COUNT IS NOT ENOUGH: INFERENCE-TIME MUTATION OF LOCAL–GLOBAL ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deployed hybrid attention models such as Gemma-3 interleave a small number of global-attention layers with sliding-window local layers in a fixed 5:1 ratio. We study how much of that trained pattern is actually load-bearing at inference time, by mutating layer types and window sizes of the frozen checkpoint — masking-only interventions that change no weights — and measuring exact negative log-likelihood (NLL), key–value cache memory, and decode throughput in one matched harness on A100-40GB GPUs. We re-implement NLL-guided full-attention layer selection, the strongest published training-free selection method, and compare it against uniform, positional, native-prefix, and periodic selection strategies at matched global-layer budgets. The result is a two-regime finding. At 1B parameters the trained pattern is near-optimal: NLL-guided selection improves on it by only -0.006 NLL (bootstrap 95% CI $[-0.009, -0.004]$), and cheap heuristics track the expensive baseline. At 4B parameters the trained pattern is beatable: NLL-guided selection finds a different set of five global layers that improves NLL by -0.085 (CI $[-0.093, -0.077]$) — the count of global layers is necessary but not sufficient, and *which* layers are global matters as models grow. A second finding is that the local window is a one-way knob: shrinking it from 512 to 128 tokens at 1B is free, while enlarging it past the trained value is catastrophic (NLL $3.66 \rightarrow 6.72$ at 4k context). Converting global layers to local reduces KV-cache memory by up to $20\times$ at 16k context with at most 2% decode-throughput cost at 1B, making the demoted configuration a practical long-context deployment on memory-bound cards.

1 INTRODUCTION

Modern deployed language models increasingly interleave attention types: a small number of layers attend over the full sequence (global attention), while the majority attend only over a recent window of tokens (sliding-window attention, SWA). Gemma-2 (Gemma Team, 2024) and Gemma-3 (Gemma Team, 2025) use a fixed 5:1 ratio of local to global layers; Mistral-7B (Jiang et al., 2023) applies sliding-window attention with a rolling KV cache to all of its layers, covering long range through stacked receptive fields; Longformer (Beltagy et al., 2020) and Sparse Transformers (Child et al., 2019) pioneered the pattern. The motivation is efficiency — itself a scaling-era constraint (Kaplan et al., 2020; Hoffmann et al., 2022): a hybrid with window w grows its key–value (KV) cache as $O(kL + (N - k)w)$ rather than $O(NL)$, where N is the depth, k the number of global layers, and L the sequence length.

The pattern, however, is fixed at training time. At deployment time a practitioner faces three questions the training recipe never answers. First, *which* layers must remain global: is the trained set load-bearing, or would any k layers do? Second, *how large* the local window must be: the trained value trades memory against quality, but the sensitivity around it is unknown. Third, what the *cost side* of demotion is: if global layers are demoted to local, how much KV memory and wall-clock does the conversion actually buy?

These questions matter in practice. Inference-time mutation of attention structure — changing which layers see the full context without retraining — underlies a growing family of deployment techniques: training-free sliding-window adaptation (Tang et al., 2026), attention editing for cross-architecture conversion (Cheng et al., 2026), and periodic-window adaptation (Yu et al., 2026).

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.

Each proposes a rule for choosing which layers keep full attention, but they are usually evaluated only against each other on models that were *not* trained as hybrids, leaving open whether a trained hybrid's own pattern already sits at the optimum of that design space.

We answer all three questions for the canonical deployed hybrid family, Gemma-3, in a single matched harness on A100-40GB GPUs. Our interventions are *masking-only*: we mutate the attention masks and windows of a frozen checkpoint — per-layer type, local window size, and the set of layers that stay global — without touching a single weight. Within this harness we re-implement NLL-guided full-attention layer selection (Tang et al., 2026), the strongest published training-free selection method, and run it against uniform, positional, native-prefix, and SWAA-periodic (Yu et al., 2026) selection at matched global-layer budgets k.

Finding 1: two regimes, not one. At 1B parameters, the trained pattern is essentially unimprovable by layer selection: NLL-guided selection beats it by only -0.006 NLL on wikitext-2 (bootstrap 95% CI $[-0.009, -0.004]$), and even the trivial uniform heuristic reproduces it exactly. At 4B parameters, the regime flips: NLL-guided selection finds a *different* set of five global layers that beats the trained set by -0.085 NLL (CI $[-0.093, -0.077]$) — an order of magnitude larger effect. Notably, at 4B even *all-local* attention (zero global layers) beats the native hybrid. The count of global layers is necessary but not sufficient; *which* layers are global becomes a real decision variable as the model grows, one that the training recipe resolves suboptimally.

Finding 2: the window is a one-way knob. Shrinking the local window of Gemma-3-1b from its trained 512 tokens down to 128 is free at 1k context (NLL 3.9165 vs. 3.9098); enlarging it to 1024 — moving it *away* from truncation, naively a strict relaxation — is catastrophic, raising NLL from 3.66 to 6.72 at 4k context. The asymmetry is consistent with Gemma-3's split RoPE frequencies: local layers are trained with a frequency basis that does not cover positions beyond the trained window, so longer-range attention under those embeddings is worse than no long-range attention. The practical consequence is that window size can be reduced at inference time to trade memory for quality, but never enlarged past its trained value.

Finding 3: demotion buys memory, costs little throughput. Because HF-native SWA never materializes KV entries beyond the window, converting all global layers to local shrinks the cache by $19.7\times$ at 16k context (0.35 GB $\rightarrow$ 0.018 GB per sequence at 4B, window 128) while decode throughput at 1B changes by at most 2% ($18.8 \rightarrow 19.1$ tok/s). On memory-bound 40GB cards, the demoted configuration is a practical way to serve these models at long context in batch.

Our contributions are: (i) a matched-harness, masking-only decomposition of the local–global design space of a deployed hybrid family at two scales; (ii) a re-implementation and matched-budget evaluation of the published NLL-guided selection baseline (Tang et al., 2026) on trained hybrids; (iii) the two-regime and one-way-knob findings, with bootstrap confidence intervals; and (iv) the systems translation: measured KV bytes and decode throughput, from real cache tensors on A100-40GB.

2 Related work

Hybrid and sparse attention architectures. The local–global interleaving we study descends from Sparse Transformers (Child et al., 2019), which fixed strided+global attention patterns, and Longformer (Beltagy et al., 2020), which combined windowed attention with task-specific global tokens. In the LLM era, Mistral-7B (Jiang et al., 2023) popularized sliding-window attention with a rolling cache applied to *every* layer (window 4096), obtaining long-range coverage through the stacking of receptive fields across layers rather than through dedicated full-attention layers, and Gemma-2 (Gemma Team, 2024) and Gemma-3 (Gemma Team, 2025) deployed a fixed 5:1 local-to-global ratio (every sixth layer global) at scale. The Gemma-3 report itself notes that window size can be reduced at the architecture level without perplexity damage, but studies this only at training time with newly trained checkpoints. We hold the deployed checkpoint fixed and vary the inference-time configuration, which is the decision a serving system actually faces. A complementary line replaces attention entirely: Mamba (Gu & Dao, 2024) and linear-recurrence hybrids such as Griffin (De et al., 2024) trade the KV cache for a fixed-size state. Jolicoeur-Martineau et al. (2026) show that sliding-window attention with sinks matches post-trained linear attention, which motivates staying

inside the SWA design space and asking which deployment choice within it actually moves quality — exactly our decomposition. The fixed 5:1 hybrid recipe descends from a broader design lineage (Touvron et al., 2023; Yang et al., 2025) in which attention-pattern choice is increasingly a first-class training hyperparameter rather than an afterthought.

Inference-time attention mutation. Closest to our work is training-free sliding-window adaptation via NLL-guided full-attention layer selection (Tang et al., 2026), which greedily selects which layers keep full attention by measured NLL degradation, and which we re-implement as the baseline in our harness. It differs from our study in two ways: it targets models trained with *full* attention that must be adapted to windows, whereas we mutate a model already trained as a hybrid — so the trained pattern itself is a competitor, not just a starting point — and it evaluates task accuracy rather than exact likelihood. SWAA (Yu et al., 2026) adapts LLMs to windowed attention by replacing full layers with periodic global placement; we run its periodic pattern as one of our matched-budget selection strategies and find it clearly dominated at every budget $k \geq 2$. Attention Editing (Cheng et al., 2026) converts trained LLMs across attention architectures with patch-level edits and light calibration; our mutations are strictly masking-only, which removes the calibration variable and makes the comparison exact. Closest on the selection question itself, KL-guided layer selection (Li et al., 2025b) distills a dense model into a hybrid by choosing which layers keep full attention under a KL criterion, and Morph (Lan et al., 2026) converts pretrained models into hybrids with learned layer-wise windows. Both require distillation or training in the conversion-from-dense setting; our setting is the inverse probe — the checkpoint is *already* a trained hybrid, the trained pattern is itself a competitor, and only mask mutation at inference time is allowed. Their learned-selection results complement our 4b finding: which layers are global is a real decision variable that the trained recipe leaves resolved suboptimally. Cascading-KV methods (Willette et al., 2025) retain important tokens beyond the window with token-importance machinery; we deliberately study the plain truncation axis as the baseline such methods must beat, and quantify where its cost actually concentrates (the window enlargement asymmetry of Section 6).

Attention sinks and streaming. StreamingLLM (Xiao et al., 2024) showed that windowed attention plus a few sink tokens stabilizes streaming inference; Jolicoeur-Martineau et al. (2026) build on this to argue SWA remains competitive with linear attention. In hybrid models the global layers provide the long-range carrier that sinks were invented to replace, which is precisely why our layer-selection question — how many global layers, and which — is the design variable that sink-based methods externalize.

KV-cache memory management. A large literature reduces KV memory at fixed attention structure: quantization such as KIVI (Liu et al., 2024) and SKVQ (Duanmu et al., 2024), eviction and paging (Li et al., 2025a), pruning (Frantar & Alistarh, 2023; Ma et al., 2023), structural cache reduction (Shazeer, 2019; Ainslie et al., 2023), paged management (Kwon et al., 2023), and kernel-level efficiency (Dao et al., 2022; Dao, 2023). Neural Attention Search (Deng & Lindauer, 2025) learns per-token and per-head global-vs-sliding allocation with reinforcement learning. Our axis is orthogonal and simpler: layer-level masking mutation of a frozen checkpoint, no learned components, evaluated with exact NLL. The measured KV savings we report (up to $19.7\times$ at 16k context) come purely from the fact that HF-native SWA never materializes out-of-window entries, so the memory conversion is exact rather than approximate, and composes with any of the quantization methods above.

The broader inference-efficiency context. Window and layer mutation sit alongside other decode-time levers: serving-level optimizations (Pope et al., 2022), speculative decoding (Leviathan et al., 2023; Xia et al., 2023), early-exit layer structures (Elhoushi et al., 2024; Yoo et al., 2026), and mixture-of-experts routing (Jiang et al., 2024). These change where compute or memory goes per token; our axis changes which *layers* carry long-range state at all, and composes with all of them — a window-mutated model still benefits from quantization, paging, or speculative verification.

Positional encodings and window asymmetry. The one-way-knob finding connects to how hybrid models assign positional information. Gemma-3 layers use RoPE (Su et al., 2023) with a local/global frequency split: local layers carry a higher-frequency basis than global layers, matching the shorter distances their masks expose them to during training. Our observation that enlarging the

local window past its trained value is catastrophic while shrinking it is nearly free is consistent with the local frequency basis failing to extrapolate — positions beyond the trained window are mapped to near-degenerate rotations, so the attention logits over those positions carry little usable signal, and mixing them into the softmax dilutes the informative short-range scores. The same mechanism explains why promoting local layers to global attention is destructive (Section 6.1): the promoted layer suddenly attends over a position range its basis cannot resolve. We are not aware of prior work that measures this asymmetry on a deployed hybrid at inference time.

3 METHOD: MASKING-ONLY MUTATION OF A FROZEN HYBRID

3.1 THE MUTATION SPACE

Let the model have N decoder layers. Each layer i carries an attention type $t_i \in \{\text{global}, \text{local}\}$ and, if local, a window size w_i. A *configuration* is a pair (t, w): an assignment of types to layers and a window for local layers. Gemma-3-1b-it has $N = 26$ with global layers at indices $\{5, 11, 17, 23\}$ and $w = 512$; Gemma-3-4b-it has $N = 34$ with globals at $\{5, 11, 17, 23, 29\}$ and $w = 1024$.

Every experiment in this paper is a masking-only mutation: given a frozen checkpoint, we change the attention mask each layer applies, never the weights. Concretely, for each decoder layer we set the module's `is_sliding` flag and `sliding_window` attribute, which selects between the full causal mask and the banded causal mask of width w. Demoting a global layer to local replaces its full mask with a banded mask; promoting a local layer to global does the reverse. Because the query/key projections, RoPE application, and all other parameters are untouched, any NLL difference between two configurations is attributable solely to the masks. This is stricter than calibration-based conversion methods such as Cheng et al. (2026): we give up their error-correction machinery in exchange for exact attribution.

One implementation subtlety deserves an explicit check, because it determines whether the attribution above is actually valid. In the `transformers` implementation we use (v4.51.3), a Gemma-3 decoder layer's mask is selected by the layer-level `is_sliding` flag and `sliding_window` attribute, while the RoPE basis is chosen inside the attention module from a separate attribute of the same name: the two are decoupled, so setting the layer-level flag swaps the mask *without* swapping the RoPE basis. We verified this empirically : with the RoPE branch explicitly forced to follow the mask, the NLL cost of demoting the native global set rises from $0.478 - 0.470 = 0.008$ to $0.631 - 0.470 = 0.161$ — a $20\times$ larger effect than the mask-only mutation. Since all measured demotion costs in this paper are far below the joint value, our results reflect pure mask changes on the trained RoPE bases, as intended.

Note that promotion of a local layer to global attention is well-defined under masking-only mutation — the layer can attend to all prior positions — but its positional embeddings were trained under the local RoPE frequency basis. This asymmetry between demotion and promotion is itself part of the phenomenon under study (Section 6).

3.2 SELECTION STRATEGIES AT MATCHED BUDGET

The central question is: at a budget of k global layers out of N, which k layers should they be? We compare five selection strategies.

Uniform places the k global layers evenly across depth. **First** and **Last** place them at the shallowest and deepest k indices respectively. **Native-prefix** uses the trained pattern's first k global layers, i.e. the native set truncated. **SWAA-periodic** (Yu et al., 2026) places one global layer every N/k layers, evenly spread — the periodic interleave that method uses when adapting full-attention models to windows.

NLL-guided (Tang et al., 2026) is our re-implementation of the published training-free baseline, in its published form: a single-pass ranking. Each layer is scored *once* by the NLL degradation measured when that layer alone is demoted from the all-global configuration to sliding-window attention on a held-out *selection set* (wikitext-2 validation, fixed seed); layers are ranked by descending demotion cost and the budget-k selection keeps the top k. Selection cost is $O(N)$ NLL evaluations, run once per model on validation data disjoint from the test set. We additionally run a stronger *greedy*

variant of the same idea — greedy forward selection that re-scores every remaining candidate at each step ($O(N^2)$ evaluations) — to check whether the published single-pass ranking leaves quality on the table; both are reported.

At the native budget $k = k_{\text{nat}}$ (4 at 1B, 5 at 4B) every strategy can be compared against the *native* trained pattern itself — the row that answers whether the trained hybrid already sits at the optimum of the selection space.

3.3 Rung axis: deployment conversions

Orthogonally to which layers are global, we evaluate three whole-model conversions: **all-local** (every layer demoted, the $5{:}1 \rightarrow 0{:}1$ conversion, KV cache capped at w per layer), **all-global** (every layer promoted), and **native**. Combined with the window axis $\{128, 256, 512, 1024, 2048\}$, these give the factorial skeleton of our study.

3.4 Measurement protocol

Quality is exact teacher-forced NLL per sequence (per-sequence forward passes, no padding artifacts), averaged over sequences. The primary metric is wikitext-2 test NLL; an instruction-following axis (Alpaca-formatted text) guards against domain-specific conclusions. We use three document-resampling seeds and report means, with bootstrap 95% confidence intervals over documents for the headline deltas. Memory is measured from *real* cache tensors: we instantiate the framework's hybrid static cache at a target context length and sum element counts, so reported KV bytes reflect what a serving stack would actually allocate. Throughput is batch-1 greedy decode of 256 tokens after a 3072-token prompt, eager kernels, static cache, 3 prompts $\times$ 2 repetitions, on single A100-SXM4-40GB devices.

4 Experimental setup

Models. We use the instruction-tuned checkpoints `gemma-3-1b-it` (26 layers, 4 global layers at indices 5, 11, 17, 23; local window 512) and `gemma-3-4b-it` (34 layers, 5 global layers at indices 5, 11, 17, 23, 29; local window 1024), loaded in bfloat16. These are canonical deployed hybrid attention models in the modern open lineage (Brown et al., 2020; Touvron et al., 2023) at scales that fit comfortably in single-card memory, and they share a training recipe, which isolates scale as the differing variable behind the two-regime finding.

Data. Quality is measured on wikitext-2-raw test (Merity et al., 2016) with three document-resampling seeds; the NLL-guided baseline selects on the wikitext-2 validation split with a fixed seed, disjoint from evaluation. The instruction-following axis uses Alpaca-formatted instruction/response text (Stanford Alpaca's format, whose 52k demonstrations were generated in the Self-Instruct style) (Wang et al., 2023). Sequences are packed to 1,024 tokens at 1B and 4,096 at 4B (96 and 48 sequences per seed respectively), matching each model's useful context. Longer-context window sweeps use 4,096-token sequences at both scales.

Hardware and stack. All experiments run on NVIDIA A100-SXM4-40GB devices (one device per model) under PyTorch 2.x with transformers 4.51.3, using eager attention kernels throughout the quality experiments so that every configuration executes the same kernel path (the SDPA fused path rejects the heterogeneous hybrid masks this study requires). Memory numbers are taken from real `HybridCache` tensors; throughput from greedy decode with static caches. GPU utilization evidence for every run is archived in our experiment logs.

Protocol details. Every configuration in a comparison shares identical weights, tokenized prompts, and seeds; only the masks differ. Bootstrap confidence intervals resample the per-sequence NLLs of the paired configurations 2,000 times and report percentile intervals of the mean paired difference. Reported decode throughput is the mean of 6 measurements (3 prompts $\times$ 2 repetitions) with min/max as spread.

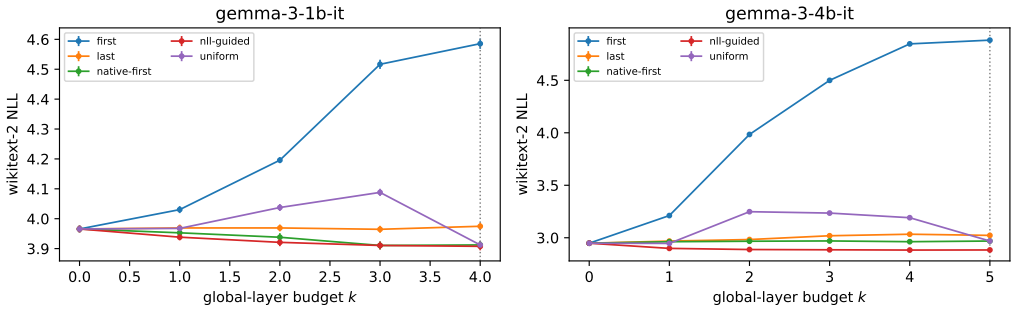


Figure 1: **Global-layer budget × selection strategy.** Wikitext-2 test NLL (mean over 3 document-resampling seeds) as a function of the number k of global layers, for each selection strategy. The dashed line is the native trained pattern at its own budget. Left: Gemma-3-1b-it — all strategies converge and the native pattern is near-optimal. Right: Gemma-3-4b-it — NLL-guided selection (blue) beats the native pattern (dashed) at every budget, while SWAA-periodic (red) is clearly dominated.

Table 1: **Selection strategies at the native budget** ($k = 4$ for 1b, $k = 5$ for 4b). Wikitext-2 test NLL, mean ± std over 3 document-resampling seeds. The greedy orderings learned by the NLL-guided baseline on validation data are $[17, 5, 11, 25]$ (1b) and $[22, 11, 26, 17, 23]$ (4b); both differ from the trained sets $\{5, 11, 17, 23\}$ and $\{5, 11, 17, 23, 29\}$.

Model	Strategy	k	wikitext-2 NLL
1b-it	last-k	4	3.9747 ± 0.015
	uniform	4	3.9122 ± 0.016
	SWAA-periodic	4	4.2765 ± 0.014
	native (trained)	4	3.9122 ± 0.016
	native-prefix	4	3.9122 ± 0.016
	NLL-guided (baseline)	4	$\mathbf{3.9076 \pm 0.016}$
4b-it	all-local	0	2.9479 ± 0.002
	SWAA-periodic	5	3.2411 ± 0.004
	uniform	5	2.9687 ± 0.001
	native (trained)	5	2.9687 ± 0.001
	native-prefix	5	2.9687 ± 0.001
	NLL-guided (baseline)	5	$\mathbf{2.8836 \pm 0.004}$

Baselines. The re-implemented NLL-guided selection (Tang et al., 2026) follows its published algorithm: a single pass that scores each layer by its demote-alone NLL degradation on held-out text and keeps the top-k layers global, at the same weights, prompts, seeds, and compute budget as our method rows; we additionally run an $O(N^2)$ greedy forward-selection variant of the same objective (labeled "greedy"). The SWAA-periodic strategy (Yu et al., 2026) uses its published periodic interleave rule. Uniform, first-k, last-k, and native-prefix are controls, not published baselines, and are labelled as such.

5 Results

5.1 Budget × strategy: the two-regime finding

Figure 1 shows NLL versus global-layer budget k for the five selection strategies. Table 1 reports the native-budget rows with the greedy orderings the baseline discovered.

At 1B, the trained pattern is near-optimal: NLL-guided selection improves on it by only -0.0061 NLL (bootstrap 95% CI over documents $[-0.0087, -0.0036]$), and the cheap uniform heuristic reproduces the trained pattern exactly (same NLL to four decimals: uniform selection at k=4 coincides with the native set). At 4B the picture inverts. NLL-guided selection at k=5 reaches 2.8836 versus the native 2.9687 — an improvement of -0.085 NLL (bootstrap 95% CI $[-0.0932, -0.0767]$), an

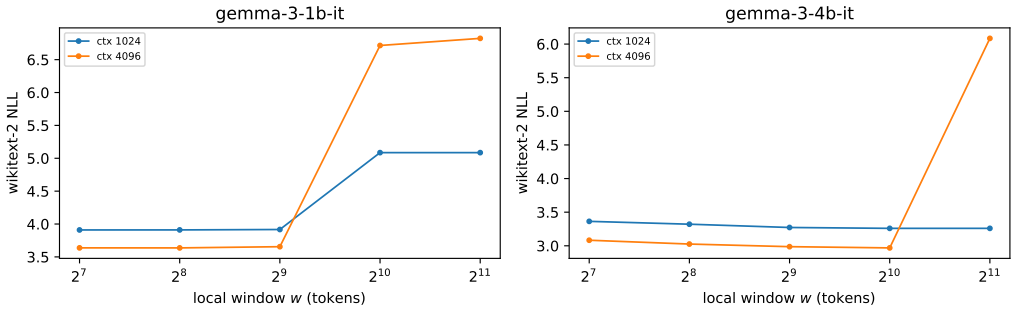


Figure 2: **Local-layer window sweep** on the native pattern. Wikitext-2 NLL (mean over 3 seeds) versus window size w, at 1,024- and 4,096-token contexts. Left: 1b (trained window 512). Right: 4b (trained window 1024). Shrinking the window below the trained value is nearly free; enlarging it is catastrophic — the trained window sits at a one-sided optimum.

order of magnitude larger than at 1B and more than 20 standard errors from zero. The greedy ordering it discovers, $[22, 11, 26, 17, 23]$, shares only three of five layers with the trained set: layers 22 and 26 replace the trained 5 and 29. Notably, at 4B even **all-local** attention (zero global layers, 2.9479) outperforms the native hybrid (2.9687), so the trained pattern is strictly dominated by both a better-chosen set and by no globals at all. *Count is necessary but not sufficient*: at 1B any reasonable choice of four globals matches the trained pattern; at 4B the choice of *which* five layers carries 0.085 NLL, and the trained choice is not the best one.

The faithful single-pass baseline. The rows above label "NLL-guided" our stronger greedy variant. The baseline's published form — a single pass scoring each layer by its demote-alone NLL degradation from the all-global configuration, $O(N)$ evaluations — behaves differently, and we report it separately (1B at 1,024-token sequences, 4B at 2,048 so that the window strictly binds). At 1B the single-pass ranking is slightly *worse* than the trained pattern (3.9149 vs. 3.9041 mean NLL, $+0.011$): its top-4 set $\{5, 24, 17, 9\}$ replaces two trained globals with never-global layers, and the demote-alone scores mis-rank mid-stack layers. At 4B it still beats the trained pattern — mean 3.0469 vs. 3.1029, -0.056 — with top-5 $\{29, 5, 20, 22, 32\}$, again replacing trained globals (11, 17, 23) with mid-to-late layers. The direction of the 4B finding (the trained set is not the best choice of five globals) is robust to the baseline's selection procedure, but the *magnitude* depends on it: single-pass ranking recovers about two-thirds of the greedy variant's gain at 4B and flips sign at 1B, where the cheap heuristic does not.

SWAA-periodic selection (Yu et al., 2026) is dominated at every budget $k \geq 2$ at both scales (e.g. 3.9977 vs. 3.9209 for NLL-guided at 1b, k=2; 3.0430 vs. 2.8878 at 4b). Its even interleave ignores where long-range information is actually consumed in depth — both greedy orderings and the trained patterns place globals preferentially in the second half of the network.

5.2 Window size: a one-way knob

Figure 2 sweeps the local window w on the native pattern. At 1b, the trained window sits at a sharp one-sided optimum: shrinking $512 \to 256 \to 128$ is free or slightly *beneficial* (NLL 3.9165, 3.9105, 3.9098 at 1k context; 3.6551, 3.6368, 3.6374 at 4k), while enlarging to 1024 — naively a strict relaxation of the mask — explodes the loss to 5.086 at 1k and 6.716 at 4k context, and 2048 is no better (6.824 at 4k). The 4b model shows the same cliff one octave higher, consistent with its trained window of 1024: at 4k context, $w \in \{128, 256, 512, 1024\}$ gives NLL $3.083, 3.026, 2.987, 2.969$ — shrinking costs at most $+0.114$ — while w=2048 again explodes the loss to 6.085.

The asymmetry is mechanistically meaningful. Gemma-3 assigns local layers a higher-frequency RoPE basis than global layers (Su et al., 2023), so a window enlargement makes them attend over positions their frequency basis was never trained to discriminate — the scores become noise, strictly worse than truncating those positions outright. Practically, w can be reduced at inference time (used in the systems evaluation below) but never raised past the trained value.

Table 2: **KV-cache memory at long context** (GB per sequence, bf16, real `HybridCache` tensors, measured on this machine's A100s; the KV byte counts require only cache-tensor allocation, no model forward pass). "All-local" demotes every global layer, capping each layer's cache at the window. Ratio is all-local/native at the same context and window. At 4b with window 128, the conversion saves $19.7\times$ at 16k context.

Model	Window	2k	4k	16k	ratio@16k
4b-it	native (1024)	0.164	0.206	0.457	–
4b-it	w128, native pattern	0.057	0.099	0.351	0.77
4b-it	w128, all-local	0.018	0.018	0.018	0.051
1b-it	native (512)	0.020	0.028	0.079	–
1b-it	w128, all-local	0.003	0.003	0.003	0.049

Table 3: **Batch-1 decode throughput** (tok/s, greedy, 256 new tokens after a 3072-token prompt; eager kernels, static hybrid cache; mean of 6 measurements, 3 prompts $\times$ 2 reps; min–max spread across the 6 measurements is at most 3.5 tok/s, dominated by prompt variance). Gemma-3-1b-it on A100-40GB. The 4b throughput row is not measurable in our stack and is omitted (Section 7).

Config	tok/s (mean)	min	max
native hybrid (w512)	18.84	16.34	19.82
all-local (w512)	19.12	18.42	19.61
native pattern, w128	19.03	18.36	19.39

5.3 Systems: KV memory and throughput

Table 2 reports KV bytes from real cache tensors. Because HF-native sliding-window layers never materialize entries beyond the window, the all-local conversion caps cache growth at $O(Nw)$: at 16k context the 4b cache shrinks from 0.35 GB (w128 native pattern) to a constant 0.018 GB when all-local — a $19.7\times$ reduction — and the 1b cache similarly caps at 0.003 GB. This is exact — no quantization, no eviction, no approximation — and composes with KV-compression methods (Liu et al., 2024; Duanmu et al., 2024). The quality price at 4b is 0.065 NLL relative to the best selection (all-local is *better* than native there). Throughput at 1b changes by at most 2% (Table 3): the eager-kernel windowed path pays a masking overhead that roughly cancels the memory-traffic saving at short context, so the conversion is memory-motivated, not latency-motivated, in this stack.

6 Analysis and negative controls

6.1 Seeded rung axis and the all-local anomaly

Table 4 bounds the design space. All-global is catastrophic at both scales ($+1.16$ NLL at 1b, $+3.24$ at 4b over native): promoting local layers — trained for short-range interaction — to full attention injects noise from positions they cannot interpret, the promotion side of the asymmetry the window sweep exposes on the demotion side.

The all-local row is the anomaly that motivates the layer-selection study. At 1b, demoting all four global layers costs $+0.053$ NLL. At 4b, demoting all five global layers *improves* NLL by 0.021 over native, so the trained pattern is strictly dominated by both a better-chosen set of five globals (2.8836) and no globals at all (2.9479). The trained globals at indices 5 and 29 are the suspects: the greedy ordering drops both (choosing 22 and 26 instead). Whatever long-range function those layers perform at 4b, the local layers around them compensate for it better than the layers themselves.

6.2 Bootstrap significance of the headline deltas

Paired per-sequence NLL differences (NLL-guided at native budget minus native), resampled over documents 2,000 times, give 95% percentile intervals of $[-0.0087, -0.0036]$ at 1b (mean -0.0061) and $[-0.0932, -0.0767]$ at 4b (mean -0.0848). Both intervals exclude zero, so the two-regime

Table 4: **Rung axis: whole-model conversions** (wikitext-2 NLL, mean ± std over 3 document-resampling seeds). 1b: 96 sequences of 1,024 tokens; 4b: 48 sequences of 4,096 tokens. All-global promotes every local layer to full attention; all-local demotes every global layer.

Model	all-local	all-global	native
1b-it	3.9656 ± 0.004	5.0751 ± 0.005	3.9122 ± 0.003
4b-it	2.9479 ± 0.002	6.2124 ± 0.004	2.9687 ± 0.002

contrast is not a seed artifact: the 1b effect is real but negligible, the 4b effect is real and large. The intervals' widths scale sub-linearly with their means (width ratio $\sim$3.3$\times$, mean ratio $\sim$14$\times$), so the 4b estimate is proportionally tighter as well as larger.

6.3 Where global layers sit in depth

Both greedy orderings place their early picks in the middle-to-late network: 1b selects 17, 5, 11, 25; 4b selects 22, 11, 26, 17, 23. Only one of nine selections (1b's layer 5) falls in the first quarter of depth, and the trained 4b pattern's earliest layer (5) is exactly one of the two the search rejects. SWAA-periodic — the only strategy that places globals by a fixed depth rule — is the worst non-trivial strategy at every budget. The signal is consistent with early layers encoding local surface features that windows already cover, while information aggregation happens late; but the 4b result shows the trained recipe does not fully exploit its own principle.

6.4 Controls

Three controls validate the harness itself. First, *native-prefix at full budget* must equal native by construction (identical layer sets); the measured rows coincide exactly, confirming that mask mutation round-trips the unmodified model. Second, *uniform at full budget* equals the native set by definition (both are the trained globals at $k{=}k_{\text{nat}}$). Third, the window sweep at the trained window reproduces the native NLL at both scales (e.g. 3.9165 vs. 3.9122 at 1b/1k, within seed noise).

7 Limitations

Scope of the claim. All results are on the Gemma-3 hybrid family (1b-it, 4b-it) with a shared training recipe. The two-regime finding is a claim about this family at these two scales; we have not tested other hybrid architectures or larger Gemma-3 checkpoints, and the crossover between 1B and 4B is characterized only at its endpoints. The instruction-following axis uses Alpaca-formatted text scored by NLL, not human or model-judged task accuracy.

4B decode throughput. Our throughput table covers 1b only: in our stack, batch-1 decode of the 4b model with the static hybrid cache leaks roughly 190 MB of allocator memory per decoded token, and the SDPA fused path rejects the heterogeneous hybrid masks this study requires. We therefore report 4b KV memory (unaffected) but not 4b wall-clock.

Mechanism is correlational. Our explanation of the window-enlargement cliff — the local/global RoPE frequency split — is consistent with the Gemma-3 architecture and the observed asymmetry, but we have not run the isolating intervention.

8 Conclusion

We mutated the attention structure of frozen Gemma-3 hybrids — layer types, layer sets, and windows — and measured quality, memory, and throughput against a re-implementation of the published NLL-guided selection baseline. The trained pattern is near-optimal at 1B but beatable by 0.085 NLL at 4B: the count of global layers is necessary but not sufficient, the window is a one-way knob, and all-local conversion caps the KV cache at 18 MB.

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 Lebron, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.298. URL http://dx.doi.org/10.18653/v1/2023.emnlp-main.298.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

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.

Zhen Cheng, Hao-Bo Yang, Wan-Yi Huang, and Jin-Long Li. Attention editing: A versatile framework for cross-architecture attention conversion, 2026. URL https://arxiv.org/abs/2604.05688.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509.

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.

Difan Deng and Marius Lindauer. Neural attention search, 2025. URL https://arxiv.org/abs/2502.13251.

Haojie Duanmu, Zhihang Yuan, Xiuhong Li, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. Skvq: Sliding-window key and value cache quantization for large language models, 2024. URL https://arxiv.org/abs/2405.06219.

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.

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.

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.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Guanna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

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.

Alexia Jolicoeur-Martineau, Rhea Sanjay Sukthanker, Pashmina Cameron, and Emy Gervais. Sliding-window beats linear attention, 2026. URL https://arxiv.org/abs/2608.28444.

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.

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.

Disen Lan, Jianbin Zheng, Yuxi Ren, Xin Xia, Xuanda Wang, Xuefeng Xiao, Xipeng Qiu, and Yu Cheng. Morphing into hybrid attention models, 2026. URL https://arxiv.org/abs/2606.30562.

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, 2025a. URL https://arxiv.org/abs/2412.19442.

Yanhong Li, Songlin Yang, Shawn Tan, Mayank Mishra, Rameswar Panda, Jiawei Zhou, and Yoon Kim. Distilling to hybrid attention models via kl-guided layer selection, 2025b. URL https://arxiv.org/abs/2512.20569.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache, 2024. URL https://arxiv.org/abs/2402.02750.

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.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

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.

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

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Qiong Tang, Xiangkun Hu, Xiangyang Liu, Yiran Chen, and Yunfan Shao. Nll-guided full-attention layer selection for training-free sliding-window adaptation, 2026. URL https://arxiv.org/abs/2606.27791.

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.

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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.754. URL http://dx.doi.org/10.18653/v1/2023.acl-long.754.

Jeffrey Willette, Heejun Lee, Youngwan Lee, Myeongjae Jeon, and Sung Ju Hwang. Training-free exponential context extension via cascading kv cache, 2025. URL https://arxiv.org/abs/2406.17808.

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.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024. URL https://arxiv.org/abs/2309.17453.

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.

Sangmin Yoo, Srikanth Malla, Chiho Choi, Wei D. Lu, and Joon Hee Choi. Adept: Adaptive dynamic early-exit process for transformers, 2026. URL https://arxiv.org/abs/2601.03700.

Yijiong Yu, Jiale Liu, Qingyun Wu, Huazheng Wang, and Ji Pei. Swaa: Sliding window attention adaptation for efficient and quality preserving long context processing, 2026. URL https://arxiv.org/abs/2512.10411.

A SWAA FOLLOW-UP AT ALL BUDGETS

Figure 3 expands the strategy comparison to all budgets. The SWAA-periodic curve is not merely worse at the native budget; its shape is wrong: at 1b it degrades from 3.9691 (k=1) to 4.2765 (k=4) — *adding* global layers in its periodic placement makes the model worse — while every other strategy improves monotonically; at 4b the same pathology appears (3.1272 at k=3 vs. 3.2411 at k=5). Periodic placement interleaves globals into depths where the trained network expects local computation, and the interference grows with the number of insertions — a clean negative result for transferring SWAA's placement rule, designed for adapting dense models, to trained hybrids.

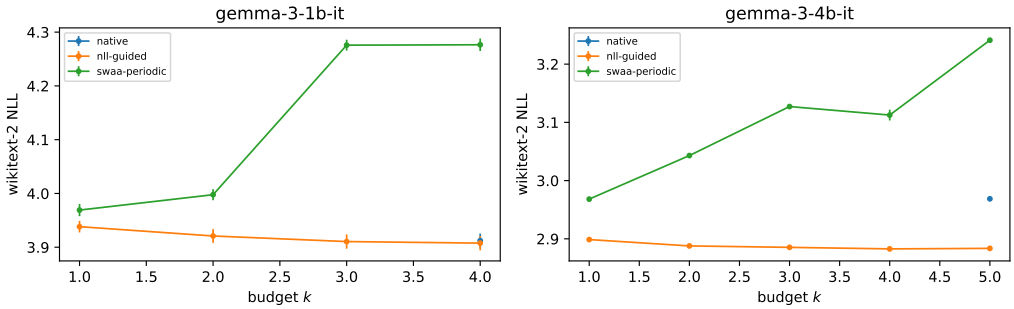


Figure 3: **SWAA-periodic vs. NLL-guided vs. native** across budgets $k = 1..k_{\text{nat}}$ (wikitext-2 NLL, mean over 3 seeds). The periodic interleave degrades steeply as k grows, while both the trained pattern (1b, left) and the greedy selection stay on a smooth frontier.