# Understanding Task Vectors in In-Context Learning: Emergence, Functionality, and Limitations

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Task vectors offer a compelling mechanism for accelerating inference in in-context learning (ICL) by distilling task-specific information into a single, reusable representation. Despite their empirical success, the underlying principles governing their emergence and functionality remain unclear. This work proposes the *Linear Combination Conjecture*, positing that task vectors act as single in-context demonstrations formed through linear combinations of the original ones. We provide both theoretical and empirical support for this conjecture. First, we show that task vectors naturally emerge in linear transformers trained on triplet-formatted prompts through loss landscape analysis. Next, we predict the failure of task vectors on representing high-rank mappings and confirm this on practical LLMs. Our findings are further validated through saliency analyses and parameter visualization, suggesting an enhancement of task vectors by injecting multiple ones into few-shot prompts. Together, our results advance the understanding of task vectors and shed light on the mechanisms underlying ICL in transformer-based models.

## 1 Introduction

In-context learning (ICL) is a core capability of large language models (LLMs), allowing them to perform new tasks without parameter updates by conditioning on a few input-output examples in the prompt [2]. Unlike traditional training, ICL relies on attention-based mechanisms to infer task structure directly from context. This surprising generalization ability has led to growing interest in uncovering the principles of learning purely from contextual examples [21, 3, 4, 15, 5].

A recent work investigates the task vector method [7] (concurrent works include function vectors [16] and in-context vectors [13]), a technique that distills underlying task information from ICL demonstrations into a single vector. Typically, ICL prompts are structured as sequences of triplets, each encoding a semantic mapping, in addition to a query at the end (e.g., *"hot* $\rightarrow$ *cold, up* $\rightarrow$ *down, day* $\rightarrow$ *night, dark* $\rightarrow$*"*). Task vectors are then extracted from the hidden states of the last ($\rightarrow$) token. Once obtained, these vectors can be injected into the same position in new prompts (e.g., *"big* $\rightarrow$*"*), enabling the model to generalize to unseen inputs in a zero-shot fashion.

Task vectors have been shown to naturally emerge even in small transformer models trained from scratch on synthetic data [24], suggesting that their formation is a general property of attention-based architectures. Recent studies further demonstrate that task vectors can be enhanced by aggregating hidden states across multiple layers and multiple arrow tokens [12]. Beyond language models, task vectors are also found effective in large-scale visual [8] and multi-modal [9] models.

Despite their empirical effectiveness, the underlying mechanism of task vectors, especially how they emerge, function, and encode task information, remains poorly understood. This paper takes a step toward unveiling the principles behind it by introducing the following conjecture:

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

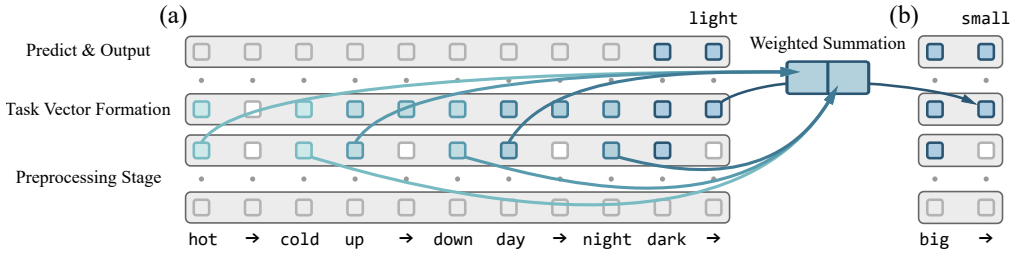



Figure 1: Overview of task vector and our main conjecture. (a) Task vector emerges during ICL as a linear combination of preceding in-context demonstrations. (b) It can then be injected into zero-shot prompts and functions as a single, representative demonstration, facilitating efficient prediction.

**Linear Combination Conjecture**

*The injected task vector functions as a single in-context demonstration, formed through a linear combination of the original demonstrations (hidden states).*

Figure 1 provides an intuitive illustration for our conjecture. In the following sections, we validate this conjecture through various empirical and theoretical perspectives. These analyses comprehensively explain how task vectors naturally emerge within attention-based model architectures, effectively encode task-related information, and facilitate inference in zero-shot prompts. Our work advances the understanding of the underlying mechanisms behind ICL, clarifying both the efficacy and limitations of task vectors in transformer-based LLMs. The highlights of this paper are as follows:

- **Theoretical Justification in Linear Transformers:** We theoretically characterize the critical points of linear-attention transformers and demonstrate how they solve random linear regression tasks through embedding concatenation and gradient descent. With a triplet-formatted input prompt structure, task vectors naturally emerge at arrow tokens as linear combinations of the in-context demonstrations. These vectors serve as redundancy against information loss induced by dropout, thereby improving robustness. Empirically, the learned linear model parameters closely align with the predicted structure and successfully replicate the task vector mechanism.
- **Empirical Verification in Practical LLMs:** We visualize the information flow in LLMs with saliency analysis and observe patterns consistent with linear models, suggesting they share similar underlying mechanisms. According to our conjecture, inference with task vectors is analogous to 1-shot ICL, which is inherently limited to rank-one meta-predictors under the gradient descent perspective. To validate this, we introduce a series of bijection tasks that are provably unsolvable by rank-one predictors, and empirically confirm this failure in real-world transformers. Building on these insights, we enhance the standard task vector method by injecting multiple vectors into few-shot prompts, resulting in consistent performance gains across a range of ICL tasks.

## 2 Setting: Random Linear Regression with Linear-Attention Transformers

**Notations:** We write $[n] = \{1, \cdots, n\}$. The Hadamard product is denoted by $\circ$, and the Kronecker product by $\otimes$. The identity matrix of dimension $n$ is denoted by $I_n$, while $0_n$ and $0_{m\times n}$ represent zero vectors or matrices of the corresponding dimensions. Subscripts are omitted when the dimensions are clear from context. We define $\mathcal{M}(M) = \left\{\Lambda \in \mathbb{R}^{\dim(M)} \;\middle|\; \Lambda = M \circ A,\ A \in \mathbb{R}^{\dim(M)}\right\}$ as the set of masked matrices induced by the binary mask $M$. For a general matrix $A$, the element at the $i$-th row and $j$-th column is denoted by $A_{i,j}$, and the sub-block from rows $i$ to $k$ and columns $j$ to $l$ is denoted by $A_{i:k,j:l}$. $\mathrm{diag}(A_1, \cdots, A_n)$ represents the block-diagonal matrix constructed by $\{A_i\}_{i=1}^n$.

**Random Linear Regression:** Following the settings in literature [6, 17, 1, 20], we consider training linear transformers on random instances of linear regression. Let $\{x_i\}_{i=1}^{n+1}$, where $x_i \in \mathbb{R}^d$, denote covariates drawn i.i.d. from distribution $P_x$, and let $\{w_i\}_{i=1}^d$, where $w_i \in \mathbb{R}^d$, denote coefficients drawn i.i.d. from distribution $P_w$. Define the coefficient matrix as $W = [w_1 \quad \cdots \quad w_d]^\top \in \mathbb{R}^{d\times d}$. The responses are then generated as $y_i = Wx_i$ for $i \in [n+1]$. We denote by $X, Y \in \mathbb{R}^{d\times n}$ the matrices whose columns are $x_i$ and $y_i$, respectively, for $i \in [n]$. The query covariate and response are denoted by $x_{\text{test}} = x_{n+1}$ and $y_{\text{test}} = y_{n+1}$ respectively.

**Linear Self-Attention Transformer:** Following prior works [17, 1, 20], we consider transformers composed of linear self-attention layers. Let $Z_0 \in \mathbb{R}^{2d \times d_p}$ denote the input matrix constructed from $X$, $Y$ and $x_{\text{test}}$ but excluding $y_{\text{test}}$, where $d_p$ denotes the number of tokens. The transformer is defined by stacking $L$ attention blocks with skip connections, where the $l$-th layer is expressed as:

$$Z_l = Z_{l-1} + \tfrac{1}{n}\,\mathrm{Attn}_{V_l,Q_l}(Z_{l-1}), \qquad \mathrm{Attn}_{V,Q}(Z) = VZM\big(Z^\top QZ\big). \tag{1}$$

Here, the trainable parameters are $\{V_l, Q_l\}_{l=1}^L$, where $V_l \in \mathbb{R}^{2d\times 2d}$ represents a reparameterization of the projection and value matrices, and $Q_l \in \mathbb{R}^{2d\times 2d}$ denotes the query and key matrices. Following the work [1], we adopt a masking matrix $M = \mathrm{diag}(I_{d_p-1}, 0)$ to prevent attention from earlier tokens to the final one. The output of the transformer is defined as $\mathsf{TF}\big(Z_0; \{V_l, Q_l\}_{l=1}^L\big) = (Z_L)_{(d+1:2d),d_p}$ (i.e., the latter half of the last column). This definition aligns with the structure of the input $Z_0$, which will be further discussed in subsequent sections. During training, the parameters are optimized to minimize the expected ICL risk over random linear regression instances:

$$\mathcal{L}\big(\{V_l, Q_l\}_{l=1}^L\big) = \mathbb{E}_{Z_0,W}\left\|\mathsf{TF}\big(Z_0; \{V_l, Q_l\}_{l=1}^L\big) + Wx_{\text{test}}\right\|_2^2. \tag{2}$$

# 3 Emergence of Task Vectors in Linear-Attention Transformers

Firstly, we present theoretical evidence indicating that task vectors naturally arise even in simple linear transformers. Specifically, we analyze the loss landscape of the in-context risk, focusing on the properties of its critical points. As a startup, recall the standard linear regression setup [1, 20], where the $(x_i, y_i)$ pairs for each demonstration are concatenated to form the input prompt:

$$Z_0 = \begin{bmatrix} X & x_{\text{test}} \\ Y & 0 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n & x_{\text{test}} \\ y_1 & y_2 & \cdots & y_n & 0 \end{bmatrix} \in \mathbb{R}^{2d\times(n+1)}. \tag{3}$$

According to existing analyses [1, 25, 14], each attention layer in this setting performs one step of gradient descent on the coefficient matrix $W$. Specifically, the theoretically optimal single-layer (possibly nonlinear) attention [10] implements the following predictive function [1] when the covariates are drawn from $P_x = \mathcal{N}(0, I_d)$, by selecting $V_1 \propto \mathrm{diag}(0_{d\times d}, I_d)$ and $Q_1 \propto \mathrm{diag}(I_d, 0_{d\times d})$:

$$\mathsf{TF}(Z_0; (V_1, Q_1)) = -\tfrac{1}{n} Y\sigma(X)^\top \sigma(x_{\text{test}}), \quad \text{where } \sigma : \mathbb{R}^d \mapsto \mathbb{R}^r \text{ is a kernel function.} \tag{4}$$

Here, we abbreviate $[\sigma(x_1) \quad \cdots \quad \sigma(x_n)]$ as $\sigma(X)$. Equation (4) employs $W' \propto Y\sigma(X)^\top$ as an estimate of coefficient matrix $W$, yielding prediction $\hat{y}_{\text{test}} = W'\sigma(x_{\text{test}})$. In this paper, we consider alternative settings more reflective of practical scenarios, where $x_i$ and $y_i$ are separated as distinct tokens. As noted [26], such separation necessitates the usage of position encodings for bi-directional attention. Following prior analysis [11], we assume that position encodings are appended to the input tokens, and reformulate the layer-wise update rule of self-attention as:

$$\mathrm{Attn}_{V,Q}(Z) = VZM\begin{bmatrix} Z^\top & P^\top \end{bmatrix} Q \begin{bmatrix} Z \\ P \end{bmatrix}, \quad \text{where } P \in \mathbb{R}^{d_p\times d_p}. \tag{5}$$

For analytical tractability, we take $P = I_{d_p}$ as one-hot position encodings. Inspired by the parameter structure in [1] and eq. (4), we further impose the following constraints on the trainable parameters:

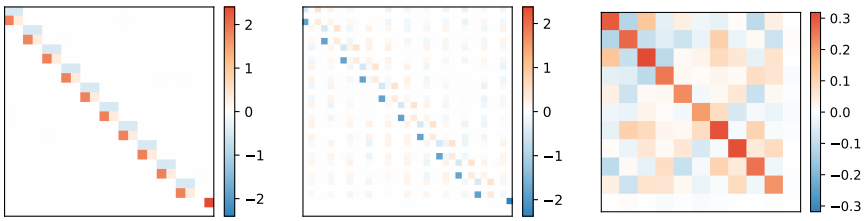
$$V_l = \mathrm{diag}(A_l, B_l), \quad Q_l = \mathrm{diag}(C_l, 0_{d\times d}, D_l), \quad \text{where } A_l, B_l, C_l \in \mathbb{R}^{d\times d},\ D_l \in \mathbb{R}^{d_p\times d_p}. \tag{6}$$

These parameterizations ensure that the projection and attention operations act independently on the covariate, response, and positional components of the input. This structural decoupling is essential for understanding how the transformer identifies the dependency between each $(x_i, y_i)$ pair and revealing the actual optimization algorithm being executed by the model. The proofs for the main theoretical results in this paper are available in Appendix B.

## 3.1 Warm-up: Learning with Pairwise Demonstrations

We begin by analyzing the optimization of linear transformers on pairwise demonstrations. Following previous approach [6, 19, 22], we decompose each demonstration in eq. (3) into a pair of tokens $Z_0^i = \left[\begin{smallmatrix} x_i & 0 \\ 0 & y_i \end{smallmatrix}\right] \in \mathbb{R}^{2d\times 2}$ to better reflect the practical ICL prompt structure:

$$Z_0 = \begin{bmatrix} Z_0^1 & \cdots & Z_0^n & Z_0^{\text{test}} \end{bmatrix} = \begin{bmatrix} x_1 & 0 & \cdots & x_n & 0 & x_{\text{test}} & 0 \\ 0 & y_1 & \cdots & 0 & y_n & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(2d)\times(2n+2)}. \tag{7}$$



(a) $D_l$ (Pairwise) (b) $D_l$ (Triplet) (c) $\Lambda_4\Lambda_4^\top$

Figure 2: Visualization of learned $D_l$ weights. (a) Pairwise demonstrations yield a block-diagonal structure aligned with Theorem 1. (b) Triplet demonstrations yield a richer structure aligned with Theorem 2. (c) The learned matrix $\Lambda_4$ has nearly orthonormal rows as suggested by Proposition 3.

The following theorem suggests that certain critical points of the in-context risk effectively solve the regression problem by first concatenating each pair of $(x_i, y_i)$ into the same tokens, and then executing a variant of the gradient descent algorithm to compute the prediction. To simplify notation, we denote $A = \{A_l\}_{l=1}^L$ (similarly for $B$, $C$, and $D$) and present:

**Theorem 1 (Critical Points; Pairwise Demonstrations).** *Assume $P_x = \mathcal{N}(0, \Sigma)$, $P_w = \mathcal{N}(0, \Sigma^{-1})$ with some $\Sigma \in \mathbb{R}^{d\times d}$ satisfying $\Sigma \succ 0$. Define $\mathcal{S}_I, \mathcal{S}_\Sigma \subset \mathbb{R}^{d\times d}$ and $\mathcal{S}_P \subset \mathbb{R}^{d_p\times d_p}$ as*

$$\mathcal{S}_I = \{\lambda I_d \mid \lambda \in \mathbb{R}\}, \quad \mathcal{S}_\Sigma = \left\{\lambda\Sigma^{-1} \;\middle|\; \lambda \in \mathbb{R}\right\}, \quad \mathcal{S}_P = \left\{\mathrm{diag}(I_n \otimes \Lambda_1, \Lambda_2) \;\middle|\; \Lambda_1, \Lambda_2 \in \mathbb{R}^{2\times 2}\right\}.$$

*Consider optimizing an $L$-layer linear transformer with pairwise demonstrations and parameter configuration given in eq.* (6)*, we then have*

$$\inf_{A,B\in\mathcal{S}_I^L,\, C\in\mathcal{S}_\Sigma^L,\, D\in\mathcal{S}_P^L} \sum_{H\in A\cup B\cup C\cup D} \left\|\nabla_H \mathcal{L}\left(\{V_l, Q_l\}_{l=1}^L\right)\right\|_F^2 = 0.$$

To understand the behavior of these critical points within a self-attention layer, we fix $\Sigma = I_d$ and take $A_l, B_l = I_d$, $C_l = -\lambda I_d$, and $D_l = \mathrm{diag}(I_n \otimes \Lambda_1, \Lambda_2)$. Let the first and last $d$ rows of $Z_l$ be denoted by $X_l$ and $Y_l$, respectively. Under these settings, the update rule of each layer becomes:

$$Z_l = Z_{l-1} - \lambda Z_{l-1} M X_{l-1}^\top X_{l-1} + \begin{bmatrix} Z_{l-1}^1\Lambda_1 & \cdots & Z_{l-1}^n\Lambda_1 & Z_{l-1}^{\text{test}}\,\mathrm{diag}(1,0)\Lambda_2 \end{bmatrix}. \tag{8}$$

The above update can be decomposed into the following two distinct components:

- **Gradient Descent:** The first component, $Z_l \leftarrow Z_{l-1} - \lambda Z_{l-1} M X_{l-1}^\top X_{l-1}$, implements the GD++ algorithm [17]. This variant enhances convergence speed over standard gradient descent by improving the condition number of the Gram matrix $X_{l-1}^\top X_{l-1}$. Notably, this operation modifies only $X_l$ but not $Y_l$ for the first layer, as implied by the structure of $Q_l$ (eq. (6)).
- **Embedding Concatenation:** The second component, $Z_l^i \leftarrow Z_{l-1}^i + Z_{l-1}^i\Lambda_1$ for $i \in [n]$, mixes each pair of $(x_i, y_i)$ tokens. Given that $x_i$ and $y_i$ tokens are initially linearly separable as in our formulation, this operation concatenates each $(x_i, y_i)$ pair, thereby *transforming pairwise demonstrations into the original single-token format*. For the query token $Z_l^{\text{test}}$, this operation copies $x_{\text{test}}$ into the final token, reconstructing the structure in eq. (3), where each non-final token directly concatenates $(x_i, y_i)$ of a demonstration, and the final token contains only $x_{\text{test}}$.

In summary, our analysis reveals that for pairwise demonstrations, the first attention layer leverages position encodings to distinguish between covariate and response tokens, subsequently concatenating them to form a single-token prompt structure. The remaining layers then apply the GD++ algorithm, mirroring the learning dynamics on single-token demonstrations. As a result, **an $L$-layer linear transformer allocates one layer for embedding concatenation and utilizes the remaining $L-1$ layers to perform gradient descent**. In Figure 2a, we visualize the learned $D_l$ weights under the setting of Theorem 1, and observe that they closely match the critical point structure of $\mathcal{S}_P$.

### 3.2 Emergence of Task Vectors with Triplet Demonstrations

Next, to better reflect the prompt structure of practical ICL, we insert additional zero tokens between each pair of $(x_i, y_i)$ to simulate the arrow $(\to)$ tokens. This reformulates each demonstration as a triplet $(x_i, \to, y_i)$, enabling us to analyze the critical points with these triplet demonstrations:

$$Z_0 = \begin{bmatrix} x_1 & 0 & 0 & \cdots & x_n & 0 & 0 & x_{\text{test}} & 0 & 0 \\ 0 & 0 & y_1 & \cdots & 0 & 0 & y_n & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(2d)\times(3n+3)}. \tag{9}$$

**Theorem 2 (Critical Points; Triplet Demonstrations).** *Assume $P_x = \mathcal{N}(0, \Sigma)$, $P_w = \mathcal{N}(0, \Sigma^{-1})$ with some $\Sigma \in \mathbb{R}^{d\times d}$ satisfying $\Sigma \succ 0$. Define $\mathcal{S}_I, \mathcal{S}_\Sigma \subset \mathbb{R}^{d\times d}$ and $\mathcal{S}_P \subset \mathbb{R}^{d_p \times d_p}$ as*

$$\mathcal{S}_I = \{\lambda I_d \mid \lambda \in \mathbb{R}\}, \quad \mathcal{S}_\Sigma = \left\{\lambda \Sigma^{-1} \;\middle|\; \lambda \in \mathbb{R}\right\},$$

$$\mathcal{S}_P = \Big\{ \operatorname{diag}(I_n \otimes \Lambda_1, \Lambda_2) + I_{n+1} \otimes \Lambda_3 + \Lambda_4 \otimes \Lambda_5 \Big| \\ \Lambda_1, \Lambda_2 \in \mathcal{M}\left(\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{smallmatrix}\right), \Lambda_3 \in \mathcal{M}\left(\begin{smallmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{smallmatrix}\right), \Lambda_4 \in \mathbb{R}^{(n+1)\times(n+1)}, \Lambda_5 \in \mathcal{M}\left(\begin{smallmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right) \Big\}.$$

*Consider optimizing an $L$-layer linear transformer with triplet demonstrations and parameter configuration given in eq.* (6)*, we then have*

$$\inf_{A,B\in\mathcal{S}_I^L,\, C\in\mathcal{S}_\Sigma^L,\, D\in\mathcal{S}_P^L} \sum\nolimits_{H\in A\cup B\cup C\cup D} \left\|\nabla_H \mathcal{L}\big(\{V_l, Q_l\}_{l=1}^L\big)\right\|_F^2 = 0.$$

To analyze the behavior of each attention layer, we note that the critical points for the matrices $A_l$, $B_l$, and $C_l$ remain consistent with Theorem 1, thereby implementing the GD++ algorithm. For the matrix $D_l$, we decompose its structure into three distinct components:

- **Embedding Concatenation:** The first component, $\operatorname{diag}(I_n \otimes \Lambda_1, \Lambda_2)$, mixes each pair of $(x_i, y_i)$ tokens, effectively concatenating them — analogous to the operation analyzed in the previous section. This converts all non-arrow tokens into single-token demonstrations.
- **Self Magnification:** The second component, $I_{n+1} \otimes \Lambda_3$, scales the embeddings corresponding to each arrow ($\to$) token by a fixed constant and adds them back to themselves.
- **Task Vector Formation:** The third component, $\Lambda_4 \otimes \Lambda_5$, performs a weighted summation across all demonstrations in the prompt. This operation is central to the emergence of task vectors. Let $[\beta_1 \quad \cdots \quad \beta_{n+1}] \in \mathbb{R}^{n\times(n+1)}$ denote the first $n$ rows of $\Lambda_4$ (we will soon show that the last row of $\Lambda_4$ converges to zero), *the first self-attention layer then outputs $n+1$ linear combinations of the demonstrations as the hidden states for the arrow tokens*, expressed as $z_{\text{tv}}^i = \left[\begin{smallmatrix} \alpha_1 X\beta_i \\ \alpha_2 Y\beta_i \end{smallmatrix}\right]$ for $i \in [n+1]$, where $\alpha_1, \alpha_2 \in \mathbb{R}$ are the two non-zero entries of $\Lambda_5$. These vectors can then be injected into zero-shot prompts and function as single-token demonstrations.

This mechanism provides strong theoretical evidence for our linear combination conjecture, demonstrating that **task vectors naturally emerge from the optimization dynamics of linear-attention transformers operating on triplet-formatted prompts**. Notably, the structure of $\mathcal{S}_P$ closely aligns with our visualization of $D_l$ in Figure 2b, confirming our theoretical analysis. We now further investigate the structure of the weight matrix $\Lambda_4$, and present the following result:

**Proposition 3 (Optimal Task Vector Weights).** *Assume $P_x, P_w = \mathcal{N}(0, I_d)$. Consider optimizing a 2-layer linear-attention transformer with triplet demonstrations and parameter configuration given in eq.* (6)*, and assume $C_1 = 0$. Let*

$$D_1 = \operatorname{diag}(I_n \otimes \Lambda_1, \Lambda_2) + I_{n+1} \otimes \Lambda_3 + \Lambda_4 \otimes \Lambda_5 \in \mathcal{S}_P$$

*be any minimizer of the in-context risk $\mathcal{L}\big(\{V_l, Q_l\}_{l=1}^L\big)$, we then have $\Lambda_4 \in \mathcal{S}_U$, where*

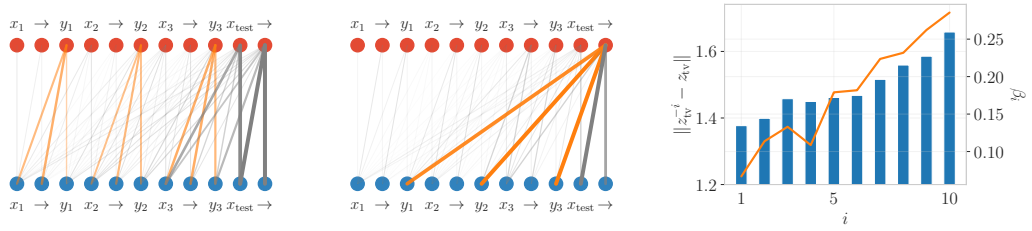
$$\mathcal{S}_U = \left\{\Lambda \;\middle|\; \Lambda\Lambda^\top = \lambda \operatorname{diag}(I_n, 0), \lambda \in \mathbb{R}\right\}.$$

This result suggests that the optimal $\Lambda_4$ weight matrix satisfies two key properties: (1) the last row is zero, and (2) the first $n$ rows are mutually orthonormal. These conditions imply that the learned weight vectors $\beta_1, \cdots, \beta_{n+1}$ are likely to be distinct. Therefore, the $n+1$ task vectors produce diverse linear combinations of the demonstrations, thereby enriching the representation within the input prompt. This implication is verified in Figure 2c. While task vectors are typically extracted from the final arrow ($\to$) token in standard usage, here we consider all arrow tokens as task vectors as bi-directional attention allows each to aggregate information from the full prompt.

# 4 Validating the Linear Combination Conjecture on Bijection Tasks

We then present an empirical observation that supports our conjecture. Consider the setting where task vectors are injected into zero-shot prompts. Based on our prior analysis, the injected task vector $z_{\text{tv}}$ is formed as a linear combination of the original demonstrations. As a result, we show that the injected prompt reconstructs the single-token structure in eq. (3) with only 1 demonstration:

$$Z_0 = [z_{\text{test}} \quad z_{\text{tv}} \quad 0] = \begin{bmatrix} x_{\text{test}} & x_{\text{tv}} & 0 \\ 0 & y_{\text{tv}} & 0 \end{bmatrix} = \begin{bmatrix} x_{\text{test}} & X\beta & 0 \\ 0 & Y\beta & 0 \end{bmatrix} \in \mathbb{R}^{2d\times 3}, \tag{10}$$




(a) Saliency Map ($l = 10$) (b) Saliency Map ($l = 12$) (c) Task Vector Weights

Figure 3: Visualization of saliency matrices as bipartite graphs between layer $l$ (●) and $l + 1$ (●), where edge widths indicate saliency magnitude; and variations in the extracted task vector after perturbing the $i$-th demonstration (▮), alongside the predicted weights (—) obtained by optimizing Proposition 6. (a) Each $y_i$ token is primarily attending to its corresponding $(x_i, y_i)$ pair, reflecting embedding concatenation. (b) The final ($\to$) token attends broadly to all $y_i$ tokens, indicating task vector formation — this occurs just before the optimal injection layer ($l = 13$). (c) The predicted task vector weights closely match the trend of empirical results, validating our theoretical model.

where the weight vector $\beta \in \mathbb{R}^n$ comes from the last column of $\Lambda_4$ (Theorem 2). After the first layer, the $\Lambda_2$ matrix of $\mathcal{S}_P$ moves $x_{\text{test}}$ to the last token, reducing the prompt to a single-shot, single-token demonstration. According to the optimal single-layer transformer (eq. (4)), the estimated coefficient matrix is now $W' = Y\beta(X\beta)^\top$, which is rank-one. Therefore, if our main conjecture holds, task vectors will be inherently limited in their expressiveness: *they can only realize rank-one coefficient matrices*. This implication also naturally extends to multi-layer transformers.

While our analysis is conducted on linear-attention transformers, we demonstrate that similar learning patterns also emerge within practical LLMs. Specifically, we visualize the layer-wise information flow between tokens using saliency maps [18], where the saliency score for each attention matrix is computed as $S(A_l) = \sum_h |A_{l,h} \cdot \partial\mathcal{L}/\partial A_{l,h}|$, $A_{l,h}$ denotes the attention matrix of the $h$-th head at layer $l$, and $\mathcal{L}$ is the ICL loss (i.e., the cross-entropy loss for predicting $y_{\text{test}}$). As demonstrated in Figures 3a and 3b, the saliency maps reveal certain patterns matching the ones of embedding concatenation and weighted summation. Importantly, the latter occurs immediately before the optimal task vector injection layer. This suggests that real-world models implement a similar algorithm to solve ICL tasks and, consequently, inherit the same expressiveness limitation.

To verify this, we construct a specialized class of ICL tasks, named bijection tasks. Specifically, given a bijective mapping from domain $\mathcal{X}$ to codomain $\mathcal{Y}$, one can combine it with its inverse mapping to form a new task that maps $\mathcal{X} \cup \mathcal{Y}$ onto itself. For instance, combining the "to uppercase" task with its inverse "to lowercase" yields a bijection task that maps each letter to its opposite case, and a valid ICL prompt takes the form: *"a $\to$ A, B $\to$ b, c $\to$ C, D $\to$"*. Note that this differs from task superposition [23], as each input corresponds to a unique, well-defined output. We then establish a key limitation of rank-one coefficient matrices in addressing such tasks:

**Proposition 4.** *Let $x, y \in \mathbb{R}^d$ be non-zero vectors. Then the following are equivalent: (1) There exists a rank-one matrix $W \in \mathbb{R}^{d\times d}$ such that $y = Wx$ and $x = Wy$; (2) $x = y$ or $x = -y$.*

This result highlights that *rank-one coefficient matrices cannot solve general bijection tasks*, and are restricted to only the identity mapping ($x = y$) or the negation mapping ($x = -y$). We further verify this implication in real-world LLMs: as summarized in Table 1, both ICL and the task vector method perform well on the original tasks and their inverses. Nevertheless, for the bijection tasks, while ICL preserves performance in many cases, the task vector method consistently fails, confusing examples from the two domains and yielding near-random predictions (50%). For instance, in the "to uppercase" task, task vectors can predict the correct letter but fail to distinguish between uppercase and lowercase. The only notable exceptions are the copy task (corresponding to the $x = y$ case in Proposition 4) and the antonym task (corresponding to $x = -y$).

Together, these findings empirically validate our conjecture: **the task vector approach, which is restricted to rank-one coefficient matrices, cannot solve general bijection tasks**. While a variety of ICL tasks have been explored to assess the capabilities of task vectors [7, 16, 12], the fundamental limitation of task vectors in addressing these bijection tasks has not been previously identified.

Table 1: Comparison of the accuracies of ICL and task vector on bijection tasks (Llama-7B, $n = 10$). We use gray text to indicate accuracies lower than $60\%$.

| **Task** | **Domain $\mathcal{X}$** | **Domain $\mathcal{Y}$** | **Example** | $\mathcal{X} \to \mathcal{Y}$ | | $\mathcal{Y} \to \mathcal{X}$ | | $\mathcal{X} \leftrightarrow \mathcal{Y}$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | ICL | TV | ICL | TV | ICL | TV |
| To Upper | $\{a, \cdots, z\}$ | $\{A, \cdots, Z\}$ | a $\to$ A | 1.00 | 0.91 | 1.00 | 0.99 | 1.00 | 0.55 |
| Translation | English | French | hello $\to$ bonjour | 0.83 | 0.84 | 0.82 | 0.70 | 0.54 | 0.35 |
| | English | Italian | hello $\to$ ciao | 0.84 | 0.78 | 0.82 | 0.74 | 0.70 | 0.47 |
| | English | Spanish | hello $\to$ hola | 0.92 | 0.88 | 0.89 | 0.75 | 0.64 | 0.43 |
| Linguistic | Present | Gerund | go $\to$ going | 0.99 | 0.95 | 1.00 | 0.97 | 0.80 | 0.41 |
| | Present | Past | go $\to$ went | 0.98 | 0.91 | 0.99 | 0.96 | 0.52 | 0.33 |
| | Present | Past Perfect | go $\to$ gone | 0.82 | 0.82 | 0.94 | 0.65 | 0.55 | 0.33 |
| | Singular | Plural | dog $\to$ dogs | 0.88 | 0.78 | 0.94 | 0.89 | 0.76 | 0.51 |
| Copy | $\{a, \cdots, z, A, \cdots, Z\}$ | | A $\to$ A | - | | - | | 1.00 | 0.98 |
| Antonym | Adjectives | | happy $\to$ sad | 0.89 | 0.83 | - | | 0.83 | 0.73 |

# 5 Further Discussions

**Inseparable Covariates and Responses.** In our main analysis, we assume that $x_i$ and $y_i$ embeddings are linearly separable, allowing the addition $x_i + y_i$ to act a concatenation operation. However, recognizing that this assumption does not generally hold for real-world transformers, we extend our analysis to the following setting, where $x_i$ and $y_i$ are no longer linearly separable. While this still imposes a $2d$-dimensional requirement on the hidden space, such a constraint is easily satisfied in practical transformers, given the high dimensionality of their internal representations.

$$Z_0 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ x_1 & y_1 & \cdots & x_n & y_n & x_{\text{test}} & 0 \end{bmatrix} \in \mathbb{R}^{(2d)\times(2n+2)}. \tag{11}$$

We slightly modify the sparsity constraints for the first layer, and require $(D_0)_{2i,:} = 0$ for $i \in [n+1]$:

$$V_0 = \begin{bmatrix} 0 & A_0 \\ 0_{d\times d} & 0 \end{bmatrix}, \quad Q_0 = \begin{bmatrix} 0_{2d\times 2d} & 0 \\ 0 & D_0 \end{bmatrix}, \quad \text{where } A_0 \in \mathbb{R}^{d\times d},\ D_0 \in \mathbb{R}^{d_p\times d_p}. \tag{12}$$

With these conditions, we are ready to establish the critical points for inseparable demonstrations. Note that $V_0$ and $Q_0$ do not involve $B_0$ and $C_0$, so the sequences $B$ and $C$ have size $L-1$.

**Theorem 5.** *Under the same settings as Theorem 1, define $\mathcal{S}_I, \mathcal{S}_\Sigma \subset \mathbb{R}^{d\times d}$ and $\mathcal{S}_P \subset \mathbb{R}^{d_p\times d_p}$ as*

$$\mathcal{S}_I = \{\lambda I_d \mid \lambda \in \mathbb{R}\}, \quad \mathcal{S}_\Sigma = \left\{\lambda\Sigma^{-1} \,\middle|\, \lambda \in \mathbb{R}\right\}, \quad \mathcal{S}_P = \left\{\text{diag}(I_n \otimes \Lambda_1, \Lambda_2) \,\middle|\, \Lambda_1, \Lambda_2 \in \mathbb{R}^{2\times 2}\right\}.$$

*Consider optimizing an $L$-layer linear transformer with inseparable pairwise demonstrations and parameter configuration given in eq.* (12) *for the first layer and eq.* (6) *for the remaining layers, then*

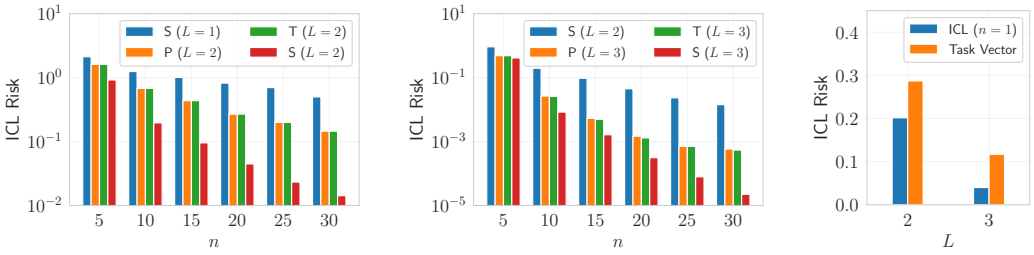
$$\inf_{A\in\mathcal{S}_I^L,\, B\in\mathcal{S}_I^{L-1},\, C\in\mathcal{S}_\Sigma^{L-1},\, D\in\mathcal{S}_P^L} \sum\nolimits_{H\in A\cup B\cup C\cup D} \left\|\nabla_H \mathcal{L}\left(\{V_l, Q_l\}_{l=1}^L\right)\right\|_F^2 = 0.$$

This result suggests that for inseparable demonstrations, the first layer performs a functionally similar concatenation operation by "moving" the embedding of each $x_i$ to the corresponding $y_i$ position. This enables the model to reconstruct the single-token structure without linear separability.

**Optimal Weights for Causal Task Vectors.** While task vectors naturally emerge in linear transformers, their embeddings do not directly help minimize the ICL risk, as evidenced by the identical performance between pairwise and triplet formatted prompts (Figures 4a and 4b). Instead, we show that task vectors contribute to minimizing the training (i.e., LLM pretraining) risk when token-wise dropout is applied, acting as redundancies for in-context demonstrations that may be randomly dropped during training. This redundancy ensures that essential task information is preserved and continues to facilitate accurate prediction despite partial context loss.

**Proposition 6.** *Under the same settings as Proposition 3, consider adding token-wise dropouts $O_l$:*

$$Z_l = Z_{l-1}O_l + \tfrac{1}{n}\,\text{Attn}_{V_l,Q_l}(Z_{l-1})O_l, \quad \textit{where } O_l = \text{diag}(o_l^1, \cdots, o_l^{d_p}),\ o_l^i \overset{\textit{i.i.d.}}{\sim} \text{Bern}(p).$$




(a) Single- vs. Multi-Token ($L = 2$) (b) Single- vs. Multi-Token ($L = 3$) (c) ICL vs. TV

Figure 4: (a, b) Comparison of the best ICL risk achieved using single (S), pairwise (P), and triplet (T) formatted prompts. (c) Performance comparison between 1-shot ICL and task vector.

*Then any minimizer $\Lambda_4$ of the in-context risk $\mathcal{L}\big(\{V_l, Q_l\}_{l=1}^L\big)$ satisfies $(\Lambda_4)_{n+1,:} = 0$ and:*

$$(\Lambda_4)_{1:n,:} \propto \arg\min_{\Lambda} \; c_1\|\Lambda\|_4^4 + c_2 \sum_{i=1}^{n} \|\Lambda_{i,:}\|_2^4 + c_3 \sum_{j=1}^{n+1} \|\Lambda_{:,j}\|_2^4 + c_4 \big\|\Lambda\Lambda^\top\big\|_F^2, \;\; \textit{s.t.}\; \|\Lambda\|_F^2 = 1.$$

*where $c_1, \cdots, c_4$ are non-negative constants depending on $V_l$, $Q_l$, and $p$.*

This result suggests that dropout introduces additional higher-order regularization on the task vector weights, encouraging them to distribute more uniformly across demonstrations. Furthermore, when considering causal attention (i.e., enforcing $\Lambda_4$ to be upper-triangular), it induces a decaying weight pattern from later to earlier demonstrations, which is also consistently observed in practical transformer models as evidenced in Figure 3c. While dropout is not always applied during LLM pre-training or fine-tuning, the injection of position encodings and use of normalization act as alternative sources of perturbation, thereby promoting the emergence of such redundancy.

**Extra EOS Tokens.** In our theoretical analysis, we consistently impose an additional zero token at the end of the input prompt. While this token can be interpreted as an EOS token in practical models, such a design choice is uncommon in standard ICL tasks. We justify this modeling decision with:

**Proposition 7** (Informal)**.** *Given any $L$-layer, single-head, $d$-dimensional linear-attention transformer with EOS tokens, there exists an equivalent $L$-layer, two-head, $2d$-dimensional linear-attention transformer operating without EOS tokens.*

This equivalence suggests that the same learning dynamics can be realized through multi-head architectures without relying on explicit EOS tokens. Specifically, one head in this setting is dedicated to task vector formation, while the other handles ICL prediction. This separation allows the model to retain the functional role of the EOS token implicitly within its hidden states. Consequently, our prior theoretical analysis can be naturally extended to practical models that omit explicit EOS tokens.

# 6 Experimental Studies

## 6.1 Synthetic Results with Random Linear Regression

In this section, we validate our critical points analysis with synthetic linear regression tasks. Specifically, we examine the achievable ICL risk of linear transformers trained with single-token (eq. (3)), pairwise (eq. (7)), and triplet (eq. (9)) demonstrations. We set the input dimension to $d = 4$ and $P_x = P_w = \mathcal{N}(0, I_d)$. For each setting, we train multiple models with different random seeds and report the minimum ICL risk achieved as a proxy for the global optimum. The comparative results across different numbers of layers $L$ and demonstration formats are shown in Figures 4a and 4b.

These results support our theoretical analysis: when trained with pairwise or triplet demonstrations, the transformer recovers the GD++ algorithm similar to the single-token case. Notably, the performance of $L$-layer transformers with pairwise (P) and triplet (T) demonstrations closely aligns, indicating a shared underlying learning pattern. Moreover, their performance consistently lies between that of single-token (S) case $L$-layer and $(L-1)$-layer models. The observed improvement over the $(L-1)$-layer single-token baselines comes from the additional GD++ performed solely on $x_i$ tokens in the first layer, effectively acting as a "half-step" of gradient descent.

Table 2: Accuracy comparison between standard ICL (Baseline), the task vector method (TaskV), and our strategy (TaskV-M). The experiment is conducted on Llama-13B with $n = 10$.

| Method | | Knowledge | Algorithmic | Translation | Linguistic | Bijection | Average |
|---|---|---|---|---|---|---|---|
| 0-shot | Baseline | 6.90 ± 2.08 | 15.60 ± 1.72 | 7.00 ± 1.65 | 12.44 ± 1.74 | 8.27 ± 1.33 | 10.28 ± 0.98 |
| | TaskV | **68.80** ± 2.66 | **86.20** ± 1.61 | **73.53** ± 0.91 | **85.24** ± 1.80 | **50.67** ± 2.32 | **72.26** ± 1.01 |
| 1-shot | Baseline | 69.50 ± 3.86 | 73.67 ± 1.56 | 57.80 ± 2.01 | 56.22 ± 1.57 | 44.76 ± 2.44 | 58.11 ± 0.63 |
| | TaskV | 79.50 ± 2.35 | 88.47 ± 0.75 | **80.67** ± 2.56 | **89.11** ± 0.84 | 60.44 ± 2.07 | 78.79 ± 0.77 |
| | TaskV-M | **81.30** ± 2.80 | **89.53** ± 0.65 | 80.13 ± 2.14 | 88.71 ± 0.62 | **61.78** ± 0.96 | **79.34** ± 0.37 |
| 2-shot | Baseline | 78.80 ± 3.30 | 85.07 ± 1.37 | 75.67 ± 2.64 | 76.80 ± 1.18 | 56.49 ± 2.87 | 72.92 ± 0.59 |
| | TaskV | 84.60 ± 2.11 | 88.40 ± 0.68 | **84.33** ± 0.92 | **90.13** ± 0.92 | 62.44 ± 2.16 | 80.82 ± 0.42 |
| | TaskV-M | **85.70** ± 1.63 | **89.27** ± 1.10 | 84.13 ± 1.15 | 89.64 ± 0.86 | **64.49** ± 2.02 | **81.48** ± 0.37 |
| 3-shot | Baseline | 86.20 ± 2.69 | 88.07 ± 1.06 | 80.00 ± 1.67 | 84.04 ± 1.19 | 62.18 ± 1.52 | 78.51 ± 0.42 |
| | TaskV | 90.20 ± 2.23 | 88.67 ± 0.89 | **86.27** ± 2.31 | 92.31 ± 0.48 | 66.53 ± 0.94 | 83.53 ± 0.41 |
| | TaskV-M | **90.30** ± 1.50 | **89.87** ± 0.83 | 86.07 ± 2.17 | **92.36** ± 0.72 | **68.13** ± 0.76 | **84.15** ± 0.52 |
| 4-shot | Baseline | 84.80 ± 2.06 | 88.07 ± 0.61 | 83.27 ± 1.82 | 88.89 ± 1.91 | 67.16 ± 1.47 | 81.52 ± 0.66 |
| | TaskV | 88.70 ± 1.69 | 89.53 ± 1.34 | 86.27 ± 1.08 | **92.76** ± 0.54 | 70.44 ± 1.35 | 84.66 ± 0.39 |
| | TaskV-M | **89.60** ± 1.43 | **91.00** ± 1.01 | **87.20** ± 0.62 | 92.36 ± 1.44 | **72.53** ± 0.94 | **85.64** ± 0.29 |

Additionally, we successfully reproduce the task vector method in linear transformers. Specifically, we extract the hidden state of the final ($\rightarrow$) token from triplet demonstrations after the first layer, and inject this vector into zero-shot prompts consisting of only $x_{\text{test}}$. To simulate the effect of layer normalization used in practical transformers, we normalize the task vectors before inference and the output vectors before ICL risk evaluation. As shown in Figure 4c, the performance of task vectors is parallel to that of standard ICL with a single in-context example. This validates our conjecture that the injected task vector effectively acts as a single demonstration.

### 6.2 Enhancing the Task Vector Method

We further explore an enhancement to the original task vector method. According to our previous analysis, a single injected task vector may not provide sufficient information for inference on complex tasks (e.g., bijection tasks). Moreover, in linear-attention models, each ($\rightarrow$) token functions as an individual in-context demonstration during the gradient descent phase and thus contributes equally to the ICL risk. Motivated by this, we extend the standard task vector method, which modifies only the final arrow token, and propose a multi-vector variant that injects into every single arrow token in few-shot prompts. This enriched injection scheme enables the model to leverage multiple new demonstrations, thereby providing a more informative and distributed context for prediction.

We compare our multi-vector injection strategy (TaskV-M) against standard $N$-shot ICL (Baseline) and the original task vector method (TaskV). For each $N$-shot prompt, we generate $N + 1$ distinct ICL prompts to produce $N + 1$ task vectors, which are then used to replace the embeddings of all arrow tokens in the input. For each task, performance is evaluated over $50$ randomly sampled prompts, with mean accuracy and standard deviation reported across $5$ independent trials. The final results, summarized in Table 2, span a diverse set of ICL task types, including Knowledge, Algorithmic, Translation, Linguistic, and Bijection, showing that TaskV-M consistently outperforms TaskV, especially on the more challenging bijection tasks. These findings support our analysis that every arrow token contributes meaningfully to the model's ICL capability.

## 7 Conclusion

This paper proposes the linear combination conjecture as a plausible explanation for the emergence and functionality of task vectors in ICL. We support this conjecture with both empirical observations and theoretical analysis, demonstrating how task vectors naturally arise under triplet-formatted demonstrations in simple linear transformer models, and why this method inherently fails on general bijection tasks. While the conjecture may not yet offer a complete characterization of ICL dynamics, it provides a new perspective on the underlying mechanisms and offers a promising direction for interpreting intermediate hidden states in modern transformer-based language models.

# References


[1] 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, 2023.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891, 2022.

[4] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, 2023.

[5] Gilad Deutch, Nadav Magar, Tomer Natan, and Guy Dar. In-context learning and gradient descent revisited. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1017–1028, 2024.

[6] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[7] Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333, 2023.

[8] Alberto Hojel, Yutong Bai, Trevor Darrell, Amir Globerson, and Amir Bar. Finding visual task vectors. In *European Conference on Computer Vision*, pages 257–273. Springer, 2024.

[9] Brandon Huang, Chancharik Mitra, Leonid Karlinsky, Assaf Arbelle, Trevor Darrell, and Roei Herzig. Multimodal task vectors enable many-shot multimodal in-context learning. *Advances in Neural Information Processing Systems*, 37:22124–22153, 2024.

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

[11] Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928, 2023.

[12] Dongfang Li, Xinshuo Hu, Zetian Sun, Baotian Hu, Min Zhang, et al. In-context learning state vector with inner and momentum optimization. *Advances in Neural Information Processing Systems*, 37:7797–7820, 2024.

[13] Sheng Liu, Haotian Ye, Lei Xing, and James Y Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *International Conference on Machine Learning*, pages 32287–32307. PMLR, 2024.

[14] Arvind V. Mahankali, Tatsunori Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *The Twelfth International Conference on Learning Representations*, 2024.

[15] Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. Position: Do pretrained transformers learn in-context by gradient descent? In *Proceedings of the 41st International Conference on Machine Learning*, pages 44712–44740. PMLR, 2024.

[16] 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.

[17] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

[18] Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, 2023.

[19] Kevin Christian Wibisono and Yixin Wang. On the role of unstructured training data in transformers' in-context learning capabilities. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

[20] Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? In *The Twelfth International Conference on Learning Representations*, 2024.

[21] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022.

[22] Yue Xing, Xiaofeng Lin, Chenheng Xu, Namjoon Suh, Qifan Song, and Guang Cheng. Theoretical understanding of in-context learning in shallow transformers with unstructured data. *arXiv preprint arXiv:2402.00743*, 2024.

[23] Zheyang Xiong, Ziyang Cai, John Cooper, Albert Ge, Vasilis Papageorgiou, Zack Sifakis, Angeliki Giannou, Ziqian Lin, Liu Yang, Saurabh Agarwal, et al. Everything everywhere all at once: Llms can in-context learn multiple tasks in superposition. *arXiv preprint arXiv:2410.05603*, 2024.

[24] Liu Yang, Ziqian Lin, Kangwook Lee, Dimitris Papailiopoulos, and Robert Nowak. Task vectors in in-context learning: Emergence, formation, and benefit. *arXiv preprint arXiv:2501.09240*, 2025.

[25] 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.

[26] Chunsheng Zuo, Pavel Guerzhoy, and Michael Guerzhoy. Position information emerges in causal transformers without positional encodings via similarity of nearby embeddings. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9418–9430, 2025.

# A Auxiliary Lemmas

**Lemma 8** (Proposed in [1]). *Given positive objective function $f(A)$ taking parameters $A = \{A_i\}_{i=1}^n$, where $A_i \in \mathbb{R}^{d_i \times d_i}$. Let $\mathcal{S} = \Pi_{i=1}^n \mathcal{S}_i \subset \Pi_{i=1}^n \mathbb{R}^{d_i \times d_i}$ be a predefined parameter subspace. Define $\widetilde{A}(t, R_i) = \{A_1, \cdots, A_i + tR_i, \cdots, A_n\}$ given $i \in [1, n]$, $R_i \in \mathbb{R}^{d_i \times d_i}$ and $t \in \mathbb{R}$. If for any $A \in \mathcal{S}$ and $R_i \in \mathbb{R}^{d_i \times d_i}$, there exists $\widetilde{R}_i \in \mathcal{S}_i$ such that*

$$\frac{\mathrm{d}}{\mathrm{d}t} f\Big(\widetilde{A}(t, \widetilde{R}_i)\Big)\Big|_{t=0} \leq \frac{\mathrm{d}}{\mathrm{d}t} f\Big(\widetilde{A}(t, R_i)\Big)\Big|_{t=0},$$

*then we have*

$$\inf_{A \in \mathcal{S}} \sum_{i=1}^n \|\nabla_{A_i} f(A)\|_F^2 = 0.$$

*Proof.* This lemma is proved as part of the main theorems in [1]. We rearrange the proof here to accommodate arbitrary function of matrices. Firstly, notice that for any $R = \{R_i\}_{i=1}^n \in \Pi_{i=1}^n \mathbb{R}^{d_i \times d_i}$,

$$\sum_{i=1}^n \frac{\mathrm{d}}{\mathrm{d}t} f\Big(\widetilde{A}(t, \widetilde{R}_i)\Big)\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t} f(A + tR)\Big|_{t=0}.$$

Therefore, the provided precondition is equivalent to stating that for any $A \in \mathcal{S}$ and $R \in \Pi_{i=1}^n \mathbb{R}^{d_i \times d_i}$, there exists $\widetilde{R} \in \mathcal{S}$ such that:

$$\frac{\mathrm{d}}{\mathrm{d}t} f\Big(A + t\widetilde{R}\Big)\Big|_{t=0} \leq \frac{\mathrm{d}}{\mathrm{d}t} f(A + tR)\Big|_{t=0}.$$

Let $R = -\nabla_A f(A)$, we then have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} f(A + tR)\Big|_{t=0} &= \left\langle \frac{\mathrm{d}f(A - t\nabla_A f(A))}{\mathrm{d}(A - t\nabla_A f(A))}, \frac{\mathrm{d}(A - t\nabla_A f(A))}{t} \right\rangle\Big|_{t=0} \\ &= \langle \nabla_A f(A), -\nabla_A f(A) \rangle = -\|\nabla_A f(A)\|_F^2.\end{aligned}$$

If the infimum of $\|\nabla_A f(A)\|_F^2$ is not zero but some positive value $p$, then the $\mathcal{S}$-constrained gradient flow induced by $\widetilde{R}$ will lead to unbounded descent:

$$\frac{\mathrm{d}}{\mathrm{d}t} f\Big(A + t\widetilde{R}\Big)\Big|_{t=0} \leq -p.$$

This contradicts the fact that $f(A) \geq 0$ and concludes the proof. □

The following lemma is an extension of Lemma 5 in [1] by accommodating multivariate $y$ samples as well as enabling a wider range of demonstration and transformer parameter configurations.

**Lemma 9.** *Let $x_1, \cdots, x_{n+1}$ be i.i.d. samples from an input distribution, and let $W$ be sampled independently of $\{x_i\}_{i=1}^{n+1}$. Let $Z_0 \in \mathbb{R}^{(2d) \times N}$, where $N \in \mathbb{Z}$, be constructed of form*

$$Z_0 = \begin{bmatrix} * & \cdots & * & * \\ * & \cdots & * & 0_d \end{bmatrix} \in \mathbb{R}^{(2d) \times N},$$

*where the $*$ parts can be arbitrarily constructed from $\{x_i\}_{i=1}^{n+1}$ and $W$. Let $\widetilde{Z}_0$ be defined as replacing the zero part of $Z_0$ by $y_{n+1}$:*

$$\widetilde{Z}_0 = \begin{bmatrix} * & \cdots & * & * \\ * & \cdots & * & y_{n+1} \end{bmatrix} \in \mathbb{R}^{(2d) \times N}.$$

*Let $\widetilde{Z}_l$ be the output of the $l$-th layer of the linear transformer, and let $\widetilde{X}_l, \widetilde{Y}_l \in \mathbb{R}^{d \times N}$ be the first and last $d$ rows of $\widetilde{Z}_l$, respectively. Suppose that the $\{Q_l\}_{l=1}^L$ matrices are of form*

$$Q_l = \begin{bmatrix} \underbrace{*}_{d \text{ columns}} & 0_{(2d+d_p) \times d} & \underbrace{*}_{d_p \text{ columns}} \end{bmatrix},$$

*Then the in-context risk of this $L$-layer linear transformer is equivalent to*

$$\mathcal{L}\big(\{V_l, Q_l\}_{l=1}^L\big) = \mathbb{E}_{\widetilde{Z}_0, W}\Big[\mathrm{tr}\Big((I_N - M)\widetilde{Y}_L^\top \widetilde{Y}_L (I_N - M)\Big)\Big]. \tag{13}$$

*Proof.* Let the $V_l$ and $Q_l$ matrices be represented as:

$$V_l = \begin{bmatrix} V_l^1 \\ V_l^2 \end{bmatrix}, \quad Q_l = \begin{bmatrix} Q_l^1 & 0 & Q_l^2 \end{bmatrix},$$

where $V_l^1, V_l^2 \in \mathbb{R}^{d\times 2d}, Q_l^1 \in \mathbb{R}^{(2d+d_p)\times d}, Q_l^2 \in \mathbb{R}^{(2d+d_p)\times d_p}$. Then the update rule in eq. (5) can be rephrased as

$$\begin{aligned} X_l &= X_{l-1} + \frac{1}{n} V_l^1 Z_{l-1} M \big[Z_{l-1}^\top, P\big] \big(Q_l^1 X_{l-1} + Q_l^2 P\big), \\ Y_l &= Y_{l-1} + \frac{1}{n} V_l^2 Z_{l-1} M \big[Z_{l-1}^\top, P\big] \big(Q_l^1 X_{l-1} + Q_l^2 P\big). \end{aligned}$$

Let $\Delta_Z = \widetilde{Z}_0 - Z_0$, i.e. an all-zero matrix except that the last half of the last column is $y_{n+1}$. Let $\Delta_X$ and $\Delta_Y$ be its first and last $d$ rows respectively, then $\Delta_X = 0$ and $\Delta_Y = [0 \quad \cdots \quad 0 \quad y_{n+1}]$. Note that $\widetilde{Z}_l = Z_l + \Delta_Z$ holds for $l = 0$ trivially. Now suppose it holds for some $l = k-1$, then

$$\begin{aligned} \widetilde{X}_k &= \widetilde{X}_{k-1} + \frac{1}{n} V_k^1 \widetilde{Z}_{k-1} M \Big[\widetilde{Z}_{k-1}^\top, P\Big] \Big(Q_k^1 \widetilde{X}_{k-1} + Q_k^2 P\Big) \\ &= X_{k-1} + \frac{1}{n} V_k^1 Z_{k-1} M \big[Z_{k-1}^\top, P\big] \big(Q_k^1 X_{k-1} + Q_k^2 P\big) \\ &\quad + \frac{1}{n} V_k^1 \Delta_Z M \big[Z_{k-1}^\top, P\big] \big(Q_k^1 X_{k-1} + Q_k^2 P\big) \\ &\quad + \frac{1}{n} V_k^1 Z_{k-1} M \big[\Delta_Z^\top, 0_{d_p\times d_p}\big] \big(Q_k^1 X_{k-1} + Q_k^2 P\big) \\ &\quad + \frac{1}{n} V_k^1 \Delta_Z M \big[\Delta_Z^\top, 0_{d_p\times d_p}\big] \big(Q_k^1 X_{k-1} + Q_k^2 P\big) \\ &= X_{k-1} + \frac{1}{n} V_k^1 Z_{k-1} M \big[Z_{k-1}^\top, P\big] \big(Q_k^1 X_{k-1} + Q_k^2 P\big) = X_k, \end{aligned}$$

where the last step holds by noticing that $\Delta_Z M = 0$. Similarly, one can prove that

$$\widetilde{Y}_k = Y_{k-1} + \Delta_Y + \frac{1}{n} V_k^2 Z_{k-1} M \big[Z_{k-1}^\top, P\big] \big(Q_k^1 X_{k-1} + Q_k^2 P\big) = Y_k + \Delta_Y.$$

Therefore, it holds that for any $l \in [1, L]$, $\widetilde{Z}_l = Z_l + \Delta_Z$. Recall the in-context risk in eq. (2):

$$\begin{aligned} \mathcal{L}\big(\{V_l, Q_l\}_{l=1}^L\big) &= \mathbb{E}_{Z_0, W} \left\| (Z_L)_{(d+1:2d),N} + y_{n+1} \right\|_2^2 \\ &= \mathbb{E}_{Z_0, W} \left\| (Y_L + \Delta_Y)(I_N - M) \right\|_2^2 \\ &= \mathbb{E}_{\widetilde{Z}_0, W} \Big[ \mathrm{tr}\Big( (I_N - M) \widetilde{Y}_L^\top \widetilde{Y}_L (I_N - M) \Big) \Big]. \end{aligned}$$

The proof is complete. □

# B Proof of Theoretical Results

## B.1 Proof of Proposition 4

*Proof.* We will first prove sufficiency. Let $W = ab^\top$ be a rank-one matrix, where $a, b \in \mathbb{R}^d$. The given conditions imply that $x = Wy = WWx = ab^\top ab^\top x$, we then have $b^\top x = b^\top ab^\top ab^\top x = (b^\top a)^2 b^\top x$. Since $b^\top x \neq 0$, we can conclude that $b^\top a = \pm 1$. Then, $x = ab^\top ab^\top x = \pm ab^\top x = \pm y$.

To prove the necessity, it suffices to show that selecting $W = xx^\top / \|x\|_2^2$ when $x = y$ satisfies the given conditions (alternatively, select $W = -xx^\top / \|x\|_2^2$ when $x = -y$). □

## B.2 Proof of Theorem 1

*Proof.* To enhance the readability of the notations in this proof, we will drop the constant $\frac{1}{n}$ factor in linear attention. Furthermore, we will simplify $\widetilde{Z}_0$, $\widetilde{X}_0$ and $\widetilde{Y}_0$ in Lemma 9 as $Z_0$, $X_0$ and $Y_0$

respectively. This results in different definitions compared to the original ones, but we will not refer to the original definitions in the remainder of this proof.

$$Z_0 = \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \begin{bmatrix} x_1 & 0 & \cdots & x_n & 0 & x_{\text{test}} & 0 \\ 0 & y_1 & \cdots & 0 & y_n & 0 & y_{\text{test}} \end{bmatrix} \in \mathbb{R}^{(2d)\times(2n+2)}.$$

Let $Z_l$ be the output of the $l$-th layer of the transformer, and let $X_l, Y_l \in \mathbb{R}^{d\times(2n+2)}$ denote the first and last $d$ rows of $Z_l$, respectively. Under the constraint in eq. (6), we can verify that

$$\begin{aligned} X_l &= X_{l-1} + A_l X_{l-1} M (X_{l-1}^\top C_l X_{l-1} + D_l), \\ Y_l &= Y_{l-1} + B_l Y_{l-1} M (X_{l-1}^\top C_l X_{l-1} + D_l). \end{aligned} \tag{14}$$

In the following analysis, we will use $f(A \leftarrow B)$ to denote the result of the function $f$ of $A$ when replacing the value of $A$ with $B$. Additionally, we denote $f(A \leftarrow B * A)$ as $f(A \overset{*}{\leftarrow} B)$ for any operator $*$. Therefore, $f(A \overset{+}{\leftarrow} B) = f(A \leftarrow A + B)$. We also denote $f(A \overset{\times}{\leftarrow} B) = f(A \leftarrow BA)$ and $f(A \overset{\diamond}{\leftarrow} B) = f(A \leftarrow AB)$ for convenience.

Our goal is proving that, for any $E \in A \cup B \cup C \cup D$ and an arbitrary matrix $R \in \mathbb{R}^{d\times d}$ ($\mathbb{R}^{d_p\times d_p}$ for $D$), there exists $\widetilde{R} \in \mathcal{S}_I$ ($\mathcal{S}_\Sigma$ for $C$, $\mathcal{S}_P$ for $D$) such that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(E \overset{+}{\leftarrow} t\widetilde{R})\Big|_{t=0} \le \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(E \overset{+}{\leftarrow} tR)\Big|_{t=0}. \tag{15}$$

Let $\overline{X}_0 = [0, x_1, \cdots, 0, x_{\text{test}}]$ be a function of $X_0$, we then have $Y_0 = W\overline{X}_0$. Let $U_\perp \in \mathbb{R}^{d\times d}$ be a uniformly sampled random orthonormal matrix, and let $U_\Sigma = \Sigma^{1/2} U_\perp \Sigma^{-1/2}$. One can verify that $U_\Sigma^{-1} = \Sigma^{1/2} U_\perp^\top \Sigma^{-1/2}$. By applying Lemma 9 and the fact that $X_0 \overset{d}{=} U_\Sigma X_0$, we have that for any given matrix $R$,

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(E \overset{+}{\leftarrow} tR)\Big|_{t=0} \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\,\mathbb{E}_{X_0,W}\Big[\mathrm{tr}\Big((I-M)Y_L^\top (E \overset{+}{\leftarrow} tR) Y_L (E \overset{+}{\leftarrow} tR)(I-M)\Big)\Big]\Big|_{t=0} \\ &= 2\,\mathbb{E}_{X_0,W}\left[\mathrm{tr}\left((I-M)Y_L^\top \frac{\mathrm{d}}{\mathrm{d}t} Y_L(E \overset{+}{\leftarrow} tR)\Big|_{t=0} (I-M)\right)\right] \\ &= 2\,\mathbb{E}_{X_0,W,U_\perp}\left[\mathrm{tr}\left((I-M)Y_L^\top (X_0 \overset{\times}{\leftarrow} U_\Sigma) \frac{\mathrm{d}}{\mathrm{d}t} Y_L(X_0 \overset{\times}{\leftarrow} U_\Sigma, E \overset{+}{\leftarrow} tR)\Big|_{t=0} (I-M)\right)\right]. \end{aligned}$$

Next, we will show that eq. (15) holds for each one of $A_i, B_i, C_i, D_i$ for any $i \in [1, L]$.

**1. Equation (15) holds for $A_i$.**

We first show that for any $l \in [1, L]$, the following equations hold:

$$X_l(X_0 \overset{\times}{\leftarrow} U_\Sigma) = U_\Sigma X_l, \tag{16}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} X_l(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} = U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} X_l(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0}. \tag{17}$$

It is straightforward to verify that eq. (16) holds for $l = 0$. Now suppose that eq. (16) holds for some $l = k-1$, we then have

$$\begin{aligned} &X_k(X_0 \overset{\times}{\leftarrow} U_\Sigma) \\ &= X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) + A_l X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) M\Big(X_{k-1}^\top (X_0 \overset{\times}{\leftarrow} U_\Sigma) C_l X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) + D_l\Big) \\ &= U_\Sigma X_{k-1} + A_l U_\Sigma X_{k-1} M\big(X_{k-1}^\top U_\Sigma^\top C_l U_\Sigma X_{k-1} + D_l\big) \\ &= U_\Sigma\big(X_{k-1} + A_l X_{k-1} M\big(X_{k-1}^\top C_l X_{k-1} + D_l\big)\big) = U_\Sigma X_k, \end{aligned}$$

where the third equality follows by noticing that when $A_l = a_l I_d$ and $C_l = c_l \Sigma^{-1}$, we have $A_l U_\Sigma = U_\Sigma A_l$ and $U_\Sigma^\top C_l U_\Sigma = C_l$. This concludes the proof of eq. (16).

We now turn to the proof of eq. (17). Notice that when $l < i$, we naturally have

$$\frac{\mathrm{d}}{\mathrm{d}t} X_l(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} = U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} X_l(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} = 0.$$

When $l = i$, it is easy to verify that

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} X_l(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} &= R U_\Sigma X_{l-1} M (X_{l-1}^\top U_\Sigma^\top C_l U_\Sigma X_{l-1} + D_l) \\
&= U_\Sigma \cdot U_\Sigma^{-1} R U_\Sigma M (X_{l-1}^\top C_l X_{l-1} + D_l) \\
&= U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} X_l(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0}.
\end{aligned}$$

Now suppose that eq. (17) holds for some $l = k - 1 \geq i$, one can verify that:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t} X_k(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} + \frac{\mathrm{d}}{\mathrm{d}t} A_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR) M \\
&\quad \cdot \Big( X_{k-1}^\top (X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR) C_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR) + D_k \Big)\Big|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\
&\quad + A_k \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} M \Big( X_{k-1}^\top (X_0 \overset{\times}{\leftarrow} U_\Sigma) C_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) + D_k \Big) \\
&\quad + A_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) M \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}^\top (X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} C_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) \\
&\quad + A_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) M X_{k-1}^\top (X_0 \overset{\times}{\leftarrow} U_\Sigma) C_k \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\
&= U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} \\
&\quad + U_\Sigma A_k \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} M \big( X_{k-1}^\top C_k X_{k-1} + D_k \big) \\
&\quad + U_\Sigma A_k X_{k-1} M \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}^\top (A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} C_k X_{k-1} \\
&\quad + U_\Sigma A_k X_{k-1} M X_{k-1}^\top C_k \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} \\
&= U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} + U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} A_k X_{k-1}(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma) M \\
&\quad \cdot \Big( X_{k-1}^\top (A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma) C_k X_{k-1}(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma) + D_k \Big)\Big|_{t=0} \\
&= U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} X_k(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0}.
\end{aligned}$$

This completes the proof of eq. (17).

Under the condition that $B_l = b_l I_d$ for some $b_l \in \mathbb{R}$, we can simplify eq. (14) as

$$\begin{aligned}
Y_l &= Y_{l-1} + b_l Y_{l-1} M (X_{l-1}^\top C_l X_{l-1} + D_l) \\
&= Y_{l-1} \big( I + b_l M (X_{l-1}^\top C_l X_{l-1} + D_l) \big) \\
&= Y_0 \prod_{j=1}^{l} \big( I + b_j M (X_{j-1}^\top C_j X_{j-1} + D_j) \big).
\end{aligned}$$

Define $G_l = \overline{X}_0 \prod_{j=1}^{l} \big(I + b_j M(X_{j-1}^\top C_j X_{j-1} + D_j)\big)$, then it satisfies that $Y_l = WG_l$. We are ready to prove that similar results to eqs. (16) and (17) also hold for $G_l$, $l \in [1, L]$:

$$G_l(X_0 \overset{\times}{\leftarrow} U_\Sigma) = U_\Sigma G_l, \tag{18}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} G_l(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} = U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} G_l(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0}. \tag{19}$$

Notice that eq. (18) holds trivially for $l = 0$ as $G_0 = \overline{X}_0$. Now suppose that eq. (18) holds for some $l = k - 1$, we then have

$$\begin{aligned} G_k(X_0 \overset{\times}{\leftarrow} U_\Sigma) &= G_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma)\Big(I + b_k M(X_{k-1}^\top(X_0 \overset{\times}{\leftarrow} U_\Sigma) C_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) + D_k)\Big) \\ &= U_\Sigma G_{k-1}\big(I + b_k M(X_{k-1}^\top C_k X_{k-1} + D_k)\big) = U_\Sigma G_k. \end{aligned}$$

This concludes eq. (18). As for eq. (19), notice that both sides equal 0 when $l \le i$. Now suppose that eq. (19) holds for some $l = k - 1 \ge i$, we then have:

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t} G_k(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\ &= \frac{\mathrm{d}}{\mathrm{d}t} G_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} + \frac{\mathrm{d}}{\mathrm{d}t} b_k G_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR) M \\ &\quad \cdot \Big(X_{k-1}^\top(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR) C_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR) + D_k\Big)\Big|_{t=0} \\ &= \frac{\mathrm{d}}{\mathrm{d}t} G_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\ &\quad + b_k \frac{\mathrm{d}}{\mathrm{d}t} G_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} M\Big(X_{k-1}^\top(X_0 \overset{\times}{\leftarrow} U_\Sigma) C_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) + D_k\Big) \\ &\quad + b_k G_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) M \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}^\top(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} C_k X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) \\ &\quad + b_k G_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) M X_{k-1}^\top(X_0 \overset{\times}{\leftarrow} U_\Sigma) C_k \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\ &= U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} G_{k-1}(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} \\ &\quad + b_k U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} G_{k-1}(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} M\big(X_{k-1}^\top C_k X_{k-1} + D_k\big) \\ &\quad + b_k U_\Sigma G_{k-1} M \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}^\top(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} C_k X_{k-1} \\ &\quad + b_k U_\Sigma G_{k-1} M X_{k-1}^\top C_k \frac{\mathrm{d}}{\mathrm{d}t} X_{k-1}(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} \\ &= U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} G_k(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0}. \end{aligned}$$

This concludes the proof of eq. (19). Consider the in-context risk:

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t} \mathcal{L}(A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\ &= 2\,\mathbb{E}_{X_0, W, U_\perp}\left[\mathrm{tr}\left((I - M) Y_L^\top(X_0 \overset{\times}{\leftarrow} U_\Sigma) \frac{\mathrm{d}}{\mathrm{d}t} Y_L(X_0 \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} (I - M)\right)\right] \\ &= 2\,\mathbb{E}_{X_0, W, U_\perp}\left[\mathrm{tr}\left((I - M) G_L^\top U_\Sigma^\top W^\top W U_\Sigma \frac{\mathrm{d}}{\mathrm{d}t} G_L(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big|_{t=0} (I - M)\right)\right] \\ &= 2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I - M) G_L^\top \Sigma^{-1} \frac{\mathrm{d}}{\mathrm{d}t} \mathbb{E}_{U_\perp}\Big[G_L(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1} R U_\Sigma)\Big]\Big|_{t=0} (I - M)\right)\right] \end{aligned}$$

$$
\begin{aligned}
&= 2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top\Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_L(A_i \overset{+}{\leftarrow} \mathbb{E}_{U_\perp}\left[tU_\Sigma^{-1}RU_\Sigma\right])\Big|_{t=0}(I-M)\right)\right]\\
&= 2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top\Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_L(A_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0}(I-M)\right)\right]\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\,\mathbb{E}_{X_0,W}\left[\mathrm{tr}\Big((I-M)Y_L^\top(A_i \overset{+}{\leftarrow} trI_d)Y_L(A_i \overset{+}{\leftarrow} trI_d)(I-M)\Big)\right]\Big|_{t=0}\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(A_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0},
\end{aligned}
$$

where $r = \mathbb{E}_{U_\perp}[U_\Sigma^{-1}RU_\Sigma] = \frac{1}{d}\,\mathrm{tr}\big(\Sigma^{-1/2}R\Sigma^{1/2}\big)$, and we used the fact that $U_\Sigma^\top\Sigma^{-1}U_\Sigma = \Sigma^{-1}$, and $\frac{\mathrm{d}}{\mathrm{d}t}G_L(A_i \overset{+}{\leftarrow} tR)\Big|_{t=0}$ is affine in $R$. This concludes that eq. (15) holds for $A_i$, $i \in [1, L]$.

**2. Equation (15) holds for $B_i$.**

From the recursive expressions in eq. (14), we can conclude that the values of $X_l$ do not depend on $B_i$. Therefore, we naturally have

$$
X_l(B_i \overset{+}{\leftarrow} tR) = X_l. \tag{20}
$$

Next, we would like to show that for any $l \in [1, L]$,

$$
\mathbb{E}_W\left[W^\top\ \frac{\mathrm{d}}{\mathrm{d}t}Y_l(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}\right] = \Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_l(b_i \overset{+}{\leftarrow} t\,\mathrm{tr}(R))\Big|_{t=0}. \tag{21}
$$

When $l < i$, we can easily verify eq. (21) since both sides equal 0. When $l = i$, we can get

$$
\begin{aligned}
\mathbb{E}_W\left[W^\top\ \frac{\mathrm{d}}{\mathrm{d}t}Y_l(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}\right] &= \mathbb{E}_W\left[W^\top RY_{l-1}M\big(X_{l-1}^\top C_l X_{l-1} + D_l\big)\right]\\
&= \mathbb{E}_W\left[W^\top RW\right]G_{l-1}M\big(X_{l-1}^\top C_l X_{l-1} + D_l\big)\\
&= \mathrm{tr}(R)\Sigma^{-1}G_{l-1}M\big(X_{l-1}^\top C_l X_{l-1} + D_l\big)\\
&= \Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_l(b_i \overset{+}{\leftarrow} t\,\mathrm{tr}(R))\Big|_{t=0}.
\end{aligned}
$$

Suppose that eq. (21) holds for some $l = k - 1 \geq i$. One can then verify

$$
\begin{aligned}
&\mathbb{E}_W\left[W^\top\ \frac{\mathrm{d}}{\mathrm{d}t}Y_k(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}\right]\\
&= \mathbb{E}_W\left[W^\top\ \frac{\mathrm{d}}{\mathrm{d}t}Y_{k-1}(B_i \overset{+}{\leftarrow} tR)\big(I + b_k M(X_{k-1}^\top C_k X_{k-1} + D_k)\big)\Big|_{t=0}\right]\\
&= \mathbb{E}_W\left[W^\top\ \frac{\mathrm{d}}{\mathrm{d}t}Y_{k-1}(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}\right]\big(I + b_k M(X_{k-1}^\top C_k X_{k-1} + D_k)\big)\\
&= \Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_{k-1}(b_i \overset{+}{\leftarrow} t\,\mathrm{tr}(R))\Big|_{t=0}\big(I + b_k M(X_{k-1}^\top C_k X_{k-1} + D_k)\big)\\
&= \Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_k(b_i \overset{+}{\leftarrow} t\,\mathrm{tr}(R))\Big|_{t=0}.
\end{aligned}
$$

The proof of eq. (21) is complete. Now, look at the in-context risk, we have

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0} &= 2\,\mathbb{E}_{X_0,W}\left[\mathrm{tr}\left((I-M)Y_L^\top\ \frac{\mathrm{d}}{\mathrm{d}t}Y_L(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}(I-M)\right)\right]\\
&= 2\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top\,\mathbb{E}_W\left[W^\top\ \frac{\mathrm{d}}{\mathrm{d}t}Y_L(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}\right](I-M)\right)\right]\\
&= 2\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top\Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_L(b_i \overset{+}{\leftarrow} t\,\mathrm{tr}(R))\Big|_{t=0}(I-M)\right)\right]\\
&= 2\,\mathbb{E}_{X_0,W}\left[\mathrm{tr}\left((I-M)Y_L^\top\ \frac{\mathrm{d}}{\mathrm{d}t}Y_L(B_i \overset{+}{\leftarrow} t\,\mathrm{tr}(R)I_d)\Big|_{t=0}(I-M)\right)\right]
\end{aligned}
$$

$$= \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(B_i \overset{+}{\leftarrow} t\,\mathrm{tr}(R)I_d)\Big|_{t=0}.$$

This concludes that eq. (15) holds for $B_i$, $i \in [1, L]$.

**3. Equation (15) holds for $C_i$.**

Similar to the $A_i$ case, we will first prove that for any $l \in [1, L]$,

$$\frac{\mathrm{d}}{\mathrm{d}t}X_l(X_0 \overset{\times}{\leftarrow} U_\Sigma, C_i \overset{+}{\leftarrow} tR)\Big|_{t=0} = U_\Sigma\,\frac{\mathrm{d}}{\mathrm{d}t}X_l(C_i \overset{+}{\leftarrow} tU_\Sigma^\top RU_\Sigma)\Big|_{t=0}. \tag{22}$$

The equation above holds trivially for $l < i$. For the case $l = i$, we have

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}X_l(X_0 \overset{\times}{\leftarrow} U_\Sigma, C_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\
&= A_j X_{l-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) M X_{l-1}^\top(X_0 \overset{\times}{\leftarrow} U_\Sigma) R X_{l-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) \\
&= U_\Sigma A_j X_{l-1} M X_{l-1}^\top U_\Sigma^\top R U_\Sigma X_{l-1} = U_\Sigma\,\frac{\mathrm{d}}{\mathrm{d}t}X_l(C_i \overset{+}{\leftarrow} tU_\Sigma^\top RU_\Sigma)\Big|_{t=0}.
\end{aligned}$$

One can conclude the proof of eq. (22) through a similar reduction as eq. (17) for $l > i$ layers. Next, we establish the corresponding result for $G_l$:

$$\frac{\mathrm{d}}{\mathrm{d}t}G_l(X_0 \overset{\times}{\leftarrow} U_\Sigma, C_i \overset{+}{\leftarrow} tR)\Big|_{t=0} = U_\Sigma\,\frac{\mathrm{d}}{\mathrm{d}t}G_l(C_i \overset{+}{\leftarrow} tU_\Sigma^\top RU_\Sigma)\Big|_{t=0}. \tag{23}$$

This equation holds trivially for $l < i$. When taking $l = i$, we can verify that

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}G_l(X_0 \overset{\times}{\leftarrow} U_\Sigma, C_i \overset{+}{\leftarrow} tR)\Big|_{t=0} &= b_l G_{l-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) M X_{l-1}^\top(X_0 \overset{\times}{\leftarrow} U_\Sigma) R X_{l-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) \\
&= b_l U_\Sigma G_{l-1}(X_0 \overset{\times}{\leftarrow} U_\Sigma) M X_{l-1}^\top U_\Sigma^\top R U_\Sigma X_{l-1} \\
&= U_\Sigma\,\frac{\mathrm{d}}{\mathrm{d}t}G_l(C_i \overset{+}{\leftarrow} tU_\Sigma^\top RU_\Sigma)\Big|_{t=0}.
\end{aligned}$$

For $l > i$ layers, one can follow similar reductions as eq. (19) to finish the proof. We then consider the in-context risk:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(C_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\
&= 2\,\mathbb{E}_{X_0,W,U_\perp}\left[\mathrm{tr}\left((I-M)Y_L^\top(X_0 \overset{\times}{\leftarrow} U_\Sigma)\,\frac{\mathrm{d}}{\mathrm{d}t}Y_L(X_0 \overset{\times}{\leftarrow} U_\Sigma, C_i \overset{+}{\leftarrow} tR)\Big|_{t=0}(I-M)\right)\right] \\
&= 2\,\mathbb{E}_{X_0,W,U_\perp}\left[\mathrm{tr}\left((I-M)G_L^\top U_\Sigma^\top W^\top W U_\Sigma\,\frac{\mathrm{d}}{\mathrm{d}t}G_L(C_i \overset{+}{\leftarrow} tR)\Big|_{t=0}(I-M)\right)\right] \\
&= 2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top\Sigma^{-1}\,\frac{\mathrm{d}}{\mathrm{d}t}\,\mathbb{E}_{U_\perp}\left[G_L(C_i \overset{+}{\leftarrow} tU_\Sigma^\top RU_\Sigma)\right]\Big|_{t=0}(I-M)\right)\right] \\
&= 2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top\Sigma^{-1}\,\frac{\mathrm{d}}{\mathrm{d}t}G_L(C_i \overset{+}{\leftarrow} tr\Sigma^{-1})\Big|_{t=0}(I-M)\right)\right] \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\,\mathbb{E}_{X_0,W}\left[\mathrm{tr}\Big((I-M)Y_L^\top(C_i \overset{+}{\leftarrow} tr\Sigma^{-1})Y_L(C_i \overset{+}{\leftarrow} tr\Sigma^{-1})(I-M)\Big)\right]\Big|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(C_i \overset{+}{\leftarrow} tr\Sigma^{-1})\Big|_{t=0},
\end{aligned}$$

where $r = \mathbb{E}_{U_\perp}[U_\Sigma^\top RU_\Sigma] = \frac{1}{d}\,\mathrm{tr}\big(\Sigma^{1/2}R\Sigma^{1/2}\big)$. This concludes that eq. (15) holds for $C_i$.

**4. Equation (15) holds for $D_i$.**

Let $U_p \in \mathbb{R}^{n\times n}$ be a uniformly sampled permutation matrix, i.e., a binary matrix that has exactly one $1$ entry in each row and column with all other entries $0$. Let $U_\circ = \mathrm{diag}(U_p \otimes I_2, I_2) \in \mathbb{R}^{(2n+2)\times(2n+2)}$.

One can verify that by multiplying $X_0 U_\circ$, it is equal to shuffling the first $n$ 2-column sub-blocks of $X_0$ and keeping the last 2 columns unchanged.

Then, consider a matrix $U_\xi = \mathrm{diag}(\xi_1, \ldots, \xi_{n+1}) \in \mathbb{R}^{(n+1)\times(n+1)}$ where $\xi_i \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}\{\pm 1\}$, i.e., a diagonal matrix with random $\pm 1$ entries. Let $U_\pm = U_\xi \otimes I_2 \in \mathbb{R}^{(2n+2)\times(2n+2)}$. Thus, $U_\pm = U_\pm^\top$ and $X_0 U_\pm$ is randomly flipping the sign of each 2-column sub-block in $X_0$.

We are going to prove that for any $l \in [1, L]$, recalling that $f(A \overset{\diamond}{\leftarrow} B) = f(A \leftarrow AB)$,

$$X_l(X_0 \overset{\diamond}{\leftarrow} U_\pm U_\circ) = X_l U_\pm U_\circ, \tag{24}$$

$$G_l(X_0 \overset{\diamond}{\leftarrow} U_\pm U_\circ) = G_l U_\pm U_\circ. \tag{25}$$

Equation (24) holds trivially for $l = 0$. When eq. (24) holds for some $l = k - 1$, we can verify that

$$\begin{aligned}
&X_k(X_0 \overset{\diamond}{\leftarrow} U_\pm U_\circ) \\
&= X_{k-1} U_\pm U_\circ + A_k X_{k-1} U_\pm U_\circ M\big(U_\circ^\top U_\pm^\top X_{k-1}^\top C_k X_{k-1} U_\pm U_\circ + D_k\big) \\
&= X_{k-1} U_\pm U_\circ + A_k X_{k-1} U_\pm U_\circ M U_\circ^\top U_\pm^\top \big(X_{k-1}^\top C_k X_{k-1} + U_\pm U_\circ D_k U_\circ^\top U_\pm^\top\big) U_\pm U_\circ \\
&= X_{k-1} U_\pm U_\circ + A_k X_{k-1} M\big(X_{k-1}^\top C_k X_{k-1} + D_k\big) U_\pm U_\circ \\
&= \big(X_{k-1} + A_k X_{k-1} M\big(X_{k-1}^\top C_k X_{k-1} + D_k\big)\big) U_\pm U_\circ = X_k U_\pm U_\circ.
\end{aligned}$$

It uses the fact that there exists some $D_i^1, D_i^2 \in \mathbb{R}^{2\times 2}$ such that $D_i = \mathrm{diag}(I_n \otimes D_i^1, D_i^2)$, so shuffling the first $n$ $2 \times 2$ diagonal sub-blocks of $D_i$ does not change the matrix, and we have $U_\circ D_i U_\circ^\top = D_i$. Similarly, we have $U_\pm D_k U_\pm^\top = D_k$. This concludes eq. (24), and eq. (25) could be acquired similarly.

Next, we will establish the following equalities for $X_l$ and $G_l$:

$$\frac{\mathrm{d}}{\mathrm{d}t} X_l(X_0 \overset{\diamond}{\leftarrow} U_\pm U_\circ, D_i \overset{+}{\leftarrow} tR)\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t} X_l(D_i \overset{+}{\leftarrow} tU_\pm U_\circ R U_\circ^\top U_\pm^\top)\Big|_{t=0} U_\pm U_\circ, \tag{26}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} G_l(X_0 \overset{\diamond}{\leftarrow} U_\pm U_\circ, D_i \overset{+}{\leftarrow} tR)\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t} G_l(D_i \overset{+}{\leftarrow} tU_\pm U_\circ R U_\circ^\top U_\pm^\top)\Big|_{t=0} U_\pm U_\circ. \tag{27}$$

The proof follows by similar reductions as proving eqs. (17) and (19).

Finally, we consider the in-context risk under the permutation of $U_p$ and $U_\xi$. Since each pair of $(x_i, y_i)$ is equivalently sampled from Gaussian distributions, we have $X_0 \overset{d}{=} X_0 U_\pm U_\circ$. Therefore,

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(D_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\
&= 2\,\mathbb{E}_{X_0,W}\left[\mathrm{tr}\left((I-M)Y_L^\top \frac{\mathrm{d}}{\mathrm{d}t} Y_L(D_i \overset{+}{\leftarrow} tR)\Big|_{t=0} (I-M)\right)\right] \\
&= 2\,\mathbb{E}_{X_0,W,U_p,U_\xi}\left[\mathrm{tr}\left((I-M)Y_L^\top(X_0 \overset{\diamond}{\leftarrow} U_\pm U_\circ) \frac{\mathrm{d}}{\mathrm{d}t} Y_L(X_0 \overset{\diamond}{\leftarrow} U_\pm U_\circ, D_i \overset{+}{\leftarrow} tR)\Big|_{t=0} (I-M)\right)\right] \\
&= 2d\,\mathbb{E}_{X_0,U_p,U_\xi}\left[\mathrm{tr}\left((I-M)U_\circ^\top U_\pm^\top G_L^\top \Sigma^{-1} \frac{\mathrm{d}}{\mathrm{d}t} G_L(D_i \overset{+}{\leftarrow} tU_\pm U_\circ R U_\circ^\top U_\pm^\top)\Big|_{t=0} U_\pm U_\circ (I-M)\right)\right] \\
&= 2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top \Sigma^{-1} \frac{\mathrm{d}}{\mathrm{d}t}\,\mathbb{E}_{U_p,U_\xi}\Big[G_L(D_i \overset{+}{\leftarrow} tU_\pm U_\circ^\top R U_\circ U_\pm)\Big]\Big|_{t=0} (I-M)\right)\right] \\
&= 2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top \Sigma^{-1} \frac{\mathrm{d}}{\mathrm{d}t} G_L(D_i \overset{+}{\leftarrow} t\widetilde{R})\Big|_{t=0} (I-M)\right)\right] = \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(D_i \overset{+}{\leftarrow} t\widetilde{R})\Big|_{t=0},
\end{aligned}$$

where $\widetilde{R} = \mathbb{E}_{U_p,U_\xi}[U_\pm U_\circ^\top R U_\circ U_\pm] = \mathrm{diag}(I_n \otimes R^1, R^2)$, $R^1 = \frac{1}{n}\sum_{j=1}^n R_j$, $R^2 = R_{n+1}$, and $R_j$ is the $j$-th $2 \times 2$ diagonal block of $R$. The 4th equality uses the fact that $\mathrm{tr}[(I-M)A(I-M)]$ is extracting the right-bottom element of $A$, so it should be equal to $\mathrm{tr}\big[(I-M)U_\circ^\top U_\pm^\top A U_\pm U_\circ (I-M)\big]$ for any matrix $A$. This concludes that eq. (15) holds for $D_i$.

Till now, we have proved that eq. (15) holds for each one of $A_i, B_i, C_i, D_i$. The proof of the whole theorem is then completed by applying Lemma 8. □

### B.3 Proof of Theorem 2

*Proof.* In this proof, we follow the same notations as the proof of Theorem 1, where the constant $\frac{1}{n}$ factor is dropped and $\widetilde{Z}_0$, $\widetilde{X}_0$, $\widetilde{Y}_0$ are simplified as $Z_0$, $X_0$, $Y_0$ respectively.

$$Z_0 = \begin{bmatrix} x_1 & 0 & 0 & \cdots & x_n & 0 & 0 & x_{\text{test}} & 0 & 0 \\ 0 & 0 & y_1 & \cdots & 0 & 0 & y_n & 0 & 0 & y_{\text{test}} \end{bmatrix} \in \mathbb{R}^{(2d)\times(3n+3)}. \tag{28}$$

Let $Z_l \in \mathbb{R}^{2d\times(3n+3)}$ be the $l$-th layer's output and let $X_l, Y_l \in \mathbb{R}^{d\times(3n+3)}$ be its first and last $d$ rows. Our goal is to prove that, for any $E \in A \cup B \cup C \cup D$ and an arbitrary matrix $R \in \mathbb{R}^{d\times d}$ ($\mathbb{R}^{d_p\times d_p} for D$), there exists $\widetilde{R} \in \mathcal{S}_I$ ($\mathcal{S}_\Sigma$ for C, $\mathcal{S}_P$ for D) such that

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\mathcal{L}(E \xleftarrow{+} t\widetilde{R})\Big|_{t=0} \le \frac{\mathrm{d}}{\mathrm{d}t}\,\mathcal{L}(E \xleftarrow{+} tR)\Big|_{t=0}. \tag{29}$$

The proofs of eq. (29) for $A_i$, $B_i$ and $C_i$ are identical with the proof of Theorem 1 so we omit them. We will be focusing on $D_i$ for the rest of the proof.

Let $U_p^s \in \mathbb{R}^{n\times n}$ and $U_p^t \in \mathbb{R}^{(n+1)\times(n+1)}$ be uniformly sampled permutation matrices. Let $U_\circ^s = \mathrm{diag}(U_p^s, 1) \otimes \mathrm{diag}(1,0,1)$ and $U_\circ^t = U_p^t \otimes \mathrm{diag}(0,1,0)$. Therefore, $X_0U_\circ^s$ is shuffling the 1-st and 3-rd columns among each 3-column sub-block of $X_0$ (except for the last 3-column sub-block), and $X_0U_\circ^s$ is shuffling the 2-nd column among each 3-column sub-block. Next, let $U_\xi^s, U_\xi^t \in \mathbb{R}^{(n+1)\times(n+1)}$ be diagonal matrices with uniformly sampled $\pm1$ entries. Define $U_\pm^s = U_\xi^s \otimes \mathrm{diag}(1,0,1)$ and $U_\pm^t = U_\xi^t \otimes \mathrm{diag}(0,1,0)$. It can then be verified that $X_0U_\pm^sU_\pm^t \overset{d}{=} X_0$.

To simplify the notations, let $U_\equiv$ denote $U_\pm^sU_\pm^tU_\circ^sU_\circ^t$. We will focus on a subset of $\mathcal{S}_P$:

$$\mathcal{S}_P' = \Big\{\mathrm{diag}(I_n \otimes \Lambda_1, \Lambda_2) + I_{n+1} \otimes \Lambda_3 \;\Big|\; \Lambda_1, \Lambda_2 \in \mathcal{M}\left(\begin{smallmatrix}1&0&1\\0&0&0\\1&0&1\end{smallmatrix}\right), \Lambda_3 \in \mathcal{M}\left(\begin{smallmatrix}0&0&0\\0&1&0\\0&0&0\end{smallmatrix}\right)\Big\}.$$

Assume $D_k = \mathrm{diag}(I_n \otimes \Lambda_1, \Lambda_2) + I_{n+1} \otimes \Lambda_3 \in \mathcal{S}_P'$ as defined above, one can verify that it is a block-diagonal matrix constructed from the same $3\times3$ sub-blocks, and thus is invariant under $U_\equiv D_k U_\equiv^\top$. We will then prove that for any $l \in [1, L]$,

$$X_l(X_0 \xleftarrow{\diamond} U_\equiv) = X_lU_\equiv, \tag{30}$$

$$G_l(X_0 \xleftarrow{\diamond} U_\equiv) = G_lU_\equiv, \tag{31}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}X_l(X_0 \xleftarrow{\diamond} U_\equiv, D_i \xleftarrow{+} tR)\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}X_l(D_i \xleftarrow{+} tU_\equiv RU_\equiv^\top)\Big|_{t=0} U_\equiv, \tag{32}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}G_l(X_0 \xleftarrow{\diamond} U_\equiv, D_i \xleftarrow{+} tR)\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}G_l(D_i \xleftarrow{+} tU_\equiv RU_\equiv^\top)\Big|_{t=0} U_\equiv. \tag{33}$$

These results can be acquired by similar proofs as eqs. (24) to (27). We then consider the in-context risk under the permutations of $U_\equiv$. Similarly, we have $X_0 \overset{d}{=} X_0U_\equiv$ and

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(D_i \xleftarrow{+} tR)\Big|_{t=0} \\
&= 2\,\mathbb{E}_{X_0,W}\left[\mathrm{tr}\left((I-M)Y_L^\top \frac{\mathrm{d}}{\mathrm{d}t}Y_L(D_i \xleftarrow{+} tR)\Big|_{t=0}(I-M)\right)\right] \\
&= 2d\,\mathbb{E}_{X_0,U_\equiv}\left[\mathrm{tr}\left((I-M)G_L^\top(X_0 \xleftarrow{\diamond} U_\equiv)\Sigma^{-1}\frac{\mathrm{d}}{\mathrm{d}t}G_L(X_0 \xleftarrow{\diamond} U_\equiv, D_i \xleftarrow{+} tR)\Big|_{t=0}(I-M)\right)\right] \\
&= 2d\,\mathbb{E}_{X_0,U_\equiv}\left[\mathrm{tr}\left((I-M)U_\equiv^\top G_L^\top\Sigma^{-1}\frac{\mathrm{d}}{\mathrm{d}t}G_L(D_i \xleftarrow{+} tU_\equiv RU_\equiv^\top)\Big|_{t=0}U_\equiv(I-M)\right)\right] \\
&= 2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top\Sigma^{-1}\frac{\mathrm{d}}{\mathrm{d}t}G_L(D_i \xleftarrow{+} t\,\mathbb{E}_{U_\equiv}[U_\equiv RU_\equiv^\top])\Big|_{t=0}(I-M)\right)\right] \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(D_i \xleftarrow{+} t\widetilde{R})\Big|_{t=0}.
\end{aligned}$$

Let $R_j$ be the $j$-th $3\times3$ diagonal block of $R$, then $R^1 = \frac{1}{n}\sum_{j=1}^{n} R_j \circ \left(\begin{smallmatrix}1&0&1\\0&0&0\\1&0&1\end{smallmatrix}\right)$, $R^2 = R_{n+1} \circ \left(\begin{smallmatrix}1&0&1\\0&0&0\\1&0&1\end{smallmatrix}\right)$, $R^3 = \frac{1}{n+1}\sum_{j=1}^{n+1} R_j \circ \left(\begin{smallmatrix}0&0&0\\0&1&0\\0&0&0\end{smallmatrix}\right)$ and $\widetilde{R} = \mathbb{E}_{U_{\equiv}}\left[U_{\equiv} R U_{\equiv}^{\top}\right] = \operatorname{diag}(I_n \otimes R^1, R^2) + I_{n+1} \otimes R^3$. This indicates that eq. (29) holds for each $D_i \in \mathcal{S}_P'$, and thus the proof of the whole theorem completes by applying Lemma 8 and noticing that $\mathcal{S}_P' \subset \mathcal{S}_P$. □

### B.4 Proof of Theorem 5

*Proof.* We keep the same notations as the proof of Theorem 1, dropping the $\frac{1}{n}$ factor and simplifying $\widetilde{X}_0$, $\widetilde{Y}_0$, $\widetilde{Z}_0$ as $X_0$, $Y_0$, $Z_0$, as follows:

$$Z_0 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ x_1 & y_1 & \cdots & x_n & y_n & x_{\text{test}} & y_{\text{test}} \end{bmatrix} \in \mathbb{R}^{(2d)\times(2n+2)}. \tag{34}$$

Note that we now have $X_0$ and $Y_0$ containing both $x_i$ and $y_i$. Define

$$\begin{aligned} X &= [x_1 \quad 0 \quad \cdots \quad x_n \quad 0 \quad x_{\text{test}} \quad 0], \\ \overline{X} &= [0 \quad x_1 \quad \cdots \quad 0 \quad x_n \quad 0 \quad x_{\text{test}}], \\ Y &= [0 \quad y_1 \quad \cdots \quad 0 \quad y_n \quad 0 \quad y_{\text{test}}]. \end{aligned}$$

we then have $Y_0 = X + Y = X + W\overline{X}$. From the parameter configuration in eq. (12), the update rule of the first attention layer is

$$X_1 = A_1 Y_0 M D_1 = A_1 X M D_1, \quad Y_1 = Y_0 = X + W\overline{X}. \tag{35}$$

The update rule for the following layers is the same as eq. (14). We are going to prove that, for any $E \in A \cup B \cup C \cup D$ and an arbitrary matrix $R \in \mathbb{R}^{d\times d}$ ($\mathbb{R}^{d_p\times d_p}$ for $D$), there exists $\widetilde{R} \in \mathcal{S}_I$ ($\mathcal{S}_\Sigma$ for $C$, $\mathcal{S}_P$ for $D$) such that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(E \overset{+}{\leftarrow} t\widetilde{R})\Big|_{t=0} \le \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(E \overset{+}{\leftarrow} tR)\Big|_{t=0}. \tag{36}$$

Similarly to Theorem 1, we uniformly sample $U_\perp \in \mathbb{R}^{d\times d}$ as an orthonormal random matrix, and let $U_\Sigma = \Sigma^{1/2} U_\perp \Sigma^{-1/2}$. Under the condition that $B_l = b_l I_d$ for some $b_l \in \mathbb{R}$, we have

$$Y_l = Y_1 \prod_{j=2}^{l} \big(I + b_j M\big(X_{j-1}^{\top} C_j X_{j-1} + D_j\big)\big).$$

Let $F_l = X\prod_{j=2}^{l}\big(I + b_j M\big(X_{j-1}^{\top} C_j X_{j-1} + D_j\big)\big)$, $G_l = \overline{X}\prod_{j=2}^{l}\big(I + b_j M\big(X_{j-1}^{\top} C_j X_{j-1} + D_j\big)\big)$, we then have $Y_l = F_l + WG_l$. According to Lemma 9,

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(E \overset{+}{\leftarrow} tR)\Big|_{t=0} \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\,\mathbb{E}_{X_0,W}\Big[\operatorname{tr}\Big((I-M)Y_L^{\top}(E \overset{+}{\leftarrow} tR)Y_L(E \overset{+}{\leftarrow} tR)(I-M)\Big)\Big]\Big|_{t=0} \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\,\mathbb{E}_{X_0,W}\Big[\operatorname{tr}\Big((I-M)F_L^{\top}(E \overset{+}{\leftarrow} tR)F_L(E \overset{+}{\leftarrow} tR)(I-M)\Big)\Big]\Big|_{t=0} \\ &\quad + \frac{\mathrm{d}}{\mathrm{d}t}\,\mathbb{E}_{X_0,W}\Big[\operatorname{tr}\Big((I-M)G_L^{\top}(E \overset{+}{\leftarrow} tR)W^{\top}WG_L(E \overset{+}{\leftarrow} tR)(I-M)\Big)\Big]\Big|_{t=0} \\ &= 2\,\mathbb{E}_{X_0}\left[\operatorname{tr}\left((I-M)F_L^{\top}\,\frac{\mathrm{d}}{\mathrm{d}t}F_L(E \overset{+}{\leftarrow} tR)\Big|_{t=0}(I-M)\right)\right] \\ &\quad + 2d\,\mathbb{E}_{X_0}\left[\operatorname{tr}\left((I-M)G_L^{\top}\Sigma^{-1}\,\frac{\mathrm{d}}{\mathrm{d}t}G_L(E \overset{+}{\leftarrow} tR)\Big|_{t=0}(I-M)\right)\right]. \end{aligned}$$

Next, we will show that eq. (36) holds for each one of $A_i, B_i, C_i, D_i$ for any $i \in [1, L]$.

**1. Equation (36) holds for $A_i$.**

One can easily verify that eqs. (16) and (17) still hold. Furthermore, eqs. (18) and (19) hold for both $F_l$ and $G_l$. With these observations, we can then verify

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(A_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\
&= 2\,\mathbb{E}_{X_0,U_\perp}\left[\mathrm{tr}\left((I-M)F_L^\top(X \overset{\times}{\leftarrow} U_\Sigma)\ \frac{\mathrm{d}}{\mathrm{d}t}F_L(X \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0}(I-M)\right)\right] \\
&\quad + 2d\,\mathbb{E}_{X_0,U_\perp}\left[\mathrm{tr}\left((I-M)G_L^\top(X \overset{\times}{\leftarrow} U_\Sigma)\Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_L(X \overset{\times}{\leftarrow} U_\Sigma, A_i \overset{+}{\leftarrow} tR)\Big|_{t=0}(I-M)\right)\right] \\
&= 2\,\mathbb{E}_{X_0,U_\perp}\left[\mathrm{tr}\left((I-M)F_L^\top U_\Sigma^\top U_\Sigma\ \frac{\mathrm{d}}{\mathrm{d}t}F_L(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1}RU_\Sigma)\Big|_{t=0}(I-M)\right)\right] \\
&\quad + 2d\,\mathbb{E}_{X_0,U_\perp}\left[\mathrm{tr}\left((I-M)G_L^\top U_\Sigma^\top\Sigma^{-1}U_\Sigma\ \frac{\mathrm{d}}{\mathrm{d}t}G_L(A_i \overset{+}{\leftarrow} tU_\Sigma^{-1}RU_\Sigma)\Big|_{t=0}(I-M)\right)\right] \\
&= 2\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)F_L^\top\ \frac{\mathrm{d}}{\mathrm{d}t}F_L(A_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0}(I-M)\right)\right] \\
&\quad + 2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_L^\top\Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_L(A_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0}(I-M)\right)\right] \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(A_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0},
\end{aligned}
$$

where $r = \mathbb{E}_{U_\perp}[U_\Sigma^{-1}RU_\Sigma] = \frac{1}{d}\,\mathrm{tr}\big(\Sigma^{-1/2}R\Sigma^{1/2}\big)$.

**2. Equation (36) holds for $B_i$.**

From the definition of $F_l$ and $G_l$, we can verify that

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}Y_l(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0} \\
&= R(F_{i-1}+WG_{i-1})M(X_{i-1}^\top C_i X_{i-1}+D_i)\prod_{j=i+1}^{l}\big(I+b_jM(X_{j-1}^\top C_jX_{j-1}+D_j)\big).
\end{aligned}
$$

Define

$$
\begin{aligned}
\overline{F}_l^i &= \big(F_{i-1}+B_iF_{i-1}M(X_{i-1}^\top C_iX_{i-1}+D_i)\big)\prod_{j=i+1}^{l}\big(I+b_jM(X_{j-1}^\top C_jX_{j-1}+D_j)\big), \\
\overline{G}_l^i &= \big(WG_{i-1}+B_iWG_{i-1}M(X_{i-1}^\top C_iX_{i-1}+D_i)\big)\prod_{j=i+1}^{l}\big(I+b_jM(X_{j-1}^\top C_jX_{j-1}+D_j)\big),
\end{aligned}
$$

We then have

$$
\frac{\mathrm{d}}{\mathrm{d}t}Y_l(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}\overline{F}_l^i(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0} + \frac{\mathrm{d}}{\mathrm{d}t}\overline{G}_l^i(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}.
$$

Similar to eqs. (19) and (21), we can prove that

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\overline{F}_l^i(X_0 \overset{\times}{\leftarrow} U_\Sigma, B_i \overset{+}{\leftarrow} tR)\Big|_{t=0} &= U_\Sigma\ \frac{\mathrm{d}}{\mathrm{d}t}\overline{F}_l^i(B_i \overset{+}{\leftarrow} tU_\Sigma^{-1}RU_\Sigma)\Big|_{t=0}, \\
\mathbb{E}_W\left[W^\top\ \frac{\mathrm{d}}{\mathrm{d}t}\overline{G}_l^i(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}\right] &= \Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}\overline{G}_l^i(B_i \overset{+}{\leftarrow} t\,\mathrm{tr}(R)I_d)\Big|_{t=0}.
\end{aligned}
$$

Without loss of generality, we assume that $r = \frac{1}{d}\,\mathrm{tr}\big(\Sigma^{-1/2}R\Sigma^{1/2}\big) \le \frac{1}{d}\,\mathrm{tr}(R)$, and let $\gamma = rd/\,\mathrm{tr}(R) \le 1$. Then, one can verify that

$$
\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}
$$

$$
\begin{aligned}
&= 2\,\mathbb{E}_{X_0,U_\perp}\left[\mathrm{tr}\left((I-M)F_l^\top(X \overset{\times}{\leftarrow} U_\Sigma)\ \frac{\mathrm{d}}{\mathrm{d}t}\overline{F}_l^i(X \overset{\times}{\leftarrow} U_\Sigma, B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}(I-M)\right)\right] \\
&\quad + 2\,\mathbb{E}_{X_0,W}\left[\mathrm{tr}\left((I-M)G_l^\top W^\top\ \frac{\mathrm{d}}{\mathrm{d}t}\overline{G}_l^i(B_i \overset{+}{\leftarrow} tR)\Big|_{t=0}(I-M)\right)\right] \\
&= 2\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)F_l^\top\ \frac{\mathrm{d}}{\mathrm{d}t}\overline{F}_l^i(B_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0}(I-M)\right)\right] \\
&\quad + 2\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_l^\top\Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}\overline{G}_l^i(B_i \overset{+}{\leftarrow} t\,\mathrm{tr}(R)I_d)\Big|_{t=0}(I-M)\right)\right] \\
&= 2\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)F_l^\top\ \frac{\mathrm{d}}{\mathrm{d}t}F_l(B_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0}(I-M)\right)\right] \\
&\quad + \frac{1}{\gamma}2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_l^\top\Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_l(B_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0}(I-M)\right)\right] \\
&= \left(\frac{1}{\gamma}-1\right)2d\,\mathbb{E}_{X_0}\left[\mathrm{tr}\left((I-M)G_l^\top\Sigma^{-1}\ \frac{\mathrm{d}}{\mathrm{d}t}G_l(B_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0}(I-M)\right)\right] \\
&\quad + \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(B_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0} \geq \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(B_i \overset{+}{\leftarrow} trI_d)\Big|_{t=0}.
\end{aligned}
$$

The last inequality assumes the positivity of the term involving $G_l$. Otherwise, one can simply flip the numerator and denominator of $\gamma$ and scale the derivative of $F_l$ instead of $G_l$ to yield an additional positive term besides the risk term to finish the proof.

**3. Equation (36) holds for $C_i$, $D_i$.**

Similarly, one can verify that eqs. (22) and (23) still hold (also eqs. (24) to (27)), and finish the proof by following the same reductions as Theorem 1 with $F_l$ and $G_l$. □

### B.5 Proof of Proposition 3

*Proof.* Let $A_l = a_l I_d$, $B_l = b_l I_d$, $C_l = c_l I_d$ and $D_l = \mathrm{diag}(I_n \otimes D_l^1, D_l^2) + I_{n+1} \otimes D_l^3 + D_l^4 \otimes D_l^5$ for $l \in [1, 2]$. Let $Z_l \in \mathbb{R}^{2d\times(3n+3)}$ be the output of the $l$-th attention layer, and let $X_l, Y_l \in \mathbb{R}^{d\times(3n+3)}$ be its first and last $d$ rows respectively. Note that $Y_l$ in this proof does not contain $y_{\mathrm{test}}$.

Let $D_1^1 = \begin{pmatrix} d_x^x & 0 & d_x^y \\ 0 & 0 & 0 \\ d_y^x & 0 & d_y^y \end{pmatrix}$, $D_1^2 = \begin{pmatrix} s_x & 0 & s_y \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ (note that the last row of $D_1^2$ is masked out by $M$, so we simply set it to 0), and $D_1^5 = \begin{pmatrix} 0 & t_x & 0 \\ 0 & 0 & 0 \\ 0 & t_y & 0 \end{pmatrix}$. We use $D$ as an abbreviation for $D_1^4$, and use $d_{i,j}$ to denote the elements in $D$. One can verify that

$$
\begin{aligned}
X_1 &= X_0 + a_1 X_0 M\big(\mathrm{diag}(I_n \otimes D_1^1, D_1^2) + I_{n+1} \otimes D_1^3 + D_1^4 \otimes D_1^5\big) \\
&= \begin{bmatrix} (1+a_1 d_x^x)x_1 & a_1 t_x \sum_{i=1}^{n+1} d_{i,1} x_i & a_1 d_x^y x_1 \\ & \cdots & \\ (1+a_1 d_x^x)x_n & a_1 t_x \sum_{i=1}^{n+1} d_{i,n} x_i & a_1 d_x^y x_n \\ (1+a_1 d_x^x)x_{\mathrm{test}} & a_1 t_x \sum_{i=1}^{n+1} d_{i,n+1} x_i & a_1 d_x^y x_{\mathrm{test}} \end{bmatrix}.
\end{aligned}
$$

Similarly, we have

$$
\begin{aligned}
Y_1 &= Y_0 + b_1 Y_0 M\big(\mathrm{diag}(I_n \otimes D_1^1, D_1^2) + I_{n+1} \otimes D_1^3 + D_1^4 \otimes D_1^5\big) \\
&= \begin{bmatrix} b_1 d_y^x y_1 & b_1 t_y \sum_{i=1}^{n} d_{i,1} y_i & (1+b_1 d_y^y)y_1 \\ & \cdots & \\ b_1 d_y^x y_n & b_1 t_y \sum_{i=1}^{n} d_{i,n} y_i & (1+b_1 d_y^y)y_n \\ 0 & b_1 t_y \sum_{i=1}^{n} d_{i,n+1} y_i & 0 \end{bmatrix}.
\end{aligned}
$$

By the definition of linear attention, we can show that

$$
\begin{aligned}
\mathsf{TF}(Z_0; \{V_l, Q_l\}_{l=1}^2) = (Y_2)_{3n+3} &= b_2 Y_1 M\big(c_2 X_1^\top (X_1)_{3n+3} + (D_2)_{3n+3}\big) \\
&= b_2 c_2 a_1 d_x^y \left(\sum_{i=1}^{3n+2} (Y_1)_i (X_1)_i^\top\right) x_{\mathrm{test}}.
\end{aligned}
$$

Define $\Delta X_1 = [0 \quad a_1 t_x d_{n+1,1} x_{\text{test}} \quad 0 \quad \cdots \quad 0 \quad a_1 t_x d_{n+1,n+1} x_{\text{test}} \quad 0]$, and let $\overline{X}_1 = X_1 - \Delta X_1$, then $\mathsf{TF}(Z_0; \{V_l, Q_l\}_{l=1}^2) = \mathsf{TF}(Z_0; \{V_l, Q_l\}_{l=1}^2, X_1 \leftarrow \overline{X}_1) + \mathsf{TF}(Z_0; \{V_l, Q_l\}_{l=1}^2, X_1 \leftarrow \Delta X_1)$. Let $b_1 d_y^x (1 + a_1 d_x^x) + (1 + b_1 d_y^x) a_1 d_x^x = a$, $b_1 t_y a_1 t_x = b$, $b_2 c_2 a_1 d_x^y = c$, we then have

$$\begin{aligned}
\mathsf{TF}(Z_0; \{V_l, Q_l\}_{l=1}^2, X_1 \leftarrow \overline{X}_1) &= c\left(a \sum_{i=1}^n y_i x_i^\top + b \sum_{i=1}^{n+1} \left(\sum_{j=1}^n d_{j,i} y_j\right)\left(\sum_{j=1}^n d_{j,i} x_j^\top\right)\right) x_{\text{test}} \\
&= c\left(a \sum_{i=1}^n y_i x_i^\top + b \sum_{j=1}^n \sum_{k=1}^n \left(\sum_{i=1}^{n+1} d_{j,i} d_{k,i}\right) y_j x_k^\top\right) x_{\text{test}}, \quad (37) \\
\mathsf{TF}(Z_0; \{V_l, Q_l\}_{l=1}^2, X_1 \leftarrow \Delta X_1) &= bc \sum_{i=1}^{n+1} \sum_{j=1}^n d_{j,i} y_j d_{n+1,i} x_{\text{test}}^\top x_{\text{test}} \\
&= bc \sum_{j=1}^n \left(\sum_{i=1}^{n+1} d_{j,i} d_{n+1,i}\right) y_j x_{\text{test}}^\top x_{\text{test}}. \quad (38)
\end{aligned}$$

Now consider the in-context risk,

$$\begin{aligned}
&\mathcal{L}(V, Q) = \mathbb{E}_{Z_0, W} \left\| \mathsf{TF}(Z_0; \{V, Q\}) + W x_{\text{test}} \right\|_2^2 \\
&= \mathbb{E}_{Z_0, W} \Big[ \big(\mathsf{TF}(Z_0; \{V, Q\}) + W x_{\text{test}}\big)^\top \big(\mathsf{TF}(Z_0; \{V, Q\}) + W x_{\text{test}}\big) \Big] \\
&= \mathbb{E}_{Z_0, W} \Big[ \big(\mathsf{TF}(Z_0; \{V, Q\}, X_1 \leftarrow \overline{X}_1) + W x_{\text{test}}\big)^\top \big(\mathsf{TF}(Z_0; \{V, Q\}, X_1 \leftarrow \overline{X}_1) + W x_{\text{test}}\big) \Big] \\
&\quad + 2\, \mathbb{E}_{Z_0, W} \big[ \mathsf{TF}(Z_0; \{V, Q\}, X_1 \leftarrow \Delta X_1)^\top \big(\mathsf{TF}(Z_0; \{V, Q\}, X_1 \leftarrow \overline{X}_1) + W x_{\text{test}}\big) \big] \\
&\quad + \mathbb{E}_{Z_0, W} \big[ \mathsf{TF}(Z_0; \{V, Q\}, X_1 \leftarrow \Delta X_1)^\top \mathsf{TF}(Z_0; \{V, Q\}, X_1 \leftarrow \Delta X_1) \big].
\end{aligned}$$

In the equation above, the 3-rd part is always positive. We then examine the second part:

$$\begin{aligned}
&\mathbb{E}_{Z_0, W} \big[ \mathsf{TF}(Z_0; \{V, Q\}, X_1 \leftarrow \Delta X_1)^\top \big(\mathsf{TF}(Z_0; \{V, Q\}, X_1 \leftarrow \overline{X}_1) + W x_{\text{test}}\big) \big] \\
&= \mathbb{E}_{Z_0, W} \big[ x_{\text{test}}^\top x_{\text{test}} v_1 x_{\text{test}} + x_{\text{test}}^\top x_{\text{test}} v_2 x_{\text{test}} \big] = 0,
\end{aligned}$$

where $v_1 = bc \sum_{j=1}^n \Big(\sum_{i=1}^{n+1} d_{j,i} d_{n+1,i}\Big) y_j^\top c \Big(a \sum_{i=1}^n y_i x_i^\top + b \sum_{j=1}^n \sum_{k=1}^n \Big(\sum_{i=1}^{n+1} d_{j,i} d_{k,i}\Big) y_j x_k^\top\Big)$ and $v_2 = bc \sum_{j=1}^n \Big(\sum_{i=1}^{n+1} d_{j,i} d_{n+1,i}\Big) y_j^\top W$ are independent of $x_{\text{test}}$. Therefore, $\mathcal{L}(V, Q)$ attains its minimum only if $\mathsf{TF}(Z_0; \{V, Q\}, X_1 \leftarrow \Delta X_1) = 0$, implying $d_{n+1,i} = 0$ for $i \in [1, n+1]$.

In the following analysis, we will assume that the last row of $D$ is 0, and let $M \in \mathbb{R}^{n \times (n+1)}$ be the first $n$ rows of $D$. Additionally, we will drop the $c$ factor in eq. (37), since its position could be substituted by $a$ and $b$. We then define $\widetilde{W} = a \sum_{i=1}^n y_i x_i^\top + b \sum_{j=1}^n \sum_{k=1}^n \Big(\sum_{i=1}^{n+1} d_{j,i} d_{k,i}\Big) y_j x_k^\top$, $X = [x_1 \quad \cdots \quad x_n]$ and $Y = [y_1 \quad \cdots \quad y_n]$. One can verify that

$$\widetilde{W} = aYX^\top + bYMM^\top X^\top = aWXX^\top + bWXMM^\top X^\top. \quad (39)$$

Furthermore, the in-context risk could be expanded as

$$\begin{aligned}
\mathcal{L}(V, Q) &= \mathbb{E}_{Z_0, W} \left\| \widetilde{W} x_{\text{test}} + W x_{\text{test}} \right\|_2^2 = \mathbb{E}_{Z_0, W} \Big[ x_{\text{test}}^\top (\widetilde{W} + W)^\top (\widetilde{W} + W) x_{\text{test}} \Big] \\
&= \mathbb{E}_{Z_0, W} \Big[ \operatorname{tr}\Big( (\widetilde{W} + W)^\top (\widetilde{W} + W) \Big) \Big] \\
&= \mathbb{E}_{Z_0, W} \Big[ \operatorname{tr}\Big( \widetilde{W}^\top \widetilde{W} \Big) + 2 \operatorname{tr}\Big( W^\top \widetilde{W} \Big) + \operatorname{tr}\big( W^\top W \big) \Big].
\end{aligned}$$

We will use the identity $\mathbb{E}_X[XAX^\top XBX^\top] = \big(\operatorname{tr}(A) \operatorname{tr}(B) + \operatorname{tr}\big(AB^\top\big) + d \operatorname{tr}(AB)\big) I_d$ for any $A, B \in \mathbb{R}^{n \times n}$, which can be acquired by expanding each element and applying Isserlis' theorem. Let $T_1 = \operatorname{tr}\big(MM^\top\big)$ and $T_2 = \operatorname{tr}\big(MM^\top MM^\top\big)$, then

$$\mathbb{E}_{Z_0, W} \big[ \operatorname{tr}\big( (aWXX^\top + bWXMM^\top X^\top)^\top (aWXX^\top + bWXMM^\top X^\top) \big) \big]$$

$$
\begin{aligned}
&= \mathbb{E}_{Z_0,W}\left[a^2 \operatorname{tr}\left(XX^\top W^\top WXX^\top\right) + 2ab \operatorname{tr}\left(XX^\top W^\top WXMM^\top X^\top\right)\right] \\
&\quad + \mathbb{E}_{Z_0,W}\left[b^2 \operatorname{tr}\left(XMM^\top X^\top W^\top WXMM^\top X^\top\right)\right] \\
&= d\,\mathbb{E}_{Z_0}\left[a^2 \operatorname{tr}\left(XX^\top XX^\top\right) + 2ab \operatorname{tr}\left(XX^\top XMM^\top X^\top\right) + b^2 \operatorname{tr}\left(XMM^\top X^\top XMM^\top X^\top\right)\right] \\
&= a^2d^2n(n+1+d) + 2abd^2(n+1+d)T_1 + b^2d^2(T_1^2 + (1+d)T_2).
\end{aligned}
$$

Simultaneously, we can verify that $\mathbb{E}_{Z_0,W}[\operatorname{tr}\left(W^\top W\right)] = d^2$ and

$$
\mathbb{E}_{Z_0,W}\left[\operatorname{tr}\left(W^\top \widetilde{W}\right)\right] = \mathbb{E}_{Z_0,W}\left[aW^\top WXX^\top + bW^\top WXMM^\top X^\top\right] = ad^2n + bd^2T_1.
$$

Combining the results above, we aim to find the optimal $a, b, M$ that minimize

$$
\frac{1}{d^2}\mathcal{L}(V,Q) = c_0 + c_1T_1 + c_2T_1^2 + c_3T_2,
$$

where

$$
\begin{gathered}
c_0 = a^2n(n+1+d) + 1 + 2an, \quad c_1 = 2ab(n+1+d) + 2b, \\
c_2 = b^2, \quad c_3 = b^2(1+d).
\end{gathered}
$$

Since $c_3 \geq 0$, to minimize $\mathcal{L}(V,Q)$ we need to minimize $T_2$. Given that $MM^\top$ is symmetric, we denote its $n$ eigenvalues as $\lambda_i$, $i \in [1,n]$. Then by Cauchy–Schwarz inequality,

$$
\operatorname{tr}\left(MM^\top MM^\top\right) = \sum_{i=1}^n \lambda_i^2 \geq \frac{1}{n}\left(\sum_{i=1}^n \lambda_i\right)^2 = \frac{1}{n}\operatorname{tr}^2(MM^\top).
$$

Therefore, $\mathcal{L}(V,Q)$ is minimized only if the inequality above holds with equality, which implies that $\lambda_i = \lambda_j$ for any $i \neq j$. This concludes the proof by showing that there exists $\lambda \in \mathbb{R}$ such that $MM^\top = \lambda I_d$, and thus $DD^\top = \operatorname{diag}(\lambda I_d, 0)$. □

### B.6 Proof of Proposition 6

*Proof.* We will continue from eqs. (37) and (38). After applying token-wise dropout, we have

$$
\begin{aligned}
&\mathsf{TF}(Z_0; \{V_l, Q_l\}_{l=1}^2, X_1 \leftarrow \overline{X}_1) = \sum_{i=1}^n (ao_2^{3i-2} + bo_2^{3i}) o_1^{3i-2} o_1^{3i} y_i x_i^\top o_1^{3n+1} o_2^{3n+3} x_{\text{test}} \\
&\quad + c\sum_{j=1}^n \sum_{k=1}^n \left(\sum_{i=1}^{n+1} o_2^{3i-1} d_{j,i} d_{k,i}\right) o_1^{3j} o_1^{3k-2} y_j x_k^\top o_1^{3n+1} o_2^{3n+3} x_{\text{test}}, \qquad (40) \\
&\mathsf{TF}(Z_0; \{V_l, Q_l\}_{l=1}^2, X_1 \leftarrow \Delta X_1) = co_2^{3n+3} \sum_{j=1}^n \left(\sum_{i=1}^{n+1} d_{j,i} d_{n+1,i}\right) o_1^{3j} o_1^{3n+1} y_j x_{\text{test}}^\top x_{\text{test}},
\end{aligned}
$$

where $a = b_2c_2a_1d_x^y b_1 d_y^x (1 + a_1 d_x^x)$, $b = b_2c_2a_1d_x^y(1 + b_1d_y^x)a_1d_x^x$ and $c = b_2c_2a_1d_x^y b_1 t_y a_1 t_x$. One can verify that our previous analysis about $\mathsf{TF}(Z_0; \{V_l, Q_l\}_{l=1}^2, X_1 \leftarrow \Delta X_1)$ still holds and we thus have $d_{n+1,:} = 0$. We then define:

$$
\begin{gathered}
O_l^1 = \operatorname{diag}(o_l^1, \cdots, o_l^{3n-2}) \in \mathbb{R}^{n\times n}, \quad O_l^2 = \operatorname{diag}(o_l^3, \cdots, o_l^{3n}) \in \mathbb{R}^{n\times n}, \quad \text{for } l \in [2], \\
O_2^3 = \operatorname{diag}(o_2^2, \cdots, o_2^{3n+2}) \in \mathbb{R}^{(n+1)\times(n+1)}.
\end{gathered}
$$

By defining

$$
\widetilde{W} = \sum_{i=1}^n (ao_2^{3i-2} + bo_2^{3i}) o_1^{3i-2} o_1^{3i} y_i x_i^\top + c\sum_{j=1}^n \sum_{k=1}^n \left(\sum_{i=1}^{n+1} o_2^{3i-1} d_{j,i} d_{k,i}\right) o_1^{3j} o_1^{3k-2} y_j x_k^\top,
$$

One can verify that

$$
\widetilde{W} = A + B + C \triangleq aYO_1^2O_2^1O_1^1X^\top + bYO_1^2O_2^2O_1^1X^\top + cYO_1^2MO_2^3M^\top O_1^1X^\top.
$$

Then, we will compute the expectation of each term in the following decomposition:

$$\mathcal{L}(V,Q) = \mathbb{E}_{Z_0,W}\Big[\mathrm{tr}\Big(\widetilde{W}^\top\widetilde{W}\Big) + 2\,\mathrm{tr}\Big(W^\top\widetilde{W}\Big) + \mathrm{tr}\big(W^\top W\big)\Big],$$

Specifically, let $T_1 = \mathrm{tr}\big(MM^\top\big)$, $T_2 = \mathrm{tr}\big(MM^\top MM^\top\big)$, $T_3 = \|M\|_4^4$, $T_4 = \sum_{i=1}^n \|M_{i,:}\|_2^4$, $T_5 = \sum_{j=1}^{n+1} \|M_{:,j}\|_2^4$, we then have

$$\begin{aligned}
\mathbb{E}[\mathrm{tr}\big(A^\top A\big)] &= a^2d^2(np^3 + n(n-1)p^6 + (1+d)np^3),\\
\mathbb{E}[\mathrm{tr}\big(B^\top B\big)] &= b^2d^2(np^3 + n(n-1)p^6 + (1+d)np^3),\\
\mathbb{E}[\mathrm{tr}\big(C^\top C\big)] &= c^2d^2(p^6T_1^2 + (1+d)(p^4-p^6)T_4 + (1+d)(p^5-p^6)T_5\\
&\quad + (1+d)(p^3-p^4-p^5+p^6)T_3 + (p^3-p^4)T_4 + p^4T_2 + dp^6T_2),\\
\mathbb{E}[\mathrm{tr}\big(A^\top B\big)] &= abd^2(np^4 + n(n-1)p^6 + (1+d)np^4),\\
\mathbb{E}[\mathrm{tr}\big(A^\top C\big)] &= acd^2((p^4 + (n-1)p^6)T_1 + (1+d)p^4T_1),\\
\mathbb{E}[\mathrm{tr}\big(B^\top C\big)] &= bcd^2((p^4 + (n-1)p^6)T_1 + (1+d)p^4T_1),\\
\mathbb{E}[\mathrm{tr}\big(W^\top A\big)] = ad^2np^3, &\quad \mathbb{E}[\mathrm{tr}\big(W^\top B\big)] = bd^2np^3, \quad \mathbb{E}[\mathrm{tr}\big(W^\top C\big)] = cd^2p^3T_1.
\end{aligned}$$

Summarizing our analysis above, $\min_M \mathcal{L}(V,Q)$ is equivalent to:

$$\min_M \big\{c_0 + c_1T_1 + c_2T_2 + c_3T_3 + c_4T_4 + c_5T_5 + c_6T_1^2\big\},$$

where

$$\begin{aligned}
c_0 &= 1 + n(2+d)p^3(a^2+b^2) + 2np^3(a+b) + 2n(2+d)p^4ab + n(n-1)p^6(a+b)^2,\\
c_1 &= 2(a+b)c(p^4 + (n-1)p^6 + (1+d)p^4) + 2cp^3,\\
c_2 &= c^2(p^4 + dp^6),\\
c_3 &= c^2(1+d)(p^3 - p^4 - p^5 + p^6),\\
c_4 &= c^2((1+d)(p^4-p^6) + (p^3-p^4)),\\
c_5 &= c^2(1+d)(p^5-p^6),\\
c_6 &= c^2p^6.
\end{aligned}$$

It is easy to verify that $c_2, c_3, c_4, c_5, c_6 \geq 0$. □

### B.7 Proof of Proposition 7

**Proposition 7** (Restate)**.** *Let $d_p$ denote the number of non-EOS tokens. Given any $L$-layer, single-head, $d$-dimensional linear-attention transformer with EOS tokens:*

$$\mathsf{TF}\big(Z_0; \{V_l, Q_l, P_l\}_{l\in[L]}\big) = (Z_L)_{:,d_p+1}, \quad (Z_0)_{:,d_p+1} = 0,$$

*where*

$$\begin{aligned}
&Z_l \in \mathbb{R}^{d\times(d_p+1)},\ V_l, Q_l \in \mathbb{R}^{d\times d},\ P_l \in \mathbb{R}^{(d_p+1)\times(d_p+1)},\\
&Z_l = Z_{l-1} + V_lZ_{l-1}M(Z_{l-1}^\top Q_l Z_{l-1}^\top + P_l), \quad M = \mathrm{diag}(I_{d_p}, 0).
\end{aligned}$$

*There exists an $L$-layer, two-head, $2d$-dimensional linear-attention transformer operating without EOS tokens:*

$$\mathsf{TF}\big(\overline{Z}_0; \{\overline{V}_l^h, \overline{Q}_l^h, \overline{P}_l^h\}_{l\in[L],h\in[2]}\big) = (\overline{Z}_L)_{d:2d,d_p},$$

*where*

$$\begin{aligned}
&\overline{Z}_l \in \mathbb{R}^{2d\times d_p},\ \overline{V}_l^h, \overline{Q}_l^h \in \mathbb{R}^{2d\times 2d},\ \overline{P}_l^h \in \mathbb{R}^{d_p\times d_p},\\
&\overline{Z}_l = \overline{Z}_{l-1} + \sum_{h=1}^{2} \overline{V}_l^h\overline{Z}_{l-1}(\overline{Z}_{l-1}^\top\overline{Q}_l^h\overline{Z}_{l-1}^\top + \overline{P}_l^h).
\end{aligned}$$

*Such that for any $Z \in \mathbb{R}^{d\times d_p}$, by letting $Z_0 = [Z \quad 0]$ and $\overline{Z}_0 = \begin{bmatrix} Z \\ 0 \end{bmatrix}$, we have*

$$\mathsf{TF}\big(Z_0; \{V_l, Q_l, P_l\}_{l\in[L]}\big) = \mathsf{TF}\big(\overline{Z}_0; \{\overline{V}_l^h, \overline{Q}_l^h, \overline{P}_l^h\}_{l\in[L],h\in[2]}\big).$$

*Proof.* We construct $\overline{V}_l^h$, $\overline{Q}_l^h$, and $\overline{P}_l^h$ as follows:

$$\overline{V}_l^1 = \begin{bmatrix} V_l & 0 \\ 0 & 0 \end{bmatrix}, \quad \overline{Q}_l^1 = \begin{bmatrix} Q_l & 0 \\ 0 & 0 \end{bmatrix}, \quad \overline{P}_l^1 = (P_l)_{1:d_p,1:d_p},$$
$$\overline{V}_l^2 = \begin{bmatrix} 0 & 0 \\ V_l & 0 \end{bmatrix}, \quad \overline{Q}_l^2 = \begin{bmatrix} 0 & Q_l \\ 0 & 0 \end{bmatrix}, \quad \overline{P}_l^2 = \begin{bmatrix} 0 & (P_l)_{:,d_p+1} \end{bmatrix}.$$

We will show that for any $l \in [L]$, it satisfies $\overline{Z}_l = \begin{bmatrix} (Z_l)_{:,(1:d_p-1)} & (Z_l)_{:,d_p} \\ 0 & (Z_l)_{:,d_p+1} \end{bmatrix}$. One can verify that it holds trivially for $l = 0$. Then, suppose it holds for some $l = k-1$, we have

$$\begin{aligned}
\overline{Z}_k &= \overline{Z}_{k-1} + \overline{V}_k^1 \overline{Z}_{k-1} (\overline{Z}_{k-1}^\top \overline{Q}_k^1 \overline{Z}_{k-1}^\top + \overline{P}_k^1) + \overline{V}_k^2 \overline{Z}_{k-1} (\overline{Z}_{k-1}^\top \overline{Q}_k^2 \overline{Z}_{k-1}^\top + \overline{P}_k^2) \\
&= \overline{Z}_{k-1} + \begin{bmatrix} V_k (Z_{k-1})_{:,1:d_p} \Big( (Z_{k-1})_{:,1:d_p}^\top Q_k (Z_{k-1})_{:,1:d_p} + (P_k)_{1:d_p,1:d_p} \Big) \\ 0 \end{bmatrix} \\
&\quad + \begin{bmatrix} 0 \\ V_k (Z_{k-1})_{:,1:d_p} \end{bmatrix} \Big( \begin{bmatrix} 0 & (Z_{k-1})_{:,1:d_p}^\top Q_k (Z_{k-1})_{:,d_p+1} \end{bmatrix} + \begin{bmatrix} 0 & (P_k)_{:,d_p+1} \end{bmatrix} \Big) \\
&= \overline{Z}_{k-1} + \begin{bmatrix} V_k Z_{k-1} M \big( Z_{k-1}^\top Q_k (Z_{k-1})_{:,1:d_p} + (P_k)_{:,1:d_p} \big) \\ 0 \end{bmatrix} \\
&\quad + \begin{bmatrix} 0 & 0 \\ 0 & V_k Z_{k-1} M \big( Z_{k-1}^\top Q_k (Z_{k-1})_{:,d_p+1} + (P_k)_{:,d_p+1} \big) \end{bmatrix} \\
&= \begin{bmatrix} (Z_k)_{:,1:d_p} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & (Z_k)_{:,d_p+1} \end{bmatrix}.
\end{aligned}$$

The proof is complete. □

# C Experiment Details and Additional Results

In this section, we present experiment details and additional results not included in the main text due to space limitations. Our experiments are conducted on an A100 40G GPU. It takes around 30 GPU hours to fully reproduce our results[1].

## C.1 Synthetic Experiments on Linear Transformers

We consider training linear-attention transformers on random linear regression instances. We take embedding dimension $d = 4$, and the distributions for generating $x_i$ and $w_i$ are both $P_x = P_w = \mathcal{N}(0, I_d)$. We optimize the ICL risk for $L$-layer linear transformers with $n$ in-context demonstrations using AdamW, where $L \in [3]$ and $n \in [5, 30]$. Each gradient step is computed from a batch size of 1000. We additionally apply $\ell_1$ regularization to simplify the found solutions. For training efficiency and stability, we restrict the $A_l$, $B_l$, and $C_l$ matrices to $\mathcal{S}_I$ during training, and initialize $D_l \in \mathbb{R}^{d_p \times d_p}$ with i.i.d. Gaussian matrices. For each case, we train 40 models with different random seeds, and report the minimum achieved ICL risk to approximate the global minimum.

To reproduce the task vector mechanism, we focus on transformers trained with triplet-formatted prompts. The training procedure is identical to the above. For inference, we restrict $P_w$ to rank-one coefficient matrices, by letting $W = w_1 w_2^\top$, where $w_1, w_2 \sim \mathcal{N}(0, I_d)$. We first generate normal ICL prompts to generate task vectors as the hidden states of the last arrow token after the first attention layer, and then inject them into zero-shot prompts after normalization. The final outputs $\hat{y}_{\text{test}}$ are taken as the output of these injected zero-shot prompts after being processed with the same transformer model. We compute the final risk as $\mathbb{E} \left\| \frac{\hat{y}_{\text{test}}}{\|\hat{y}_{\text{test}}\|} + \frac{y_{\text{test}}}{\|y_{\text{test}}\|} \right\|$ to simulate the layer normalization blocks in practical LLMs. The reported scores are averaged for $n \in [5, 30]$.

## C.2 Experiments on Practical LLMs

**Datasets.** Following the settings of the original task vector method [7], our study covers 33 tasks in 5 categories. The detailed description for each task is provided in Table 3.

[1]The source code is available in supplementary materials.

Table 3: Descriptions of the tasks used in our empirical studies.

| Category | Task | Example | Description |
|---|---|---|---|
| Knowledge | Contry to Capital | France $\rightarrow$ Paris | Output the capital city of the given country. |
| | Person to Language | Macron $\rightarrow$ French | Output the native language of the given person. |
| | Location to Continent | Paris $\rightarrow$ Europe | Output the corresponding continent of the given location. |
| | Religion | Saladin $\rightarrow$ Muslim | Output the associated religion of the given location or person. |
| Algorithmic | List First | [a,b,c] $\rightarrow$ a | Output the first item in the given list. |
| | List Last | [a,b,c] $\rightarrow$ c | Output the last item in the given list. |
| | Next Letter | a $\rightarrow$ b | Output the next letter of the given letter in the alphabet. |
| | Prev Letter | b $\rightarrow$ a | Output the previous letter of the given letter in the alphabet. |
| | To Upper | a $\rightarrow$ A | Output the corresponding uppercase letter of the given lowercase letter. |
| | To Lower | A $\rightarrow$ a | Output the corresponding lowercase letter of the given uppercase letter. |
| Translation | English to French | hello $\rightarrow$ bonjour | Translate the given word in English to French. |
| | English to Italian | hello $\rightarrow$ ciao | Translate the given word in English to Italian. |
| | English to Spanish | hello $\rightarrow$ hola | Translate the given word in English to Spanish. |
| | French to English | bonjour $\rightarrow$ hello | Translate the given word in French to English. |
| | Italian to English | ciao $\rightarrow$ hello | Translate the given word in Italian to English. |
| | Spanish to English | hola $\rightarrow$ hello | Translate the given word in Spanish to English. |
| Linguistic | Present to Gerund | go $\rightarrow$ going | Output the corresponding gerund form of the given verb in present simple tense. |
| | Present to Past | go $\rightarrow$ went | Output the corresponding past simple form of the given verb in present simple tense. |
| | Present to Past Perfect | go $\rightarrow$ gone | Output the corresponding past perfect form of the given verb in present simple tense. |
| | Gerund to Present | going $\rightarrow$ go | Output the corresponding present simple form of the given verb in gerund form. |
| | Past to Present | went $\rightarrow$ go | Output the corresponding present simple form of the given verb in past simple tense. |
| | Past Perfect to Present | gone $\rightarrow$ go | Output the corresponding present simple form of the given verb in past perfect tense. |
| | Singular to Plural | dog $\rightarrow$ dogs | Output the corresponding plural form of the given noun in singular form. |
| | Plural to Singular | dogs $\rightarrow$ dog | Output the corresponding singular form of the given noun in plural form. |
| | Antonym | happy $\rightarrow$ sad | Output the antonym of the given adjective. |
| Bijection | To Upper & Lower | a $\leftrightarrow$ A | Output the given letter in uppercase if it is in lowercase, and vice versa. |
| | English & French | hello $\leftrightarrow$ bonjour | Translate the given word to French if it is in English, and vice versa. |
| | English & Italian | hello $\leftrightarrow$ ciao | Translate the given word to Italian if it is in English, and vice versa. |
| | English & Spanish | hello $\leftrightarrow$ hola | Translate the given word to Spanish if it is in English, and vice versa. |
| | Present & Gerund | go $\leftrightarrow$ going | Output the given verb in gerund form if it is in present simple tense, and vice versa. |
| | Present & Past | go $\leftrightarrow$ went | Output the given verb in past simple form if it is in present simple tense, and vice versa. |
| | Present & Past Perfect | go $\leftrightarrow$ gone | Output the given verb in past perfect form if it is in present simple tense, and vice versa. |
| | Singular & Plural | dog $\leftrightarrow$ dogs | Output the given noun in plural form if it is in singular form, and vice versa. |

Table 4: Accuracy comparison between standard ICL (Baseline), the task vector method (TaskV), and our strategy (TaskV-M). The experiment is conducted on Pythia-12B with $n = 10$.

| Method | | Knowledge | Algorithmic | Translation | Linguistic | Bijection | Average |
|---|---|---|---|---|---|---|---|
| 0-shot | Baseline | 6.60 ± 1.59 | 14.07 ± 1.45 | 8.60 ± 0.68 | 12.53 ± 1.57 | 10.31 ± 0.70 | 10.82 ± 0.48 |
| | TaskV | **63.30** ± 2.62 | **84.73** ± 1.22 | **62.07** ± 0.98 | **82.58** ± 1.22 | **42.27** ± 0.92 | **66.40** ± 0.96 |
| 1-shot | Baseline | 61.80 ± 5.45 | 72.80 ± 1.15 | 43.27 ± 2.92 | 57.07 ± 1.15 | 41.91 ± 2.83 | 53.95 ± 1.02 |
| | TaskV | 76.40 ± 2.40 | **84.20** ± 1.05 | **71.47** ± 1.41 | **87.16** ± 2.04 | 53.11 ± 2.37 | 73.59 ± 0.79 |
| | TaskV-M | **77.70** ± 2.52 | 83.73 ± 1.37 | 71.00 ± 1.48 | 86.80 ± 1.59 | **53.87** ± 2.90 | **73.68** ± 0.90 |
| 2-shot | Baseline | 70.30 ± 3.71 | 82.13 ± 0.54 | 60.80 ± 1.81 | 81.16 ± 1.57 | 50.76 ± 2.17 | 68.41 ± 0.64 |
| | TaskV | 80.30 ± 2.46 | **87.00** ± 1.63 | 76.13 ± 3.77 | 89.33 ± 0.70 | 58.67 ± 2.44 | 77.41 ± 0.50 |
| | TaskV-M | **81.60** ± 1.56 | 86.47 ± 0.40 | **77.27** ± 2.53 | **89.51** ± 0.88 | **59.24** ± 2.48 | **77.87** ± 0.76 |
| 3-shot | Baseline | 77.60 ± 2.40 | 81.87 ± 0.81 | 68.13 ± 2.02 | 86.31 ± 1.93 | 55.73 ± 1.60 | 73.20 ± 0.31 |
| | TaskV | 84.00 ± 2.76 | 86.33 ± 1.17 | **79.53** ± 2.27 | 92.00 ± 0.67 | 58.76 ± 1.53 | 79.06 ± 0.67 |
| | TaskV-M | **85.40** ± 2.31 | **87.07** ± 1.18 | 78.13 ± 1.86 | **92.84** ± 0.68 | **59.56** ± 1.27 | **79.54** ± 0.35 |
| 4-shot | Baseline | 78.40 ± 1.83 | 82.73 ± 0.44 | 72.40 ± 1.24 | 88.89 ± 1.25 | 57.91 ± 1.46 | 75.46 ± 0.64 |
| | TaskV | 83.80 ± 1.12 | 87.60 ± 1.81 | **80.20** ± 2.39 | **92.18** ± 0.96 | 59.38 ± 0.47 | 79.59 ± 0.62 |
| | TaskV-M | **84.30** ± 1.50 | **88.13** ± 0.81 | 80.00 ± 2.67 | 91.87 ± 1.25 | **60.31** ± 0.86 | **79.87** ± 0.51 |

**Prompt Template.** The template used to construct ICL demonstrations is "Example:$\{x_i\} \to \{y_i\}$, where $x_i$ and $y_i$ are subsequently replaced by the input and output of the semantic mapping. For the query part, $y_i$ is omitted from the prompt. After concatenating each demonstration with "\n", an example of the full input prompt is:

$$\text{Example:}\{x_1\} \to \{y_1\}\text{\n} \cdots \text{Example:}\{x_n\} \to \{y_n\}\text{\nExample:}\{x_{\text{test}}\} \to$$

**Evaluation.** To evaluate the $N$-shot performance, we generate $50 \times (N + 1)$ i.i.d. prompts for each task with number of demonstrations $n = 10$ for task vector extraction. The hidden states of the last $\to$ token, which is also literally the last token in the prompt, are recorded for every layer in the transformer. Thereafter, we generate another $50$ i.i.d. prompts with $N$ demonstrations, where $x_{\text{test}}$ is selected to be distinct from the previous chosen ones. The final accuracy is measured by whether the next word predicted matches the expected answer. The performance of the standard ICL method (Baseline) is acquired by inferring without interference. For the task vector method (TaskV) and our multi-vector variant (TaskV-M), the extracted task vectors are injected to replace the hidden states of the arrow $\to$ tokens at a specified layer $l$. For TaskV, only the last arrow token is injected, while for TaskV-M, each of the $N + 1$ arrow tokens is injected with the $N + 1$ extracted task vectors for the same task. The performance is reported for the one layer $l \in L$ achieving the highest accuracy. For each case, the mean and standard deviation are evaluated through $5$ independent trials.

**Additional Results.** Besides Llama-13B, we also observe consistent accuracy improvement of our TaskV-M method on the Pythia-12B model, as reported in Table 4.

# D Additional Discussions

## D.1 Last Task Vector Weights the Most

While our analysis of linear-attention models suggests that each formed task vector (i.e., the hidden state at each arrow token) contributes equally to the final prediction, this assumption does not fully hold in practical LLMs. As demonstrated by the conflicting tasks experiment in [7], injecting a task vector from task $B$ into an ICL prompt designed for task $A$ causes the model to predominantly perform task $B$. This behavior indicates that LLMs largely rely on the last arrow token to determine the task identity. We attribute this to the causal attention mechanism used in practical LLMs, which is not captured by our current theoretical analysis. In causal attention, only the final arrow token can aggregate information from the entire preceding context, making it the most informative and influential for prediction. This explains why our multi-vector strategy offers modest, though consistent, performance gains. The improvement suggests that intermediate arrow tokens do participate in the inference process, albeit less effectively. Enhancing how LLMs utilize information from all arrow tokens remains a promising direction for improving task vector accuracy and robustness.

### D.2 Decoding the Vocabulary of Task Vectors

Multiple prior works [7, 16] have observed an interesting phenomenon: when task vectors are extracted and passed through the final classification layer, the top predicted tokens often belong to the output space of the corresponding task. This effect is particularly prominent in the GPT-J model. Interestingly, we find that this behavior can be naturally explained by our analysis of linear models. Specifically, we assume that the hidden state space has dimensionality at least $2d$, where the first $d$ dimensions represent the input ($x_i$) and the last $d$ dimensions represent the output ($y_i$). Task vectors constructed under this architecture preserve this layout: the first half encodes a linear combination of $x_i$, and the second half encodes a linear combination of $y_i$. In the final layer, the model predicts $y_{\text{test}}$ by extracting the last $d$ dimensions of the final token. When this same mechanism is applied to a task vector, it naturally produces a linear combination of the $y_i$ values, thereby generating outputs aligned with the task's output space. This indicates that practical LLMs adopt a similar partition in the hidden state space, justifying our prompt structure for linear model analysis.

### D.3 Limitations

While our analysis provides new insights into the emergence and functionality of task vectors, it is primarily conducted on simplified linear-attention transformers and synthetic tasks, which may not fully capture the complexity of real-world LLMs. Moreover, our theoretical framework focuses on middle-layer representations and does not fully account for deeper interactions across layers or the role of fine-tuned components such as layer normalization and multi-head attention.

### D.4 Broader Impacts

This work advances the theoretical understanding of in-context learning and task vector mechanisms, which can lead to more efficient and interpretable language models. By enabling faster inference through task vectors, it may reduce the computational cost and energy consumption of large-scale deployment, thereby making AI systems more accessible and environmentally sustainable. Improved interpretability could also enhance trust and transparency in AI applications across education, healthcare, and other socially beneficial domains.

As task vector methods improve efficiency and transferability, they may also be misused to replicate or extract functionality from proprietary models without authorization, raising concerns around model intellectual property. Additionally, while interpretability is often framed as a benefit, deeper insights into model internals could be exploited to engineer adversarial inputs or extract sensitive training data. Careful consideration and mitigation strategies are essential to ensure that such work aligns with the broader goals of safe and beneficial AI.

# NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).

- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.