Causal In-Context Learning in Transformers: Training Dynamics Across Heterogeneous Interventional Data

Shanyun Gao
Purdue University

Murat Kocaoglu
Johns Hopkins University

Qifan Song
Purdue University

Abstract

Large language models (LLMs) have shown strong abilities in generalization, adaptation, and reasoning. Among these capabilities, in-context learning (ICL) has emerged as a defining property of LLMs. ICL enables models to separate and exploit heterogeneous data arising from different environments or tasks while producing context-dependent solutions. In this work, we show that such in-context learning capabilities can arise even in a simple two-layer Transformer with a single attention head per layer when trained on heterogeneous data with causal structure. Through an analysis of training dynamics during next-token prediction on diverse interventional datasets, we observe two phases: an initial phase that learns the parents of a target variable, followed by a later phase that captures the full Markov boundary. By leveraging the crucial role of positional embeddings, we show that in-context learning can be used for local causal discovery under certain interventions without knowing from which environment or interventional distribution each prompt is drawn.

1 Introduction

Transformers [Vaswani et al., 2023] have gained a lot of attention because of the remarkable abilities they show, such as in context learning [Dong et al., 2024, Han et al., 2023, Brown et al., 2020]. In-context learning is notable because the model can learn to learn from examples in the prompt without updating its weights. A growing body of research investigates how Transformers perform in-context learning across different tasks [Hendel et al., 2023], such as linear regression [Akyürek et al., 2023, Zhang et al., 2023, Bai et al., 2023] and Markov chains [Edelman et al., 2024].

Not only the final results matter. The training dynamics of the training process also contain information about the data and about how the model learns an optimal solution step by step [Singh et al., 2023, Olsson et al., 2022, Reddy, 2023, Nanda et al., 2023]. For example, work on induction heads and Markov chain in context learning gives concrete analyses of training time behavior in small transformer settings [Chen et al., 2024, Zhang et al., 2025, He et al., 2025].

In the causal context, transformers are being used as tools for causal discovery, and some work demonstrates that the attention score can be used to recover causal relations, either by directly detecting relations or by extracting causal structure from model signals [Rohekar et al., 2023, Nichani et al., 2024, Butkus and Kriegeskorte, 2025, Montagna et al., 2025].

We consider training a simple two-layer Transformer with a standard next token prediction objective on heterogeneous interventional data. In this setting, existing work has not characterized the training dynamics, and it remains unclear whether those dynamics can be used to discover causal relations.

1.1 Contributions

Our contributions are as follows.

- We introduce a novel in-context learning problem, termed *heterogeneous discrete prompts with causal structure*. In this task, the underlying directed acyclic graph (DAG) is fixed but unknown to the Transformer. Each prompt consists of IID samples from a single causal mechanism, while different prompts correspond to different causal mechanisms induced by multiple soft interventions on specific variables in the causal graph.

- We observe that next-token prediction training dynamics exhibit two phases. The first phase corresponds to the model learning the invariant causal predictors of the target variable, while the second corresponds to learning its Markov boundary. The Transformer converges to an optimal in-context solution by learning, for each interventional distri-

bution, the conditional probability of the target variable given its Markov boundary.

- We propose a local causal discovery algorithm that leverages positional embeddings to identify the parents of the target variable under certain interventions, without knowing from which environment or interventional distribution each prompt is drawn.

2 Setup

2.1 Transformer Architecture

For a detailed description of a multi-layer Transformer, we refer readers to Section 2.1 of [Nichani et al., 2024]. We emphasize that the two-layer Transformer we use is a standard Transformer, with a linear token embedding, positional embeddings, and a single attention head in each layer.

2.2 Problem Setup: Discrete-valued Prompt with Causal Structure

Denote $[a] \coloneqq \{1, \cdots, a\}$ and $[a, b] \coloneqq \{a, \cdots, b\}$, where $a, b \in \mathbb{N}^+$. Sets are usually denoted by bold letters, while individual variables are denoted by non-bold letters. Let $\mathcal{G}(\mathbf{V}, \mathbf{E})$ denote the underlying causal graph with the variable set $\mathbf{V}$ and edge set $\mathbf{E}$. For each variable $X \in \mathbf{V}$, the variables with directed edges pointing into X form its parent set, denoted by $\mathrm{Pa}(X)$. We denote a single parent as $\mathrm{pa}(X) \in \mathrm{Pa}(X)$. Similarly, the variables to which directed edges point out from X form its set of children, denoted by Child(X), and an individual child is denoted by $\mathrm{child}(X) \in \mathrm{Child}(X)$. $\mathrm{MB}(X)$ denotes the Markov boundary of X.

In our task, we use $\mathbf{X}$ to denote the predictor variables and Y to denote the target variable to be predicted. Accordingly, we have $\mathbf{X} \cup \{Y\} = \mathbf{V}$. Note that in this work all variables share the same discrete-valued support, $S = \{0, 1, 2, \cdots\}$; therefore, the only distinction between the variables in $\mathbf{X}$ and Y lies in their positions within the input data. Specifically, Y appears as the final token of each sample unit in every prompt.

ICL Next-Token Prediction Task (heterogeneous discrete-valued prompts with causal structure)

1. Generate a random Directed Acyclic Graph $\mathcal{G}$ given the number of variables $m \coloneqq |\mathbf{V}|$.

2. Generate a set of random conditional distributions corresponding to each causal mechanism implied by $\mathcal{G}$, given the size of the discrete-valued support $|S|$. Repeat this process E times to obtain E distinct sets of conditional distributions. Each such generation can be interpreted as a *soft intervention*, in which the functional form of the conditional distributions is modified while the underlying causal graph $\mathcal{G}$ remains unchanged.

For each predictor $X_i \in \mathbf{X}$ with $i \in [m-1]$, there are E distinct conditional distributions $P(X_i \mid \mathrm{Pa}(X_i))$, indicating that the causal mechanism of each predictor X_i varies across the E environments, while the conditional distribution $P(Y \mid \mathrm{Pa}(Y))$ remains invariant across all E environments.

3. Specify a probability mass function $P(E)$ over a finite or countable set of environments.

4. Sample an environment $E \sim P(E)$ prior to generating each training input sequence. Condition on the realized environment $E = e$ to generate discrete-valued IID samples from the corresponding Structural Causal Model (SCM) associated with environment e. Finally construct prompts spanning multiple environments, where each prompt consists of n complete IID samples, followed by a final sample in which the predictors $\mathbf{X}$ are observed and the target variable Y is separated and treated as the prediction target.

As a result, we obtain heterogeneous prompts that share the same causal graph $\mathcal{G}$ but differ in their causal mechanisms across E distinct environments.

Therefore, the k-th input prompt can be formalized as:

$$\mathrm{D}^k \coloneqq \{\mathbf{X}_1^k, Y_1^k, \mathbf{X}_2^k, Y_2^k, \cdots, \mathbf{X}_{n+1}^k\}, k \in [K]$$

where k indexes the prompt. We further denote $|\mathbf{X}_i^k| = m - 1$ for any prompt index $k \in [K]$ and sample index $i \in [n+1]$.

The loss function is therefore defined as

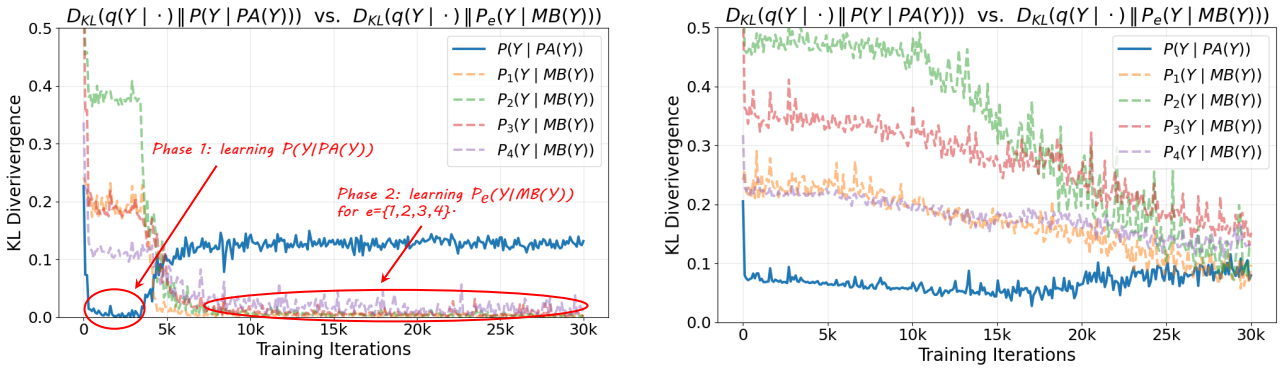
$$\mathcal{L}_\theta = \mathbb{E}_P \left[\sum_{k=1}^{K} \ell\Big(f_\theta(\mathrm{D}^k), Y_{n+1}^k \Big) \right], \tag{1}$$

where $f_\theta(\cdot)$ denotes the output of a model parameterized by θ that aims to minimize the loss $\mathcal{L}_\theta$, and $\ell(\cdot, \cdot)$ is a chosen loss function. Let P denote the joint distribution over prompts.

3 In-context Learning: Dynamic Training Process

By training a two-layer Transformer on the ICL next-token prediction task with the objective in Eq. 1, we observe that positional embeddings play a critical role in the model's performance.

Consider two Transformers with identical architecture (e.g., two layers, one head per layer, same width and MLP), denoted by $G_{\mathrm{p}}(\cdot; \theta)$ (with positional embeddings) and $G_{\mathrm{w/p}}(\cdot; \theta)$ (without positional embeddings). The only difference is the inclusion/exclusion of positional embeddings in the input representation.



(a) Training dynamics of $G_{\mathrm{p}}(\cdot;\theta)$

(b) Training dynamics of $G_{\mathrm{w/p}}(\cdot;\theta)$

Figure 1: Training dynamics on discrete-valued sequences generated from 4 environments (SCMs). Panel (a) shows the dynamics of $G_{\mathrm{p}}(\cdot;\theta)$, where Phase I learns $\mathrm{Pa}(Y)$ and Phase II captures $\mathrm{MB}(Y)$. Panel (b) shows that $G_{\mathrm{w/p}}(\cdot;\theta)$ fails to learn either $\mathrm{Pa}(Y)$ or $\mathrm{MB}(Y)$.

With positional embedding, we observe that the training dynamics exhibit two distinct phases. At the end of Phase I, the model learns $\mathrm{Pa}(Y)$, and at the end of Phase II, the model learns $\mathrm{MB}(Y)$. The performance evaluated throughout the training process is illustrated in Fig. 1.

3.1 Positional Embedding

Observation 3.1 (Faster Convergence to Lower Loss with Positional Embedding)**.** *Let $\{\theta_t^{\mathrm{p}}\}_{t\geq 0}$ and $\{\theta_t^{\mathrm{w/p}}\}_{t\geq 0}$ be the parameter sequences produced by the same training procedure (including optimizer, learning-rate schedule, batch size, and initialization distribution) applied to G_{p} and $G_{\mathrm{w/p}}$, respectively. Let $\mathcal{L}_{\mathrm{test}}(\theta)$ denote the test loss.*

For any accuracy level $\varepsilon > 0$, define the hitting times

$$T_{\mathrm{p}}(\varepsilon) := \inf\Big\{t \in \mathbb{N} : \ \mathbb{E}\big[\mathcal{L}_{\mathrm{test}}(\theta_t^{\mathrm{p}})\big] \leq \varepsilon\Big\}, \tag{2}$$

$$T_{\mathrm{w/p}}(\varepsilon) := \inf\Big\{t \in \mathbb{N} : \ \mathbb{E}\big[\mathcal{L}_{\mathrm{test}}(\theta_t^{\mathrm{w/p}})\big] \leq \varepsilon\Big\}. \tag{3}$$

Empirically, we observe that for a range of target accuracies $\varepsilon \in (0, \varepsilon_0]$, the positional-embedding model tends to reach the target test loss in many fewer training steps, i.e.,

$$T_{\mathrm{p}}(\varepsilon) \ < \ T_{\mathrm{w/p}}(\varepsilon) \tag{4}$$

Empirically, we observe a substantial performance gap between $G_{\mathrm{p}}(\cdot;\theta)$ and $G_{\mathrm{w/p}}(\cdot;\theta)$ both in terms of the learned distribution, as shown in Fig. 1, and training loss, as shown in Fig. 3. In particular, $G_{\mathrm{w/p}}(\cdot;\theta)$ struggles to reduce the loss and quickly plateaus at a high level early in training, whereas $G_{\mathrm{p}}(\cdot;\theta)$ continues to reduce the loss and approaches the cross-entropy induced by the true conditional distribution $P(Y \mid \mathbf{X})$.

3.2 Two-Phase Training Dynamics under Causal Structure

Here, we state a claim and a theorem that guarantees the observed two-phase training dynamics. We note that Claim 3.2 is not fully proved as of the submission date.

Claim 3.2 (Phase I: Parent-Preferential Gradient on Positional Embeddings)**.** *Let $g_{\mathrm{pe}}(t) := \nabla_{w_{\mathrm{pe}}} \mathcal{L}(\theta(t))$ be the positional embedding gradient at training time t. Assume there exists an environment-identification ability $\alpha(t) \in [0,1]$ of the model at time t, interpreted as how well the model can infer which environment or prompt source generated the current context, with $\alpha(t)$ nondecreasing in t and $\alpha(0) = 0$, meaning that at initialization the model cannot distinguish heterogeneous prompts at all.*

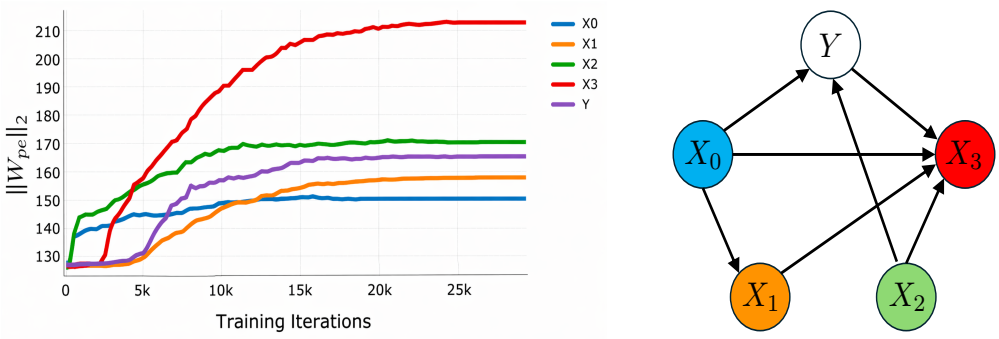
Then we have for all $t \in (0, T_I]$,

$$\big\|g_{\mathrm{pe}}^{Pa}(t)\big\|_2 \ - \ \big\|g_{\mathrm{pe}}^{\mathbf{X}\setminus Pa}(t)\big\|_2 \ \geq \ \frac{c}{\alpha(t)}, \tag{5}$$

for some constant $c > 0$ independent of t. Here $g_{\mathrm{pe}}^{Pa}(t)$ and $g_{\mathrm{pe}}^{\mathbf{X}\setminus Pa}(t)$ denote the components of the positional-embedding gradient attributable to the parent variables of Y and the non-parent variables, respectively.

Moreover, as $\alpha(t)$ increases with training, the threshold $\frac{c}{\alpha(t)}$ decreases, capturing a transition from treating data as drawn from the mixture distribution (small $\alpha(t)$) toward environment- or prompt-conditioned behavior (large $\alpha(t)$), more specifically toward in-context learning.

Claim 3.2 states that for early training time $t \in [0, T_I]$, the positional-embedding gradients are dominated by contributions from the parent variables of Y, while



(a) Positional embedding weights aggregated over embedding dimensions for each variable

(b) Causal graph used in the corresponding experiment

Figure 2: The dynamics of the learned positional embedding structure and the underlying causal graph. Panel (a) shows how the positional embedding weights, aggregated over embedding dimensions for each variable, evolve over training steps. Panel (b) shows the corresponding causal graph used in the experiment. This comparison and the color matching illustrate how the learned positional embedding patterns reflect the structural roles of variables in the causal graph. In particular, the model first captures the parents of the target variable Y, $\mathrm{Pa}(Y) = \{X_0, X_2\}$, and then identifies the remaining variables in the Markov boundary, including the child X_3 and the spouse X_1.

non-parent variables have weaker influence. As training progresses and the model improves its ability to distinguish environments, this preference gradually diminishes, reflecting a transition from learning invariant structure to environment-specific behavior.

A toy example is shown in Fig. 2. For better visualization, instead of showing the gradient directly, we plot in Fig. 2(a) the norm of the positional embedding weights, aggregated over embedding dimensions for each variable, and track how it evolves throughout training. The underlying causal graph for this example is shown in Fig. 2(b). In Fig. 2(a), the blue and green curves, corresponding to $\mathrm{Pa}(Y) = \{X_0, X_2\}$, increase immediately at the beginning of training and almost simultaneously, which corresponds to Phase I. Around 1,500 training steps, the model begins to capture the child of Y, namely X_3 (red curve), and around 4,000 steps it starts to identify the spouse X_1 (orange curve). The time gap between learning $\mathrm{Pa}(Y)$ and learning $\mathrm{MB}(Y) \setminus \mathrm{Pa}(Y)$ corresponds to the transition from Phase I to Phase II. Moreover, the gap in learning time between X_3 and X_1 arises because the path $Y \to X_3 \leftarrow X_1$ becomes informative only **after** conditioning on X_3.

Theorem 3.3 (Phase II: Optimal Solution)**.** *By the end of Phase II, for each environment e, the model learns the optimal conditional distribution and uses all predictive information available in the input, so that*

$$q_e^*(Y \mid \mathbf{X}) \;=\; P(Y \mid \mathrm{MB}(Y), e)\,. \tag{6}$$

The theorem above characterizes the model's behavior in Phase II: for each environment e, the predictor becomes optimal by using all information contained in $\mathrm{MB}(Y)$.

3.3 Local Causal Discovery Algorithm

Based on the observed training dynamics of the model and the way positional embeddings capture the information associated with Phase I and Phase II, we propose a local causal discovery algorithm for identifying the parents of a target variable under interventions. The procedure is described in Algorithm 1.

Initial experiments on arbitrary six-node causal graphs achieve an average F_1 score of 0.93 for correctly identifying $\mathrm{Pa}(Y)$, with a standard deviation of 0.07, evaluated over 25 random graphs with 5 environments each. Moreover, the model also estimates $\widehat{P}(Y \mid \mathrm{Pa}(Y))$ and $\widehat{P}(Y \mid \mathrm{MB}(Y), e)$, achieving an average KL divergence of 0.04 ± 0.02 from the ground-truth distributions.

4 Conclusions

Through our analysis of next-token prediction training dynamics on diverse interventional datasets, we identify a two-phase training pattern: Phase I learns $\mathrm{Pa}(Y)$, while Phase II captures $\mathrm{MB}(Y)$. We further observe that positional embeddings are not only critical for in-context learning ability, but also encode information about the underlying graph structure. By leveraging this training dynamic and its manifestation in positional embeddings, ICL can be used for local causal discovery. We leave the extension to latent environment classification and intervention set identification for future work.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning, 2024. URL https://arxiv.org/abs/2301.00234.

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. Understanding in-context learning via supportive pretraining data, 2023. URL https://arxiv.org/abs/2306.15091.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors, 2023. URL https://arxiv.org/abs/2310.15916.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models, 2023. URL https://arxiv.org/abs/2211.15661.

Ruiqi Zhang, Spencer Frei, and Peter L. Bartlett. Trained transformers learn linear models in-context, 2023. URL https://arxiv.org/abs/2306.09927.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection, 2023. URL https://arxiv.org/abs/2306.04637.

Benjamin L. Edelman, Ezra Edelman, Surbhi Goel, Eran Malach, and Nikolaos Tsilivis. The evolution of statistical induction heads: In-context learning markov chains, 2024. URL https://arxiv.org/abs/2402.11004.

Aaditya K. Singh, Stephanie C. Y. Chan, Ted Moskovitz, Erin Grant, Andrew M. Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers, 2023. URL https://arxiv.org/abs/2311.08360.

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

Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task, 2023. URL https://arxiv.org/abs/2312.03002.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability, 2023. URL https://arxiv.org/abs/2301.05217.

Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Unveiling induction heads: Provable training dynamics and feature learning in transformers, 2024. URL https://arxiv.org/abs/2409.10559.

Yedi Zhang, Aaditya K. Singh, Peter E. Latham, and Andrew Saxe. Training dynamics of in-context learning in linear attention, 2025. URL https://arxiv.org/abs/2501.16265.

Jianliang He, Xintian Pan, Siyu Chen, and Zhuoran Yang. In-context linear regression demystified: Training dynamics and mechanistic interpretability of multi-head softmax attention, 2025. URL https://arxiv.org/abs/2503.12734.

Raanan Y. Rohekar, Yaniv Gurwicz, and Shami Nisimov. Causal interpretation of self-attention in pre-trained transformers, 2023. URL https://arxiv.org/abs/2310.20307.

Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent, 2024. URL https://arxiv.org/abs/2402.14735.

Eivinas Butkus and Nikolaus Kriegeskorte. Causal discovery and inference through next-token prediction. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=tHr0vFbS3K.

Francesco Montagna, Max Cairney-Leeming, Dhanya Sridhar, and Francesco Locatello. Demystifying amortized causal discovery with transformers, 2025. URL https://arxiv.org/abs/2405.16924.

Causal In-Context Learning in Transformers: Training Dynamics Across Heterogeneous Interventional Data: Supplementary Materials

A Partial Proof

Theorem A.2 (Phase II: Optimal Solution)**.** *By the end of Phase II, for each environment e, the model learns the optimal conditional distribution and uses all predictive information available in the input. In particular, this information is fully captured by the Markov boundary of Y, so that*

$$q_e^*(Y \mid \mathbf{X}) \;=\; P(Y \mid \mathrm{MB}(Y), e)\,. \tag{7}$$

Proof. The environment index variable E has a marginal distribution:

$$E \sim P(E) \tag{8}$$

Given an environment $E = e$, predictors $\mathbf{X}$ and the target variable Y have a joint distribution with underlying factorized causal mechanisms:

$$P(\mathbf{X}, Y|E = e) = \prod_{X_i \in \mathbf{X}} P(X_i|\mathrm{Pa}(X_i), e)\; P(Y|\mathrm{Pa}(Y), e) \tag{9}$$

For any conditional model $q(Y|\cdot)$, define the expected cross-entropy loss as follows:

$$\mathcal{L}_p \coloneqq \mathbb{E}_{(\mathbf{X},Y,E)\sim p}\big[-\log q(Y|\mathbf{X}, E)\big] \tag{10}$$

where p here denotes the joint distribution over $\mathbf{X}, Y, E$. In order to get the optimal solution for minimizing the expected cross-entropy loss, we can write the loss into two terms shown below:

$$\begin{aligned}
\mathcal{L}_q &= \sum_{\mathbf{x},y,e} p(\mathbf{x}, y, e)(-\log q(y|\mathbf{x}, e)) && (11)\\
&= \sum_{\mathbf{x},e} p(\mathbf{x}, e) \sum_{y} p(y|\mathbf{x}, e)(-\log q(y|\mathbf{x}, e)) && (12)\\
&= -\sum_{\mathbf{x},e} p(\mathbf{x}, e) \sum_{y} p(y|\mathbf{x}, e) \log p(y|\mathbf{x}, e) && (13)\\
&\quad + \sum_{\mathbf{x},e} p(\mathbf{x}, e) \sum_{y} p(y|\mathbf{x}, e) \log \frac{p(y|\mathbf{x}, e)}{q(y|\mathbf{x}, e)} && (14)\\
&= H(Y|\mathbf{X}, E) + E_{\mathbf{x},e}\left[D_{KL}(p(Y|\mathbf{X}, E)\|q(Y|\mathbf{X}, E))\right] && (15)
\end{aligned}$$

Since KL divergence is non-negative, we have:

$$\mathcal{L}_q \geq H(Y|\mathbf{X}, E) \tag{16}$$

with equality iff

$$\begin{aligned}
&E_{\mathbf{x},e}\left[D_{KL}(p(Y|\mathbf{X}, E)\|q(Y|\mathbf{X}, E))\right] = 0 && (17)\\
&\Rightarrow p(Y|\mathbf{x}, e) = q(Y|\mathbf{x}, e) \;\text{ for any } \mathbf{x}, e && (18)
\end{aligned}$$

Therefore, the optimal solution is:

$$q_e^*(Y|\mathbf{X}) \coloneqq p(Y|\mathbf{X}, e) \tag{19}$$

Under Causal Markov Condition, we have:

$$Y \perp\!\!\!\perp \mathbf{X} \setminus \mathrm{MB}(Y) \,|\mathrm{MB}(Y) \tag{20}$$

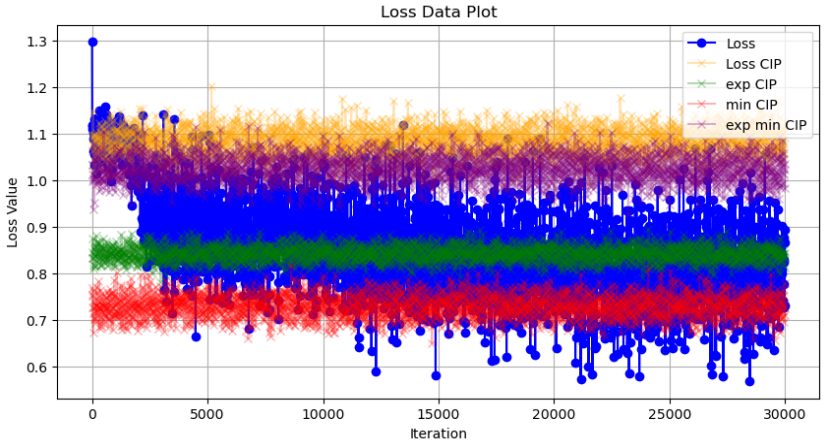
Therefore we have:

$$q_e^*(Y|\mathbf{X}) \coloneqq p(Y|\mathrm{MB}, e) \tag{21}$$

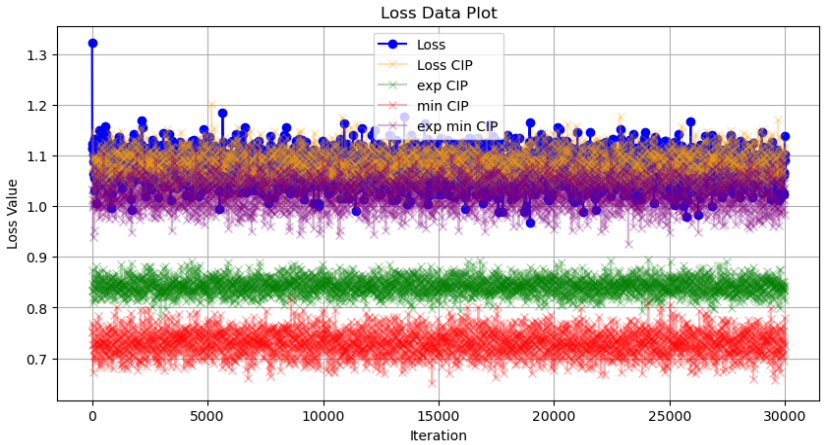
□

B Training Loss

Fig. 3 shows the training loss for models trained with and without positional embeddings, while keeping all other settings identical.



(a) Training loss with positional embeddings



(b) Training loss without positional embeddings

Figure 3: The blue curve shows the training loss over 30,000 optimization steps for two-layer Transformers trained on the ICL next-token prediction task, with and without positional embeddings. The model with positional embeddings exhibits faster convergence and reaches a lower training loss, suggesting that positional information plays an important role in optimization and predictive performance.

Below we explain the meaning of each legend entry in Fig. 3.

$$\text{Loss} = -\sum_k \Big[y_k^e \log(\hat{p}_k^e) + (1 - y_k^e) \log(1 - \hat{p}_k^e) \Big], \tag{22}$$

$$\text{Loss CIP} = -\sum_k \Big[y_k^e \log(p_k^e) + (1 - y_k^e) \log(1 - p_k^e) \Big], \tag{23}$$

$$\text{exp CIP} = -\sum_k \Big[p_k^e \log(p_k^e) + (1 - p_k^e) \log(1 - p_k^e) \Big], \tag{24}$$

$$\text{min CIP} = -\sum_k \Big[p_k'^e \log(p_k'^e) + (1 - p_k'^e) \log(1 - p_k'^e) \Big], \tag{25}$$

$$\text{exp min CIP} = -\sum_k \Big[p_k'^e \log(wp_k') + (1 - p_k'^e) \log(1 - wp_k') \Big]. \tag{26}$$

Here, $y_k^e \in \{0, 1, \ldots\}$ denotes the true value in test prompt $k \in [K]$ drawn from environment $e \in [E]$.

For each prompt k associated with its environment index e,

- $\hat{p}_k^e$ denotes the estimated conditional probability produced by the trained model for that environment;
- p_k^e denotes the true conditional probability given the true parents of the target variable in environment e;
- $p_k'^e$ denotes the true conditional probability given all other observed variables in environment e;
- wp_k' denotes the weighted conditional probability given all other observed variables. This quantity is computed from a weighted conditional probability table (CPT). In particular, for one batch of prompts, there is a single weighted CPT used to obtain $wp_k'^e$. In contrast, the quantities p_k^e and $p_k'^e$ are computed from environment-specific CPTs, so for each batch, there are multiple such CPTs, one for each environment.

In all cases, the quantities y_k^e, $\hat{p}_k^e$, p_k^e, and $p_k'^e$ correspond to the same environment e from which prompt k is drawn.

C Algorithm

Algorithm 1 Local Parent Discovery via Positional Embedding Dynamics

Input: Target variable Y, variables $\mathbf{X}$, training horizon T, early-training window T_{I}, threshold $\tau > 0$

Output: Estimated parent set $\widehat{\mathrm{Pa}}(Y)$

Train the two-layer Transformer on the next-token prediction task with target Y up to step T.

Record the positional embedding weights $W_{\mathrm{pe}}(t)$ over training.

for all $X_j \in \mathbf{X}$ **do**

Compute the aggregated positional embedding magnitude

$$s_j(t) \leftarrow \big\|W_{\mathrm{pe}}(X_j, t)\big\|, \qquad t \in [0, T_{\mathrm{I}}],$$

where the norm is taken after aggregating over embedding dimensions.

Define the estimated parent set as

$$\widehat{\mathrm{Pa}}(Y) \leftarrow \Big\{X_j \in \mathbf{X} : s_j(t) > \tau,\ t \in [T_{\mathrm{I}}]\Big\}.$$