BEYOND 4D FFNS: HESSIAN POTENTIAL MLP FOR MEMORY AND LANGUAGE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Position-wise feed-forward networks (FFNs) dominate Transformer cost, yet remain standard two-layer MLPs expanding to width $4d$ before linear projection. We introduce the Hessian Feed-Forward Network (H-FFN), H-FFN$(x) = \nabla^2\Psi(x)\,x = \sum_j \lambda_j(x)\,(u_j^\top x)\,u_j$, expressing the sublayer as a sum of rank-one directional units derived from a scalar potential. For associative retrieval, a coaxial Memory-Track employs multi-scale contextual gating; for language modeling, Decoupled H-FFN separates content, routing, and emission. Under matched training budgets, Memory-Track reaches $96.5 \pm 0.7\%$ Needle-in-a-Haystack accuracy versus $75.7 \pm 5.8\%$ for a matched MLP, and Decoupled H-FFN achieves 1.06–$1.38\times$ perplexity gains across causal language modeling benchmarks ($1.17\times$ on WikiText-103; $\approx 1.15\times$ on FineWeb-Edu). In transfer, analytic GELU matching improves GPT-2 fine-tuning ($1.07\times$) and accelerates ViT-B/16 convergence ($+2.4\%$ top-1 with $3\times$ fewer steps).

1 INTRODUCTION

The Transformer architecture (Vaswani et al., 2017) remains the backbone of large language and multimodal models. Each block interleaves multi-head self-attention with a position-wise feed-forward network (FFN). In the canonical design the FFN is a two-layer multi-layer perceptron (MLP) that expands the hidden state and linearly reconstructs it at model dimension, FFN$(x) = \sigma(xW_1)W_2$, with intermediate width typically $d_{\text{ff}} = 4\,d_{\text{model}}$ and elementwise nonlinearity σ. This layer dominates cost: FFNs account for over 60% of parameters and FLOPs in standard Transformers (Wei et al., 2024), often outnumbering the attention sublayer by an 8:3 parameter ratio (Gerber, 2025). The same dense map is applied to every token, which is a known bandwidth constraint at decode time (Vaswani et al., 2017; Wei et al., 2024).

Efforts to improve FFNs have largely targeted the nonlinearity and multiplicative gates. Smooth activations such as GELU (Hendrycks & Gimpel, 2016) and Swish (Ramachandran et al., 2017) reduced inactive units relative to ReLU. Gated linear units (Shazeer, 2020) introduced a multiplicative branch, as in SwiGLU in PaLM (Chowdhery et al., 2023) and LLaMA (Touvron et al., 2023). The surrounding map remains a single path through a fixed intermediate width. Pires et al. (2023) and Csordás et al. (2023) further document cross-layer redundancy, with every token still forced through that same intermediate space.

A complementary view treats FFNs as key-value associative memories (Geva et al., 2021), where the first matrix detects patterns and the second emits outputs. This creates an operational dilemma: the FFN must simultaneously transform hidden states and store long-term facts without an explicit index, leading to representational interference. While Mixture-of-Experts (MoE) (Lepikhin et al., 2021; Fedus et al., 2022) and low-rank or structured matrix approximations (Csordás et al., 2023; Wei et al., 2024) scale capacity or reduce inference bandwidth, they still rely on static, unindexed projections.

Work on the attention sublayer has connected sequence mixing to linear kernels (Katharopoulos et al., 2020; Choromanski et al., 2021), cosine geometries (Qin et al., 2022; Mongaras et al., 2024), and continuous Modern Hopfield networks (Ramsauer et al., 2021; Hoover et al., 2023; Hu et al., 2024). The feed-forward sublayer has fewer such derivations. Second-order optimization further shows that dense layers often traverse ill-conditioned Hessian surfaces when the forward map is a static projection (Martens & Grosse, 2015).

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.

These observations point to three structural limits of the standard FFN: a static dense map, a single unindexed intermediate width, and associative storage without an explicit key. Rather than further tuning that width, the feed-forward sublayer can be written as a curvature operator derived from a scalar potential.

We propose the Hessian Feed-Forward Network (H-FFN). Each rank-one unit projects the hidden state x onto a coordinate $s_j = u_j^\top x + b_j$, gates the projection with localized curvature $\lambda_j = \psi''(s_j)$, and emits along the same axis, at the same asymptotic cost as a two-layer MLP. A coaxial Memory-Track adds multi-scale context gating for associative retrieval; decoupled variants separate content, routing, and emission for language modeling.

Scratch and supplementary evaluations use native signed curvature from $\psi = \tanh^2$; only pretrained fine-tuning (Section 3.3) uses the analytic warm-start. Under matched schedules, Memory-Track raises Needle-in-a-Haystack accuracy from $75.7 \pm 5.8\%$ to $96.5 \pm 0.7\%$, and channel-split H-FFN lowers WikiText-103 test perplexity from 455.3 ± 7.6 to 388.0 ± 2.3. Sections 4 and 5 then analyze the mechanism in trained coordinates and consolidate supplementary probes across additional corpora, model widths, and training budgets.

2 METHODOLOGY

Standard feed-forward networks apply a fixed linear map and an elementwise nonlinearity, without reference to local geometry of the hidden state. Applying the Hessian of a scalar potential, $\mathcal{H}(x)x = \nabla^2\Psi(x)\,x$, would give input-dependent second-order modulation, but an unconstrained dense Hessian is $O(d^3)$ per token and $O(d^2)$ in memory. H-FFN writes Ψ as a sum of one-dimensional projections so that $\mathcal{H}(x)$ is a sum of rank-one outer products. Contracting with x then costs $O(md)$ inner products and scalings, matching a two-layer MLP while making curvature explicit.

2.1 HESSIAN FORMULATION FOR FEED-FORWARD LAYERS

To realize this operator, we parameterize the scalar potential $\Psi(x) : \mathbb{R}^d \to \mathbb{R}$ as a sum over a dictionary of m spectral units:

$$\Psi(x) = \sum_{j=1}^{m} \psi(s_j), \qquad s_j \triangleq u_j^\top x + b_j, \tag{1}$$

where $u_j \in \mathbb{R}^d$ denotes a learnable projection vector forming the j-th row of weight matrix $U \in \mathbb{R}^{m\times d}$, $b_j \in \mathbb{R}$ is a learnable scalar bias, and $s_j \in \mathbb{R}$ is the affine coordinate of x along projection axis u_j. The kernel $\psi : \mathbb{R} \to \mathbb{R}$ is a smooth, twice-differentiable scalar potential function.

Differentiating $\Psi(x)$ with respect to x via the chain rule ($\nabla_x s_j = u_j$) yields the gradient and Hessian:

$$\nabla\Psi(x) = \sum_{j=1}^{m} \psi'(s_j)u_j = U^\top \psi'(s), \tag{2}$$

$$\mathcal{H}(x) = \nabla^2\Psi(x) = \sum_{j=1}^{m} \psi''(s_j)\, u_j u_j^\top = U^\top \operatorname{diag}\big(\lambda(x)\big)U, \tag{3}$$

where $\lambda_j(x) = \psi''(s_j)$ denotes the curvature gate along coordinate u_j. The core H-FFN transformation evaluates the contraction of the input vector x with its local Hessian operator:

$$\text{H-FFN}(x) = \mathcal{H}(x)x = \sum_{j=1}^{m} \lambda_j(x)(u_j^\top x)u_j. \tag{4}$$

Geometrically, Equation equation 4 decomposes the transformation into m rank-one directional projectors $u_j u_j^\top$. The input x is projected onto coordinate axis u_j, scaled by its local second-order curvature $\lambda_j(x) = \psi''(s_j)$, and emitted along the same direction. In neural implementations, the linear contraction is typically evaluated over the affine coordinate $s_j = u_j^\top x + b_j$ (with an optional

output bias b_{out}), maintaining parameter compatibility with standard MLPs while preserving the directional second-order curvature.

The choice of potential kernel ψ dictates the resulting curvature landscape. In general, any twice-differentiable scalar potential $\psi : \mathbb{R} \to \mathbb{R}$ generates a state-dependent gating operator through its second derivative $\lambda_j(x) = \psi''(s_j)$. While canonical activations apply monotonic scalar projections, the potential formulation allows designing localized curvature profiles. Monotonic convex potentials (such as quadratic $\psi(s) = \frac{1}{2}s^2$ with constant $\psi'' = 1$) yield strictly non-negative second derivatives, functioning purely as positive amplification factors. To achieve selective subspace excitation alongside active inhibition without numerical instability, the kernel must yield a bounded, signed second derivative that decays smoothly to zero for extreme coordinates, preventing outlier activations from destabilizing representation dynamics. We instantiate this formulation with $\psi(s) = \tanh^2(s)$, which naturally satisfies these geometric criteria.[1]

To capture multi-resolution curvature across diverse feature magnitudes without relying on a single localized scale, we construct a Multi-Scale Curvature Ladder:

$$\lambda_j^{\text{ladder}}(s_j) = \sum_{k=0}^{K} w_k \, \psi'' \left(\frac{s_j}{2^k} \right), \qquad w_k = 2^{-k}, \tag{5}$$

where $K = 3$ (four geometric dyadic scales) by default. Dyadic scaling widens the effective receptive band across activation magnitudes, allowing the layer to simultaneously modulate fine-grained local variations and broader macroscopic feature shifts throughout deep Transformer stacks.

While canonical Transformers enforce intermediate capacity via a fixed expansion width $d_{\text{ff}} = 4d$, H-FFN capacity is governed by the dictionary size m. Because $\mathcal{H}(x)$ is assembled from rank-one outer products, expanding m beyond $4d$ (such as $m = 6d$) directly broadens the directional basis without altering sequence length or increasing attention complexity.

2.2 Directional Decoupling and the Gating Coordinate Dictionary

In sequential modeling, token representations depend on temporal context. Evaluating curvature purely from instantaneous state x_t restricts gating to local token magnitude. To incorporate broader context, we condition curvature on an aggregated state vector $z_t \in \mathbb{R}^d$ computed via Multi-Scale Prefix Aggregation (Prefix-MS):

$$z_t = \sum_{k=1}^{K} \omega_k \, \bar{x}_{t,W_k} + \omega_{\text{global}} \left(\frac{1}{t} \sum_{i=1}^{t} x_i \right), \tag{6}$$

where $\bar{x}_{t,W_k} = \frac{1}{|W_k|} \sum_{i \in W_k(t)} x_i$ averages over causal sliding windows of length W_k, and $\{\omega_k\}$ are learned mixture weights.

In encoder architectures and associative memory retrieval, representations operate in a shared coordinate system where queries and keys recall patterns in the same space. Here, preserving coaxial orientation ($u_j u_j^\top$) between input projection and output emission stabilizes representation geometry:

$$\Phi_{\text{mem}}(x_t; z_t) = \sum_{j=1}^{m} \lambda_j(z_t)(u_j^\top x_t) u_j, \tag{7}$$

[1] Analytically, $\psi''(s) = 6\,\text{sech}^4(s) - 4\,\text{sech}^2(s) = 2(1-\tanh^2(s))(1-3\tanh^2(s))$, bounded in $[-2/3, 2]$ and vanishing as $|s| \to \infty$. It provides central positive excitation ($\lambda_j > 0$) for $|s| < \text{artanh}(1/\sqrt{3}) \approx 0.66$ peaking at $\psi''(0) = 2$, crosses zero at $|s| \approx 0.66$, and forms negative damping wells ($\lambda_j < 0$) in flanking shoulders $|s| \in (0.66, 1.5)$ reaching a minimum of $-2/3$ at $|s| = \text{artanh}(\sqrt{2/3}) \approx 1.15$. The effective transmission $h(s) = \psi''(s) \cdot s$ shares qualitative traits with activations like GELU, giving near-linear pass-through around the origin while suppressing negative coordinates. In functional screening across curvature constructions—spanning soft softened profiles (temperature τ, broader signed floors, softplus blends) versus hard single-scale ψ'' kernels, as well as rational forms and shifted lobes—training remained broadly stable across smooth second-order potentials without catastrophic divergence. However, softening or shifting the gate alters the curvature well and acts primarily as an empirical activation tune; the core advantage stems from the bounded, signed Hessian formulation itself rather than delicate kernel hardening.

where $\lambda_j(z_t) = \psi''_{\text{ladder}}(u_j^\top z_t + b_j)$. In this Memory-Track, u_j serves as a shared basis for both measuring input features and emitting outputs.

In decoder architectures and autoregressive language modeling, representations must actively rotate and transform across layers into subsequent token distributions. The coaxial constraint ($u_j u_j^\top$) limits each unit to scaling along its input axis without coordinate rotation. To enable flexible directional transformation, we decouple output emission vectors from the input projection dictionary, introducing an independent emission matrix $V \in \mathbb{R}^{m\times d}$:

$$\text{H-FFN}_{\text{decouple}}(x_t; z_t) = \sum_{j=1}^{m} \lambda_j(z_t)(u_j^\top x_t)v_j. \tag{8}$$

Breaking coaxiality ($U \neq V$) relaxes the self-adjoint symmetry constraint without abandoning the second-order Hessian formulation: each unit remains a discrete spectral channel whose throughput is governed by local second derivative $\psi''(\cdot)$. Decoupling simply allows coordinate rotation between measurement basis $\{u_j\}$ and emission basis $\{v_j\}$ while fully preserving curvature-driven spectral gating.

In Equation equation 8, the matrix U carries a dual burden: its rows u_j must simultaneously extract local token features ($u_j^\top x_t$) and parameterize the curvature state ($u_j^\top z_t + b_j$). Under high-learning-rate regimes, gradients from these two objectives can collide within U. To resolve this coupling, we define a dedicated gating dictionary $A \in \mathbb{R}^{m\times d}$, specifying the affine coordinate hyperplanes along which curvature is evaluated. While our causal implementation conditions on prefix context z_t, the formulation is general: A can evaluate curvature over any conditioning state $\tilde{x} \in \mathbb{R}^d$ (including context summaries, normalized token states, or cross-attention representations). This yields a three-channel factorization (U, A, V):

$$\lambda_j(\tilde{x}) = \psi''_{\text{ladder}}(a_j^\top \tilde{x} + b_j), \qquad \text{H-FFN}_{\text{split}}(x; \tilde{x}) = \sum_{j=1}^{m} \lambda_j(\tilde{x})\,(u_j^\top x)\,v_j, \tag{9}$$

where a_j denotes the j-th row of A. This factorization cleanly delineates three functional channels: (1) U determines what content is extracted ($u_j^\top x$); (2) A determines when and how strongly each channel opens ($\lambda_j = \psi''(a_j^\top \tilde{x} + b_j)$); and (3) V determines where the representation is steered (v_j). Initializing $A = U$ and $V = U$ ensures identity with the symmetric operator at step 0. The three channels are trained with channel-specific learning rates paired with a residual linear readout ($d \to d$). Setting $m = 4d$ maintains exact parameter parity with canonical $4d$ two-matrix MLPs.

2.3 Spectral Persistence and Morse Pruning

In differential topology, Morse theory characterizes the topology of a manifold through the non-degenerate critical points of its Hessian, distinguishing persistent geometric directions from degenerate coordinates that contribute little to the landscape. In H-FFN, the potential is assembled from m rank-one spectral units. Units with negligible projection or emission norms behave as lazy parameters with minimal contribution to output variance. Measuring this spectral persistence enables identifying the dominant curvature subspace.

For each spectral unit $j \in \{1, \ldots, m\}$, we define its spectral persistence as the product of its projection and emission norms:

$$p_j = \|u_j\|_2 \cdot \|v_j\|_2, \tag{10}$$

where $\|\cdot\|_2$ denotes the Euclidean vector norm. This scalar measures the joint throughput capacity of the j-th unit: $\|u_j\|_2$ sets input sensitivity, while $\|v_j\|_2$ governs output emission energy.

To prune inactive directions, we sort units by descending persistence p_j and extract the submatrix index set $\mathcal{S} \subset \{1, \ldots, m\}$ corresponding to the top m_{keep} values:

$$\tilde{U} = U_{\mathcal{S},:}, \qquad \tilde{V} = V_{\mathcal{S},:}, \qquad \tilde{A} = A_{\mathcal{S},:}, \qquad \tilde{b} = b_{\mathcal{S}}, \tag{11}$$

yielding reduced parameter matrices in $\mathbb{R}^{m_{\text{keep}}\times d}$ and bias $\tilde{b} \in \mathbb{R}^{m_{\text{keep}}}$.

Empirically, pruning an overcomplete dictionary from $m = 6d$ down to $m_{\text{keep}} = 4d$ preserves the full perplexity and retrieval accuracy of the unpruned model. This confirms that the effective

curvature subspace is naturally low-dimensional and sparse, concentrating representation energy along dominant persistent directions. Whether the dictionary can be compressed even further below $4d$ remains an open question.

In addition to post-hoc Morse pruning, we also evaluate native parameter control during model construction: directly instantiating the architecture with native unit width (e.g., $m = 4d$ or $m = 512$) across (U, A, V). This native parametrization enforces strict parameter parity with standard $4d$ MLPs from initialization without requiring an overcomplete pre-training phase.

2.4 Analytic Equivalence to Standard Networks

A two-layer MLP is the same rank-one sum once the gate is taken from the activation. That identity interprets the MLP inside the Decoupled Hessian family and supplies a step-0 match for fine-tuning.

A standard two-layer MLP maps hidden state $x \in \mathbb{R}^d$ through weights $W_1 \in \mathbb{R}^{d_{\text{ff}} \times d}$, $W_2 \in \mathbb{R}^{d \times d_{\text{ff}}}$, biases $b_1 \in \mathbb{R}^{d_{\text{ff}}}$, $b_2 \in \mathbb{R}^d$, and activation σ (e.g., GELU):

$$\text{MLP}(x) = W_2\sigma(W_1x + b_1) + b_2 = \sum_{j=1}^{d_{\text{ff}}} \sigma(s_j)v_j + b_2, \qquad s_j \triangleq w_j^\top x + b_{1,j}, \tag{12}$$

where $w_j = W_{1,j,:}^\top \in \mathbb{R}^d$ denotes the j-th row of W_1 and $v_j = W_{2,:,j} \in \mathbb{R}^d$ denotes the j-th column of W_2.

In Decoupled H-FFN, the layer computation is $\sum_{j=1}^m \lambda_j(s_j)\, s_j\, v_j$. Factoring a pointwise activation as $\sigma(s) \equiv \lambda_{\text{oracle}}(s) \cdot s$ yields:

$$\lambda_{\text{oracle}}(s) \triangleq \begin{cases} \dfrac{\sigma(s)}{s}, & s \neq 0, \\ \sigma'(0), & s = 0. \end{cases} \tag{13}$$

For GELU, $\sigma'(0) = 0.5$. Substituting $\sigma(s_j) = \lambda_{\text{oracle}}(s_j) \cdot s_j$ into Equation equation 12 gives:

$$\text{MLP}(x) = \sum_{j=1}^{d_{\text{ff}}} \lambda_{\text{oracle}}(s_j)\, (w_j^\top x + b_{1,j})\, v_j + b_2. \tag{14}$$

Equation equation 14 establishes an exact correspondence between the standard MLP and a Decoupled Hessian layer with $U = W_1$, $b = b_1$, $V = W_2^\top$, tied routing ($A = U$), and $\lambda = \lambda_{\text{oracle}}$. Because an existing checkpoint already stores representations through monotonic GELU activations, initializing from a native second-derivative potential ψ'' disrupts the trained forward trajectory at step 0. Setting $\lambda(s) = \text{GELU}(s)/s$ is therefore a necessary bridge to achieve exact step-0 warm-start fidelity on pretrained weights; it serves as an initialization strategy rather than a true second-order curvature gate. All other evaluations use native signed curvature $\lambda = \psi''$, preserving the full second-order curvature operator; only Section 3.3 relies on this warm-start map.

3 Experiments and Results

We evaluate H-FFN across three settings: (1) associative retrieval (Needle-in-a-Haystack and Gap Copy) testing key-value recall interference; (2) causal language modeling (WikiText-103) trained from random initialization; and (3) fine-tuning from pretrained MLPs (GPT-2 on WikiText-103; ViT-B/16 on Galaxy10) using the algebraic map $\lambda(s) = \text{GELU}(s)/s$.

The baseline Standard 4D is a two-layer MLP with intermediate width $d_{\text{ff}} = 4\, d_{\text{model}}$ and GELU. Across primary native scratch evaluations, models share hidden dimension $d_{\text{model}} = 128$, four-head scaled dot-product self-attention ($d_{\text{head}} = 32$), AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay 0.01), and gradient clipping at norm 1.0 (with pretrained fine-tuning preserving native model dimensions, $d = 768$). H-FFN matches the baseline in depth, attention, embeddings, batch tokens, and training steps, with dictionary size $m = 4\, d_{\text{model}}$ and a residual linear readout so that parameter counts remain aligned.

3.1 Associative Memory Retrieval

To evaluate pattern storage and retrieval in feed-forward layers, we test synthetic memory benchmarks: Gap Copy (sequential recall across temporal delays) and Needle-in-a-Haystack (NIAH; localized key-value retrieval from distractor tokens, following the standard protocol (Kamradt, 2023)). In dense MLPs, pattern storage relies on unindexed weight matrices, inducing memory conflicts when distractors disperse hidden states.

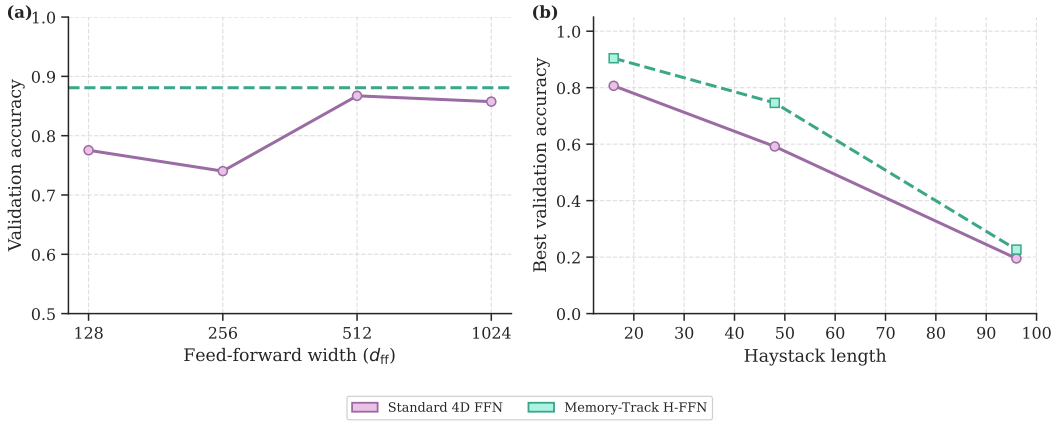

Figure 1: Associative retrieval probes across architectural width and sequence length. **(a)** Feed-forward width sweep: expanding standard MLP width fails to match the performance of the base symmetric H-FFN. **(b)** Sequence capacity curve up to 96 tokens for depth-5 encoder-decoder models. At standard context length (16 tokens), Memory-Track maintains a strong lead (90.4% vs 80.7%), and retains a 10–15 percentage point margin across intermediate lengths (48 tokens). At extreme sequence length (96 tokens), representation capacity degrades for both architectures under compact hidden dimension $d = 128$ (22.7% vs 19.5%).

Figure 1 presents width and sequence length stress probes on deep encoder-decoder models. In Figure 1(a), expanding standard MLP intermediate width up to $16d$ yields marginal accuracy improvements, remaining below the base symmetric H-FFN. This confirms that the retrieval gap is architectural, not explained by intermediate width alone. Figure 1(b) examines retrieval retention as sequence length is scaled from 16 to 96 tokens. Memory-Track consistently outperforms the baseline across short and medium contexts; only at the extreme context limit of 96 tokens do both compact architectures experience representation saturation, declining to $\approx$ 20% accuracy due to hidden state capacity limits at $d = 128$.

Table 1: Memory retrieval results on synthetic associative benchmarks.

(a) Architecture component ladder on Needle-in-a-Haystack (best validation accuracy at 600 steps).

Feed-Forward Arch.	Component Added	Val. Acc.
Standard 4D	Dense MLP baseline	80.9%
Base Symmetric H-FFN	Directional projection dictionary	88.1%
Memory-Track H-FFN	+ Prefix-MS Context Gating	97.1%
Memory-Track H-FFN	+ Multi-Scale Curvature Ladder	99.8%

(b) Encoder-decoder memory retrieval (best validation accuracy, mean$\pm$std over n=3 seeds).

Benchmark	Standard 4D	Memory-Track
Gap Copy	$90.3 \pm 1.0\%$	$96.4 \pm 2.1\%$
NIAH	$75.7 \pm 5.8\%$	$96.5 \pm 0.7\%$

Table 1 reports synthetic memory retrieval. The component ladder (Table 1a) shows layer-wise progression on single-layer NIAH: the base symmetric H-FFN lifts retrieval accuracy from 80.9% to 88.1%; Prefix-MS routing (z_t) reaches 97.1%; the curvature ladder achieves 99.8% ceiling accuracy. The encoder–decoder summary (Table 1b) at $d_{\text{model}} = 128$, depth 2, stabilizes NIAH at $96.5 \pm 0.7\%$ versus $75.7 \pm 5.8\%$ for Standard 4D, with a similar margin on Gap Copy.

3.2 Causal Language Modeling on WikiText-103

We evaluate autoregressive causal language modeling on WikiText-103 (4 layers, context window 256, batch size 32, 3000 training steps, n=3 seeds). Because standard MLPs are sensitive to

schedule choices (shifting from conventional cosine to flat-cosine lowers perplexity from ≈ 565 to 455.3 ± 7.6), all models are strictly evaluated under the same schedule-matched flat-cosine protocol. Applying factor-specific learning rates across U, A, V naturally leverages the modular groupings of decoupled architectures without altering parameter parity ($m = 4d$, Table 2).

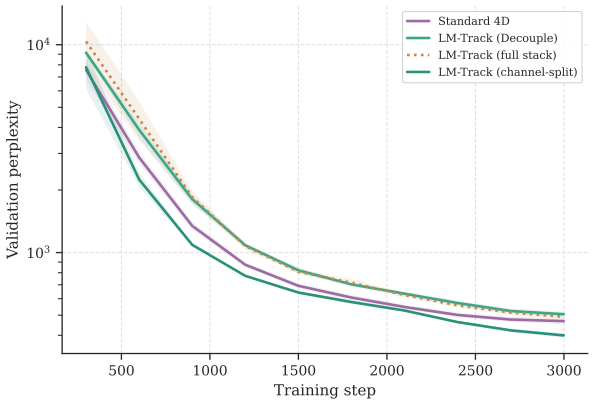

Table 2: WikiText-103 test perplexity at 3000 steps (mean±std, n=3). Gain is $\text{PPL}_{\text{Standard}}/\text{PPL}_{\text{Model}}$.

Config	Params	Test PPL	Gain
Standard 4D	4.92M	455.3 ± 7.6	1.00×
LM Decoupled (U/V)	5.28M	488.4 ± 4.8	0.93×
+ Ladder	5.25M	462.2 ± 4.2	0.99×
Channel-Split ($U/A/V$)	5.25M	388.0 ± 2.3	1.17×

Figure 2: WikiText-103 validation perplexity over 3000 training steps (log scale; mean and shaded ± 1 std over three seeds). All curves use the same flat-cosine schedule. Channel-split denotes the $U/A/V$ factorization of Equation equation 9 at $m = 4d$.

Table 2 and Figure 2 trace a clear stepwise progression: decoupling emission V (U/V, 488.4 ± 4.8; ladder, 462.2 ± 4.2) separates output steering from input projection, while additionally decoupling routing A ($U/A/V$) yields a substantial leap to 388.0 ± 2.3 ($1.17\times$ over the matched MLP). Validation trajectories highlight contrasting dynamics: Standard 4D plateaus early, whereas Channel-Split overtakes it near step 600 ($\sim$20

Channel-split keeps descending after Standard 4D has flattened. What the factorization changes is recorded in Section 4: coaxial units move along one gate coordinate, and a free routing matrix A lets that coordinate leave the content axis. The perplexity curve is the matched-schedule counterpart of that split.

3.3 Fine-tuning from Pretrained GPT-2 and Vision Transformers

To transfer onto pretrained checkpoints without disrupting step-0 representations, we apply the analytic bridge derived in Section 2.4 ($\lambda(s) = \text{GELU}(s)/s$ with $A = U$). We examine this adaptation across two distinct architectures.

On the 12-layer autoregressive GPT-2 model (Radford et al., 2019), we replace the position-wise FFN modules in the final four blocks (layers 8–11) and fine-tune on WikiText-103 (BPE tokenization) for 500 steps. Across three random seeds, the analytic Hessian initialization achieves test perplexities of 28.38, 26.76, and 27.67 (mean 27.60), outperforming matched standard MLP fine-tuning at 30.65, 28.58, and 29.73 (mean 29.65), a $1.07\times$ reduction.

To test a contrasting production-scale bidirectional domain, we evaluate the 86M-parameter Vision Transformer ViT-B/16 (Dosovitskiy et al., 2021) operating on non-causal 2D image patches. Transferring ImageNet-1K pretrained weights (TorchVision ViT-B/16) to the Galaxy10 DECaLS benchmark (Leung & Bovy, 2021) under the standardized AION-1 morphology split (10 classes), matched standard MLP fine-tuning reaches 70.9% validation accuracy at 1000 steps and requires 3000 steps to reach 76.6% test top-1. In contrast, the Hessian warm-start achieves 77.2% validation accuracy and 79.0% test top-1 at just 1000 steps ($+2.4\%$ ahead of the full 3000-step MLP baseline). This shows that factorized Hessian parameterization supports rapid adaptation in large-scale visual backbones.

4 Geometric Trace of the Hessian Layer

In a standard MLP, content projection, gating, and emission are coupled within a shared intermediate representation. In contrast, H-FFN decomposes the sublayer into rank-one units (Section 2.1),

giving each unit explicit coordinates for content extraction ($u_j^\top x$), contextual gating (s_j or $a_j^\top \tilde{x}$), and emission (v_j, Section 2.2). Figure 3 uses this structure to trace the joint density of content and routing coordinates across tokens during training, directly visualizing whether gating remains tied to content magnitude or explores an independent routing dimension.

The dashed line $y = x$ marks the coaxial manifold where gate coordinate and content projection coincide. In the top row, Memory-Track keeps probability density concentrated along $y = x$ as validation accuracy climbs to 0.96, showing that associative recall is maintained along a shared directional basis. In the bottom row, channel-split training causes correlation between $|Ux|$ and $|Az|$ to drop from 0.23 toward zero as density leaves the diagonal, reflecting the progressive decoupling of contextual routing from content magnitude. Together with the curvature profiles in Appendix A (Figure 4), these traces offer a geometric account consistent with the empirical trends in Section 3: coaxial alignment stabilizes associative retrieval (Section 3.1), whereas decoupling routing from content is necessary to bypass the plateau in language modeling (Section 3.2). Because these heatmaps diagnose coordinate coupling rather than asymptotic task metrics, compact step budgets are sufficient.

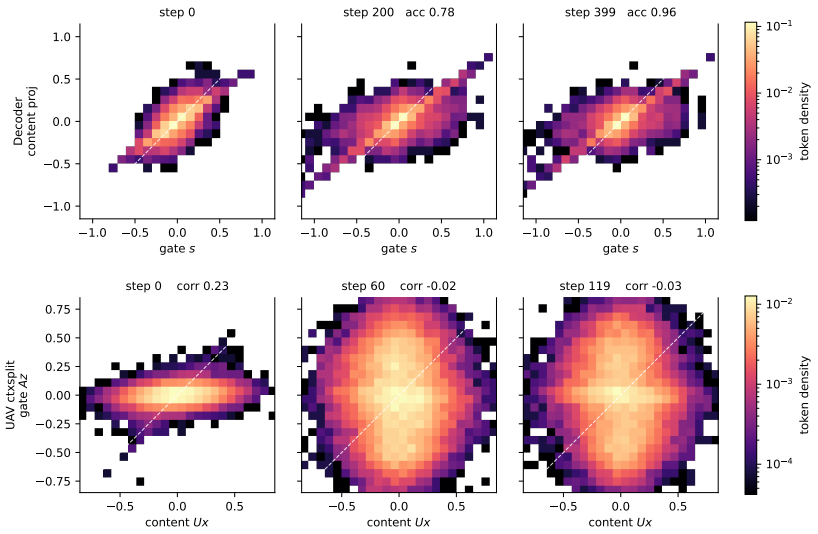

Figure 3: Empirical token density in named H-FFN coordinates (log scale; brighter indicates higher density). Dashed line $y = x$ denotes the coaxial manifold. Top: Memory-Track decoder on Needle-in-a-Haystack (steps 0, 200, 399), plotting context gate $s(z)$ against content projection; density remains along $y = x$. Bottom: channel-split ($U/A/V$) causal LM probe on WikiText-103 (steps 0, 60, 119), plotting $|Ux|$ against $|Az|$; density leaves the diagonal as correlation falls from 0.23 to 0.00.

5 Discussion

A key numerical question is whether second-order curvature gates preserve gradient stability during backpropagation. Layer-wise probes at step 0, normalized to Standard 4D, keep every H-FFN variant well-conditioned. Total gradient norms run from $0.87\times$ (symmetric H-FFN) to $1.45\times$ (decoupled ladder); peak tensor gradients run from $0.49\times$ to $1.41\times$. The feed-forward share of the block gradient stays in 0.12–0.47, against 0.20 for Standard 4D. Removing the block-level identity residual ($x+\text{FFN}(x)$) collapses total throughput by $12.2\times$, so the outer skip is the gradient highway. H-FFN therefore omits an internal residual inside the sublayer.

This design, however, exposes structural trade-offs during multi-layer fine-tuning. A 200-step single-layer probe on GPT-2 (layer 11) yields a $1.23\times$ perplexity reduction (32.64 vs. 40.10); Section 3.3 extends this to 500 steps and four-layer swaps. Single-layer shifts are buffered by surrounding standard layers and the outer skip; replacing deeper blocks narrows the margin as coordinate mismatches compound. Pretrained transfer therefore relies on the analytic map $\lambda(s) = \text{GELU}(s)/s$ with $A{=}U$ (Section 2.4) solely to preserve step-0 checkpoint consistency.

Complementary Marian OPUS-MT fine-tunes show the same constraint: SwiGLU and GeGLU baselines remain stable only under step-0-aligned swaps, so the modest GPT-2 margin in Section 3.3 reflects warm-start fidelity rather than a strictly better λ. Independent routing is a sharper bottleneck—standard GELU blocks admit no closed-form initialization for a separate context matrix A, and native ψ'' or scratch $U/A/V$ replacements destroy step-0 parity (layer-11 validation perplexity $\sim$128). Native scratch training learns these coordinates from initialization (Sections 3.1 and 3.2), whereas full-depth pretrained replacement remains the brittle regime (Section 3.3). Scratch OPUS-100 runs exhibit the same architectural role asymmetry: a Memory-Track encoder with a standard decoder improves BLEU from 5.22 ± 0.42 to 5.89 ± 0.46 (d=128, depth 2, 1200 steps, n=3), whereas a decoupled LM-Track decoder falls to 3.95 (9.02 vs. 8.69 at d=256, 2400 steps, seed 0).

Beyond the primary d=128 benchmark (Section 3.2), supplementary runs confirm the advantage under the same flat-cosine schedule across dimensions and corpora. On WikiText-103 at d=256 (depth 4, 500 steps), Channel-Split ($U/A/V$) lowers validation perplexity from 2833 to 2048 (1.38×), maintaining gains across a learning-rate grid (1.29–1.38×). Transferring to FineWeb-Edu (Penedo et al., 2024) (100M tokens, d=128), Channel-Split achieves 1.11–1.15× perplexity reductions across seeds (451 and 464 vs. 518 and 514). At d=256 (150M tokens), this margin compresses to 1.06× (281 vs. 298). Channel-Split typically trails Standard 4D early and overtakes later: in Section 3.2 near step 600 (about 20% of 3000), on FineWeb-Edu at d=128 near 8% of steps, but at d=256 only after about 87% (step 16,000 of 18,000). Because the primary benchmark crosses during the flat learning-rate phase (flat-cosine ratio 0.6), the delay points more plausibly to a coordination cost for aligning independent U, A, and V, which widens with scale while Standard 4D optimizes a single tied GELU path immediately. A scratch mini-LM at d=384 (depth 4, about 130M tokens, seed 0; word-level WikiText-103) extends this pattern: Channel-Split trails through step 9000 and first overtakes Standard near step 9500 (about 90% of 10,570), ending at validation perplexity 67.1 vs. 72.1 (1.07×). Test perplexity is instead slightly worse (104.0 vs. 100.5), plausibly reflecting a remaining generalization gap, single-seed holdout variance, and a short post-crossover tail rather than fully converged out-of-sample fit. The late crossing is consistent with delayed decoupling of the routing matrix A from the content axis (Section 4). Before scaling to wider production pretraining, these width probes indicate that the decoupled architecture and its optimization recipe still merit refinement alongside the primary d=128 evidence. These supplementary evaluations remain mechanistic validations rather than production pretraining.

6 Conclusion

This work formulates the feed-forward layer as $\mathcal{H}(x)\,x$: each unit projects the hidden state x onto a coordinate $s_j = u_j^\top x + b_j$, scales the projection by a localized gate $\lambda_j = \psi''(s_j)$, and emits along the same axis. A coaxial Memory-Track preserves this geometry for associative retrieval; decoupling separates content, contextual routing, and output emission.

Native curvature gating succeeds most clearly when trained from scratch under matched parameter and schedule parity. In Section 3.1, the coaxial Memory-Track resolves representational interference in unindexed weight matrices, raising Needle-in-a-Haystack recall from $75.7 \pm 5.8\%$ to $96.5 \pm 0.7\%$ (99.8% with the curvature ladder). In Section 3.2, three-channel decoupling ($U/A/V$) achieves a 1.17× perplexity reduction over Standard 4D (388.0 ± 2.3 vs. 455.3 ± 7.6). In transfer, ViT-B/16 reaches 79.0% test top-1 at 1000 steps, +2.4% ahead of the 3000-step MLP baseline, whereas multilayer GPT-2 fine-tuning narrows as coordinate shifts compound. The GELU warm-start helps only where tied (U, V) transfer suffices.

Overall, H-FFN reframes the feed-forward block as an explicit, traceable curvature operator whose factorization can be inspected in trained coordinates. Scratch training under the matched schedule carries the native operator; supplementary checks show relative gains of 1.06–1.38× (with $\approx$1.15× as a mid-scale reference), while co-scaling to larger depth and width remains to be explored.

AI Use Disclosure

The early completion of the code was assisted by Cursor 2.5, and later the wording of the manuscript was revised with Gemini 3.8 Flash. Related-work discovery for FFN and Transformer literature also

used LLM-assisted search. The authors raised the core ideas and assume full responsibility for the content of this paper.

REFERENCES

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH. arXiv:2009.14794.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. URL https://jmlr.org/papers/v24/22-1144.html.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. Approximating two-layer feedforward networks for efficient transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 674–692, 2023. URL https://aclanthology.org/2023.findings-emnlp.49/. arXiv:2310.10837.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://jmlr.org/papers/v23/21-0998.html.

Isaac Gerber. Attention is not all you need: The importance of feedforward networks in transformer models. *arXiv preprint arXiv:2505.06633*, 2025. URL https://arxiv.org/abs/2505.06633.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021. URL https://arxiv.org/abs/2012.14913.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. URL https://arxiv.org/abs/1606.08415.

Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed J. Zaki, and Dmitry Krotov. Energy transformer. *arXiv preprint arXiv:2302.07253*, 2023. URL https://arxiv.org/abs/2302.07253.

Jerry Yao-Chiang Hu, Pei-Hsuan Chang, Haozheng Luo, Hong-Yu Chen, Weijian Li, Wei-Po Wang, and Han Liu. Outlier-efficient hopfield layers for large transformer-based models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 19123–19152, 2024. URL https://proceedings.mlr.press/v235/hu24a.html.

Greg Kamradt. Needle-in-a-haystack benchmark. https://github.com/gkamradt/needle-in-a-haystack, 2023.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5156–5165, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxime Krikun, Noam Shazeer, Zhifeng Chen, et al. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb. arXiv:2006.16668.

Henry W. Leung and Jo Bovy. Galaxy10 DECaLS. Zenodo, 2021. URL https://doi.org/10.5281/zenodo.10845026.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2408–2417, 2015. URL https://proceedings.mlr.press/v37/martens15.html.

Gabriel Mongaras, Trevor Dohm, and Eric Larson. Cottention: Linear transformers with cosine attention. *arXiv preprint arXiv:2409.18747*, 2024. URL https://arxiv.org/abs/2409.18747.

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

Telmo Pessoa Pires, António V. Lopes, Yannick Assogba, and Hendra Setiawan. One wide feedforward is all you need. In *Proceedings of the Eighth Conference on Machine Translation*, pp. 1031–1044, 2023. URL https://aclanthology.org/2023.wmt-1.98/. arXiv:2309.01826.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosFormer: Rethinking softmax in attention. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Bl8CQrx2Up4. arXiv:2202.08791.

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

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. URL https://arxiv.org/abs/1710.05941.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tL89RnzIiCd. arXiv:2008.02217.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. URL https://arxiv.org/abs/2002.05202.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017. URL https://arxiv.org/abs/1706.03762.

Xiuying Wei, Skander Moalla, Razvan Pascanu, and Caglar Gulcehre. Building on efficient foundations: Effectively training LLMs with structured feedforward layers. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.16450.

A Coaxial Gate Mass

Figure 4 is the coaxial companion to Figure 3. It records which gate shape was used, the empirical token density on s, and how large a residual is written back.

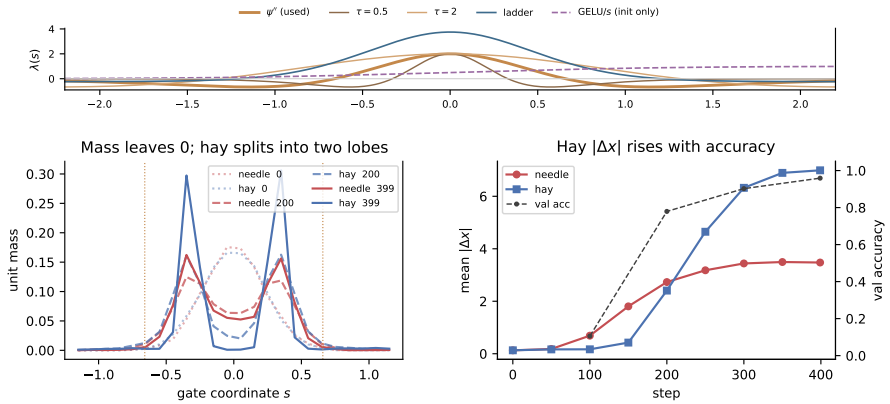

Figure 4: Coaxial Memory-Track on Needle-in-a-Haystack (encoder, seed 0). Top: gates that were tried. The solid ψ'' curve is the operator below; τ and the ladder only reshape that lobe, and $\mathrm{GELU}(s)/s$ is the warm-start, not the trained gate. Bottom left: empirical density on s. By step 399 hay splits into two lobes inside the central well, while needle stays broader. Bottom right: $|\Delta x|$ versus step. Hay's write grows after step 150 to about twice the needle write as accuracy reaches 0.97.

B Reference Implementation (Python)

Listings 1–3 are the forward maps in Section 2. Shapes: token $x \in \mathbb{R}^d$, batch omitted. Native training uses `psi_second` ($\psi = \tanh^2$). Pretrained transfer uses $\lambda = \mathrm{GELU}(s)/s$ with $A = U$, as in Section 2.4; Morse pruning is Equation equation 11.

Listing 1: Curvature gate and rank-one units.

```
import torch

def psi_second(s: torch.Tensor) -> torch.Tensor:
    """Second derivative of sum_j tanh(s_j)^2; bounded signed gate."""
    t = torch.tanh(s)
    return 2.0 * (1.0 - t * t) * (1.0 - 3.0 * t * t)

def curvature_ladder(s: torch.Tensor, K: int = 3) -> torch.Tensor:
    """Multi-scale gate: sum_k 2^{-k} psi''(s / 2^k)."""
    lam = torch.zeros_like(s)
    for k in range(K + 1):
        lam = lam + (0.5 ** k) * psi_second(s / (2.0 ** k))
    return lam

def hffn_coaxial(x, U, b):
    """Memory-Track core: same rows u_j read x and emit output."""
    # U: (m, d), b: (m,), x: (d,)
    s = x @ U.T + b              # affine coords, shape (m,)
    lam = curvature_ladder(s)    # (m,)
    proj = s                     # u_j^T x + b_j
    return (lam * proj) @ U      # sum_j lam_j * proj_j * u_j
```

Listing 2: Prefix-MS context (causal LM / memory tasks).

```
def prefix_ms(x_seq, t, window_sizes=(4, 16, 64), mix_weights=None):
    """Causal multi-scale prefix z_t from states x_1..x_t."""
    if mix_weights is None:
        mix_weights = [1.0 / len(window_sizes)] * len(window_sizes)
    parts = []
    for w, W in zip(mix_weights, window_sizes):
        start = max(0, t + 1 - W)
        parts.append(x_seq[start : t + 1].mean(dim=0))
    z = sum(w * p for w, p in zip(mix_weights, parts))
    z = z + x_seq[: t + 1].mean(dim=0)   # global prefix mean
    return z

def memory_track_ffn(x_t, x_seq, t, U, b):
    z = prefix_ms(x_seq, t)
    s_gate = z @ U.T + b
    lam = curvature_ladder(s_gate)
    proj = x_t @ U.T + b         # content from current token
    return (lam * proj) @ U      # coaxial emission
```

Listing 3: Three-channel split for causal LM (U, A, V).

```
def channel_split_ffn(x, z, U, A, V, b, readout):
    """Content U, gate A on context z, emission V, then d->d readout."""
    lam = curvature_ladder(z @ A.T + b)   # A does not read x
    proj = x @ U.T + b
    y = (lam * proj) @ V
    return readout(y)                       # nn.Linear(d, d) in paper
    runs
```