WHICH STATE SHOULD PREDICTION READ? A MECHANISTIC ANALYSIS OF STATE–PREDICTION SEPARATION

Anonymous authors
Paper under double-blind review

ABSTRACT

State–Prediction Separation (SPS) improves language modeling with separate persistent-state and next-token prediction slots, but sharing all weights leaves it unclear whether and how their roles specialize. We trace information flow after giving the streams separate weights. State concentrates predecessor information, while prediction shows stronger induction matching, retrieving continuations of earlier token occurrences. Ablating state's previous-token heads selectively weakens prediction's retrieval, and restricting access to older states worsens loss. Restricting prediction to the deepest state levels costs far less than restricting it to the shallowest, yet SPS never reads its top state layer. We therefore give every prediction layer the final state (a final-state read), a model we call Sequential. With each tower as deep as the Transformer, Sequential lowers seed-mean loss by 0.031 nats/token versus the same-depth read (Two-tower) at matched parameters and floating-point operations (FLOPs). A smaller Sequential model with half the layers per tower comes within 0.005 nats/token of SPS at 55% of its FLOPs and 0.064 below the Transformer at matched non-embedding parameters and FLOPs.

1 INTRODUCTION

In an autoregressive Transformer, each token's hidden state does two jobs: it stores context that later tokens read as keys and values (K/V), and it predicts the next token. Encoder–decoders separate contextualization from prediction (Vaswani et al., 2017), and K/V reuse across layers separates context construction from subsequent reads (Sun et al., 2024; Wu & Tu, 2024; Brandon et al., 2024). Machine translation has studied which encoder layers a decoder should read (Bapna et al., 2018; He et al., 2018; Domhan, 2018; Liu et al., 2021). However, this read-depth question has not been posed for causal language models that separate the two jobs.

State–Prediction Separation (SPS, Monea et al., 2026) is a decoder-only model that directly separates state and prediction into different token slots, forming state and prediction *streams*. SPS improves on the Transformer at about $1.8\times$ its FLOPs, but the streams share weights, obscuring their individual roles. Furthermore, prediction at each layer reads the state entering that layer, so the output of the top state layer is never read. We therefore ask: which computations does each stream perform, and which state should prediction read?

Mechanistic interpretability supplies tools for the first question. In single-stream Transformers, it has identified previous-token heads that encode predecessor information and induction heads that use it to retrieve continuations of earlier token occurrences (Elhage et al., 2021; Olsson et al., 2022; Wang et al., 2023). Activation and path patching test causal contributions (Meng et al., 2022; Goldowsky-Dill et al., 2023; Zhang & Nanda, 2024; Heimersheim & Nanda, 2024), ablation tests whether a component is needed although self-repair can mask it (McGrath et al., 2023), and probes and lenses show what information intermediate states make readable, though not whether the model uses it (Alain & Bengio, 2017; Belinkov, 2022; Belrose et al., 2023). In a single stream, however, storage and prediction share each hidden state, so these methods cannot cleanly assign a computation to either role. SPS's separate streams make that distinction explicit. We give the two streams separate weights and compare two ways for prediction to read state: a same-depth read (*Two-tower*), close

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.

to SPS's, and a read of the final state at every prediction layer (*Sequential*). We train each at 12+12 and 6+6, where $a{+}b$ means a state and b prediction layers. Our contributions show:

1. **Stream roles (§3).** Once the streams are separated, state carries predecessor information and prediction does more of the matching, with higher induction lift (attention to earlier continuations). Ablating state's previous-token heads selectively weakens prediction's retrieval, most clearly in the 12+12 Sequential model.
2. **What prediction needs from state (§4.1–§4.2).** Prediction relies on distant state held in the state stream, and losing its deep reads costs far more than losing its shallow ones. SPS never reads its top state layer.
3. **A read derived from the analysis (§5–§5.1).** Based on (1) and (2), we give every prediction layer the final state (Sequential). With separate weights at matched parameters and FLOPs, it lowers seed-mean loss versus the same-depth read (Two-tower) by 0.031 nats/token at 12+12 and 0.037 at 6+6.
4. **Efficiency (§6).** The smaller separate-weight Sequential 6+6 comes within 0.005 nats/token of SPS at the Transformer's FLOPs.

2 A Controlled Setting for State and Prediction

2.1 Transformer and SPS

The Transformer (Fig. 1a) is the single-stream baseline, with one slot per token and attention plus a multilayer perceptron (MLP) in each of 12 layers. SPS (Fig. 1b) separates token roles while sharing weights, using $2T$ alternating state and prediction slots for T tokens. Prediction slots receive a learned ⟨predict⟩ placeholder instead of the current token's embedding. Both slot types attend causally to same-document state and to preceding prediction slots within 64 token positions (the prediction-slot window), and our retrieval tests reach beyond this window. A *state level* is a state layer's output, with level 0 the embedding. Prediction slots in layer ℓ attend to state as it enters that layer (level $\ell{-}1$), so the top state layer's output is never read.

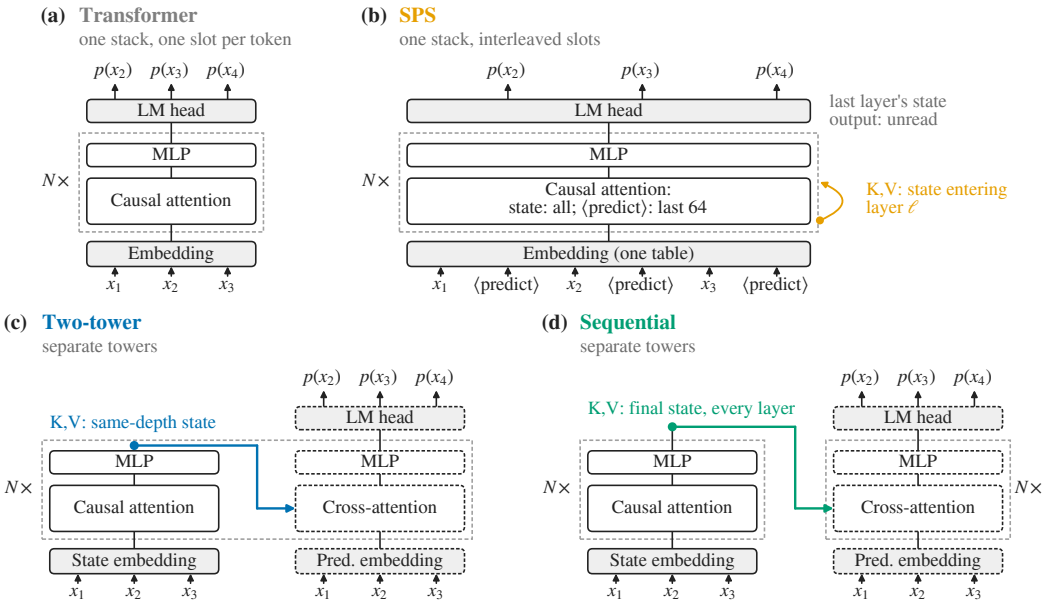


Figure 1: Architectures: (a) Transformer, (b) interleaved SPS slots, (c) Two-tower's same-depth read, and (d) Sequential's final-state read. Layer norms and residual connections are omitted. In (b), the 64-token window restricts prediction-slot keys only; state slots see all earlier state slots. Dashed boxes mark prediction towers, and grey dashed frames mark repeated layers.

2.2 TWO-TOWER AND SEQUENTIAL

To expose the stream roles obscured by SPS's shared weights, we compare separate-weight Two-tower and Sequential, which we call the *separated models* (Tab. 1, Fig. 1c,d). The main depths (a+b) are 12+12 and 6+6.[1] Two-tower uses a *same-depth read*, with prediction layer ℓ reading state level $r(\ell) = \ell$. Sequential uses a *final-state read*, with $r(\ell) = a$ at every prediction layer. Sequential is the design our analysis leads to (§5); we define it here so that §§3–4.2 can compare both reads. Each prediction layer has its own K/V projections of that state. State is causal over the same sequence, and prediction at t reads only state at positions $\leq t$, never attends to other prediction positions, and produces the loss. Shared Two-tower 12+12 and Shared Sequential 12+12 instead run SPS's single stack twice, over separate state and prediction sequences without interleaving or the prediction-slot window, and read each layer's input state or the final state, respectively (App. D.3). At 12+12, the separated models have twice the Transformer's non-embedding parameters at about SPS's FLOPs. At 6+6, they match the Transformer in both (§5).

Controls vary MLP allocation, attention sharing, tower splits, prediction input and head tying (definitions and per-seed results: App. A.2). AF-SPS (Almost-Free SPS, Langford et al., 2026) provides a lower-cost shared-MLP reference (App. A.3).

Table 1: Main models and controls: weights, state level read by prediction layer ℓ, prediction input, language-model (LM) output head, and seed count. Main depths: $a = b \in \{6, 12\}$. A tied head reuses input-embedding weights; an untied head has separate weights.

Model	Weights	State level read by prediction layer ℓ	Prediction input	LM head	Seeds
Main					
Transformer	–	–	–	untied	2
SPS	shared	$\ell-1$ (input to layer ℓ)	⟨predict⟩ token	untied	2
Two-tower a+b	separate	ℓ	own embedding	untied	2
Sequential a+b	separate	final	own embedding	untied	2
Controls					
Shared Two-tower 12+12	shared	$\ell-1$ (input to layer ℓ)	⟨predict⟩ token	tied	1
Shared Sequential 12+12	shared	final	⟨predict⟩ token	tied	2
Two-tower 12+12 (MLP realloc.)	separate	ℓ	own embedding	untied	1
Sequential 6+6 (⟨predict⟩ input)	separate	final	⟨predict⟩ token	untied	1
Sequential 6+6 (tied head, ⟨predict⟩ input)	separate	final	⟨predict⟩ token	tied	2
Two-tower 12+6 (top-aligned)	separate	ℓ+6	own embedding	untied	1

2.3 TRAINING AND EVALUATION

All models train on FineWeb-Edu, a filtered web-text corpus, for 20B tokens using the GPT-2 tokenizer, with width 768, 12 heads and context length 4096. The common learning-rate schedule decays over the final 2B tokens (App. A.1). We report negative log-likelihood (NLL) in nats/token, forward FLOPs/token, and total and non-embedding parameters. Results are seed means over two seeds unless marked, and $\pm$ is half the range over the two seeds, so mean $\pm$ spans both. Differences under 0.005 nats/token count as ties (App. A.1).

2.4 ANALYSIS METHODS

The analyses first separate predecessor encoding from matching, then test which state prediction needs (methods and control gates: App. A.4). *Previous-token mass*, the attention weight on the preceding token, measures whether heads encode each position's predecessor, and a *previous-token head* attends mainly there. *Induction lift*, attention to continuations of earlier occurrences of the current token divided by the uniform-attention baseline, locates matching (Olsson et al., 2022; Elhage et al., 2021). *Induction tokens* repeat an earlier bigram's continuation beyond SPS's prediction-slot window and could be copied from context.

[1] Layers and heads are numbered from 1.

To test whether prediction's retrieval depends on these heads, we ablate the three state heads with highest previous-token mass and compare the drop in induction-token accuracy against low-mass, random and pattern-matched control heads. Low-mass controls are three heads with low previous-token mass from the targets' layers, random controls are 100 draws of any three non-target heads, and pattern-matched controls draw among the heads nearest the targets in attention entropy, attended distance, first-token attention mass and output norm (App. A.4). Holm–Bonferroni correction (Holm, 1979) covers four tests: two control pools (random, pattern-matched), each with and without layer 1. *Activation patching* tests whether restoring one clean state-head output after corrupting an earlier repeated segment recovers retrieval.

Read restrictions limit prediction to the nearest N state positions or to the first or last m state levels, testing the reach and depth it needs. *Whole-layer lesions* zero a layer's residual contribution within a stream. Linear *role probes* measure, by held-out NLL, the previous-token and next-token information readable at each layer (baselines: App. C.3).

3 How the Separated Streams Divide Retrieval

Once the streams have separate weights, they divide retrieval. In all four separated models, state heads carry most of the previous-token attention (peak mass 0.71–0.89 in state versus 0.17–0.39 in prediction), while prediction heads show higher mean induction lift (Tab. 2). State records each position's predecessor and prediction matches against that record, so the state prediction reads determines what it can match. The division is not strict: in Two-tower 12+12, state heads reach a mean lift of 3.02 against prediction's 4.38, and about as many state heads as prediction heads exceed lift 3 (Tab. 6, App. B.1). SPS instead reaches the previous token through prediction-slot keys (mass 0.80 versus 0.26 on state-slot keys), which already carry it.

Table 2: Seed-mean peak previous-token mass, mean induction lift, and ΔNLL (nats/token) from ablating three previous-token state heads versus low-mass control heads. S/P: state/prediction query stream (the Transformer's single stream under S). SPS's "state keys" and "pred. keys" rows restrict each measure to state-slot or prediction-slot keys. Holm columns give one code per seed for random and pattern-matched control pools: all = Holm-significant both with and without layer 1, excl. = only with layer 1 excluded, – = neither.

Model	Prev. mass	Mean lift	ΔNLL, abl./ctrl.		Holm-specific, s1/s2	
	S/P	S/P	induct. tok.	all tok.	random	pattern
Transformer	0.75/–	3.68/–	0.300/0.018	0.096/0.025	–/all	all/all
Two-tower 12+12	0.85/0.37	3.02/4.38	0.095/0.015	0.032/0.015	–/all	all/all
Sequential 12+12	0.71/0.17	2.12/4.73	0.137/0.010	0.046/0.009	all/all	all/all
Two-tower 6+6	0.89/0.39	3.51/4.67	0.151/0.020	0.058/0.014	excl./–	excl./excl.
Sequential 6+6	0.82/0.21	3.17/5.17	0.202/0.007	0.108/0.018	all/–	excl./excl.
SPS, state keys	0.26/0.47	2.25/4.87	0.120/0.007	0.035/0.003	–/excl.	excl./excl.
pred. keys	0.80/0.33	4.06/4.49				

Removing state's previous-token heads selectively hurts prediction's retrieval. In the four separated models, ablating the three strongest previous-token state heads raises induction-token loss 1.9–3.0× as much as all-token loss, against 0.4–1.4× for low-mass control heads. Against random and pattern-matched controls, the effect passes every Holm-corrected test in Sequential 12+12 and some tests in the other separated models, partly through layer-1 exclusions added after seeing the results (App. B.2). Prediction's retrieval depends on predecessor information from state heads, so what prediction reads from state matters.

Patching supports this. Across seeds of the separated models, restoring the strongest previous-token state head recovers 0.09–0.50 of the corruption-induced change in answer-versus-distractor logits, while random donor heads stay within ± 0.10 (Tab. 8). Recovery is small in SPS and near zero in the shared-weight models, so the effect is clear only once the streams are separated. Restoring previous-token key slots alone recovers less than restoring all heads, so the key path carries part of the effect but not all of it. Path tracing, which attributes query–key (QK) scores to paths from state heads into matching, shows the score spread over several paths: the largest single path carries

30–67% of the matching-minus-control score (App. B.3). Matching draws on state that combines many heads, consistent with reading processed state.

3.1 Predecessor Information Reaches the Top State Layer

Previous-token probes show that this predecessor information survives to the top of state. In SPS and the 12+12 separated models, state's previous-token probe loss stays low through the final layer, whereas the Transformer's rises near the output (Fig. 2b: the colored curves stay low while the black curve climbs). The 6+6 separated models agree (App. C.3). State does not turn into a next-token predictor along the way: its last-layer next-token probes (4.8–5.4 nats/token) stay near the bigram baseline (5.24), far above prediction's 3.5–3.7. Deep state still holds what retrieval needs, so reading the final state loses none of it. Which level prediction actually reads is tested next.

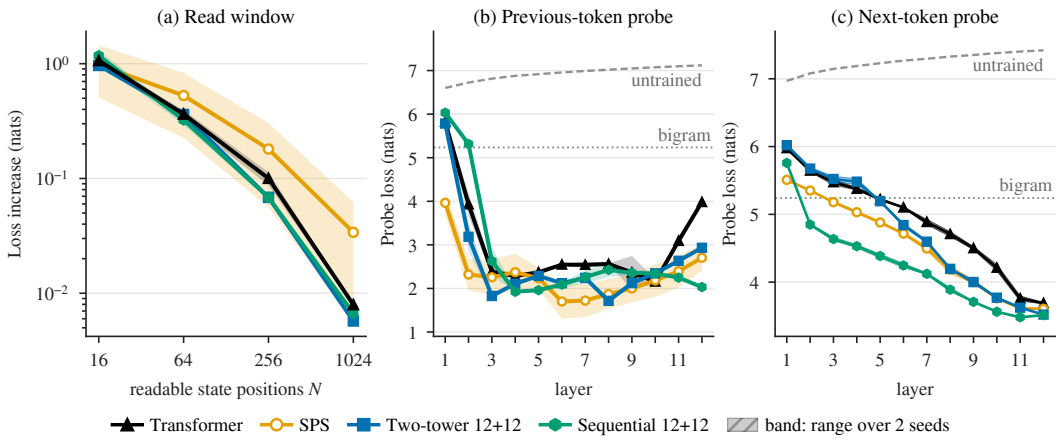


Figure 2: SPS, Two-tower 12+12, Sequential 12+12, and Transformer (black). (a) Loss increase over each model's own unrestricted full-read loss (nats/token, log scale; zero means no cost). In SPS and the separated models only prediction's reads of state are limited to the nearest N positions, whereas in the Transformer every layer's attention is restricted. (b) State-stream previous-token probe loss and (c) prediction-stream next-token probe loss by layer (nats/token); the Transformer uses its single stream in both. Dashed grey/dotted lines: untrained/bigram references. Curves: seed means; shading: two-seed range.

4 What State Prediction Needs

Section 3 showed that prediction's retrieval uses predecessor information persisting to the top of the state stream. We now ask which state positions and levels prediction actually needs, testing this by restricting what a trained model's prediction may read and by lesioning whole layers.

4.1 Prediction Needs Distant State

We first restrict prediction's reads to the nearest N state positions and measure the loss increase (Fig. 2a). Keeping only 64 positions raises loss by 0.32–0.40 nats/token in the four separated models (Tab. 10), so prediction relies on state far back in the context. SPS also pays a cost, although its size varies widely between seeds (App. C.6). Relying on distant context is not specific to separation, since restricting every Transformer layer the same way costs 0.36 nats/token (App. C.1). With separate streams, that distant context can only come from state, so prediction's read of state must span it.

4.2 Deep State Levels Matter More

To test which state levels prediction needs, we keep all positions and the trained K/V projections of Two-tower 12+12 but restrict prediction to a subset of state levels, pairing the m deepest with

the $m{+}1$ shallowest. Prediction is far more sensitive to losing its deep reads: keeping only the shallowest levels costs 2.5–5.4× as much as keeping only the deepest (Fig. 3a, where the dashed shallowest-level curve lies well above the solid deepest-level curve; details: App. C.2). Whole-layer lesions show why deep reads matter: Two-tower 12+12 relies on both streams, with state costs concentrated in the bottom layers and prediction costs rising toward the top (Fig. 3b; details and controls in App. C.4).

SPS, by contrast, makes progressively less use of its deeper state layers: its state-lesion costs fall roughly two-fold from early to late interior layers, and its top state layer costs nothing because nothing reads it (Fig. 3c). The top of SPS's state stream holds predecessor information that nothing downstream exploits.

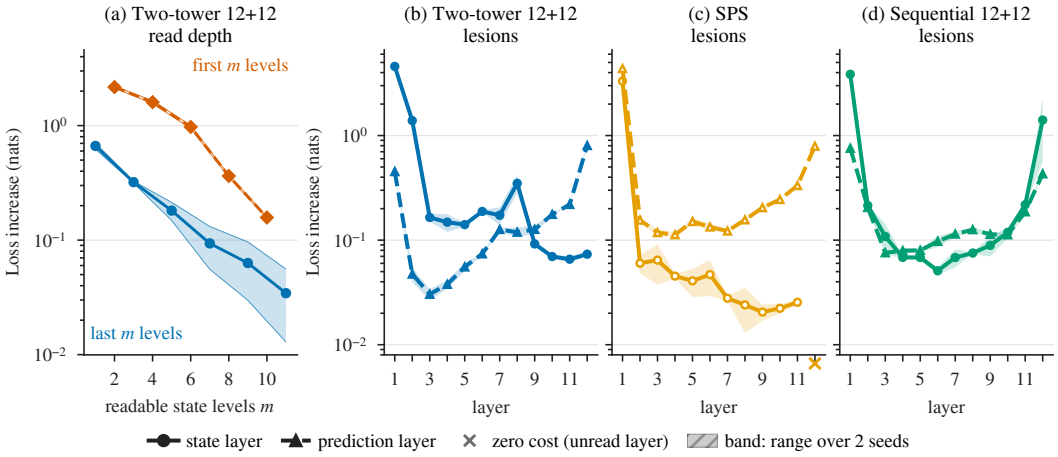


Figure 3: Loss increase (nats/token, log scale). (a) Two-tower 12+12: the sweep pairs the m deepest levels (blue circles, solid) with the $m{+}1$ shallowest (vermilion diamonds, dashed). (b–d) Whole-layer lesion costs by layer for Two-tower 12+12, SPS and Sequential 12+12, sharing one y-axis scale. Solid circles denote state and dashed triangles prediction. SPS's top state layer costs nothing because nothing reads it. Curves: seed means; shading: two-seed range.

5 Final-State Reads Lower Loss with Separate Weights

These results lead to one change. State writes predecessor information and prediction matches against it (§3), and ablation shows this read path matters and draws on many state heads. That information persists to state's top layer (§3.1), prediction needs it from distant positions (§4.1), and prediction is most sensitive to losing its deep reads (§4.2). Yet Two-tower gives the final state only to its last prediction layer, and SPS never reads it at all. Sequential gives it to every prediction layer (Fig. 1d). Testing this requires retraining, since switching a trained Two-tower 6+6 to final-state reads at evaluation raises its loss by 0.68 nats/token (App. C.2). We therefore train both models from scratch at matched parameters and FLOPs.

With separate weights, Sequential lowers validation loss in both seeds, by 0.031 nats/token at 12+12 and 0.037 at 6+6 (Tab. 3, per-seed losses in Tab. 5). That is about half of the 0.068 gain SPS achieves over the Transformer, and well above our 0.005 tie threshold. Capacity does not explain it: moving MLP width from state to prediction leaves Two-tower 12+12 at 2.737, far from Sequential 12+12's 2.702 (seed 1, App. A.2). Separate attention matters too, since sharing attention between the two towers raises Two-tower 12+12 to 2.767 (seed 1, App. D.3). Balanced towers work best: Sequential 6+6 beats 3+9 and 9+3 at equal cost, and a top-aligned Two-tower 12+6 with a shorter prediction tower falls short of Two-tower 12+12 (seed 1 each, App. D.5).

With shared weights, the ordering reverses: Shared Two-tower 12+12 (seed 1) beats Shared Sequential 12+12's seed mean by 0.012 nats/token. One hypothesis is looping: a shared stack runs twice, first for state and then for prediction, and a final-state read may make those two uses less compatible. Gradient diagnostics are consistent with this, with slight alignment in Shared Two-tower 12+12 and

unrelated directions in Shared Sequential 12+12 (App. C.5). The shared pair also differs in head tying and prediction input (App. D.3).

Table 3: Two-tower and Sequential validation loss (nats/token), difference $\Delta = \mathrm{Seq.} - \mathrm{TT}$ (Sequential minus Two-tower), non-embedding and total parameters (M), and GFLOPs/token. Heads and inputs as in Tab. 1. Shared parameter and FLOP pairs are Two-tower/Sequential (Shared Sequential 12+12 adds a 1.18M-parameter K/V projection). Shared Two-tower 12+12 reads level $\ell-1$, as in SPS. Sequential 6+6 ($\langle$predict$\rangle$ input) removes the prediction embedding table. $\pm$ as in §2.3, and loss entries without $\pm$ have one seed.

		Read		Δ	Params (M)		GFLOPs
Weights	Towers	Two-tower	Sequential	Seq.−TT	non-emb.	total	/token
Separate	12+12	$2.733_{\pm.001}$	$2.702_{\pm<.001}$	−0.031	184.1	300	0.751
	6+6	$2.838_{\pm<.001}$	$2.801_{\pm.002}$	−0.037	92.0	208	0.414
Two-tower 12+12 (MLP realloc.)		2.737	–	–	184.1	300	0.751
Sequential 6+6 ($\langle$predict$\rangle$ input)		–	2.805	–	92.0	169	0.414
Shared	12+12	2.814	$2.826_{\pm.001}$	+0.012	92.0/93.2	131/132	0.722/0.725
Transformer		$2.865_{\pm.003}$		–	92.0	169	0.414
SPS		$2.797_{\pm.001}$		–	92.0	169	0.756

5.1 Where the Gain Comes From

Sequential's gain over Two-tower is a reduction in validation NLL on held-out FineWeb-Edu. If it came from better copying, the mechanism of §3, it would concentrate on tokens copyable from context, but it does not. In the repeat-distance sweep (512 sequences), first-occurrence tokens (targets absent earlier in the document) are 42.8% of tokens but take 58–61% of the NLL reduction. In the induction sweep (16 sequences), induction tokens, which could be copied, are 19.3% of tokens but take only 8–14% (Tab. 4). Relative gains are about 1–2% at every distance to a token's last occurrence (Tab. 17), so the larger first-occurrence share mostly reflects their higher baseline loss.

Table 4: Token shares versus shares of Sequential's gain over Two-tower, for first-occurrence tokens (absent earlier in the document) and copyable induction tokens. A group's share of gain is its summed NLL reduction divided by the sweep's total reduction. R: repeat-distance sweep (512 sequences, 2.09M tokens). I: induction sweep (16 sequences, 65,520 tokens). Rows come from different sweeps and do not partition the tokens (App. D.4). Ranges span both depths (6+6 and 12+12) and, for I, the same-seed pairs.

Token group	Share of tokens	Share of Sequential's gain
First-occurrence tokens (R)	42.8%	58–61%
Induction tokens (I)	19.3%	8–14%

Two diagnostics show how Sequential uses its read. First, its prediction stream becomes predictive earlier, as expected if it works on processed state sooner: in Fig. 2c, Sequential 12+12's next-token probe reaches 5.0 and 4.5 nats/token three to four layers before Two-tower 12+12, and Sequential 6+6 leads by one layer (App. D.4). The probe curves of both models then converge at the final prediction layer.

Second, every layer in both of Sequential 12+12's towers matters. In Fig. 3d its state and prediction lesion profiles share a U shape, with costly first and last layers and a flat middle, and the two towers cost about the same at each depth (within about 1.6× on average), whereas in Two-tower and SPS one stream dominates (3–5× apart). Both towers are thus used about equally. Its top state layer, unread in SPS, is its costliest state layer after layer 1 (0.56–2.26 nats/token, persisting under mean ablation; App. C.4), consistent with reliance on the read.

Another variable that may have affected performance is processing depth: the final-state read gives Sequential more layers to work on fully contextualized state, letting it make better use of its layers.

6 Efficiency at the Transformer's Compute

If the final-state read makes better use of state, a smaller Sequential model should also be efficient, and we test this. Separate-weight Sequential 6+6 matches the Transformer's non-embedding parameters and FLOPs and finishes 0.005 nats/token above SPS (2.801 versus 2.797, with the Transformer at 2.865), with an approximate 95% interval of 0.0002–0.0094 for the gap (App. A.1). It recovers 93% of SPS's gain over the Transformer (88–97% across seed pairings, App. D.6) at 55% of SPS's training compute (Fig. 4b,c). Its prediction embedding table gives it 208M total parameters against 169M for SPS and the Transformer (Tab. 3), but that table is not what drives the result: replacing it with a single learned ⟨predict⟩ token matches total parameters as well and changes loss only from 2.803 to 2.805 in the same seed (App. D.5). It keeps the Transformer's K/V cache size and roughly its training throughput (App. D.7).

Before learning-rate decay the advantage is clearer. At the dotted matched-compute line in Fig. 4a, Sequential 6+6 lies 0.038 nats/token below SPS (SPS interpolated, App. D.6). At the end of training and matched total parameters, this ⟨predict⟩-input Sequential 6+6 is 0.008 above SPS and recovers 88% of its gain. With tied heads in all three models, Sequential 6+6 finishes 0.010 nats/token above tied-head SPS and recovers 72–82% of its gain over the tied-head Transformer (App. D.2). This tied-head Sequential 6+6 also beats AF-SPS (2.815/2.819 versus 2.826) with fewer GFLOPs/token (0.414 versus 0.595) and higher training throughput (1.02× versus 0.68× the tied-head Transformer, Tab. 21 and App. A.3).

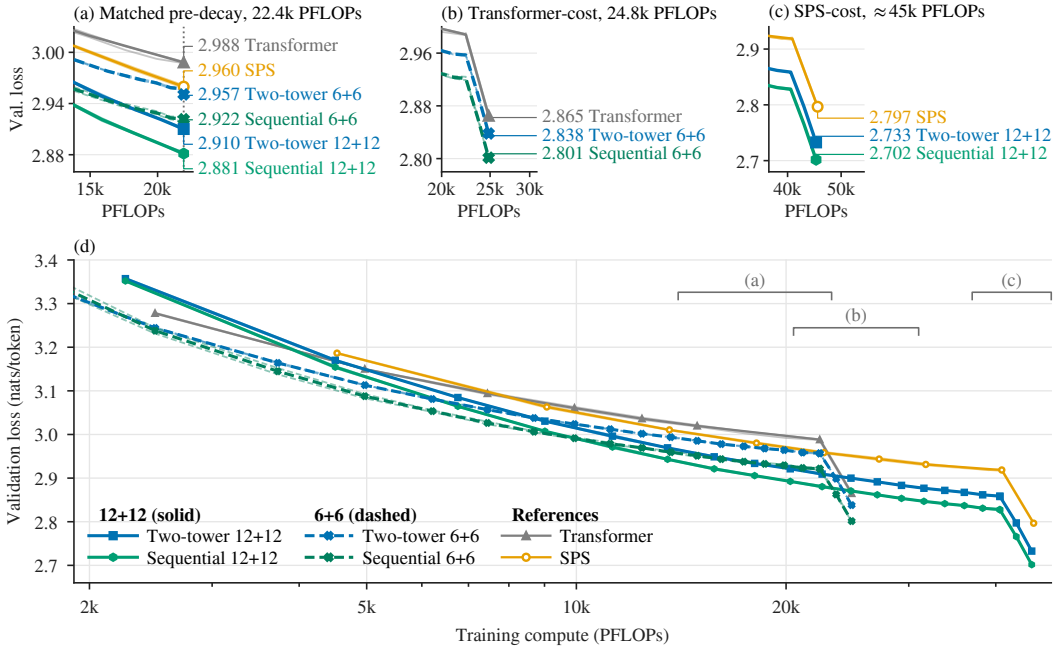


Figure 4: Validation loss (nats/token) versus cumulative training compute (PFLOPs, 10^{15} floating-point operations; log x-axis) for the Transformer, SPS and the four separated models. Bold lines are seed means and thin lines individual seeds. Top: (a) matched trajectories before learning-rate decay (pre-decay) through 22.4k PFLOPs (dotted line), with SPS and 12+12 interpolated in compute; (b) run ends at Transformer cost (24.8k PFLOPs; Transformer and both 6+6 models); (c) run ends at SPS cost (45.3k PFLOPs for SPS, 45.0k for both 12+12 models). End markers in (a–c) give loss values, and final seed ranges (≤ 0.006) are not drawn. Bottom: (d) full runs with a marker at each checkpoint, where grey brackets mark the ranges of (a)–(c).

7 Related Work

Extra computation for prediction. SPS (Monea et al., 2026) gives each token a separate prediction slot, and later work makes that computation cheaper: Almost-Free SPS (AF-SPS, Langford et al., 2026, reimplemented in App. A.3) shares an MLP between the streams, and Decode-Branch Transformers (DBT, Liu et al., 2026b) move prediction compute off the prefill path. Parallel and looped designs add per-token computation through shared weights (Wu et al., 2025b;a; Chen et al., 2025; Liu et al., 2026a). These methods vary how much computation prediction gets. We instead fix the cost, separate the weights, and vary which state prediction reads.

Final-state readers and encoder–decoders. Encoder–decoder models read the encoder's final output, and recent work revisits them for language modeling, comparing pretraining setups (Wang et al., 2022), targeting small models (Elfeki et al., 2025), adapting decoder-only models (Zhang et al., 2025), or decoding locally from block-level context (Ho et al., 2024). Machine translation also asked which encoder layers a decoder should read (Bapna et al., 2018; He et al., 2018; Domhan, 2018; Liu et al., 2021). These works change whole architectures or read a separate source sequence, whereas our state encodes the same sequence causally, prediction at t reads only positions $\leq t$ without self-attention, and read level is the only difference between the compared models (App. E).

Cross-layer K/V reuse. Layer-Condensed KV Cache (LCKV), You Only Cache Once (YOCO) and Cross-Layer Attention reuse keys and values across layers within one stream to save memory (Wu & Tu, 2024; Sun et al., 2024; Brandon et al., 2024). Sequential instead reads the final state from a separate stream with its own weights and input, and its read level comes from an analysis of what prediction needs rather than from cache savings.

Interpretability. Our analyses build on induction circuits (Olsson et al., 2022; Elhage et al., 2021), activation and path patching (Meng et al., 2022; Wang et al., 2023; Goldowsky-Dill et al., 2023; Zhang & Nanda, 2024; Heimersheim & Nanda, 2024), and probing (Alain & Bengio, 2017; Belinkov, 2022; Belrose et al., 2023). These methods were built for single-stream Transformers, where storage and prediction share each hidden state, and applying them to fully separated streams lets us attribute each computation to one role.

8 Conclusion

Which state should prediction read? Once the streams are separated, state writes predecessor information and keeps it through its top layer, while prediction does the matching using distant, deep state. These findings suggested giving every prediction layer the most processed state, which SPS never reads and Two-tower gives only to its last prediction layer. A controlled test confirmed the design: with separate weights, Sequential's final-state read lowers loss at matched parameters and FLOPs, and a smaller Sequential model approaches SPS at the Transformer's compute. Interpretability can thus guide architecture design, and where prediction reads state is a strong design choice for separated streams. Testing the final-state read at larger scale is a natural next step.

Limitations. Our models are small (130–300M parameters, width 768, 20B FineWeb-Edu tokens, context 4096), and we evaluate validation loss up to 4096 tokens. Extending the comparison to larger models and downstream tasks is left to future work.

AI use statement

In this work, we used generative AI tools (Claude and GPT models) to implement methods (training, evaluation and analysis code), create figures, run and monitor experiments, search for and summarize background literature, verify and format references, and draft and edit the paper's text. We did not use generative AI to generate synthetic data sets, formulate mathematical claims or proofs, translate, or clean datasets. The research question, direction and choice of experiments were driven by the authors, who reviewed all AI-assisted work: all reported results come from our own logged training runs, every number in the paper was re-derived from those logs and checked against the underlying data, and code was tested with automated reproducibility checks. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

Reproducibility statement

The appendices document training and evaluation (App. A.1), model definitions with parameter and FLOP counts (Apps. A.2 and A.3), every analysis method and its control gates (App. A.4), and known issues (App. A.5). Code, configurations and result files will be released as an anonymized archive in the supplementary material.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2017. URL https://arxiv.org/abs/1610.01644.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3028–3033, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1338. URL https://aclanthology.org/D18-1338/.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422. URL https://arxiv.org/abs/2102.12452.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2023. URL https://arxiv.org/abs/2303.08112.

William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan-Kelley. Reducing transformer key-value cache size with cross-layer attention. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, pp. 86927–86957, 2024. doi: 10.52202/079017-2758. URL https://arxiv.org/abs/2405.12981.

Mouxiang Chen, Binyuan Hui, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Jianling Sun, Junyang Lin, and Zhongxin Liu. Parallel scaling law for language models. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, pp. 131892–131932, 2025. doi: 10.52202/085713-3970. URL https://arxiv.org/abs/2505.10475.

Tobias Domhan. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1799–1808, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1167. URL https://aclanthology.org/P18-1167/.

Mohamed Elfeki, Rui Liu, and Chad Voegele. Return of the encoder: Maximizing parameter efficiency for SLMs, 2025. URL https://arxiv.org/abs/2501.16273.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Angela Fan, Thibaut Lavril, Edouard Grave, Armand Joulin, and Sainbayar Sukhbaatar. Addressing some limitations of transformers with feedback memory, 2020. URL https://arxiv.org/abs/2002.09402.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching, 2023. URL https://arxiv.org/abs/2304.05969.

Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Layer-wise coordination between encoder and decoder for neural machine translation. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/4fb8a7a22a82c80f2c26fe6c1e0dcbb3-Paper.pdf.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024. URL https://arxiv.org/abs/2404.15255.

Namgyu Ho, Sangmin Bae, Taehyeon Kim, Hyunjik Jo, Yireun Kim, Tal Schuster, Adam Fisch, James Thorne, and Se-Young Yun. Block transformer: Global-to-local language modeling for fast inference. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, pp. 48740–48783, 2024. doi: 10.52202/079017-1545. URL https://arxiv.org/abs/2406.02657.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2006.10369.

John Langford, Nathan Godey, Giovanni Monea, Yoav Artzi, Harry Dong, Ying Fan, Gustavo de Rosa, and Zheng Zhan. Almost free state prediction separation, 2026. URL https://arxiv.org/abs/2609.03807.

Aiwei Liu, Cheng Shi, Chuhan Wu, Ci Lei, Di Lu, Donald He, Fan Zhang, Fanhao Kong, Feifei Zhang, Guan Wang, Haicheng Wang, Haoyu Liu, Houjin Yu, Jiachen Ding, Jiayi Feng, Jie Zhou, Jijun Chi, Jindi Shi, Jing Lei, Junjie Zhang, Laiyi Li, Le Tian, Linhao Zhang, Miao Fan, Sijun Zhang, Wei Jia, Weiwei Shi, Wenhan Li, Wentao Zhao, Wenteng Liang, Xiao Zhou, Xiaojin Zhou, Xihuai Wang, Xinyu Gao, Xuanliang Wang, Xuyang Ao, Yang Yu, Yangxiu You, Yinuo Zhao, Yufei Kuang, Yufei Wang, Yuan Liu, Yuan Liu, Yuwen Chen, Zhencong Tian, Zhongyin Zhao, Zilin Yu, and Zitao Wang. Hidden decoding at scale: Latent computation scaling for large language models, 2026a. URL https://arxiv.org/abs/2607.08186.

Liming Liu, Mingze Wang, and Tuo Zhao. Decode-branch transformers: Decoupling the primary prefill path from additional decode computation, 2026b. URL https://arxiv.org/abs/2608.12385.

Xuebo Liu, Longyue Wang, Derek F. Wong, Liang Ding, Lidia S. Chao, and Zhaopeng Tu. Understanding and improving encoder layer fusion in sequence-to-sequence learning. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2012.14768.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1711.05101.

Thomas McGrath, Matthew Rahtz, János Kramár, Vladimir Mikulik, and Shane Legg. The Hydra effect: Emergent self-repair in language model computations, 2023. URL https://arxiv.org/abs/2307.15771.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pp. 17359–17372, 2022. doi: 10.52202/068431-1262. URL https://arxiv.org/abs/2202.05262.

Giovanni Monea, Nathan Godey, Kianté Brantley, and Yoav Artzi. The state-prediction separation hypothesis, 2026. URL https://arxiv.org/abs/2607.01218.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane

Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024 Datasets and Benchmarks Track)*, pp. 30811–30849, 2024. doi: 10.52202/079017-0970. URL https://arxiv.org/abs/2406.17557.

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

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://arxiv.org/abs/2104.09864.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. You only cache once: Decoder-decoder architectures for language models. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, pp. 7339–7361, 2024. doi: 10.52202/079017-0235. URL https://arxiv.org/abs/2405.05254.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2211.00593.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective works best for zero-shot generalization? In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22964–22984. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wang22u.html.

Bohong Wu, Mengzhao Chen, Xiang Luo, Shen Yan, Qifan Yu, Fan Xia, Tianqi Zhang, Hongrui Zhan, Zheng Zhong, Xun Zhou, Siyuan Qiao, and Xingyan Bin. Parallel loop transformer for efficient test-time computation scaling, 2025a. URL https://arxiv.org/abs/2510.24824.

Bohong Wu, Shen Yan, Sijun Zhang, Jianqiao Lu, Yutao Zeng, Ya Wang, and Xun Zhou. Efficient pretraining length scaling, 2025b. URL https://arxiv.org/abs/2504.14992.

Haoyi Wu and Kewei Tu. Layer-condensed KV cache for efficient inference of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11175–11188, 2024. doi: 10.18653/v1/2024.acl-long.602. URL https://arxiv.org/abs/2405.10637.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019. URL https://arxiv.org/abs/1906.08237.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pp. 5824–5836, 2020. URL https://arxiv.org/abs/2001.06782.

Biao Zhang, Fedor Moiseev, Joshua Ainslie, Paul Suganthan, Min Ma, Surya Bhupatiraju, Fede Lebron, Orhan Firat, Armand Joulin, and Zhe Dong. Encoder-decoder Gemma: Improving the quality-efficiency trade-off via adaptation, 2025. URL https://arxiv.org/abs/2504.06225.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.16042.

App. A gives the training setup, model definitions and analysis methods behind §2. App. B adds the controls and patching behind the stream roles in §3, and App. C the read restrictions, probes, lesions and seed checks behind §4.1–§5.1. App. D covers the architecture controls and efficiency measurements for §5–§6, and App. E extends §7.

A Setup and Reproducibility

A.1 Training and Evaluation Details

Every run in Tab. 5 shares the setup below, which completes §2.3. The Transformer has a 2304-unit swish-gated linear unit (SwiGLU) MLP, rotary position embeddings (RoPE, Su et al., 2024), root-mean-square normalization (RMSNorm) and 38.6M input-embedding parameters. Data come from FineWeb-Edu 100BT (Penedo et al., 2024), tokenized with GPT-2 (Radford et al., 2019).

Training uses Adam with decoupled weight decay (AdamW, Loshchilov & Hutter, 2019) for 20B tokens:

- Learning rate: a 50M-token linear warmup, a plateau at 6×10^{-4}, then linear decay over the final 2B tokens (10%) to 6×10^{-5}.
- Optimizer: betas $(0.9, 0.95)$, weight decay 0.1 and gradient clipping at 1.0.
- Batch: 96×4096 tokens on 2×4 or 8 H100s, with micro-batch 6.

Seeds 1 and 2 are random seeds 1337 and 1338, and single-seed runs use seed 1. Tied-head SPS instead uses seeds 2 and 3 (1338 and 1339, App. A.5). The chunk-shuffle seed equals the model seed, so same-seed runs see the same data order.

Validation uses bfloat16 (bf16) on non-overlapping 4096-token sequences, with 13.1M scored tokens in the full sweep and 512 sequences (2,094,978 tokens) in the paired sub-sweeps used for interventions. On these sweeps, no-op controls change loss by at most 0.0001 nats/token, and we call an intervention magnitude replicated when both seeds agree in sign and differ by at most 1.5×. Forward FLOPs are computed analytically from each instantiated configuration.

The 0.005 tie threshold rests on the pooled seed standard deviation (SD), estimated from eight two-seed arms and three tied-head SPS runs (including seed 1337, which is excluded from reported means). With 10 degrees of freedom and a chi-square 95% confidence interval (CI) of [0.0014, 0.0036] under common variance, the threshold is about 2.4 pooled SDs. Variance is uneven across arms: the untied Transformer contributes about 36% of the sum of squares. Tables round seed ranges, while differences in the text use unrounded losses.

For the efficiency comparison in §6, independent arm means give Sequential 6+6 minus SPS an approximate 95% Student-t interval (t_{10}) of [0.0002, 0.0094] nats/token around $+0.0048$, and the gain-recovery delta-method interval is approximately ± 6.5 percentage points. Both ignore same-seed covariance. The loss interval extends past the 0.005 tie threshold, so the gap is small but not shown to be a tie.

A.2 Model Definitions and Full Results

This section defines the separate-weight models precisely and the controls built from them, and Tab. 5 lists every run. Let a, b be state and prediction depths, x_t the token, and $s_t^0 = E_s(x_t)$, $p_t^0 = E_p(x_t)$ its separate state and prediction embeddings. State is a stack of pre-norm causal blocks, and the read map is $r(\ell) = \ell$ for Two-tower and $r(\ell) = a$ for Sequential. With learned projections W_Q, W_{KV}, W_O, scaled dot-product attention Attn and normalization inputs P, R, M (prediction, read, MLP), prediction updates are

$$q_t^\ell = W_Q^\ell \,\mathrm{RMSNorm}_P^\ell(p_t^{\ell-1}), \qquad (k_j^\ell, v_j^\ell) = W_{KV}^\ell \,\mathrm{RMSNorm}_R^\ell(s_j^{r(\ell)}),$$
$$\widetilde{p}_t^\ell = p_t^{\ell-1} + W_O^\ell \,\mathrm{Attn}(q_t^\ell, k_{j\le t}^\ell, v_{j\le t}^\ell), \qquad p_t^\ell = \widetilde{p}_t^\ell + \mathrm{MLP}^\ell(\mathrm{RMSNorm}_M^\ell(\widetilde{p}_t^\ell)).$$

Queries and keys receive rotary position embeddings, and the attention mask excludes other documents. Each prediction layer has its own read normalization and K/V projection, and nothing from prediction is written back to state. A final normalization and LM head predict x_{t+1} from p_t^b.

The controls change one aspect of these models at a time (seeds and head settings in Tab. 5):

- Two-tower 12+12 (MLP realloc.) uses state/prediction MLP widths 1152/3456 at unchanged cost, to test whether moving capacity to prediction explains the gain (§5).
- Sequential's ⟨predict⟩ input replaces its prediction embedding table with a learned ⟨predict⟩ token, removing the extra embedding parameters. Tying also reuses the state embedding as the LM head (Apps. D.5, D.2).
- Two-tower 12+6 (top-aligned) reads levels 7–12, testing whether deep state suffices for a shorter prediction tower. Sequential 3+9 and 9+3 vary the depth split, retaining separate weights, their own prediction embedding tables and untied heads (App. D.5).
- Tied-head Transformer/SPS reuse their input embeddings as LM heads. Shared Two-tower/Sequential 12+12 reuse one stack, with a learned ⟨predict⟩ token and tied head (App. D.3). AF-SPS also shares a gated MLP (App. A.3).

Table 5 lists 28 of the 49 runs trained to 20B tokens. The full evaluation ledger holds 61 distinct runs, including 12 shorter ones, and excludes unlogged pilots.

Table 5: Model and seed runs: head and input variant, shared weights, the state level read by prediction layer ℓ ($\ell-1$: the state entering layer ℓ), total and non-embedding parameters (M), GFLOPs/token, training speed and validation NLL (nats/token). Speed is relative to the tied-head Transformer with the same attention kernel, from the benchmark behind Tab. 21 (–: not measured). * marks the extra tied-head SPS seed-1 (1337) run, outside the two-seed mean.

Model	Variant	Shared	Read	Params (M)		GFLOPs	Speed	NLL
				total	non-emb.	/token		(nats/token)
Transformer (s1)	untied	–	–	169.3	92.0	0.414	1.00	2.862
Transformer (s2)								2.868
Transformer	tied	–	–	130.7	92.0	0.414	1.00	2.851
AF-SPS	tied / ⟨predict⟩	attn., pooled MLP, norms	$\ell-1$	130.7	92.1	0.595	0.68	2.826
SPS (s1)	untied / ⟨predict⟩	one stack	$\ell-1$	169.3	92.0	0.756	0.45	2.798
SPS (s2)								2.795
SPS (s1)	tied / ⟨predict⟩	one stack	$\ell-1$	130.7	92.0	0.756	0.45	2.805*
SPS (s2)								2.806
SPS (s3)								2.809
Two-tower 6+6 (s1)	untied / own table	none	ℓ	207.9	92.0	0.414	–	2.838
Two-tower 6+6 (s2)								2.838
Two-tower 12+6 (top-aligned)	untied / own table	none	$\ell+6$	251.3	135.4	0.577	0.74	2.757
Two-tower 12+12 (shared attn. only)	untied / own table	attn.	ℓ	271.7	155.8	0.751	0.57	2.767
Two-tower 12+12 (s1)	untied / own table	none	ℓ	300.0	184.1	0.751	0.57	2.732
Two-tower 12+12 (s2)								2.733
Two-tower 12+12 (MLP realloc.)	untied / own table	none	ℓ	300.0	184.1	0.751	–	2.737
Sequential 3+9	untied / own table	none	final	207.9	92.0	0.414	1.02	2.813
Sequential 6+6 (s1)	untied / own table	none	final	207.9	92.0	0.414	1.02	2.803
Sequential 6+6 (s2)								2.800
Sequential 6+6	untied / ⟨predict⟩	none	final	169.3	92.0	0.414	1.02	2.805
Sequential 6+6 (s1)	tied / ⟨predict⟩	none	final	130.7	92.0	0.414	1.02	2.815
Sequential 6+6 (s2)								2.819
Sequential 9+3	untied / own table	none	final	207.9	92.0	0.414	1.02	2.815
Sequential 12+12 (s1)	untied / own table	none	final	300.0	184.1	0.751	0.57	2.702
Sequential 12+12 (s2)								2.701
Shared Two-tower 12+12	tied / ⟨predict⟩	attn., MLP, norms	$\ell-1$	130.7	92.0	0.722	–	2.814
Shared Sequential 12+12 (s1)	tied / ⟨predict⟩	attn., MLP, norms	final	131.8	93.2	0.725	–	2.827
Shared Sequential 12+12 (s2)								2.826

A.3 Almost-Free SPS Reimplementation

AF-SPS (Langford et al., 2026) lowers the cost of separation by sharing the MLP between streams. We reimplement it as an efficiency reference and compare it (tied head, seed 1) with the tied-head Transformer (seed 1) and tied-head SPS (seeds 2/3). Prediction uses a learned ⟨predict⟩ token without prediction-to-prediction attention, and attention and norms are shared. For normalized residuals $\bar{s}, \bar{p}$, the mixing gate is $g = \sigma(W_{\text{in}}[\bar{s}; \bar{p}])$ and the shared output is $f = \text{MLP}(g\bar{s} + (1-g)\bar{p})$, with per-stream output gates. The head is tied, prediction reads the state entering each layer as in SPS, and only prediction supplies the loss. Because state depends on prediction through the shared MLP, a final-state read is not possible without changing the architecture.

AF-SPS trades some of SPS's loss for lower compute (Tab. 5). This comes from our reimplementation, which differs from the original in scale (131M/20B versus 1B/100B), multi-head versus grouped-query attention, AdamW, dense attention and context (4096 versus 8192), and omits tail-phasing and the two-pass attention kernel.

The shared MLP changes more than cost: unlike the separated models, AF-SPS has no head with induction lift above 3 and no induction-selective ablation (Tab. 6), and its gradient alignment grows during training (App. C.5). Per-stream MLP lesions cost nothing because the MLP is shared.

A.4 Analysis Methods and Control Gates

This section details the methods of §2.4, and each results subsection (Apps. B–D) states its models and seeds.

Circuit scores and ablation targets. Induction lift measures where matching happens. On natural text, it compares attention to matching keys with uniform attention over the causally visible keys, summing matches in the scored key stream and counting only queries that have an allowed match. Stream means average all heads, including those below the threshold of 3. Ablation effects are scored on induction tokens, targets x_{t+1} with an earlier $x_j = x_{t+1}$ and $x_{j-1} = x_t$, where $j \leq t-64$ excludes SPS's prediction-slot window.

Head-ablation controls. The controls ask whether ablating any three comparable heads would cost as much as the three previous-token state heads. Each pool draws three distinct non-target heads 100 times. Random draws sample any such heads, while pattern-matched draws pick among the five nearest unused heads in standardized entropy, log attended distance, first-token attention mass and output norm, excluding heads with high previous-token mass, so the same triple can recur. Specificity is the loss of top-1 accuracy on induction tokens the intact model predicts correctly, tested with $p = (k+1)/101$, where k counts control draws at least as costly as the target. Holm correction at $\alpha = 0.05$ covers the four tests per model and seed (App. B.2). The all-layer pools, metric and p-value were fixed before measurement, and the layer-1 exclusions were added afterward. Neither pool matches the target's all-token cost, so the tests measure selectivity for induction tokens, not equal overall damage.

Patching and path tracing. Patching asks whether restoring one clean state-head output recovers retrieval after corruption. It uses duplicated 256-token segments and scores mean answer-versus-distractor logits over 96 examples (App. B.3), with monotone trends across models specified beforehand. Path tracing asks which routes carry the matching. It decomposes matching-minus-control query–key (QK) scores by source and requires residual reconstruction relative error $\leq$0.01 and QK reconstruction absolute error $\leq$1.0.

Read restrictions. Read restrictions test which state prediction needs. Read reach keeps the nearest 16, 64, 256 or 1024 state positions and masks or mean-replaces farther keys, with controls for native NLL and untrained initialization. The read-depth tests were specified for Two-tower 12+12 before measurement, and as a sanity check, the unrestricted read reproduces native loss exactly. App. C.2 defines the first/last m restrictions and their alignment confound.

Role probes. Probes measure which token information is linearly readable at each layer. Linear next-, current- and previous-token probes train on sequences 0–511 and evaluate on 512–1023 at every fourth position, using standardized features, the top 8192 tokens plus "other", and Adam for 12 epochs. Baselines are trained embeddings, untrained architectures, and unigram and bigram cross-entropies on the same positions and labels (bigram losses 5.241 and 5.236 nats/token for next and previous token). As a sanity check, final next-token probes must do worse than the LM head.

Layer lesions. Lesions show which layers each model depends on. They zero a whole layer's, attention's or MLP's residual contribution in one stream (in SPS, the corresponding slot rows). Mean ablation instead uses clean per-channel means from 256 disjoint held-out sequences, and the unlesioned control reproduces full-sweep loss to measurement precision.

Gradient alignment. Gradient alignment asks whether the two streams pull shared weights in compatible directions. We treat the state and prediction uses of each shared tensor as independent copies and check that $g_{\text{total}} = g_{\text{state}} + g_{\text{pred}}$, with an exact per-slot split for SPS. Controls include three estimators, same-stream agreement across split batches, initialization and untied tensors.

A.5 Sampler Behavior and Checkpoint Availability

Three bookkeeping issues touch the runs in Tab. 5, and none changes the main Two-tower–Sequential comparisons. The chunked sampler repeats about 4.28% of one of eight ranks' streams, or 0.5% of all tokens, identically in every 20B run. On resume, all ranks adopt rank 0's offset, so ranks 1–7 replay or skip at most 0.04% of chunks. Two runs resumed: Sequential 9+3 (seed 1) at 7.52B tokens, after its gap to Sequential 6+6 had already appeared, and tied-head Sequential 6+6 ($\langle$predict$\rangle$ input, seed 2) at 0.40B tokens.

The tied-head SPS seed-1 checkpoint was overwritten before upload. Its logged loss stays in Tab. 5 but is excluded from the tied-head mean, which uses seeds 2/3. The main SPS analyses use the untied model and are unaffected.

B Evidence for State and Prediction Roles

B.1 Circuit Statistics and Head Ablation

To test whether the roles of §3 generalize, Tab. 6 extends Tab. 2 to every model and seed, including tied-head variants, AF-SPS and other depth splits. The roles mostly hold, as a division of work rather than exclusive head types. In Sequential 9+3 and top-aligned Two-tower 12+6, state has more heads with lift above 3 than prediction, Two-tower 12+12's state has nearly as many, and AF-SPS reverses the previous-token ordering.

Table 6: By model and seed: previous-token mass, induction lift, heads with lift > 3 (out of 12 per layer in the tower), and ablation/control ΔNLL (nats/token) for induction and all tokens. S/P mark state and prediction queries, and SPS rows distinguish key slots. Uncorrected p columns compare random/pattern-matched controls with and without layer 1. – means not run.

Model	Prev. mass	Mean lift	Lift > 3	ΔNLL, abl./ctrl.		p, rand./match.	
	S/P	S/P	S/P	induct. tok.	all tok.	all layers	excl. L1
Transformer, tied head	0.75/–	4.03/–	30/–	0.334/0.061	0.107/0.043	0.010/0.010	0.010/0.010
Transformer (s1)	0.74/–	3.74/–	31/–	0.276/0.026	0.087/0.042	0.030/0.010	0.030/0.010
Transformer (s2)	0.75/–	3.61/–	34/–	0.324/0.010	0.105/0.009	0.010/0.010	0.010/0.010
SPS, tied head (s2), state keys	0.31/0.18	2.03/4.62	11/37	0.099/0.000	0.030/0.003	0.079/0.198	0.030/0.198
pred. keys	0.94/0.26	5.12/5.25	107/110				
SPS, tied head (s3), state keys	0.67/0.15	2.45/5.56	17/40	0.297/0.001	0.096/0.002	0.010/0.010	0.010/0.010
pred. keys	0.77/0.27	5.11/5.16	97/108				
SPS (s1), state keys	0.34/0.43	2.12/4.28	15/35	0.076/0.010	0.019/0.005	0.099/0.287	0.030/0.010
pred. keys	0.64/0.36	3.91/4.44	88/98				
SPS (s2), state keys	0.17/0.50	2.39/5.45	10/37	0.165/0.003	0.051/0.002	0.040/0.257	0.010/0.010
pred. keys	0.95/0.31	4.20/4.54	92/95				
AF-SPS	0.25/0.49	1.11/1.12	0/0	0.022/0.002	0.063/0.005	–	–
Two-tower 12+12 (s1)	0.91/0.38	2.96/4.38	30/34	0.072/0.029	0.026/0.017	0.050/0.010	0.050/0.010
Two-tower 12+12 (s2)	0.80/0.37	3.07/4.38	29/30	0.118/0.000	0.038/0.013	0.010/0.010	0.010/0.010
Two-tower 12+12 (shared attn. only)	0.77/0.35	2.45/3.30	22/19	0.303/0.048	0.120/0.022	–	–
Two-tower 6+6 (s1)	0.79/0.41	3.49/4.58	11/21	0.202/0.021	0.066/0.017	0.198/0.653	0.010/0.010
Two-tower 6+6 (s2)	1.00/0.37	3.53/4.76	17/20	0.101/0.019	0.049/0.012	0.248/0.792	0.069/0.010
Two-tower 12+6 (top-aligned)	0.89/0.30	3.01/5.85	30/23	0.082/0.008	0.042/0.006	–	–
Sequential 12+12 (s1)	0.71/0.20	2.06/4.69	12/40	0.150/0.009	0.055/0.007	0.040/0.010	0.020/0.010
Sequential 12+12 (s2)	0.71/0.14	2.17/4.77	11/44	0.124/0.012	0.037/0.010	0.010/0.010	0.010/0.010
Shared Sequential 12+12 (s1)	0.91/0.17	2.04/4.97	10/45	0.245/0.016	0.092/0.016	–	–
Shared Sequential 12+12 (s2)	0.84/0.21	1.99/5.43	12/49	0.294/0.095	0.105/0.047	–	–
Sequential 3+9	0.66/0.18	2.22/4.95	4/33	0.342/0.025	0.142/0.036	–	–
Sequential 6+6 (s1)	0.76/0.21	3.20/5.25	15/23	0.273/0.015	0.130/0.018	0.010/0.436	0.010/0.010
Sequential 6+6 (s2)	0.87/0.21	3.13/5.10	10/28	0.131/−0.001	0.086/0.018	0.079/0.386	0.079/0.010
Sequential 6+6 ($\langle$predict$\rangle$ input)	0.75/0.27	2.90/5.45	12/26	0.257/0.029	0.141/0.015	–	–
Sequential 6+6 (tied head, $\langle$predict$\rangle$ input) (s1)	0.80/0.32	3.58/4.74	11/22	0.231/0.032	0.112/0.020	–	–
Sequential 6+6 (tied head, $\langle$predict$\rangle$ input) (s2)	0.81/0.30	3.54/5.03	12/24	0.187/0.025	0.066/0.030	–	–
Sequential 9+3	0.81/0.16	3.27/5.98	24/13	0.251/0.003	0.077/0.012	–	–
Shared Two-tower 12+12	0.75/0.22	2.16/4.07	13/29	0.180/0.027	0.061/0.032	0.059/0.010	0.010/0.010

Synthetic repeated random tokens remove natural-text structure, so they check that matching is not a text artifact. On them, prediction still holds the largest lifts in the four separated models, with more seed variation (Tab. 7).

Table 7: Induction lift on synthetic repeated random tokens. Ranges span seeds 1/2 of the separated models.

Model	Max. prediction	Max. state	Mean prediction
Two-tower 12+12	447–744	3.0–26.1	7.0–8.2
Sequential 12+12	153–824	1.1–1.5	2.7–7.1
Two-tower 6+6	260–812	1.9–7.8	4.1–17.3
Sequential 6+6	122–683	0.7–1.5	3.2–12.5

B.2 Head-Ablation Controls

These controls support §3's claim that removing state's previous-token heads selectively weakens prediction's retrieval. For the main models, each target's cost is compared with random and pattern-matched control draws (App. A.4), and the distributions explain the layer-1 exclusions. Figure 5 shows why layer 1 matters: in SPS and the 6+6 separated models the costly controls are layer-1 heads, and excluding them (lower strips) drops Two-tower 6+6's pattern-matched k from 65 to 0. Per-seed Holm results are in Tab. 2. One caveat applies to that table's ΔNLL controls: Two-tower 6+6's low-mass control contains only two distinct heads per seed, whereas random and pattern-matched triples always use three.

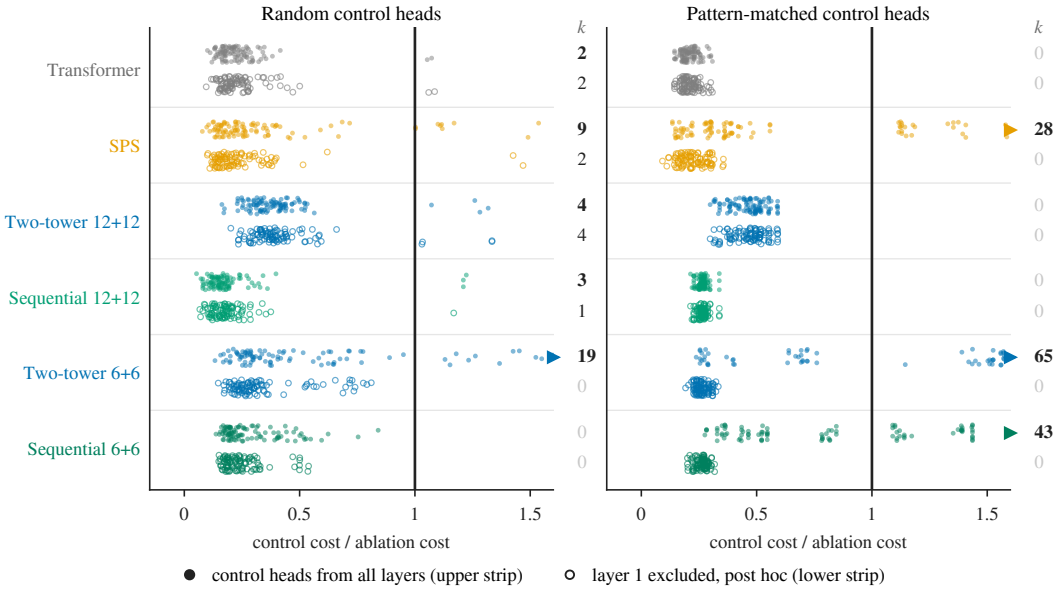


Figure 5: Seed-1 random and pattern-matched control costs divided by the three-head target-ablation cost in induction-token accuracy, for each main model (100 draws per strip). Dots are draws, triangles collect ratios above 1.6, the solid line marks 1, and k counts controls at least as costly as the target. Filled upper strips draw controls from all layers, and open lower strips exclude layer 1 (post hoc).

B.3 Activation Patching and Path Tracing

Patching supports §3's claim that prediction's retrieval draws on predecessor information from state heads, asking whether a single state head carries what retrieval needs (Tab. 8). The corruption swaps two unique tokens with different successors in the first copy of a duplicated segment. This moves the matching predecessor from the answer to a distractor while keeping the later query and the token inventory fixed. We then restore the strongest previous-token head at the answer slot, with a donor from another example as the random control. Recovery is the ratio of mean restored to mean clean-minus-corrupted logit differences.

Table 8: Answer-versus-distractor logit-difference recovery (fraction of the clean–corrupted difference) from patching each model's strongest previous-token head at the answer slot, or a random donor, on validation text with a duplicated 256-token segment. Two-seed rows give seeds 1/2 (tied-head SPS: seeds 2/3), and one-seed rows use seed 1 and add within-run 95% bootstrap intervals in brackets.

Model	Head patched in	Random donor
Transformer	0.50 / 0.26	−0.04 / −0.01
Transformer, tied head	0.48 [0.14, 0.90]	−0.01 [−0.04, 0.01]
AF-SPS	0.00 [−0.01, 0.01]	0.00 [−0.03, 0.00]
SPS	0.01 / 0.16	0.00 / −0.01
SPS, tied head	0.10 / 0.02	0.00 / 0.01
Two-tower 6+6	0.23 / 0.09	−0.02 / −0.10
Two-tower 12+6 (top-aligned)	0.10 [0.03, 0.21]	0.02 [−0.01, 0.04]
Two-tower 12+12 (shared attn. only)	0.04 [0.00, 0.14]	0.02 [0.00, 0.06]
Two-tower 12+12	0.10 / 0.37	0.00 / −0.02
Sequential 3+9	0.29 [0.15, 0.61]	−0.02 [−0.04, 0.03]
Sequential 6+6	0.50 / 0.23	0.00 / −0.04
Sequential 6+6 (⟨predict⟩ input)	0.73 [0.55, 0.83]	0.00 [−0.13, 0.04]
Sequential 6+6 (tied head, ⟨predict⟩ input)	0.19 / 0.08	−0.02 / 0.00
Sequential 9+3	0.38 [0.10, 0.86]	−0.01 [−0.03, 0.00]
Sequential 12+12	0.35 / 0.34	−0.05 / −0.08
Shared Two-tower 12+12	0.00 [−0.01, 0.01]	−0.01 [−0.01, 0.00]
Shared Sequential 12+12	0.02 / 0.00	0.02 / 0.00

One head restores part of the signal in the separated models, while AF-SPS and the shared-weight models recover little, which suggests the effect depends on separated streams. None of the pre-specified monotone trends across models held, so patching supports the role split but not a graded ordering of models.

Path tracing is observational: it attributes the matching-minus-control QK score to its sources (Tab. 9), with the 80% head count taken over the state-head QK score and the other shares over total QK. Two-tower 12+12 (shared attn. only), Shared Two-tower 12+12, AF-SPS and Shared Sequential 12+12 fail the decomposition exactness gate and are excluded. In the remaining models, the signal spreads over several state heads, and attention sources carry most of it, except in Two-tower 6+6 seed 2, where the largest source is an MLP. Key-slot patching in Two-tower 12+12 is the causal counterpart: restoring all heads' key slots recovers far more than restoring only the three previous-token heads, so matching draws on many state heads, not just the previous-token ones.

Table 9: Observational QK shares (first six rows) and causal key-slot patch recovery (last two rows) in separated models. Model and seed coverage is given per row, and paired values are seeds 1/2. QK scores are matching-minus-control, with denominators defined in the text.

Model and seeds	Measurement	Value
Two-tower/Sequential, both depths, 1/2	State heads carrying 80% of state-head QK	3–12
Two-tower 12+12, 1/2	Largest QK contribution	32% / 62%
Two-tower 6+6; Sequential 12+12 and 6+6, 1/2	Largest QK contribution	30–67%
Two-tower 12+12, 1/2	Three previous-token heads, QK share	36% / 68%
Two-tower 12+12, 1/2	All attention sources, QK share	87% / 91%
Two-tower 6+6, 1/2	All attention sources, QK share	85% / 67%
Two-tower 12+12, 1	Previous-token key-slot patch recovery	25%
Two-tower 12+12, 1	All-head key-slot patch recovery	77%

C Read Access, Representation and Replication

C.1 Read Reach and Gains Beyond Repeated Tokens

To check that §4.1's claim that prediction needs distant state holds beyond the main models, we apply the same recent-position restriction to SPS, AF-SPS, Two-tower, Sequential and their variants (Tab. 10), and to the Transformer with untied and tied heads (Tab. 11). Main models use seeds 1/2, tied-head SPS seeds 2/3, and single-run controls seed 1.

Masking could hurt simply because it removes attention targets, so we also replace distant state with its mean. At $N{=}64$ this costs about as much in the separated models (0.34–0.63 versus 0.32–0.40 masked, Tab. 10 top), so the dependence is not a masking artifact.

Table 10: Loss increase (nats/token): recent-position restrictions (top) and first/last m state levels (middle/bottom). For 12-level state towers, rows $\leq k$ are the first $m = k$ levels and rows $\geq k$ the last $m = 13 - k$ levels. * marks models whose native read is not same-depth (Sequential $a{+}b$ reads level a, top-aligned Two-tower 12+6 reads $\ell{+}6$). Their restrictions start from $\ell \mapsto \ell$ and so also change alignment. –: unchanged read or beyond state depth. Omitted architectures: App. C.2. $\pm$ as in §2.3, and entries without it have one seed.

	masked, $N{=}$				mean-replaced, $N{=}$			
Model	16	64	256	1024	16	64	256	1024
AF-SPS	1.37	0.41	0.083	0.006	1.33	0.80	0.26	0.013
SPS	$0.98_{\pm.47}$	$0.53_{\pm.30}$	$0.18_{\pm.12}$	$0.034_{\pm.028}$	$1.63_{\pm.04}$	$0.91_{\pm.23}$	$0.36_{\pm.22}$	$0.073_{\pm.065}$
SPS, tied head	$1.32_{\pm.35}$	$0.82_{\pm.27}$	$0.33_{\pm.19}$	$0.078_{\pm.059}$	$1.35_{\pm.16}$	$1.02_{\pm.17}$	$0.58_{\pm.17}$	$0.15_{\pm.10}$
Two-tower 6+6	$1.04_{\pm.01}$	$0.40_{\pm.02}$	$0.091_{\pm.016}$	$0.009_{\pm.003}$	$0.71_{\pm.01}$	$0.34_{\pm.01}$	$0.11_{\pm<.01}$	$0.010_{\pm.002}$
Two-tower 12+6 (top-aligned)	1.07	0.37	0.069	0.006	0.79	0.40	0.11	0.009
Two-tower 12+12 (shared attn. only)	1.02	0.36	0.069	0.006	0.86	0.42	0.11	0.009
Two-tower 12+12	$0.97_{\pm<.01}$	$0.36_{\pm<.01}$	$0.068_{\pm<.001}$	$0.006_{\pm<.001}$	$0.78_{\pm<.01}$	$0.40_{\pm<.01}$	$0.12_{\pm<.01}$	$0.008_{\pm<.001}$
Sequential 3+9	1.28	0.36	0.070	0.006	1.10	0.51	0.12	0.009
Sequential 6+6	$1.24_{\pm.02}$	$0.37_{\pm.01}$	$0.070_{\pm.001}$	$0.007_{\pm<.001}$	$0.96_{\pm.02}$	$0.46_{\pm.01}$	$0.11_{\pm<.01}$	$0.010_{\pm<.001}$
Sequential 6+6 (⟨predict⟩ input)	0.84	0.30	0.067	0.006	0.78	0.34	0.093	0.008
Sequential 6+6 (tied head, ⟨predict⟩ input)	$0.96_{\pm.03}$	$0.33_{\pm<.01}$	$0.071_{\pm.001}$	$0.007_{\pm<.001}$	$0.81_{\pm.02}$	$0.37_{\pm.01}$	$0.10_{\pm.01}$	$0.009_{\pm<.001}$
Sequential 9+3	1.21	0.50	0.10	0.007	0.71	0.37	0.12	0.011
Sequential 12+12	$1.17_{\pm.13}$	$0.32_{\pm.03}$	$0.068_{\pm.002}$	$0.007_{\pm<.001}$	$1.29_{\pm.15}$	$0.63_{\pm.13}$	$0.14_{\pm.01}$	$0.011_{\pm<.001}$
Shared Two-tower 12+12	1.07	0.33	0.063	0.006	1.14	0.55	0.12	0.009
Shared Sequential 12+12	$0.95_{\pm.01}$	$0.32_{\pm.01}$	$0.067_{\pm.002}$	$0.006_{\pm<.001}$	$0.95_{\pm.01}$	$0.43_{\pm.01}$	$0.098_{\pm.004}$	$0.007_{\pm<.001}$

	levels $\leq k$, $k{=}$					
Model	2	4	6	8	10	12
Two-tower 6+6	$1.21_{\pm.01}$	$0.39_{\pm.04}$	–	–	–	–
Two-tower 12+6 (top-aligned)*	1.82	1.54	1.39	1.39	1.39	1.39
Two-tower 12+12 (shared attn. only)	2.04	1.48	1.06	0.54	0.16	–
Two-tower 12+12	$2.17_{\pm.05}$	$1.60_{\pm.05}$	$0.97_{\pm.03}$	$0.36_{\pm.01}$	$0.16_{\pm<.01}$	–
Sequential 3+9*	0.62	–	–	–	–	–
Sequential 6+6*	$1.47_{\pm<.01}$	$1.07_{\pm.05}$	$0.94_{\pm.06}$	–	–	–
Sequential 6+6 (⟨predict⟩ input)*	1.29	0.98	0.88	–	–	–
Sequential 6+6 (tied head, ⟨predict⟩ input)*	$1.51_{\pm.12}$	$1.03_{\pm.04}$	$0.90_{\pm.04}$	–	–	–
Sequential 9+3*	2.25	2.16	2.16	2.16	–	–
Sequential 12+12*	$4.07_{\pm.83}$	$2.82_{\pm.50}$	$2.66_{\pm.52}$	$2.58_{\pm.53}$	$2.56_{\pm.57}$	$2.44_{\pm.50}$

	levels $\geq k$, $k{=}$					
Model	2	4	6	8	10	12
Two-tower 6+6	$0.010_{\pm<.001}$	$0.10_{\pm.01}$	$0.68_{\pm.06}$	–	–	–
Two-tower 12+6 (top-aligned)*	1.34	1.08	0.66	0.43	0.20	0.31
Two-tower 12+12 (shared attn. only)	0.059	0.063	0.11	0.14	0.28	0.75
Two-tower 12+12	$0.034_{\pm.022}$	$0.063_{\pm.033}$	$0.094_{\pm.038}$	$0.18_{\pm.03}$	$0.32_{\pm.01}$	$0.66_{\pm.04}$
Sequential 6+6*	$0.86_{\pm.06}$	$0.44_{\pm.05}$	–	–	–	–
Sequential 6+6 (⟨predict⟩ input)*	0.79	0.38	–	–	–	–
Sequential 6+6 (tied head, ⟨predict⟩ input)*	$0.78_{\pm.05}$	$0.35_{\pm.02}$	–	–	–	–
Sequential 9+3*	2.03	1.33	0.72	0.31	–	–
Sequential 12+12*	$2.67_{\pm.48}$	$2.82_{\pm.74}$	$2.67_{\pm.73}$	$2.38_{\pm.77}$	$1.98_{\pm1.07}$	–

The Transformer, restricted at every layer on the same 512 sequences, loses about as much at $N{=}64$ under both head settings (Tab. 11), so needing long context is general. What the separated models add is that this context is held in state.

Table 11: Transformer loss increase (nats/token) when every layer is restricted to the nearest N positions. Untied values are means of seeds 1/2, except the mean-replacement pair, and tied values use seed 1.

Transformer head / intervention	N=16	N=64	N=256	N=1024
Untied / mask	1.063	0.364 ± 0.030	0.100	0.008
Tied / mask	1.176	0.342	0.089	0.006
Untied / mean replacement (seeds 1/2)	–	0.617 / 0.517	–	–
Tied / mean replacement	–	0.656	–	–

C.2 Read-Depth Tests and Their Scope

The read-depth test supports §4.2's claim that prediction is most sensitive to losing its deep reads. It restricts prediction, at evaluation, to the first or last m state levels while keeping the trained K/V projections and all positions. We run it on the separated models and on the controls of App. A.2 (seeds 1/2 for main models and tied-head Sequential, seed 1 otherwise). For state depth a the read maps are $\min(\ell, m)$ and $\max(\ell, a - m + 1)$. In Two-tower 12+12 the sweep pairs the m deepest levels with the $m{+}1$ shallowest, and Tab. 10 (middle, bottom) extends Fig. 3a to the other models.

Some restrictions also change which level a layer was trained to read, mixing depth with an untrained read. This affects the starred models and the extreme settings: limiting Two-tower 6+6 to levels ≥ 6 turns it into a final-state read, and limiting Sequential 12+12 to levels ≤ 12 into a same-depth read (costing 2.44). Both raise loss, which is why §5 retrains models rather than swapping reads. Training each prediction layer to read one fixed level would isolate depth directly.

Four models are omitted from the level tables. SPS, AF-SPS and Shared Two-tower 12+12 read state-layer inputs rather than outputs, and Shared Sequential 12+12's shared projections were never trained on the substituted levels.

C.3 Role Probes and Baselines

The probes behind §3 and §5.1 are compared with untrained and bigram baselines, which separates what training adds from what the architecture preserves (Tab. 12). They cover the untied Transformer, SPS and the separated models (seeds 1/2). Trained state keeps far more predecessor information than untrained state, while current-token readability falls with depth even before training.

Table 12: Last-layer linear-probe loss (nats/token) for the next token from prediction and the previous token from state, in trained and untrained models. The Transformer uses its single stream for both. $\pm$ as in §2.3, and untrained entries without $\pm$ use one initialization.

	Next token (prediction)		Previous token (state)	
Model	trained	untrained	trained	untrained
Transformer	$3.69_{\pm.01}$	$7.16_{\pm<.01}$	$3.98_{\pm.01}$	$7.11_{\pm<.01}$
SPS	$3.61_{\pm.01}$	$7.57_{\pm<.01}$	$2.70_{\pm.30}$	$7.19_{\pm<.01}$
Two-tower 12+12	$3.52_{\pm<.01}$	$7.56_{\pm<.01}$	$2.93_{\pm.06}$	$7.09_{\pm<.01}$
Sequential 12+12	$3.51_{\pm<.01}$	7.39	$2.03_{\pm<.01}$	7.09
Two-tower 6+6	$3.64_{\pm.01}$	7.73	$2.09_{\pm.12}$	7.13
Sequential 6+6	$3.63_{\pm.02}$	7.73	$1.95_{\pm.03}$	7.13

State, by contrast, predicts the next token much worse than prediction does, near the bigram baseline (Tab. 13), so what it preserves is specific to predecessors rather than a next-token forecast.

Even the final prediction probes stay above the LM head's loss, so linear probes miss some of what the model uses.

Table 13: Last-layer next-token probe loss (nats/token), means over seeds 1/2. The Prediction row's range spans SPS and the four separated models.

Model	Stream	Probe loss
SPS	State	5.35
Sequential 12+12	State	5.06
Two-tower 12+12	State	4.81
Two-tower 6+6	State	5.01
Sequential 6+6	State	5.00
Transformer	Single stream	3.69
SPS, Two-tower/Sequential at both depths	Prediction	3.5–3.7
Bigram baseline (same evaluation tokens)	–	5.24

C.4 Layer and Module Lesions

Whole-layer lesions support the claims of §4.2 and §5.1 about which layers matter, giving the profiles of Fig. 3b–d for SPS and the separated models (and the untied Transformer in the tables) (seeds 1/2, SPS mean ablation seed 1). To check that the profiles do not depend on how a layer is removed, we compare zero and mean ablation. They rank layers similarly but differ in absolute cost (Tab. 14), because mean replacement removes position-specific structure while adding a nonzero input. Which layers matter is more reliable than how much.

Table 14: By model, Spearman ρ between zero- and mean-ablation whole-layer lesion profiles for state and prediction, and the loss increase (nats/token) from lesioning state or prediction layer 1 under each ablation. Zero ablation removes a layer's residual contribution, and mean ablation replaces it with its per-channel mean on clean text. The Transformer, with one stream, is omitted. $\pm$ as in §2.3, and SPS has one seed.

	Spearman ρ		Layer 1, state		Layer 1, prediction	
Model	state	pred.	zero	mean	zero	mean
SPS	0.97	0.98	3.45	0.80	4.28	3.27
Two-tower 12+12	$0.97_{\pm<.01}$	$0.97_{\pm.02}$	$4.59_{\pm.32}$	$8.33_{\pm.78}$	$0.45_{\pm.01}$	$0.42_{\pm<.01}$
Sequential 12+12	$0.89_{\pm.01}$	$0.93_{\pm.07}$	$3.85_{\pm.18}$	$6.16_{\pm2.68}$	$0.76_{\pm.01}$	$0.57_{\pm.03}$
Two-tower 6+6	$0.97_{\pm.03}$	$0.86_{\pm.09}$	$5.71_{\pm.34}$	$6.60_{\pm.55}$	$0.86_{\pm.05}$	$0.50_{\pm.01}$
Sequential 6+6	$1.00_{\pm<.01}$	$0.91_{\pm.03}$	$3.79_{\pm.08}$	$4.03_{\pm.02}$	$2.05_{\pm.20}$	$0.86_{\pm.08}$

Sequential 12+12's final state layer stays costly under mean ablation (2.38/0.71 in seeds 1/2), far above Two-tower 12+12's (0.07–0.09), so its reliance on the read does not depend on the ablation type. In the 6+6 models no state layer after the first is cheap (Tab. 15), with state attention and MLP both contributing throughout (Fig. 7), so the cheap interior state layers of the 12+12 models are a property of depth.

Table 15: Minimum whole-layer zero-ablation state cost after layer 1 (nats/token), across layers and seeds 1/2.

Separate-weight model	Minimum cost
Two-tower 6+6	0.148
Sequential 6+6	0.236
Sequential 12+12	0.047
Two-tower 12+12	0.062

C.5 Gradient Alignment

Gradient alignment backs the looping hypothesis of §5 by asking whether state and prediction push shared weights in the same direction. We compare their gradient cosine per weight matrix at trained

weights and at matched initialization (Yu et al., 2020) (App. A.4). Positive cosines mean agreement, negative conflict, and near zero unrelated directions. The test covers every model with shared weights, with seeds as listed in Fig. 6.

Training moves Two-tower 12+12 (shared attn. only), AF-SPS and tied-head SPS toward positive alignment (Fig. 6), so where streams share weights they usually learn to push them the same way. Shared Two-tower 12+12 moves only slightly, and Shared Sequential 12+12 stays near zero.

Shared Two-tower 12+12's mean cosine is 0.030, with a bootstrap interval that excludes zero, while Shared Sequential 12+12 shows no detectable alignment. In the looped model, the streams' gradients are unrelated rather than conflicting, consistent with the looping account without testing it. Intervals reflect variation across matrices within a run, not across seeds.

In the separated 12+12 models (seeds 1/2), where no weights are shared, read-edge gradients ask whether final-state reads at least reduce gradient conflict within the state tower. A read-edge gradient is the part of a state matrix's gradient that arrives through one prediction layer's read, and in Two-tower the own-read edge is the read of that block's own output. Edges oppose the total gradient somewhat more often in Sequential (Tab. 16), so final-state reads do not reduce gradient conflict.

Table 16: Read-edge gradient statistics for the 12+12 separated models. Paired values are seeds 1/2.

Statistic	Two-tower 12+12	Sequential 12+12
Edges opposing total gradient	4.1% (both)	7.0% / 5.8%
Median edge/total cosine	0.54 / 0.55	0.45 / 0.48
Median own-read/deeper-read cosine	0.21 / 0.24	–
Negative own-read/deeper-read cosines	9%	–

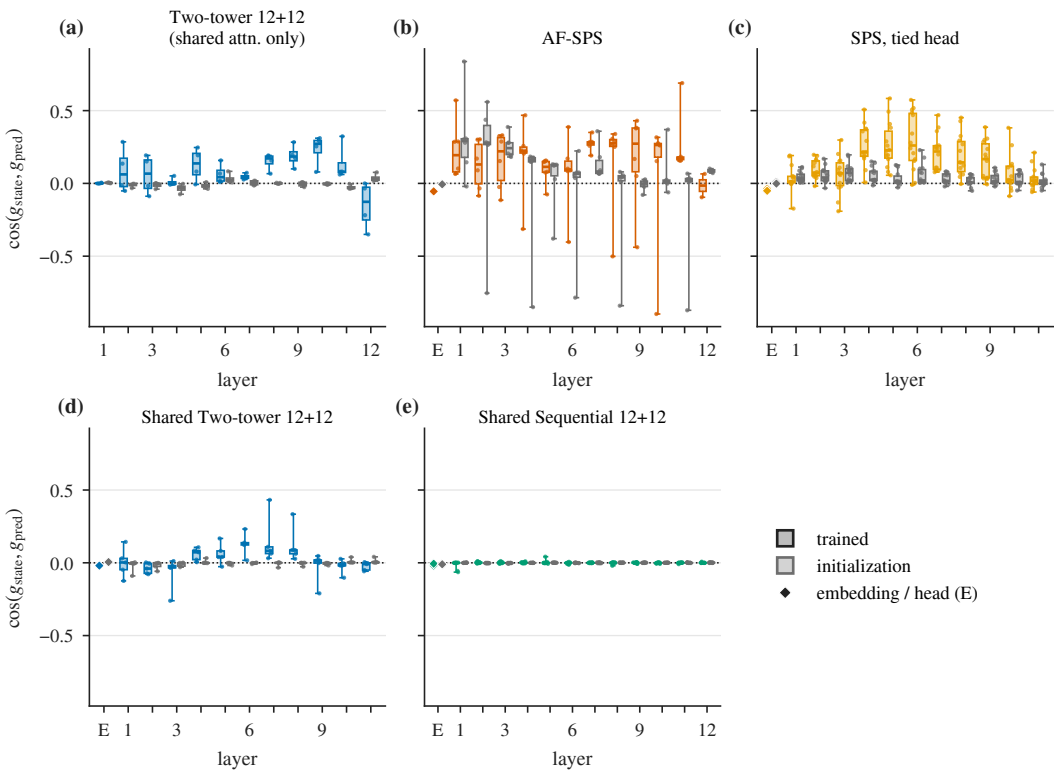


Figure 6: State–prediction gradient cosine per shared weight matrix and layer for (a) Two-tower 12+12 (shared attn. only), (b) AF-SPS, (c) tied-head SPS, (d) Shared Two-tower 12+12, and (e) Shared Sequential 12+12. Points denote individual matrices, colored boxes trained and grey boxes initialization cosines, and diamonds at E the embedding/head matrix, which is excluded from the summaries. (c) pools seeds 2/3, (e) seeds 1/2, and (a,b,d) use seed 1. The dotted line marks zero.

C.6 Seed Replication

Seed replication shows which intervention effects the claims of §§3–5 can rely on. We apply the 1.5× rule of App. A.1 to the untied Transformer and SPS, tied-head SPS (seeds 2/3), the separated models, and Shared Sequential 12+12 (seeds 1/2 unless noted).

Replicate in sign and magnitude: $N{=}64$ reach in the separated models and the Transformer, layer-1 whole-layer lesions (Tab. 14), first-m read restrictions in Two-tower 12+12 and Sequential 6+6, role probes, and Sequential 12+12's head ablations.

Same sign, varying magnitude: SPS head ablation and reach, Sequential 12+12's top-state-layer lesion (2.26/0.56), and several patching effects (Tabs. 6, 8, 10).

Do not replicate: head-ablation specificity outside Sequential 12+12, where corrected tests depend on seed or control pool (App. B.2).

The main-text claims about reach, read depth and Sequential 12+12's head specificity rest on replicating effects, while lesion and patching magnitudes support direction only.

D Architecture Controls and Efficiency

D.1 What the Variants Reveal

The architecture variants also clarify what the final-state read changes. Capacity and parameters do not explain the gain. Moving MLP width from state to prediction leaves Two-tower 12+12 at 2.737, against 2.733 before and 2.702 for Sequential 12+12 (Tab. 3), and replacing Sequential 6+6's

prediction embedding table with one learned ⟨predict⟩ token moves its loss only from 2.803 to 2.805 (App. D.5). Depth in both towers does matter. Shifting layers from either tower to the other costs Sequential 6+6 about the same (3+9 at 2.813, 9+3 at 2.815), and Two-tower 12+6 reads the deepest state levels yet trails Two-tower 12+12 by 0.024 (App. D.5). Reading deep state is not sufficient on its own, and prediction also needs layers to work with it, which is consistent with the processing-depth account of §5.1. Separate weights matter as well. Sharing only attention raises Two-tower 12+12 to 2.767, and with fully shared weights the ordering reverses (App. D.3), although these comparisons are exploratory. Finally, the efficiency result survives the cheaper designs we tested: with tied heads, Sequential 6+6 still recovers 72–82% of SPS's gain (App. D.2) and beats AF-SPS at fewer FLOPs (App. A.3, Tab. 21). Overall, the variants point to the read and to enough prediction depth to use it, rather than to extra capacity or parameters.

D.2 Models with Tied LM Heads

Head tying checks whether the extra output matrix of the untied main models drives §6's efficiency result, using the Transformer (untied seeds 1/2, tied seed 1), SPS (untied seeds 1/2, tied seeds 2/3) and tied-head Sequential 6+6 with a ⟨predict⟩ input (seeds 1/2). Tying helps the Transformer (2.865 to 2.851) but hurts SPS (2.797 to 2.807, Tab. 5). Tied-head Sequential 6+6 reaches 2.815/2.819, 0.010 above tied-head SPS, and its per-seed gains over the tied-head Transformer (0.036/0.032) recover 82%/72% of tied-head SPS's 0.044. The efficiency result holds with matched heads, at somewhat lower recovery.

D.3 Shared-Weight Models

The shared-weight models ask whether the read preference survives weight sharing, supporting §5's exploratory finding that it reverses. Shared Two-tower 12+12 (seed 1) and Shared Sequential 12+12 (seeds 1/2) run SPS's stack twice, without interleaving or a prediction-slot window. Shared Two-tower 12+12 reads level $\ell-1$, and Shared Sequential 12+12 reads final state through an extra K/V projection (1.18M parameters, 2.4M FLOPs/token). The references are tied-head SPS (seeds 2/3), the tied-head Transformer (seed 1) and Two-tower 12+12 (shared attn. only, seed 1).

The reversal already appears before decay: at matched compute (22.4k PFLOPs, interpolated), Shared Two-tower 12+12 reaches 2.970, level with tied-head SPS (2.969/2.970), while Shared Sequential 12+12 trails at 2.988/2.987. Sharing only attention also costs Two-tower 12+12 (2.733 to 2.767, seed 1, Tab. 5), so what is shared matters, not only how many parameters. With the confounds in §5 and a single Shared Two-tower 12+12 seed, this comparison cannot isolate looping (gradient evidence: App. C.5).

Sharing also weakens the causal retrieval evidence. Shared Two-tower 12+12's head ablation depends on the control pool (Tab. 6), and head patching recovers nothing (Tab. 8), so the division of retrieval in §3 needs separate weights.

D.4 Where Sequential's Gain Falls

Breaking down Sequential's gain over Two-tower (§5.1) by token type tests whether it comes from copying. We use the separated models and the untied Transformer, over seeds 1/2.

Repeat distance. Bucketing target tokens by how far back they last occurred shows whether the gain concentrates on copyable tokens (Tab. 17).

Every bucket improves in all four seed pairings at both depths with similar relative gains, and first-occurrence tokens take 59–63% of the gain at 12+12 and 55–61% at 6+6 in every pairing, so the breakdown does not depend on how seeds are paired.

Versus the Transformer. Against the Transformer, SPS and Sequential 12+12 also improve on first-occurrence tokens in both seeds, though less in relative terms than on repeated tokens (1.9–2.2% versus 2.3–3.5% for SPS, 5.1–5.4% versus 5.7–6.8% for Sequential). First-occurrence tokens carry 53–55% of SPS's gain and 58% of Sequential's.

Table 17: Sequential minus Two-tower loss by target-token repeat distance within a document for 12+12 and 6+6: token share, loss difference (nats/token), relative difference (%), and share of gain. Values are seed means on the repeat-distance sweep (512 sequences, 2.09M tokens), so the "all" row differs slightly from Tab. 3's full-sweep Δ.

		12+12			6+6		
Last occurrence	tokens	Δ	rel.	gain	Δ	rel.	gain
First-occurrence	42.8%	-0.043	-1.1%	61%	-0.050	-1.2%	58%
1–16 back	13.9%	-0.012	-0.9%	5%	-0.017	-1.2%	6%
17–64	21.0%	-0.017	-1.2%	12%	-0.022	-1.5%	13%
65–256	14.5%	-0.026	-1.3%	12%	-0.032	-1.5%	13%
257–1024	6.6%	-0.035	-1.3%	8%	-0.042	-1.4%	8%
$>$1024	1.2%	-0.040	-1.2%	2%	-0.063	-1.7%	2%
all	100%	-0.030			-0.036		

Induction tokens. The induction sweep (12,651 qualifying tokens) additionally requires a preceding-bigram match (App. A.4). A group's gain share is $(n_{\text{induction}}/n_{\text{all}})(\Delta_{\text{induction}}/\Delta_{\text{all}})$, where n counts tokens and Δ is Sequential minus Two-tower loss. The same-seed estimates in Tab. 18 all fall below the induction tokens' share of tokens, so induction tokens receive less than their share of the gain in every seed.

Table 18: Share of Sequential's gain over Two-tower on induction tokens, by matched training seed and depth.

Depth	Seed 1	Seed 2
12+12	9.9%	14.4%
6+6	8.2%	10.6%

Prediction depth. Sequential's prediction stream becomes next-token readable earlier: its probe advantage over Two-tower is largest in early and middle layers and closes at the output (Tab. 19), and it crosses fixed probe-loss thresholds several layers sooner (Tab. 20), as expected if prediction receives processed state earlier.

Table 19: Two-tower minus Sequential next-token probe loss (nats/token). Ranges span all four pairings of seeds 1/2 at each depth.

Depth	Prediction layer	Probe-loss gap
12+12	1	0.245–0.280
12+12	4	0.887–1.018
12+12	11	0.136–0.148
12+12	12	0.005–0.009
6+6	1	0.763–0.811
6+6	6	-0.006 to 0.041

Table 20: First prediction layer reaching each next-token probe-loss threshold (nats/token), in seeds 1/2. The Transformer uses its single stream, and – denotes a threshold not reported here.

Model	5.000	4.500
Sequential 12+12	2	5
Two-tower 12+12	6	8
Sequential 6+6	2	3
Two-tower 6+6	3	4
Transformer (untied)	7	–

D.5 Depth, Input and Pre-Registered Controls

These controls support §5's claim that balanced towers work best and §6's that the prediction embedding table does not drive the gain, testing alternatives to the read explanation: tower balance, the prediction input and the pre-registered targets. Sequential 3+9, Sequential 9+3, Sequential 6+6 (⟨predict⟩ input) and Two-tower 12+6 (top-aligned) use seed 1, the tied-head SPS reference uses seed 2, and the separated models, Transformer and SPS use seeds 1/2.

Tower balance. Shifting layers between the towers at fixed cost does not help: balanced Sequential 6+6 (2.801) beats 3+9 (2.813) and 9+3 (2.815) (Tab. 5), and every layer of the shortened towers contributes (Fig. 7). The gain does not come from giving either role more depth. The two unbalanced splits cost about the same (0.010 and 0.012, seed 1), which indicates that both towers carry comparable load, as the balanced lesion profiles of Sequential 12+12 also suggest (§5.1).

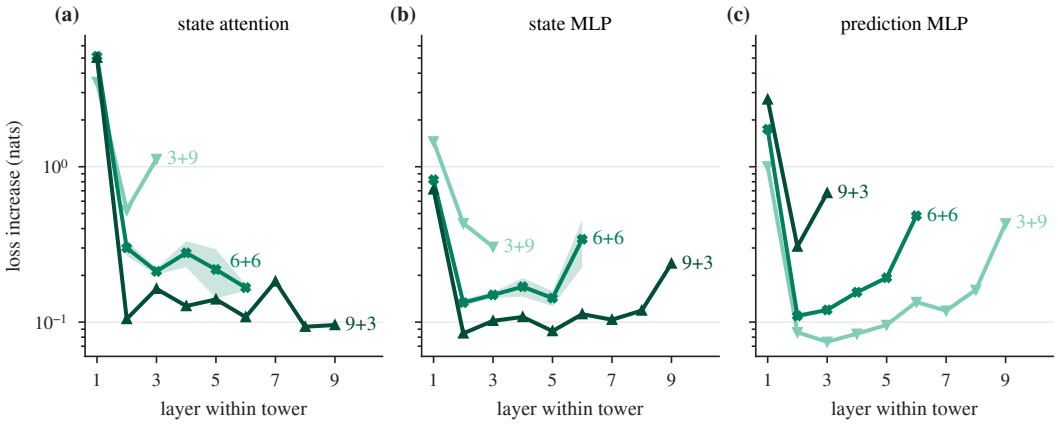


Figure 7: Module lesion costs (nats/token, log scale) by layer within each tower for Sequential 3+9, 6+6 and 9+3: (a) state attention, (b) state MLP, (c) prediction MLP. Shaded bands show the range over two seeds for 6+6, and Sequential 3+9 and 9+3 use seed 1.

Prediction input. Training with a single learned ⟨predict⟩ token instead of a prediction embedding table stays within the tie threshold (seed 1: 2.805 versus 2.803, Tab. 5), so the extra table is not what drives the gain. The prediction stream draws what it needs from state rather than from its own token input. Swapping the input only at inference costs 0.935–1.333 nats/token, so the model still learns to use whichever input it is trained with, even though that input does not change the final loss.

Pre-registered targets. The registered Sequential 6+6 seed-1 test used an untied head and its own prediction embedding table at Transformer FLOPs. It targeted Two-tower 12+12 seed 1, with tied-head SPS seed 2 as reference: success $L \leq 2.752$, partial success $2.752 < L \leq 2.782$, failure $L \geq 2.806$, and no claim for $2.782 < L < 2.806$. Its loss of 2.803 fell in the no-claim band: halving both towers costs Sequential about 0.07 nats/token relative to the 12+12 reference, more than the read recovers. That miss is why the efficiency result is framed against SPS and the Transformer instead. Untied SPS was trained afterward, so §6's head-matched comparison is post hoc.

Top-aligned Two-tower 12+6 asks whether reading deep state is enough when the prediction tower is halved. Its registered targets, relative to Two-tower 12+12 seed 1, were success $L \leq 2.737$ and partial success $L \leq 2.752$, and its loss of 2.757 misses both. Deep access alone does not recover 12+12 performance: with the deepest state available but half the prediction layers, Two-tower 12+6 still trails Two-tower 12+12's seed mean by 0.024, which indicates that prediction depth, not only access to deep state, limits the loss.

D.6 Compute Matching and Full Trajectories

The efficiency comparison of §6 depends on how compute is counted. Compute in Fig. 4 is tokens $\times 3\times$ forward FLOPs/token, for the untied Transformer, SPS and the separated models (seeds 1/2, seed means unless paired). The pre-decay comparison uses the 18B-token checkpoints of the Transformer and the 6+6 models and interpolates SPS and the 12+12 models near 9.9B tokens in cumulative compute before averaging seeds. Run endpoints include each run's own decay schedule, so the two comparisons answer different questions. At the endpoints, same-seed Sequential 6+6 minus SPS gaps are $+0.0050/+0.0045$ nats/token, with 92%/94% gain recovery and 88–97% across all seed pairings (intervals: App. A.1).

D.7 Wall-Clock Throughput and Decode Cost

Throughput checks whether FLOP savings become wall-clock savings. We time each model in Tab. 21 in its 20B configuration with random weights on one H100, using `torch.compile`, bf16 and fused AdamW, and report the median of 30 steps after 10 warmup steps at micro-batch 12×4096, over two repetitions (spread $\leq$1.4%).

Sequential 6+6 trains at about the Transformer's speed and prefills (processes prompts) faster than the 12+12 separated models (Tab. 21). With Triton, SPS slows by more than its FLOP increase, because causal attention over $2T$ positions takes 212 of 406 ms per step against the Transformer's 57 ms.

Sequential 6+6 stores 12 K/V sets, six from state layers and one final-state projection per prediction layer, so its cache matches the Transformer's. The separated models have no incremental decode implementation, so their decode speed is unmeasured.

Table 21: Training and prefill (prompt-processing) throughput (thousand tokens/s), training speed relative to the tied-head Transformer using the same attention kernel, bf16 K/V cache (KiB/token), and forward GFLOPs/token, by model, head setting and kernel (Triton or flex attention, as marked). The flex-kernel Transformer reference ran on a different node, so flex-kernel speed ratios are cross-node. Superscripts refer to the notes below the table.

			Train		Prefill	K/V	GFLOPs
Model	Head	Kernel	ktok/s	Speed	ktok/s	KiB/token	/token
Transformer	tied	Triton	270	1.00	734	36	0.414
Transformer[b]	tied	flex	343	1.00	–	36	0.414
SPS	tied	Triton	121	0.45	393	36	0.756
Two-tower 12+12[a]	untied	flex	197	0.57	548	72	0.751
Sequential 12+12[a]	untied	flex	196	0.57	550	72	0.751
Sequential 6+6[a]	untied	flex	350	1.02	976	36	0.414
Sequential 6+6 (tied head, ⟨predict⟩ input)[a]	tied	flex	351	1.02	978	36	0.414
AF-SPS[a]	tied	flex	234	0.68	647	36	0.595

[a]No incremental decode implementation. [b]No successful prefill measurement in this benchmark run.

E Related Work Details

The architectures below separate or reuse computation in different ways, but none varies the read level with separate weights, the one variable Sequential changes.

Shared computation. DBT (Liu et al., 2026b) reads same-depth primary K/V without writing K/V of its own, and shares attention, MLP and output matrices. PHD-Transformer (Wu et al., 2025b), PLT (Wu et al., 2025a), ParScale (Chen et al., 2025) and Hidden Decoding (Liu et al., 2026a) also add per-token compute through shared weights. None gives the streams separate weights, so none can isolate the read choice (AF-SPS details: App. A.3).

Cross-layer K/V reuse. LCKV (Wu & Tu, 2024) reads top-layer K/V after warmup layers and trains through iterative parallel computation. YOCO's cross-decoder reads the final K/V of its self-decoder (Sun et al., 2024), and CLA shares K/V between adjacent layers (Brandon et al., 2024). All three continue a single residual stream rather than separate state and prediction streams. The Feedback Transformer goes further, reading a weighted mixture of past-token layers at the cost of sequential training (Fan et al., 2020).

Encoder–decoder reads. Machine-translation studies compare same-depth reads (He et al., 2018), learned mixtures over encoder layers (Bapna et al., 2018), lower-layer access (Domhan, 2018), embedding-driven fusion (Liu et al., 2021) and encoder/decoder depth splits (Kasai et al., 2021), all over a separate source sequence. Decoder-only versus encoder–decoder comparisons (Wang et al., 2022; Elfeki et al., 2025; Zhang et al., 2025; Ho et al., 2024) read final state but change the whole architecture rather than only the read level. XLNet shares weights across two attention streams (Yang et al., 2019).