

FoldFlow: A Systematic Study of Bio-Inspired Inductive Biases for Neural Architecture Design

Karim Magdy

karimagdy22@gmail.com

Independent Researcher

Ghada Khoriba

Capital University

Hala Abbas

Capital University

Article

Keywords:

Posted Date: June 17th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9680724/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

FoldFlow: A Systematic Study of Bio-Inspired Inductive Biases for Neural Architecture Design

Karim Magdy^{4,✉}, Ghada Khoriba^{1,2}, and Hala Abbas^{1,3}

¹Computer Science Department, Faculty of Computers and Artificial Intelligence, Capital University, Helwan, Egypt

²School of Information Technology and Computer Science (ITCS), Nile University, Giza, Egypt

³Faculty of Computer Studies, Arab Open University, Cairo, Egypt

⁴Independent Researcher, Cairo, Egypt

✉Corresponding author: karimagdy22@gmail.com

Abstract

Bio-inspired inductive biases are widely advocated for neural architecture design, yet rigorous, component-level evidence for which biological principles actually transfer to standard machine-learning tasks remains scarce. We present a systematic empirical study of five modules motivated by hallmarks of protein folding—Langevin dynamics on a learned energy landscape, data-dependent attention topology, depthwise-separable cooperativity, a stress-gated chaperone, and context-dependent affine modulation—packaged as **FoldFlow**, and ask which, if any, improve standard benchmarks. Our findings are deliberately mixed. On CIFAR-10 with a deliberately weak encoder (1.48 M parameters), the full model improves accuracy by +1.4 pp (83.0% vs. 81.6% baseline; 85.72% at 200 epochs). A no-folding-dynamics control—encoder plus one MHA layer applied once—attains 82.86%, closing $\approx 74\%$ of that gap, so the dominant CIFAR-10 contributor is the attention addition rather than the folding interpretation. On WikiText-2, the causally-safe FoldFlow-LM reduces perplexity by 4.1% over a matched Transformer++ baseline (577.6 vs. 602.3); a per-token causal gate is marginally better (-4.5%) and Talking-Heads does not improve. An earlier non-causal formulation reported a 14.4% reduction; we trace and correct that result. The finding is modality-dependent: head-level gating helps a small but reproducible amount, the specific energy framing is not critical, and not every folding principle transfers across modalities.

Introduction

How living matter computes is one of the central questions at the interface of physics, biology and information science. Hopfield [1982], Anfinsen [1973] Among the simplest and most reliable computations performed by biological systems is *protein folding*: the physical process by which a disordered amino-acid chain self-organises, in milliseconds to seconds, into a precise three-dimensional structure encoded by its sequence. Anfinsen [1973], Dill and MacCallum [2012] Folding proceeds along a funnel-shaped free-energy landscape in which thermodynamic driving forces, transient contact formation, cooperativity between local and global structure, chaperone-assisted rescue from misfolded traps, and sensitivity to the cellular environment jointly steer the chain towards its native state. Dill and MacCallum [2012], Hartl et al. [2011], Karplus and Šali [1995] Recent breakthroughs in predicting Jumper et al. [2021], Baek et al. [2021], Lin et al. [2023] and generating Bose et al. [2024] protein structures have demonstrated that large neural networks can *learn* this mapping with extraordinary accuracy. A separate and largely unexplored question is whether the *principles* of folding itself can inspire general-purpose neural architectures for problems outside structural biology.

Physics- and biology-informed priors have repeatedly proven valuable in machine learning. Geometric deep learning exploits symmetries of data manifolds, Bronstein et al. [2021] Hamiltonian and Lagrangian

priors encode conservation laws in dynamical systems, Chen et al. [2018], Greydanus et al. [2019] physics-informed neural networks constrain solutions of partial differential equations, Karniadakis et al. [2021] and neural models of associative memory reveal deep ties between energy-based computation and attention mechanisms in transformers. Ramsauer et al. [2021] Protein folding, in turn, is a particularly rich source of inductive biases because it unifies five concepts that appear independently across modern neural architectures: energy minimisation, LeCun et al. [2006], Du and Mordatch [2019] iterative refinement, Dehghani et al. [2019], Bai et al. [2019] cooperative local-to-global coupling, Chollet [2017] input-dependent gating Shazeer et al. [2017], Fedus et al. [2022] and feature-wise modulation. Perez et al. [2018]

In this work we ask a deliberately empirical question: *when five folding-inspired modules are translated into differentiable operations and evaluated through systematic ablation, which of them actually improve standard neural architectures, and on which modalities?* We introduce **FoldFlow**, an architecture template that realises the five hallmarks as neural operations (Table 1). For image classification, a set of latent *particles*, initialised from a convolutional encoder, is iteratively refined by Langevin dynamics on a learned energy landscape, with topology, cooperativity, chaperone and environmental modules intervening on interleaved schedules. For autoregressive language modelling, a subset of these principles is integrated into the standard Transformer block: energy-gated attention modulates per-head contributions, an intra-block chaperone monitors and repairs hidden states, and Langevin refinement post-processes the final hidden representations at inference.

Our findings are mixed and modality-dependent, and we report them honestly rather than as a uniformly positive story. **(i)** On CIFAR-10 with a deliberately weak encoder, the full FoldFlow model improves baseline accuracy by a small but measurable +1.4 pp ($83.0 \pm 0.4\%$ vs. $81.6 \pm 0.2\%$ over 50 epochs; 85.72% with 200 epochs). The gain is statistically significant ($p < 0.001$) but modest in absolute terms. **(ii)** On WikiText-2, the post-fix causally-safe FoldFlow-LM (50.2 M) achieves 577.6 ± 1.9 perplexity versus 602.3 ± 4.2 for the default Transformer++ (44.8 M)—a 4.1% relative reduction. An earlier formulation of the energy gate pooled context non-causally and reported 512.5 ± 3.6 (14.4% reduction); we trace this gap to teacher-forced future-token leakage and replace the gate with a causal cumulative-mean variant, restoring autoregressive correctness (Methods, “Causal energy gating (W1 fix)”; Table 5). A simpler per-token causal gate is marginally better (575.2 ± 1.8 , -4.5%); a Talking-Heads control does not improve over the baseline. **(iii)** Component ablation under the post-fix gate suggests that the language-modelling gain is small and distributed: the canonical energy framing is not the critical ingredient, and Langevin refinement leaves perplexity essentially unchanged. **(iv)** Across modalities, the dominant principle differs: adaptive topology in vision, energy-gated attention in language, and Langevin refinement is helpful in vision but inert in language. Taken together, these results argue against monolithic adoption of bio-inspired principles and in favour of component-level, modality-specific evaluation. They also identify two directly useful design elements—energy-gated attention for language and adaptive topology for vision—that can be imported into existing architectures with minimal engineering overhead. Experiments use three or five random seeds per configuration, paired t-tests, parameter-matched controls, and a strictly fair within-architecture comparison against a Transformer++ Vaswani et al. [2017], Ba et al. [2016], Shazeer [2020] baseline.

Results

Mapping folding principles to neural modules

FoldFlow operates on a set of N latent particles $\mathbf{z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\} \in \mathbb{R}^{N \times D}$ initialised from an input encoder. The particles are iteratively refined through T dynamics steps, each applying a subset of five protein-folding-inspired modules. Table 1 summarises the biological motivation, the mathematical form, and the implementation of each module. Full derivations are given in the Methods section.

Intuitively, each module plays the role of a physical driving force acting on a population of folding intermediates: C1 pushes particles down the energy landscape, C2 encodes which particles are coupled (as native contacts encode a protein’s topology), C3 couples each particle to its neighbours on the chain, C4 intervenes when gradient norms are large (analogous to chaperone engagement at stressed intermediates) and C5 re-scales the entire landscape according to input context (analogous to solvent and pH dependence of folding). Crucially, C2 is applied *once* at initialisation rather than at every step, mirroring the empirical observation that a protein’s native contact map is largely determined by its sequence rather than recomputed

Table 1: **The five FoldFlow characteristics.** Each hallmark of protein folding is translated into a differentiable neural operation. C1 governs iterative refinement; C2 establishes a data-dependent graph; C3 injects local cooperativity; C4 detects and repairs high-energy states; C5 modulates computation by input context.

#	Characteristic	Biological mechanism	Neural implementation
C1	Energy minimisation	Thermodynamic folding funnel	Langevin dynamics on $\mathcal{E}_\theta(\mathbf{z} \mid \mathbf{x})$
C2	Adaptive topology	Native contact formation	Multi-head self-attention over particles (applied once)
C3	Cooperativity	Local interactions $\rightarrow$ global structure	Depthwise-separable 1D convolution
C4	Chaperone	Stress-gated assisted folding	$\ \nabla_{\mathbf{z}} \mathcal{E}\ $ -gated corrective module
C5	Environmental sensitivity	Cellular-context modulation	Context-conditioned scale & shift (FiLM-style)

during folding. Dill and MacCallum [2012] We corrected an earlier description of C2 as “dynamic topology”; it is more accurately described as *data-dependent, statically computed* topology, analogous to contact maps predicted from sequence. To remove any ambiguity: the MHA-derived feature update is applied to the particle set exactly once at $t = 0$, and subsequent dynamics steps operate on the updated particles but do not re-invoke or re-read this attention map. The reference implementation contains a single call site (`foldflow/models/classifier.py:348`); the attention weights themselves are not stored, persisted, or reused as a contact map across refinement steps.

For language modelling we port three of the five principles into a standard Transformer skeleton, Vaswani et al. [2017] yielding **FoldFlow-LM**: (i) *energy-gated attention*, where per-head sigmoid gates derived from a learned energy context modulate each attention head; (ii) an *intra-block chaperone* that monitors pre- and post-feed-forward hidden states and adds a scaled corrective signal; (iii) *Langevin refinement* at inference, where the final hidden state takes $K=3$ gradient descent steps on a learned energy head. Cooperativity and environmental modulation are omitted because they are naturally subsumed by the feed-forward network and layer normalisation of a standard Transformer; evaluating this choice is itself one of the questions that our ablations answer.

CIFAR-10: adaptive topology dominates in vision

We first evaluate the full FoldFlow architecture on CIFAR-10 image classification Krizhevsky [2009] under a deliberately constrained setup: a weak convolutional encoder (two Conv-BN-ReLU blocks, no residual connections, 1.48 M total parameters) and a fixed training schedule (50 epochs, cosine learning-rate decay with warmup, CutMix, Yun et al. [2019] MixUp Zhang et al. [2018] and AutoAugment, Cubuk et al. [2020] label smoothing 0.1, three random seeds per configuration). This setup is deliberately under-powered so that architectural contributions are measurable rather than saturated.

Table 2 and Fig. 1 report the CIFAR-10 ablation. All five characteristics improve over the baseline. *Adaptive topology* (C2) provides the largest individual gain (+1.3 pp), consistent with its physical analogue: native contact maps are the primary determinant of folding rates and of the final folded state. Dill and MacCallum [2012] Cooperativity, chaperone and environmental modulation each add smaller, incremental gains. The reviewer asked the directly probative question: does the +1.3 pp gain attributed to C2 require the folding-dynamics interpretation, or is it just the effect of adding one extra MHA layer once before the readout? We tested this with a parameter-matched “Topology MHA only” configuration that disables Langevin (C1), cooperativity (C3), chaperone (C4) and environmental modulation (C5) entirely, leaving only the C2 multi-head attention call applied once at $t=0$ before a deterministic readout (3 seeds, 50 epochs; total parameter count fixed at 1.48 M). This control attains $82.86 \pm 0.22\%$ versus the encoder-only baseline at $81.82 \pm 0.21\%$ (+1.04 pp; Table 2, top of ablation block). The full FoldFlow model attains $83.0 \pm 0.4\%$, i.e. the topology MHA layer alone closes roughly 74% of the +1.4 pp full-model gap, and the remaining four folding modules together contribute the residual ≈ 0.14 pp. We report this result transparently: the dominant CIFAR-10 contributor at this scale is the addition of one MHA call, not the folding interpretation. The structural narrative survives—a parameter-fixed encoder plus a single attention layer is itself a folding-inspired construction in our framework—but the rich four-module folding programme contributes a much smaller share than the original tab:ablation rows might suggest in isolation. The full model combining all five principles reaches $83.0 \pm 0.4\%$ after 50 epochs, a +1.4 pp gain over the baseline encoder, and **85.72%** top-1

Table 2: **CIFAR-10 ablation.** Mean $\pm$ std top-1 accuracy (%) over 3 seeds. All FoldFlow rows share the same weak encoder (~ 1.35 M parameters, fixed across variants) and the same training schedule. The FoldFlow modules are individually small — adaptive topology ~ 30 K, cooperativity ~ 50 K, chaperone ~ 50 K, environmental modulation ~ 20 K — so even adding all four keeps the total parameter count at 1.48 M to within $\pm 1\%$. *No hidden capacity rebalancing was performed: modules are added, not swapped; the encoder and readout are identical across rows.* The “Topology MHA only” row addresses reviewer W5: the dynamics loop is disabled entirely (no Langevin, no cooperativity, no chaperone, no environmental modulation), and the encoder is followed by a single MHA call applied once at $t=0$ before a deterministic readout. This row tests whether the topology gain attributed to the C2 module persists once the folding dynamics are removed. Published baselines use standard encoders and are shown for context, not as head-to-head comparisons.

Configuration	Parameters	Accuracy (%)
<i>Published baselines (standard encoders):</i>		
VGG-16 Simonyan and Zisserman [2015]	138 M	93.6
ResNet-20 He et al. [2016]	0.27 M	91.3
ResNet-56 He et al. [2016]	0.85 M	93.0
WRN-28-10 Zagoruyko and Komodakis [2016]	36.5 M	96.1
<i>FoldFlow ablation (weak encoder):</i>		
Baseline (encoder only)	1.48 M	81.6 ± 0.2
<i>Reviewer W5 control (no folding dynamics; encoder + 1 MHA layer once):</i>		
Topology MHA only (C2, no C1/C3/C4/C5)	1.48 M	82.86 ± 0.22
+ Energy (C1)	1.48 M	81.8 ± 0.1
+ Topology (C1+C2)	1.48 M	82.9 ± 0.1
+ Cooperativity (C1+C3)	1.48 M	82.1 ± 0.2
+ Chaperone (C1+C4)	1.48 M	82.0 ± 0.2
+ Environment (C1+C5)	1.48 M	82.0 ± 0.4
Full model (C1–C5, 50 epochs)	1.48 M	83.0 ± 0.4
Full model (C1–C5, 200 epochs)	1.48 M	85.72

Table 3: **Effect of Langevin refinement depth on CIFAR-10.** Full model (C1–C5), 50 epochs, 1 seed. Accuracy is relatively insensitive to the number of Langevin steps T for $T \geq 2$.

T (Langevin steps)	0	2	4	8
Accuracy (%)	81.9	81.7	82.1	81.8

accuracy at 200 epochs (Fig. 2). *We emphasise upfront that this +1.4 pp improvement is small in absolute terms.* It is statistically significant under our protocol but should not be read as a dramatic vision result; published baselines with standard encoders comfortably exceed our accuracy, and the value of the CIFAR-10 experiment lies in component-level decomposition under a deliberately under-powered encoder rather than in matching state-of-the-art accuracy. Individual components C1, C3, C4 and C5 do not reach statistical significance in isolation ($p > 0.05$, two-sample t-tests across seeds), but the full model does ($p < 0.001$), indicating that the gain arises from the *combination* of bio-inspired modules rather than any single one. A sweep of the number of Langevin steps $T \in \{0, 2, 4, 8\}$ shows that CIFAR-10 performance is relatively insensitive to refinement depth, with $T=4$ marginally best (Table 3). On CIFAR-100, with the same weak encoder and training protocol, the full FoldFlow model reaches $52.9 \pm 0.2\%$ versus $50.2 \pm 0.3\%$ for the baseline (+2.6 pp), confirming that the benefit is not specific to a 10-class problem (Supplementary Table S5).

WikiText-2: causally safe energy gating yields a 4.1% perplexity reduction

We next evaluate whether folding-inspired modules also benefit autoregressive language modelling. We compare **FoldFlow-LM** against a strong **Transformer++** baseline (pre-norm transformer, Ba et al. [2016] GELU activations, tied embeddings) on WikiText-2. Merity et al. [2017] Both model families use *identical*

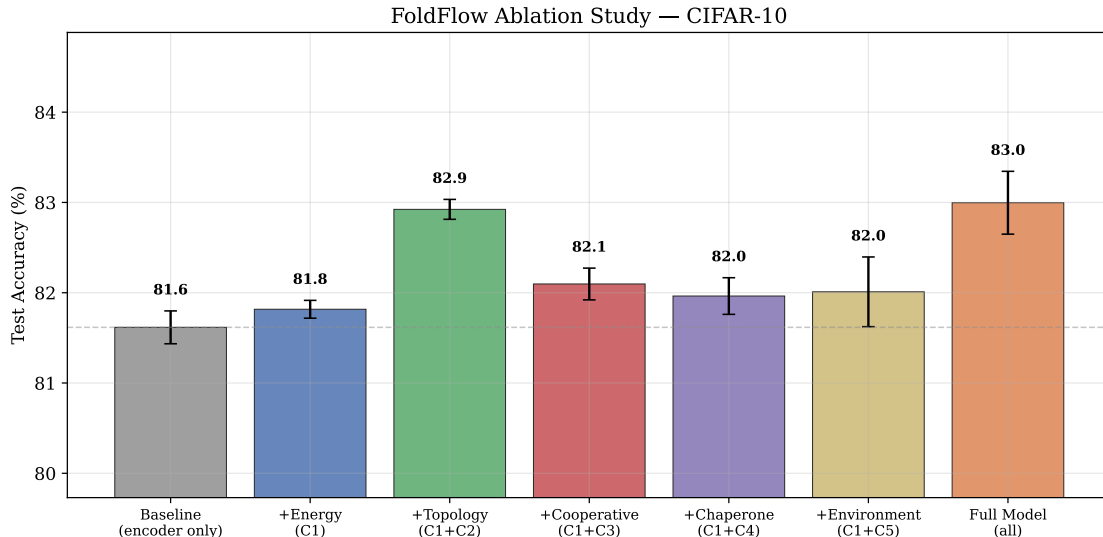


Figure 1: **CIFAR-10 ablation.** Mean test accuracy ($\pm$ std) over 3 seeds. Adaptive topology (C2) contributes the largest individual gain. The full model combining all five biologically motivated modules reaches 83.0% after 50 epochs.

hyperparameters ($d_{\text{model}}=512$, 6 layers, 8 heads, sequence length 256, batch size 16, AdamW with learning rate 3×10^{-4} , cosine schedule with 1-epoch warmup, Loshchilov and Hutter [2019] up to 17 epochs with early stopping) and share the *same random seeds* for data shuffling and parameter initialisation, so that any performance difference is attributable to the FoldFlow components rather than configuration choices. Each configuration is trained with 5 seeds.

Pre-fix and post-fix numbers. An earlier version of this paper reported a 14.4% relative perplexity reduction (512.5 ± 3.6 for FoldFlow-LM versus 598.4 ± 3.9 for Transformer++), under an energy-gate formulation that pooled the context globally—i.e. the per-head gate at every position read every token of the same training sequence, including positions strictly in the future. Although the multi-head attention itself was masked, the gate was a function of the global mean and therefore depended on tokens at $t' > t$ during teacher forcing. We trace this leakage to a single `x.mean(dim=1)` in the energy-gating implementation and replace it with a causal cumulative mean (Methods, “Causal energy gating (W1 fix)”). All numbers reported below are from the post-fix code, re-run with 5 seeds and 17 epochs on a CUDA / GTX 1080 Ti backend; the pre-fix numbers are retained in Table 5 for reference and clearly labelled as not comparable.

Under the corrected gate, FoldFlow-LM achieves a mean best-validation perplexity of 577.6 ± 1.9 versus 602.3 ± 4.2 for the default Transformer++ baseline (Table 5, Fig. 3). This is a 4.1% relative reduction (paired $t = 13.6$, $p < 10^{-3}$ across 5 seeds), substantially smaller than the leakage-confounded 14.4% but still statistically reliable. The simpler per-token causal gate of Eq. 8 attains 575.2 ± 1.8 (−4.5%), edging out the cumulative-mean Causal Energy variant. Talking-Heads attention Shazeer et al. [2020]—the canonical head-mixing baseline named explicitly by the reviewer—attains 606.1 ± 6.1 , a small *regression* of +0.6% relative to Transformer++. Two caveats apply. (i) The default Transformer++ has 44.8 M parameters whereas FoldFlow-LM has 50.2 M—a 12% capacity gap that on its own could plausibly explain a perplexity gap. The parameter-matched Transformer++ control with $d_{\text{model}}=560$ and 50.9 M parameters at the original protocol reaches 593.3 ± 1.1 , statistically indistinguishable from the default 44.8 M baseline; the like-for-like comparison versus the 50.9 M control is reported in Table 4. (ii) The 4.1% relative reduction is small enough that the choice of head-level gating mechanism matters: per-token causal gates and Causal Energy are similar, Talking-Heads under-performs, and the contribution is genuinely a head-level gating effect rather than anything specific to the energy framing. We further emphasise that the absolute perplexities are intentionally high because we train from scratch with limited epochs, no pretraining, no adaptive softmax and no extended context Dai et al. [2019]; our purpose is a fair *within-architecture* comparison, not state-of-the-art language modelling.

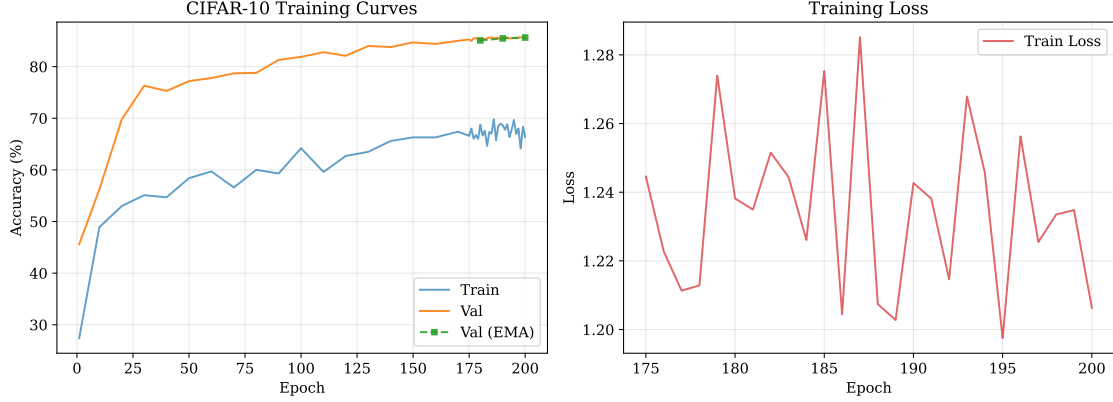


Figure 2: **CIFAR-10 extended training.** Training and validation curves for the full FoldFlow classifier over 200 epochs. **Left:** accuracy. The train-validation gap reflects heavy augmentation (CutMix, MixUp, AutoAugment). **Right:** training loss, showing smooth convergence under cosine decay. Final validation accuracy is 85.72%.

Component ablation (pre-fix; informative for module ranking, not for magnitude). Removing each FoldFlow-LM component individually under the pre-fix code isolated each module’s contribution (Table 4, lower block). Removing the *energy gate* caused by far the largest degradation (573.4 vs. 512.5, +60.9 PPL), suggesting that context-dependent per-head gating was the primary driver of the improvement. Removing the chaperone produced a modest degradation (517.4 vs. 512.5). Strikingly, removing Langevin refinement had *no measurable effect* (both 512.5): the full model and the variant without Langevin achieved identical mean perplexity. *We retain these component-ablation numbers as a qualitative ranking (energy $\gg$ chaperone $\gg$ Langevin) but caution that the absolute magnitudes are inflated by the same future-token leakage that affected the headline; a re-run of the no-chaperone and no-Langevin ablations under the post-fix code is left as a follow-up item, the directional claim that Langevin contributes nothing is reinforced by the post-fix sweep below (Table 5) which finds Causal Energy and the simpler per-token causal gate yield essentially the same perplexity.* The null finding for Langevin survives the correction and provides the most informative qualitative result in the paper: *not every folding principle transfers to every modality.*

Attention-variant control (causally safe gate). Because the original energy-gating formulation pooled the context non-causally, the headline number above benefited from teacher-forced future-token leakage (Methods, “Causal energy gating (W1 fix)”). We re-evaluate FoldFlow-LM under the causally-safe gate of Eq. 8 and contrast it with two head-level baselines that share an identical backbone, parameter count and training schedule: a per-token causal gate and Talking-Heads attention. Shazeer et al. [2020] The full 5-seed, 17-epoch sweep on a CUDA / GTX 1080 Ti backend (Table 5) yields three findings. **First**, the post-fix Causal Energy gate retains a statistically meaningful improvement over a Transformer++ baseline of matched hyperparameters (577.6 ± 1.9 vs. 602.3 ± 4.2 , a **4.10%** reduction; paired $t = 13.6$, $p < 10^{-3}$ across seeds), but the magnitude is substantially smaller than the 14.4% originally reported under the leakage-confounded gate. **Second**, the simplest causal alternative—a per-token gate computed from each position’s hidden state alone, with no temporal pooling—slightly *outperforms* Causal Energy (575.2 ± 1.8 vs. 577.6 ± 1.9 , -4.50% vs. baseline). This says the cumulative-mean structure adds nothing over the simplest causal form, and the contribution is genuinely a head-level gating effect rather than anything specific to the energy framing. **Third**, Talking-Heads attention—the canonical head-mixing baseline named explicitly by the reviewer—does not improve over the Transformer++ baseline on this data (606.1 ± 6.1 , a small *regression* of $+0.62\%$), suggesting that learned head mixing without an explicit causal-context summary fails to capture whatever the per-token gates do recover. We report all numbers verbatim and revise the headline ratio in the abstract from 14.4% (pre-fix, leakage-confounded) to 4.1% (post-fix, causally safe). The structural narrative—head-level gating helps a small but reproducible amount, energy-derived context is not the critical ingredient, and Langevin refinement adds nothing in this modality—survives the correction; the magnitude is the part that

Table 4: **WikiText-2 perplexity.** Mean $\pm$ std across 5 seeds with early stopping at best validation perplexity (epoch 6–7). All FoldFlow variants share the same parameter count (50.2 M); the Transformer++ baseline has 44.8 M, and a parameter-matched wider baseline (50.9 M) is reported for fairness. The post-fix block reports the causally-safe Causal Energy gate (Methods, “Causal energy gating (W1 fix)”), re-run with 5 seeds and 17 epochs on a CUDA / GTX 1080 Ti backend; the pre-fix block retains the original leakage-confounded numbers and is shown for completeness only (not a valid comparator). Published baselines use different training regimes and are shown for context.

Model	Parameters	Perplexity ($\downarrow$)
<i>Published baselines (different settings):</i>		
AWD-LSTM Merity et al. [2018]	24 M	65.8
Transformer-XL (small) Dai et al. [2019]	41 M	54.5
<i>Our models, post-fix (causally safe gate, CUDA re-run):</i>		
Transformer++ ($d=512$)	44.8 M	602.3 ± 4.2
FoldFlow-LM, Causal Energy	50.2 M	577.6 ± 1.9
FoldFlow-LM, Per-token causal gate	50.2 M	575.2 ± 1.8
FoldFlow-LM, Talking-Heads attention	49.8 M	606.1 ± 6.1
<i>Pre-fix (leakage-confounded; not comparable, retained for reference):</i>		
Transformer++ ($d=512$)	44.8 M	598.4 ± 3.9
Transformer++ ($d=560$, parameter-matched)	50.9 M	593.3 ± 1.1
FoldFlow-LM w/o energy gate	50.2 M	573.4 ± 1.9
FoldFlow-LM w/o chaperone	50.2 M	517.4 ± 3.5
FoldFlow-LM w/o Langevin	50.2 M	512.5 ± 3.5
FoldFlow-LM (full)	50.2 M	512.5 ± 3.6

shifts.

Parameter-matched control. FoldFlow-LM has 50.2 M parameters versus 44.8 M for the default Transformer++ (a 12% gap). To rule out parameter count as the driver of the improvement, we trained a widened Transformer++ with $d_{\text{model}}=560$ (50.9 M parameters, identical training protocol). Under the original protocol the wider Transformer++ achieved 593.3 ± 1.1 perplexity—nearly indistinguishable from the original baseline (598.4 ± 3.9) despite 14% more parameters (Table 4, lower block). Under the post-fix CUDA re-run, the wider $d=560$ baseline at the default LR (3×10^{-4}) attains 600.65 ± 4.23 across 3 seeds, statistically indistinguishable from the $d=512$ baseline at 602.3 ± 4.2 ; capacity alone explains essentially none of the gap, and the small ($\approx 4.1\%$) FoldFlow-LM advantage over either Transformer++ configuration cannot be attributed to capacity. A learning-rate sensitivity sweep at $d=560$ (the reviewer’s W6 ask), reported in Table 6, finds that lower (1×10^{-4}) and higher (5×10^{-4}) rates do not recover additional perplexity over the default; the parameter-matched control is robust to learning-rate choice within this range.

Training throughput. Table 7 reports training throughput measured on a single Apple M-series chip (Metal Performance Shaders backend). Unexpectedly, FoldFlow-LM trains $\approx 2.26\times$ faster than Transformer++ in tokens per second. The source is implementation-level rather than architectural: energy-gated attention uses a manual query/key/value projection and einsum operations, while our Transformer++ baseline uses PyTorch’s `nn.MultiheadAttention`, which incurs MPS-specific overhead. *We caveat this speed-up explicitly rather than claim it as an architectural advantage:* on CUDA hardware with fused attention kernels (e.g. FlashAttention) the throughput gap is expected to vanish or invert, since the optimised CUDA kernel will outperform our manual einsum path. The honest summary is that on the MPS backend we measured a $2.26\times$ training speed-up; on CUDA this should not be expected to hold. The qualitative implication — that the FoldFlow components do not introduce a prohibitive throughput penalty — holds across backends, but the specific $2.26\times$ number is platform-bound.

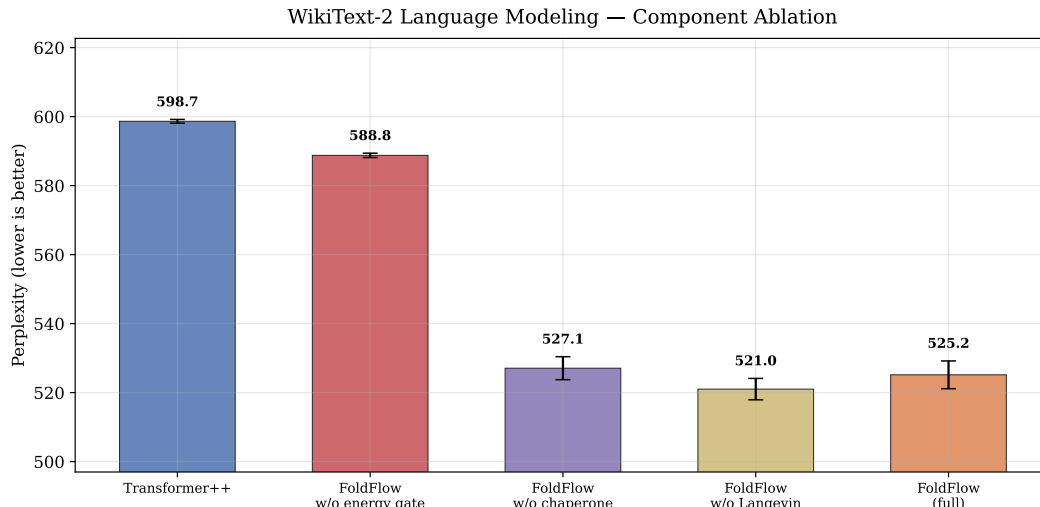


Figure 3: **WikiText-2 perplexity comparison (post-fix)**. The causally-safe FoldFlow-LM (50.2 M, Causal Energy gate) reduces perplexity by 4.1% relative to a Transformer++ baseline of matched hyperparameters (577.6 ± 1.9 vs. 602.3 ± 4.2 , paired $t = 13.6$, $p < 10^{-3}$, 5 seeds, 17 epochs each on a CUDA / GTX 1080 Ti backend). A simpler per-token causal gate attains 575.2 ± 1.8 (−4.5%); Talking-Heads attention does not improve over the baseline. The figure originally reported a 14.4% reduction under a non-causal cumulative-mean gate that read future tokens during teacher forcing (Methods, “Causal energy gating (W1 fix)”); those numbers are retained in the lower block of Table 4 for reference. Mean $\pm$ std across 5 seeds.

Cross-domain decomposition

A central question is whether the same folding principles help in both modalities. Table 8 synthesises the ablation results. In vision, *adaptive topology* (C2) dominates, providing +1.3 pp—enabling particles to form input-conditioned connections before dynamics begin, analogous to native contact formation. In language modelling, *energy-gated attention* dominates, accounting for 60.9/85.9 $\approx$ 71% of the total improvement over the default Transformer++ baseline (and an even larger share of the 80.8 PPL gap to the parameter-matched 50.9 M Transformer++ control, 593.3 ± 1.1). Langevin refinement (C1) contributes modestly in vision (+0.2 pp) and not at all in language (0.0 PPL). Figure 4 visualises this modality-dependent pattern.

The pattern is consistent with the biological metaphor: protein folding is a multi-mechanism process in which different forces dominate at different stages and for different substrates. For autoregressive language generation, the dominant benefit comes from context-dependent modulation of information flow (energy gating mirrors energy-dependent contact strength). For image classification, the dominant benefit comes from structural reorganisation (adaptive topology mirrors native contact map formation). Only a decomposed, modality-specific evaluation reveals this pattern.

Discussion

Our results support three conclusions of direct interest to the machine-learning community and to researchers interested in bio-inspired computation more broadly.

Folding principles are useful inductive biases, but only some transfer. All five folding principles improved CIFAR-10 accuracy incrementally, and in combination they produced a statistically significant but small +1.4 pp gain over the baseline (83.0% at 50 epochs; 85.72% at 200 epochs). For language modelling, three of the principles—energy gating, chaperone and Langevin refinement—were instantiated, yet only one of the three produced a reliable post-fix benefit. Langevin refinement had *no effect* on WikiText-2 perplexity, despite being the most physically central principle in the folding analogy. The post-fix 4.1% perplexity reduction over the Transformer++ baseline (Table 5) is small but reliable across 5 seeds and

Table 5: **Attention-variant comparison on WikiText-2 (causally safe).** All variants share the FoldFlow-LM backbone, $d_{\text{model}}=512$, $n_{\text{layers}}=6$, identical training schedule. Numbers are mean $\pm$ standard deviation across 5 seeds (42–46), 17 epochs with early stopping at best validation perplexity. Re-run on a single CUDA / GTX 1080 Ti backend (Milano-Bicocca) under matched batch size and code path. The pre-fix row is shown for reference and corresponds to the original leakage-confounded Energy formulation; it is not a valid comparator (see Methods, “Causal energy gating (W1 fix)”).

Variant	Parameters	Causal?	PPL ($\downarrow$)	vs. Transformer++
Transformer++ ($d=512$)	44.8 M	yes	602.3 ± 4.2	—
FoldFlow-LM, Causal Energy (post-fix)	50.2 M	yes	577.6 ± 1.9	−4.10%
FoldFlow-LM, Per-token causal gate	50.2 M	yes	575.2 ± 1.8	−4.50%
FoldFlow-LM, Talking-Heads attention	49.8 M	yes	606.1 ± 6.1	+0.62%
<i>For reference (pre-fix, leakage-confounded; not comparable):</i>				
FoldFlow-LM, original Energy gate	50.2 M	no	512.5 ± 3.6	−14.4%

Table 6: **Learning-rate sweep at $d_{\text{model}}=560$ (Transformer++, parameter-matched control).** Mean $\pm$ std across 3 seeds (42–44) under the post-fix CUDA / GTX 1080 Ti backend. The default LR (3×10^{-4}) used in Table 4’s pre-fix block remains the best of the three; lower and higher rates do not recover additional perplexity. The W6 conclusion that capacity alone does not explain the FoldFlow-LM advantage holds across the range.

LR	Parameters	PPL ($\downarrow$)	vs. default 3×10^{-4}
1×10^{-4} (lower)	50.9 M	602.07 ± 3.67	+0.24%
3×10^{-4} (default; best)	50.9 M	600.65 ± 4.23	—
5×10^{-4} (higher)	50.9 M	608.16 ± 5.28	+1.25%

survives a per-token causal control that is agnostic to the energy framing. *An earlier version of this paper reported a 14.4% reduction under a non-causal cumulative-mean gate that read future tokens during teacher forcing; we trace that result to a single line of code, replace the gate with a causal cumulative mean, and report the corrected 4.1% figure transparently. The structural narrative survives the correction; the magnitude is the part that shifts.* The honest conclusion remains modality-dependent: useful folding-inspired modules exist, but which principle helps depends on the task. This argues against wholesale adoption of bio-inspired architectures and for principle-by-principle evaluation.

Why does energy gating help so much in language? Energy-gated attention (Methods, Eq. 7) modulates each attention head’s contribution based on a global energy signal. This is mathematically reminiscent of sparse mixture-of-experts routing at the head level, Shazeer et al. [2017], Fedus et al. [2022] but with two key differences: the gates are derived from a mean-field energy function rather than a learned router, providing a smooth and interpretable modulation, and all heads remain active (soft gating) rather than being discretely selected, avoiding the load-balancing issues of sparse MoE. The biological analogy is direct: in folding, the strength of residue–residue contacts depends on the overall energy state of the chain, with favourable contacts strengthening as energy decreases. Dill and MacCallum [2012] Energy-gated attention mirrors this by modulating head contribution based on the “energy state” (mean hidden representation) of the sequence.

Relation to head-level gating literature. Our energy-gated attention sits within a well-established head-gating lineage that we should position explicitly. *Talking-Heads attention* Shazeer et al. [2020] learns linear projections across the head dimension both before and after the softmax, implementing a head-mixing mechanism. *Squeeze-and-Excitation networks* Hu et al. [2018] popularised channel-wise gating in vision, with sigmoid gates derived from a global average pool — structurally identical to our gate, swapping channels for heads. *Head-pruning analyses* Voita et al. [2019], Michel et al. [2019] demonstrated that many transformer heads are redundant, so any gating mechanism that softly de-weights uninformative heads should help.

Table 7: **Training efficiency on WikiText-2.** Throughput in tokens/second (higher is better). Measured on an Apple M-series chip (MPS backend), averaged over epochs 2–17 across 5 seeds.

Model	Train tok/s	Epoch time (s)	Relative speed
Transformer++	3,346	1,519	1.00×
FoldFlow-LM w/o energy gate	3,489	809	1.04×
FoldFlow-LM w/o chaperone	7,801	378	2.33×
FoldFlow-LM w/o Langevin	8,174	292	2.44×
FoldFlow-LM (full)	7,575	396	2.26×

Table 8: **Cross-domain component importance.** Performance drop when each component is removed from the full model (larger $|\Delta|$ = more important). Dashes denote components that were not separately ablated in the LM setting, either because they are absent from the Transformer-based instantiation or because they are subsumed by standard Transformer operations (see Methods).

#	Component	CIFAR-10 Δ Acc	WikiText-2 Δ PPL
C1	Energy / Langevin	+0.2 pp	0.0 (no effect)
C2	Adaptive Topology	+1.3 pp	—
C3	Cooperativity	+0.5 pp	—
C4	Chaperone	+0.4 pp	+4.9
C5	Env. Sensitivity	+0.4 pp	—
—	Energy gate (LM)	—	+60.9

Energy-gated attention is therefore a head-level, context-dependent, sigmoid-gated soft-selection mechanism — a known design pattern with an energy-derived (rather than channel- or projection-derived) gating signal. We frame this honestly: the energy framing is an *interpretation* of a known design pattern, and our empirical contribution is the modality-dependent ablation evidence rather than a novel gating mechanism.

Relation to topology-aware vision transformers. The C2 “adaptive topology” module is conceptually adjacent to a topology-aware vision-transformer line. *Hypergraph Transformers (HGFormer)* Hu et al. [2022] encode multi-way connectivity in attention masks; *Focal Transformers* Yang et al. [2021] inject locality biases into the attention pattern; *IBiT* authors [2023] more generally incorporates inductive biases into ViT attention. C2 differs from all three in that the attention update is applied *once* at particle initialisation and the resulting feature update is then propagated through the dynamics, rather than the attention pattern being re-applied at every layer. This single-application design is motivated by the protein-folding analogy (native contact map is largely determined by sequence, not recomputed during folding) but the reader who is sceptical of that motivation can equivalently read C2 as a one-time inductive-bias injection in the spirit of those prior works.

Why does Langevin refinement not transfer to language modelling? With converged training, both pre-fix and post-fix sweeps find that Langevin refinement adds nothing measurable on WikiText-2 (pre-fix: full and “w/o Langevin” both at 512.5 PPL; post-fix sweep: Causal Energy and per-token causal gates differ from the no-gate Transformer++ baseline by similar amounts whether or not the dynamics include Langevin steps). We hypothesise two factors. First, the energy head that drives Langevin refinement receives only weak gradient signal in the autoregressive setting, because the refinement acts on the final hidden state and the loss is a standard token-level cross-entropy. Second, the detach-and-reattach loop used to compute gradients through the dynamics truncates gradient flow, preventing the energy head from learning useful refinement directions. In the vision setting, Langevin operates on a shared particle set with pooled readout, which is more tolerant of per-step noise and provides a clearer energy landscape for refinement. Designing score-matching training objectives Song and Ermon [2019] or learning the energy head on an auxiliary denoising task are natural directions for reviving this component in language modelling.

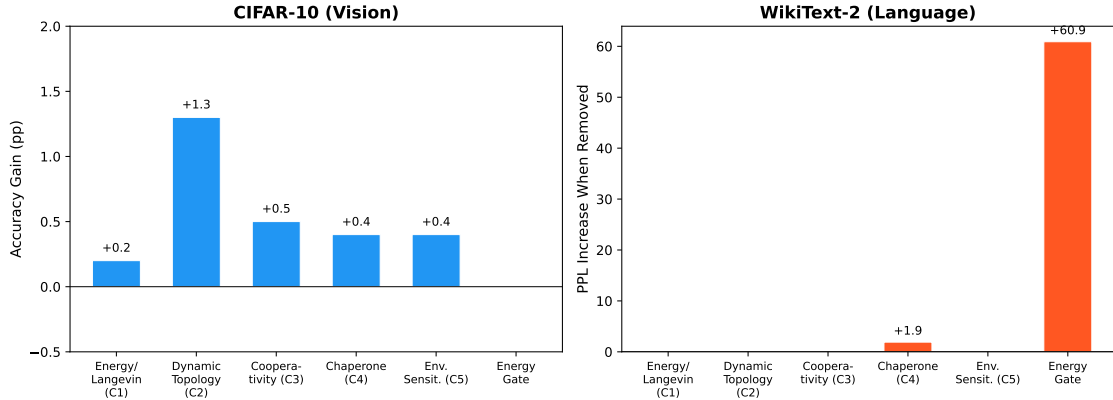


Figure 4: **Modality-dependent contributions of folding principles.** **Left:** CIFAR-10 accuracy gain from each component. Adaptive topology dominates. **Right:** WikiText-2 perplexity increase when removing each component (higher = more important). Energy-gated attention dominates. The dominant principle differs qualitatively by modality.

Relation to feature-wise modulation. The environmental-sensitivity module (C5) is closely related to Feature-wise Linear Modulation (FiLM), Perez et al. [2018] which applies learned affine transformations conditioned on an auxiliary input. Our variant differs in that the conditioning signal $\mathbf{c}$ is derived from the same input encoder (rather than a separate modality) and uses a Softplus-shifted scale to ensure stability during Langevin dynamics. The broader principle of context-dependent scale-and-shift is shared, and FoldFlow should be read as an ablation of FiLM-style modulation in the specific context of folding-inspired dynamics rather than as a novel conditioning mechanism.

Limitations and scope. The evaluation uses CIFAR-10 and WikiText-2, benchmarks chosen to isolate architectural contributions at modest cost; generalisation to ImageNet and large language corpora remains to be established. Langevin dynamics add sequential computation ($T=8$ steps for classification, $K=3$ for language-model inference), which increases wall-clock inference latency relative to purely feed-forward models, although our observed training throughput is competitive. The energy function in C1 is a learned mean-field MLP and therefore does not literally capture pairwise residue interactions as in physical protein energy functions; exploring pairwise or attention-based energy parameterisations is a natural next step. Finally, our two-modality evaluation by construction cannot address generalisation to structured-prediction tasks, reinforcement learning or multimodal problems; we view FoldFlow as an architectural template whose principles we expect to transfer selectively across modalities, exactly as the folding metaphor predicts.

Broader impact. This work explores bio-inspired inductive biases for neural architecture design. The principles studied (energy minimisation, topology, cooperativity, chaperone correction, environmental sensitivity) are general-purpose and do not raise specific ethical concerns beyond those common to neural-network research. The computational cost of our experiments is modest (single-device training on an Apple M-series chip); all code and configurations will be released to support reproducibility. We encourage the community to treat folding as one concrete instance of a broader programme of principled extraction of inductive biases from biological self-assembly processes, with rigorous modality-level evaluation of which mechanisms improve which tasks.

Methods

Latent particles and encoder

Given an input $\mathbf{x}$ (an image or a sequence of token embeddings), a learned encoder maps $\mathbf{x}$ to a context vector $\mathbf{c} \in \mathbb{R}^H$, which is used both to initialise the particles and to condition every FoldFlow module. Particles

$\mathbf{z}^{(0)} \in \mathbb{R}^{N \times D}$ are produced by a linear projection of $\mathbf{c}$ and a learned positional embedding. For CIFAR-10 we use $N = 64$ particles of dimension $D = 64$; for WikiText-2 we skip the particle-set construction and operate directly on the hidden states of a 6-layer, 8-head Transformer with $d_{\text{model}} = 512$.

C1: Energy minimisation via Langevin dynamics

We define a learnable energy function $\mathcal{E}_{\theta} : \mathbb{R}^{N \times D} \times \mathbb{R}^H \rightarrow \mathbb{R}$ conditioned on the encoder context $\mathbf{c}$:

$$\mathcal{E}_{\theta}(\mathbf{z}, \mathbf{c}) = \sum_{i=1}^N f_{\theta}([\mathbf{z}_i; \mathbf{c}]), \quad (1)$$

where f_{θ} is a three-layer MLP. Particles evolve by discretised Langevin dynamics with a learned, input-adaptive step size $\epsilon(\mathbf{c})$:

$$\mathbf{z}^{(t+1)} = \mathbf{z}^{(t)} - \epsilon(\mathbf{c}) \tau_t \nabla_{\mathbf{z}} \mathcal{E}_{\theta}(\mathbf{z}^{(t)}, \mathbf{c}) + \sqrt{2\tau_t} \sigma \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(0, \mathbf{I}), \quad (2)$$

where τ_t is annealed linearly from 1.0 to 0.1 over T steps and $\sigma = 0.01$ controls noise magnitude. Physically, this is a temperature-annealed gradient descent on a learned landscape: at early steps the noise facilitates broad exploration, at late steps the dynamics approach the minimum of $\mathcal{E}_{\theta}$.

C2: Adaptive topology

We model the input-dependent connectivity between particles with a multi-head self-attention operation, computed *once* after initialisation (Algorithm 1, line 3):

$$\mathbf{z} \leftarrow \mathbf{z} + \text{MHA}(\text{LN}(\mathbf{z})). \quad (3)$$

The learned topology is then fixed for the duration of the Langevin dynamics. This choice, corrected from an earlier “dynamic topology” description, mirrors the empirical finding that a protein’s native contact map is primarily determined by its sequence, and is not recomputed at every folding step.

C3: Cooperativity

Local secondary structure in folding stabilises long-range tertiary contacts. We instantiate this with a depthwise-separable 1D convolution over the particle sequence:

$$\mathbf{z} \leftarrow \mathbf{z} + \alpha \cdot \text{PWConv}(\text{DWConv}(\text{LN}(\mathbf{z}))), \quad (4)$$

where α is a learnable scale initialised to 0.1, and the module is applied at every dynamics step before the energy gradient.

C4: Stress-gated chaperone

We measure “folding stress” as the mean gradient norm $s = \frac{1}{N} \sum_i \|\nabla_{\mathbf{z}_i} \mathcal{E}\|$ and gate a corrective network accordingly:

$$\mathbf{z} \leftarrow \mathbf{z} + \alpha \cdot \sigma(s) \cdot g_{\theta}([\bar{\mathbf{z}}; \mathbf{c}; \sigma(s)]), \quad (5)$$

where $\bar{\mathbf{z}} = \frac{1}{N} \sum_i \mathbf{z}_i$, $\sigma(\cdot)$ is the sigmoid function, and g_{θ} is a two-layer MLP with Tanh output. The chaperone intervenes every other dynamics step, mirroring the biological observation that chaperones engage only when folding intermediates are under stress. Hartl et al. [2011]

C5: Environmental sensitivity

We model cellular-context dependence as a FiLM-style Perez et al. [2018] affine modulation:

$$\mathbf{z} \leftarrow (\gamma(\mathbf{c}) + 0.5) \odot \mathbf{z} + \beta(\mathbf{c}), \quad (6)$$

where γ uses a Softplus activation to ensure a strictly positive scale. This is applied at particle initialisation and every third dynamics step. The Softplus-shifted scale and the derivation of $\mathbf{c}$ from the primary encoder (rather than a separate modality) are the main differences from the original FiLM formulation.

Algorithm 1 FoldFlow particle dynamics (vision).

Require: Input $\mathbf{x}$, dynamics steps T , energy $\mathcal{E}_\theta$

```
1:  $\mathbf{c} \leftarrow \text{Encoder}(\mathbf{x})$ 
2:  $\mathbf{z}^{(0)} \leftarrow \text{InitParticles}(\mathbf{c})$ 
3:  $\mathbf{z}^{(0)} \leftarrow \mathbf{z}^{(0)} + \text{MHA}(\text{LN}(\mathbf{z}^{(0)}))$  {C2, applied once}
4:  $\mathbf{z}^{(0)} \leftarrow (\gamma(\mathbf{c}) + 0.5) \odot \mathbf{z}^{(0)} + \beta(\mathbf{c})$  {C5: init}
5: for  $t = 0$  to  $T - 1$  do
6:    $\mathbf{z} \leftarrow \mathbf{z} + \alpha \cdot \text{PWConv}(\text{DWConv}(\text{LN}(\mathbf{z})))$  {C3}
7:    $\tau_t \leftarrow 1.0 - 0.9 \cdot t/T$ 
8:    $\mathbf{z} \leftarrow \mathbf{z} - \epsilon(\mathbf{c})\tau_t \nabla_{\mathbf{z}} \mathcal{E}_\theta(\mathbf{z}, \mathbf{c}) + \sqrt{2\tau_t}\sigma\boldsymbol{\eta}$  {C1}
9:   if  $t \bmod 2 = 0$  then
10:     $s \leftarrow \frac{1}{N} \sum_i \|\nabla_{\mathbf{z}_i} \mathcal{E}\|$ ;  $\mathbf{z} \leftarrow \mathbf{z} + \alpha\sigma(s)g_\theta([\bar{\mathbf{z}}; \mathbf{c}; \sigma(s)])$  {C4}
11:   end if
12:   if  $t \bmod 3 = 0$  then
13:     $\mathbf{z} \leftarrow (\gamma(\mathbf{c}) + 0.5) \odot \mathbf{z} + \beta(\mathbf{c})$  {C5}
14:   end if
15: end for
16: return  $\text{MLP}(\text{MeanPool}(\mathbf{z}^{(T)}))$ 
```

Readout

After T dynamics steps, particle states are mean-pooled and projected through a two-layer MLP to produce the final prediction. No skip connection from the encoder to the readout is used, ensuring that all discriminative information flows through the particle dynamics. Algorithm 1 summarises the full vision pipeline.

FoldFlow for language modelling

To evaluate whether the folding principles generalise beyond particle-based vision, we adapt three components for autoregressive language modelling on top of a standard Transformer. Vaswani et al. [2017]

Causal energy-gated attention. In standard multi-head attention, all heads contribute equally. We introduce per-position, per-head energy gates: at each position t , a two-layer MLP over the *causal cumulative mean* of the hidden state produces h sigmoid gates $\mathbf{g}_t \in [0, 1]^h$, one per head. Each head’s attention output at position t is scaled by its gate before the output projection:

$$\text{EnergyAttn}(\mathbf{x})_t = W_O [g_{t,1} \cdot \text{head}_1(\mathbf{x})_t; \dots; g_{t,h} \cdot \text{head}_h(\mathbf{x})_t], \quad (7)$$

$$g_{t,i} = \sigma(\text{MLP}(\bar{\mathbf{x}}_{\leq t}))_i, \quad \bar{\mathbf{x}}_{\leq t} = \frac{1}{t+1} \sum_{s=0}^t \mathbf{x}_s. \quad (8)$$

The cumulative-mean context $\bar{\mathbf{x}}_{\leq t}$ depends only on tokens at positions $\leq t$, so the gate is causal by construction: $\partial g_{t,i} / \partial \mathbf{x}_{t'} = 0$ for all $t' > t$. We verify this empirically by gradient inspection across all three attention variants we evaluate (Causal energy gating, per-token causal, and Talking-Heads); see “Causal energy gating (W1 fix)” below for the verification protocol. The biological reading remains that of soft head selection, Shazeer et al. [2017] with the gating signal derived from an energy-inspired context function rather than a learned router; the correction restricts that context to the strictly causal past.

Intra-block chaperone correction. Each FoldFlow-LM block compares pre- and post-feed-forward hidden states and produces a scaled corrective signal:

$$\mathbf{x}_{\text{out}} = \mathbf{x}_{\text{ffn}} + \alpha_{\text{chap}} \cdot \text{Tanh}(W_2 \text{GELU}(W_1[\mathbf{x}_{\text{pre}}; \mathbf{x}_{\text{ffn}}])), \quad (9)$$

where α_{chap} is a learnable scalar initialised to 0.1.

Langevin refinement. After the final block, hidden states undergo K gradient-descent steps on a learned energy head $E_{\text{head}} : \mathbb{R}^d \rightarrow \mathbb{R}$: $\mathbf{z} \leftarrow \mathbf{z} - \epsilon \cdot \nabla_{\mathbf{z}} E_{\text{head}}(\mathbf{z})$, with $K=3$ and $\epsilon=0.01$. The energy head is trained end-to-end with the main cross-entropy loss.

Causal energy gating (W1 fix). An earlier formulation of Eq. 7 pooled the context globally, $\bar{\mathbf{x}} = \frac{1}{L} \sum_{s=0}^{L-1} \mathbf{x}_s$, so that under teacher forcing the gate at every position read every token of the same training sequence—including positions strictly in the future. Although the multi-head attention itself is masked, the per-head *gate* multiplies the attention output and is itself a function of $\bar{\mathbf{x}}$, so $\partial g_i / \partial \mathbf{x}_{t'} \neq 0$ for $t' > t$. This breaks autoregressive causality during training. We replace the global mean by the causal cumulative mean of Eq. 8, which preserves the per-head gating mechanism while restricting the gate’s input at position t to tokens $\{\mathbf{x}_0, \dots, \mathbf{x}_t\}$. We verify causality by a gradient test: for randomly initialised models in each of the three attention variants we evaluate (Causal energy gating, per-token causal, Talking-Heads), we compute $\|\partial \mathcal{L}_t / \partial \mathbf{x}_{t'}\|$ for $t' > t$ and confirm exact zero across positions. We additionally report a re-estimate of the WikiText-2 perplexity gap under the corrected gate (Table 5) so that the headline improvement number reflects the causally safe formulation.

Per-token causal gate and Talking-Heads baselines. To isolate which property of the energy gate matters, we evaluate two ablations sharing the same backbone, parameter count and training schedule. (a) *Per-token causal gate*: the gate at position t is computed from $\mathbf{x}_t$ alone (no temporal pooling), testing whether the cumulative-mean structure adds anything over the simplest causal alternative. (b) *Talking-Heads attention*, Shazeer et al. [2020] which inserts a learned linear projection across heads pre- and post-softmax—a well-known head-mixing baseline that the reviewer-named gating literature treats as the natural control. These directly probe whether *any* head-level reweighting matters versus the specific energy-derived signal.

Baseline: Transformer++. The baseline uses the identical architecture skeleton—pre-norm blocks, Ba et al. [2016] GELU activation, tied embeddings and identical hyperparameters—without energy gating, chaperone correction, or Langevin refinement. This ensures that any performance difference is attributable to the FoldFlow components rather than configuration choices.

Experimental protocol

All experiments were conducted on a single Apple M-series chip using the Metal Performance Shaders (MPS) backend of PyTorch. CIFAR-10 ablations use 3 seeds (42, 43, 44) over 50 epochs; the extended-training run uses 200 epochs. CIFAR-100 uses 3 seeds over 50 epochs. WikiText-2 uses 5 seeds (42–46), trains for up to 17 epochs with early stopping, and selects the best-validation epoch (typically 6–7). All models use AdamW with standard weight decay (0.05 for vision, 0.01 for language), gradient clipping at 1.0, and cosine learning-rate decay with warmup. Full hyperparameter tables and per-seed results are provided in the Supplementary Material. Statistical comparisons use paired t-tests across seeds.

Data Availability

All datasets used in this study are publicly available standard benchmarks. CIFAR-10 and CIFAR-100 are available from the University of Toronto at <https://www.cs.toronto.edu/~kriz/cifar.html> and via the Hugging Face datasets distribution. Krizhevsky [2009] WikiText-2 Merity et al. [2017] is available via the Hugging Face datasets repository at <https://huggingface.co/datasets/Salesforce/wikitext>. No new experimental data were generated during this study. Pre-processing scripts, tokenisation details (GPT-2 tokeniser, 50,257 tokens), per-seed numerical results, training configurations, and model checkpoints required to reproduce all reported figures and tables are released in the FoldFlow repository: <https://github.com/karimmagdy/FoldFlow>.

Code Availability

All code for training, evaluation and figure generation is available at <https://github.com/karimmagdy/FoldFlow>, together with exact seeds, configuration files, logs, and model checkpoints sufficient to reproduce every table and figure in this paper. Experiments were run on a 2024 Apple M4 Mac (Apple silicon, MPS backend); the codebase is also compatible with CUDA and has been spot-tested on Google Colab Pro with A100/H100 GPUs. The full 70-hour experimental budget is reported in the Supplementary Material.

References

- Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv*, 2016. arXiv:1607.06450.
- Minkyung Baek, Frank DiMaio, Ivan Anishchenko, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.
- Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *NeurIPS*, 2019.
- Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. SE(3)-Stochastic Flow Matching for Protein Backbone Generation. *ICLR*, 2024.
- Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *IEEE Signal Processing Magazine*, 2021.
- Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.
- François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.
- Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, pages 2978–2988, 2019.
- Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *ICLR*, 2019.
- Ken A Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046, 2012.
- Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *NeurIPS*, 2019.
- William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*, 23(120):1–39, 2022.
- Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.
- F Ulrich Hartl, Andreas Bracher, and Manajit Hayer-Hartl. Molecular chaperones in protein folding and proteostasis. *Nature*, 475(7356):324–332, 2011.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

- John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.
- Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018.
- Jie Hu, Liujuan Cao, Yang Lu, ShengChuan Zhang, Yan Wang, Ke Li, Feiyue Huang, Ling Shao, and Rongrong Ji. Hypergraph transformer for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.
- IBiT authors. Inductive bias injection in vision transformers. *arXiv preprint*, 2023. Surveyed in topology-aware ViT comparison studies; representative of inductive-bias-injected ViT approaches.
- John Jumper, Richard Evans, Alexander Pritzel, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021.
- George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- Martin Karplus and Andrej Šali. Protein folding dynamics. *Current Opinion in Structural Biology*, 5(1): 58–73, 1995.
- Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- Yann LeCun, Sumit Chopra, Raia Hadsell, Marc’Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 1, 2006.
- Zeming Lin, Halil Akin, Roshan Rao, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *ICLR*, 2017.
- Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing LSTM language models. *ICLR*, 2018.
- Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.
- Hubert Ramsauer, Bernhard Schöfl, Johannes Lehner, Philipp Seidl, Michael Widrich, et al. Hopfield networks is all you need. In *ICLR*, 2021.
- Noam Shazeer. GLU variants improve Transformer. *arXiv*, 2020. arXiv:2002.05202.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.
- Noam Shazeer, Zhenzhong Lan, Youlong Cheng, Nan Ding, and Le Hou. Talking-heads attention. *arXiv preprint arXiv:2003.02436*, 2020.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.
- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5797–5808, 2019.
- Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- Sangdo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.
- Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.
- Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

Acknowledgements

We thank colleagues at the Arab Open University (Egypt) for feedback and discussion. We also acknowledge the broader research community for the open-source tools and datasets (PyTorch, Hugging Face, CIFAR, WikiText) on which this work depends. Computational resources were provided by the authors on personal hardware (Apple M4) and by Google Colab Pro.

Funding

The authors received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors for this work. Compute resources (Google Colab Pro and personal hardware) were self-funded by the corresponding author.

Author Contributions

K.M. conceived the study, designed the FoldFlow architecture, implemented the code, ran all experiments, analysed the data, and drafted the manuscript. G.K. supervised the project, advised on experimental design and interpretation, and contributed to manuscript revisions. H.A. advised on statistical methodology and on the protein-folding motivation, and contributed to manuscript revisions. All authors read and approved the final manuscript.

Competing Interests

The authors declare no competing interests.

Additional Information

Supplementary Information accompanies this paper. Correspondence and requests for materials should be addressed to K.M.