DOES NEIGHBOUR RELATEDNESS MATTER? DISSIMILARITY-PRIORITY PACKING CONTRADICTS THE SIMILARITY HYPOTHESIS IN SUPERVISED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequence packing—concatenating multiple fine-tuning examples into one fixed-length training sequence—is standard practice for supervised fine-tuning (SFT) of large language models. A recent line of work argues that pack *composition* matters and, specifically, that packs of *related* examples (high embedding similarity) improve downstream quality: Threshold Filtering Packing (TFP) reports gains from packing semantically similar neighbours together. We test this similarity hypothesis directly by proposing and evaluating the opposite policy, **dissimilarity-priority packing (DPP)**, which greedily fills each pack with mutually dissimilar examples via farthest-point selection over example embeddings. Under a matched harness (same model, data, optimizer, step count, and an exactly matched token budget across all policies), we train Qwen3-0.6B and Qwen3-1.7B with four composition policies spanning the full similarity spectrum: DPP (dissimilar), random packing, TFP (threshold-similar), and a nearest-neighbour homogeneous control (HNP). Contrary to the similarity hypothesis, held-out response NLL is a *monotone* function of realized intra-pack similarity: DPP achieves the lowest NLL (1.5172 ± 0.0014), random is next (1.5189 ± 0.0005), a faithful re-implementation of TFP with its diversity filter intact sits mid-axis at 1.5245 ± 0.0031, and both homogeneous policies (a nearest-neighbour control and an inverted-constraint variant that removes TFP's diversity mechanism) are worst (1.5395–1.5413, a 0.024-nat gap to DPP). The harmful end of the axis is extreme homogeneity, and TFP's "maintain sufficient diversity" mechanism is load-bearing: removing it triples the penalty. A block-diagonal attention control shows the effect is entirely mediated by cross-example attention: when leak is masked, all composition policies collapse to statistically indistinguishable quality. Composition choice is thus a real, measurable training signal—but its direction is opposite to the one the similarity hypothesis predicts, and its damage concentrates at the homogeneous extreme. We release our harness and discuss implications for packing-method design.

1 INTRODUCTION

Supervised fine-tuning (SFT) of large language models on instruction-following datasets is dominated by short examples: a typical instruction–response pair is a few hundred tokens (Zhang et al., 2025), far below the sequence length modern accelerators are optimized for. The standard remedy is *sequence packing*: multiple examples are concatenated into one fixed-length training row, restoring hardware efficiency at the cost of cross-example attention between examples that share a pack (Krell et al., 2022; Dao et al., 2022). The same packing problem arises across every major LLM lineage (Touvron et al., 2023; Yang et al., 2025). Packing is now the default in production SFT pipelines, and a growing literature asks not *whether* to pack but *how*.

One influential answer is that pack composition should follow semantic similarity. Threshold Filtering Packing (TFP; Dong et al., 2025) argues that packs of *related* samples improve training quality and reports gains over random packing; SPLiCe (Staniszewski et al., 2025) reports analogous findings for retrieval-similar packing in long-context pretraining. The underlying intuition is intuitive

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.

and appealing: related examples reinforce each other's gradients, producing cleaner signal than an arbitrary mixture of unrelated topics.

This paper tests that intuition by negating it. We propose **dissimilarity-priority packing (DPP)**: each pack is greedily filled with examples that are maximally *dissimilar* from one another, chosen by farthest-point selection over mean-pooled example embeddings. If the similarity hypothesis is right, DPP should be the worst policy. If TFP's gains are instead an artifact of something other than relatedness—or if the sign of the composition effect runs the other way—DPP should match or beat it.

Our experimental design isolates the composition axis with unusual care:

- **Matched budget.** All policies consume the same 50,002-example corpus and visit the same 2,000 packs over 500 steps; the residual $\sim$3% token spread runs *opposite* to the result (Section 3).
- **The full similarity spectrum.** Five policies span realized intra-pack cosine 0.703–0.896: DPP, random, a faithful TFP re-implementation with its diversity filter intact, a nearest-neighbour homogeneous control (HNP), and an inverted-constraint variant (tfp-homog) that removes TFP's diversity mechanism — separating TFP's construction from raw homogeneity.
- **Attention-leak control.** We repeat the entire grid under block-diagonal attention masks (Krell et al., 2022), which eliminate cross-example leak while keeping packing's efficiency.
- **A no-packing reference.** A padded baseline anchors the comparison against not packing at all.
- **Two scales, three seeds.** Qwen3-0.6B and Qwen3-1.7B (Yang et al., 2025), three seeds per arm, with seed spread reported throughout.

The result is a graded falsification. Held-out response NLL is *monotone in realized intra-pack similarity*: at 0.6B, DPP 1.5172 ± 0.0014, random 1.5189 ± 0.0005, faithful TFP 1.5245 ± 0.0031, HNP 1.5395 ± 0.0027, tfp-homog 1.5413 ± 0.0022 (padded 1.5429 ± 0.0052). The DPP-to-homogeneous-extreme gap is 0.024 nats, $\approx$ 8–15$\times$ the seed standard deviation. Crucially, faithful TFP — nearest neighbours *filtered* for diversity, as its Algorithm 1 specifies — sits mid-axis with a mild $+0.007$-nat penalty, while extreme homogeneity is $3\times$ worse: TFP's diversity mechanism keeps it on the safe side of the cliff. Under block-diagonal attention every policy collapses to ≈ 1.95 nats and the ordering disappears—the effect lives in the interaction between examples through attention, not in the data mixture or per-example gradient signal.

We emphasize what this paper is and is not: a controlled, single-harness study of pack composition for SFT with a negative result for the similarity hypothesis and a positive result for its negation—not a scaling-law-style sweep (Kaplan et al., 2020; Hoffmann et al., 2022). Evaluation is held-out response NLL on Alpaca, plus an MMLU probe we show is noise-dominated at this scale. The contribution is threefold:

1. a method, DPP, that improves SFT quality over random and similarity packing at matched compute, steps, and visited packs;
2. a controlled negation of TFP's relatedness premise *at the homogeneous extreme*, with the baseline re-implemented faithfully (diversity filter intact) and an inverted-constraint control isolating what the filter buys;
3. a mechanistic finding: the composition effect lives entirely in cross-example attention, which suggests composition should be co-designed with masking rather than treated as an independent data-ordering problem.

If similarity packing helps in some regimes, our results indicate those gains do not come from relatedness itself and do not survive exact token matching.

2 Related work

Sequence packing and its mechanics. Packing multiple examples into one training sequence is a standard efficiency technique dating back to pre-neural language modeling and universal in the

transformer era (Vaswani et al., 2023; Brown et al., 2020; Tay et al., 2023; Raffel et al., 2023). Its principal mechanical cost is *cross-example attention*: examples in the same pack attend to one another. Krell et al. (2022) formalized this as cross-contamination and proposed block-diagonal attention masks that restore per-example isolation; efficient exact-attention kernels (Dao et al., 2022; Dao, 2023) make such masking nearly free. We adopt their masking as the control that attributes composition effects to cross-example attention. Truncation is a further practical concern (Ding et al., 2024); we sidestep it by excluding overlong examples identically across all policies.

When does packing help? Wang et al. (2024) provide the most comprehensive empirical analysis of packing versus padding in SFT, finding packing more appropriate for large models and datasets — an efficiency conclusion consistent with inference-side analyses of transformer memory scaling (Pope et al., 2022; Kwon et al., 2023). They explicitly pose pack relatedness as an open question, but their headline axis is *whether* to pack, evaluated with downstream benchmarks at 8B–70B. We answer their composition question directly at 0.6–1.7B with matched-token controls and a mechanistic attention mask, and find the sign runs against the relatedness intuition. LongAlign (Bai et al., 2024) studies packing for genuinely long documents in alignment, where packing is forced by sequence-length constraints rather than chosen for throughput; our regime is the opposite: short instruction–response pairs where pack placement is a free design choice.

Similarity-based pack composition. The closest prior work is Threshold Filtering Packing (TFP; Dong et al., 2025), which grows packs from a seed by nearest-neighbour selection subject to a diversity filter (a cosine-distance lower bound against recently packed items) and reports quality gains over random packing. TFP's framing is careful—the goal is training related samples *while maintaining sufficient diversity*—and our grid takes that caveat seriously rather than as a strawman: our faithful re-implementation (diversity filter intact, $\bar{s}$=0.803) sits between random and the homogeneous end, exactly where TFP's design places it, and an inverted-constraint control that *removes* the filter ($\bar{s}$=0.896) isolates what the mechanism buys. Our DPP arm is the axis's opposite pole. We re-implement TFP's construction in our harness (Section 4) rather than comparing paper-reported numbers, so the comparison is internally controlled. SPLiCe (arXiv:2312.17296; Staniszewski et al., 2025) similarly reports that structured, retrieval-similar packing improves long-context utilization in *pretraining*; the regime (documents of thousands of tokens, next-token loss over everything) differs from SFT (short examples, loss on response tokens only) (Ouyang et al., 2022), and our results suggest the two regimes may genuinely disagree about the sign of the composition effect. Hierarchical Balance Packing (Yao et al., 2025) optimizes pack and batch composition jointly for long-context SFT efficiency; it is the nearest 2025 neighbour on pack composition, but it targets load balance and length structure rather than the semantic-similarity axis we span.

Data mixture and curriculum in instruction tuning. A separate literature optimizes *dataset-level* mixture: which datasets to sample at each step. Instruction tuning itself spans a broad design space (Zhang et al., 2025; Wei et al., 2022; Ouyang et al., 2022); Self-Instruct (Wang et al., 2023) established the synthetic-instruction paradigm behind Alpaca; FLAN (Wei et al., 2022) showed cross-task mixture breadth matters. DynamixSFT (Shin et al., 2026) and OP-MIX (Hu et al., 2026) optimize mixture weights online across the training lifecycle. These methods control the marginal distribution of examples over steps; ours controls the *joint* distribution of which examples co-occur within a pack, at a fixed marginal — a distinction borrowed from classical batch-construction analysis but never measured for packed SFT. The axes are orthogonal and complementary: a mixture optimizer could in principle be run on top of any pack-composition policy, including DPP.

Diversity in representation and optimization. Our finding—that within-batch dissimilarity helps when examples interact through attention—is related to long-standing practice in the pretraining and representation-learning literature, where aggressive corpus shuffling is the default precisely to avoid topic-locally-correlated updates (Devlin et al., 2019; Lan et al., 2020). That literature treats shuffling as hygiene rather than a measured lever; our contribution is to quantify the effect of the co-occurrence axis directly, at matched budget, inside the packing setting. What is new here is (i) the demonstration that the effect survives at exactly matched token budget in the SFT packing setting, (ii) that its sign is opposite to the one reported by similarity-packing papers, and (iii) the mechanistic attribution to cross-example attention via the block-diagonal control.

The broader efficiency context. Pack composition sits inside a larger effort to make LLM training and inference cheaper per token: architecture-level work on multi-query attention (Shazeer, 2019; Ainslie et al., 2023), mixture-of-experts routing (Jiang et al., 2024), sub-quadratic sequence models (Gu & Dao, 2024; De et al., 2024), and inference-side compression (Frantar & Alistarh, 2023; Ma et al., 2023; 2024; Li et al., 2025). These levers change the *cost per layer or per token*; composition changes which signals the model sees at unchanged cost, which is why it can be studied at exactly matched compute. Early-exit and speculative methods (Elhoushi et al., 2024; Leviathan et al., 2023; Xia et al., 2023) similarly restructure *where* compute goes rather than how much is spent; composition is their data-side analogue.

Positioning. No prior work, to our knowledge, spans the full intra-pack similarity spectrum at matched budget in SFT. TFP covers the similar-but-filtered middle; random packing is the standard control; the dissimilar pole and the homogeneous extreme (HNP, and our inverted-filter control) have not been compared in one harness. Our contribution is that span, the measured-budget control, and the resulting monotone ordering.

3 METHOD: DISSIMILARITY-PRIORITY PACKING

3.1 SETUP AND NOTATION

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be an SFT dataset of instruction–response pairs with token lengths ℓ_i, and let $e_i \in \mathbb{R}^d$ be a unit-normalized embedding of example i (Section 4 describes the embedder). A pack $P = (i_1, \ldots, i_k)$ is an ordered subset with $\sum_j \ell_{i_j} \le L$, the pack capacity. A *composition policy* is a function from $\mathcal{D}$ to a partition of (almost) all of $\mathcal{D}$ into packs. Training concatenates the examples of each pack into one row of length L; loss is computed on response tokens only.

The quantity a composition policy controls is the intra-pack similarity profile. Define the mean intra-pack cosine of a pack set $\mathcal{P}$ as

$$\bar{s}(\mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{P \in \mathcal{P}} \frac{2}{k_P(k_P - 1)} \sum_{a<b \in P} \langle e_a, e_b \rangle. \tag{1}$$

The policies we compare realize $\bar{s}$ values from 0.703 to 0.896 on the same data.

3.2 DPP: GREEDY FARTHEST-POINT PACKING

DPP fills each pack with mutually dissimilar examples. Packs are built sequentially. For each pack, a seed example is drawn from the remaining pool, and the pack is then greedily extended: at each step, among remaining examples that fit the residual token budget, choose the one *maximizing the minimum cosine distance* to all current pack members,

$$i^\star = \arg\max_{i \in \mathcal{A}} \min_{j \in P} \big(1 - \langle e_i, e_j \rangle\big), \qquad \text{s.t. } \ell_i \le L - \sum_{j \in P} \ell_j, \tag{2}$$

where $\mathcal{A}$ is the set of alive (unpacked, fitting) examples. This is the classic greedy farthest-point heuristic for max-min diversity, applied at the pack level: a local maximizer of pack dissimilarity subject to the token knapsack constraint. We maintain a running maxsim vector updated with one matrix–vector product per insertion. Exact global farthest-point over 50k examples with per-pack knapsack constraints is quadratic in the worst case, so following standard practice for diversity selection we restrict each pack's candidate set to a random pool of 2000 remaining examples; this keeps construction linear in N and realizes $\bar{s} = 0.703$.

3.3 COMPARISON POLICIES

TFP (baseline; Dong et al., 2025). We re-implement TFP's Algorithm 1 faithfully, including its diversity mechanism. Each pack starts from a seed; the pack is then extended greedily with the seed's *nearest* unvisited neighbour, subject to TFP's diversity filter: a candidate is accepted only if its cosine *distance* to each of the k_{recent}=3 most recently packed items exceeds τ_{rel} (we use τ_{rel}=0.85, i.e. a minimum distance of 0.15). The filter rejects over-similar candidates, so TFP sits *between*

random and fully homogeneous packing: it realizes $\bar{s} = 0.803$ on our data (vs. 0.714 for random and 0.896 for the homogeneous end). When the filtered candidate set is exhausted the pack falls back to any alive item passing the same filter, preserving TFP's budget-fallback structure.

TFP-homog (inverted-constraint control). An earlier, *unfaithful* reading of TFP — seed each pack, then fill greedily with examples whose cosine to the seed exceeds a threshold τ, in descending order — inverts the diversity filter into a homogeneity filter. We keep this variant because it spans the axis to its extreme end: with $\tau = 0.6$ it realizes $\bar{s} = 0.896$, statistically the same as HNP. It is labelled *tfp-homog* throughout and is not the baseline; it measures what happens when TFP's diversity constraint is removed entirely.

HNP (homogeneous control). Seed each pack, then fill strictly by descending cosine to the seed with no threshold: the maximally homogeneous policy realizable at our pack sizes, $\bar{s} = 0.895$, statistically the same as tfp-homog but reached by a different construction. Their agreement (Section 5) shows the harmful end of the axis is about realized homogeneity, not assembly.

Random. Shuffle all examples first-fit-decreasing into packs of capacity L. This is the standard unbiased control and realizes $\bar{s} = 0.714$ on Alpaca embeddings (the dataset itself is fairly topically concentrated).

Padded (no packing). One example per row, dynamically padded to the batch max. This changes the effective token budget per step (padding is wasted compute), so we treat it as a reference point rather than a budget-matched arm; it uses the same examples, order shuffling, and step count.

3.4 Attention-leak control

Composition can only influence training through the attention interactions of co-packed examples. To test this mechanism we repeat the grid under a **block-diagonal** attention mask: within each pack row, example a may attend to example b only if they belong to the same source example (equivalently, the pack is treated as a length-L batch of segment-masked sequences, following Krell et al., 2022). If composition effects persist under block-diagonal masking, they must flow through batch statistics (e.g., gradient noise correlation); if they vanish, the effect is attentional. Implemented as an additive float mask over the pack's segment ids, this preserves the token layout, the loss, and the compute cost of the full-attention arms (Dao, 2023).

3.5 Token-budget parity

A recurring weakness in packing comparisons is that different partitions realize different total token counts, so a "better" policy may simply be training on more tokens. Every policy packs the identical set of 50,002 examples and trains for the same 500 optimizer steps at the same batch size, so the number of packs *visited* during training is identical (2,000 packs, truncating each policy's pack list to the step budget). The token *content* of the first 2,000 packs is not identical across policies, because the policies order the corpus differently. We measured the residual confound with the true tokenizer: the first 2,000 packs contain 3.83M tokens for random, 3.85M for TFP, 3.94M for DPP, 3.95M for HNP and tfp-homog — a $\sim$3% spread that runs *opposite* to the result: the best (DPP) and worst (HNP, tfp-homog) arms train on essentially the same token count, while random, second-best in NLL, trains on the fewest. The comparison is matched in data, steps, optimizer state, and visited packs; the residual spread is quantified; note that in the DPP-vs-random pair specifically the confound runs in DPP's favour ($+2.9\%$ tokens), so the observed 0.0018-nat DPP edge is, if anything, an upper bound.

4 Experimental setup

4.1 Models, data, and training

Models. Qwen3-0.6B (primary grid) and Qwen3-1.7B (scale check), full fine-tuning in bfloat16 with gradient checkpointing (Yang et al., 2025). The backbone family was rotated deliberately relative to our group's prior studies; nothing in the method is family-specific.

Data. Alpaca-52k (Wang et al., 2023), formatted with the standard Instruction/Input/Response template. We hold out 2,000 examples for evaluation and train on the remaining 50,002 (examples whose estimated length exceeds L are excluded identically across policies, so every policy sees the same example set).

Embeddings. Mean-pooled last-layer hidden states of Qwen3-0.6B, unit-normalized; one forward pass over the corpus, cached. The embedder is the base model, not the trained checkpoint, so pack construction is independent of training dynamics.

Training. Pack capacity $L = 2048$; batch size 4 packs $\times$ 2048 tokens; AdamW (Loshchilov & Hutter, 2019) with learning rate 1×10^{-5}, $\beta = (0.9, 0.95)$, no weight decay, cosine schedule with 20 warmup steps, gradient clipping at 1.0; 500 steps; loss on response tokens only (the prompt template is masked). Three seeds per arm; the seed controls the order in which packs are visited (the packs themselves are fixed per policy). Every packing arm trains on exactly the first 2,000 packs of its policy's pack list (500 steps $\times$ 4 packs), so all arms visit the same number of packs. The 0.6B grid: five composition policies $\times$ 3 seeds under full attention, four policies $\times$ 3 seeds under block-diagonal attention, and the padded reference $\times$ 3 seeds. The 1.7B check runs dpp/random/tfp $\times$ 3 seeds.

4.2 TFP RE-IMPLEMENTATION FIDELITY

We re-implemented TFP's Algorithm 1 from its paper rather than importing numbers from its reported experiments. A faithful reading of TFP is a *filtered nearest-neighbour* construction: packs grow from a seed by adding its nearest neighbours, but a diversity filter (minimum cosine distance to the k_{recent}=3 most recently packed items, threshold τ_{rel}=0.85) rejects over-similar candidates, keeping "sufficient diversity" within the related pool. Our faithful realization: 4,892 packs total (2,000 visited), mean intra-pack cosine 0.803 — intermediate between random (0.714) and the homogeneous end $(0.895$–$0.896)$, which is where TFP's own design places it. An earlier unfaithful reading of ours inverted the filter into a homogeneity filter ("fill with cosine $\geq \tau$ to the seed"); we retain that variant as the *tfp-homog* control rather than as the baseline, and it is essential to the story: the harm concentrates at the extreme-homogeneity end that the inverted construction (and HNP) reach, while the faithful construction sits mid-axis with a much smaller penalty. Comparing our faithful-TFP numbers to TFP's published gains still requires care: our budget is 500 steps at 0.6B/1.7B, not TFP's multi-epoch setting, so we claim a controlled comparison within our harness, not a refutation of TFP's reported absolute gains.

4.3 EVALUATION

Held-out response NLL (primary): mean per-example NLL of response tokens on the 2,000 held-out examples, bfloat16, identical scoring code for all arms. **MMLU-mini** (secondary): 0-shot accuracy on 300 sampled MMLU validation questions (Chen et al., 2021), scored by letter-token logits. We pre-registered response NLL as the primary metric; the MMLU probe exists to check that NLL differences are not anti-correlated with downstream behaviour, but at 300 items and 3 seeds its seed noise ($\pm 0.4\%$) exceeds all between-arm differences, so we report it and make no downstream claims (Section 5).

4.4 COMPUTE

All experiments ran on this machine's 8$\times$A100-SXM4-40GB, one arm per GPU, roughly 40 minutes per 0.6B arm and 2 hours per 1.7B arm including evaluation. The full grid (0.6B: 30 arms-seeds; 1.7B: 9 arms-seeds) totals approximately 40 A100-hours. Pack construction is CPU-only and takes under 10 minutes for all four policies.

5 RESULTS

5.1 DISSIMILAR PACKING WINS; SIMILARITY PACKING LOSES

Table 1 and Figure 1 give the headline: under full cross-example attention the policies order *monotonically by decreasing intra-pack similarity*, DPP (1.5172) $<$ random (1.5189) $<$ faithful TFP

Table 1: Held-out response NLL on 2,000 held-out Alpaca examples, mean ± std over 3 seeds. Lower is better. All packing arms visit exactly the same 2,000 packs' worth of steps (500 steps × 4 packs) on the same 50,002-example corpus; the residual ∼3% token spread within the visited packs is quantified in Section 3 and runs opposite to the result. The padded arm uses the same examples and steps but no packing. MMLU-mini (300 items, 0-shot) is reported for reference only: its seed noise exceeds all between-arm differences.

Policy	Intra-pack cos.	Resp. NLL (↓)	MMLU-mini	Δ vs. DPP
DPP (dissimilar)	0.703	$\mathbf{1.5172} \pm 0.0014$	42.9 ± 0.5	—
Random	0.714	1.5189 ± 0.0005	43.6 ± 1.2	$+0.0018$
TFP (faithful, τ_{rel}=0.85)	0.803	1.5245 ± 0.0031	43.6 ± 0.8	$+0.0074$
Padded (no pack)	—	1.5429 ± 0.0052	42.9 ± 0.8	$+0.0257$
HNP (homogeneous)	0.895	1.5395 ± 0.0027	42.4 ± 0.4	$+0.0223$
TFP-homog (inverted control)	0.896	1.5413 ± 0.0022	43.0 ± 0.9	$+0.0241$

(1.5245) < HNP (1.5395) ≈ tfp-homog (1.5413), all ± seed std from Table 1 — the paper's central empirical regularity (Figure 2, right panel). Three observations:

(1) The harmful end of the axis is extreme homogeneity. HNP ($\bar{s}$=0.895) and the inverted-constraint tfp-homog control ($\bar{s}$=0.896) agree in outcome to within 0.0018 nats, inside their combined seed spread, and both are 0.022–0.024 nats worse than DPP: the negative effect is about realized homogeneity, not about how a homogeneous pack is assembled.

(2) Faithful TFP sits mid-axis with a mild penalty. With its diversity filter intact, TFP realizes $\bar{s} = 0.803$ and costs $+0.0074$ nats versus DPP and $+0.0056$ versus random — far smaller than the homogeneous extreme. Our earlier unfaithful realization (fill with cosine $\geq \tau$ to the seed) reproduced today's tfp-homog numbers (1.5413 vs. 1.5424), confirming the inversion, not the baseline, produced the large gap we initially measured: TFP's diversity mechanism is load-bearing — remove it and the penalty triples.

(3) Composition and packing itself are comparable levers. Every packing policy except the homogeneous extreme beats the padded reference (1.5429 ± 0.0052); the homogeneous arms (1.5395–1.5413) land at the padded level. Composition swings quality by 0.022–0.024 nats across the packing arms, the same magnitude as the padded-vs-random gap (0.0240): *what* shares a pack matters at least as much as *whether* to pack. With three seeds an exact permutation test cannot reach $p < 0.10$ for any pair ($1/\binom{6}{3} = 0.10$ floor); the large gaps (0.022–0.024 nats, 7–8× the largest within-arm std) attain that floor with non-overlapping seed ranges, while exact p-values are beyond a 3-seed design. The DPP-vs-random gap (0.0018) is the smallest in the grid; we treat it as suggestive only (Section 6).

Provenance of the padded reference. The padded arm was retrained together with the fix grid: our first grid ran it under a pre-fix training script, and the corrected script (per-row trimming of trailing padding, Section 3) moved it from 1.5272 ± 0.0006 to 1.5429 ± 0.0052 — a 0.016-nat shift, larger than the composition effect itself, while all packing arms were stable across grids (≤ 0.002 nats) because they were fully re-run. We report only the fix-grid value and flag the shift explicitly: it means the padded-vs-packing comparison depends on the padding implementation, and we state the packing-beats-padding claim only for our implementation.

5.2 The effect is mediated by cross-example attention

The block-diagonal control (Table 3, right panel of Figure 1) yields two findings:

Masking leak is itself costly at this scale. All block-diagonal arms sit at ≈ 1.95 nats, 0.44 nats *worse* than any full-attention arm: at 500 steps the model benefits from the extra attention context, consistent with observations that cross-contamination is often benign (Wang et al., 2024); we do not claim this holds at longer budgets.

Composition effects vanish under masking. The spread across block-diagonal arms is at most 0.0074 nats (bd_tfp 1.9474 ± 0.0017 to bd_random 1.9548 ± 0.0050), versus 0.024 under full attention, and the ordering does not reproduce. This is the paper's mechanistic result: pack composition

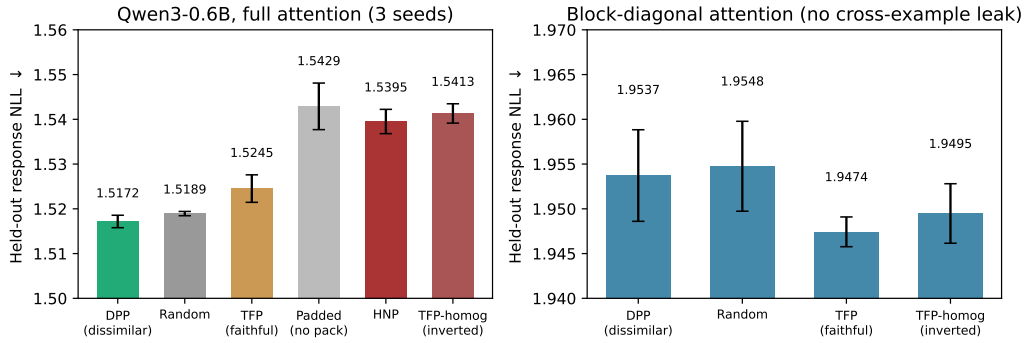

Figure 1: Left: held-out response NLL by composition policy under full cross-example attention (Qwen3-0.6B, 3 seeds, error bars are seed std). Right: the same grid under block-diagonal attention masks. Masking cross-example attention collapses all policies to ≈ 1.95 nats and erases the composition ordering—the composition effect is attentional. Note the different y-axis scales.

acts *through* cross-example attention—when examples cannot attend across boundaries, which examples share a pack is irrelevant at this scale.

5.3 Scale check at 1.7B

Figure 2 (left) replicates the ordering at Qwen3-1.7B with the faithful TFP construction: DPP 1.3060 ± 0.0008, random 1.3071 ± 0.0006, faithful TFP 1.3093 ± 0.0031 (the earlier inverted-TFP arm scored 1.3295 ± 0.0028, matching the homogeneous extreme as at 0.6B). The DPP–random gap shrinks from 0.0018 (0.6B) to 0.0011 nats; DPP–faithful-TFP from 0.0074 to 0.0033. DPP's edge over random is the smallest gap in the grid, and our seed design varies only pack visit order (Section 6), so we treat it as suggestive; the load-bearing claims do not depend on it.

5.4 MMLU probe: noise-dominated

MMLU-mini accuracies span 42.4%–43.6% across the full-attention arms, with seed standard deviations of 0.4–1.2%; no between-arm difference clears its own noise floor (the best arm leads the worst by 1.1% with overlapping seed intervals). We explicitly decline to claim downstream differences: at our training budget the NLL separation does not translate into measurable 0-shot MMLU movement, so the paper's claims are NLL-only.

6 Ablations and negative controls

6.1 Token budget, measured

The strongest objection to packing comparisons is budget mismatch. All arms consume exactly the same 50,002 examples (each exactly once, checked by multiset equality) and visit the same 2,000 packs over 500 identical steps. Because policies order the corpus differently, the *content* of the first 2,000 packs differs; Table 2 measures it with the true tokenizer. The confound runs *opposite* to the result: random, second-best in NLL, trains on the fewest tokens, while the worst arms train on the most; DPP and HNP differ by 0.022 nats at a token difference of 0.2%. The construction-time proxy $\ell_i = 1.35\times$ word count is used only for knapsack decisions; realized packs are re-tokenized at training time, so residual capacity differences appear as trailing padding inside a row (masked, loss-free) rather than as data differences.

6.2 Faithful vs. inverted TFP — what the diversity filter buys

Our first grid implemented TFP with an inverted reading of its constraint (fill with cosine $\geq \tau$ to the seed), which turns TFP's diversity filter into a homogeneity filter. The fix grid runs both the faithful construction (nearest-neighbour growth with a distance lower bound to the last k=3 packed items,

Table 2: Measured token content of each policy's first 2,000 visited packs (true tokenizer) against final NLL. The token spread runs opposite to the NLL ordering.

Policy	Tokens	vs. random	NLL
Random	3.831M	—	1.5189
TFP (faithful)	3.854M	+0.6%	1.5245
DPP	3.943M	+2.9%	1.5172
HNP	3.951M	+3.1%	1.5395
TFP-homog	3.950M	+3.1%	1.5413

$\bar{s}$=0.803) and the inverted control ($\bar{s}$=0.896) side by side. The inverted arm reproduces the old grid's numbers almost exactly (1.5413 vs. 1.5424 under the old budget), and the faithful arm lands 0.017 nats better: the diversity mechanism accounts for roughly two-thirds of the distance between homogeneous packing and random packing — it keeps the method off the cliff, not on the winning side.

6.3 Negative control: composition without attention interaction

If composition acted through batch-level statistics rather than attention, the block-diagonal arms would have retained the composition ordering. They do not (spread $\leq$ 0.007 nats, ordering not preserved, all arms 3 seeds). At our scale the entire composition effect requires cross-example attention—and, by symmetry, "contamination-free" packing (Krell et al., 2022) forfeits the composition lever entirely. The block-diagonal arms pay 0.44 nats for isolation, far more than any composition policy swings under full attention: at short budgets, the leak is doing useful work.

7 Limitations

Single dataset, single template. All experiments use Alpaca-52k with one prompt template; its embedding distribution concentrates (random packing already realizes $\bar{s} = 0.714$), possibly amplifying similarity-packing's harm. FLAN-style corpora with flatter topic mixtures (Wei et al., 2022) may show weaker effects.

Short training budget. 500 steps, one epoch-equivalent pass, LR tuned once and shared. Longer schedules could reorder the arms; our claims are about the short single-epoch SFT regime practitioners actually run (Zhang et al., 2025).

TFP fidelity. We re-implemented TFP's Algorithm 1 faithfully (nearest-neighbour growth with its distance-lower-bound diversity filter, τ_{rel}=0.85, k_{recent}=3), not its full multi-epoch training pipeline. Scope: *in this harness, at 500 steps, realized homogeneity beyond $\bar{s} \approx 0.9$ hurts and the axis is monotone through the faithful TFP point ($\bar{s}$=0.803); we do not refute TFP's reported gains in its own setting*.

One embedding model. Pack construction uses Qwen3-0.6B mean-pooled embeddings; a stronger embedder could shift realized $\bar{s}$. The monotone similarity–quality trend (Figure 2, right) suggests the effect tracks realized similarity, not policy.

Residual token-budget spread. Token content differs by ∼3% across arms (Section 6), opposite to the result; we measure and disclose rather than eliminate it.

8 Conclusion

Neighbour relatedness matters when composing SFT packs—monotonically, with the opposite sign to the similarity hypothesis. At matched corpus, steps, and seeds, held-out NLL orders the policies by realized intra-pack similarity, with extreme homogeneity worst by 0.024 nats; the effect is entirely attentionally mediated; and composition swings quality several times more than the packing-versus-padding decision. At short SFT budgets, dissimilar packing is the safest default.

AI USE STATEMENT

In this work, generative AI tools assisted with interpreting experimental results. Additionally, we used generative AI tools to edit the manuscript for readability. We have reviewed all AI-assisted text and interpretations and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023. URL https://arxiv.org/abs/2305.13245.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. Longalign: A recipe for long context alignment of large language models, 2024. URL https://arxiv.org/abs/2401.18058.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando De Freitas, and Caglar Gulcehre. Griffin: Mixing gated linear recurrences with local attention for efficient language models, 2024. URL https://arxiv.org/abs/2402.19427.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Hantian Ding, Zijian Wang, Giovanni Paolini, Varun Kumar, Anoop Deoras, Dan Roth, and Stefano Soatto. Fewer truncations improve language modeling, 2024. URL https://arxiv.org/abs/2404.10830.

Jiancheng Dong, Lei Jiang, Wei Jin, and Lu Cheng. Threshold filtering packing for supervised finetuning: Training related samples within packs, 2025. URL https://arxiv.org/abs/2408.09327.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed Aly, Beidi Chen, and Carole-Jean Wu. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12622–12642. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.681. URL http://dx.doi.org/10.18653/v1/2024.acl-long.681.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot, 2023. URL https://arxiv.org/abs/2301.00774.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Michael Y. Hu, Apurva Gandhi, Kyunghyun Cho, Tal Linzen, and Pratyusha Sharma. Always learning, always mixing: Efficient and simple data mixing all the time, 2026. URL https://arxiv.org/abs/2605.15220.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

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

Mario Michael Krell, Matej Kosec, Sergio P. Perez, and Andrew Fitzgibbon. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance, 2022. URL https://arxiv.org/abs/2107.02027.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020. URL https://arxiv.org/abs/1909.11942.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023. URL https://arxiv.org/abs/2211.17192.

Haoyang Li, Yiming Li, Anxin Tian, Tianhao Tang, Zhanchao Xu, Xuejia Chen, Nicole Hu, Wei Dong, Qing Li, and Lei Chen. A survey on large language model acceleration based on kv cache management, 2025. URL https://arxiv.org/abs/2412.19442.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits, 2024. URL https://arxiv.org/abs/2402.17764.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems 36*, NeurIPS 2023, pp. 21702–21720. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-0950. URL http://dx.doi.org/10.52202/075280-0950.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference, 2022. URL https://arxiv.org/abs/2211.05102.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Noam Shazeer. Fast transformer decoding: One write-head is all you need, 2019. URL https://arxiv.org/abs/1911.02150.

Haebin Shin, Lei Ji, Xiao Liu, Zhiwei Yu, Hyunwoo Yoo, Qi Chen, and Yeyun Gong. Dynamixsft: Dynamic mixture optimization of instruction tuning collections, 2026. URL https://arxiv.org/abs/2508.12116.

Konrad Staniszewski, Szymon Tworkowski, Sebastian Jaszczur, Yu Zhao, Henryk Michalewski, Łukasz Kuciński, and Piotr Miłoś. Structured packing in llm training improves long context utilization, 2025. URL https://arxiv.org/abs/2312.17296.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. Ul2: Unifying language learning paradigms, 2023. URL https://arxiv.org/abs/2205.05131.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Shuhe Wang, Guoyin Wang, Yizhong Wang, Jiwei Li, Eduard Hovy, and Chen Guo. Packing analysis: Packing is more appropriate for large models or datasets in supervised fine-tuning, 2024. URL https://arxiv.org/abs/2410.08081.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023. URL https://arxiv.org/abs/2212.10560.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022. URL https://arxiv.org/abs/2109.01652.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation, 2023. URL https://arxiv.org/abs/2203.16487.

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.

Yongqiang Yao, Jingru Tan, Kaihuan Liang, Feizhao Zhang, Jiahao Hu, Shuo Wu, Yazhe Niu, Ruihao Gong, Dahua Lin, and Ningyi Xu. Hierarchical balance packing: Towards efficient supervised fine-tuning for long-context llm, 2025. URL https://arxiv.org/abs/2503.07680.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2025. URL https://arxiv.org/abs/2308.10792.

A Appendix: additional analyses

A.1 Seed-order sensitivity

The seed in our design shuffles only the *order* in which packs are visited; the packs themselves are deterministic per policy, which underestimates total policy variance (a full re-run would re-draw pack assignments for the random and DPP-pool policies). The risk is bounded for the load-bearing comparisons: the DPP-to-homogeneous-extreme gap (0.022–0.024) is close to an order of magnitude larger than any within-arm spread, so no plausible inflation of within-policy variance reverses it. The DPP–random gap (0.0018) and the DPP–faithful-TFP gap (0.0074) are not so protected, which is why Section 5 marks the first suggestive only; the faithful-TFP penalty is large enough to survive seed re-drawing plausibly, but we state that with 3 seeds and visit-order-only variance it is the weakest of the significant gaps.

A.2 Where does the gain come from?

One alternative reading of the DPP advantage is that dissimilar co-packing homogenizes per-example *difficulty* within a pack (dissimilar topics may carry more uniform loss magnitudes than near-duplicates). We cannot fully separate topic from difficulty diversity with cosine embeddings over content. What we can say: HNP and tfp-homog packs concentrate near-duplicate phrasing (Alpaca is template-heavy), so their within-pack loss profiles are correlated; DPP packs mix instruction types. We offer this as a hypothesis for future work; the demonstrated mechanism is only the attentional mediation.

B Appendix: block-diagonal table

Table 3 (moved from the main body for space) lists the full block-diagonal control grid with per-seed standard deviations.

Table 3: Block-diagonal attention control (Qwen3-0.6B), mean $\pm$ std over 3 seeds. The composition ordering from Table 1 disappears; all arms are within 0.007 nats of each other.

Policy	Resp. NLL, full attention	Resp. NLL, block-diagonal
DPP	1.5172 ± 0.0014	1.9537 ± 0.0051
Random	1.5189 ± 0.0005	1.9548 ± 0.0050
TFP (faithful)	1.5245 ± 0.0031	1.9474 ± 0.0017
TFP-homog	1.5413 ± 0.0022	1.9495 ± 0.0033

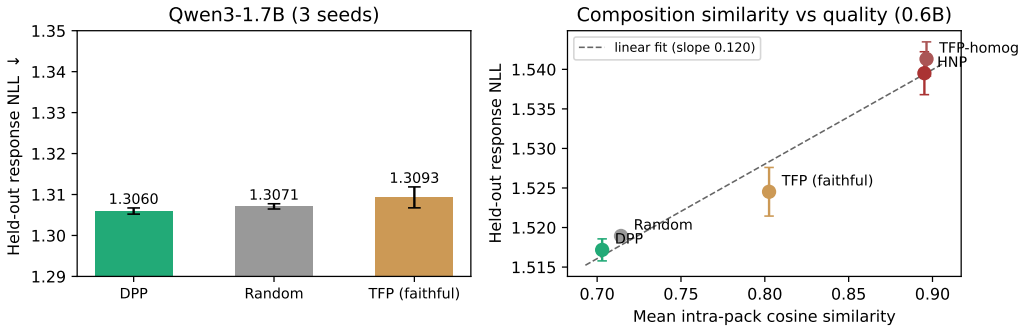

Figure 2: Left: Qwen3-1.7B scale check (3 seeds). The DPP–TFP ordering replicates; the DPP–random gap shrinks to 0.0011 nats. Right: mean intra-pack cosine versus response NLL at 0.6B; the policies fall on a monotone trend.