THE CONVEX CORRIDOR: LEARNING ATTENTION–LAPLACIAN MIXTURES HELPS PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention computes input-dependent weighted averages of token vectors through the softmax probability matrix P. Recently proposed Laplacian heads instead use $I - P$, where I is the identity matrix, to compute the deviation of each token vector from its corresponding average. Although using both Laplacian and attention heads in transformers has been shown to improve performance on many tasks, how to best combine the two operators is still unclear. We address this problem by parameterizing each head's operator as $\delta(\gamma I + P)$, where $\gamma, \delta \in \mathbb{R}$ are learnable parameters. This modification consistently leads to lower validation loss in causal language modelling pretraining across tested model scales. To understand why, we examine the evolution of γ and δ throughout training, and find that most heads beyond the first layer learn $\gamma \in (-1, 0)$, an interval we call the *convex corridor*. When γ falls within this corridor, $\delta(\gamma I + P) = \delta(|\gamma|(P - I) + (1 - |\gamma|)P)$, so each head's operator becomes a scaled convex combination of attention P and the negative Laplacian $P - I$, with γ controlling the weights of the two operators and δ controlling the size of the head's output. This pattern emerges across different initializations and alternative parameterizations of the operator such as $\alpha_0 I + \alpha_1 P$ and $\alpha P + \beta(I - P)$, which also improve pretraining. We perform a series of controlled experiments to show that γ being in the convex corridor contributes to the improvements in pretraining: restricting $\gamma \geq 0$ increases loss, while fixing γ to be some value within the convex corridor already improves upon standard attention even with δ fixed to 1. Together, these findings suggest that attention should be viewed as one endpoint of the convex corridor, and that exploring the corridor's interior is a promising direction for future transformer architectures.

1 INTRODUCTION

Multi-head attention is at the heart of the transformer architecture (Vaswani et al., 2017). Given a sequence of token vectors $X \in \mathbb{R}^{T \times d}$, each head h computes a softmax attention probability matrix

$$P^{(h)} = \text{softmax}\Big(XW_{Q_h}W_{K_h}^{\top}X^{\top}/\sqrt{d_k}\Big) \in \mathbb{R}^{T \times T}$$

and outputs $P^{(h)}XW_{V_h}$[1], where $W_{Q_h}, W_{K_h} \in \mathbb{R}^{d \times d_k}, W_{V_h} \in \mathbb{R}^{d \times d_h}$ are the query, key, and value projections, respectively. Because $P^{(h)}$ is row-stochastic, attention can be interpreted as computing a weighted average of XW_{V_h}.

Zhang & Papyan (2026) recently proposed Laplacian heads, where each head instead computes the deviation of token vectors from their attention-weighted means through $(I - P^{(h)})XW_{V_h}$. The motivation was to give the model more direct control over the within-sequence variance of token representations. Based on a different interpretation of transformers as unrolled inference steps of a probabilistic Laplacian Eigenmaps model, Ravuri & Lawrence (2026) also argue that the negative Laplacian $P - I$ is more natural for transformers than P. Incorporating Laplacian heads into the transformer has been shown to improve performance on various tasks.

However, it remains unclear how transformers should balance attention heads, which compute weighted token averages, with Laplacian heads, which compute deviations from those averages.

[1]Here, we assume X has already been normalized, following the pre-layer normalization convention (Xiong et al., 2020).

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.

Prior work replaced a fixed number of attention heads with Laplacian heads in each layer (Zhang & Papyan, 2026), but this might be a suboptimal strategy.

In this paper, we address this problem by modifying attention to $\delta^{(h)}(\gamma^{(h)}I + P^{(h)})XW_{V_h}$, where $\gamma^{(h)}, \delta^{(h)} \in \mathbb{R}$ are learnable coefficients for each head. This parameterization allows each head to learn its own balance between attention and Laplacian and how to scale the size of its output.

Our central finding across 1,715 language modelling pretraining runs spanning different initialization schemes and hyperparameters is that most heads beyond the first layer learn $\gamma^{(h)} \in (-1, 0)$.

We name the interval $(-1, 0)$ the *convex corridor*, because when $\gamma \in (-1, 0)$, the operator is a scaled convex combination of standard attention and the negative Laplacian, as shown in the following equation:

$$\delta^{(h)}(\gamma^{(h)}I + P^{(h)}) = \delta^{(h)}\big(|\gamma^{(h)}|(P^{(h)} - I) + (1 - |\gamma^{(h)}|)P^{(h)}\big). \tag{1}$$

Under this decomposition, the learnable parameter $\gamma^{(h)}$ can be interpreted as controlling the weight of attention or negative Laplacian in each head, while $\delta^{(h)}$ controls the size of each head's output.

Contributions. In summary, our first contribution is showing that the proposal improves pretraining loss for causal language modelling, adding only a tiny number of parameters (Section 3).

Our second contribution is an in-depth empirical analysis showing that γ belonging to the convex corridor contributes to the improvements in pretraining (Sections 4–7), with the following evidence:

- Different parameterizations such as $\alpha_0 I + \alpha_1 P$ and $\alpha P + \beta(I - P)$ also improve pretraining loss and learn $\gamma^{(h)} \in (-1, 0)$ (Sections 5 and 6);
- Constraining $\gamma^{(h)}$ to be nonnegative throughout training increases loss (Section 7.1);
- Fixing $\gamma^{(h)}$ to be a constant value within the convex corridor already improves pretraining loss upon standard attention, even with $\delta^{(h)}$ fixed to 1 (Section 7.2).

2 Related work

Laplacian heads. Zhang & Papyan (2026) proposed Laplacian heads $I - P$ and showed that using a fixed number of Laplacian heads in each layer alongside attention heads P improves performance on various tasks, with stronger token smoothing observed across modalities. Ravuri & Lawrence (2026) interpret transformers as unrolled inference steps of a probabilistic Laplacian Eigenmaps model and argue that $P - I$ is more natural for transformers than P. We propose a learnable parameterization that lets each head learn how to combine attention and the Laplacian through $\delta(\gamma I + P)$, and find that most heads learn $-1 < \gamma < 0$, giving the scaled convex combination of attention and the negative Laplacian in Equation 1.

Combinations of identity and attention. The combination $\alpha_0 I + \alpha_1 P$ already appears in ValueSkipInit by He et al. (2023), which initializes $(\alpha_0, \alpha_1) = (1, 0)$ and learns both coefficients to train transformers *without* skip connections. He et al. (2024) study fixed positive combinations $I + \beta P$ to reduce the dominance of unusually large feature activations. We instead show that the proposed $\delta(\gamma I + P)$ parameterization improves pretraining of transformers *with* skip connections. We also use a series of intervention experiments to show that $\delta(\gamma I + P)$ contributes to improvements in pretraining by learning the scaled convex combination in Equation 1.

Polynomial attention. Polynomial attention uses linear combinations of I and higher powers of P. TabPSA learns coefficients for these polynomial terms (Kim et al., 2024), while GFSA combines I and P with an approximation to a higher power of P (Choi et al., 2024). Their evaluations include fine-tuning pretrained BERT and GPT-2 models. Our experiments instead focus on causal language modelling pretraining, and our interpretation is centered on the distinct roles of attention and Laplacian rather than higher powers of P.

Frequency-based attention modulation. FDAM and FAiT interpret P and $I - P$ as low-pass and high-pass filters, combining them to recover high-frequency information in dense visual prediction and time-series forecasting, respectively (Chen et al., 2025; He et al., 2026). Both methods also introduce explicit Fourier-domain modulation of feature representations. Our study instead examines how attention and Laplacian are balanced during causal language-model pretraining. Our key contribution is showing that most heads beyond the first layer learn a scaled convex combination of attention and the negative Laplacian across different initializations and parameterizations. Through intervention experiments, we provide evidence that this pattern contributes to improved pretraining loss.

Centering and differential attention. Shaped Transformer uses $I + P_\tau - U$, where $U = \frac{1}{T}\mathbf{1}\mathbf{1}^\top$ and P_τ denotes attention with a width-dependent softmax temperature. It also studies $aI + P_\tau - bU$ with learnable coefficients a, b (Noci et al., 2023). Simplifying Transformer Blocks uses $a_h I + b_h P^{(h)} - c_h C$, with all three coefficients learned separately for each head (He & Hofmann, 2024), where C averages uniformly over tokens allowed by the attention mask. AttnScale learns how strongly to weigh the difference between the attention output and the uniform mean, then adds the uniform mean back (Wang et al., 2022). Centered Self-Attention uses the difference directly (Ali et al., 2025), while Differential Transformer subtracts a second attention-weighted mean computed from another attention matrix (Ye et al., 2025). In contrast, our proposal does not rely on a uniform mean, and instead allows each head to balance between P and $I - P$.

Values from earlier layers. NeuTRENO and ResFormer use value vectors from the first layer to modify attention in later layers. NeuTRENO adds a scaled difference between first-layer and current-layer values to the attention output (Nguyen et al., 2023), while ResFormer combines these values before computing the attention-weighted mean (Zhou et al., 2025). Our proposal instead allows each head to apply a scaled convex combination of P and $P - I$ to the current layer's value vectors when $\gamma \in (-1, 0)$, without relying on value vectors from earlier layers.

3 Language-model pretraining experiments

We used the nanochat repository (Karpathy, 2025) to pretrain language models from scratch at different model sizes and training budgets. Specifically, we trained transformers with 12 and 20 layers (d12 and d20), containing approximately 185.6M and 561.0M parameters, respectively. We trained d12 on 2.10B tokens and d20 on 5.73B tokens from FineWeb-Edu (Lozhkov et al., 2024). At each model size, we compared the following three models:

- Attention, which uses standard attention P for all heads.
- Laplacian, which uses the Laplacian operator $I - P$ for all heads.
- Trainable Mixture, which uses the proposed $\delta(\gamma I + P)$ for each head, with (δ, γ) initialized to $(1, 0)$ for heads in the first layer, and $(1, -1)$ for heads in the remaining layers.

We optimized the matrix weights in the transformer blocks with Muon (Jordan et al., 2024), and the embeddings, output weights, and learned coefficients with AdamW (Loshchilov & Hutter, 2019). The full training setting is specified in Appendix A. For each of the three models, we report validation bits per byte (BPB) across three random seeds.

Figure 1 shows the same ordering at both model sizes: Laplacian heads achieve lower validation BPB than standard attention, and the proposed $\delta^{(h)}(\gamma^{(h)} I + P)$ parameterization improves further. The improvement of the proposal over the Laplacian is approximately twice as large as the improvement of Laplacian heads over standard attention. This ordering holds for all three seeds at each model size. These results demonstrate the benefit of the proposed parameterization and motivate our subsequent analysis of how the learned coefficients evolve during training.

4 Evolution of γ and δ throughout training

Does the proposed parameterization learn a scaled convex combination of attention and the negative Laplacian, as described in Equation 1, or are the learned heads much closer to either attention or

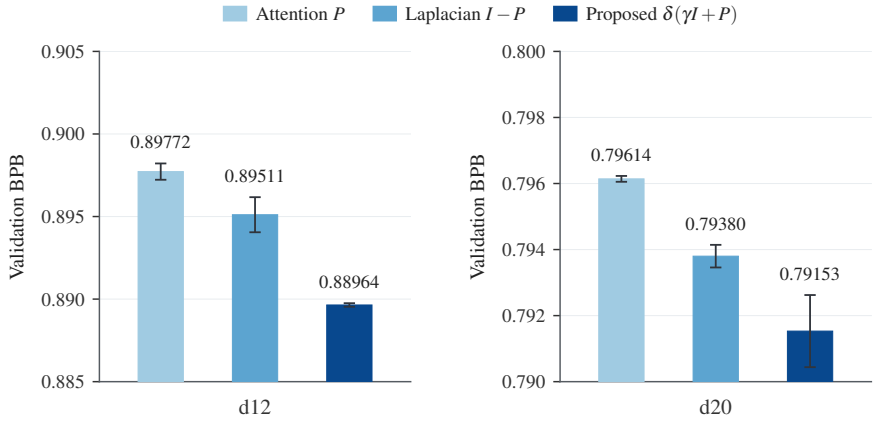
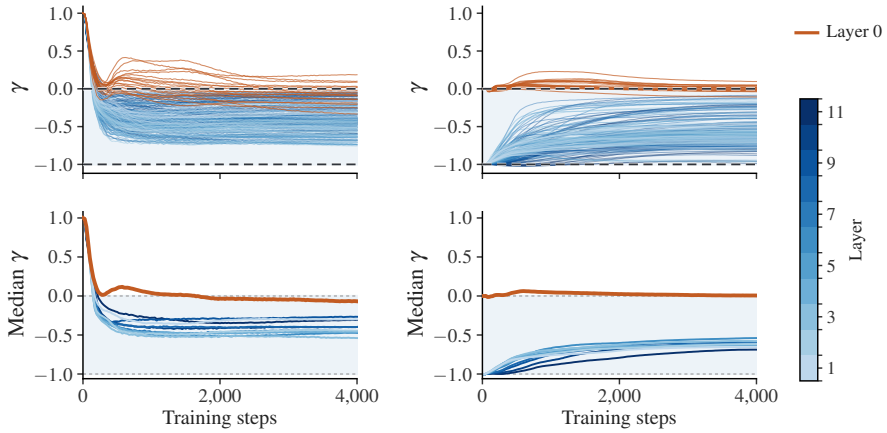


Figure 1: **Validation BPB across two tested model sizes.** We compare standard attention P, Laplacian $I - P$, and the proposed $\delta^{(h)}(\gamma^{(h)} I + P)$. The proposal achieves the lowest mean BPB at both tested model sizes.

Laplacian? To answer this, we trained d12 models on 2.10B tokens under several initialization schemes, with the initial values $\gamma_{\text{init}} \in \{-1.5, -1, -0.75, -0.5, 0, 0.5, 0.75, 1, 1.5\}$, $\delta_{\text{init}} = 1$, and tracked the evolution of $\gamma^{(h)}$ and $\delta^{(h)}$ for every head throughout training. Below, we refer to the first layer as layer 0 and the remaining layers as layers 1–11.

Figures 2 and 3 present the evolution of γ and δ, respectively, for two particular initializations.

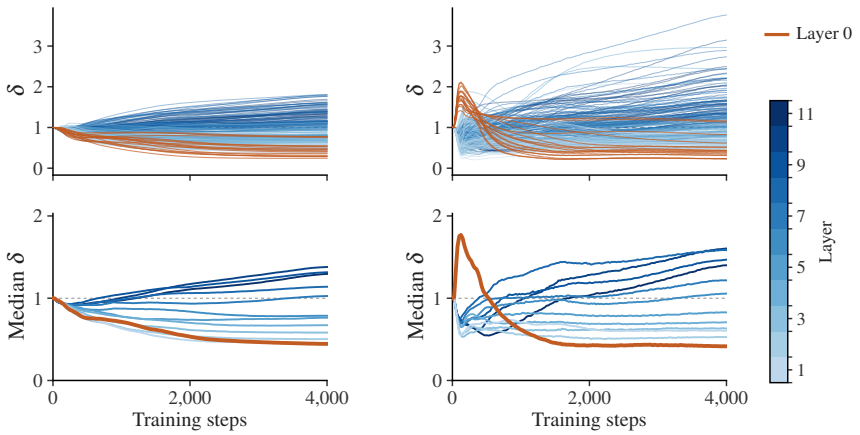


(a) $\gamma_{\text{init}} = 1$ in every layer. Throughout training, later layers' γ moves into the convex corridor $(-1, 0)$ while in layer 0 it approaches zero.

(b) $\gamma_{\text{init}} = 0$ in layer 0; -1 in layers 1–11. Throughout training, layer 0's γ stays close to 0, while later layers' γ starts at -1 and moves into the convex corridor $(-1, 0)$.

Figure 2: **Evolution of γ throughout training.** Top: one curve per head and seed (six heads per layer, three seeds). Bottom: each layer's median across these 6 heads and 3 seeds at each training step. Shading marks the convex corridor $(-1, 0)$. Both initializations use $\delta_{\text{init}} = 1$ in every layer. Heads in layer 0 converge towards attention, while heads in later layers converge to a scaled convex combination of attention and the negative Laplacian.

Figure 2 shows that under both initializations, for heads in layers 1–11, $\gamma^{(h)}$ moves into the convex corridor $(-1, 0)$ early in training and converges to some value within $(-1, 0)$. In particular, Figure 2(a) shows that if $\gamma^{(h)}$ is initialized at 1, then it crosses zero and moves into the convex corridor, showing that the observed pattern emerges without imposing it at initialization. On the other hand, Figure 2(b) shows that if $\gamma^{(h)}$ is initialized at -1, then it converges closer to the centre of the convex



(a) $\gamma_{\text{init}} = 1$ in every layer.

(b) $\gamma_{\text{init}} = 0$ in layer 0; -1 in layers 1–11.

Figure 3: **Evolution of δ throughout training.** Top: one curve per head and seed (six heads per layer, three seeds). Bottom: each layer's median across these 18 values at each training step. In deeper layers, the coefficients $\delta^{(h)}$ learn larger values, increasing the size of each head's output.

corridor. For most heads in layer 0, $\gamma^{(h)}$ moves towards or stays close to zero throughout training, which implies that many heads are close to attention.

Meanwhile, Figure 3 shows that in deeper layers, $\delta^{(h)}$ converges to larger values, which implies that the model learns to increase the size of each head's output.

These observed patterns hold consistently across the additional initializations we tested (Appendix B). Across 417 distinct, completed, unrestricted d12 runs under the $\delta(\gamma I + P)$ parameterization, 97.0% of later-layer heads learned $-1 < \gamma^{(h)} < 0$ (Table 3), which implies that the scaled convex combination of standard attention and negative Laplacian described in Equation 1 applies to most heads beyond the first layer.

5 Evolution of coefficients under alternative parameterizations

A natural question is whether the observed coefficient pattern is sensitive to the $\delta(\gamma I + P)$ parameterization of each head, and whether different parameterizations would lead to different patterns. To investigate this, we trained models using two alternative parameterizations.

$\alpha_0 I + \alpha_1 P$ parameterization. Each head is parameterized as $\alpha_0^{(h)} I + \alpha_1^{(h)} P$, where $\alpha_0^{(h)}$ and $\alpha_1^{(h)}$ are learnable coefficients. When $\alpha_1^{(h)} \neq 0$, we can write

$$\alpha_0^{(h)} I + \alpha_1^{(h)} P = \alpha_1^{(h)} \big(\frac{\alpha_0^{(h)}}{\alpha_1^{(h)}} I + P\big). \tag{2}$$

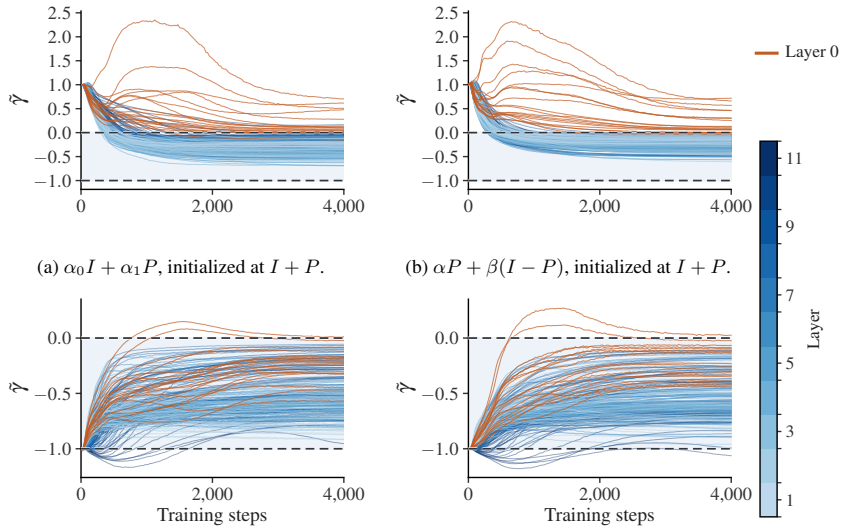
Thus, we can rewrite $\alpha_0^{(h)} I + \alpha_1^{(h)} P$ in the form $\tilde{\delta}^{(h)}(\tilde{\gamma}^{(h)} I + P)$ by defining $\tilde{\gamma}^{(h)} = \alpha_0^{(h)}/\alpha_1^{(h)}$ and $\tilde{\delta}^{(h)} = \alpha_1^{(h)}$.

$\alpha P + \beta(I - P)$ parameterization. Each head is parameterized as $\alpha^{(h)} P + \beta^{(h)}(I - P)$, with $\alpha^{(h)}$ and $\beta^{(h)}$ being learnable coefficients. When $\alpha^{(h)} - \beta^{(h)} \neq 0$, we can write

$$\alpha^{(h)} P + \beta^{(h)}(I - P) = (\alpha^{(h)} - \beta^{(h)}) \big(\frac{\beta^{(h)}}{\alpha^{(h)} - \beta^{(h)}} I + P\big). \tag{3}$$

Thus, we can rewrite $\alpha^{(h)} P + \beta^{(h)}(I - P)$ in the form $\tilde{\delta}^{(h)}(\tilde{\gamma}^{(h)} I + P)$ by defining $\tilde{\gamma}^{(h)} = \beta^{(h)}/(\alpha^{(h)} - \beta^{(h)})$ and $\tilde{\delta}^{(h)} = \alpha^{(h)} - \beta^{(h)}$.

Evolution of $\tilde{\gamma}^{(h)}$ and $\tilde{\delta}^{(h)}$ throughout training. Figure 4 shows that for most heads beyond the first layer, $\tilde{\gamma}^{(h)}$ converges to the convex corridor $(-1, 0)$. In particular, for the $I + P$ initialization, we see $\tilde{\gamma}^{(h)}$ crossing over zero. Meanwhile, Figure 12 in Appendix C shows that in deeper layers, $\tilde{\delta}^{(h)}$ learns larger magnitudes. These patterns are consistent with the results in Figures 2 and 3.



(a) $\alpha_0 I + \alpha_1 P$, initialized at $I + P$.
(b) $\alpha P + \beta(I - P)$, initialized at $I + P$.
(c) $\alpha_0 I + \alpha_1 P$, initialized at $I - P$.
(d) $\alpha P + \beta(I - P)$, initialized at $I - P$.

Figure 4: Individual-head $\tilde{\gamma}$ trajectories across both parameterizations and initializations. Columns show the $\alpha_0 I + \alpha_1 P$ and $\alpha P + \beta(I - P)$ parameterizations; rows show the $I + P$ and $I - P$ initializations. Each panel contains one curve per head and seed (six heads per layer, three seeds). Blue shades identify the layer. For most later-layer heads, $\tilde{\gamma}^{(h)}$ converges to the convex corridor $(-1, 0)$, consistent with the pattern for the $\delta(\gamma I + P)$ parameterization.

6 Correlation between the convex corridor and pretraining improvement

Section 4 and Section 5 show that the pattern of $\tilde{\gamma}^{(h)}$ converging to the convex corridor emerges consistently across different initializations and parameterizations. Furthermore, Figure 5 shows that all of these tested parameterizations and initializations achieved lower validation loss than both attention P and Laplacian $I - P$. Together with Section 3, these results establish a correlation between learning γ in the convex corridor and improvements in pretraining loss.

7 Intervention experiments

Section 6 established a correlation between pretraining improvements and two learned coefficient patterns: $\tilde{\gamma}^{(h)}$ converges to the convex corridor and $\tilde{\delta}^{(h)}$ learns larger magnitudes in deeper layers. To test the causal contribution of these patterns, we perform a series of intervention experiments that restrict what values $\gamma^{(h)}$ and $\delta^{(h)}$ can learn. We then observe whether these interventions increase or decrease validation loss.

7.1 Clipping γ throughout training

First, for all heads in layers 1–11, we restricted $\gamma^{(h)}$ to be outside of the convex corridor via clipping:

- We initialized $\gamma^{(h)} \in \{0, 0.5, 1\}$ and set $\gamma^{(h)} \leftarrow \max(0, \gamma^{(h)})$ after each optimizer update.
- We initialized $\gamma^{(h)} = -1$ and set $\gamma^{(h)} \leftarrow \min(-1, \gamma^{(h)})$ after each optimizer update.

We initialized $\gamma^{(h)} = 0$ in layer 0 and $\delta^{(h)} = 1$ in all layers, and applied no restriction to these parameters. We trained models under either restriction using two sets of learning rates: $(\eta_\gamma, \eta_\delta) =$

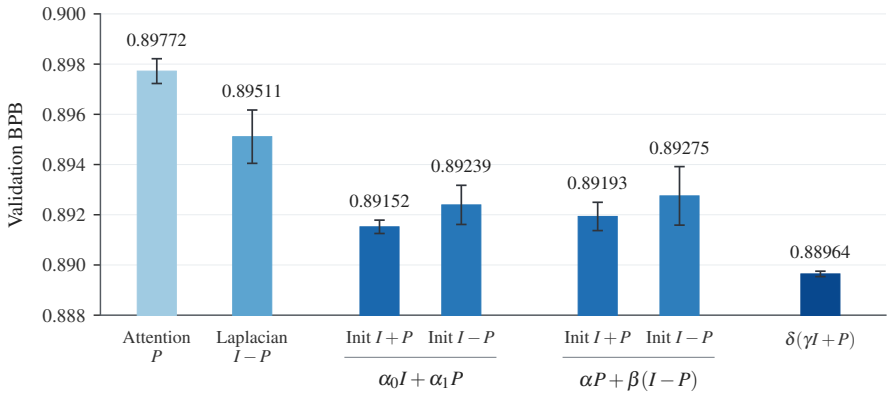


Figure 5: Validation loss (mean ± standard deviation) across parameterizations and initializations. Attention and Laplacian use P and $I-P$ in all heads, respectively. The parameterizations $\alpha_0 I+\alpha_1 P$ and $\alpha P+\beta(I-P)$ use the same runs as Figure 4, and $\delta(\gamma I+P)$ uses the same runs as Figure 2(b).

$(0.0025, 0.02)$ and $(0.005, 0.005)$, where η_γ is the learning rate for γ and η_δ is the learning rate for δ. We compared the validation loss of these restricted models to their unrestricted counterparts, where $\gamma^{(h)}$ learns freely and can converge to the convex corridor.

Figure 6 shows that for each initialization and learning rate pair, clipping $\gamma^{(h)}$ throughout training increases validation loss compared to learning $\gamma^{(h)}$ with no restriction. This is true for both restrictions $\gamma^{(h)} \geq 0$ and $\gamma^{(h)} \leq -1$. Because learnable $\gamma^{(h)}$ converges to $(-1, 0)$ for most heads in layers 1–11, these results suggest that learning $\gamma^{(h)}$ in the convex corridor contributes to improvements in validation loss, while preventing this learning hurts validation loss.

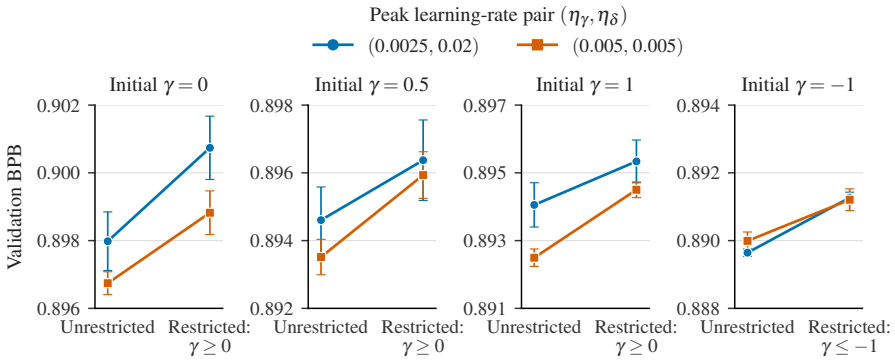


Figure 6: Clipping γ in layers 1–11 to be outside of the convex corridor increases validation BPB at both learning-rate pairs for every initialization. Each line connects unrestricted and restricted mean validation BPB at one learning-rate pair. Error bars show sample standard deviations.

7.2 Fixing $\gamma^{(h)}$ inside the convex corridor

The preceding experiments show that clipping $\gamma^{(h)}$ to be outside of the convex corridor throughout training increases validation loss. We next ask whether fixing $\gamma^{(h)}$ to be a constant inside the convex corridor throughout training can already improve validation loss.

To investigate, we fixed $\delta^{(h)} = 1$ and $\gamma^{(h)} = \gamma_0$ for some $\gamma_0 \in \{-1.5, -0.75, -0.5, 0.5, 0.75, 1.5\}$ in every head of every layer, including layer 0, and only allowed the remaining model weights to be learned, so each head becomes $\gamma_0 I + P$.

Figure 7 compares the validation BPB between $\gamma_0 I + P$ and the standard attention P for each γ_0. It shows that even when $\delta^{(h)}$ is fixed, the two tested values inside the convex corridor ($\gamma_0 = -0.75$ and $\gamma_0 = -0.5$) still achieved lower validation BPB than attention. $\gamma_0 = 0$ is the same as attention,

but every other tested value of γ_0 outside the convex corridor resulted in higher validation loss than attention.

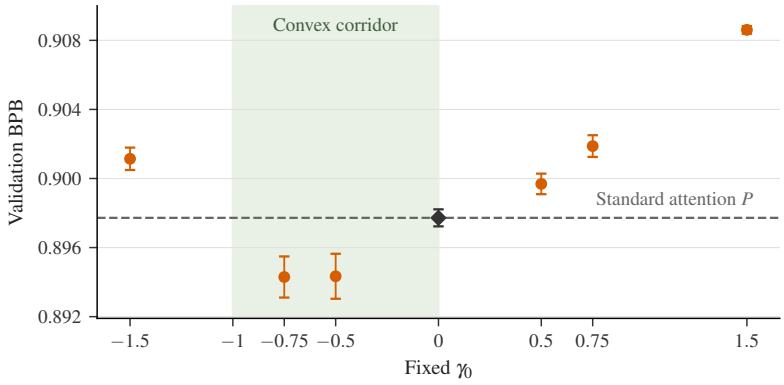


Figure 7: All heads use $\gamma_0 I + P$ for some fixed γ_0, with each point showing mean validation BPB. Even with $\delta^{(h)}$ fixed, constant values of γ_0 inside the convex corridor improve validation loss compared to attention, while the tested fixed values below -1 or above 0 yield higher validation loss than attention.

7.3 Learning $\delta^{(h)}$ across fixed $\gamma^{(h)}$ values

Next, we investigate whether allowing $\delta^{(h)}$ to learn when $\gamma^{(h)}$ is fixed can improve validation loss. We fixed $\gamma^{(h)} = \gamma_0$ for some $\gamma_0 \in \{-1.5, -1, -0.75, -0.5, 0, 0.5, 1\}$ in layers 1–11. We then compared the validation loss of fixing $\delta^{(h)} = 1$ with learning $\delta^{(h)}$. Comparisons at the same fixed γ_0 therefore isolate the effect of learning $\delta^{(h)}$ in layers 1–11 under a common layer-0 setup.

In all experiments, we allowed layer 0 to learn both $\gamma^{(h)}, \delta^{(h)}$ freely, with initialization $(\gamma^{(h)}, \delta^{(h)}) = (0, 1)$. In later layers, $\delta^{(h)}$ was also initialized to 1. We used learning rates $(\eta_\gamma, \eta_\delta) = (0.0025, 0.02)$.

Figure 8 shows that even when $\delta^{(h)}$ is learnable, values of γ_0 below -1 or above 0 fail to improve validation loss compared to attention. In contrast, values of γ_0 inside the convex corridor improve validation loss, regardless of whether $\delta^{(h)}$ is fixed or learnable. The endpoint $\gamma_0 = -1$ also improves over attention: it gives negative-Laplacian heads in layers 1–11, consistent with the benefit of Laplacian heads observed earlier.

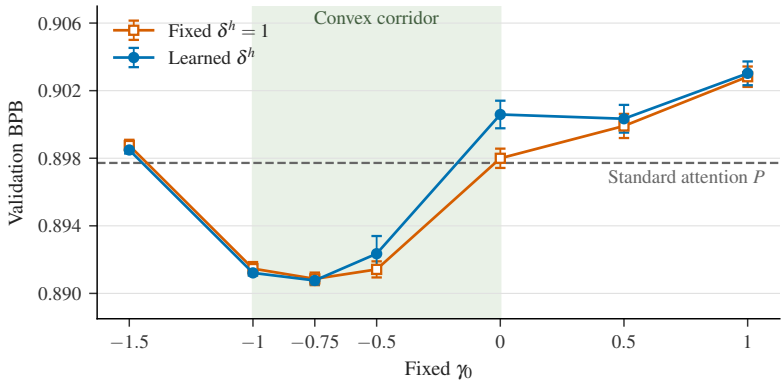


Figure 8: **Effects of learning or fixing $\delta^{(h)}$ when $\gamma^{(h)}$ is fixed at γ_0.** Each head in layers 1–11 uses $\delta^{(h)}(\gamma_0 I + P)$ with γ_0 fixed. The blue curve shows learned $\delta^{(h)}$, while the orange curve uses a fixed $\delta^{(h)} = 1$. Even when $\delta^{(h)}$ is learnable, values of γ_0 below -1 or above 0 performed worse than attention.

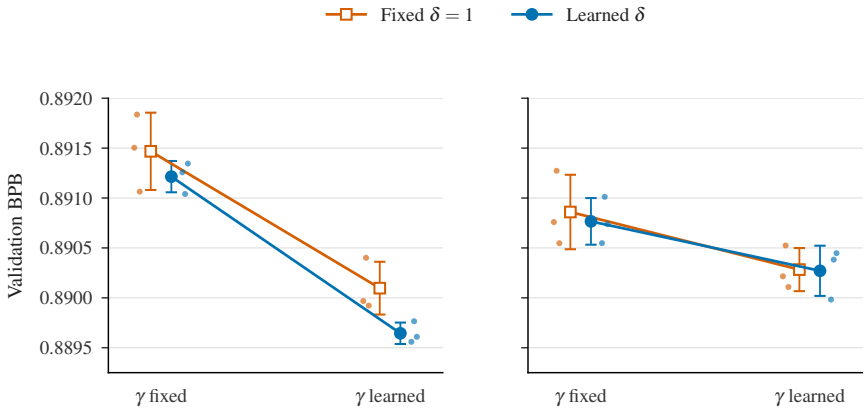
7.4 LEARNING BOTH $\gamma^{(h)}$ AND $\delta^{(h)}$

The preceding experiments suggest that requiring both $\gamma^{(h)}$ and $\delta^{(h)}$ to be learnable is not necessary to improve validation loss. To assess whether both $\gamma^{(h)}$ and $\delta^{(h)}$ should be learnable, we extend the experiments in Sections 7.2 and 7.3 to include:

- Learning only $\gamma^{(h)}$ in layers 1–11, with $\delta^{(h)} = 1$ fixed in layers 1–11.
- Learning both $\gamma^{(h)}$ and $\delta^{(h)}$ in layers 1–11, with $\delta^{(h)}$ initialized to 1 in all layers.

We allowed layer 0 to learn both $\gamma^{(h)}$ and $\delta^{(h)}$ under the same setup as in Section 7.3.

Figure 9 shows that when $\gamma^{(h)}$ is initialized at -1 in layers 1–11, learning both $\gamma^{(h)}$ and $\delta^{(h)}$ achieves lower validation loss than fixing either coefficient, consistently across seeds. The benefit of learning $\delta^{(h)}$ is larger when $\gamma^{(h)}$ is also learnable.



(a) Initial $\gamma_0 = -1$. (b) Initial $\gamma_0 = -0.75$.

Figure 9: **Effects of learning both $\gamma^{(h)}$ and $\delta^{(h)}$ on validation loss.** The panels compare two initializations for layers 1–11: $\gamma^{(h)} = -1$ (left) and $\gamma^{(h)} = -0.75$ (right). Each panel varies whether $\gamma^{(h)}$ and $\delta^{(h)}$ in layers 1–11 are fixed or learned. Large markers show mean validation BPB, error bars show sample standard deviations, and small dots show individual seeds.

When $\gamma^{(h)}$ is initialized at -0.75 in layers 1–11, learning $\gamma^{(h)}$ remains helpful, but learning both $\gamma^{(h)}$ and $\delta^{(h)}$ achieves nearly the same mean loss as learning $\gamma^{(h)}$ alone. These results indicate that the additional benefit of learning $\delta^{(h)}$ depends on initialization.

Overall, the results in Section 7 provide strong evidence that beyond the first layer, learning $\gamma^{(h)}$ in the convex corridor, thereby making each head a scaled convex combination of attention and negative Laplacian, contributes to improvements in validation loss, and this contribution cannot be replaced by learning $\delta^{(h)}$ alone.

8 CONCLUSION

We proposed parameterizing each head as $\delta(\gamma I + P)$ to learn the balance between the Laplacian and attention. This modification improves pretraining loss at both tested model sizes, adding only two parameters per head. Most heads beyond the first layer learn $\gamma \in (-1, 0)$, giving a scaled convex combination of attention and the negative Laplacian. Our intervention experiments provide evidence that this balance is not merely an artifact of training, but truly contributes to loss improvements.

These findings suggest that finer control over the balance between attention and the Laplacian can further improve transformers. The fundamental reason why training favors the convex corridor, and why the first layer behaves differently, remain open questions. Additionally, it would be interesting to measure the impact of the learned convex combinations on the geometry of token embeddings.

REPRODUCIBILITY STATEMENT

Appendix A describes the full details of the training recipe used in our experiments and contains the necessary information to reproduce our results. The premise of each experiment in the main text is defined in its corresponding section. Appendix B, C, D, E, F, and G also describe the details of the experiments within each section.

AI USAGE STATEMENT

We used LLMs to assist with code implementations and analysis of results, such as creating figures. We did not use LLMs to come up with the core hypothesis. Core experimental design was done by humans, and LLMs were only used to provide feedback on some details. Other required disclosures are not applicable to this work. LLMs were used to help with formatting the paper and conducting literature search. We take responsibility for the final content of this work, including anything generated with the aid of LLMs.

REFERENCES

Ameen Ali, Tomer Galanti, and Lior Wolf. Centered self-attention layers. In *Image Analysis: 23rd Scandinavian Conference, SCIA 2025*, volume 15725 of *Lecture Notes in Computer Science*, pp. 60–73. Springer, 2025. doi: 10.1007/978-3-031-95911-0_5. URL https://arxiv.org/abs/2306.01610.

Linwei Chen, Lin Gu, and Ying Fu. Frequency-dynamic attention modulation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22620–22632, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Chen_Frequency-Dynamic_Attention_Modulation_For_Dense_Prediction_ICCV_2025_paper.html.

Jeongwhan Choi, Hyowon Wi, Jayoung Kim, Yehjin Shin, Kookjin Lee, Nathaniel Trask, and Noseong Park. Graph convolutions enrich the self-attention in transformers! In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2312.04234.

Bobby He and Thomas Hofmann. Simplifying transformer blocks. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2311.01906.

Bobby He, James Martens, Guodong Zhang, Aleksandar Botev, Andrew Brock, Samuel L. Smith, and Yee Whye Teh. Deep transformers without shortcuts: Modifying self-attention for faithful signal propagation. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2302.10322.

Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. Understanding and minimising outlier features in transformer training. In *Advances in Neural Information Processing Systems*, volume 37, pp. 83786–83846, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/986292a930c3692168b177a770025ab3-Abstract-Conference.html.

Peng He, Yao Liu, Yanglei Gan, Run Lin, Yuxiang Cai, and Qiao Liu. FAiT: Frequency-aware inverted transformer for multivariate time series forecasting. *arXiv preprint arXiv:2606.01306*, 2026. URL https://arxiv.org/abs/2606.01306.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Andrej Karpathy. nanochat: The best ChatGPT that $100 can buy, 2025. URL https://github.com/karpathy/nanochat.

Jayoung Kim, Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. Polynomial-based self-attention for table representation learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning*

Research, pp. 24509–24526. PMLR, 2024. URL https://proceedings.mlr.press/v235/kim24ae.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

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

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. FineWeb-Edu: the finest collection of educational content, 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu.

Tam Nguyen, Tan M. Nguyen, and Richard G. Baraniuk. Mitigating over-smoothing in transformers via regularized nonlocal functionals. In *Advances in Neural Information Processing Systems*, volume 36, pp. 80233–80256, 2023. URL https://arxiv.org/abs/2312.00751.

Lorenzo Noci, Chuning Li, Mufan Li, Bobby He, Thomas Hofmann, Chris J. Maddison, and Daniel M. Roy. The shaped transformer: Attention models in the infinite depth-and-width limit. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.17759.

Aditya Ravuri and Neil D. Lawrence. Transformers as unrolled inference in probabilistic Laplacian Eigenmaps. In *Proceedings of UniReps: the Third Edition of the Workshop on Unifying Representations in Neural Models*, volume 322 of *Proceedings of Machine Learning Research*, pp. 159–167. PMLR, 2026. URL https://proceedings.mlr.press/v322/ravuri26a.html.

David So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V. Le. Searching for efficient transformers for language modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 6010–6022, 2021. URL https://proceedings.neurips.cc/paper/2021/file/2f3c6a4cd8af177f6456e7e51a916ff3-Paper.pdf.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

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*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the Fourier domain analysis: From theory to practice. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2203.05962.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10524–10533. PMLR, 2020. URL https://proceedings.mlr.press/v119/xiong20b.html.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.05258.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/1e8a19426224ca89e83cef47f1e7f53b-Abstract.html.

Yuchong Zhang and Vardan Papyan. Laplacian heads improve transformers by smoothing token representations. *arXiv preprint arXiv:2602.09297*, 2026. URL https://arxiv.org/abs/2602.09297v3.

Zhanchao Zhou, Tianyi Wu, Zhiyun Jiang, Fares Obeid, and Zhenzhong Lan. Value residual learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28341–28356. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1375. URL https://aclanthology.org/2025.acl-long.1375/.

A PRETRAINING RECIPE

Tables 1 and 2 summarize the `nanochat` (Karpathy, 2025) configuration used in our experiments.

Architecture	d12	d20
Layers / model width / heads	$12/768/6$	$20/1280/10$
Baseline parameters	185.6M	561.0M
Learned mixture coefficients	144	400
Attention windows	Full context	SSSL
Context length / vocabulary	2,048 positions / 65,536 tokens	
Normalization	RMSNorm (Zhang & Sennrich, 2019) before each sublayer, and on embeddings and final hidden states	
Position encoding	RoPE (Su et al., 2021), base 100,000; normalized queries and keys scaled by 1.2	
Feedforward network	Width $4d$, squared-ReLU activation (So et al., 2021)	
Embedding / output weights	Untied	
Residual branch initialization	Zero attention and feedforward output projections	
Biases / dropout	None	
Logit cap	$15\tanh(z/15)$	

Table 1: Model architecture. Parameter counts include embeddings. SSSL repeats three layers with a 512-position local attention window, followed by one full-context layer.

Training setting	d12	d20
Tokens per update	524,288	1,048,576
Training tokens	2.097B	5.725B
Peak matrix LR (Muon)	0.02	0.028284
Peak embedding LR (AdamW)	0.3	0.328634
Peak output LR (AdamW)	0.008	0.008764
Peak $(\eta_\gamma, \eta_\delta)$ (AdamW)	$(0.0025, 0.02)$	$(0.003536, 0.028284)$
Initial matrix weight decay	0.28	0.112267
Data	FineWeb-Edu; documents packed into fixed-length sequences	
LR schedule	40-update warmup; constant to 35% of the schedule; linear decay to 5% of peak	
Mixture initialization (γ, δ)	$(0, 1)$ in layer 0; $(-1, 1)$ in later layers	
Precision	BF16 activations; FP32 matrix master weights and mixture coefficients	
Validation	41.94M held-out token positions	

Table 2: Training recipe for Figure 1.

B COEFFICIENT EVOLUTION ACROSS INITIALIZATIONS

Additional trajectories. Figures 10 and 11 extend Section 4 to four initializations: $\gamma \in \{-1.5, -0.5, 0, 1.5\}$, with $\delta = 1$ in every head. Unlike the default initialization in the main text, these runs use the same initial γ in layer 0 and later layers.

Distribution at the end of training. Table 3 combines the unrestricted d12 runs under the three parameterizations. For the proposal, it includes all 417 distinct runs in the completed initialization and learning-rate screens. These include uniform starts, starts with $\gamma = 0$ only in layer 0, and three runs with initial layer-0 $\delta = 2$. Fixed, clipped, and shared-coordinate controls are excluded. The 400 and 240 nonzero-initialization runs from the grids in Appendix F are included.

For $\delta(\gamma I + P)$, 26,693 of 27,522 later-layer heads (97.0%) end inside $(-1, 0)$, and every run has a majority of later-layer heads inside this interval. The two alternative parameterizations show the same concentration. Layer 0 has a different, more variable distribution, so we report it separately.

We form $\tilde{\gamma}$ separately for each head using Equations 2 and 3, without clipping or adding an epsilon to the denominator. All ratios in the table are finite. Percentages are across heads.

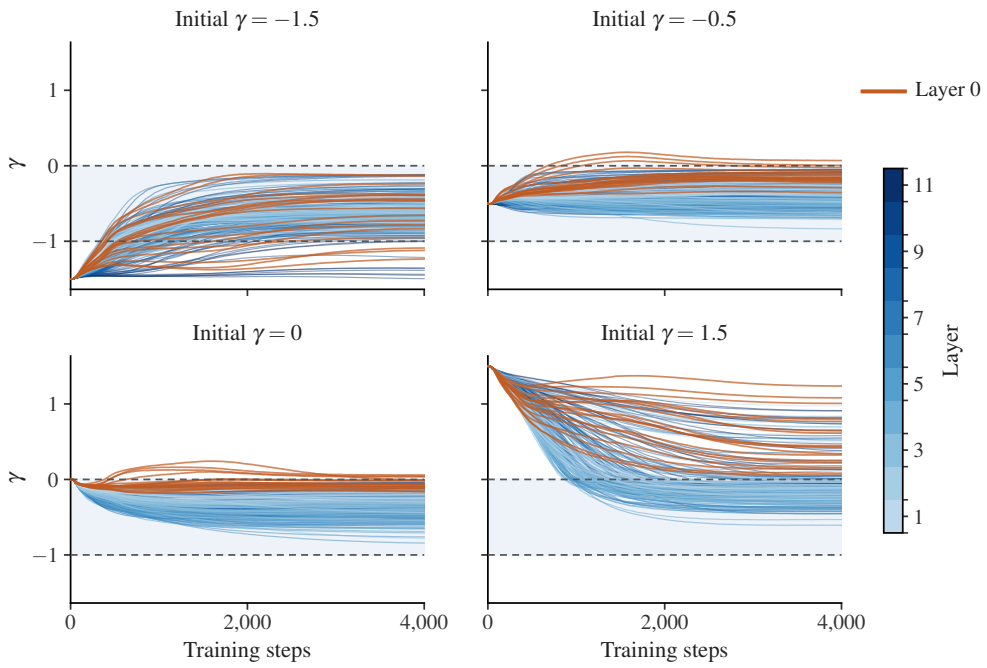


Figure 10: Evolution of γ from four additional initializations. Each panel shows all six heads in all twelve layers for three seeds. Orange denotes layer 0, and blue shades denote later layers. The shaded interval is the convex corridor $(-1, 0)$.

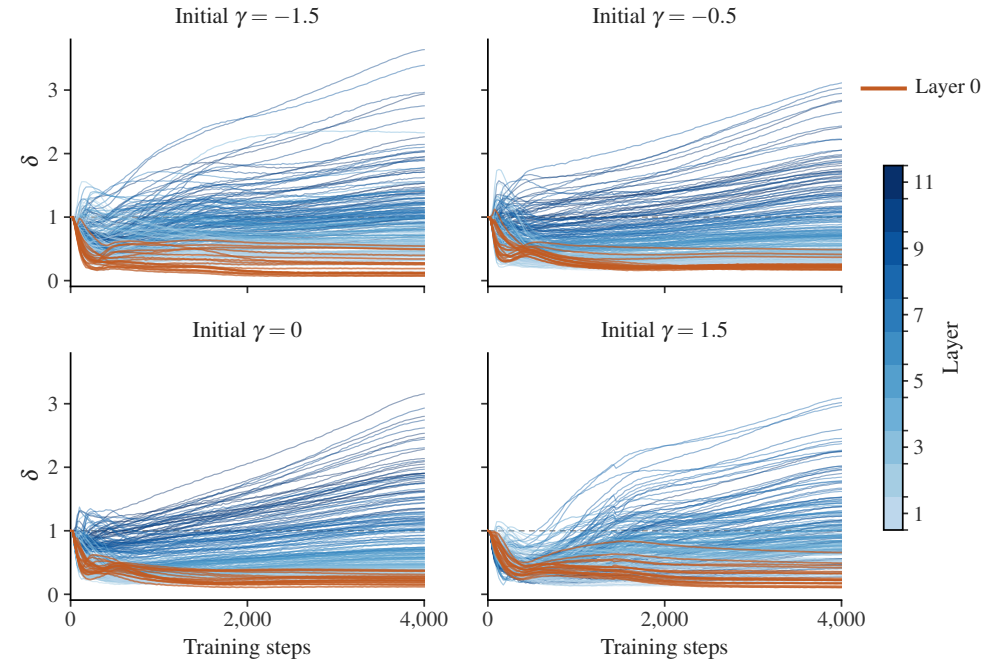


Figure 11: Evolution of δ for the same runs as Figure 10. Each curve is one head and seed. Colors identify layers.

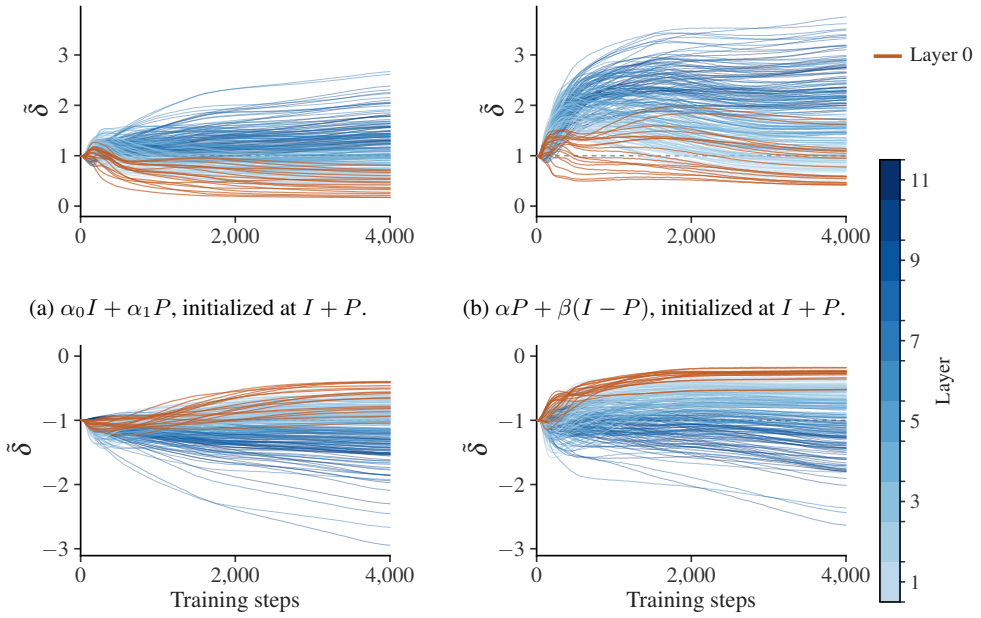
C δ TRAJECTORIES UNDER ALTERNATIVE PARAMETERIZATIONS

Figure 12 supplies the $\tilde{\delta}$ trajectories referred to in Section 5. It uses the same twelve d12 runs as Figure 4: two parameterizations, initializations $I + P$ and $I - P$, and seeds 0–2. Both learned

Table 3: Final per-head γ at update 4,000. Each row pools all heads in the stated layers. The eight zero-operator runs (576 heads) have undefined ratios and are excluded.

Parameterization	Layers	Heads	Median γ	$\gamma \le -1$ (%)	$-1 < \gamma < 0$ (%)	$\gamma \ge 0$ (%)
$\delta(\gamma I + P)$	0	2,502	-0.040	0.12	68.39	31.49
	1–11	27,522	-0.421	0.78	96.99	2.23
$\alpha_0 I + \alpha_1 P$	0	2,400	-0.023	2.42	55.83	41.75
	1–11	26,400	-0.379	0.89	97.31	1.80
$\alpha P + \beta(I - P)$	0	1,440	-0.066	6.46	65.90	27.64
	1–11	15,840	-0.409	0.76	98.35	0.88

coefficients use peak learning rate 0.005. We plot $\tilde{\delta} = \alpha_1$ for $\alpha_0 I + \alpha_1 P$ and $\tilde{\delta} = \alpha - \beta$ for $\alpha P + \beta(I - P)$.



(a) $\alpha_0 I + \alpha_1 P$, initialized at $I + P$.

(b) $\alpha P + \beta(I - P)$, initialized at $I + P$.

(c) $\alpha_0 I + \alpha_1 P$, initialized at $I - P$.

(d) $\alpha P + \beta(I - P)$, initialized at $I - P$.

Figure 12: Signed $\tilde{\delta}$ trajectories for the runs in Figure 4. Each panel contains all 216 head/seed curves, recorded every 25 updates. Columns identify the parameterization; rows identify the initial operator. Orange denotes layer 0 and blue shades denote layers 1–11. Deeper layers tend to learn larger $|\tilde{\delta}|$ under both starts.

D Initialization and learning-rate sensitivity of the proposal

We varied initialization and coefficient learning rates while keeping the d12 recipe fixed. Every head starts with $\delta = 1$, and layer 0 starts with $\gamma = 0$. In layers 1–11, we tested $\gamma \in \{-1.5, -1, -0.5, 0.5, 1, 1.5\}$. For each start, we tested all 16 peak learning-rate pairs from $\{0.0025, 0.005, 0.01, 0.02\}^2$, with seeds 0–2. Both coefficients then learn freely in each head.

Figure 13 shows that all 96 configurations improve upon standard attention. Initializing later layers at $\gamma = -1$ gives the lowest loss throughout the tested grid: even its worst mean BPB (0.890338) is below the best mean for any other tested initialization (0.890683). Learning rates still matter, especially for positive initializations. For example, the mean BPB range is 0.891304–0.895307 at initial $\gamma = 1.5$, compared with 0.889641–0.890338 at $\gamma = -1$.

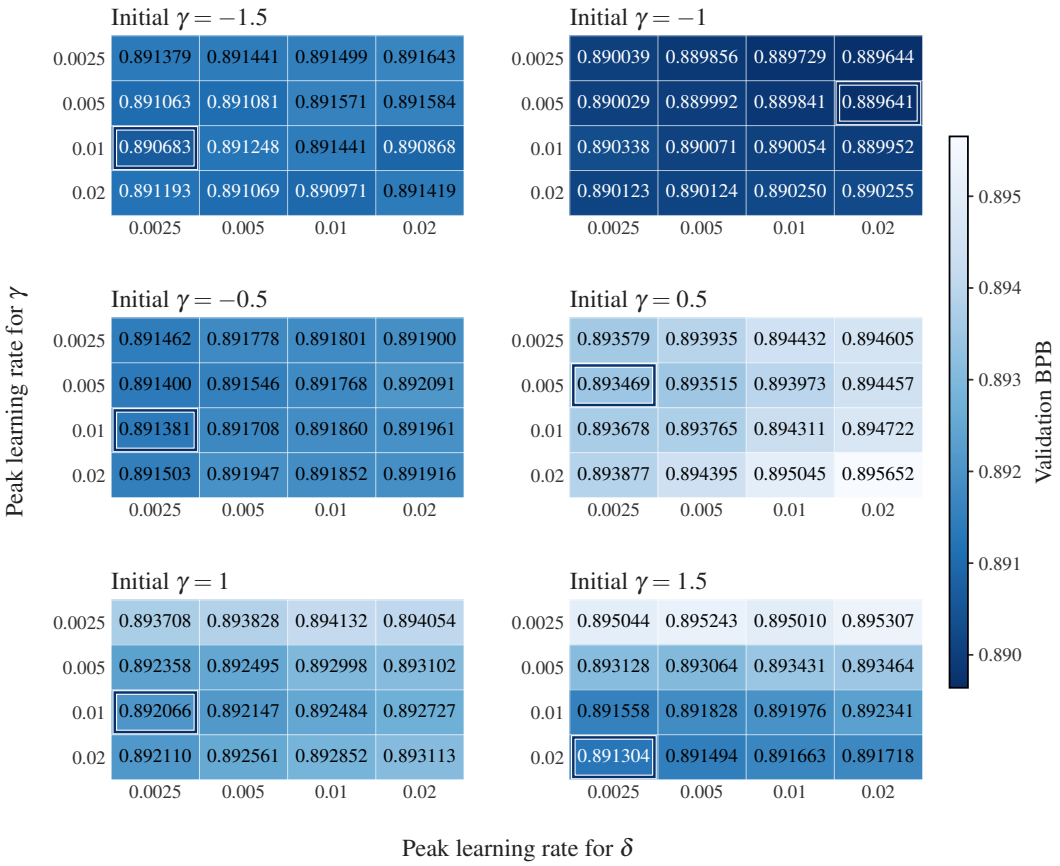


Figure 13: Initialization and learning rates both affect validation BPB. All tested settings improve over standard attention (mean BPB 0.897721). Each panel uses the indicated initial γ in layers 1–11; cells show three-seed means. Outlined cells mark the lowest full-precision mean in each panel.

E MIXING FLEXIBILITY

Does each head need its own mixing coefficients? We independently varied the sharing of γ and δ in layers 1–11: one scalar across all later heads, one per layer, or one per head. Crossing these choices gives nine policies. Layer 0 always retains separate γ and δ for each head, initialized at $(0, 1)$; later heads start at $(-1, 1)$.

We tested each policy at peak learning rates $(0.0025, 0.02)$ and $(0.005, 0.005)$, across 3 seeds. For a shared coordinate, gradients are summed over its heads before a single Adam update (Kingma & Ba, 2015). Each shared scalar has its own Adam state.

Table 4: Sharing γ and δ across layers 1–11. G shares one scalar globally, L uses one per layer, and H uses one per head. Layer 0 retains separate per-head coefficients, included in the scalar count. Entries are validation BPB (mean $\pm$ sample SD over three seeds).

γ	δ	Scalars	$(\eta_\gamma, \eta_\delta) = (0.0025, 0.02)$	$(0.005, 0.005)$
G	G	14	0.89045 ± 0.00042	0.89051 ± 0.00042
G	L	24	0.89013 ± 0.00048	0.89029 ± 0.00035
G	H	79	0.89006 ± 0.00032	0.89030 ± 0.00046
L	G	24	0.89021 ± 0.00027	0.89041 ± 0.00034
L	L	34	0.88993 ± 0.00022	0.89026 ± 0.00035
L	H	89	0.88984 ± 0.00036	0.89027 ± 0.00032
H	G	79	0.89017 ± 0.00033	0.89029 ± 0.00014
H	L	89	0.88979 ± 0.00012	0.89001 ± 0.00009
H	H	144	0.88964 ± 0.00011	0.88999 ± 0.00026

Table 4 shows that all nine policies improve over attention (mean BPB 0.897721). Independent coefficients per head give the lowest mean at both learning-rate pairs. However, much of the im-

provement remains with less freedom: at $(0.0025, 0.02)$, sharing both coordinates by layer gives 0.889927 BPB, compared with 0.889644 for separate coordinates per head. Even sharing both globally gives 0.890449.

Thus, independent mixing in every head is helpful but is not required to improve on attention. The small differences between some policies do not establish equal performance; this comparison uses one initialization and two learning-rate pairs.

F Performance of alternative parameterizations

Initialization grids. We tested $\alpha_0 I + \alpha_1 P$ and $\alpha P + \beta(I - P)$ on the d12 recipe, initializing each learned coefficient on $\{-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2\}$. Every head shares the stated initialization and then learns independently. Both coordinates use peak learning rate 0.005. The $\alpha_0 I + \alpha_1 P$ grid uses 5 seeds, and the $\alpha P + \beta(I - P)$ grid uses 3 seeds.

Figure 14 shows the change in final validation BPB relative to attention. Each parameterization improves for 78 of its 80 nonzero starts, but the improvement depends on initialization. In particular, starts close to a small multiple of pure attention can give little improvement. The zero operator is a separate failure: with both mixing coefficients and the attention output projection initialized to zero, the attention branch receives no gradient and remains zero.

Learning-rate sensitivity. An initialization that performs poorly can be improved with appropriate learning rates. For $\alpha_0 I + \alpha_1 P$ initialized at $(2, 0.5)$, changing peak rates from $(0.005, 0.005)$ to $(0.02, 0.005)$ reduces the mean gap to the $(1, 1)$ reference from 0.002965 to -0.000349 BPB on reserved text and fresh seeds (Figure 15a).

For the $(0, 1)$ initialization, selecting $(0.04, 0.0025)$ from the 18 tested rate pairs reduces the gap from 0.004887 to 0.003358 BPB (Figure 15b). This second comparison selects and reports on the same validation text and 3 seeds. It shows partial recovery within the tested grid.

The $\alpha P + \beta(I - P)$ grid tests initialization at one rate pair and does not provide a comparable learning-rate sweep. Although both parameterizations describe the same linear operators, changing coordinates changes Adam's updates. Their initialization grids therefore support broad improvements under both choices, without establishing that either has the best attainable loss after tuning.

G Additional intervention experiments

G.1 Restricting alternative parameterizations

We tested whether the cost of excluding negative γ persists under other coordinates. For $\gamma I + P$, we initialized $\gamma = 1$ in every head and compared free learning with clipping $\gamma \geq 0$ in layers 1–11. Layer 0 remained free. Both settings used the same four peak learning rates, $\{0.0025, 0.005, 0.01, 0.02\}$, and 3 seeds.

For $\alpha_0 I + \alpha_1 P$, all heads started at $(0, 1)$. We compared free learning (F), fixing $\alpha_0 = 0$ while learning α_1 (A), and requiring $\alpha_0 \alpha_1 \geq 0$ (N). For N, an update proposing opposite signs has its smaller-magnitude coordinate set to zero; ties set $\alpha_0 = 0$. Both restrictions exclude negative $\tilde{\gamma}$ wherever the ratio is defined. Projections follow each Adam update and retain its moments. Choosing the layer-0 and later-layer conditions independently gives nine combinations, and each was tested at the same 18 learning-rate pairs and 3 seeds. Writing the rates as $0.005(m_0, m_1)$, the tested pairs are

$$\{0.5, 1, 2, 4, 8\} \times \{0.25, 0.5, 1\} \cup \{(0.5, 2), (1, 2), (1, 4)\}.$$

Figure 16 shows that restricting later layers increases loss under both parameterizations. For $\gamma I + P$, this holds at every matched rate. For the direct coefficients, it holds after selecting the best tested rate pair for each condition. Restricting layer 0 does not have the same consistent cost.

We then fixed the selected policies and evaluated new seeds 10–12 on reserved text. Table 5 confirms the loss increase from restricting later layers. The layer-0 comparison remains inconclusive. This

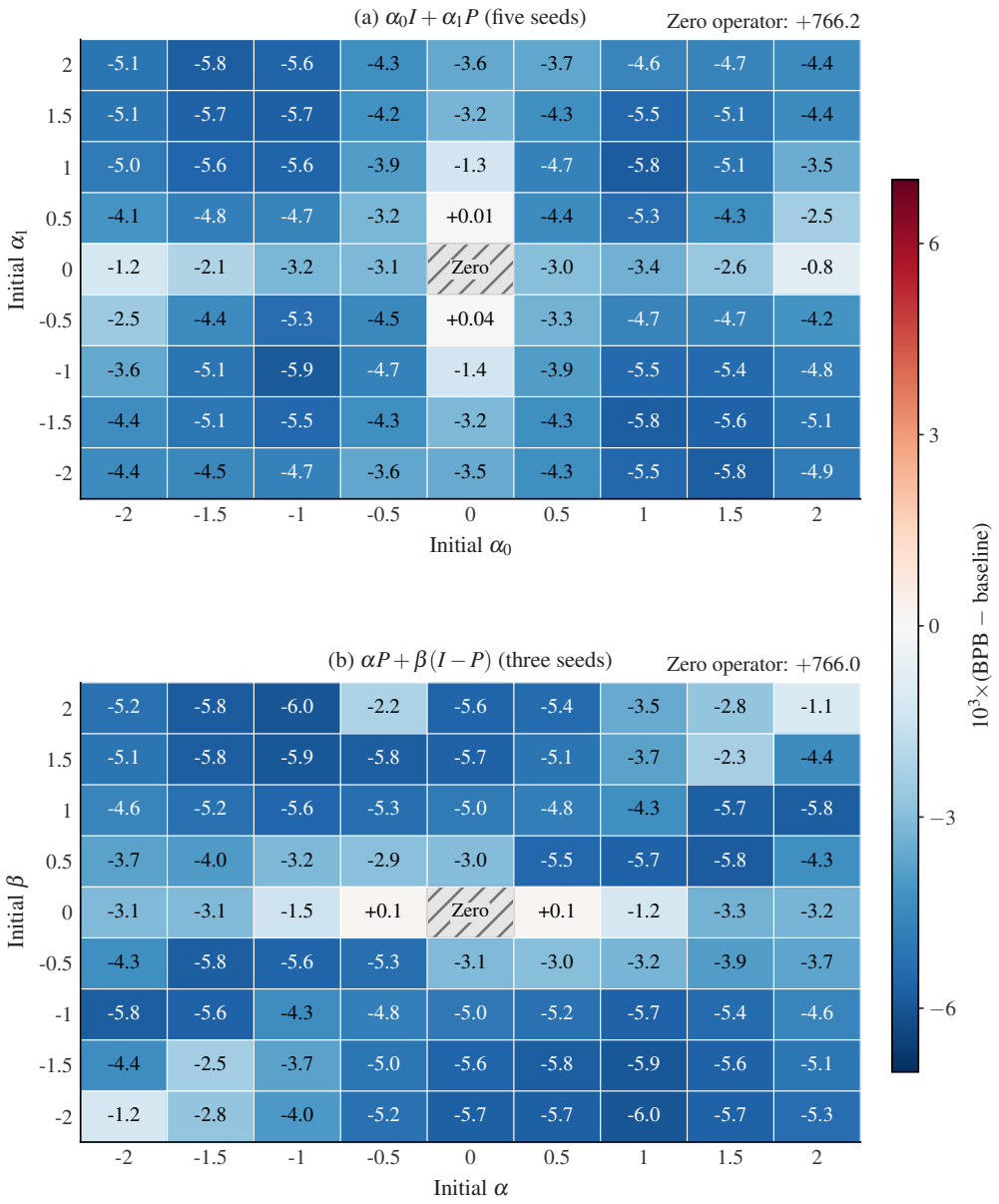


Figure 14: Initialization sensitivity of both alternative parameterizations. Each cell shows 10^3 times the mean run-minus-attention BPB difference at update 4,000, subtracting the baseline with the same seed label. Negative values favor the learned mixture. Both panels use the same color scale; outlined cells are worse than attention. The zero-operator cell is gray, with its value stated above each panel on the same $10^3\Delta$BPB scale. Axes give the native learned coefficients, so corresponding cells across panels need not represent the same initial operator.

strengthens the evidence that the later-layer effect extends beyond the proposed coordinates and the original tuning seeds.

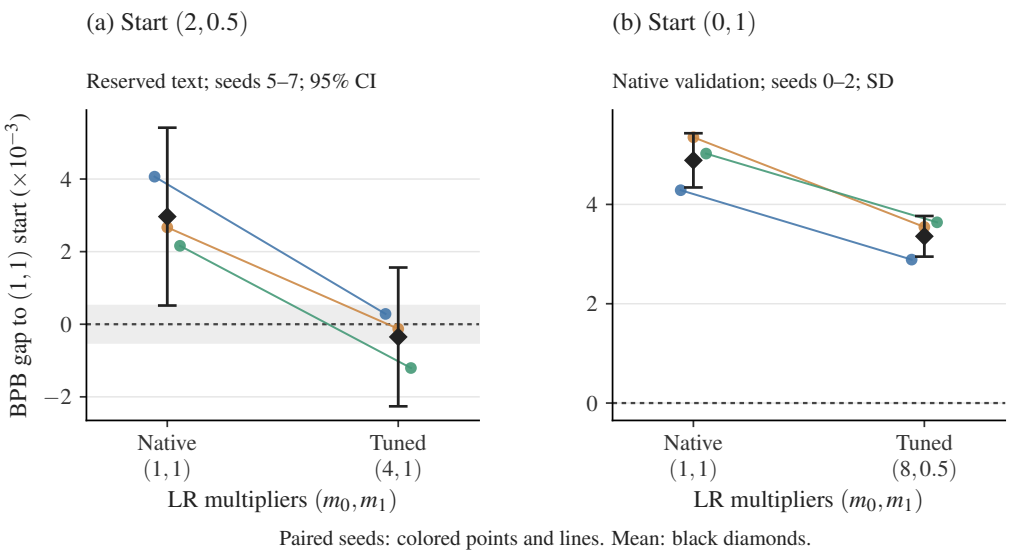


Figure 15: LR tuning reduces both initialization gaps; the $(0, 1)$ gap remains. Gaps compare with $(1, 1)$ initialization at native rates on each panel's evaluation text. Points show paired seeds and diamonds their mean. Bars are 95% paired t intervals in (a) and sample seed SD in (b). The gray band is ± 0.0005 BPB; (a) does not establish recovery within the predefined confidence bound.

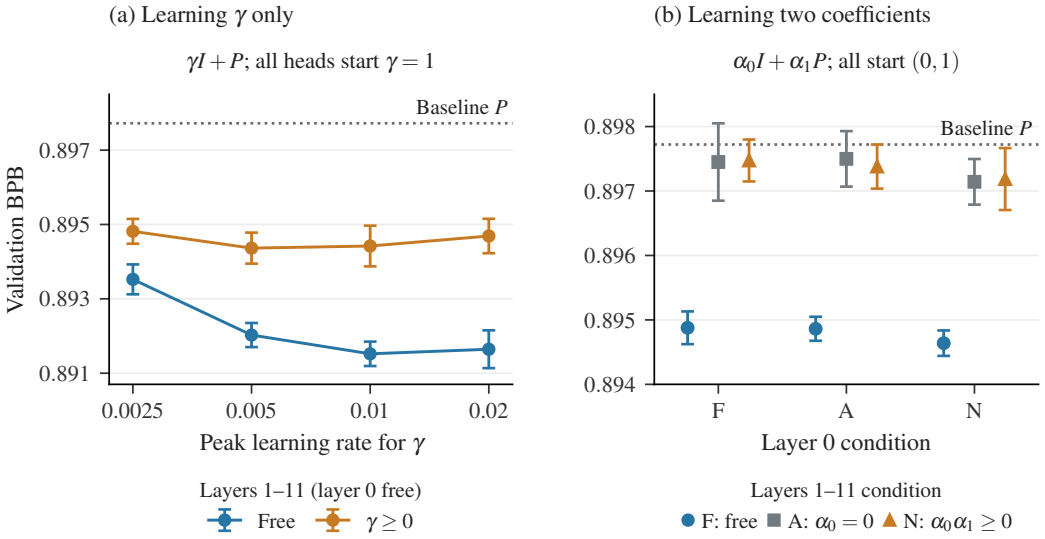


Figure 16: Restricting the coefficients in layers 1–11 increases validation BPB under both parameterizations. (a) For $\gamma I + P$, learning γ freely gives lower BPB than constraining $\gamma \geq 0$ at every tested learning rate; both settings improve over standard attention. Layer 0 learns freely in both cases. (b) Learning both coefficients freely (F) gives lower BPB than fixing $\alpha_0 = 0$ (A) or requiring $\alpha_0\alpha_1 \geq 0$ (N) in layers 1–11, for every condition in layer 0. Each combination uses the best of the same 18 learning-rate pairs. Points show means and error bars show sample standard deviations across three seeds; dotted lines mark standard attention P.

Table 5: Independent restriction confirmation. Later-layer restrictions increase BPB on reserved text. Differences pair seeds 10–12 under policies selected before confirmation; the layer-0 contrast remains inconclusive.

Comparison	Mean ΔBPB	95% paired t interval
Direct: later attention vs free	+0.002080	[+0.001104, +0.003056]
Direct: later no opposite signs vs free	+0.002242	[+0.001503, +0.002980]
Direct: layer-0 attention vs free	−0.000025	[−0.000229, +0.000179]
$\gamma I + P$: later nonnegative vs free	+0.003122	[+0.002207, +0.004036]