WHEN DOES DEPTH MATTER FOR IN-CONTEXT LEARNING? ADAPTIVE INFERENCE IN DEEP TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers perform computations through many successive attention and feedforward blocks, allowing them to learn complex correlations between a large collection of coupled variables. When does stacking successive attention-feedforward computations over many layers provide a computational advantage over a single transformer block? We address this question by examining in-context learning in generalized linear attention transformers. We first introduce a general theory of distributed inference in such transformers, subject to constraints on communication and depth. We show that such systems can exploit internal representations ('function vectors') to infer a latent context variable at increasingly finer scales over its layers. For an in-context linear regression task, the theory predicts that while one-layer transformers without feedforward blocks are optimal for Gaussian priors over the context variable, multi-layer transformers are superior for non-Gaussian, tree-like priors. Trained linear attention transformers reproduce quantitative predictions from the theory. Using causal key-patching experiments, we verify that function vectors in intermediate layers mediate adaptive routing of information. Our results suggest that depth and feedforward blocks enable transformers to implement adaptive inference, and this is advantageous when the distribution over latent variables has hierarchical structure.

1 INTRODUCTION

Often, one is interested in describing the joint distribution of a large collection of coupled random variables, $P(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N)$. Many classical techniques have been developed to estimate this distribution from data, often requiring simplifying ansätze that allow for efficient inference (Bishop, 2006). These techniques have been largely superceded in recent years by learning methods that optimize a flexible architecture using large datasets (Ortega et al., 2019; Müller et al., 2021; Genewein et al., 2026). The transformer architecture has emerged as one such general-purpose substrate (Vaswani et al., 2017), and now forms the basis of foundation models in many domains beyond language, including vision (Dosovitskiy et al., 2021; Radford et al., 2021; Kirillov et al., 2023), speech (Baevski et al., 2020; Hsu et al., 2021; Radford et al., 2023), materials design (Qu et al., 2024) and molecule generation (Irwin et al., 2022; Bagal et al., 2022; Chandra et al., 2023; Wei et al., 2023).

The flexibility of a transformer arises from two operations that are repeated across layers. The first is a global interaction, mediated by self-attention, in which each vector (or 'token') in the input sequence extracts information from other tokens through learned interaction maps. The second is a local nonlinear computation, usually implemented as a multi-layer perceptron (MLP), which acts independently and identically on each vector. Once its parameters are optimized, a transformer can adapt its predictions to a novel sequence or dataset in a single forward pass, without further updates to its parameters. This phenomenon, called in-context learning (ICL), suggests that the transformer has learned a generic procedure for inferring structure from context (Brown et al., 2020; Olsson et al., 2022). In-context learning has served as a paradigm for probing the internal mechanisms of transformers (Xie et al., 2022; Garg et al., 2022; Akyürek et al., 2023; Bai et al., 2023). We will use this paradigm for a similar purpose.

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.

While in large language models, inference is typically performed through many successive layers of attention and feedforward (MLP) blocks, mechanistic understanding of ICL is restricted to shallow models and typically without feedforward blocks (reviewed below). This leaves us with the following question: **when does depth provide a computational advantage?** Our hope is that addressing this question in controlled, simplified scenarios will help explain the remarkable success of deep models on practically relevant tasks. We approach this question for a class of architectures we call *generalized linear attention transformers*. Predictions from theory are tested on an in-context linear regression task using one derived variant from this class.

1.1 Related Work

Algorithmic and mechanistic accounts of ICL. A plethora of work has attempted to characterize the computation implemented by transformers during ICL. This work has identified two classes of motifs that implement ICL: (1) attention circuits that implement specific algorithmic operations, such as copying (Olsson et al., 2022; Reddy, 2023; Bietti et al., 2023) or gradient descent (Oswald et al., 2023; Ahn et al., 2023; Park et al., 2025; Zhang et al., 2024), and (2) by forming *function vectors* (Todd et al., 2024; Hendel et al., 2023; Hojel et al., 2024; Yin & Steinhardt, 2025), which are internal representations of the function or task the model deems is being solved. While recent theoretical efforts have shed light on the former class of mechanisms (Reddy, 2023; Lu et al., 2025; Ahn et al., 2023; Oswald et al., 2023; Zhang et al., 2024; 2025; He et al., 2025; Giannou et al., 2024), the transformer circuits that enable the latter have remained unclear. First identified in large language models (Hendel et al., 2023; Todd et al., 2024), function (or task) vectors are consistently found in the middle layers of billion-parameter models of different families, suggesting a common underlying principle that favors this organization. Recent empirical work (Gibson et al., 2026) finds that, in contrast to pure attention-based circuits, leveraging function vectors necessarily entails a multi-layer interaction between the attention and MLP blocks. The current work builds on function-vector-based ICL.

The role of depth in ICL. Recent work has characterized benefits of depth arising from iterative computation within in-context learning. Bordelon et al. (2026) show that deep linear attention transformers can help in-context linear regression when the input covariance varies across contexts, thus preventing effective preconditioning of the data before applying gradient descent. Depth can help by applying multiple steps of unconditioned gradient descent. Sun et al. (2026) show that MLPs can implement nonlinear regression by constructing polynomial-kernel features, so that in-context gradient descent operates on nonlinear features. Here, we propose a general theory that highlights how the structure of latent context variables influences the advantage offered by depth. We show that MLPs play other computational roles beyond nonlinear feature construction. In particular, MLPs use intermediate task representations to route which information is exchanged in subsequent attention layers and further act as decoders that apply the inferred function on a novel query.

1.2 Summary of Contributions

Here, we propose that depth enables a transformer to perform adaptive inference. Particularly, a shallow model must choose all of the information it extracts from the context simultaneously, while a deep model can refine later measurements based on information acquired at earlier layers. In this sequential layerwise approach, the accumulated function vector acts as a state variable, feedforward (MLP) blocks determine which statistic should be extracted next, and attention pools that statistic across the context. Depth can therefore enable an adaptive search over the hypothesis space, focusing computation on hypotheses which remain plausible given earlier observations. In summary, our main contributions are:

- **A framework for isolating the role of depth in ICL**. We formulate in-context inference under a fixed communication budget, allowing shallow and deep models to be compared on equal footing while controlling for the total amount of information extracted from the context.
- **A theory of when depth provides an advantage**. We frame layerwise inference as a sequential decision problem and show that depth is useful only when information acquired at one layer changes the choice of information extracted next.

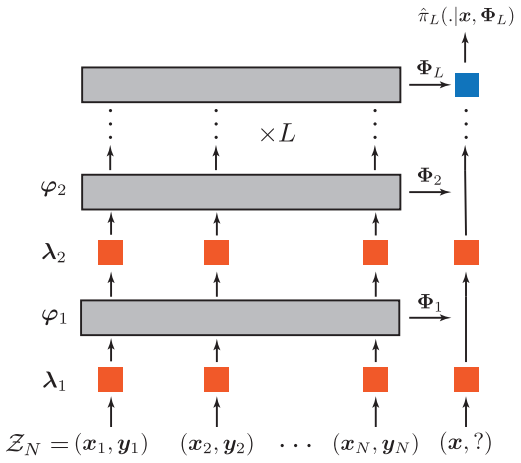


Figure 1: **Model schematic.** Information about a latent contextual variable $\boldsymbol{\theta}$ implicit in the context $\mathcal{Z}_N$ is compressed through L rounds of interaction into a function vector $\boldsymbol{\Phi}_L$, which is subsequently used by a query for prediction (through a decoder; blue square). At every layer $\ell+1$, the local nonlinearity (red square) uses $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and the intermediate function vector $\boldsymbol{\Phi}_\ell$ to construct an embedding function $\boldsymbol{\lambda}_{\ell+1}(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{\Phi}_\ell)$. The embedding of the N contextual tokens is pooled (gray rectangle) to form a d-dimensional statistic $\boldsymbol{\varphi}_{\ell+1}$. This pooled statistic is concatenated: $\boldsymbol{\Phi}_{\ell+1} = \boldsymbol{\Phi}_\ell \oplus \boldsymbol{\varphi}_{\ell+1}$.

- **A mechanistic realization in trained transformer models**. Deep transformers learn the predicted adaptive inference strategy, and causal key-patching experiments show that intermediate context representations control the routing decisions made at subsequent layers.

2 In-context learning in deep transformers

In-context learning. We wish to learn the joint distribution of a collection of N random variables ('tokens'), $\boldsymbol{Z}_N = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$. We consider a standard in-context learning paradigm (Garg et al., 2022; Ahn et al., 2023; Zhang et al., 2024; Lu et al., 2025). Specifically, the tokens are independent conditional on a latent context variable $\boldsymbol{\theta}$ (drawn from $\rho_0(\boldsymbol{\theta})$), which allows us to write

$$P(\{\boldsymbol{z}_i\}) = \int d\boldsymbol{\theta} \rho_0(\boldsymbol{\theta}) \prod_i p(\boldsymbol{z}_i|\boldsymbol{\theta}). \tag{1}$$

The model is trained to predict a masked-out component in a separate query token using the context $\boldsymbol{Z}_N$. Specifically, each token is split into two components, $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$, where $p(\boldsymbol{z}|\boldsymbol{\theta}) = q(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$. Given $\boldsymbol{Z}_N$ and an independently drawn query token $\boldsymbol{x}$ from the same $\boldsymbol{\theta}$, the model is trained to construct a predictive distribution $\hat{\pi}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{Z}_N) \approx \pi(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ by optimizing a prediction loss $\langle \mathcal{L}(\hat{\pi}) \rangle$. $\mathcal{L}$ is typically a cross-entropy loss or a mean-squared-error loss, and the expectation is over samples of $\boldsymbol{\theta} \sim \rho_0$, a new set of contextual tokens $\boldsymbol{Z}_N$ given $\boldsymbol{\theta}$, and a new query $\boldsymbol{x} \sim q(\cdot|\boldsymbol{\theta})$.

Generalized linear attention transformer. We consider transformer architectures that consist of L layers, where each layer includes one step of pooling interactions between tokens and one step of a local nonlinear operation that processes each token separately (Figure 1). Importantly, interactions are constrained to a dimension d. This constraint will allow us to compare architectures of different depths under a fixed communication budget. We assume a common masking scheme where the context tokens cannot interact with the query, whereas the query $\boldsymbol{x}$ can access the shared communication channel among the context tokens. Since a pooling operation is linear in the number of tokens N, the computational complexity of this architecture scales linearly with N. Thus, this architecture can be viewed as a *generalized linear attention transformer*. Indeed, we will see later that commonly used linear attention transformers are special cases with interaction dimension $d = d_K d_V$, where d_K, d_V are the key and value dimensions respectively (Wang et al., 2020; Katharopoulos et al., 2020; Schlag et al., 2021; Choromanski et al., 2022). We then show that *optimal* in-context learning with this architecture involves adaptive inference of the latent context variable $\boldsymbol{\theta}$.

One-layer model. Let's begin with a one-layer model, $L = 1$. Since $\boldsymbol{\theta}$ is a shared variable, the tokens should exchange information to perform inference. We assume interactions are mediated through a shared pool

$$\boldsymbol{\varphi}_1 = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\lambda}_1(\boldsymbol{z}_i), \tag{2}$$

where $\boldsymbol{\lambda}_1$ is a d-dimensional nonlinear embedding of each token and $\boldsymbol{\varphi}_1$ is a globally accessible statistic that carries information about $\boldsymbol{\theta}$. At optimality, the query $\boldsymbol{x}$ can access $\boldsymbol{\varphi}_1$ to form a posterior $\rho(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{\varphi}_1)$ and thereby generate a predictive distribution $\hat{\pi}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{Z}_N) = \int d\boldsymbol{\theta}\rho(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{\varphi}_1)\pi(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$. Since information about the context is shared to the query entirely through $\boldsymbol{\varphi}_1$, we define $\boldsymbol{\varphi}_1$ as the *function vector* corresponding to the context $\boldsymbol{Z}_N$. This definition of a function vector aligns with empirically motivated definitions of function/task vectors (Todd et al., 2024; Hendel et al., 2023; Gibson et al., 2026): an internal representation of a model is deemed a function vector if patching in the function vector from context $\boldsymbol{Z}_N$ into a different context $\boldsymbol{Z}'_N$ produces a response to a novel query $\boldsymbol{x}$ in context $\boldsymbol{Z}'_N$ that aligns with context $\boldsymbol{Z}_N$. In other words, a function vector is a compact summary of the information in the context for predicting the target $\boldsymbol{y}$ for any novel query $\boldsymbol{x}$.

Function vectors as internal state variables. Now, suppose $L \geq 1$. After the first ℓ layers, define

$$\boldsymbol{\Phi}_\ell = \boldsymbol{\varphi}_1 \oplus \boldsymbol{\varphi}_2 \oplus \cdots \oplus \boldsymbol{\varphi}_\ell, \qquad \boldsymbol{\Phi}_0 = \varnothing, \tag{3}$$

where $\oplus$ denotes a direct sum of vectors and the function vector at layer ℓ is $\boldsymbol{\varphi}_\ell = \sum_{i=1}^{N} \boldsymbol{\lambda}_\ell(\boldsymbol{z}_i, \boldsymbol{\Phi}_\ell)/N$. Crucially, $\boldsymbol{\Phi}_\ell$ acts as a state variable for implementing adaptive inference. To see this, consider the following heuristic argument. Let us say we want to adaptively select the embedding at the next layer $\boldsymbol{\lambda}_{\ell+1}$ to optimize the final prediction loss after L layers. A careful choice is necessary because of the communication constraint imposed by the finite dimension d of $\boldsymbol{\lambda}_{\ell+1}$. One may imagine, however, that making this choice requires keeping track of the embeddings at all preceding layers, $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_\ell$ (which, problematically, are functions rather than statistics). However, storing this history is not necessary in principle. Specifically, since $\boldsymbol{\Phi}_\ell$ preserves all statistics acquired so far, and because the embedding at each layer is chosen as a function of the previous state, $\boldsymbol{\Phi}_\ell$ *implicitly* records the choices of embeddings $\{\boldsymbol{\lambda}_{1:\ell}\}$ that produced those statistics.

Adaptive inference. We now examine how the state variable $\boldsymbol{\Phi}_\ell$ can be used to optimally choose $\boldsymbol{\lambda}_{\ell+1}$. Denote $\hat{\pi}_L(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Phi}_L) = \int d\boldsymbol{\theta}\, \rho(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{\Phi}_L)\pi(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ for the predictive distribution after the final layer, and define the terminal value $V_L(\boldsymbol{\Phi}_L) = \int d\boldsymbol{\theta}\, \rho(\boldsymbol{\theta}|\boldsymbol{\Phi}_L) \int d\boldsymbol{z}\, p(\boldsymbol{z}|\boldsymbol{\theta})\mathcal{L}(\hat{\pi}_L)$, which is the expected loss averaged over the posterior of possible latent context variables $\boldsymbol{\theta}$ and queries $\boldsymbol{x}$.

For $\ell < L$, say $\boldsymbol{\varphi}$ is the random statistic generated by a candidate next-layer embedding $\boldsymbol{\lambda}$, and let $P_{\boldsymbol{\lambda}}(\boldsymbol{\varphi}|\boldsymbol{\Phi}_\ell)$ denote its conditional distribution. $P_{\boldsymbol{\lambda}}(\boldsymbol{\varphi}|\boldsymbol{\Phi}_\ell)$ implicitly includes a complicated average over all possible contexts $\boldsymbol{Z}_N$ compatible with $\boldsymbol{\Phi}_\ell$. By the principle of dynamic programming (Bellman, 1957), the optimal value formally satisfies the equation:

$$V_\ell(\boldsymbol{\Phi}_\ell) = \min_{\boldsymbol{\lambda}} \langle V_{\ell+1}(\boldsymbol{\Phi}_\ell \oplus \boldsymbol{\varphi})\rangle_{\boldsymbol{\varphi}\sim P_{\boldsymbol{\lambda}}(\cdot|\boldsymbol{\Phi}_\ell)}\,. \tag{4}$$

This so-called Bellman equation is central to optimal control and appears here because one has to choose interactions at an intermediate layer to minimize the prediction loss using partial information from earlier layers. While this equation is challenging to solve analytically in most cases, it nevertheless shows that the optimal embedding $\boldsymbol{\lambda}_{\ell+1}$, which is the minimizer in Eq. 4, should be interpreted as an experimental design strategy for adaptive inference (Chaloner & Verdinelli, 1995; Chernoff, 1992). That is, the information exchanged by tokens at each layer should constrain the posterior distribution over $\boldsymbol{\theta}$ such that the model ultimately minimizes prediction loss at the final layer. Once $\boldsymbol{\lambda}_{\ell+1}$ is chosen, the model appends the corresponding statistic $\boldsymbol{\varphi}_{\ell+1}$ to the state, so that $\boldsymbol{\Phi}_{\ell+1} = \boldsymbol{\Phi}_\ell \oplus \boldsymbol{\varphi}_{\ell+1}$. After L layers, the query $\boldsymbol{x}$ uses $\boldsymbol{\Phi}_L$ to form $\hat{\pi}_L(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Phi}_L)$.

The role of attention and feedforward blocks at optimality. This discussion suggests that, if the transformer achieves optimality after training, then the feedforward blocks at convergence play at least two distinct computational roles. Their role depends on whether they are in intermediate layers or in the final layer: i) in the intermediate layers, they act as *routers* that choose which information in the token should be conveyed to other tokens, i.e., $(\boldsymbol{z}, \boldsymbol{\Phi}_\ell) \to \boldsymbol{\lambda}_{\ell+1}$, and ii) in the final layer, the MLP acts as a *decoder* that processes information gathered from other tokens to generate a prediction, $(\boldsymbol{x}, \boldsymbol{\Phi}_L) \to \hat{\pi}$. The attention operation mediates interactions and builds the function vector by pooling information across tokens, $\{\boldsymbol{\lambda}_\ell(\boldsymbol{z}_i, \boldsymbol{\Phi}_{\ell-1})\} \to \boldsymbol{\varphi}_\ell$. To achieve optimality, the

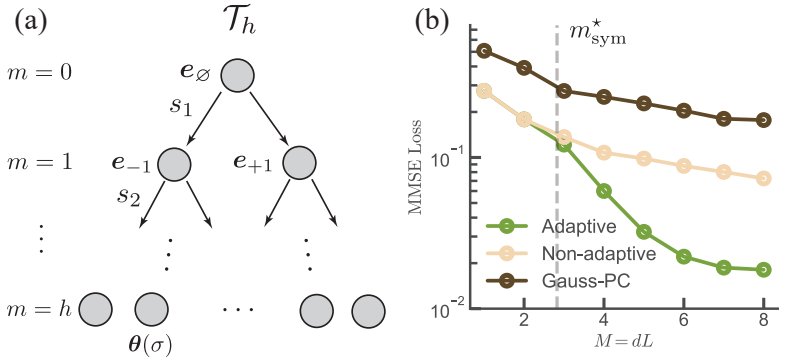


Figure 2: **Adaptive and non-adaptive inference strategies for the tree prior.** (a) Each internal node of the depth h tree is a coordinate. The context vectors $\boldsymbol{\theta}(\sigma)$ are weighted sums of coordinates along the path to a leaf $\sigma = s_1 s_2 \dots s_h$. (b) The MMSE loss from three strategies. Adaptive: A multi-layer strategy that adaptively infers the path towards the leaf by making a single measurement at every layer. Non-Adaptive: An optimized single-layer strategy with a measurement $\boldsymbol{W}$ whose rows are in the span of the depth-symmetric axes and the top $2^4 - 1$ eigenvectors of the prior covariance. Gauss-PC: The expected MMSE loss from a one-layer rank-constrained linear attention transformer without MLP blocks. Here, $h = 8, \alpha = 0.5, \tau = 0.1$. For these values, $m^\star_{\text{sym}} \approx 2.8$ and $m^\star_{\text{ada}} \approx 5.6$.

discussion further implies that the residual stream should be sufficiently large to retain information about the original token $\boldsymbol{z}_i$ and all the function vectors from previous layers. Motivated by this picture, we introduce a novel linear attention transformer architecture in Section 4 and discuss its relationship with commonly used architectures in Appendix B.5.

3 In-context linear regression and the role of depth

The adaptive inference framework allows us to derive a non-trivial relationship between the structure of the prior ρ_0 and the advantage offered by depth and MLP blocks in transformers. We examine this relationship in more detail in an in-context linear regression task (Garg et al., 2022). In each context, we generate tokens such that

$$y = \boldsymbol{\theta}^\top \boldsymbol{x} + \eta, \quad \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_K), \quad \eta \sim \mathcal{N}(0, \sigma^2), \quad \boldsymbol{\theta} \sim \rho_0. \tag{5}$$

For a context $\mathcal{Z}_N$ drawn from $\boldsymbol{\theta}$, the loss is $\mathcal{L} = (\boldsymbol{\theta}^\top \boldsymbol{x} - \hat{y}(\boldsymbol{x}, \mathcal{Z}_N))^2$, where $\hat{y}(\boldsymbol{x}, \mathcal{Z}_N)$ is the predicted output for query $\boldsymbol{x}$. For the theoretical analysis, we work in the limit $N \to \infty$ and the noise variance is scaled by fixing $\sigma^2/N = \tau^2$ to retain noise in this limit.

To derive predictions, we consider embeddings of the form $\boldsymbol{\lambda}_\ell = y \boldsymbol{W}_\ell \boldsymbol{x}$. This functional form for $\boldsymbol{\lambda}_\ell$ is motivated by the key-value outer product structure in linear attention architectures, which we will later use to test predictions. For this choice, the function vectors can be easily calculated:

$$\boldsymbol{\varphi}_\ell = \boldsymbol{W}_\ell(\boldsymbol{\theta} + \boldsymbol{\xi}), \quad \boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \tau^2 \boldsymbol{I}_K), \tag{6}$$

where $\boldsymbol{\xi}$ is due to the noise η and is fixed for all layers. One can interpret the rows of $\boldsymbol{W}_\ell$ as forming the subspace along which $\boldsymbol{\theta}$ is measured and the function vector $\boldsymbol{\varphi}_\ell$ as the outcome of the measurement. From the definition of $\boldsymbol{\Phi}_\ell$, we have $\boldsymbol{\Phi}_\ell = \boldsymbol{A}_\ell(\boldsymbol{\theta} + \boldsymbol{\xi})$, where $\boldsymbol{A}_\ell = \boldsymbol{W}_1 \oplus \boldsymbol{W}_2 \oplus \cdots \oplus \boldsymbol{W}_\ell$. Recall that for an adaptive strategy, $\boldsymbol{A}_\ell$ depends on $\boldsymbol{\Phi}_{\ell-1}$. After L layers, the network constructs $\boldsymbol{\Phi}_L$. If we denote $\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}_L}$ for the covariance of the Bayesian posterior $\rho(\boldsymbol{\theta}|\boldsymbol{\Phi}_L)$, then it can be shown that the Bayesian minimum mean squared error (MMSE) loss averaged over query tokens for a given $\boldsymbol{\Phi}_L$ is $\text{Tr}(\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}_L})$ (App. B.1). The average MMSE loss is $\langle \text{Tr}(\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}_L}) \rangle$, where the expectation is over contexts $\mathcal{Z}_N$. Next, we compare the MMSE loss of one-layer and multi-layer architectures while keeping the total communication budget $M = dL$ fixed. We perform this analysis for a Gaussian prior and a novel Tree prior over $\boldsymbol{\theta}$.

3.1 GAUSSIAN PRIOR

Consider a Gaussian prior with covariance $\boldsymbol{C}_0$, $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_0)$, for which we find (details in App. B.2)

$$\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}_\ell} = \boldsymbol{C}_0 - \boldsymbol{C}_0 \boldsymbol{A}_\ell^\top \left[\boldsymbol{A}_\ell (\boldsymbol{C}_0 + \tau^2 \boldsymbol{I}_K) \boldsymbol{A}_\ell^\top\right]^\dagger \boldsymbol{A}_\ell \boldsymbol{C}_0. \tag{7}$$

Crucially, the expression for $\mathrm{Tr}(\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}_L})$ does not depend on the function vector $\boldsymbol{\Phi}_L$ for a fixed measurement matrix $\boldsymbol{A}_L$. The expected posterior covariance is fully determined by the choices of which measurements the strategy chooses in subsequent layers rather than the observed outcomes of measurements. The optimal adaptive strategy at layer ℓ is thus to pick a $\boldsymbol{W}_{\ell+1}$ with d rows in the span of the top $d(L-\ell)$ eigenvectors of $\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}_\ell}$. However, the total MMSE loss from this adaptive strategy is identical to the non-adaptive strategy of choosing $\boldsymbol{A}_L$ from the top dL eigenvectors of $\boldsymbol{C}_0$ from the outset. Thus, given a communication budget M, L layers of interaction dimension d cannot outperform a single layer with interaction dimension dL. That a one-layer model is optimal for Gaussian priors was also shown in Bordelon et al. (2026).

3.2 TREE PRIOR

The property that the posterior covariance does not depend on the measurement is a special feature of Gaussian priors. The theory suggests that depth is useful in the opposite regime where the posterior covariance is strongly informed by the outcome of the measurement. Can one construct such a prior?

Consider a balanced binary tree $\mathcal{T}_h$ of depth h (Figure 2a), where each of the $K = 2^h - 1$ internal nodes represent orthogonal coordinates. Each leaf σ corresponds to a regression vector $\boldsymbol{\theta}$, which is specified by the path from the root of the tree to this leaf. Specifically, a leaf σ, represented as a sequence of spins $s_1 s_2 \cdots s_h$ ($s_m = \pm 1$), corresponds to an element in the prior:

$$\boldsymbol{\theta}(\sigma) = \kappa \sum_{m=0}^{h-1} \alpha^{m/2} s_{m+1} \boldsymbol{e}_{s_1 s_2 \ldots s_m}, \tag{8}$$

where $0 < \alpha < 1$, κ is a constant that normalizes $\boldsymbol{\theta}$, and $\boldsymbol{e}_{s_1 s_2 \ldots s_m}$ is the coordinate of internal node $s_1 s_2 \ldots s_m$. The prior is uniform over the 2^h leaves of the tree. Given this prior, we would like to predict the MMSE of one-layer and multi-layer models. According to our theory, this MMSE should correspond to the MMSE achieved by the best non-adaptive and adaptive strategies, respectively.

Non-adaptive strategies. Consider the non-adaptive strategy of choosing $\boldsymbol{A}_L$ as the top M eigenvectors of the prior covariance, as in the Gaussian case. The prior covariance is diagonal, and all coordinates at depth m have variance proportional to $(\alpha/2)^m$. The top eigenvectors are the shallowest tree coordinates. Because the multiplicity of eigenvalues at level m is 2^m, this strategy resolves spins up to tree depth $m^\star_{\text{PC}} \sim \log_2 M$, and the total posterior variance (and thus MMSE loss) scales as $\alpha^{m^\star_{\text{PC}}}$, namely, a power of M.

However, it turns out there are more effective non-adaptive strategies; for example, one can measure along axes that are symmetric with respect to the tree coordinates at a particular depth m ($0 \le m \le h-1$),

$$\boldsymbol{\mu}_m = 2^{-m/2} \sum_{v \in \mathcal{V}_m} \boldsymbol{e}_v, \tag{9}$$

where $\mathcal{V}_m$ are the 2^m nodes at depth m and the prefactor is chosen so that $\{\boldsymbol{\mu}_m\}_{m=0}^{h-1}$ are orthonormal. We have $\boldsymbol{\mu}_m^\top \boldsymbol{\theta}(\sigma) = \kappa(\sqrt{\alpha/2})^m s_{m+1}$. If $\boldsymbol{A}_L = \boldsymbol{\mu}_0 \oplus \boldsymbol{\mu}_1 \oplus \cdots \oplus \boldsymbol{\mu}_{M-1}$ ($M \le h$), then the function vector can be viewed as a noisy representation of σ in base $\sqrt{\alpha/2}$. Noise prevents resolving the spin s_{m+1} if $\kappa(\alpha/2)^{m/2} \lesssim \tau$, which means σ is resolved up to tree depth $m^\star_{\text{sym}} \approx 2\log(\tau/\kappa)/\log(\alpha/2)$. The posterior $\rho(\boldsymbol{\theta}|\boldsymbol{\Phi}_L)$ is concentrated on leaves that share the prefix until level $m^\star_{\text{sym}}$, which implies that the MMSE loss is $\sim \alpha^{\min(M, m^\star_{\text{sym}})}$, i.e., an exponential scaling with M and a cutoff at $m^\star_{\text{sym}}$. While this argument shows that there are more effective non-adaptive strategies, it still does not provide the optimal non-adaptive strategy. We instead resort to numerics (described in App. A.3.1) to compute near-optimal non-adaptive strategies for a given budget M.

Adaptive strategies. The optimal adaptive strategy is difficult to compute, but a simpler heuristic provides a useful upper bound to the MMSE and is sufficient to explain why depth is useful. Assume

$d = 1$ (so $L = M$). The heuristic strategy first measures the root coordinate $\boldsymbol{e}_{\varnothing}$. Now, if φ_1 is positive, the next measurement routes to the $+1$ child, and otherwise to the -1 child. Repeating this procedure layer by layer, the strategy measures the coordinate of the node selected by the previous signs and routes to the next child according to the sign of that noisy measurement. Thus, the M measurements trace a data-dependent path through the tree until depth M. Since $\boldsymbol{\theta}$ will be resolved up to depth M, the MMSE loss scales as α^M. Noise will eventually limit the level to which spins can be resolved. However, importantly, noise now acts only along a single axis. In this case, a spin cannot be resolved when $\kappa\alpha^{m/2} \lesssim \tau$, which leads to a cutoff $m^\star_{\text{ada}} \approx 2\log(\tau/\kappa)/\log\alpha$. Thus, we expect that the MMSE loss still decays exponentially as $\sim \alpha^{\min(M, m^\star_{\text{ada}})}$ with a higher cutoff $m^\star_{\text{ada}}$ than in the non-adaptive case.

Computing the MMSE loss. For the Adaptive and Non-Adaptive strategies, we first sample contexts $\boldsymbol{\theta} + \boldsymbol{\xi}$, obtain $\boldsymbol{A}_L, \boldsymbol{\Phi}_L$, estimate the posterior $\rho(\boldsymbol{\theta}|\boldsymbol{\Phi}_L)$ and average $\text{Tr}(\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}_L})$ over contexts to obtain the MMSE loss (App. B.1). For the Non-Adaptive strategy, we take $\boldsymbol{A}_L = \boldsymbol{U}\boldsymbol{D}$, $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_M$ and optimize $\boldsymbol{U}$. An exhaustive search requires $\boldsymbol{D}$ to contain all $2^h - 1$ coordinates, which is computationally challenging; we restrict the rows of $\boldsymbol{A}_L$ to the span of $\{\boldsymbol{\mu}_m\}$ and the top $2^g - 1$ eigenvectors of the prior covariance ($g = 4$ in numerics). We call Gauss-PC as the predicted loss from a one-layer linear attention transformer without MLPs (App. B.3), which depends on the spectrum of $\boldsymbol{C}_0$.

The MMSE losses for three strategies (labeled Adaptive, Non-Adaptive and Gauss-PC) are plotted in Figure 2b. An adaptive strategy should by construction have MMSE loss at most that of a non-adaptive strategy. Given a communication budget M, the heuristic Adaptive strategy splits from the Non-Adaptive strategy for $M > m^\star_{\text{sym}}$ (Figure 2b). Thus, the theory provides a specific prediction that a multi-layer transformer will outperform a one-layer model in this regime, which we now test.

4 Adaptive inference in deep linear attention transformers

Communication-constrained linear attention transformer. To test predictions from this model, we consider a communication-constrained L-layer linear attention transformer (Wang et al., 2020; Schlag et al., 2021). As discussed previously, the theory applies in a regime where the architecture can retain information about function vectors and the input; we therefore choose a model dimension $D \geq (K+1) + dL$. Denote $S = D - (K+1)$ and the residual stream activation at layer ℓ (where $1 \leq \ell \leq L+1$) and token index i as $\boldsymbol{u}_{\ell,i}$. Though not strictly necessary in practice, to maintain interpretability, we explicitly disentangle the input $\boldsymbol{z}_i$ and the 'internal state' $\boldsymbol{p}_\ell$ as $\boldsymbol{u}_{\ell,i} = \boldsymbol{z}_i \oplus \boldsymbol{p}_\ell$. In the first layer, we initialize $\boldsymbol{p}_1 = \epsilon \boldsymbol{1}_S$, where ϵ is a small constant. Denote $\boldsymbol{\phi}_\ell$ as the local feedforward block (MLP) at layer ℓ, which takes in $\boldsymbol{p}_\ell$ as input and outputs a $d \times (K+1)$ matrix $\boldsymbol{K}_\ell$. $\boldsymbol{Q}_\ell, \boldsymbol{V}_\ell, \boldsymbol{O}_\ell$ are $d \times S$, $d \times (K+1)$ and $S \times d$ matrices. For each layer ℓ, we iterate

$$\boldsymbol{\varphi}_\ell = \frac{1}{N}\sum_{j=1}^{N} (\boldsymbol{V}_\ell \boldsymbol{z}_j) \odot (\boldsymbol{K}_\ell \boldsymbol{z}_j), \quad \boldsymbol{K}_\ell = \bar{\boldsymbol{\phi}}_\ell(\bar{\boldsymbol{p}}_\ell),$$

$$\boldsymbol{p}_{\ell+1} = \boldsymbol{p}_\ell + \frac{1}{\sqrt{d}} \boldsymbol{O}_\ell (\boldsymbol{\varphi}_\ell \odot \boldsymbol{Q}_\ell \bar{\boldsymbol{p}}_\ell), \tag{10}$$

where the bar denotes row/layer normalization and $\odot$ represents elementwise multiplication. Finally, we pass the activation $\boldsymbol{u}_{L+1,q}$ at the query token through another feedforward block $\boldsymbol{\phi}^{\text{out}}$ to generate the output prediction: $\boldsymbol{u}_{\text{out},q} = \boldsymbol{u}_{L+1,q} + \boldsymbol{\phi}^{\text{out}}(\bar{\boldsymbol{u}}_{L+1,q})$. Eq. 10 is a linear attention transformer with three special features compared to common formulations: (1) the elementwise multiplication implements d heads that each convey a single scalar, allowing us to fix the interaction dimension to d, (2) the token representation $\boldsymbol{z}$ is never updated and is the only information conveyed by the attention heads, and (3) the MLP $\boldsymbol{\phi}_\ell$ reads from the internal state variable $\boldsymbol{p}_\ell$ and directly affects the key matrix, $\boldsymbol{K}_\ell$. Under our theoretical framework, we would expect the MLP $\boldsymbol{\phi}_\ell$ to read the internal state $\bar{\boldsymbol{p}}_\ell$ (where the function vectors from earlier layers are written) and decide on an appropriate measurement, here specified by $\boldsymbol{K}_\ell$.

4.1 Depth enables an adaptive search over the hypothesis space

Predictions from the theory are tested using the architecture in Eq. 10 in three different configurations A, B and C (Figure 4, with details in App. A.1). Model A has a standard arrangement of

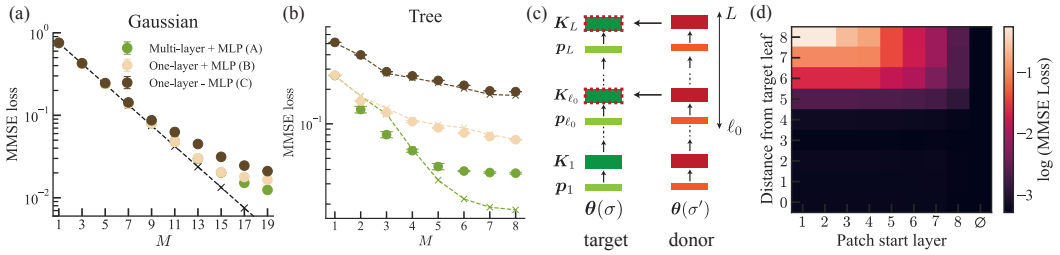


Figure 3: **Transformers learn adaptive inference and context-dependent routing.** (a) MMSE loss of Models A–C for a Gaussian prior. (b) MMSE loss for the tree prior. The dashed curves denote the theoretical strategies from Section 3. Error bars denote one standard deviation over training seeds. (c) Key-patching experiment where keys $\boldsymbol{K}_\ell$ from a donor leaf (σ') are patched into a target leaf (σ) at every layer $\ell \geq \ell_0$. (d) MMSE loss as a function of donor-to-target tree distance and the intervention layer ℓ_0. The rightmost column is the no-patch control and the bottommost row is a control where donor contexts are resampled from the target leaf σ.

interleaved attention and MLP blocks with $L = M$ layers and interaction dimension $d = 1$. Each $\boldsymbol{\phi}_\ell$ is an MLP and $\boldsymbol{\phi}^{\text{out}}$ is three stacked MLPs. Model B is a one-layer model with interaction dimension M with identical MLPs as Model A. All MLPs have a single hidden layer with GeLU activation. Model C is a standard one-layer linear attention model with interaction dimension M and no feed-forward blocks. It can be shown that Model C implements $\hat{y}(\boldsymbol{x}, \boldsymbol{Z}_N) = (\boldsymbol{W}(\boldsymbol{\theta} + \boldsymbol{\xi}))^\top \boldsymbol{x}$, where $\boldsymbol{W}$ is a rank-M matrix (Ahn et al., 2023; Zhang et al., 2024). Once optimized, $\boldsymbol{W}$ projects onto the span of the top M eigenvectors of $\boldsymbol{C}_0$ (App. B.3).

Figure 3a shows the results of training these models on a Gaussian prior ρ with covariance $\boldsymbol{C}_0 = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ where $\boldsymbol{\Lambda}$ is diagonal with $\boldsymbol{\Lambda}_{kk} \propto \alpha^k$ (up to a normalization constant) for $k = 1, 2, \ldots, K$ and $\boldsymbol{U}$ is a random orthogonal matrix (details provided in App. A.2). Consistent with the theory, all models achieve approximately the same loss: neither MLPs nor depth offer any advantage in this scenario.

Figure 3b shows the results for the tree prior. Consistent with theory, the MMSE loss of Models B and C aligns with the MMSE loss of the Non-Adaptive and Gauss-PC strategies, respectively. The remarkable match of the numerically computed MMSE loss for the Non-Adaptive strategy and the loss of Model B hints that MLP-based one-layer models do achieve optimality. Importantly, Model A outcompetes the optimized Non-Adaptive strategy, showing that the model indeed executes an adaptive inference strategy over its layers. However, the difference between the MMSE loss for Model A and the Adaptive strategy suggests that the network does not achieve optimality. We verified that this gap is not due to finite N effects or an insufficiently expressive MLP ($\boldsymbol{\phi}_\ell$). Thus, while deep models do appear to implement an adaptive strategy, there is further room for improvement. It is an interesting question whether an alternative architecture can achieve better performance under similar constraints.

4.2 Causal evidence for adaptive routing

As a direct causal test of the adaptive routing mechanism, we performed a key-patching experiment on Model A ($M = 8, h = 8$). If the model performs adaptive inference, the embeddings the model chooses for a context sampled from a leaf σ' should fare worse when used to infer the context variable of another leaf σ, and this difference should depend on the tree distance between σ and σ'. To test this hypothesis, for contexts sampled from the target leaf σ and the donor leaf σ' at distance d (from 0 to h), we first ran a forward pass of the donor context and recorded the dynamic key matrices $\{\boldsymbol{K}_\ell^{\sigma'}\}_{\ell=1}^M$. We then reran the target context from σ, but from patch start layer ℓ_0 onward replaced $\boldsymbol{K}_\ell^\sigma$ by $\boldsymbol{K}_\ell^{\sigma'}$ for all $\ell \geq \ell_0$, and measured the MMSE loss (Figure 3c). A model that does not implement an adaptive strategy should show no or little change in the MMSE loss across perturbations. Figure 3d shows that donor keys whose path diverges early from the target produce a large loss when inserted early. The effect is absent for donor leaves that are proximal to the target and for late or no interventions. The rightmost column $\ell_0 = \varnothing$ is a no-perturbation control.

5 DISCUSSION

A prominent view is that in-context learning emulates Bayesian inference over a latent context variable (Müller et al., 2021; Xie et al., 2022). Mechanistic studies of in-context linear regression (for example) have shown how attention circuits can implement algorithms such as preconditioned gradient descent (Ahn et al., 2023; Oswald et al., 2023; Zhang et al., 2024). We make progress towards bridging the gap between these two pictures by showing that deep transformers implement inference as a distributed, adaptive, layerwise procedure. Our framework suggests that function vectors act as compressed state variables, MLP blocks choose which statistic should be measured next, and attention pools that statistic across the context. This mechanism places function vectors (Todd et al., 2024; Hendel et al., 2023; Hojel et al., 2024; Yin & Steinhardt, 2025; Gibson et al., 2026) in a circuit-level framework and shows that depth and MLP blocks enable a much richer class of in-context learning algorithms than preconditioned gradient descent or copying.

The theory ignores a key element of typical transformer architectures, softmax attention, which allows tokens to retrieve information from specific tokens in the context at the expense of a quadratic scaling of computation with context size (Miller et al., 2016). Softmax attention is analogous to an implementation of a key-value memory over the context (Bricken & Pehlevan, 2021; Smart et al., 2025), and plays an important role in implementing attention-based circuit computations for in-context learning (Olsson et al., 2022; Bietti et al., 2023; Reddy, 2023; Gibson et al., 2026). The pooling-based operation in our theory is linear in context size while respecting exchangeability (Diaconis & Freedman, 1980; Zaheer et al., 2017; Edwards & Storkey, 2016), namely, permutation invariance of the data conditional on the latent context variable, and thus may be sufficient to implement in-context learning in such scenarios.

Recent analyses that treat self-attention as an interacting-particle system similarly emphasize a collective description of token dynamics, but focus on the evolution of the empirical distribution of token representations and the emergence of clustering or metastable macroscopic states (Geshkovski et al., 2023; Bruno et al., 2025; Sander et al., 2022). In a complementary direction, in-context denoising identifies regimes in which attention implements a context-dependent associative-memory or score-like denoising update (Smart et al., 2025; Rosu et al., 2026). Integrating these perspectives with the constrained Bayesian framework developed here may clarify how transformers combine local memory retrieval with adaptive inference using feedforward blocks.

AI USE STATEMENT

In this work, we used generative AI tools (OpenAI Codex, GPT-5.5) for coding assistance in writing experiments. We did not use generative AI tools for other any tasks requiring disclosure, including the development of theoretical models or conceptual frameworks, formulation or proof of mathematical claims, development or refinement of hypotheses, design of research methodology or experiments, or interpretation of results. Other required-disclosure tasks, including translation, dataset cleaning or reformatting, and qualitative or thematic data analysis, are not applicable to this work. Generative AI was used to check for grammatical and typesetting errors, but was not used to generate scientific results, theoretical claims, writing, or interpretations.

REPRODUCIBILITY STATEMENT

All code used to generate the results and figures of this work is available in the supplementary code files at `icl_function_vectors.zip`.

REFERENCES

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36:45614–45650, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8ed3d610ea4b68e7afb30ea7d01422c6-Abstract-Conference.html.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International*

Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12449–12460. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html.

Viraj Bagal, Rishal Aggarwal, P. K. Vinod, and U. Deva Priyakumar. MolGPT: Molecular Generation Using a Transformer-Decoder Model. *Journal of Chemical Information and Modeling*, 62(9):2064–2076, May 2022. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00600. URL https://doi.org/10.1021/acs.jcim.1c00600.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36:57125–57211, 2023.

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1957.

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 36: 1560–1588, 2023.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006. ISBN 9780387310732.

Blake Bordelon, Mary Letey, and Cengiz Pehlevan. Theory of scaling laws for in-context regression: Depth, width, context and time. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qA42mWsnbl.

Trenton Bricken and Cengiz Pehlevan. Attention approximates sparse distributed memory. *Advances in Neural Information Processing Systems*, 34:15301–15315, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Giuseppe Bruno, Federico Pasqualotto, and Andrea Agazzi. Emergence of meta-stable clustering in mean-field transformer models. In *International Conference on Learning Representations*, volume 2025, pp. 7496–7526, 2025.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical science*, pp. 273–304, 1995.

Abel Chandra, Laura Tünnermann, Tommy Löfstedt, and Regina Gratz. Transformer-based deep learning for predicting protein properties in the life sciences. *eLife*, 12:e82819, January 2023. ISSN 2050-084X. doi: 10.7554/eLife.82819. URL https://doi.org/10.7554/eLife.82819.

Herman Chernoff. Sequential design of experiments. In *Breakthroughs in Statistics: Methodology and Distribution*, pp. 345–360. Springer, 1992.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking Attention with Performers, November 2022. URL http://arxiv.org/abs/2009.14794. arXiv:2009.14794 [cs].

Persi Diaconis and David Freedman. Finite exchangeable sequences. *The Annals of Probability*, pp. 745–764, 1980.

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

Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pp. 30583–30598, Red Hook, NY, USA, November 2022. Curran Associates Inc. ISBN 978-1-7138-7108-8.

Tim Genewein, Jordi Grau-Moya, Li Kevin Wenliang, Laurent Orseau, and Marcus Hutter. Algorithmic compression via pretrained neural networks. *Entropy*, 28(6):596, 2026.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems*, 36:57026–57037, 2023.

Angeliki Giannou, Liu Yang, Tianhao Wang, Dimitris Papailiopoulos, and Jason D Lee. How well can transformers emulate in-context Newton's method? *arXiv preprint arXiv:2403.03183*, 2024.

Cole Gibson, Wenping Cui, and Gautam Reddy. Distinct mechanisms underlying in-context learning in transformers, April 2026. URL http://arxiv.org/abs/2604.12151. arXiv:2604.12151 [cs.LG].

Jianliang He, Xintian Pan, Siyu Chen, and Zhuoran Yang. In-context linear regression demystified: Training dynamics and mechanistic interpretability of multi-head softmax attention. *arXiv preprint arXiv:2503.12734*, 2025.

Roee Hendel, Mor Geva, and Amir Globerson. In-Context Learning Creates Task Vectors, October 2023. URL http://arxiv.org/abs/2310.15916. arXiv:2310.15916 [cs].

Alberto Hojel, Yutong Bai, Trevor Darrell, Amir Globerson, and Amir Bar. Finding Visual Task Vectors, April 2024. URL http://arxiv.org/abs/2404.05729. arXiv:2404.05729 [cs].

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. ISSN 2329-9304. doi: 10.1109/TASLP.2021.3122291. URL https://ieeexplore.ieee.org/document/9585401/.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, January 2022. ISSN 2632-2153. doi: 10.1088/2632-2153/ac3ffb. URL https://doi.org/10.1088/2632-2153/ac3ffb.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003, Paris, France, October 2023. IEEE. ISBN 979-8-3503-0718-4. doi: 10.1109/ICCV51070.2023.00371. URL https://ieeexplore.ieee.org/document/10378323/.

Yue M Lu, Mary Letey, Jacob A Zavatone-Veth, Anindita Maiti, and Cengiz Pehlevan. Asymptotic theory of in-context learning by linear attention. *Proceedings of the National Academy of Sciences*, 122(28):e2502599122, 2025.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1400–1409, 2016.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context Learning and Induction Heads, September 2022. URL http://arxiv.org/abs/2209.11895. arXiv:2209.11895 [cs.LG].

Pedro A Ortega, Jane X Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alex Pritzel, Pablo Sprechmann, et al. Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*, 2019.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers Learn In-Context by Gradient Descent. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 35151–35174. PMLR, July 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. ICLR: In-context learning of representations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pXlmOmlHJZ.

Jiaxing Qu, Yuxuan Richard Xie, Kamil M. Ciesielski, Claire E. Porter, Eric S. Toberer, and Elif Ertekin. Leveraging language representation for materials exploration and discovery. *npj Computational Materials*, 10(1):58, March 2024. ISSN 2057-3960. doi: 10.1038/s41524-024-01231-8. URL https://www.nature.com/articles/s41524-024-01231-8.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763. PMLR, July 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 28492–28518. PMLR, July 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. *arXiv preprint arXiv:2312.03002*, 2023.

Paul Rosu, Lawrence Carin, and Xiang Cheng. From softmax to score: Transformers can effectively implement in-context denoising steps. *Advances in Neural Information Processing Systems*, 38: 142994–143017, 2026.

Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pp. 3515–3530. PMLR, 2022.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pp. 9355–9366. PMLR, 2021.

Matthew Smart, Alberto Bietti, and Anirvan M Sengupta. In-context denoising with one-layer transformers: Connections between attention and associative memory retrieval. *Proceedings of Machine Learning Research*, 267:55950–55971, 2025.

J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992. doi: 10.1109/9.119632.

Haoyuan Sun, Ali Jadbabaie, and Navid Azizan. On the Role of Transformer Feed-Forward Layers in Nonlinear In-Context Learning. In *Proceedings of The 37th International Conference on Algorithmic Learning Theory*, pp. 1–3. PMLR, May 2026. URL https://proceedings.mlr.press/v313/sun26a.html.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Lai Wei, Nihang Fu, Yuqi Song, Qian Wang, and Jianjun Hu. Probabilistic generative transformer language models for generative design of molecules. *Journal of Cheminformatics*, 15 (1):88, September 2023. ISSN 1758-2946. doi: 10.1186/s13321-023-00759-z. URL https://doi.org/10.1186/s13321-023-00759-z.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference, July 2022. URL http://arxiv.org/abs/2111.02080. arXiv:2111.02080 [cs].

Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? *arXiv preprint arXiv:2502.14010*, 2025.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

Yedi Zhang, Aaditya K Singh, Peter E Latham, and Andrew Saxe. Training dynamics of in-context learning in linear attention. *arXiv preprint arXiv:2501.16265*, 2025.

A Experimental Details

A.1 Transformer Architecture and Training

We optimize all transformer models using AdamW with batch size 128, learning rate 3×10^{-4}, and weight decay 10^{-4}. The model dimension is $D = 2(K + 1)$, where K is the dimensionality of $\boldsymbol{x}$ and $\boldsymbol{\theta}$.

All feedforward blocks use a single hidden layer with Gaussian Error Linear Unit (GeLU) activations and hidden dimension D. The output block $\boldsymbol{\phi}_{\text{out}}$ consists of three stacked MLPs followed by a linear projection to a scalar output, which gives the prediction $\hat{y}(\boldsymbol{x}, \mathcal{Z}_N)$ for the query token.

We write the residual stream as $\boldsymbol{u}_{\ell,i} = \boldsymbol{z}_i \oplus \boldsymbol{p}_\ell$, where $\boldsymbol{z}_i \in \mathbb{R}^{K+1}$ contains the fixed token representation and $\boldsymbol{p}_\ell \in \mathbb{R}^S$ is the shared internal state, with $S = D - (K + 1)$. The internal state is initialized as

$$\boldsymbol{p}_1 = \epsilon \mathbf{1}_S, \tag{11}$$

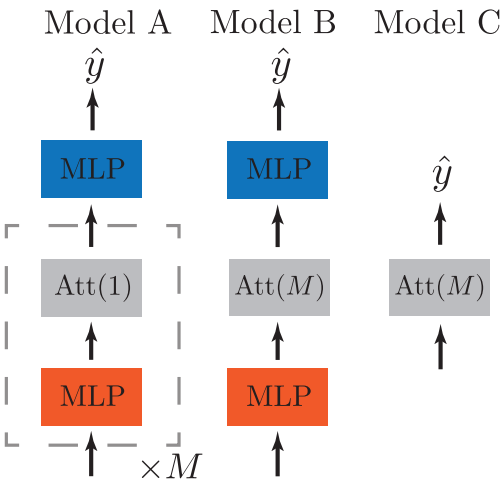


Figure 4: Three transformer configurations. A: M-layer model with interaction dimension 1 per layer with an MLP embedding function (orange) and an MLP decoder (blue). B: 1-layer model with interaction dimension M with an MLP embedding function and MLP decoder. C: 1-layer with interaction dimension M and no MLPs.

where ϵ is a small positive constant. For a model with interaction dimension d, the matrices $\boldsymbol{Q}_\ell$, $\boldsymbol{K}_\ell$, $\boldsymbol{V}_\ell$, and $\boldsymbol{O}_\ell$ have dimensions $d \times S$, $d \times (K+1)$, $d \times (K+1)$, and $S \times d$, respectively. In the implementation, the internal state supplied to the MLP and query branches is layer-normalized. Thus, the dynamic key matrix is constructed from the normalized state.

Query tokens are embedded in the same residual-stream representation as context tokens, except that their output component is masked. In particular, for a query token we set $\boldsymbol{z} = (\boldsymbol{x}, 0)$. For all transformer models trained in Sec. 4, the block diagrams of the models are shown in Fig. 4. All other architectural details follow the communication-constrained transformer defined in Appendix B.5.

A.2 Gaussian Prior Experiments

For the Gaussian-prior experiments, we sample

$$\boldsymbol{\theta} \sim \mathcal{N}(0, \boldsymbol{C}_0), \qquad \boldsymbol{C}_0 = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top, \tag{12}$$

where $\boldsymbol{U}$ is a random orthogonal matrix and $\boldsymbol{\Lambda}$ is diagonal with

$$\Lambda_{kk} = \frac{1-\alpha}{\alpha(1-\alpha^K)}\alpha^k, \qquad k = 1, \ldots, K. \tag{13}$$

For the results shown in Figure 3, we use $\alpha = 0.75$ and $\tau = 0.01$.

Each context contains $N = 4096$ context tokens, and the prediction loss for each context is evaluated using 256 query tokens. We use a relatively large value of N in these experiments to suppress finite-N effects, which otherwise introduce a lower bound to the achievable MMSE at larger communication budgets.

We train 8 independent seeds for each model configuration for 20000 iterations.

A.3 Tree Prior Experiments

For all tree-prior experiments, we use $h = 8$, $\alpha = 0.5$, and $\tau = 0.1$. For the trained transformer results in Figure 3, each context contains $N = 512$ context tokens, and the prediction loss is evaluated using 64 query tokens. We train 8 independent seeds for each model configuration for 40000 iterations.

To test whether the gap between the learned deep model and the heuristic adaptive strategy is caused by finite-context effects or insufficient model capacity, we repeated the experiments with $N = 1024$

and with additional layers in the layerwise MLPs. Neither modification produced a meaningful change in the MMSE loss.

For the theoretical strategy curves in Figure 2, the prior contains 2^h regression vectors in a $K = 2^h - 1$-dimensional space. For each measurement strategy, we estimate the expected MMSE by averaging the posterior variance over contexts and measurement noise. For a row-orthonormal measurement matrix $\boldsymbol{W}$ satisfying $\boldsymbol{W}\boldsymbol{W}^\top = \boldsymbol{I}_M$, the function vector is

$$\boldsymbol{\Phi} = \boldsymbol{W}(\boldsymbol{\theta}(\sigma) + \boldsymbol{\xi}), \qquad \boldsymbol{\xi} \sim \mathcal{N}(0, \tau^2 \boldsymbol{I}_K). \tag{14}$$

For a fixed $\boldsymbol{W}$, the posterior probability of leaf σ is

$$\rho_{\boldsymbol{W}}(\sigma|\boldsymbol{\Phi}) = \frac{\exp\left[-\|\boldsymbol{\Phi} - \boldsymbol{W}\boldsymbol{\theta}(\sigma)\|^2/(2\tau^2)\right]}{\sum_{\sigma'} \exp\left[-\|\boldsymbol{\Phi} - \boldsymbol{W}\boldsymbol{\theta}(\sigma')\|^2/(2\tau^2)\right]}. \tag{15}$$

The corresponding posterior mean is

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\Phi}) = \sum_{\sigma} \boldsymbol{\theta}(\sigma)\rho_{\boldsymbol{W}}(\sigma|\boldsymbol{\Phi}), \tag{16}$$

and the posterior MMSE is

$$\mathrm{Tr}\left(\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}}\right) = \sum_{\sigma} \left\|\boldsymbol{\theta}(\sigma) - \hat{\boldsymbol{\theta}}(\boldsymbol{\Phi})\right\|^2 \rho_{\boldsymbol{W}}(\sigma|\boldsymbol{\Phi}). \tag{17}$$

We enumerate all 2^h leaves exactly when evaluating the posterior and average this quantity over samples of the measurement noise. Since every leaf has unit norm, the posterior MMSE may equivalently be written as

$$\mathrm{Tr}\left(\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}}\right) = 1 - \left\|\hat{\boldsymbol{\theta}}(\boldsymbol{\Phi})\right\|^2. \tag{18}$$

A.3.1 Optimized non-adaptive baseline

An exhaustive optimization over all M-dimensional subspaces of the $K = 2^h - 1$ tree coordinates is computationally challenging. We therefore optimize over a restricted subspace that includes both the depth-symmetric directions and individual coordinates from the shallow levels of the tree. Specifically, define

$$\mathcal{H} = \mathrm{span}\left(\{\boldsymbol{e}_v : \mathrm{depth}(v) < g\} \cup \{\boldsymbol{\mu}_m\}_{m=0}^{h-1}\right). \tag{19}$$

The individual tree coordinates in the first set correspond to the top $2^g - 1$ eigenvectors of the prior covariance. We use $g = 4$ for the results in Figure 2. Let $\boldsymbol{D}$ be a row-orthonormal basis for $\mathcal{H}$. We parameterize the measurement matrix as

$$\boldsymbol{W} = \boldsymbol{U}\boldsymbol{D}, \qquad \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_M. \tag{20}$$

For each $\boldsymbol{U}$, we enumerate all leaves exactly and evaluate the posterior defined above. Since $\|\boldsymbol{\theta}(\sigma)\| = 1$, minimizing the expected MMSE is equivalent to maximizing

$$J(\boldsymbol{U}) = \left\langle \left\| \sum_{\sigma} \boldsymbol{\theta}(\sigma)\rho_{\boldsymbol{U}\boldsymbol{D}}\left(\sigma|\boldsymbol{U}\boldsymbol{D}(\boldsymbol{\theta}(\sigma_0) + \boldsymbol{\xi})\right) \right\|^2 \right\rangle_{\sigma_0, \boldsymbol{\xi}}. \tag{21}$$

We optimize $\boldsymbol{U}$ using simultaneous perturbation stochastic approximation (SPSA) subject to the row-orthonormality constraint $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_M$ (Spall, 1992). At each optimization step, for a random sign matrix $\boldsymbol{\Delta}$, we evaluate J at two re-orthonormalized perturbations of $\boldsymbol{U}$ and use their finite difference to construct a gradient estimate $\boldsymbol{G}$. The gradient is projected onto the tangent space of the row-orthonormal constraint using

$$\Pi_{\boldsymbol{U}}(\boldsymbol{G}) = \boldsymbol{G} - \mathrm{Sym}(\boldsymbol{G}\boldsymbol{U}^\top)\boldsymbol{U}, \tag{22}$$

where $\mathrm{Sym}(\boldsymbol{A}) = \frac{1}{2}\left(\boldsymbol{A} + \boldsymbol{A}^\top\right)$. We then update $\boldsymbol{U}$ along this projected direction and re-orthonormalize its rows. The SPSA step size and perturbation schedules are $a_k = \eta k^{-0.602}$ and $c_k = \epsilon k^{-0.101}$ respectively.

A.3.2 ADAPTIVE STRATEGY

For the heuristic adaptive strategy, we take $d = 1$ and simulate the measurements sequentially. The first measurement is taken along $\mathbf{e}_\varnothing$. At every subsequent step, the sign of the previous measurement determines which child coordinate is measured next. If

$$\hat{\sigma} = \hat{s}_1 \hat{s}_2 \cdots \hat{s}_M \tag{23}$$

denotes the path selected by these noisy measurements, the realized measurement directions are

$$\boldsymbol{W} = \boldsymbol{e}_\varnothing \oplus \boldsymbol{e}_{\hat{s}_1} \oplus \cdots \oplus \boldsymbol{e}_{\hat{s}_1 \cdots \hat{s}_{M-1}}. \tag{24}$$

For each sampled noise realization, we construct the resulting function vector, enumerate the posterior over all leaves, and compute the posterior MMSE. By symmetry of the tree prior, the noise average can be evaluated using contexts generated from a single fixed leaf.

A.3.3 GAUSS-PC BASELINE

The Gauss-PC baseline corresponds to the optimal rank-M linear estimator obtained from the prior covariance alone. If $\lambda_1 \geq \lambda_2 \geq \cdots$ are the eigenvalues of $\boldsymbol{C}_0$, its expected squared prediction loss is

$$\mathcal{L}_{\mathrm{Gauss-PC}} = \mathrm{Tr}(\boldsymbol{C}_0) - \sum_{i=1}^{M} \frac{\lambda_i^2}{\lambda_i + \tau^2}. \tag{25}$$

This baseline selects the top M eigenvectors of the prior covariance, as would be optimal for a Gaussian prior, but does not exploit the non-Gaussian hierarchical structure of the tree prior.

B THEORETICAL DETAILS

B.1 BAYESIAN POSTERIOR MMSE FOR IN-CONTEXT LINEAR REGRESSION

For the linear regression problem considered in the main text, the posterior mean estimator of the regression vector given a function vector $\boldsymbol{\Phi}$ is

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\Phi}) = \int d\boldsymbol{\theta}\, \boldsymbol{\theta}\, \rho(\boldsymbol{\theta}|\boldsymbol{\Phi}). \tag{26}$$

The corresponding Bayes-optimal prediction for a query $\boldsymbol{x}$ is

$$\hat{y}(\boldsymbol{x}, \boldsymbol{\Phi}) = \hat{\boldsymbol{\theta}}(\boldsymbol{\Phi})^\top \boldsymbol{x}. \tag{27}$$

Conditioned on $\boldsymbol{\Phi}$, the query-averaged squared prediction error is

$$\left\langle \left(\boldsymbol{\theta}^\top \boldsymbol{x} - \hat{y}(\boldsymbol{x}, \boldsymbol{\Phi})\right)^2 \right\rangle_{\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{\Phi}} = \left\langle \left[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{\Phi}))^\top \boldsymbol{x}\right]^2 \right\rangle_{\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{\Phi}} \tag{28}$$

$$= \left\langle \boldsymbol{x}^\top (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{\Phi}))(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{\Phi}))^\top \boldsymbol{x} \right\rangle_{\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{\Phi}}. \tag{29}$$

Using $\langle \boldsymbol{x}\boldsymbol{x}^\top \rangle = \boldsymbol{I}_K$, we thus obtain

$$\left\langle \left(\boldsymbol{\theta}^\top \boldsymbol{x} - \hat{y}(\boldsymbol{x}, \boldsymbol{\Phi})\right)^2 \right\rangle_{\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{\Phi}} = \mathrm{Tr}\left(\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}}\right), \tag{30}$$

where

$$\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}} = \left\langle (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{\Phi}))(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{\Phi}))^\top \right\rangle_{\boldsymbol{\theta}|\boldsymbol{\Phi}} \tag{31}$$

is the posterior covariance. Thus, minimizing the expected prediction error is equivalent to minimizing the expected trace of the posterior covariance.

B.2 GAUSSIAN PRIOR

Consider the Gaussian prior $\boldsymbol{\theta} \sim \mathcal{N}(0, C_0)$, and suppose that after ℓ layers the accumulated function vector takes the form

$$\boldsymbol{\Phi}_\ell = \boldsymbol{A}_\ell(\boldsymbol{\theta} + \boldsymbol{\xi}), \qquad \boldsymbol{\xi} \sim \mathcal{N}(0, \tau^2 \boldsymbol{I}_K), \tag{32}$$

where

$$\boldsymbol{A}_\ell = \boldsymbol{W}_1 \oplus \boldsymbol{W}_2 \oplus \cdots \oplus \boldsymbol{W}_\ell. \tag{33}$$

For a fixed choice of $\boldsymbol{A}_\ell$, $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}_\ell$ are jointly Gaussian, and therefore the posterior $\rho(\boldsymbol{\theta}|\boldsymbol{\Phi}_\ell)$ is also Gaussian. Its covariance is

$$\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}_\ell} = \boldsymbol{C}_0 - \mathrm{Cov}(\boldsymbol{\theta}, \boldsymbol{\Phi}_\ell)\, \mathrm{Cov}(\boldsymbol{\Phi}_\ell, \boldsymbol{\Phi}_\ell)^\dagger\, \mathrm{Cov}(\boldsymbol{\Phi}_\ell, \boldsymbol{\theta}). \tag{34}$$

The required covariance matrices are

$$\mathrm{Cov}(\boldsymbol{\theta}, \boldsymbol{\Phi}_\ell) = \boldsymbol{C}_0 \boldsymbol{A}_\ell^\top, \tag{35}$$

$$\mathrm{Cov}(\boldsymbol{\Phi}_\ell, \boldsymbol{\Phi}_\ell) = \boldsymbol{A}_\ell(\boldsymbol{C}_0 + \tau^2 \boldsymbol{I}_K)\boldsymbol{A}_\ell^\top. \tag{36}$$

Substituting these expressions gives

$$\boldsymbol{C}_{\boldsymbol{\theta}|\boldsymbol{\Phi}_\ell} = \boldsymbol{C}_0 - \boldsymbol{C}_0 \boldsymbol{A}_\ell^\top \left[\boldsymbol{A}_\ell(\boldsymbol{C}_0 + \tau^2 \boldsymbol{I}_K)\boldsymbol{A}_\ell^\top\right]^\dagger \boldsymbol{A}_\ell \boldsymbol{C}_0, \tag{37}$$

which is exactly the result used in the main text.

B.3 ONE-LAYER LINEAR ATTENTION BASELINE

We next derive the optimal prediction of the one-layer linear attention model without MLP blocks, corresponding to model C in Section 4. For a context summarized by $\boldsymbol{\theta}+\boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(0, \tau^2 \boldsymbol{I}_K)$, the model prediction can be written as

$$\hat{y}(\boldsymbol{x}; \boldsymbol{\theta} + \boldsymbol{\xi}) = \left[\boldsymbol{K}(\boldsymbol{\theta} + \boldsymbol{\xi})\right]^\top \boldsymbol{Q}\boldsymbol{x}, \tag{38}$$

where $\boldsymbol{Q}$ and $\boldsymbol{K}$ have rank at most M. Defining $\boldsymbol{B} = \boldsymbol{Q}^\top \boldsymbol{K}$, the query-averaged prediction loss is

$$\mathcal{L}_C(\boldsymbol{B}) = \left\langle \|\boldsymbol{\theta} - \boldsymbol{B}(\boldsymbol{\theta} + \boldsymbol{\xi})\|^2 \right\rangle_{\boldsymbol{\theta}, \boldsymbol{\xi}} \tag{39}$$

$$= \mathrm{Tr}\left[(\boldsymbol{I}_K - \boldsymbol{B})^\top (\boldsymbol{I}_K - \boldsymbol{B})\boldsymbol{C}_0\right] + \tau^2 \|\boldsymbol{B}\|_F^2, \tag{40}$$

subject to $\mathrm{rank}(\boldsymbol{B}) \le M$. Let

$$\boldsymbol{C}_0 = \sum_{i=1}^{K} \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i^\top, \qquad \lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_K. \tag{41}$$

Optimizing over rank-M matrices gives

$$\boldsymbol{B}^\star = \sum_{i=1}^{M} \frac{\lambda_i}{\lambda_i + \tau^2} \boldsymbol{e}_i \boldsymbol{e}_i^\top. \tag{42}$$

Thus, the optimal one-layer linear attention model measures the top M eigenvectors of the prior covariance, with noise-dependent shrinkage. Its minimum achievable loss is

$$\mathcal{L}_C^\star = \mathrm{Tr}(\boldsymbol{C}_0) - \sum_{i=1}^{M} \frac{\lambda_i^2}{\lambda_i + \tau^2}. \tag{43}$$

This is the *Gauss-PC* prediction used throughout the main text.

B.4 TREE PRIOR

For the tree prior introduced in Section 3, the normalization constant is

$$\kappa = \sqrt{\frac{1-\alpha}{1-\alpha^h}}, \tag{44}$$

such that $\|\boldsymbol{\theta}(\sigma)\| = 1$ for every leaf σ.

We next derive the spectrum of the prior covariance. A node at depth m lies on the path to a uniformly sampled leaf with probability 2^{-m}. Conditional on lying on the path, the corresponding component of $\boldsymbol{\theta}(\sigma)$ has magnitude $\kappa\alpha^{m/2}$. The variance of each coordinate at depth m is therefore

$$\kappa^2 \left(\frac{\alpha}{2}\right)^m. \tag{45}$$

Since there are 2^m coordinates at depth m, the prior covariance has a 2^m-fold degenerate eigenvalue proportional to $(\alpha/2)^m$ at each level, with eigenvalues decreasing from the shallowest to the deepest levels of the tree.

If the first m branch choices of a leaf are resolved exactly, the remaining posterior uncertainty is contained in the coordinates below depth m. Its total variance is

$$\kappa^2 \sum_{r=m}^{h-1} \alpha^r = \frac{\alpha^m(1-\alpha^{h-m})}{1-\alpha^h} \sim \alpha^m. \tag{46}$$

This gives the MMSE scalings used in Section 3.

B.5 Relation to Linear Attention Transformers

Here we show how the communication-constrained model introduced in Section 2 can be implemented by a linear attention transformer. Consider a residual stream activation $\boldsymbol{u}_{\ell,i} \in \mathbb{R}^D$ for token i at layer ℓ. A standard linear attention layer takes the form

$$\boldsymbol{u}_{\ell+1,i} = \boldsymbol{u}_{\ell,i} + \boldsymbol{O}_\ell \left[\left(\frac{1}{N} \sum_{j=1}^{N} (\boldsymbol{V}_\ell \boldsymbol{u}_{\ell,j})(\boldsymbol{K}_\ell \boldsymbol{u}_{\ell,j})^\top \right) \boldsymbol{Q}_\ell \boldsymbol{u}_{\ell,i} \right], \tag{47}$$

where $\boldsymbol{Q}_\ell$, $\boldsymbol{K}_\ell$, $\boldsymbol{V}_\ell$, and $\boldsymbol{O}_\ell$ are the learned query, key, value, and output matrices at layer ℓ respectively.

To restrict the interaction dimension to d, we consider d independent attention heads with one-dimensional query and key representations. Each head then communicates a single scalar statistic, and the attention update can then be written as

$$\boldsymbol{u}_{\ell+1,i} = \boldsymbol{u}_{\ell,i} + \boldsymbol{O}_\ell \left[\boldsymbol{\varphi}_\ell \odot (\boldsymbol{Q}_\ell \boldsymbol{u}_{\ell,i}) \right], \tag{48}$$

$$\boldsymbol{\varphi}_\ell = \frac{1}{N} \sum_{j=1}^{N} (\boldsymbol{V}_\ell \boldsymbol{u}_{\ell,j}) \odot (\boldsymbol{K}_\ell \boldsymbol{u}_{\ell,j}), \tag{49}$$

where $\odot$ denotes elementwise multiplication. The vector $\boldsymbol{\varphi}_\ell \in \mathbb{R}^d$ is thus a pooled statistic of the context, with each attention head contributing a single scalar communication channel. This has the same form as the pooled statistic introduced in Sec. 2.

To allow the statistic communicated at each layer to depend on information acquired in previous layers, we partition the residual stream as $\boldsymbol{u}_{\ell,i} = \boldsymbol{z}_i \oplus \boldsymbol{p}_\ell$ where $\boldsymbol{z}_i$ stores the original token and $\boldsymbol{p}_\ell$ is a shared internal state that evolves across layers. We also take $\boldsymbol{p}_\ell$ to have sufficient dimensionality to retain the statistics accumulated across previous layers. We constrain the attention operation so that only the fixed token representation $\boldsymbol{z}_i$ is communicated between tokens, while $\boldsymbol{p}_\ell$ controls how that information is processed. In particular, the feedforward block (MLP) at layer ℓ uses the current internal state to determine the key matrix $\boldsymbol{K}_\ell = \boldsymbol{\phi}_\ell(\boldsymbol{p}_\ell)$. Under this restriction, the pooled statistic defined above becomes

$$\boldsymbol{\varphi}_\ell = \frac{1}{N} \sum_{j=1}^{N} (\boldsymbol{V}_\ell \boldsymbol{z}_j) \odot (\boldsymbol{K}_\ell \boldsymbol{z}_j). \tag{50}$$

The pooled statistic is then written back into the shared internal state through

$$\boldsymbol{p}_{\ell+1} = \boldsymbol{p}_\ell + \frac{1}{\sqrt{d}} \boldsymbol{O}_\ell \left[\boldsymbol{\varphi}_\ell \odot (\boldsymbol{Q}_\ell \bar{\boldsymbol{p}}_\ell) \right], \tag{51}$$

where a factor of $1/\sqrt{d}$ is included as a form of width normalization for the residual update, ensuring that the magnitude of the communicated signal remains $\mathcal{O}(1)$ as the interaction dimension d is varied. Thus, $\boldsymbol{p}_{\ell+1}$ contains information acquired from the context at the current layer and can in turn determine the key matrix $\boldsymbol{K}_{\ell+1}$ used at the next layer.