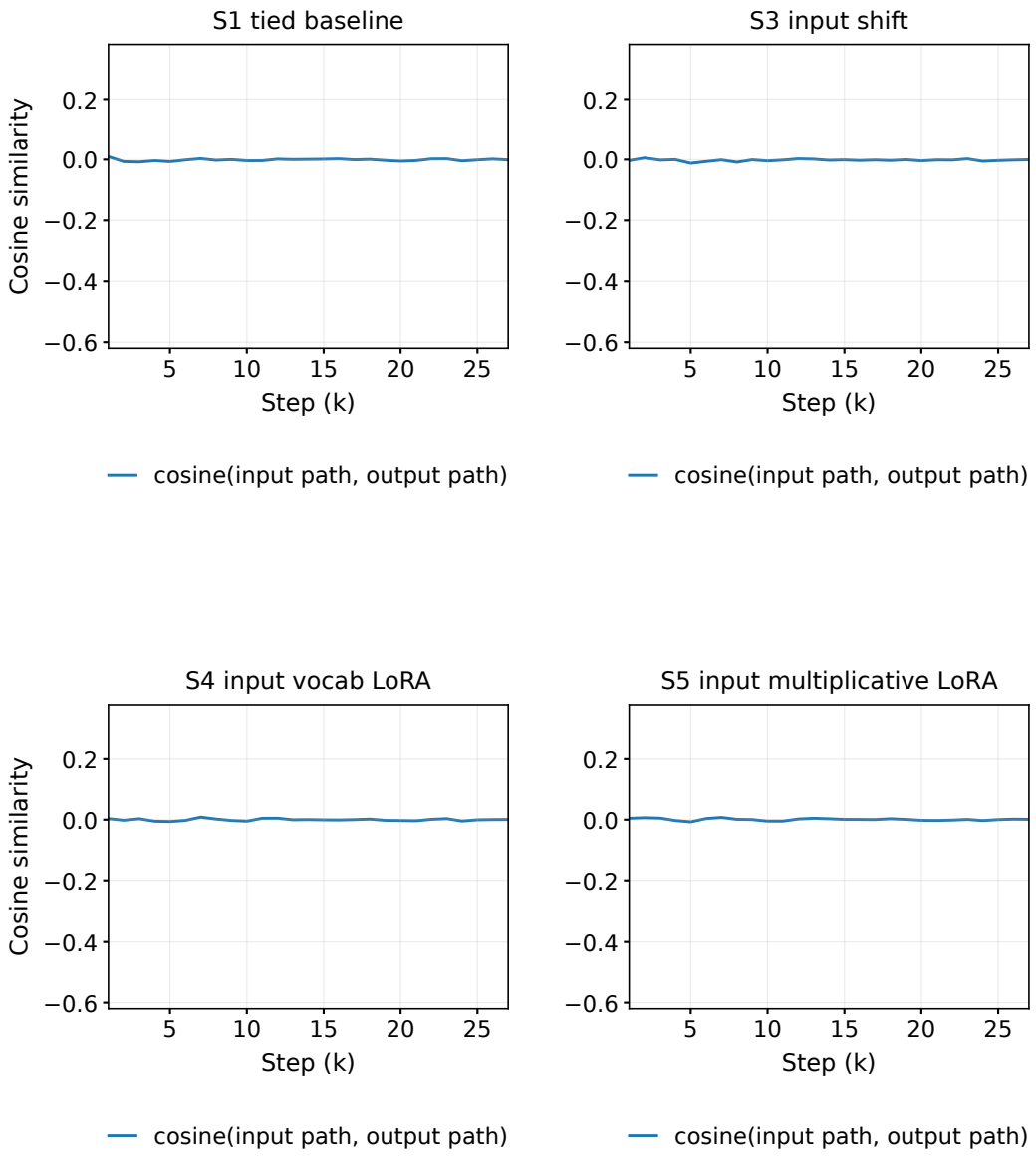


Figure 5: **Input-path versus output-path gradient directions for tied variants during FineWeb-Edu/GPT-2 pretraining.** Curves are averaged over seeds 42, 123, and 2026. Tied split cosine computes cosine similarity between input-embedding-path and output-unembedding-path gradients on the shared vocabulary matrix: near 0 means approximately orthogonal paths, positive means similar directions, negative means conflicting directions. S1/S3/S4/S5/S6 are mostly near 0, indicating no obvious systematic conflict between the two update paths on shared weights; S8 stays negative for long periods at $s = 1$, suggesting output-side multiplicative LoRA makes output-path and input-path gradients clearly oppose each other in shared weight space, which may explain its weaker performance than the S1 baseline.

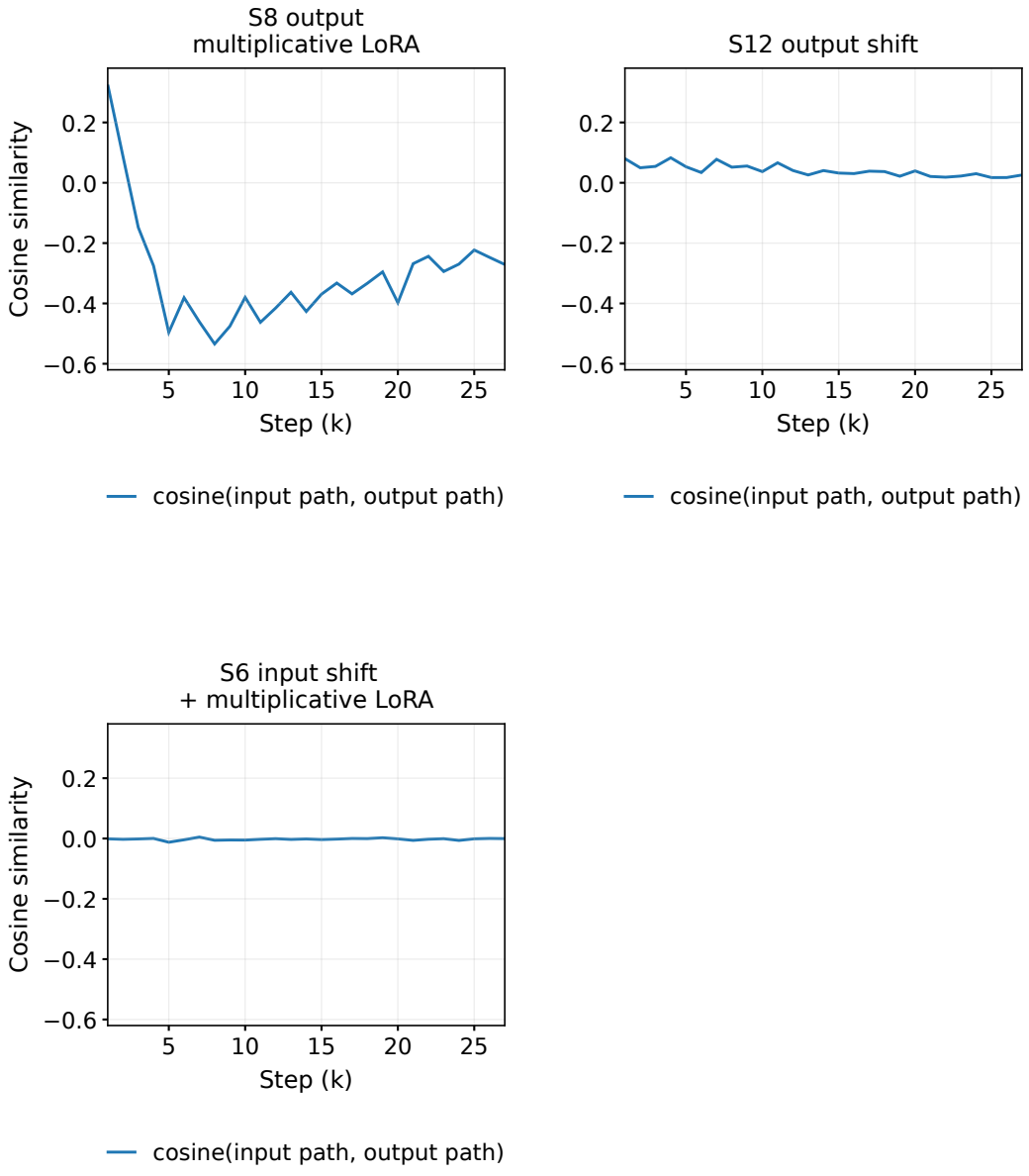


Figure 5: Continued gradient-cosine panels: S8/S12/S6. Axes and curve definitions are unchanged from the preceding panels.

Validation uses the same tokenizer as training; padding positions in labels are set to -100 to ignore loss. The following tables report mean cross-entropy (CE), its sample standard deviation across three seeds, and $\exp(\overline{\mathrm{CE}})$. The Δ column is the mean paired CE difference from same-seed S1; the main table instead reports paired PPL differences and their sample SD.

B.4.3 Full Three-Seed Validation Results

Table 23: Three-seed validation loss for the MiniMind variants. Sorted by mean validation loss from low to high.

Variant	seed42	seed123	seed2026	Mean CE	SD	$\exp(\overline{\mathrm{CE}})$	Δ vs S1
Input-Bias (S3)	1.904187	1.908604	1.897083	1.903291	0.005812	6.707934	-0.011671
Input-Affine (S6)	1.906864	1.903095	1.910587	1.906849	0.003746	6.731843	-0.008113
Input-Add-LoRA (S4)	1.915157	1.904686	1.916560	1.912134	0.006488	6.767515	-0.002828
Input-Mult-LR (S5)	1.919072	1.906038	1.913362	1.912824	0.006534	6.772186	-0.002138
Tied (S1)	1.918825	1.913490	1.912571	1.914962	0.003377	6.786681	0.000000
Output-Bias (S12)	1.915081	1.915418	1.914574	1.915024	0.000425	6.787102	0.000062
Output-Mult-LR (S8)	1.925734	1.908932	1.914385	1.916350	0.008572	6.796107	0.001388
Output-Add-LoRA (S7)	1.918207	1.927948	1.905078	1.917078	0.011477	6.801057	0.002116
Untied+Mult-LR (S11)	1.970678	1.974093	1.980755	1.975175	0.005125	7.207881	0.060213
Untied+Bias (S9)	1.981351	1.975633	1.977136	1.978040	0.002964	7.228561	0.063078
Untied+Add-LoRA (S10)	1.978650	1.981286	1.984940	1.981625	0.003159	7.254522	0.066663

Table 24: Three-seed validation loss for key variants under FineWeb-Edu/GPT-2. Sorted by mean validation loss from low to high.

Variant	seed42	seed123	seed2026	Mean CE	SD	$\exp(\overline{\mathrm{CE}})$	Δ vs S1
Untied (S2)	3.223342	3.228236	3.225045	3.225541	0.002484	25.167186	-0.012355
Input-Affine (S6)	3.228119	3.231683	3.231145	3.230316	0.001921	25.287638	-0.007581
Input-Bias (S3)	3.230755	3.234913	3.234813	3.233494	0.002372	25.368130	-0.004403
Input-Add-LoRA (S4)	3.235644	3.234921	3.235037	3.235201	0.000388	25.411471	-0.002696
Input-Mult-LR (S5)	3.232097	3.238262	3.236377	3.235579	0.003159	25.421078	-0.002318
Output-Bias (S12)	3.233531	3.241727	3.235599	3.236952	0.004262	25.456022	-0.000944
Tied (S1)	3.237538	3.240631	3.235520	3.237896	0.002574	25.480064	0.000000
Output-Mult-LR (S8)	3.238060	3.240361	3.238147	3.238856	0.001304	25.504528	0.000960

The two tables above report the complete three-seed validation results referenced in the main text and appendix figures.

B.5 Fixed-Random FineWeb-Edu/GPT-2 Replication

The replication uses the same eight-layer, $d = 768$ model and GPT-2 tokenizer on approximately 6B packed FineWeb-Edu tokens. A fixed-random 1% validation split (split seed 20260602) contains 176,470 sequences and 59,764,917 effective prediction tokens. For seeds 42, 123, and 2026, respectively, validation losses are $(3.208531, 3.210356, 3.211132)$ for S1, $(3.200723, 3.209470, 3.202984)$ for Input-Bias (S3), and $(3.199437, 3.206056, 3.201113)$ for Input-Affine (S6). Both variants improve in all three pairs. Each comparison uses the original same-seed S1 from this replication. Aggregation follows Table 5; archived evaluation files retain the historical identifier `s12` for manuscript S6.

B.6 Dynamic Centering versus Learned Input Translation

Dynamic centering uses $E_{\mathrm{eff}} = W - \mathbf{1}\mu(W)^\top$, where $\mu(W)$ is the mean of all vocabulary rows, and leaves $U_{\mathrm{eff}} = W$. The mean is recomputed in FP32 at each forward pass and remains differentiable, so the mean term also supplies gradients to rows absent from the input. No parameters are added. Input-Bias instead learns an unconstrained β from the training objective.

The MiniMind comparison uses the eight-layer, $d = 768$ configuration with a fixed-random 1% validation split (split seed 20260602), trained for two epochs with batch size 224 per GPU on eight GPUs. Validation covers 84,688 sequences and 17,625,568 prediction tokens. The 0.6B/FineWeb-Edu comparison uses the 20B-token setup in Appendix C. Within each configuration, centering and its same-seed S1/S3 controls share the data split and training recipe. The two configurations differ in both model and data; their centering results describe configuration dependence.

Table 25: Dynamic-centering ablation: per-seed validation cross-entropy. Paired Δ CE is relative to same-seed S1 (mean $\pm$ sample SD); Wins counts lower-loss seeds than S1.

Variant	seed42	seed123	seed2026	Paired Δ CE	Wins
MiniMind (fixed-random split)					
Tied (S1)	1.741829	1.738371	1.734557	–	–
Dynamic centering	1.734228	1.734176	1.731871	-0.004827 ± 0.002518	3/3
Input-Bias (S3)	1.732819	1.726621	1.732631	-0.007562 ± 0.005070	3/3
Qwen3-0.6B/FineWeb-Edu (20B tokens)					
Tied (S1)	2.755773	2.762714	2.757664	–	–
Dynamic centering	2.771955	2.799285	2.799276	$+0.031455 \pm 0.013465$	0/3
Input-Bias (S3)	2.741402	2.758531	2.751138	-0.008360 ± 0.005336	3/3

C Larger-Scale Pretraining and Supervised Fine-Tuning

Pretraining protocol S1, S3, and S6 are trained from random initialization with seeds 42, 123, and 2026; no released pretrained Qwen3 weights are used. The Qwen3-0.6B-aligned configuration is specified in Section 5.1. The FineWeb-Edu corpus contains approximately 20B packed tokens at sequence length 340. Training runs for one epoch with batch size 32 per GPU on eight GPUs (global batch 256). Low-rank branches use rank 32 and scale 1; output bias is enabled. Training uses bfloat16 and the AdamW schedule above. RoPE uses base 10^6 and RMSNorm uses $\epsilon = 10^{-6}$. A fixed-random 1% validation split (split seed 20260602) contains 588,235 sequences and 199,411,665 effective prediction tokens, shared by all runs.

SFT protocol Each pretrained checkpoint is fine-tuned on the 460,341 training conversations of HuggingFaceTB/smol-smoltalk for one epoch using AdamW, learning rate 2×10^{-5}, bfloat16, maximum sequence length 1024, and gradient clipping at 1.0. Eight GPUs with per-device batch size 12 and no gradient accumulation give global batch 96. The Qwen3 ChatML template is used; only assistant content and its immediately following `<|im_end|>` token receive supervision. User/system content, template tokens, and padding are masked. Final weights are reloaded for evaluation on all 24,229 test conversations.

Aggregation For each variant, we report the mean loss over three seeds and its exponential as aggregate PPL. Paired differences are computed within seed before averaging; their error bars use sample SD ($n - 1$ denominator). Thus paired mean PPL differences need not equal differences between aggregate PPL values.

Table 26: Per-seed cross-entropy for 0.6B/FineWeb-Edu pretraining and subsequent SmolTalk SFT.

Stage	Variant	seed42	seed123	seed2026
Pretraining	S1	2.755773	2.762714	2.757664
	S3	2.741402	2.758531	2.751138
	S6	2.745604	2.751806	2.744375
SFT	S1	1.348365	1.361640	1.349187
	S3	1.331062	1.356669	1.345507
	S6	1.338309	1.343713	1.333366