MAKING TRANSFORMERS LOGICAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work has shown that transformers operating on logical tasks will often fit a simple function that interpolates between observed examples. This means that they often fail to generalize to unseen instances of logical boolean or arithmetic problems. Generalization on the Unseen (GOTU) has therefore become a stress test for understanding the inductive bias of transformers. In this paper, we demonstrate that poor logical extrapolation is not necessarily an inherent property of transformers, and that the desired behavior can be obtained by directly modifying this inductive bias in the network architecture. We show that including a simple attention mask is often sufficient to enable near-perfect extrapolation on boolean tasks and improves sample efficiency on arithmetic tasks. We show that although a similar bias may be encoded by initializing query and key parameters, a mask-based encoding is preserved during training. We additionally explore how a similar mechanism may be applied in vision. These results show the potential for using mask-based priors to encode logical inductive biases in transformers.

1 INTRODUCTION

Attention provides transformers (Vaswani et al., 2017) with a powerful mechanism for learning long-range token interactions directly from data. This flexibility has been central to their success in natural language processing (Vaswani et al., 2017; Radford et al., 2019; Guo et al., 2025) and computer vision (Dosovitskiy et al., 2020; Liu et al., 2021; Peebles & Xie, 2023). However, it can also make their inductive bias difficult to control. Recent work on generalization on the unseen (GOTU) (Abbe et al., 2024) illustrates this issue. The GOTU problem is a test of out-of-distribution generalization where data is divided into 'seen' and 'unseen' components. Abbe et al. (2024) show that where seen data points are fully learned by transformers there is a preference for a simple interpolative function (a minimum-degree interpolator) rather than equally plausible higher degree functions. Figure 2 highlights this issue: training on three points for the boolean parity problem leads to a simple decision boundary and misclassifies the unseen data point. This raises the question: **Can we change the inductive bias of transformers to improve performance on unseen examples**?

A natural way to approach this problem is to modify self-attention. Since self-attention controls how tokens interact, altering attention patterns provides a direct mechanism to encode structural priors for desired logical interactions. We therefore investigate whether enforcing a well-designed self-attention pattern can guide a Transformer to generalize across unseen portions of a logical Boolean function.

If a logical rule can be encoded in self-attention this raises the question of *how* we can encode this as a prior. While attention patterns can be shaped through masks (Duan et al., 2024; Vaswani et al., 2017; Roy et al., 2021), an alternative strategy would be to encode this interaction through a structured initialization (Trockman & Kolter, 2023; Zheng et al., 2025). Structured initialization methods encode a desired attention pattern by modifying the query and key projections, using either prior knowledge about the dataset (Zheng et al., 2025; Shinnick et al., 2026) or posterior observations from well-trained attention maps (Trockman & Kolter, 2023). However, since query and key projections must learn task-dependent representations during training, the injected structure may be washed out based on seen examples in a dataset. As a result, it is unclear whether structured initialization methods, even when encoding a known structural prior, can learn a desired rule or if it will instead default to a simple interpolative solution in the GOTU or arithmetic problem setting.

In this paper we first establish that the use of a self-attention mask with a known logical interaction can improve the performance of transformers on logical tasks with unseen examples in the GOTU setting.

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.

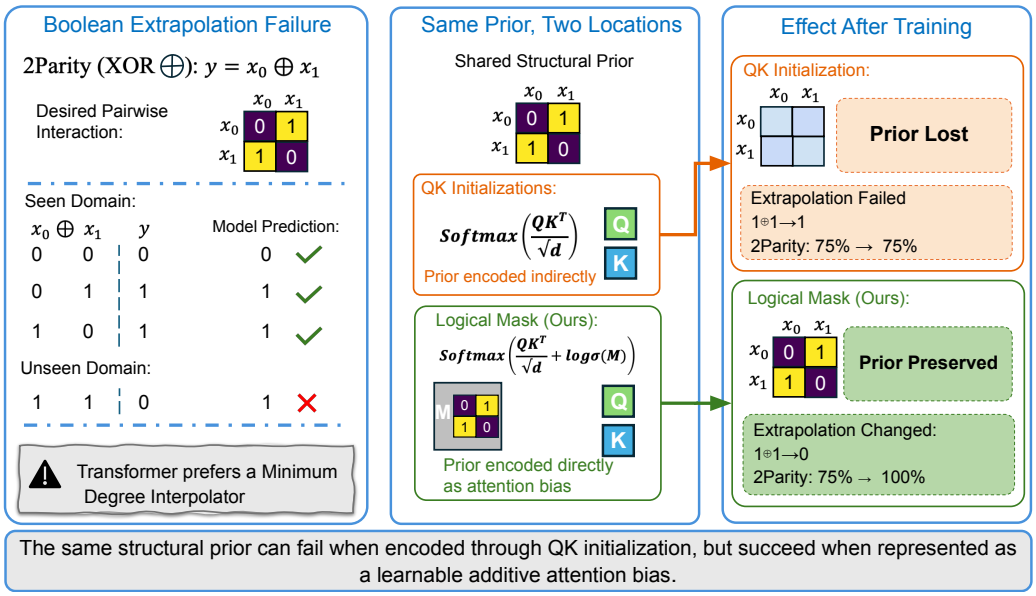


Figure 1: **Where the prior lives determines whether it changes extrapolation.** Boolean GOTU tasks expose a failure mode in which transformers fit the observed domain but extrapolate according to the wrong rule, often a simpler minimum-degree interpolator. We compare two ways of injecting the same structural prior into attention. QK-based initialization encodes the prior indirectly through content-dependent query-key projections, choosing Q and K so that $\mathrm{Softmax}(QK^\top/\sqrt{d})$ approximates the desired pattern. Mask-based initialization instead places the prior directly in the attention logits as a finite, learnable additive bias. After training, QK-based priors can be washed out, whereas mask-based priors remain explicit and persistent, leading to substantially improved extrapolation.

In particular, we demonstrate that where the prior is placed matters. QK-based structured initialization methods can produce the desired interaction pattern at initialization but do not necessarily preserve this information during training and do not reliably alter the extrapolation of the vanilla transformer. Secondly, we demonstrate that this strategy improves the sample efficiency of transformers on arithmetic tasks such as multiplication and addition. Finally, we ask whether the use of a learnable attention mask can have benefits for traditional vision problems and show that it has advantages over alternative ways to encode a structural prior such as through mimetic or QK initialization.

In summary, our main contributions are:

- We demonstrate that the tendency for transformers to learn minimum-degree solutions can be avoided by using appropriately designed attention masks. We show that this can enable higher-order interactions required to learn to solve logical problems in the GOTU setting.
- We show that although a similar inductive bias can be encoded at initialization through query and key parameters, that this does not persist over training. In contrast, using an attention mask provides a harder constraint and preserves this inductive bias during training.
- We provide a local gradient-level analysis for how inductive biases encoded in the attention mask are more persistent compared with initialization of query and key parameters.

We validate our approach for logical GOTU tasks including boolean and arithmetic problems. In addition, we demonstrate that encoding inductive biases through a masking approach has benefits over structured initialization approaches that extend to vision tasks. Our work highlights the potential for using mask-based priors for encoding specific logical inductive biases in transformers.

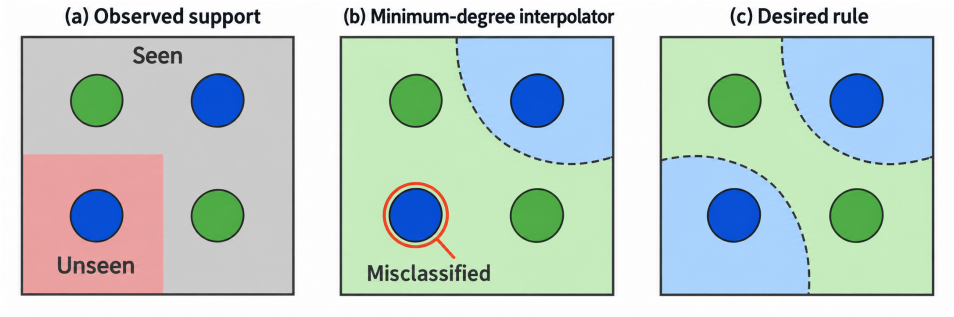


Figure 2: Generalization on the Unseen for the parity problem. (a) The model is trained on the three seen points. (b) The model converges to a simple decision rule (minimum degree interpolator) which perfectly classifies seen points but which misclassifies an unseen point requiring higher order interaction. (c) The correct decision rule that is not learned when converging to a MDI solution.

2 Related Works

2.1 Neural Network Initialization and Inductive Biases

Initialization methods have been widely explored in the deep learning community. Traditionally, they are designed to stabilize the training process, as deep networks often suffer from vanishing or exploding gradients (Glorot & Bengio, 2010; He et al., 2015). More recent work has focused on how initialization can influence learning efficiency. For instance, (Xu et al., 2023; Samragh et al., 2024; 2023) proposes using pretrained weights from large models to initialize smaller models. Similarly, (Trockman & Kolter, 2023) suggests transferring structural patterns from well-trained transformers as initialization, improving performance across a variety of tasks. Initialization is also widely applied for optimization stability, often by analyzing signal propagation under assumptions on pseudo-inputs such as Gaussian-distributed features (Sitzmann et al., 2020; Lindell et al., 2022; Kumar, 2017).

Inductive bias, which characterizes the generalization behavior of machine learning algorithms, has long been a central topic in the field. The inductive bias of transformers has been extensively studied, with many approaches seeking to explore the limits of extrapolation, generalization, and encoding priors through training(Abbe et al., 2024; Wang et al.; Zhong & Andreas, 2024; Pushkin et al.; Davidovich & Ringel, 2026; Teney et al., 2026; Shinnick et al., 2026). More recently, (Zheng et al., 2024; 2025) propose *structured initialization* to address the lack of inductive bias in standard transformer initialization, which typically yields unstructured attention maps. Their approach injects CNN-like locality into ViTs by solving for query and key matrices such that $\mathrm{Softmax}(QK^\top/\sqrt{d})$ approximates convolutional impulse patterns. This imposes spatial structure at initialization without modifying the architecture, leading to improved performance on small- and medium-scale datasets while maintaining competitive results at scale. However, this induced structure is implicitly encoded through query-key interactions, which may not preserve the desired inductive bias during training.

2.2 Attention Masks in Transformers

Attention masks, as a flexible and easily implemented component, play an important role in transformer architectures. The causal mask is a feature in the original autoregressive architecture, preventing the model from accessing future tokens and thereby avoiding shortcut solutions that could harm inference performance (Yin et al., 2024; Pei et al., 2025; Vaswani et al., 2017). Beyond causality, masks are used to restrict unwanted interactions in attention, particularly when data is limited in both size and dimensionality. This is especially important in vision models and tasks requiring length generalization (Zaheer et al., 2020; Beltagy et al., 2020; Tay et al., 2020; Child et al., 2019; Cheng et al., 2022b;a; Pang et al., 2019; Yu et al., 2022). Moreover, masking is a key tool for improving computational efficiency, as the quadratic $O(N^2)$ complexity of self-attention can be reduced by limiting the number of token interactions (Ho et al., 2019; Roy et al., 2021; Yuan et al., 2025). Related to our setting, Duan et al. (2024) use fixed, task-specific attention masks to enable length generalization for transformers applying arithmetic rules to longer inputs. In contrast, our

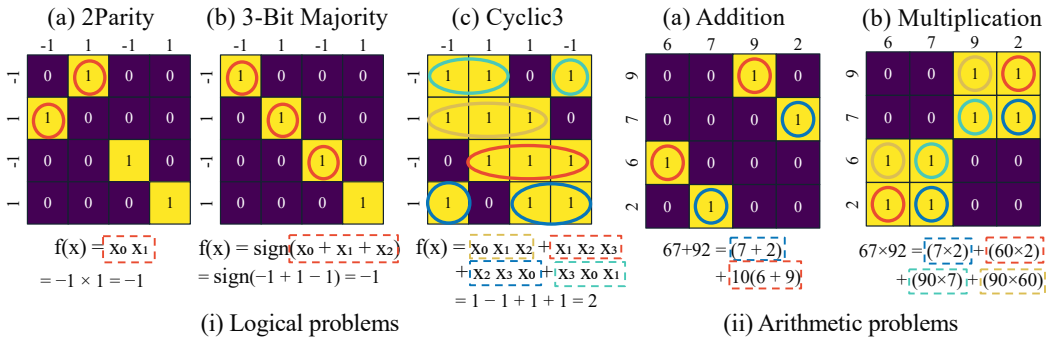


Figure 3: Visualization of task-specific mask design. **(i) Logical problems.** (a) 2Parity encodes pairwise interaction between the first two tokens; (b) 3-Bit Majority emphasizes aggregation over the first three tokens; (c) Cyclic3 captures cyclic triple interactions. **(ii) Arithmetic problems.** (a) Addition enforces digit-wise interactions between aligned positions; (b) Multiplication encodes structured cross-digit interactions corresponding to partial products. The masks define the interaction structure underlying each task, aligning attention with the functional decomposition.

work explore whether a learnable mask can enable extrapolation in the GOTU setting by encoding a persistent prior towards a higher degree interaction. Additionally, we demonstrate that how inductive biases are encoded in transformers matters, with alternative approaches to encode biases such as structured initialization failing to preserve desired attention patterns over training (see Figure 7).

3 Preliminary

Self-Attention Self-attention is a core component of transformer models (Vaswani et al., 2017). Given an input sequence $X \in \mathbb{R}^{N\times D}$, self-attention is parameterized by three learnable projection matrices: query $W_Q \in \mathbb{R}^{D\times d}$, key $W_K \in \mathbb{R}^{D\times d}$, and value $W_V \in \mathbb{R}^{D\times d}$. The output of a single attention head is defined as

$$A(X) = \text{softmax}\left(\frac{XW_Q W_K^\top X^\top}{\sqrt{d}}\right) XW_V, \tag{1}$$

where the softmax operator acts row-wise on the attention score matrix. In practice, multiple such heads are used in parallel and concatenated to form a multi-head attention layer.

Minimum-Degree Interpolator The notion of a minimum-degree interpolator (MDI) (Abbe et al., 2024) characterizes an inductive bias where neural networks prefer low-degree structure when fitting data. For Boolean functions $f : \{\pm 1\}^d \to \mathbb{R}$ with Fourier expansion

$$f(x) = \sum_{S\subseteq[d]} \hat{f}(S)\chi_S(x), \quad \chi_S(x) = \prod_{i\in S} x_i,$$

where $\hat{f}(S)$ is the coefficient for each polynomial and $[d]$ is the set $\{0, 1, \cdots, d\}$. Multiple functions may interpolate the observed data on Ω_{seen}. MDI selects the solution that explains the data using the lowest-order terms, favoring low-degree structure over higher-order interactions. As shown in (Abbe et al., 2024), transformers have a tendency to favor these low-degree solutions over the true higher-order functions for Boolean tasks such as 2Parity and 3-Bit Majority, representing a substantial limitation in the ability for transformers to perform extrapolation within logic problems.

4 Methods

4.1 Encoding Inductive Biases through a Logical Mask

For many structured tasks such as image classification and logical problems, prior knowledge about token interactions is often available. This knowledge can be expressed as a correlation map over

tokens, indicating which interactions should be encouraged or suppressed. Structured initialization methods (Zheng et al., 2025; 2024; Trockman & Kolter, 2023) aim to encode such prior knowledge into self-attention, typically by shaping the initial attention patterns through the query-key matrices.

In our approach, we directly encode prior structure as a learnable mask $M \in \mathbb{R}^{N\times N}$. The mask is initialized from manually constructed patterns that reflect the underlying problem structure. For example, Figure 3 illustrates how the desired inductive biases for Boolean and arithmetic tasks can be encoded as attention maps, where only relevant tokens (e.g., aligned digits or interacting variables) are allowed to attend. Entries corresponding to desired interactions are assigned active values, while undesired interactions are assigned smaller (more negative) values to suppress them. Through the transformation $\log\sigma(M)$, this induces a soft, learnable bias on the attention logits, enabling the model to retain the prescribed structure while adapting during training. The full formula is $\mathrm{softmax}\Big(\frac{XW_QW_K^\top X^\top}{\sqrt{d}} + \log\sigma(M)\Big)$, where $\sigma(\cdot)$ denotes the sigmoid function. Since $\log\sigma(M) \le 0$, this selectively down-weights undesired interactions while preserving the standard attention mechanism. During training, M remains learnable and is optimized jointly with the model parameters, allowing the encoded structural prior to persist throughout optimization.

4.2 Principles for Designing Logical Masks

Based on the previous mask methods (Vaswani et al., 2017; Duan et al., 2024), which blocks access to certain tokens to prevent spurious correlations and shortcut learning, we hypothesize that properly designed masks can similarly restrict irrelevant interactions in other tasks, particularly in low-data regimes where learning correlations is difficult, thereby improving generalization. To this end, we outline principles for mask-based initialization: the key idea is to encode prior knowledge about which token interactions are relevant for the task, while suppressing spurious correlations. As a simple example, consider the 3-bit majority task and its MDI solution. The non-MDI solution can be expressed as a global summation followed by a sign function, whereas the MDI solution relies on pairwise interaction terms between inputs. Therefore, as shown in Figure 3, we suppress such interaction terms by restricting attention, preventing the model from adopting the MDI solution and more uniform distributed attentions. This induces a structured attention pattern aligned with the underlying 3-Bit Majority algorithm. We visualize the corresponding mask patterns for different tasks in Figure 3, and provide more detailed design principles in Supplementary Materials Section F.

5 Experiments

5.1 Boolean GOTU Problems

Table 1: Test performance across Boolean tasks, reported as mean ± sample standard deviation over five random seeds. SVD Init and QK Opt denote task-pattern initialization via SVD (Zheng et al., 2025) and query-key optimization (Zheng et al., 2024), respectively. Bold indicates the best mean at the displayed precision, including ties.

Task	Metric	Default	Structured (SVD)	Structured (Opt)	Logical Mask (Ours)
2Parity	Loss ↓	1.41 ± 0.25	1.28 ± 0.09	1.36 ± 0.27	$\mathbf{0.01 \pm 0.02}$
	Seen Acc. (%) ↑	$\mathbf{100.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.00}$
	Unseen Acc. (%) ↑	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{100.00 \pm 0.00}$
Cyclic3	Loss ↓	4.00 ± 2.48	0.57 ± 1.01	4.12 ± 2.24	$\mathbf{0.01 \pm 0.00}$
	Seen Acc. (%) ↑	93.59 ± 4.19	94.95 ± 6.82	95.70 ± 2.27	$\mathbf{99.24 \pm 0.00}$
	Unseen Acc. (%) ↑	4.14 ± 5.07	59.06 ± 41.20	0.19 ± 0.41	$\mathbf{99.24 \pm 0.00}$
3-Bit Majority	Loss ↓	0.35 ± 0.22	0.23 ± 0.21	0.29 ± 0.26	$\mathbf{0.01 \pm 0.00}$
	Seen Acc. (%) ↑	$\mathbf{100.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.01}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.00}$
	Unseen Acc. (%) ↑	60.11 ± 22.30	64.15 ± 20.70	70.03 ± 27.36	$\mathbf{100.00 \pm 0.00}$
ThreeSym	Seen Loss (10^{-4}) ↓	$\mathbf{3.30 \pm 0.42}$	7.49 ± 3.09	3.98 ± 1.24	3.31 ± 1.44
	Unseen Loss ↓	14.98 ± 0.26	14.25 ± 1.10	15.11 ± 0.41	$\mathbf{5.44 \pm 0.21}$

We evaluate our method on four Boolean tasks under the Generalization on the Unseen (GOTU) setting Abbe et al. (2024) which are detailed described in Supplementary Section A.1. These tasks are important as they reflect the extrapolation and inductive bias properties of transformers. Performance

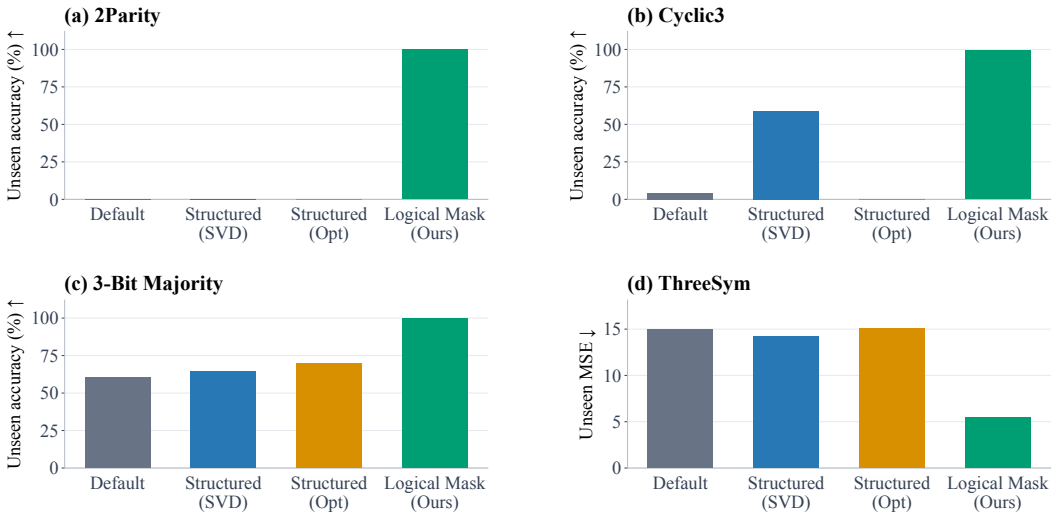


Figure 4: Test performance on Boolean reasoning tasks under an extrapolation setting. Our logical mask consistently achieves lower loss and higher accuracy compared to default and QK-based initialization methods, indicating improved extrapolation. Accuracy is not reported for ThreeSym due to its continuous outputs.

on these datasets provides a useful lens for evaluating whether a properly designed initialization, with the correct correlation patterns, can alter extrapolation behavior. Visualization and explanation of the initialization pattern can be found at Section F.

From Figure 4 and Table 1, we observe that initialization can indeed influence the inductive bias of transformers, as the GOTU setting explicitly evaluates extrapolation beyond the training domain. Moreover, even when using the same structural pattern, initialization through the query-key matrices tends to quickly forget the imposed prior during optimization, which leads to poor generalization on unseen data. In contrast, mask-based initialization preserves the prior and enables better extrapolation.

5.2 Arithmetic Problems

Arithmetic problems such as addition and multiplication follow simple and clear rules, but they are still difficult for neural networks to learn. For example, addition only requires learning basic operations such as adding single digits and handling carries. Once these rules are learned, they can be reused to solve many new examples. Therefore, we would expect a model to learn arithmetic from a relatively small number of training examples. However, neural networks often need much more data than expected, even to generalize to new examples with the same input format (Trask et al., 2018).

This makes arithmetic a natural test for sample efficiency: if a model can better use the underlying rules, it should require fewer training examples to generalize. We test whether using a logical mask improves in-distribution generalization in low-data settings. For both addition and multiplication, we enumerate all possible input pairs (10^6 combinations) and use only a fraction of them for training, with the remaining examples used for testing. We evaluate 3-by-3 digit decimal addition and multiplication, where all operands contain exactly three digits (including leading zeros), and the transformer predicts the full output sequence in a single forward pass. More details are provided in Section A.2.

As shown in Table 2, The logical mask consistently outperforms other methods on both multiplication and addition tasks. The is most pronounced in low-data regimes (e.g., addition with 0.4% of smamples), where other methods fail to generalize. As shown in Figure 5, although all methods improve with more data, the logical mask attains better generalization with fewer training samples. This provides evidence that a logical mask can accelerate learning an underlying algorithmic structure.

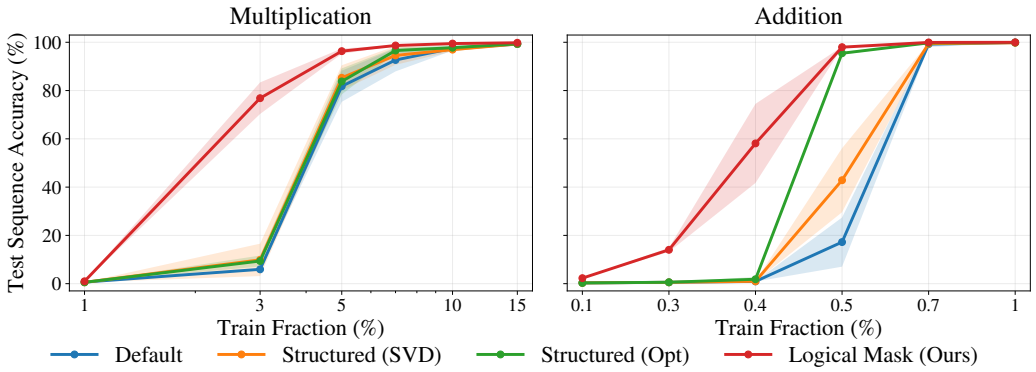


Figure 5: Test sequence accuracy under varying training fractions for multiplication (left) and addition (right). Logical mask improves data efficiency, achieving higher accuracy in low-data regimes.

Table 2: Test accuracy on arithmetic tasks, reported as mean ± sample standard deviation across five random seeds. Higher is better.

	Multiplication (Train frac = 3%)		Addition (Train frac = 0.4%)	
Method	Seq Acc (%) ↑	Digit Acc (%) ↑	Seq Acc (%) ↑	Digit Acc (%) ↑
Default	6.51 ± 0.81	71.61 ± 0.58	1.01 ± 0.19	39.49 ± 1.21
Structured (SVD)	10.04 ± 4.76	74.28 ± 3.13	1.10 ± 0.24	40.23 ± 1.60
Structured (Opt)	9.74 ± 1.69	73.96 ± 1.48	2.11 ± 0.90	44.68 ± 3.83
Logical Mask (Ours)	$\mathbf{77.51 \pm 7.51}$	$\mathbf{95.76 \pm 1.51}$	$\mathbf{60.37 \pm 16.00}$	$\mathbf{87.70 \pm 5.76}$

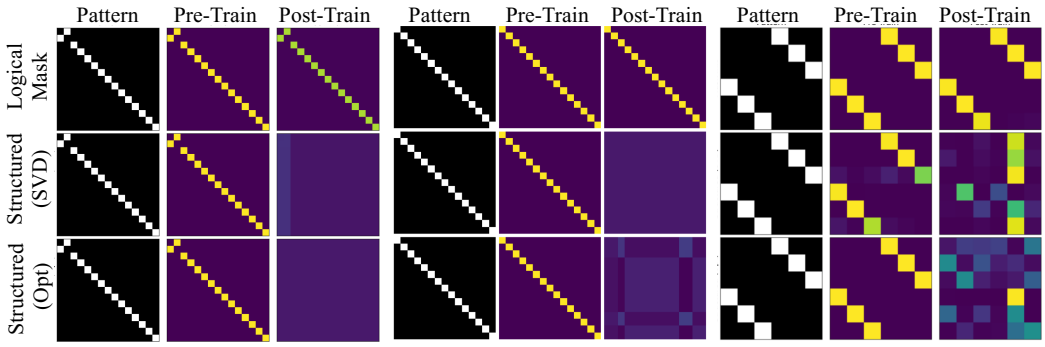
5.3 Image Classification

It is known that spatial priors for vision tasks can be encoded in attention through structured initialization (Zheng et al., 2025). Following (Zheng et al., 2025), we adopt an impulse filter pattern to initialize our logical masks (see Supplementary Figure Section H.1 for visualization of the desired pattern). In this section, we evaluate three initialization methods on CIFAR10, CIFAR100, and ImageNet100 (Russakovsky et al., 2015; Krizhevsky, 2009a). For comparison, we adopt the implementation from (Zheng et al., 2025), which provides the most up-to-date codebase. We include these benchmarks to demonstrate that the proposed patterns extend beyond toy settings and remain effective in practical scenarios. Moreover, these datasets align with the primary evaluation domain of prior structured initialization methods. We compare against mimetic initialization (Trockman & Kolter, 2023), which initializes attention close to an identity pattern with noise, reflecting a common structure observed in trained transformers, and the Impulse method (Zheng et al., 2025), which initializes each attention head to approximate a 3×3 impulse filter, resulting in a diagonal-like attention pattern. Both approaches inject locality as an inductive bias, which has been shown to improve generalization, particularly in low-data regimes.

Table 3: Best Top-1 accuracy (%) on vision benchmarks using ViT-Tiny. Mean ± sample standard deviation on three random seeds. For each seed, we report the highest accuracy during training.

Dataset	Default	Mimetic	Impulse	Logical Mask (Ours)
CIFAR10	93.93 ± 0.11	94.98 ± 0.26	95.03 ± 0.20	$\mathbf{95.95 \pm 0.12}$
CIFAR100	77.26 ± 0.22	79.01 ± 0.06	79.62 ± 0.35	$\mathbf{82.06 \pm 0.14}$
ImageNet100	83.47 ± 0.13	84.55 ± 0.14	83.51 ± 0.95	$\mathbf{85.83 \pm 0.56}$

As shown in Table 3, our logical mask consistently achieves the best performance, with higher accuracy than baseline and structured methods, indicating improved optimization and generalization. Additional visualizations are provided in Sections H.2 and H.3.



(a) 2Parity (b) 3-Bit Majority (c) Addition

Figure 6: Comparison of attention patterns before and after training for Head 0 at Layer 0. Our logical mask preserves the target interaction structure more closely after training, whereas QK-based structured initializations are not preserved despite closely matched initial attention patterns.

6 Persistence of Prior Knowledge during Optimization

In this section, we examine why structural priors encoded through additive masks are more persistent during optimization than those encoded through structured QK-initialization by examining gradient effects of the self-attention. The key distinction between QK initialization and our logical mask is that a mask represents the prior through an additive pathway in the attention logits, whereas a QK-based initialization is embedded in the query and key parameters that can overfit to seen examples during training. Consequently, changes to these representations can alter the encoded attention structure even when the desired pattern is well matched at initialization. To characterize the difference between these two pathways, we analyze the local sensitivity of the attention probabilities with respect to queries, keys, and mask parameters.

Theorem 1. *Consider a single attention row with query $q \in \mathbb{R}^d$, keys $\{k_j\}_{j=1}^N$, and mask parameters $\{m_j\}_{j=1}^N$. Define*

$$p_j = \frac{\exp(q^\top k_j/\sqrt{d} + \log \sigma(m_j))}{\sum_{\ell=1}^N \exp(q^\top k_\ell/\sqrt{d} + \log \sigma(m_\ell))}.$$

Then, for any $i, j \in \{1, \dots, N\}$,

$$\frac{\partial p_i}{\partial q} = \frac{p_i}{\sqrt{d}} \left(k_i - \sum_{\ell=1}^N p_\ell k_\ell \right), \tag{2}$$

$$\frac{\partial p_i}{\partial k_j} = \frac{1}{\sqrt{d}} p_i(\delta_{ij} - p_j) q, \tag{3}$$

$$\frac{\partial p_i}{\partial m_j} = p_i(\delta_{ij} - p_j)\sigma(-m_j). \tag{4}$$

Corollary 1. *Under the setting of Theorem 1, suppose $\|q\|_2 \le B_q$ and $\|k_j\|_2 \le B_k$ for all j. Then*

$$\left\| \frac{\partial p_i}{\partial q} \right\|_2 \le \frac{2 p_i B_k}{\sqrt{d}}, \tag{5}$$

$$\left\| \frac{\partial p_i}{\partial k_j} \right\|_2 \le \frac{p_i |\delta_{ij} - p_j| B_q}{\sqrt{d}} \le \frac{p_i B_q}{\sqrt{d}}, \tag{6}$$

$$\left| \frac{\partial p_i}{\partial m_j} \right| \le p_i |\delta_{ij} - p_j| \sigma(-m_j) \le p_i. \tag{7}$$

The Corollary above highlight a difference in local sensitivity: query-key gradients depend on the feature scales $\|q\|_2$ and $\|k_j\|_2$, whereas mask gradients do not and are further moderated by $\sigma(-m_j) \le 1$. For strongly encouraged interactions, large positive m_j reduces this sensitivity, while

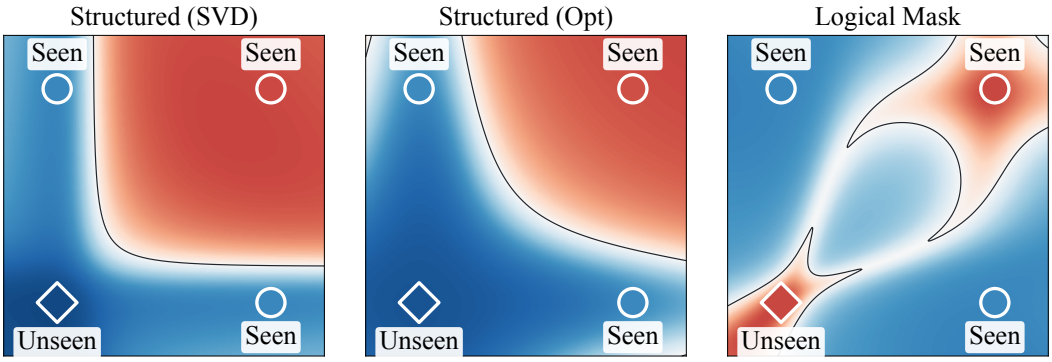


Figure 7: Decision boundaries on the GOTU 2Parity problem. While all three methods start with the same structured attention pattern, both of the structured initalization methods (SVD (Zheng et al., 2025) and Opt (Zheng et al., 2024)) converge to a MDI solution. In contrast, the logical mask is more persistent during training enabling it to achieve the desired higher-order interaction for extrapolation.

strongly suppressed interactions typically have small attention probability p_j. These properties provide a gradient-level mechanism consistent with the greater empirical persistence of mask-based priors, rather than implying universally smaller optimizer updates. Proofs are provided in Section C. We also verify this persistence directly by tracking the attention pattern during training. As shown in Figure 6 and Section E, the additive-mask parameterization retains the target interaction structure substantially more closely, whereas the attention patterns produced by QK-based structured initialization drift away from their initialized structure. Importantly, this difference is observed even when the two parameterizations produce closely matched attention patterns at initialization, suggesting that the effect is not simply due to better initial pattern fitting. The resulting difference in persistence is accompanied by substantially better extrapolation on the Boolean GOTU tasks.

7 Conclusion and Limitations

In conclusion, we have shown that applying an appropriately designed attention mask is able to encode a logical decision rule in transformers. As this mask encodes a persistent desired token interaction, this is able to avoid converging to the simple interpolative solution that present a difficult when training transformers on Boolean problems in a GOTU setting. Furthermore, we have shown that although a similar inductive bias can be encoded with a structured QK-based initialization, this initialization does not persist during the course of training. As a result, we have shown that where we encode an inductive bias matters, and that a bias encoded through a logical mask can be maintained with benefits across multiple tasks. Empirically, we have demonstrated that a logical mask is sufficient to enable near perfect performance on Boolean GOTU tasks, and that appropriately constructed masks can improve the training efficiency for arithmetic problems and image classification. Our work provides intuition on encoding inductive biases in self-attention and demonstrates the potential for logical masks to embed persistent decision-rules in transformers.

Limitations. Our approach assumes access to a meaningful interaction prior, which is natural in structured domains such as logic, arithmetic, and vision, but less clear in domains such as language. This is in effect an oracle assumption which allows us to demonstrate that useful inductive priors may be encoded within transformers using masks, but this may not be known in advance for all tasks. Designing effective masks may require knowledge of domain characteristics, and learning priors from data or pretrained models for general problems remains an important direction for future work.

8 AI USE STATEMENT

OpenAI generative AI tools were used to assist in improving the grammar, assisting with translation, and readability of the manuscript, as well as in implementing and debugging parts of the research code. All AI-assisted text was reviewed and revised by the authors, and all AI-assisted code was reviewed, tested for correctness, and verified through the resulting experiments.

Generative AI tools were not used to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in writing proofs, propose or refine research hypotheses, design or provide feedback on the research methodology or experiments, clean or reformat datasets, support qualitative or thematic data analysis, or interpret experimental results.

The authors reviewed all AI-assisted work and take full responsibility for the final content of the manuscript, including all text, claims, code, experiments, and other artifacts produced with the aid of generative AI.

REFERENCES

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, and Kevin Rizk. Generalization on the unseen, logic reasoning and degree curriculum. *Journal of Machine Learning Research*, 25(331):1–58, 2024.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1290–1299, June 2022a.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022b.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Orit Davidovich and Zohar Ringel. Algorithmic capture, computational complexity, and inductive bias of infinite transformers. *arXiv preprint arXiv:2603.11161*, 2026.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Shaoxiong Duan, Yining Shi, and Wei Xu. Attention bias as an inductive bias: How to teach transformers simple arithmetic. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=Ei4bzOt8NG.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009a.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009b. URL `https://api.semanticscholar.org/CorpusID:18268744`.

Siddharth Krishna Kumar. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*, 2017.

David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16252–16262, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Yanwei Pang, Jin Xie, Muhammad Haris Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Ling Shao. Mask-guided attention network for occluded pedestrian detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Xiaohuan Pei, Tao Huang, YanXiang Ma, and Chang Xu. Rethinking causal mask attention for vision-language inference. *arXiv preprint arXiv:2505.18605*, 2025.

Denys Pushkin, Raphaël Berthier, and Emmanuel Abbe. On the minimal degree bias in generalization on the unseen for non-boolean functions. In *Forty-first International Conference on Machine Learning*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Mohammad Samragh, Mehrdad Farajtabar, Sachin Mehta, Raviteja Vemulapalli, Fartash Faghri, Devang Naik, Oncel Tuzel, and Mohammad Rastegari. Weight subcloning: direct initialization of transformers using larger pretrained ones. *arXiv preprint arXiv:2312.09299*, 2023.

Mohammad Samragh, Iman Mirzadeh, Keivan Alizadeh Vahid, Fartash Faghri, Minsik Cho, Moin Nabi, Devang Naik, and Mehrdad Farajtabar. Scaling smart: Accelerating large language model pre-training with small model initialization. *arXiv preprint arXiv:2409.12903*, 2024.

Zachary Shinnick, Liangze Jiang, Hemanth Saratchandran, Damien Teney, and Anton van den Hengel. Can you learn to see without images? procedural warm-up for vision transformers, 2026. URL `https://arxiv.org/abs/2511.13945`.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International conference on machine learning*, pp. 9438–9447. PMLR, 2020.

Damien Teney, Liangze Jiang, Hemanth Saratchandran, and Simon Lucey. Can transformers really do it all? on the compatibility of inductive biases across tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=B08MW8oDqN.

Andrew Trask, Felix Hill, Scott E Reed, Jack Rae, Chris Dyer, and Phil Blunsom. Neural arithmetic logic units. *Advances in neural information processing systems*, 31, 2018.

Asher Trockman and J Zico Kolter. Mimetic initialization of self-attention layers. In *International Conference on Machine Learning*, pp. 34456–34468. PMLR, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *International Conference on Learning Representations*.

Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. Initializing models with larger ones. *arXiv preprint arXiv:2311.18823*, 2023.

Qingyu Yin, Xuzheng He, Xiang Zhuang, Yu Zhao, Jianhua Yao, Xiaoyu Shen, and Qiang Zhang. StableMask: Refining causal masking in decoder-only transformer. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57033–57052. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yin24a.html.

Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 288–307, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19818-2.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097, 2025.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Jianqiao Zheng, Xueqian Li, and Simon Lucey. Structured initialization for attention in vision transformers. *arXiv preprint arXiv:2404.01139*, 2024.

Jianqiao Zheng, Xueqian Li, Hemanth Saratchandran, and Simon Lucey. Structured initialization for vision transformers. *arXiv preprint arXiv:2505.19985*, 2025.

Ziqian Zhong and Jacob Andreas. Algorithmic capabilities of random transformers. *Advances in Neural Information Processing Systems*, 37:104357–104382, 2024.

A Task Definitions

A.1 Boolean GOTU Task Definitions

We consider four generalization-on-the-unseen (GOTU) Boolean tasks introduced by Abbe et al. (2024). For each task, let f denote the ground-truth target function and $\tilde{f}$ denote the minimum-degree interpolator (MDI), i.e., the lowest-degree function that agrees with f on the seen domain. These tasks are constructed such that $\tilde{f}$ matches f on all training examples but has lower degree or a simpler interaction structure. They therefore provide a controlled setting for studying whether transformers follow the MDI bias or preserve the higher-order interactions of the target function.

2Parity. $f(x) = x_0x_1$, seen domain $x_0 = 1$ or $x_1 = 1$ (i.e., excluding $(-1, -1)$). On this domain, $x_0x_1 = x_0 + x_1 - 1$, so $\tilde{f}(x) = x_0 + x_1 - 1$ (degree $2 \to 1$).

Cyclic3. $f(x) = \sum_{i=0}^{14} x_i x_{i+1} x_{i+2}$, seen domain excludes $(x_0, x_1, x_2) = (-1, -1, -1)$. The first term satisfies $x_0x_1x_2 = x_0x_1 + x_1x_2 + x_2x_0 - x_0 - x_1 - x_2 + 1$, so $\tilde{f}$ replaces this cubic term by a quadratic one (degree $3 \to 2$).

3-Bit Majority. $f(x) = \mathrm{Maj}(x_0, x_1, x_2) = \frac{x_0+x_1+x_2-x_0x_1x_2}{2}$, seen domain $x_0 = 1$ or $x_1 = 1$. On this domain, the cubic term $x_0x_1x_2$ can be rewritten into pairwise terms, yielding $\tilde{f}(x) = \frac{x_0+x_1+2x_2-x_0x_2-x_1x_2}{2}$ (degree $3 \to 2$).

ThreeSym. $f(x) = x_0x_1 - 1.25\, x_1x_2 + 1.5\, x_2x_0$, seen domain $x_0x_1x_2 = 1$. On this domain, $x_0x_1 \to x_2$, $x_1x_2 \to x_0$, $x_2x_0 \to x_1$, so $\tilde{f}(x) = x_2 - 1.25\, x_0 + 1.5\, x_1$ (degree $2 \to 1$).

A.2 Arithmetic Definitions

Addition We evaluate 3-by-3 digit decimal addition in an in-distribution setting, where all operands have exactly three digits. The transformer predicts the full output sequence in a single forward pass.

Multiplication We evaluate 3-by-3 digit decimal multiplication under the same setting. Compared to addition, multiplication is more challenging as it requires multiple intermediate digit-wise multiplications, carry operations, and the accumulation of partial sums.

B Ablation Study

B.1 Controlling for Additional Mask Parameters

To test whether the improvement comes simply from adding extra mask parameters, we compare logical mask with three controls: a randomly initialized learnable mask, an all-zero initialized learnable mask, and a fixed logical mask. The random and all-zero masks have the same number of learnable parameters as logical mask but contain no task-specific structural prior. For the all-zero mask, the log-sigmoid parameterization adds the same constant to every attention logit, so the initial attention distribution is unchanged. The fixed mask encodes the same structural prior as logical mask but is not updated during training, allowing us to distinguish the effect of the prior itself from that of its learnability.

As shown in Table 4, neither the random mask nor the all-zero mask consistently improves over the Default transformer, especially on unseen inputs. In contrast, both Fixed Mask and logical mask, which encode the correct structural prior, substantially improve extrapolation. Logical mask achieves near-perfect unseen accuracy on the three discrete tasks and also greatly reduces the test loss on ThreeSym. These results show that the gains come from the structural information encoded in the mask, rather than from simply adding more learnable parameters.

Table 4: Effect of logical mask on Boolean GOTU tasks. Results are reported as mean ± sample standard deviation over five random seeds. Test loss is the overall test MSE; accuracy (%) is reported separately on seen and unseen test inputs.

Task	Method	Loss ↓	Seen Acc ↑	Unseen Acc ↑
2Parity	Default	1.4111 ± 0.2484	100.00 ± 0.00	0.00 ± 0.00
	Random Mask	1.4196 ± 0.1913	100.00 ± 0.00	0.00 ± 0.00
	All-zero	1.3927 ± 0.2614	100.00 ± 0.00	0.00 ± 0.00
	Fixed Mask	0.0294 ± 0.0338	100.00 ± 0.00	80.00 ± 44.72
	Logical Mask	0.0138 ± 0.0172	100.00 ± 0.00	100.00 ± 0.00
Cyclic3	Default	3.9964 ± 2.4791	93.59 ± 4.19	4.14 ± 5.07
	Random Mask	3.6310 ± 1.8705	94.22 ± 2.76	0.93 ± 2.07
	All-zero	4.1167 ± 1.7160	94.96 ± 3.39	1.65 ± 2.41
	Fixed Mask	0.0100 ± 0.0010	99.24 ± 0.00	99.24 ± 0.00
	Logical Mask	0.0098 ± 0.0012	99.24 ± 0.00	99.24 ± 0.00
3-Bit Majority	Default	0.3459 ± 0.2223	100.00 ± 0.00	60.11 ± 22.30
	Random Mask	0.3258 ± 0.1914	100.00 ± 0.00	59.97 ± 22.39
	All-zero	0.3014 ± 0.2675	100.00 ± 0.00	70.25 ± 27.16
	Fixed Mask	0.0128 ± 0.0053	100.00 ± 0.00	100.00 ± 0.00
	Logical Mask	0.0148 ± 0.0012	100.00 ± 0.00	100.00 ± 0.00

B.2 Why Not Just Fix the Mask?

Another natural question is whether learnability is necessary, since fixed masks can already provide strong inductive biases on structured tasks such as Boolean and arithmetic problems. However, in more realistic settings, the available prior may only approximately reflect the underlying interaction structure. In such cases, learnability becomes important because it allows the model to refine the initial prior during training rather than being constrained to a fixed interaction pattern.

The results in Table 5 illustrate this distinction. Fixed masks improve over the default Transformer on all three vision benchmarks, showing that even a fixed structural prior can be beneficial. The gain is particularly pronounced on CIFAR100, where the fixed mask improves Top-1 accuracy from 77.26% to 79.84% (+2.58 points). However, the benefit of the fixed mask is much smaller on CIFAR10 and ImageNet100, where it improves over the default model by only +0.67 and +0.40 points, respectively. In contrast, the learnable logical mask achieves substantially larger improvements of +2.02, +4.80, and +2.36 points on CIFAR10, CIFAR100, and ImageNet100, respectively. This suggests that a fixed prior can be effective when its assumed structure is already well matched to the task, but becomes limiting when the appropriate interaction pattern differs from the predefined bias.

Overall, these results motivate making the structural prior learnable: rather than relying on the initial mask to be exactly correct, the model can preserve the useful inductive bias provided at initialization while adapting its interaction structure to the statistics of each dataset.

Table 5: Best Top-1 accuracy (%) on vision benchmarks using ViT-Tiny, reported as mean ± sample standard deviation across three random seeds. For each seed, we report the highest accuracy achieved during training. Values in parentheses denote improvement over Default in percentage points. † denotes a single-seed result (seed 0).

Dataset	Default	Mimetic	Impulse	Fixed Mask	Logical Mask
CIFAR10	93.93 ± 0.11	94.98 ± 0.26 (+1.05)	95.03 ± 0.20 (+1.10)	94.60 ± 0.42 (+0.67)	$\mathbf{95.95 \pm 0.12}$ (+2.02)
CIFAR100	77.26 ± 0.22	79.01 ± 0.06 (+1.75)	79.62 ± 0.35 (+2.36)	79.84 ± 0.23 (+2.58)	$\mathbf{82.06 \pm 0.14}$ (+4.80)
ImageNet100	83.47 ± 0.13	84.55 ± 0.14 (+1.08)	83.51 ± 0.95 (+0.04)	83.87 ± 0.25 (+0.40)	$\mathbf{85.83 \pm 0.56}$ (+2.36)

B.3 Initialization Deviation

We investigate whether the performance difference between QK-Init and logical mask can be explained by their accuracy in reproducing the desired attention pattern at initialization. QK-Init fits the query and key projections to pattern-derived logits, whereas logical mask encodes the pattern through a trainable additive attention bias.

To quantify initialization deviation, we use KL divergence from the desired attention distribution to the actual initialized attention. The desired distribution assigns equal probability to all allowed keys in each query row. We compute KL separately for each input, layer, head, and query position, then average the resulting values. Lower divergence indicates closer agreement with the desired pattern.

We evaluate both methods using four-layer, three-head transformers with sequence length 15, learnable positional embeddings, mean pooling, and no CLS token. We use seed 0 and initialization strength 20. Evaluation covers all 32,734 binary inputs that were not used to fit the QK initialization. Each layer is evaluated against its corresponding task pattern.

Table 6: Mean KL divergence between the desired attention distribution and initialized attention. Lower KL divergence indicates closer agreement with the desired distribution.

Task	QK-Init ↓	Logical Mask ↓
2Parity	7.04×10^{-4}	7.08×10^{-4}
ThreeSym	6.75×10^{-4}	6.84×10^{-4}
3-Bit Majority	7.05×10^{-4}	7.15×10^{-4}
Cyclic3	1.07×10^{-2}	2.38×10^{-2}

Both methods closely reproduce the desired distributions for 2Parity, ThreeSym, and 3-Bit Majority, with similar initialization deviations. For Cyclic3, QK-Init achieves lower divergence than logical mask. These results suggest that any downstream advantage of logical mask is not explained by more accurate initial pattern matching in this setting.

B.4 Effect of Initialization Strength

We study the sensitivity of logical mask to the magnitude of the initialized prior. Specifically, for an initialization magnitude c, the open positions of the mask are initialized with $+c$, while the closed positions are initialized with $-c$. A larger magnitude therefore corresponds to a stronger structural prior at initialization. We evaluate different initialization strengths on both Boolean GOTU tasks and ImageNet-100.

Boolean tasks. We first evaluate initialization values of ±1, ±5, ±10, and ±20 on the Boolean tasks. The results are reported in Table 7.

Table 7: Ablation on logical mask strength for Boolean GOTU tasks. Results are reported as mean test accuracy (%) $\pm$ standard deviation. Higher is better.

Task	Split	±1	±5	±10	±20
2Parity	Seen	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	Unseen	0.00 ± 0.00	0.00 ± 0.00	58.97 ± 53.87	100.00 ± 0.00
Cyclic3	Seen	95.60 ± 2.38	98.76 ± 1.03	98.21 ± 0.83	99.24 ± 0.00
	Unseen	20.40 ± 35.12	92.44 ± 13.96	96.80 ± 1.69	99.24 ± 0.00
3-Bit Majority	Seen	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	Unseen	69.95 ± 27.43	89.93 ± 22.51	100.00 ± 0.00	100.00 ± 0.00

These results suggest that the structural prior must be sufficiently strong at initialization to reliably influence the learned extrapolation behavior. The required strength is task dependent: some tasks, such as Cyclic3, benefit substantially from moderate initialization strengths, whereas 2Parity requires a stronger prior to consistently recover the target rule.

ImageNet-100. We further evaluate initialization strengths of ±1, ±2, ±5, ±10, and ±20 on ImageNet-100. The results are reported in Table 8.

A similar trend is observed on ImageNet-100. Increasing the initialization magnitude from ±1 to ±10 improves the best Top-1 accuracy from 84.10% to 85.83%. Further increasing the magnitude to ±20 does not provide an additional improvement, with the accuracy slightly decreasing to 85.41%.

Table 8: Ablation on logical mask strength for ImageNet-100. Results are reported as mean best Top-1 accuracy (%) ± standard deviation. Higher is better.

Initialization Values	Best Top-1 Accuracy
± 1	84.10 ± 0.45
± 2	84.23 ± 0.15
± 5	85.01 ± 0.35
± 10	85.83 ± 0.56
± 20	85.41 ± 0.27

Overall, these experiments show that performance is sensitive to very weak initialization values, but becomes relatively stable once the structural prior is sufficiently strong.

C Proof of Theorem 1 and Corollary 1

Theorem 1. *Consider a single attention row with query $q \in \mathbb{R}^d$, keys $\{k_j\}_{j=1}^N$, and mask parameters $\{m_j\}_{j=1}^N$. Define the softmax function working on the attention score as*

$$p_j = \frac{\exp(q^\top k_j/\sqrt{d} + \log \sigma(m_j))}{\sum_{\ell=1}^N \exp(q^\top k_\ell/\sqrt{d} + \log \sigma(m_\ell))}.$$

Then, for any $i, j \in \{1, \dots, N\}$,

$$\frac{\partial p_i}{\partial q} = \frac{p_i}{\sqrt{d}} \left(k_i - \sum_{\ell=1}^N p_\ell k_\ell \right), \tag{8}$$

$$\frac{\partial p_i}{\partial k_j} = \frac{1}{\sqrt{d}} p_i(\delta_{ij} - p_j) q, \tag{9}$$

$$\frac{\partial p_i}{\partial m_j} = p_i(\delta_{ij} - p_j)\sigma(-m_j). \tag{10}$$

Proof. We consider a single attention row with

$$p_j = \frac{\exp\Big(s_j/\sqrt{d} + \log \sigma(m_j)\Big)}{\sum_{\ell=1}^N \exp\Big(s_\ell/\sqrt{d} + \log \sigma(m_\ell)\Big)}.$$

Let

$$z_j = s_j/\sqrt{d} + \log \sigma(m_j).$$

Then $p = \mathrm{softmax}(z)$, and the standard softmax derivative gives

$$\frac{\partial p_i}{\partial z_j} = p_i(\delta_{ij} - p_j).$$

By the chain rule, we have

$$\frac{\partial z_j}{\partial q} = \frac{1}{\sqrt{d}} k_j, \qquad \frac{\partial z_j}{\partial k_j} = \frac{1}{\sqrt{d}} q, \qquad \frac{\partial z_j}{\partial m_j} = \sigma(-m_j).$$

Combining these yields

$$\frac{\partial p_i}{\partial q} = \sum_{j=1}^N \frac{\partial p_i}{\partial z_j} \frac{\partial z_j}{\partial q} = \frac{p_i}{\sqrt{d}} \left(k_i - \sum_{\ell=1}^N p_\ell k_\ell \right),$$

$$\frac{\partial p_i}{\partial k_j} = \frac{1}{\sqrt{d}} p_i(\delta_{ij} - p_j) q,$$

and

$$\frac{\partial p_i}{\partial m_j} = p_i(\delta_{ij} - p_j)\sigma(-m_j),$$

which completes the proof. □

Corollary 1. *Under the setting of Theorem* 1*, suppose* $\|q\|_2 \le B_q$ *and* $\|k_j\|_2 \le B_k$ *for all* j*. Then*

$$\left\| \frac{\partial p_i}{\partial q} \right\|_2 \le \frac{2p_i B_k}{\sqrt{d}}, \tag{11}$$

$$\left\| \frac{\partial p_i}{\partial k_j} \right\|_2 \le \frac{p_i|\delta_{ij} - p_j|B_q}{\sqrt{d}} \le \frac{p_i B_q}{\sqrt{d}}, \tag{12}$$

$$\left| \frac{\partial p_i}{\partial m_j} \right| \le p_i|\delta_{ij} - p_j|\sigma(-m_j) \le p_i. \tag{13}$$

Proof. From Theorem 1, we have

$$\frac{\partial p_i}{\partial q} = \frac{p_i}{\sqrt{d}} \left(k_i - \sum_{\ell=1}^{N} p_\ell k_\ell \right).$$

Assuming $\|k_j\|_2 \le B_k$ for all j, we bound by using triangular inequality and $\forall \ell \in [N], p_\ell \ge 0, \sum_{\ell=1}^{N} p_\ell = 1$:

$$\left\| \sum_{\ell=1}^{N} p_\ell k_\ell \right\|_2 \le \sum_{\ell=1}^{N} p_\ell \|k_\ell\|_2 \le \sum_{\ell=1}^{N} p_\ell B_k = B_k,$$

which implies (By triangular inequality)

$$\left\| k_i - \sum_{\ell=1}^{N} p_\ell k_\ell \right\|_2 \le \|k_i\|_2 + \left\| \sum_{\ell=1}^{N} p_\ell k_\ell \right\|_2 \le 2B_k.$$

Therefore,

$$\left\| \frac{\partial p_i}{\partial q} \right\|_2 = \left\| \frac{p_i}{\sqrt{d}} \left(k_i - \sum_{\ell=1}^{N} p_\ell k_\ell \right) \right\|_2 = \frac{|p_i|}{\sqrt{d}} \left\| \left(k_i - \sum_{\ell=1}^{N} p_\ell k_\ell \right) \right\|_2 \le \frac{2p_i B_k}{\sqrt{d}}.$$

Similarly, using $\|q\|_2 \le B_q$, we obtain

$$\left\| \frac{\partial p_i}{\partial k_j} \right\|_2 = \frac{1}{\sqrt{d}} p_i|\delta_{ij} - p_j|\|q\|_2 \le \frac{p_i B_q}{\sqrt{d}}.$$

Finally, since $\sigma(-m_j) \le 1$ and $|\delta_{ij} - p_j| \le 1$, we have

$$\left| \frac{\partial p_i}{\partial m_j} \right| \le p_i,$$

which completes the proof. □

D Variable-Length Inputs and Mask Parameterizations

A dense attention mask is tied to a fixed input length because it stores a separate parameter for each pair of tokens. To support variable-length inputs and reduce the number of mask parameters, we represent the mask using an implicit neural representation (INR).

INR mask parameterization. An INR uses a neural network to map coordinates to values. Instead of learning each mask entry M_{ij} independently, we compute it as

$$M_{ij} = f_\theta(G_{ij}),$$

where G_{ij} contains geometric information about tokens i and j, and f_θ is an MLP. Since the same MLP can be evaluated at any pair of coordinates, it can generate masks for different input lengths without storing a separate mask for each length. INRs are commonly used to represent continuous signals, such as images, at different resolutions (Tancik et al., 2020; Sitzmann et al., 2020).

Variable-length arithmetic tasks. For the arithmetic experiments, G_{ij} contains the coordinates of the two tokens and their relative distance. We use a ReLU MLP with a hidden dimension of 64. We first train the MLP to approximate the target mask and then jointly optimize the MLP and the Transformer.

We train on inputs with different digit lengths on multiplication task in the same run. Table 9 shows the results. The INR-based logical mask model achieves high digit-level and sequence-level accuracy across all tested input lengths, while the standard Transformer performs substantially worse.

Table 9: Performance on variable-length arithmetic tasks. d denotes the number of digits in each operand. Results are reported as digit-level and sequence-level accuracy.

Length	Method	Digit Acc. (%)	Sequence Acc. (%)
$d = 2$	MaskInit (INR)	**97.53**	**93.26**
$d = 2$	Transformer	47.02	11.52
$d = 3$	MaskInit (INR)	**99.99**	**99.96**
$d = 3$	Transformer	57.08	3.43
$d = 4$	MaskInit (INR)	**99.92**	**99.60**
$d = 4$	Transformer	43.95	0.12

Comparison with relative position bias. We further compare the INR mask with relative position bias (RPB), as used in Swin Transformer (Liu et al., 2021). RPB uses a learnable lookup table indexed by the relative spatial offset between two tokens. In contrast, our INR predicts the mask value from geometric features.

For each query-key pair, the INR receives the normalized token coordinates, relative spatial offsets, relative distance, and binary indicators for interactions involving the CLS token. A lightweight MLP hen produces one mask value for each attention head. As before, we first fit the INR to the target logical mask and then jointly train it with the Transformer.

On ImageNet-100, MaskInit (INR) achieves a top-1 accuracy of $87.32 \pm 0.05\%$, outperforming both RPB variants with a similar number of parameters. Results are summarized in Table 10.

Table 10: Comparison of mask parameterizations on ImageNet-100. Accuracy is reported as the mean and sample standard deviation of the best validation accuracy across three random seeds.

Method	Top-1 Accuracy (%) ↑	Parameters (M)
RPB (Default)	86.23 ± 0.13	5.57
RPB (Impulse Init)	86.53 ± 0.06	5.57
INR MaskInit (Impulse Init)	$\mathbf{87.32 \pm 0.05}$	5.60

E Empirical Verification of Persistency

We empirically examine the persistency of the initialized structural patterns during training on four Boolean tasks. We compare logical mask with Structured (SVD) and Structured (Opt) over five random seeds (0-4), measuring the KL divergence from the initial attention pattern to the current pattern at epochs 10, 30, 50, 70, and the final epoch (index 89). Smaller KL divergence indicates greater preservation of the initialized pattern. We additionally track the gradient norm of the mask parameters for logical mask and the query-key parameters for the structured initialization baselines.

Persistence of initialized patterns. As shown in Table 11, Logical mask remains close to its initialization across all sampled epochs. Its mean KL divergence increases gradually, reaching only 0.052, 0.032, 0.208, and 0.011 at the final epoch for 2Parity, Cyclic3, 3-Bit Majority, and ThreeSym, respectively. In contrast, both structured initialization baselines already exhibit substantial divergence by epoch 10 and retain considerably larger divergence throughout the sampled trajectory. Although

their divergence sometimes decreases later in training, their final values remain between 2.24 and 2.87 across the four tasks. These observations support greater empirical persistency of the patterns encoded by logical mask.

Table 11: KL divergence from the initial to the current attention pattern at selected training epochs. Values are means over five random seeds. Final denotes epoch index 89.

Task	Method	10	30	50	70	Final
2Parity	Structured (SVD)	2.88	2.74	2.74	2.77	2.76
	Structured (Opt)	2.83	2.67	2.69	2.72	2.69
	Logical Mask	**0.00**	**0.01**	**0.02**	**0.04**	**0.05**
Cyclic3	Structured (SVD)	1.36	2.16	2.39	2.29	2.24
	Structured (Opt)	4.46	3.70	3.15	2.81	2.66
	Logical Mask	**0.00**	**0.00**	**0.01**	**0.02**	**0.03**
3-Bit Majority	Structured (SVD)	2.99	2.75	2.91	2.91	2.87
	Structured (Opt)	2.92	2.69	2.68	2.69	2.69
	Logical Mask	**0.00**	**0.02**	**0.07**	**0.16**	**0.21**
ThreeSym	Structured (SVD)	2.72	2.77	2.73	2.73	2.73
	Structured (Opt)	2.83	2.73	2.68	2.67	2.67
	Logical Mask	**0.00**	**0.00**	**0.01**	**0.01**	**0.01**

Gradient dynamics. Table 12 reports the corresponding gradient norms. At every sampled epoch, the mask-parameter gradient norm is several orders of magnitude smaller than the query-key gradient norms of the structured initialization baselines. This separation coincides with the smaller attention-pattern drift observed for logical mask. The mask gradients remain nonzero, indicating that the initialized structure remains trainable rather than being frozen. However, these norms concern different parameterizations and parameter counts, so their magnitudes are descriptive diagnostics, not directly comparable measures of functional change. This also provides empirical evidence for our propositions.

F Design of Mask Pattern

In this section, we outline the key principles for designing effective mask-based initialization. Before presenting the design principles, we first note that low-order interactions, such as pairwise terms $x_i x_j$, can be easily represented by shallow neural networks. In particular, it is well known that a two-layer MLP can express functions such as XOR, indicating that modeling such interactions does not require deep architectural complexity.

This observation suggests that the role of attention can choose not to represent these polynomial interactions directly, but rather to select which variables should interact. However, prior work (Abbe et al., 2024) failure indicates that transformers may rely on spurious correlations that fit the training data, rather than recovering the underlying structure.

Therefore, by explicitly controlling token-wise interactions through the attention mask, we can guide the model to construct the desired extrapolation property in principle.

Design Principles for Mask Pattern. Mask-based initialization is guided by the following principles.

(1) Task-aligned interactions. The mask should allow only the interactions needed for the task. For example, in 2-parity, only the first two input components should interact. Other components may affect the prediction and hurt extrapolation.

(2) Suppression of shortcut solutions. Stacking multiple layers allows the model to build higher-order interactions from simpler ones. We can guide this process by blocking unnecessary interactions. For 3-Bit Majority, a second-order interaction is an MDI solution. Blocking off-diagonal interactions therefore encourages the model to learn the third order solution and extrapolate correctly.

Table 12: Gradient norms at selected training epochs, averaged over training batches within each epoch and then over five random seeds. Logical mask reports mask-parameter gradient norms; Structured (SVD) and Structured (Opt) report query-key gradient norms.

Task	Method	10	30	50	70	Final
2Parity	Structured (SVD)	0.09	0.05	0.01	0.01	0.01
	Structured (Opt)	0.09	0.06	0.06	0.06	0.05
	Logical Mask	8.17×10^{-6}	5.48×10^{-6}	5.40×10^{-6}	8.51×10^{-6}	1.18×10^{-5}
Cyclic3	Structured (SVD)	3.19	17.28	10.25	9.30	6.61
	Structured (Opt)	6.96	8.01	8.80	7.34	6.65
	Logical Mask	2.01×10^{-3}	1.78×10^{-3}	3.12×10^{-3}	5.71×10^{-3}	8.17×10^{-3}
3-Bit Majority	Structured (SVD)	0.22	0.03	0.04	0.04	0.03
	Structured (Opt)	0.25	0.02	0.03	0.03	0.02
	Logical Mask	8.18×10^{-6}	9.91×10^{-5}	6.11×10^{-5}	8.44×10^{-5}	1.20×10^{-4}
ThreeSym	Structured (SVD)	0.24	0.13	0.13	0.08	0.09
	Structured (Opt)	0.34	0.35	0.22	0.33	0.35
	Logical Mask	1.20×10^{-5}	3.39×10^{-5}	5.27×10^{-5}	5.45×10^{-5}	5.29×10^{-5}

(3) Multi-head specialization. A layer contains multiple attention heads, and each head can capture a different interaction. For example, ThreeSym is composed of multiple 2-parity functions. We therefore assign different masks to different heads to support correct extrapolation.

These principles define a structured way to control the interaction graph of the model, guiding it toward the desired computation.

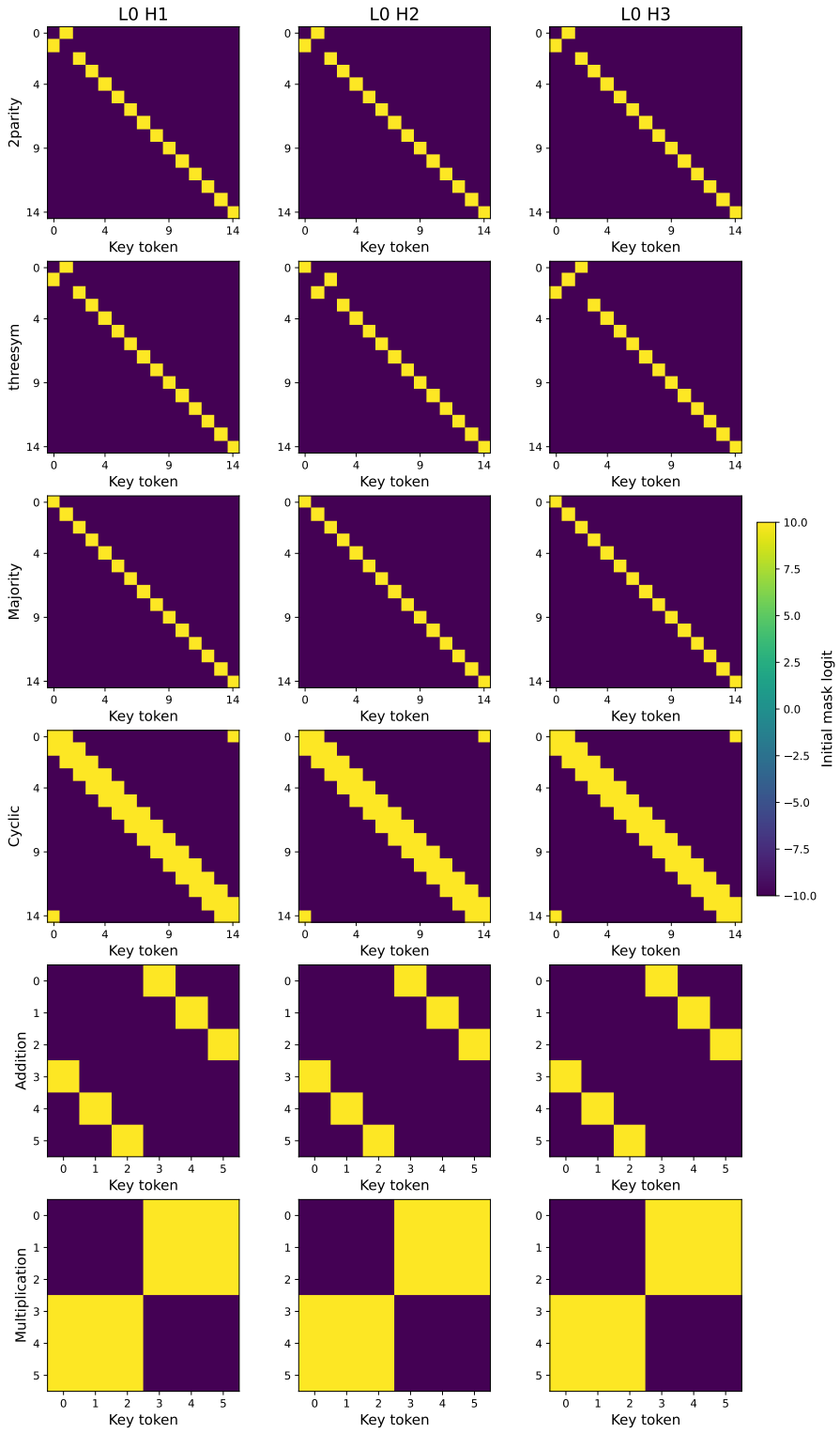


Figure 8: Initial mask logits for different tasks. Each row corresponds to a task and each column shows different attention heads at the first layer.

G EXPERIMENT DETAILS

In this section, we provide a detailed description of the experimental setup. Boolean experiments are conducted on a single NVIDIA 4090 GPU. Other experiments are conducted on a single NVIDIA A100 GPU (40GB).

G.1 BOOLEAN PROBLEMS

For Boolean tasks, we build our implementation upon the GOTU repository[1], with additional modifications such as incorporating a learning rate scheduler to improve convergence speed. The Structured (Opt) initialization is based on the official implementation[2], which optimizes the QK matrices directly via gradient descent, instead of using SVD-based decomposition. The detailed information is shown at Table 13. For logical mask, we choose open positions = 20 and close positions = -20.

Table 13: Training setup for Boolean tasks.

Task	Epochs	Layers	Heads	LR	Batch	Optimizer	Seed
2Parity	90	4	3	1×10^{-3} –1×10^{-4}	256	Adam	0–4
ThreeSym	90	4	3	1×10^{-3} –1×10^{-4}	256	Adam	0–4
3-Bit Majority	90	4	3	1×10^{-3} –1×10^{-4}	256	Adam	0–4
Cyclic3	90	4	3	1×10^{-3} –1×10^{-5}	256	Adam	0–4

G.2 ARITHMETIC PROBLEMS

We implement all arithmetic experiments from scratch due to the lack of existing code. All methods share the same architecture and training setup, differing only in the initialization strategy Table 14.

Table 14: Arithmetic task setup.

Item	Value
Task	Addition / Multiplication (3-digit)
Dataset size	10^6 (all combinations)
Model	Transformer
Layers / Heads	6 / 4
Model dim	256
MLP dim	256
Head dim	64
Positional encoding	learnable
Loss	Cross-entropy
Optimizer	AdamW
Learning rate	1×10^{-3}
Weight decay	1×10^{-3}
Batch size	256
Epochs	50
Scheduler	cosine
Logical Mask	open = 5, close = -5
Structured init (SVD)	open = 10000, close = 0
Structured init (Opt)	steps = 2000, lr = 1×10^{-4}

[1] https://github.com/aryol/GOTU/tree/main

[2] https://github.com/osiriszjq/structured_initialization

G.3 IMAGE CLASSIFICATION

We summarize the experimental setup for image classification tasks in Table 15 and section G.3 and the corresponding dataset-specific tables. All experiments are conducted using a ViT-Tiny architecture with identical training pipelines across different initialization methods to ensure fair comparison.

For CIFAR10 and CIFAR100, we follow a standard training protocol with a batch size of 512, cosine learning rate schedule, and extensive data augmentation including Mixup, CutMix, and RandAugment. For ImageNet100, we adopt a similar setup with linear learning rate scaling and a shorter warmup period.

To evaluate the effect of initialization, we compare default initialization with prior structured initialization methods (mimetic and impulse) and our proposed logical mask. For logical mask, we use a simple prior with open entries set to 10 and closed entries set to -10, which encodes a strong structural bias at initialization.

We evaluate our method on CIFAR10 (Krizhevsky, 2009b), CIFAR100 (Krizhevsky, 2009b), and ImageNet100 is constructed as a 100-class subset of ImageNet (Deng et al., 2009), following the version available at `https://huggingface.co/datasets/clane9/imagenet-100`. Our implementation is based on the official repository[3].

Table 15: Experimental setup for CIFAR10.

Item	Value
Model	vit_tiny_patch16_224
Dataset	CIFAR10
Classes	10
Input size	$3 \times 224 \times 224$
Seed	0–2
GPU	1×A100
Batch size	512
Epochs	400
Optimizer	AdamW
LR	2×10^{-3}
Scheduler	cosine
Warmup epochs	50
Weight decay	0.05
AMP	yes
Drop path	0.1
Label smoothing	0.1
Mixup / CutMix	0.8 / 1.0
Augment	RandAug (m9)

[3] `https://github.com/osiriszjq/structured_initialization`

Table 16: Experimental setup for CIFAR100.

Item	Value
Model	vit_tiny_patch16_224
Dataset	CIFAR100
Classes	100
Input size	$3 \times 224 \times 224$
Seed	0–2
GPU	1×A100
Batch size	512
Epochs	400
Optimizer	AdamW
LR	2×10^{-3}
Scheduler	cosine
Warmup epochs	50
Weight decay	0.05
AMP	yes
Drop path	0.1
Label smoothing	0.1
Mixup / CutMix	0.8 / 1.0
Augment	RandAug (m9)

Table 17: Experimental setup for ImageNet100.

Item	Value
Model	vit_tiny_patch16_224
Dataset	ImageNet100
Classes	100
Input size	$3 \times 224 \times 224$
Seed	0–2
GPU	1×A100
Batch size	512
Epochs	300
Optimizer	AdamW
Base LR	1×10^{-3}
LR scaling	linear (base size 512)
Scheduler	cosine
Warmup epochs	5
Cooldown epochs	10
Weight decay	0.05
AMP	yes
Global pool	avg
Drop path	0.1
Label smoothing	0.1
Mixup / CutMix	0.8 / 1.0
Augment	RandAug (m9)

H VISUALIZATIONS

H.1 VISUALIZATION OF STRUCTURE INITIALIZATION

We visualize structured initialization in Figure 9, where convolutional impulse filters are translated into corresponding attention masks.

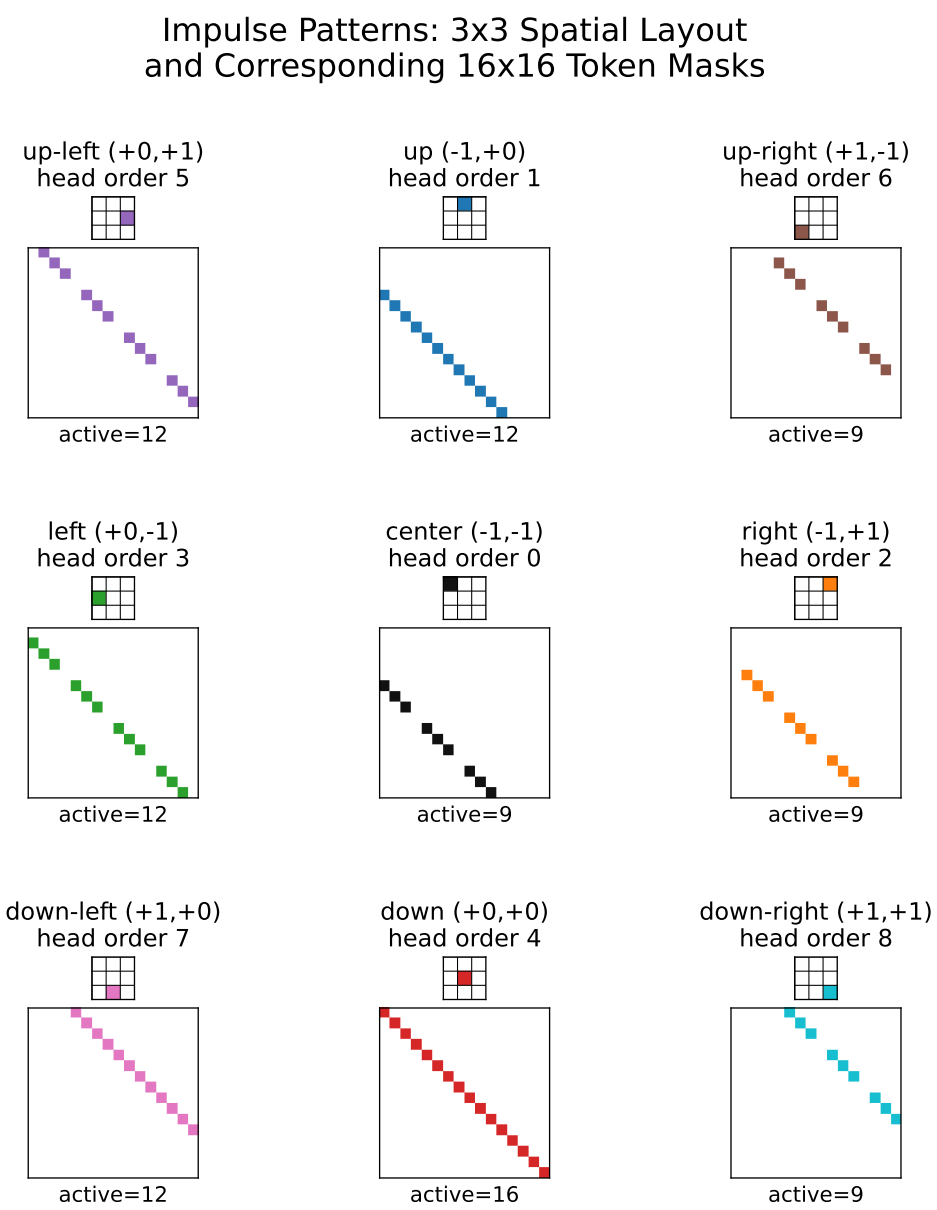


Figure 9: Structured initialization for vision transformers. convolutional impulse filters (above) are translated into corresponding token-level attention masks (below), inducing locality and spatial inductive bias.

H.2 Mask For Image Classification

We visualize the learned masks for Head 1 at different layers after training for different tasks, as shown in Figure 10. The plots illustrate the effective bias applied to the attention logits. We observe that when the initial inductive bias is not well aligned with the task, the learnable mask adapts and modifies the pattern accordingly.

H.3 Attention Map For Image Classification

We visualize the attention maps for different methods in Figures 11 to 13. All attention values are taken after the softmax operation for the first test sample. For mask-based initialization, the mask is applied before softmax, and the resulting attention probabilities are shown. We observe that the core

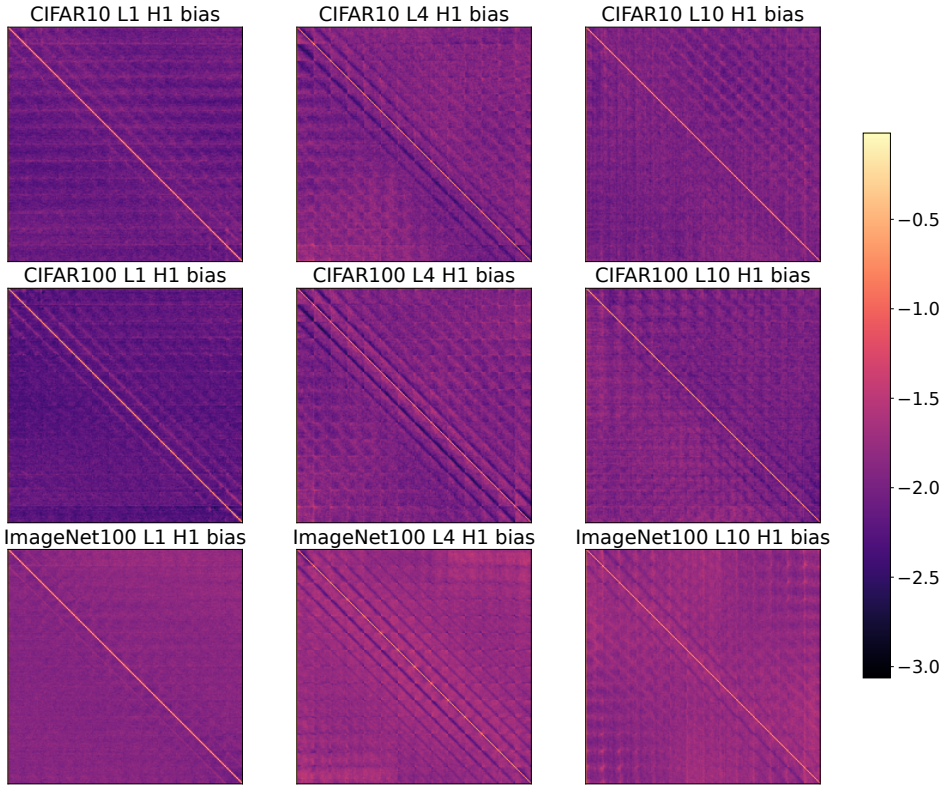


Figure 10: Learned mask (after training) for CIFAR10, CIFAR100, and ImageNet100 across different layers. The visualization shows the effective mask bias applied to the attention logits.

pattern is preserved across all tasks, while additional positions are gradually opened depending on the alignment between the task, layer, and the prior.

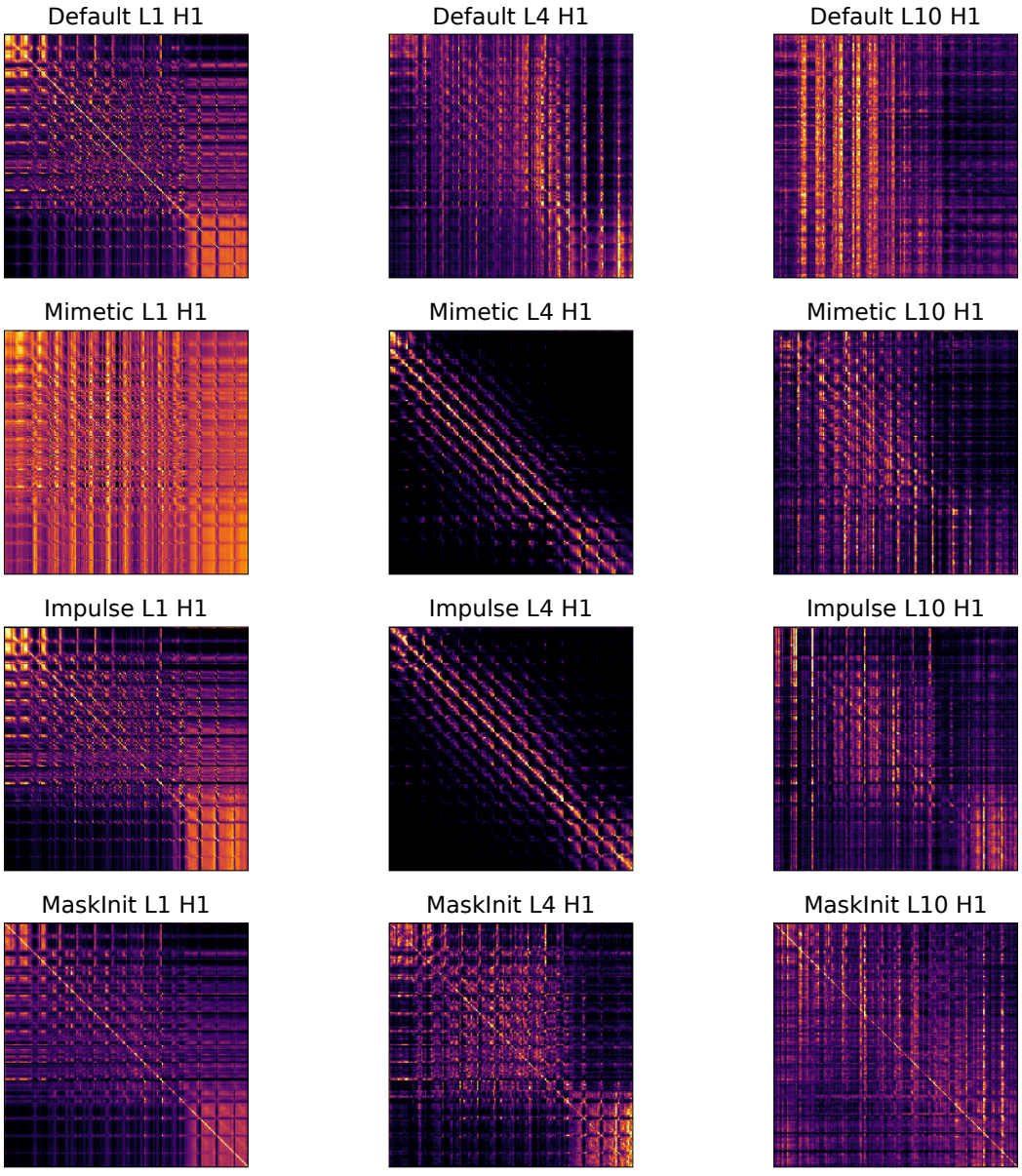


Figure 11: Attention maps for the first validation sample on CIFAR10.

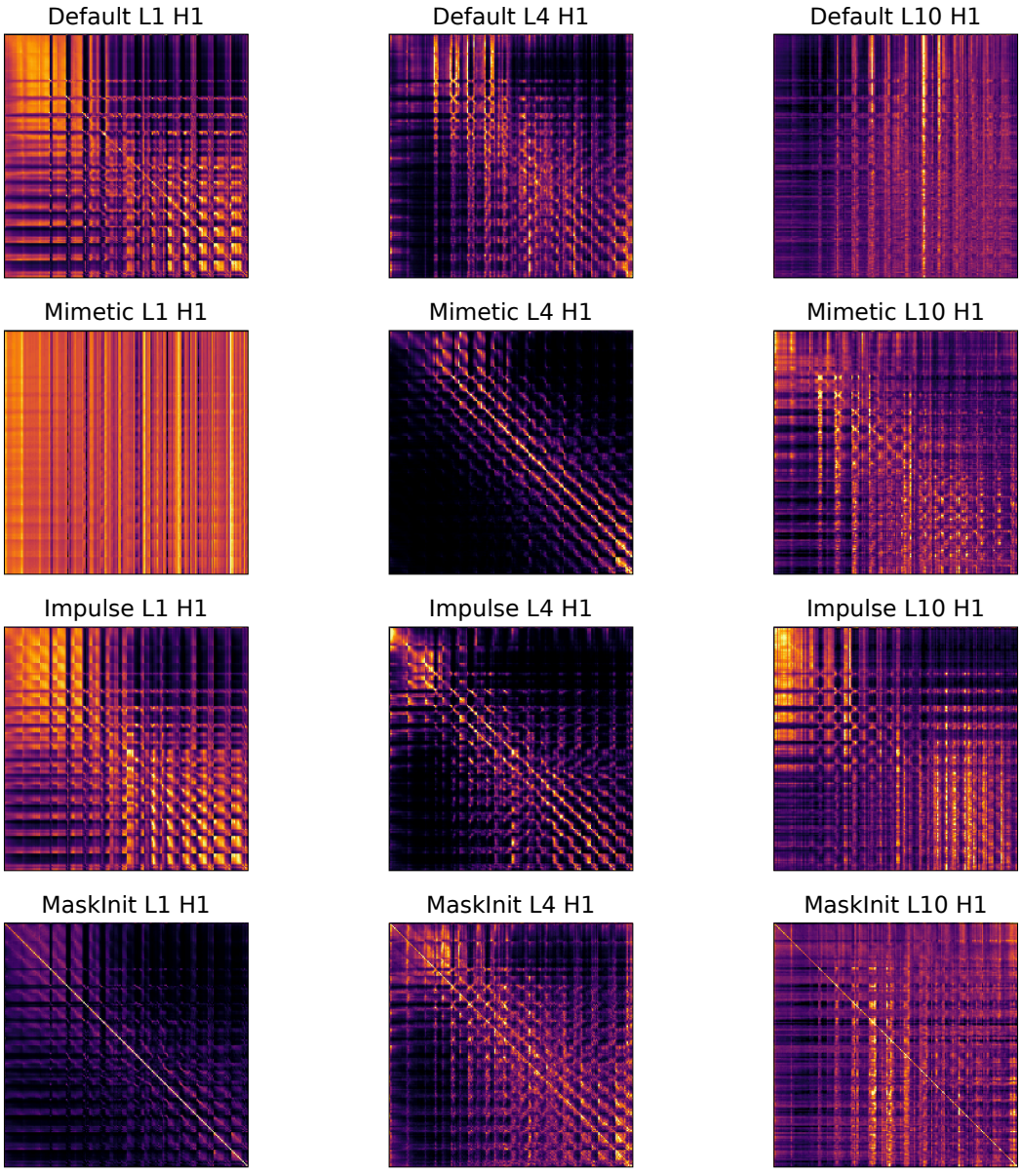


Figure 12: Attention maps for the first validation sample on CIFAR100.

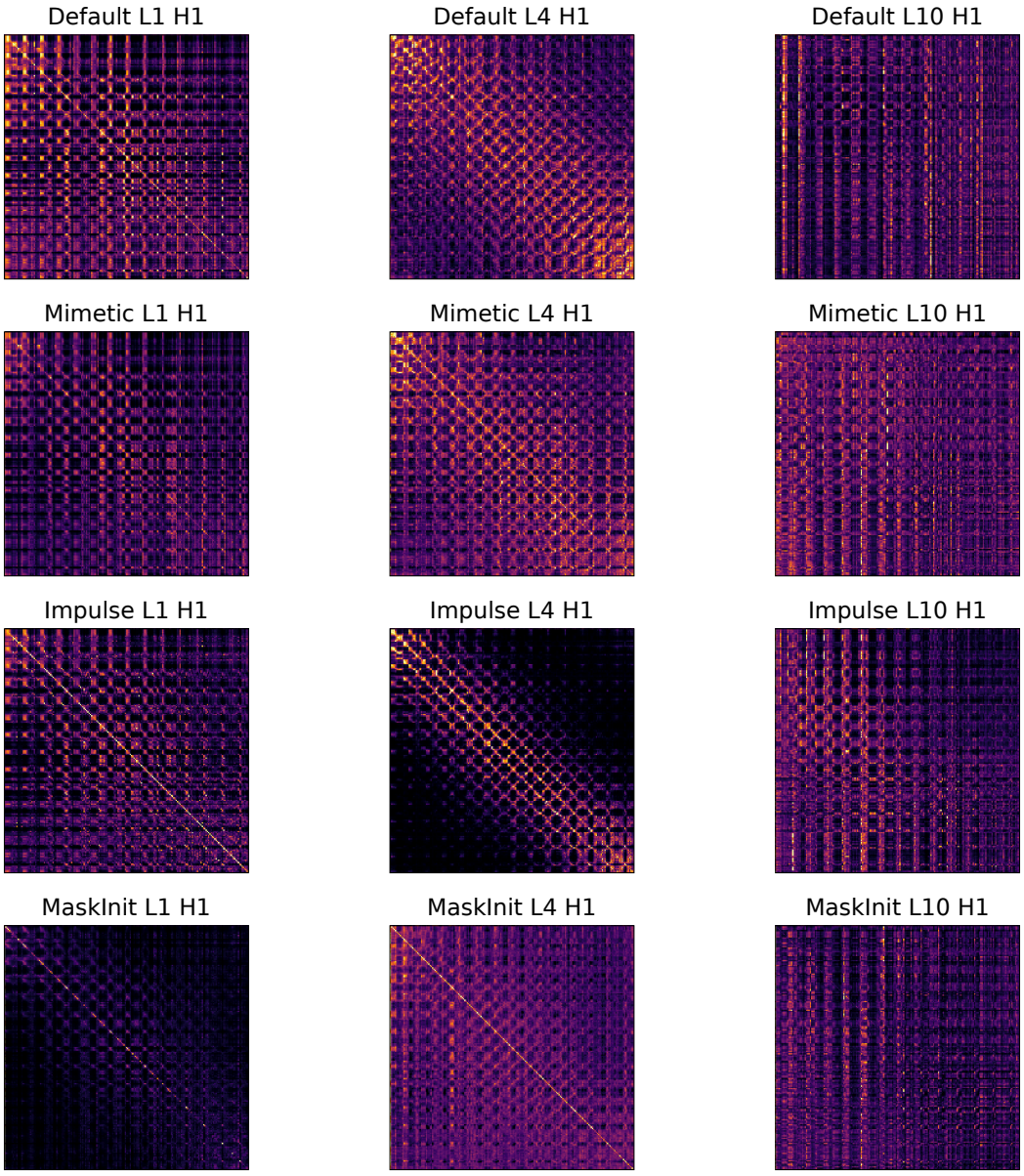


Figure 13: Attention maps for the first validation sample on ImageNet100.

H.4 Visualizations for Convergence Curves

In this section, we visualize the convergence behavior of different initialization methods on the CIFAR100 and ImageNet100 datasets as shown at Figure 14. Our method begins to outperform the baselines after around 100 epochs and maintains a faster convergence rate throughout training. This suggests a two-phase training dynamic: in the early stage, the model aligns latent features under the constraint of the mask, while in later stages, it jointly refines both the mask and the latent representations.

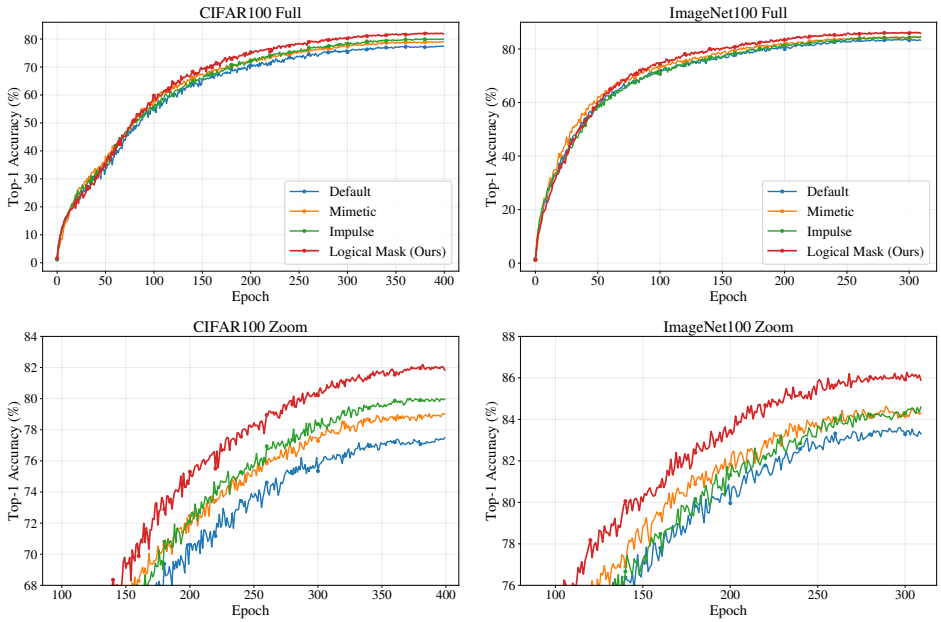


Figure 14: Training curves on CIFAR100 (left) and ImageNet100 (right) under identical training settings. The plots show validation Top-1 accuracy over training epochs. We compare default initialization, prior structured initialization methods, and the proposed logical mask. Our logical mask consistently converges faster in the early stages and achieves higher final accuracy. This indicates that mask-based initialization provides a better inductive bias, leading to improved optimization efficiency and generalization on vision benchmarks.