Open-world AI: a tale of three Requirements

Anonymous authors
Paper under double-blind review

Abstract

The closed world asks AI for specialized competence and lets it manufacture the IID data its theory assumes: a fixed criterion licenses collecting examples until the task is covered. The open world asks for versatility — taking up tasks a machine was not built for, from few examples and with little prior from a human designer — and offers data that is grouped, not shuffled. Its tasks are not known in advance, so a machine must do *inference-time learning* (1st requirement). Modeling tasks as sharing latent factors and decomposing the training objective, we get two more: a *rich representation*, retaining more than IID generalization requires, and a *predictive disentanglement* that aligns inputs and outputs into matched low-dimensional groups. The analysis further predicts the need for a consistent inner learner, and the benefit of long contexts spanning related tasks. We test these claims on language models up to 8B, asking whether a model has a property (qualitative) rather than how much it wins (quantitative). Models that meet the requirements improve as data from related tasks accumulates; the rest degrade, and neither scale nor layer-type interleaving repairs it.

1 Introduction

> *We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 [...] An attempt will be made to find how to make machines [...] solve kinds of problems now reserved for humans [...]. We think that a significant advance can be made [...] for a summer.* — McCarthy et al., 1955

Seventy years after the Dartmouth proposal (McCarthy et al., 1955), artificial intelligence has produced a long sequence of successes associated with the *closed world*: artificial environments (Samuel, 1959; Tesauro et al., 1995; Silver et al., 2016) and specific real-world tasks (Matan et al., 1991; Taigman et al., 2014). Each rests on a clear criterion of success, which does two things at once. 1) The criterion specifies the distribution the machine is measured against, licensing the collection of more IID examples and a generalization gap falling at $O(l^{-1/2})$ in the number of examples l (Vapnik, 1991). And 2) it reveals not merely *whether* a machine fails but *how* it fails, so that designers can repair the problems one after another. Together these give a loop along which capability accumulates: locate a failure, collect data against it, repair, repeat. On tasks other than the one it was trained on — even one that looks close — the machine may fail (Zech et al., 2018; Koh et al., 2021), and the loop must be run again, by a human, for each new task. After building AI this way, rather than proving that our machines are intelligent, we have more often found satisfaction in proving that we are.

Do large language models escape the *closed world*? Their competence across many specialized domains is real and impressive, and the in-context learning ability it comes with is genuinely remarkable. But two questions are still worth asking: how that competence was obtained, and what kind of in-context learning it is. On the first, the training process is eager to consume all available data and to annotate as much task-specific data as possible until the tasks of interest are likely covered (Yang et al., 2025, §4.1; DeepSeek-AI, 2026, §5.1).[1] Benchmarks then serve as both criterion and measurement; a failure identified is a dataset collected, and the loop continues. This is the closed-world loop, *industrialized* rather than escaped. On the second, the modern in-context learning ability is impressive but fragile (Li

[1] Web crawls, source code, book corpora, chain-of-thought for mathematics, queries for reasoning. The practice is general; these two reports document it explicitly, which most do not.

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.

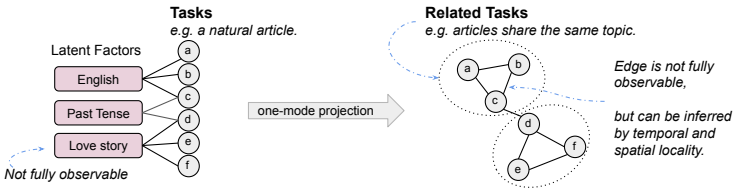


Figure 1: Latent factors, task, and related tasks.

et al., 2024): supplying *more* demonstrations frequently makes performance *worse* rather than better; and frontier models (e.g. GPT-4o) may under-perform smaller & older ones (e.g. LLAMA-2-7b). Neither more task-specific IID data nor more scale reliably improves performance on the task at hand. This is not yet learning.

The Dartmouth proposal set out to produce versatility rather than specialized competence — a machine that can take up a task it was not built for. We take that as the target:

> AI for the open world requires a machine to learn a **wide range** of new tasks, using **fewer examples** and **less task-specific prior knowledge** from human designers.

Unlike closed-world AI where the *tasks of interest* were already known at training time, open-world AI does not come with predefined lists of tasks. Thus, models must be able to adapt to new tasks on the fly.[2] This gives the first requirement, *inference-time learning*.

Just as closed-world AI relies on the IID sampling assumption to connect training and test *examples*, open-world AI needs connections between training and test *tasks* (groups of examples) (Baxter, 2000; Maurer et al., 2016; Tripuraneni et al., 2020; Du et al., 2020). Fortunately, the world is structured, and tasks are not arbitrary. For example, in language modeling, poetry and mathematics are different tasks following very different statistics. Yet these two tasks share substantial latent factors, e.g., language and grammatical construction. According to this observation, one can view tasks and latent factors as a bipartite graph (as in Figure 1), and assume two tasks to be more similar when they share more latent factors.[3]

In this work, Section 2, we first analyze *inference-time learning* under the assumption that the latent-factor overlap of *similar tasks* can be expressed as an overlap over internal representation of either the model inputs or the model outputs. We argue that this analysis suggests two additional requirements:

- *rich representation* (gather broadly): learns features richer than IID generalization requires, making a wide range of new tasks reachable;
- *predictive disentanglement* (group predictively): disentangles input and output representations into matching groups, and performs prediction within each group, making a new task learnable from fewer examples;

Furthermore, this analysis also allows us to make interesting predictions about learning algorithms for building open-world AI:

- about the importance of having a consistent learner in inference-time learning;
- the positive impact of using long contexts that potentially span multiple related tasks.

Finally, Section 3 tests these requirements and predictions on language models up to 8B. What we look for is a *qualitative* difference — whether a model has a given property or lacks it — rather than a *quantitative* margin. A margin in today's language modeling depends sensitively on engineering detail that is largely unshared and would not fit in a scientific paper in any case; a qualitative difference is robust to that detail.

What we find is that models meeting the requirements improve as data from related tasks accumulates, while the rest degrade. Neither scale nor layer-type interleaving repairs it.

2 Inference-time learning

In closed-world AI, both training and test examples are assumed to be drawn IID. This gives no need to learn at inference-time, provided the data are actually sampled that way.

[2] We consider this well understood and broadly accepted.

[3] Breaking this assumption means the world is likely to be arbitrary and unpredictable.

Real-world data, however, has been observed for half a century to be "grouped", where a group of examples (a *task*, written τ) is sampled conditionally on the choice of certain latent factors (Heckman, 1979; Zadrozny, 2004; Torralba & Efros, 2011; Koh et al., 2021; Liu & He, 2025). The latent factors are not fully observable and sparsely changed (Schölkopf et al., 2021) over tasks, resulting a series of similar but not exactly the identical tasks at inference-time. Thus, in open-world AI, adapting to new tasks on the fly — *inference-time learning* — is a must.

2.1 Modeling inference-time learning

To model inference-time learning we make two assumptions: 1) inputs and outputs both have internal representations that are shared across tasks (Finn et al., 2017; Garnelo et al., 2018; Kim et al., 2019); 2) the connection between those representations is task-specific, and must therefore be learned at inference time. The first says what is carried over between tasks; the second says what is not. Together, they ask a machine to have four parts:

1. **input feature extractor** ϕ_θ, on input x, $z = [z_1, \dots, z_G] = \phi_\theta(x)$, shared across tasks.
2. **output feature extractor** ψ, on output y, $\gamma = [\gamma_1, \dots, \gamma_G] = \psi(y)$, shared across tasks. ψ and ϕ learn representation and partition them into matching groups $\{(z_g, \gamma_g)\}$.
3. **decoder** affine A maps γ back to y: $y = A\gamma$.[4]
4. **inner learner** f, the only task-specific part, *adapted at inference-time*. It is fitted on the l support-set examples $S^{(l)} = \{(x_i, y_i)\}_{i=1}^{l} \sim \tau$, carried into latent space as $\{(z_g, \gamma_g)\}$, and predicts group-wise on a query $x^* \sim \tau$: $\hat{\gamma}_g = f_g(z_g)\,, \hat{y} = A[\hat{\gamma}_1, \dots, \hat{\gamma}_G]$, where each f_g is fitted on the group-g pairs $\{(z_g, \gamma_g)\}$ alone.

In practice a model stacks many such units; the analysis concerns the functional decomposition, not its allocation across layers.

Training objective The training process minimizes the error (for ease of analysis we take the squared error throughout) on a query (x^*, y^*) given support-set $S = \{(x, y)\}$ drawn from the same task, averaged over tasks:

$$\mathbb{E}_{\tau, S, (x^*, y^*)}[(\hat{y}^* - y^*)^2] \;=\; \mathbb{E}_{l \sim p(l)}\, \mathbb{E}_{\tau, S^{(l)}, (x^*, y^*)}[(\hat{y}^* - y^*)^2], \tag{1}$$

where $p(l)$ is the distribution of support-set sizes, which deployment does not let a machine choose.

2.2 Decompose the training objective

The reconstruction of y directly supervises decoder and output extractor to avoid collapse: if γ has dropped information about y, and equally if A cannot read it back, the reconstruction fails. Thus we leave decoder A out of the analysis below. We decompose the training objective around the *Bayes-optimal prediction* on query x^* given the support set $S^{(l)}$, formed in the latent space that ϕ_θ and ψ provide:

$$\bar{y}^* \;=\; A\, \mathbb{E}[\gamma^* \mid z^*, \{(z, \gamma)\}^{(l)}]. \tag{2}$$

It is the best any inner learner could do with what the representations hand it. Since A is affine it commutes with the conditional expectation, and since A recovers y from γ, conditioning on $\{(z, \gamma)\}^{(l)}$ is conditioning on $\{(z, y)\}^{(l)}$; hence $\bar{y}^* = \mathbb{E}[y^* \mid z^*, \{(z, y)\}^{(l)}]$. Predicting in latent space and predicting in output space share the same reference, and the decomposition can be written on y:

$$\begin{aligned}
\mathbb{E}_{\tau, S^{(l)}, (x^*, y^*)}[(\hat{y}^* - y^*)^2] &= \mathbb{E}_{\tau, S^{(l)}, (x^*, y^*)}[(\hat{y}^* - \bar{y}^* + \bar{y}^* - y^*)^2] \\
&= \mathbb{E}_\tau \underbrace{\mathbb{E}_{S^{(l)}, x^*}[(\bar{y}^* - y^*)^2]}_{\ell_\theta(l)} + \mathbb{E}_\tau \underbrace{\mathbb{E}_{S^{(l)}, x^*}[(\hat{y}^* - \bar{y}^*)^2]}_{\ell_f(\theta, l)} \\
&= \mathbb{E}_\tau \sigma_\tau^2 + \mathbb{E}_\tau[\ell_\theta(l) - \sigma_\tau^2] + \mathbb{E}_\tau \ell_f(\theta, l)
\end{aligned} \tag{3}$$

where $\sigma_\tau^2 := \mathbb{E}_{x^*}[\mathrm{Var}(y^* \mid x^*, \tau)]$ indicates the irreducible error of task τ. The cross term $\mathbb{E}_{\tau, S^{(l)}, x^*}[(\hat{y}^* - \bar{y}^*)(\bar{y}^* - y^*)] = 0$ vanishes by the definition of $\bar{y}^*$: $\hat{y}^* - \bar{y}^*$ is a function of $(z^*, S^{(l)})$, and $\bar{y}^* - y^*$ has zero mean given the same. This orthogonality is not special to squared error: it holds for any Bregman divergence, hence for cross-entropy.

[4] An affine A does not make ψ affine, since γ lives in a larger space than y.

Neither term in Eq. (3) can be optimized on its own: ℓ_f is defined against the Bayes-optimal $\bar{y}^*$, which is unknown. Only the full objective is available, and l decides which term it reaches. At large l, ℓ_f vanishes for any consistent inner learner, so reducing the objective reduces ℓ_θ — the information the representation has permanently discarded — and builds a *rich representation*. At small l, ℓ_f dominates instead, and reducing the objective presses ϕ and ψ toward coordinates in which the prediction $z_g \to \gamma_g$ factorizes group-wise: *predictive disentanglement*. The two are ordered: because $\bar{y}^*$ is defined relative to ϕ and ψ, the small-l pressure is worth applying only once the representation is rich[5] (obtained by large-l).

I. $\mathbb{E}_\tau[\ell_\theta(l) - \sigma_\tau^2]$ at large l: rich representation. Bayes-optimal prediction $\bar{y}^*$ uses the exact prior of each task (ideal case), and thus it only cares about the information retained in ϕ_θ, invariant to how features are arranged in ϕ_θ. $\mathbb{E}_\tau[\ell_\theta(l) - \sigma_\tau^2]$ removes the effect of tasks' irreducible error; at large l it measures the information about $p(y \mid x, \tau)$ that ϕ_θ has permanently discarded. A smaller $\mathbb{E}_\tau[\ell_\theta(l) - \sigma_\tau^2]$ indicates a richer representation (Zhang & Bottou, 2023), which might be redundant to a task but helpful to other broader tasks.

Furthermore, this work provides a way to learn rich representation. Call an inner learner *consistent* if $\ell_f(\theta, l) \to 0$ as $l \to \infty$ — given enough examples from a task, its prediction approaches the Bayes-optimal $\bar{y}^*$ (Györfi et al., 2002).[6] With a consistent inner learner $\mathbb{E}_\tau \ell_f$ vanishes at large l, so the reduction of $\mathbb{E}_{\tau, S^{(l)}, (x^*, y^*)}[(\hat{y}^* - y^*)^2]$ operates directly on $\mathbb{E}_\tau[\ell_\theta(l) - \sigma_\tau^2]$. *It implies that a rich representation can be constructed by minimizing* $\mathbb{E}_{\tau, S^{(l)}, (x^*, y^*)}[(\hat{y}^* - y^*)^2]$ *at large* l. A linear Gaussian case analysis on rich representation is shown in Appendix A.

II. ℓ_f at small l: predictive disentanglement. The term ℓ_f measures how far the actual inner learner is from the Bayes-optimal predictor (which uses the exact task prior) under the learned representation. Once the representation is rich, ℓ_f dominates in the small-l regime. So the goal is no longer to refine the representation, but to let the inner learner reach $\bar{y}^*$ from as few examples as possible. Without being given a task-specific prior, this asks for a prior shared across the tasks of interest. One way to supply it is to reduce the inner learner from a single joint prediction $z \to \gamma$ with $z, \gamma \in \mathbb{R}^m$, to G separate predictions $z_g \to \gamma_g$ with $z_g, \gamma_g \in \mathbb{R}^{m/G}$.[7] Feature groups are shared across tasks; tasks differ in the function mapping within each group:

$$z = \phi(x) = (z_1, \ldots, z_G), \qquad \gamma = \psi(y) = (\gamma_1, \ldots, \gamma_G),$$
$$\mathbb{E}[\gamma \mid z, \tau] = (\eta_{\tau 1}(z_1), \ldots, \eta_{\tau G}(z_G)), \qquad \eta_{\tau g}(z_g) := \mathbb{E}[\gamma_g \mid z_g, \tau]. \tag{4}$$

Under this structure the Bayes predictor itself factorizes group-wise, so a *block*-diagonal inner learner loses nothing. Therefore, minimizing ℓ_f at small l with such a learner pressures the representations ϕ, ψ to find the coordinates in which this form holds. Since (z_g, γ_g) are aligned for *prediction*, $(z_1, \ldots, z_G)$ and $(\gamma_1, \ldots, \gamma_G)$ are *disentangled* in groups; we call this *predictive disentanglement*.

Prediction in latent space. Predicting $\gamma = \psi(y)$ rather than y is the move made explicit by joint-embedding predictive architectures (JEPA) (LeCun, 2022)[8], which encode the output and measure prediction error in embedding space to avoid spending capacity on unpredictable detail. *Predictive disentanglement* shares that step but differs in two respects. First, the predictor is not dense, but group-wise as in Eq. (4). Its sample complexity is governed by the sub-problem dimension m/G rather than the full dimension m (Stone, 1985). Second, the motivation of predictive disentanglement is sample complexity at small l rather than

[5] This matches the finding of Zhang et al. (2022); Chen et al. (2023), where algorithms that seek an invariant representation directly fail, and a rich representation restores them.

[6] This excludes learners whose effective memory is limited, whether by a fixed-size state (as in standard recurrent and state-space models), a sliding window, or strongly decaying positional encodings.

[7] In analogy to the sparse mechanism shift hypothesis (Schölkopf et al., 2021): tasks vary along sparse, independent directions.

[8] Contrastive predictive coding (Oord et al., 2018) makes the same move with a contrastive objective. Joint-embedding methods (JEA) (Chen et al., 2020; Grill et al., 2020; Bardes et al., 2022) match the embeddings of two views rather than predicting one from the other; they have no predictor for Eq. (4) to factorize.

the difficulty of the output space. JEPA is recovered with a single dense group, no task conditioning, and no decoder.

2.3 Two predictions

2.3.1 Consistency of the inner learner.

The rich-representation analysis depends on the consistency of the inner learner: a consistent inner learner drives ℓ_f to zero at large l, so the optimization pressure falls on ϕ_θ and enriches it. This rich representation in turn prepares features for learning predictive disentanglement at small l.

2.3.2 Task size, and the use of related tasks.

The analysis of rich representation starts from large l. Without it a rich representation cannot be constructed, and predictive disentanglement is therefore hard to learn. The training process thus requires each task to contain enough examples.

Could we sample arbitrarily many examples from the target task, the inner learner would need only to be consistent — predictive disentanglement having already made the sample complexity manageable — and would handle new tasks at inference time without a designer's task-specific prior. We cannot. As the ImageNetv2–ImageNet comparison shows (Recht et al., 2019), unknown hidden factors shift the distribution even under a faithful replication protocol, so data from the *target* task τ_t is hard to accumulate. What accumulates instead is data from *related* tasks τ_r, and using them when the target task does not supply enough examples becomes essential.

We now analyze that case. In the feature space defined by Eq. (4), a related task helps with the target task's group g when its group-wise conditional mean is close to the target task's, $\eta_{\tau_r g}(z_g) := \mathbb{E}[\gamma_g \mid z_g, \tau_r] \approx \eta_{\tau_t g}(z_g)$ (dropping g, z and τ when unambiguous: $\eta_r \approx \eta_t$), and hurts it otherwise. Pooling every example into one estimate cannot tell the two cases apart, and the damage grows as related tasks accumulate. Two things are therefore required: estimating the target and related tasks *separately*, as $\hat{\eta}_t$ and $\hat{\eta}_r$, and mixing them *dynamically*,

$$\hat{\eta} = \lambda\, \hat{\eta}_t + (1-\lambda)\hat{\eta}_r\,,$$

where $\hat{\eta}_t$ uses target examples alone, $\hat{\eta}_r$ uses related-task examples, and λ indicates how to mix. The optimal λ grows as related tasks move further from the target task, and the learner stops borrowing once they become too distant (Grant et al., 2018).

Proof. Write $\hat{\eta}_t = \eta_t + \xi_t$ and $\hat{\eta}_r = \eta_r + \xi_r$ with independent zero-mean estimation errors, $v_t := \mathbb{E}[\xi_t^2]$, $v_r := \mathbb{E}[\xi_r^2]$, and $\delta := \eta_r - \eta_t$ the mechanism discrepancy between them (η_r being the pooled mean over related tasks when several are available). Minimizing the squared error $\mathbb{E}[(\hat{\eta} - \eta_t)^2] = \lambda^2 v_t + (1-\lambda)^2(v_r + \delta^2)$ gives the optimal weight:

$$\lambda^* = (v_r + \delta^2)/(v_t + v_r + \delta^2)\,,$$

which depends only on the statistics of the target and related task estimates. δ is not observed at inference. Treating tasks within a neighborhood as exchangeable makes $\mathbb{E}[\delta^2]$ the between-task variance, and spread among the related tasks estimates it (Gelman et al., 2007). □

3 Experiments

Qualitative effects, not quantitative ones. Section 2.2 and 2.3 derives three requirements and two predictions. The experiments in this section ask, for each, whether a model has it or lacks it — a *qualitative* comparison (*"does a model have a property needed for open-world AI?"*) rather than a *quantitative* one (*"the proposed model outperforms baselines by x percent"*). A quantitative margin in modern language modeling depends sensitively on engineering details — data curation, training schedules, evaluation schemes — that are largely unpublished, making the comparison fragile. By contrast, a qualitative comparison asks for the sign of an effect rather than its size, and a sign is robust to those details.

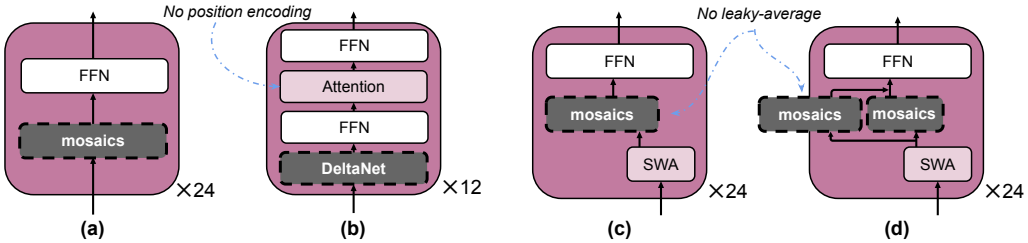


Figure 2: Model architecture groups. Architectures within each group differ only in the dark box. **(a)** *Basic*: the dark box is one of {Mosaics, attention, DeltaNet}. **(b)** *Modern hybrid*: {DeltaNet, sliding window attention (SWA)}, interleaved with full attention (NOPE). **(c)** +: {Mosaics, attention}; the dark box skips the s most recent positions, and the SWA below it uses a small window w (the leaky average in Mosaics is removed). **(d)** *Two-stage*: {Mosaics, attention}; as (c), but the target task and the related tasks are handled separately.

3.1 Connecting the experiments to the theory

Language data. The theory is stated over examples and tasks and does not depend on modality. We instantiate it in language because language corpora are large, widely shared, and the architectures we compare are established in that setting. Every model in this work is trained on exactly the same mixture: Cosmopedia (Ben Allal et al., 2024), NemotronCC-high (Su et al., 2025), open-web-math & arxiv (Azerbayev et al., 2024), Gutenberg books (Manu, 2023), and ShareGPT (Vicuna, 2023).[9]

Building tasks and related tasks. Following *tasks* and *related tasks* illustration in Figure 1, we use natural documents as *tasks*, and semantic clusters of tasks as *related tasks*. We apply k-means clustering to document embeddings (Nussbaum et al., 2024), getting 1150 clusters for training (each contains related tasks).[10]. Similarly, we partition the Simple English Wikipedia[11] evaluation corpus, and get 19 clusters.

The clusters built this way are semantically meaningful and reflect real-world groupings. To further remove any mismatch of k-means clustering, we also run the same evaluation without any post-clustering, treating Simple English Wikipedia as a single natural cluster. See Appendix C for details.

Training and evaluation Unless stated otherwise, every model at each scale uses the same hyper-parameters, optimizer, and same amount of training data, with no per-model tuning. Each sequence is built by sampling documents (tasks) from one randomly drawn cluster (related tasks), delimiting each with `[bos]` and `[eos]`[12], concatenating them, and truncating at length l for training, at length l' for evaluation. The training objective, as in Eq. (1), averages the loss from position 1 to l.

How to read the evaluation. To keep the comparison *qualitative*, and so robust to engineering choices and to the environment, we never directly compare raw performance numbers across model families. Instead, for each model or model family, we vary the other factors — training length l, evaluation length l', how to use related tasks — and ask whether the model has a property needed for open-world AI, yes or no.

3.2 Models to compare

Architecture. Two sizes, 1.5B and 8B. Every model has the same architecture layout: a token embedding, a stack of blocks each pairing an inner learner with a SwiGLU feed-forward network, and a linear decoder. Other detailed settings are in Appendix B.

Figure 2 illustrates the four groups of architectures compared in this work. Swapping the component in the "dark box" gives the different architectures within a group.

[9] Topical diversity is what matters, not the choice of sources: the same experiments on Cosmopedia alone — a single synthetic corpus, but a topically diverse one — give the same conclusions.

[10] The exact number of clusters does not affect the conclusions; anything from a few hundred to a few thousand works.

[11] https://simple.wikipedia.org/wiki/Simple_English_Wikipedia

[12] Every model can therefore tell the target task apart from the related tasks in its context.

a) Basic architecture.

- *Memory Mosaics*: The "dark box" is a Gaussian kernel regression inner-learner over keys and values — a single *long memory* spanning the whole context. This Gaussian regression inner-learner is consistent (Györfi et al., 2002). Keys and values are both built from linear projections of the input: the key k_t is a leaky average of the projections up to t, normalized so that $\|k_t\| = 1$, and the value v_t is a weighted average of the projections at t and $t+1$ ($\|v_t\| = 1$) (Zhang et al., 2025). The prediction is:

$$\frac{1}{z}\sum_{i=1}^{t-1} e^{\beta\, k_t^\top k_i}\, v_i\,, \qquad \beta = \beta_1 t^{\beta_2} + \beta_0\,,$$

  with a bandwidth that scales with position t (Zhang & Bottou, 2026). A per-head sigmoid gate as in DeltaNet is also applied. Keys summarize the past and values the near future, so each stored pair is a *predictive* association rather than a retrieval of one position's own representation.
- *Transformer*: replaces "dark box" with softmax attention (ROPE); positional decay bounds its effective memory, so it is not consistent. A Transformer using NOPE throughout is omitted: with no position encoding anywhere it cannot handle the dependence between nearby tokens.
- *DeltaNet*: replaces "dark box" with a gated delta-rule recurrence, an efficient recurrent alternative; a fixed-size state bounds its effective memory, so it is not consistent.

b) Modern hybrids, interleaving DeltaNet/sliding window attention (SWA) and full attention (NOPE, no position encoding) in successive blocks. This gives two architectures: *Hybrid DeltaNet* and *Hybrid SWA*. Both are configurations that frontier language models are converging on.

c) + architecture.

- *Memory Mosaics+*: one long memory no longer does both jobs — building the example-level (k, v) pairs and predicting from them. The long memory drops the leaky-average key extractor, queries with the unaveraged projection $\tilde{k}_t$ (normalized to 1), and skips the most recent key-value pairs:

$$\frac{1}{z}\sum_{i=1}^{t-s} e^{\beta\, \tilde{k}_t^\top k_i}\, v_i\,.$$

  A short sliding window attention of size w is inserted *below* it to build the examples instead; it may also be a Mosaics block or a DeltaNet block. Following Zhang & Bottou (2026), the skip s is drawn uniformly from 64 to 256 during training and fixed at 64 at inference, and $w = 256$; randomizing s stops the long memory from learning example-level information itself.
- *Transformer+*: as Memory Mosaics+, with attention (NOPE, no position encoding)[13] in place of the kernel regression — a short sliding window below, and above it a long-range attention that skips the most recent positions.

d) Two-stage architecture.

- *Two-stage Memory Mosaics+*: as above, but instead of one long memory covering the target task and its related tasks together, two memories cover them separately,

$$\frac{1}{z_1}\sum_{i\in\text{target}} e^{\beta\, \tilde{k}_t^\top k_i}\, v_i \;+\; \frac{1}{z_2}\sum_{i\in\text{related}} e^{\beta\, \tilde{k}_t^\top k_i}\, v_i\,.$$

  Each is normalized on its own, so the two contributions enter in proportion to their own magnitudes — an implicit form of the mixing weight λ of Section 2.3. Splitting the memory does not duplicate inner-learner computation: the two sums run over disjoint parts of the context, so each stored pair is compared against the query exactly once, as in a single long memory. Every structural choice in this model is one that Section 2 calls for: a consistent inner learner, example construction separated from prediction, and target task evidence held apart from related task evidence.
- *Two-stage Transformer+*: as two-stage Memory Mosaics+, with separate attention (no position encoding) as long memory over target and related evidence.

Groups (a) and (b) are what the field currently builds. Groups (c) and (d) add the two separations that Section 2.3 calls for.

[13] Using attention with position encoding, e.g. RoPE, degrades performance. Check Appendix D.

3.3 Experiments on Rich Representation and Predictive Disentanglement

The positive impact of large training max length l. At large l, ℓ_f vanishes for a consistent inner learner, so the training objective of Eq. (3) is dominated by what the representation has discarded. This produces a rich representation, which in turn prepares the features that disentanglement groups. That is, a large maximum training length l, holding the total amount of training data fixed, benefits tasks other than the ones trained on.

Table 1 trains Memory Mosaics models at various maximum training lengths l and compares their performance at fixed evaluation lengths l' on SimpleWiki. Every column increases downward: the model trained at $l = 64$k is the best at every evaluation length, including lengths at which the extra context it saw during training is never available.

Figure 3 takes the same trained models, fine-tunes them on mathematics (Yu et al., 2023), and evaluates them on GSM8K, whose problems are shorter than 1k tokens. The result shows a positive correlation between l and downstream math reasoning performance.

	Evaluation length l'			
Training length l	4k	8k	16k	32k
4k	48.55	—	—	—
8k	48.93	48.82	—	—
16k	48.97	49.19	49.32	—
32k	49.15	49.41	49.64	49.68
64k	**49.19**	**49.48**	**49.84**	**50.00**
$\Delta > 0$	✔	✔	✔	✔

Table 1: Memory Mosaics top-1 accuracy (%) on SimpleWiki. Rows are the training length l, columns the evaluation length l'. Cells with $l < l'$ are omitted. Δ is the 64k row minus the matched-length entry. Every column increases downward: the longest-trained model is best.

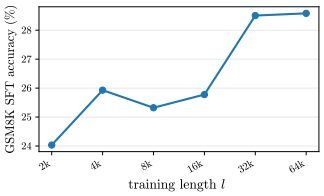


Figure 3: Memory Mosaics reasoning accuracy (%) on GSM8K. We fine-tune each model on Yu et al. (2023) datasets using 131k examples. Sweep learning rate in [3e-5, 6e-5, 9e-5], training epochs in [2, 4, 6, 8], and pick the best config for each model.

The importance of a consistent inner learner. The gain above depends on consistency. If the inner learner is not consistent, ℓ_f does not vanish at large l, and ℓ_θ loses the direct optimization pressure — let alone the learning of disentanglement.

Table 2 repeats the protocol of Table 1 for four inner learners that lack consistency: attention with RoPE, a delta-rule recurrence, and the two hybrids. The sign flips, from beneficial to harmful. None shows a systematic gain, while Memory Mosaics in Table 1 gains positive benefits from a larger maximum training length. For all four, the best configuration is to train at or just beyond the evaluation length.

	l'	*Transformer[rope]*				*DeltaNet*				*Hybrid* SWA				*Hybrid DeltaNet*			
l		4k	8k	16k	32k	4k	8k	16k	32k	4k	8k	16k	32k	4k	8k	16k	32k
4k		48.40	—	—	—	47.40	—	—	—	48.19	—	—	—	48.57	—	—	—
8k		48.48	48.70	—	—	47.49	47.48	—	—	47.88	47.80	—	—	48.49	48.70	—	—
16k		48.05	48.27	48.60	—	47.49	47.43	47.58	—	47.97	47.98	48.01	—	48.15	48.33	48.57	—
32k		47.72	48.07	48.44	48.49	47.27	47.23	47.31	47.15	47.65	47.60	47.65	47.53	46.49	46.51	46.52	46.37
64k		47.51	47.85	48.23	48.39	47.35	47.30	47.40	47.21	47.54	47.55	47.60	47.46	46.24	46.26	46.28	46.13
$\Delta > 0$		✘	✘	✘	✘	✘	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘	✘

Table 2: Top-1 accuracy (%) on SimpleWiki for four inner learners that lack consistency, under the same protocol as Table 1. A longer max training length l does not help these models learn new tasks; it hurts them.

Predictive disentanglement. Predictive disentanglement reduces the sample complexity of a new task: the G-group partition of the latent is learned from the training tasks and reused, and only the prediction within each group is recomputed for the new task (Eq. (4)). A partition that captures the real structure — the kind a consistent inner learner produces — therefore benefits from a large G; one that merely fits the training tasks does not.

Table 3 sweeps the number of heads for Memory Mosaics, Transformer, and Hybrid SWA.[14] The sign splits along consistency: Memory Mosaics gains positive benefits as the number of heads rises,[15] while the Transformer and Hybrid SWA lose.

[14] DeltaNet and the hybrids containing it are omitted, because changing the number of heads also affects the total recurrent state size; the full attention in hybrid SWA has no position encoding.

[15] The per-head gate is not the cause. See Appendix F for ablation with gate removed.

Model	Number of groups G (heads) 4	8	16	32	64	$\Delta > 0$
Memory Mosaics	48.43	48.63	48.97	**49.07**	49.02	✔
Transformer	48.31	**48.38**	48.05	47.92	47.08	✘
Hybrid SWA	**48.00**	47.95	47.97	47.67	46.85	✘

Table 3: Top-1 accuracy (%) on SimpleWiki, evaluated at length $l' = 4k$. All three are trained at length $l = 16$k. Model width is fixed, so raising the number of heads raises the number of groups G and shrinks each group. Row maxima in bold. Δ is (64 heads − 4 heads).

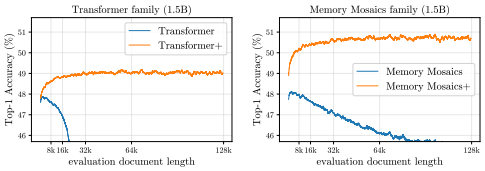


Figure 4: Left: Transformer vs Transformer+. **Right:** Memory Mosaics vs Memory Mosaics+. x-axis: evaluation per-document length. The gap widens with length, exceeding 5 points at 128k.

3.4 Inference-time learning

Scaling the per-task length at inference. Consider first the simple case in which there are no related tasks, in training or at evaluation — only isolated ones. The inner learner of Section 2.1 reads a support-set of examples drawn independently from one task. A token sequence does not come this way: neighboring tokens are strongly dependent, whereas sufficiently separated blocks behave as approximately independent (Yu, 1994; Mohri & Rostamizadeh, 2008). This is why in + architectures (Figure 2 c)) we handle neighboring tokens separately and delay the long-memory to ignore neighboring tokens.

To test this, we train four models (Transformer, Transformer+, Memory Mosaics, Memory Mosaics+) with a maximum sequence length of 16k under a document-wise causal mask, which confines attention computation within each document even if it is very short (never across-documents). We evaluate them on synthetic long documents formed by concatenating Simple English Wikipedia articles delimited by `\n\n`.

Figure 4 shows the result. Transformer+ improves on Transformer, and Memory Mosaics+ improves substantially on Memory Mosaics. Both "+" models extrapolate beyond the training length. The choice of short memory does not matter. See Appendix E for ablation.

Scaling the number of related tasks at inference. We now return to the realistic case: few examples from the target task, and many from related tasks. Pooling all of them is biased, as Section 2.3 shows; modeling them separately and combining them dynamically gives the optimal estimate. That is what the "two-stage" models do. The dynamic mixing term λ is implicitly controlled via the magnitudes of each stage. See Appendix G for details.

Figure 5 shows the result. One-stage models peak around their training length and then decline or flatten. Two-stage Transformer+ keeps improving, though slowly; two-stage Memory Mosaics+ improves quickly. The rightmost panel repeats the comparison at 8B, over established architectures, hybrids, and both + variants, and every curve keeps its shape. Capacity is not the missing ingredient; structure is.

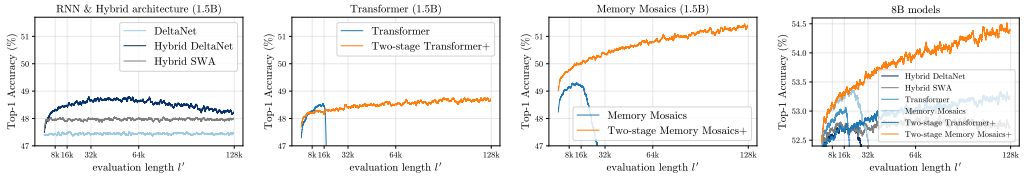


Figure 5: Accuracy as the number of related tasks in the context grows. Training length $l = 16$k. At 1.5B: **1)** RNN and hybrid architectures; **2)** Transformer versus two-stage Transformer+; **3)** Memory Mosaics versus two-stage Memory Mosaics+. The **4)** panel repeats the comparison at 8B.

4 Conclusion

Open-world AI asks for a machine that takes up tasks it was not built for — the first requirement, *inference-time learning*. Modeling the problem with tasks that share latent factors, and decomposing the training objective, this work produces two more requirements, a *rich representation* and a *predictive disentanglement*, and two predictions: that the inner learner must be consistent, and that target and related tasks must be handled separately. The experiments test each qualitatively, asking whether a model has the property or lacks it. Models with these properties improve as related tasks accumulate, while the others degrade regardless of scale.

AI DISCLOSURE

In this work, we used generative AI tools for none of the tasks with required disclosure. We have not used generative AI tools for: Generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. Additionally, we used generative AI tools for aiding or polishing writing. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix B specifies the architectures, the optimizer and schedule, the token budget, the data mixture and the clustering procedure used for every experiment reported here, and Appendix B.3 describes the selection of the evaluation clusters. Every corpus we train or evaluate on is publicly available, as is the embedding model used to build the clusters (Nussbaum et al., 2024). With the final version of this paper[16] we will release source code.

REFERENCES

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Qiaochu Jiang, Jia Deng, Stella R Biderman, and Sean Welleck. Llemma: An open language model for mathematics. In *International Conference on Learning Representations*, volume 2024, pp. 40622–40649, 2024.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Cosmopedia, February 2024. URL https://huggingface.co/datasets/HuggingFaceTB/cosmopedia.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Yongqiang Chen, Wei Huang, Kaiwen Zhou, Yatao Bian, Bo Han, and James Cheng. Towards understanding feature learning in out-of-distribution generalization. *arXiv preprint arXiv:2304.11327*, 2023.

DeepSeek-AI. Deepseek-v4.1-flash: Pushing the limits of kv cache compression. Technical report, DeepSeek-AI, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4.1-Flash.

Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International conference on machine learning*, pp. 1704–1713. PMLR, 2018.

[16]This research, including all access and use of models and datasets, was completed in authors' personal capacity.

Andrew Gelman, Jennifer Hill, et al. *Data analysis using regression and multilevel/hierarchical models*, volume 1. Cambridge university press Cambridge, 2007.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer, 2002.

James J Heckman. Sample selection bias as a specification error. *Econometrica: Journal of the econometric society*, pp. 153–161, 1979.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024.

Zhuang Liu and Kaiming He. A decade's battle on dataset bias: Are we there yet? In *International Conference on Learning Representations*, volume 2025, pp. 41433–41449, 2025.

Manu. Project gutenberg. https://huggingface.co/datasets/manu/project_gutenberg, 2023.

Ofer Matan, Christopher J Burges, Yann LeCun, and John Denker. Multi-digit recognition using a space displacement neural network. *Advances in neural information processing systems*, 4, 1991.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016. URL http://jmlr.org/papers/v17/15-242.html.

John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence. Unpublished manuscript, Dartmouth College, August 1955.

Mehryar Mohri and Afshin Rostamizadeh. Rademacher complexity bounds for non-i.i.d. processes. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Charles J Stone. Additive regression and other nonparametric models. *The annals of Statistics*, pp. 689–705, 1985.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2459–2475, 2025.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

Gerald Tesauro et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528, 2011. doi: 10.1109/CVPR.2011.5995347.

Nilesh Tripuraneni, Michael Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *Advances in neural information processing systems*, 33: 7852–7862, 2020.

Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

Vicuna. Sharegpt vicuna unfiltered. https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Bin Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, pp. 94–116, 1994.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 114, 2004.

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11): e1002683, 2018.

Jianyu Zhang and Léon Bottou. Learning useful representations for shifting tasks and distributions. In *International Conference on Machine Learning*, pp. 40830–40850. PMLR, 2023.

Jianyu Zhang and Léon Bottou. Memory mosaics at scale. *Advances in Neural Information Processing Systems*, 38:38929–38956, 2026.

Jianyu Zhang, David Lopez-Paz, and Léon Bottou. Rich feature construction for the optimization-generalization dilemma. In *International Conference on Machine Learning*, pp. 26397–26411. PMLR, 2022.

Jianyu Zhang, Niklas Nolte, Ranajoy Sadhukhan, Beidi Chen, and Léon Bottou. Memory mosaics. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IiagjrJNwF.

Open-world AI: a Tale of Three Requirements

Supplementary Material

A A linear Gaussian case study

This section makes $\mathbb{E}_\tau[\ell_\theta(l) - \sigma_\tau^2]$ of Eq. (3) explicit in a case where every quantity can be computed, and shows what a *rich representation* is there.

Data generation. Each task τ is a parameter $\omega \in \mathbb{R}^d$:

$$y = \omega^\top x + \epsilon, \qquad x \sim \mathcal{N}(0, \Sigma_x), \qquad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2), \qquad \omega \sim p(\omega),$$

with ω, x and ϵ mutually independent, $\Sigma_\omega := \mathbb{E}[\omega\omega^\top]$ finite, and $\mathrm{rank}(\Sigma_\omega) \geq m$, $m \leq d$. The irreducible error is the same for every task, $\sigma_\tau^2 = \sigma_\epsilon^2$.

The **input feature extractor** is a rank-m linear bottleneck, $\phi = V \in \mathbb{R}^{m\times d}$, $z = Vx \in \mathbb{R}^m$. The **output feature extractor** and **decoder** are set to identity. The **inner learner** f is ordinary least squares (OLS) with weight $\hat{\mu} \in \mathbb{R}^m$ and prediction $\hat{y}^* = \hat{\mu}^\top z^*$. OLS is chosen because at $l \to \infty$ it attains the Bayes prediction given z: $\ell_f(\theta, \infty) = 0$ exactly, so the whole of the remaining error is ℓ_θ.

Work in the whitened coordinates $\tilde{x} = \Sigma_x^{-1/2} x \sim \mathcal{N}(0, I)$, $\tilde{\omega} = \Sigma_x^{1/2}\omega$ and $\tilde{V} = V\Sigma_x^{1/2} \in \mathbb{R}^{m\times d}$, so that $y = \tilde{\omega}^\top\tilde{x} + \epsilon$ and $z = \tilde{V}\tilde{x}$. Write

$$M := \mathbb{E}[\tilde{\omega}\tilde{\omega}^\top] = \Sigma_x^{1/2}\,\Sigma_\omega\,\Sigma_x^{1/2}$$

for the whitened task-signal covariance, and $P_{\tilde{V}} = \tilde{V}^\top(\tilde{V}\tilde{V}^\top)^+\tilde{V}$ for the orthogonal projection onto $\mathrm{row}(\tilde{V})$, where $^+$ is the Moore–Penrose pseudoinverse.

What the representation discards. **Proposition 1.** *With OLS as the inner learner,*

$$\mathbb{E}_\tau[\ell_\theta(\infty) - \sigma_\tau^2] \;=\; \mathbb{E}_\omega[\tilde{\omega}^\top(I - P_{\tilde{V}})\tilde{\omega}] \;=\; \mathrm{tr}(M(I - P_{\tilde{V}})).$$

This is the expected squared task signal lying in the whitened directions **orthogonal to the row space of** $\tilde{V}$. It does not depend on l: no amount of task data recovers it, only a better V does. This is the quantity Section 2.2 calls the information the representation has permanently discarded.

Proof. $P_{\tilde{V}}$ is symmetric and idempotent with $\mathrm{rank}(P_{\tilde{V}}) = \mathrm{rank}(\tilde{V})$, and $(I - P_{\tilde{V}})\tilde{x} \perp \mathrm{row}(\tilde{V})$.
As $l \to \infty$ the OLS normal equations converge, $(Z^\top Z)/l \to \mathbb{E}[zz^\top]$ and $Z^\top y/l \to \mathbb{E}[zy]$, so for a task ω,

$$\hat{\mu}_\omega = (Z^\top Z)^+ Z^\top y \;\longrightarrow\; \mathbb{E}[zz^\top]^+\,\mathbb{E}[zy] = (V\Sigma_x V^\top)^+ V\Sigma_x\omega,$$

giving the prediction and the residual

$$\hat{y}^*_\infty = \hat{\mu}_\omega^\top V x^* = \tilde{\omega}^\top P_{\tilde{V}}\tilde{x}^*,$$
$$y^* - \hat{y}^*_\infty = \tilde{\omega}^\top\tilde{x}^* + \epsilon - \hat{y}^*_\infty = \tilde{\omega}^\top \underbrace{(I - P_{\tilde{V}})\tilde{x}^*}_{\perp\,\mathrm{row}(\tilde{V})} + \epsilon.$$

Squaring and taking expectations,

$$\begin{aligned}\mathbb{E}_{\omega,x^*,\epsilon}[(y^* - \hat{y}^*_\infty)^2] &= \mathbb{E}[(\tilde{\omega}^\top(I - P_{\tilde{V}})\tilde{x}^*)^2] + 2\,\underbrace{\mathbb{E}[\tilde{\omega}^\top(I - P_{\tilde{V}})\tilde{x}^*\,\epsilon]}_{=0} + \underbrace{\mathbb{E}[\epsilon^2]}_{=\sigma_\epsilon^2} \\ &= \mathbb{E}_\omega[\tilde{\omega}^\top(I - P_{\tilde{V}})\,\underbrace{\mathbb{E}[\tilde{x}^*\tilde{x}^{*\top}]}_{=I}(I - P_{\tilde{V}})\tilde{\omega}] + \sigma_\epsilon^2 \\ &= \mathbb{E}_\omega[\tilde{\omega}^\top(I - P_{\tilde{V}})\tilde{\omega}] + \sigma_\epsilon^2 \;=\; \mathrm{tr}(M(I - P_{\tilde{V}})) + \sigma_\epsilon^2.\end{aligned}$$

The cross term vanishes because ϵ is zero-mean and independent of (ω, x^*). The second-to-last equality uses idempotency, $(I - P_{\tilde{V}})^2 = I - P_{\tilde{V}}$, and the last uses $\mathbb{E}_\omega[\tilde{\omega}^\top A \tilde{\omega}] = \mathrm{tr}(\mathbb{E}[\tilde{\omega}\tilde{\omega}^\top]A)$ for symmetric A. Subtracting $\sigma_\tau^2 = \sigma_\epsilon^2$ gives the claim. □

What the best representation retains. **Proposition 2.** *Let* $M = U\Lambda U^\top$ *with* $\Lambda = \mathrm{diag}(\lambda_1 \geq \cdots \geq \lambda_d \geq 0)$. *Then*

$$\min_{V \in \mathbb{R}^{m \times d}} \mathrm{tr}(M(I - P_{\tilde{V}})) = \sum_{i>m} \lambda_i,$$

attained by any V *whose whitened row space* $\tilde{V} = V\Sigma_x^{1/2}$ *spans the top-*m *eigenvectors of* M.

Proof. Since $\mathrm{tr}(M(I - P_{\tilde{V}})) = \mathrm{tr}(M) - \mathrm{tr}(MP_{\tilde{V}})$, minimizing the former is maximizing $\mathrm{tr}(MP_{\tilde{V}})$, subject to $\mathrm{rank}(P_{\tilde{V}}) = \mathrm{rank}(\tilde{V}) \leq m$.
Put $Q := U^\top P_{\tilde{V}} U$. It is again an orthogonal projection — symmetric and idempotent, $Q^\top = Q = Q^2$ — of the same rank. With $w_i := Q_{ii}$,

$$\mathrm{tr}(MP_{\tilde{V}}) = \mathrm{tr}(U\Lambda U^\top P_{\tilde{V}}) = \mathrm{tr}(\Lambda Q) = \sum_{i=1}^{d} \lambda_i w_i.$$

Because Q is an orthogonal projection, $w_i \in [0, 1]$ and $\sum_i w_i = \mathrm{tr}(Q) = \mathrm{tr}(P_{\tilde{V}} U U^\top) = \mathrm{rank}(P_{\tilde{V}}) \leq m$. The sum $\sum_i \lambda_i w_i$ is therefore at most $\sum_{i \leq m} \lambda_i$, attained at $w_1 = \cdots = w_m = 1$ and $w_{m+1} = \cdots = w_d = 0$, that is $Q = \mathrm{diag}(I_m, 0)$ and $P_{\tilde{V}} = U\,\mathrm{diag}(I_m, 0)\,U^\top$. This is exactly the projection onto the span of the top-m eigenvectors of M, and the minimum value is $\mathrm{tr}(M) - \sum_{i \leq m} \lambda_i = \sum_{i>m} \lambda_i$. □

Why a second principle is needed. Proposition 2 determines a *subspace*, not a basis: replacing $\tilde{V}$ by $R\tilde{V}$ for any invertible $R \in \mathbb{R}^{m \times m}$ leaves $P_{\tilde{V}}$ unchanged, and with it the loss. The large-l pressure of Section 2.2 therefore fixes what the representation must retain and says nothing about how those m directions are arranged into groups. Choosing the arrangement is the separate job of predictive disentanglement at small l — the same invariance that makes $\bar{y}^*$ independent of how ψ organizes γ.

B Experiment details

B.1 Architecture

Every model shares the layout of Section 3.2: a token embedding, a stack of blocks each pairing an inner learner with a SwiGLU feed-forward network, and a linear decoder. Embeddings are untied. Table 4 gives the settings held fixed at each of the two scales.

	1.5B	8B
model dimension d	2 048	4 096
layers	24	32
head dimension	128	128
heads	16	32
FFN hidden dimension	5 632 (2.75 d)	14 336 (3.5 d)
vocabulary (Mistral-7B-v0.1)	32 000	32 000

Table 4: Architecture settings shared by all models at each scale. The feed-forward width follows the Llama SwiGLU convention (see text), and the input and output embeddings are not tied.

The feed-forward width follows the Llama SwiGLU convention: a nominal $4d$ expansion is multiplied by 2/3 to compensate for SwiGLU's three weight matrices, then by a size-dependent factor (1.0 at 1.5B, 1.3 at 8B) and rounded up to a multiple of 256. The hidden dimension is therefore 2.75 d and 3.5 d, not $4d$.

Position encoding. ROPE splits each head into $d_h/2$ channel pairs and rotates pair i of the query and key by an angle $t\,\omega_i$ at position t, with $\omega_i = \theta^{-2i/d_h}$ and $i = 0, \ldots, d_h/2 - 1$, where d_h is the head dimension.

Wherever ROPE is used, its base θ is scaled linearly with the span the layer attends over, at a fixed ratio θ/context length ≈ 122.

This applies to the short sliding window (256), to the sliding-window blocks of the hybrids (2k), and to the Transformer of Table 2 at each of its training lengths. The Transformer baseline is therefore given a correctly rescaled ROPE at every l, so its behaviour in Table 2 is not an artifact of a mistuned θ.

Hybrids. The hybrids interleave local and global blocks 1:1: even-numbered layers are local, odd-numbered layers are full attention without position encoding, so layer 0 is local and the last layer is full attention. That gives 12 of each at 1.5B and 16 of each at 8B. The local block differs by family: *Hybrid* SWA uses sliding-window attention with ROPE, while *Hybrid DeltaNet* uses a recurrence and therefore carries no position encoding at all.

The + and two-stage models. The short sliding window is fixed at $w = 256$. The long-range inner learner skips s recent positions, with s drawn uniformly from $[64, 256]$ during training and fixed at 64 at inference. In the two-stage models the same skip is applied to the related-task memory as well as to the target-task memory.

B.2 Training

All models are trained with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$) and weight decay 0.1, peak learning rate 3×10^{-4} with 200 warmup steps and a cosine decay to 1% of peak, and gradient clipping at 1.0. Weight decay is not applied to the learned bandwidth and damping coefficients, whose optima are not at zero. Training is in `bf16` under FSDP. Every run uses the same seed; no hyper-parameter is tuned per model.

Token budget. The 1.5B models train for 4000 steps and the 8B models for 8000, at a global batch of $2^{22} = 4.19$M tokens per step. For the training-length sweep of Table 1, the number of GPUs and the per-device batch are chosen together so that this global batch is identical at every l:

training length l	4k	8k	16k	32k	64k
GPUs	256	256	256	128	64
tokens / step			4.19M		

Every model in that comparison therefore sees the same number of tokens in the same number of optimizer steps, and only the length of the sequences differs.

Data. Documents are drawn from six sources in the proportions of Table 5. The percentages are shares of *documents*, not of tokens. Cosmopedia and NemotronCC-high mainly contain short documents, so their token contribution is well below their 94% share of the documents.

Source	% of documents
Cosmopedia (Ben Allal et al., 2024)	71.17
NemotronCC-high (Su et al., 2025)	22.89
open-web-math (Azerbayev et al., 2024)	4.77
arXiv (Azerbayev et al., 2024)	0.88
ShareGPT (Vicuna, 2023)	0.21
Gutenberg books (Manu, 2023)	0.07

Table 5: Training mixture, as a share of documents.

Documents are embedded with Nussbaum et al. (2024) and partitioned by k-means into 1200 clusters (1150 for training). Each training step draws from 32 clusters; within a step, a sequence is built from documents of a single cluster, delimited by `[bos]` and `[eos]`, concatenated and truncated at l.

B.3 Evaluation clusters

The same clustering procedure gives 50 clusters on Simple English Wikipedia. The corpus contains large families of near-template stubs — United States counties, Italian *comuni*, French communes — and k-means naturally collects each family into a cluster of its own:

> *Saint-Créac is a commune in the Gers department. It is in southwestern France. Related pages: Communes of the Gers department.*
>
> *Saint-Georges is a commune in the Gers department. It is in southwestern France. Related pages: Communes of the Gers department.*
>
> *Tourdun is a commune in the Gers department. It is in southwestern France. Related pages: Communes of the Gers department.*

We remove these clusters, identified from the documents alone and not from any model's predictions, to avoid trivial solutions. That leaves 19 clusters, used in all main-text SimpleWiki evaluations. Figure 6 repeats the comparison on all 50: the conclusion is unchanged.

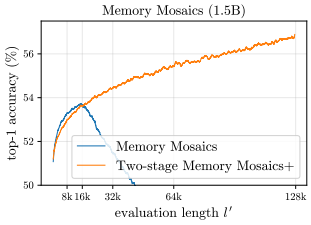


Figure 6: Memory Mosaics and two-stage Memory Mosaics+ on all 50 Simple English Wikipedia clusters, at 1.5B. As Figure 5, without the cluster selection.

C Evaluation on Simple English Wikipedia without clustering

The whole of Simple English Wikipedia is treated as one pool of related tasks, with no clustering. Only the evaluation changes. The training and the evaluation sequence construction are same as in Figure 5. The pattern is unchanged — two-stage models keep improving, one-stage models peak and decline — so the conclusion does not rest on the clustering step. The two figures cover different portions of the evaluation corpus, so their shapes are comparable and their magnitudes are not.

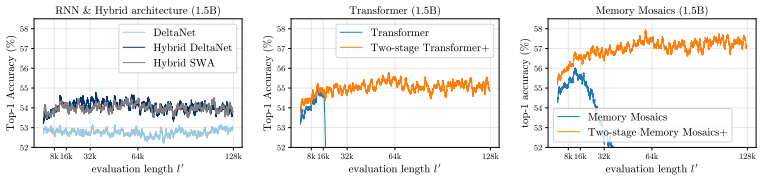


Figure 7: Accuracy as the number of related tasks in the context grows, at 1.5B. As Figure 5, but the whole of Simple English Wikipedia is treated as related tasks, with no clustering. **1)** RNN and hybrid architectures; **2)** Transformer versus two-stage Transformer+; **3)** Memory Mosaics versus two-stage Memory Mosaics+.

D Position encoding on the long-range attention

Transformer+ applies ROPE to the short sliding window and none to the long-range attention above it. Figure 8 repeats the setting of Figure 4 with a third curve, a Transformer+ whose long-range attention also carries ROPE. Transformer+ with ROPE is worse than Transformer+, confirming the correctness of using NOPE in the long-range attention, as in the main text.

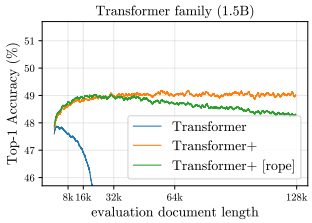


Figure 8: As Figure 4, at 1.5B, with a third curve: Transformer+ whose long-range attention also carries ROPE.

E What implements the short memory

Section 3.4 argues for a short-memory component to handle highly dependent local tokens, but says nothing about its type. Figure 9 sweeps that short-memory in Memory Mosaics+ over attention with ROPE, a Mosaics block with the leaky-average key extractor, and DeltaNet, in the setting of Figure 4. The curve "None" indicates no short-memory, i.e. Memory Mosaics. The other three curves coincide: what matters is a separated short-memory, not the particular implementation of the short memory.

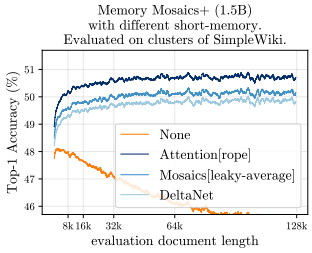


Figure 9: As Figure 4, at 1.5B, sweeping the implementation of short-memory of Memory Mosaics+: attention with ROPE, a Mosaics block with the leaky-average key extractor, and DeltaNet.

F The per-head gate

Table 6 shows the same comparison as Table 3, but adds a Memory Mosaics variant with the per-head sigmoid gate removed. It shows the same conclusion: Memory Mosaics still gains performance as G grows.

	Number of groups G (heads)					
Model	4	8	16	32	64	$\Delta > 0$
Memory Mosaics	48.43	48.63	48.97	**49.07**	49.02	✔
Memory Mosaics [no per-head gate]	48.15	48.29	48.46	**48.52**	48.47	✔
Transformer	48.31	**48.38**	48.05	47.92	47.08	✘
Hybrid SWA	**48.00**	47.95	47.97	47.67	46.85	✘

Table 6: Top-1 accuracy (%) on SimpleWiki, evaluated at length $l' = 4$k; all models trained at length $l = 16$k. As Table 3, with Memory Mosaics without the per-head gate added. Model width is fixed, so raising the number of heads raises the number of groups G and shrinks each group. Row maxima in bold. Δ is (64 heads − 4 heads).

G The implicit balancing weight

The two-stage models never compute the optimal weight $\lambda^* = (v_r + \delta^2)/(v_t + v_r + \delta^2)$ of Section 2.3; each long memory is normalized on its own, so the two contributions enter in proportion to their output magnitudes. That magnitude is informative because each memory is a normalized average of unit-norm values: its norm approaches 1 when the retrieved values agree and falls towards 0 when they disagree. Figure 10 plots the two against evaluation length. The target-task memory stays flat while the related-task memory grows — the

related evidence is becoming more concordant as it accumulates — so the balance shifts towards it with nothing computing a weight. This reproduces the direction in which λ^* moves as related tasks accumulate, though not its value; estimating the weight directly is left to future work.

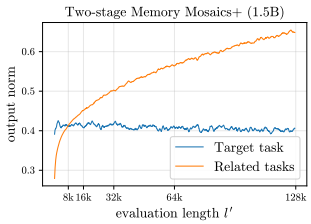


Figure 10: Output magnitude of the target-task and related-task long memories in two-stage Memory Mosaics+ at 1.5B, against evaluation length.