Fine-Tuning Dynamics of In-Context Factual Recall in Transformers

Ruomin Huang
ruomin.huang@duke.edu
Duke University

Eshaan Nichani
eshnich@princeton.edu
Princeton University

Jason D. Lee
jasondlee@berkeley.edu
UC Berkeley

Rong Ge
rongge@cs.duke.edu
Duke University

Abstract

In-context learning – performing tasks based on examples given in the prompt – is an important capability that has emerged in large language models and has received significant attention in both theory and practice. Existing theoretical work often focuses on settings where the learning uses information purely from the prompt. However, many practical instances of in-context learning require the model to retrieve factual knowledge stored in the model's parameters, with the context serving to identify which knowledge is relevant. In this work, we study how in-context learning leverages factual knowledge recall. We formalize this behavior by introducing the *in-context factual recall (IC-recall)* task, where a transformer is provided a context of (subject, answer) pairs generated from a hidden relation, along with a query subject, and must both infer this hidden relation and retrieve the corresponding answer. Factual knowledge is modeled by the transformer having access to a simple pre-constructed MLP associative memory storing (subject, relation, answer) triplets. We analyze the supervised fine-tuning dynamics of a one-layer transformer on IC-recall data and prove that the model successfully performs IC-recall by converging to a particular pairwise attention pattern. This fine-tuning stage requires a very small number of samples – only polylogarithmic in the number of stored knowledge triplets. Experiments verify our theoretical predictions and show that the pairwise attention pattern emerges even when the MLP layer is pretrained instead of constructed.

1. Introduction

Transformer-based large language models (LLMs) exhibit strong *in-context learning* (ICL) abilities: they can use examples provided in the prompt to improve prediction on a new query (Brown et al., 2020). Considerable research has sought to understand the mechanisms underlying ICL, and most existing works focus on settings in which transformers rely only on the context to make predictions. One line of work studies ICL over function classes (Garg et al., 2022; von Oswald et al., 2023; Zhang et al., 2024), where transformers are trained from scratch on sequences of $(x, f(x))$ examples for a context-dependent function f, and learn to predict $f(x_q)$ for a new query x_q. Another line of work (Nichani et al., 2024; Edelman et al., 2024) studies how transformers learn to solve particular matching or copying tasks by converging to the induction head mechanism (Olsson et al., 2022). Although these are interesting settings to study ICL from a theoretical perspective, they do not capture an important behavior common in real-world LLM applications, where the model must combine contextual information with factual knowledge, or parametric knowledge (Petroni et al., 2019; Roberts et al., 2020; Cheng et al., 2024), stored in its weights. For example, given the prompt "Albert Einstein $\to$ Germany, Isaac Newton $\to$ England, Marie Curie $\to$?", the model must first use its stored knowledge to infer that the relation between subject and answer is nationality, then answer the query by retrieving the fact (again from stored knowledge) that Marie Curie was Polish.

In this paper, we formalize this behavior through the *in-context factual recall* (IC-recall) task. Motivated by (Petroni et al., 2019; Ghosal et al., 2024; Nichani et al., 2025), we model factual knowledge as $(\text{subject}, \text{relation}, \text{answer})$ triplets.

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

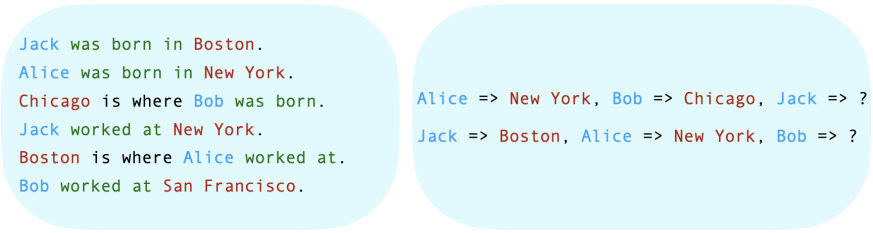


Figure 1. Left: factual knowledge corpus. Each sentence consists of three components: subject, relation and answer. We can put these components into the subject set $\mathcal{S} = \{\text{Jack, Alice, Bob}\}$, the relation set $\mathcal{R} = \{\text{was born in, worked at}\}$ and the answer set $\mathcal{A} = \{\text{Boston, New York, Chicago, San Francisco}\}$. We can view each relation $r \in \mathcal{R}$ as a mapping from $\mathcal{S}$ to $\mathcal{A}$. Right: IC-recall data requires the model to complete the sequence in a specific format: $(s_1, a_1, s_2, a_2, s_3, u_{\text{EoS}})$. Here u_{EoS} is the End-of-Sequence (EoS) token, $s_1, s_2 \in \mathcal{S}$ and $a_1 = r(s_1), a_2 = r(s_2)$ for some underlying relation r.

Each sequence in the IC-recall task consists of (subject, answer) pairs generated from the same relation, along with a query subject; the goal of the task is to output the answer corresponding to the query subject and hidden relation. Here, the context alone is insufficient to answer the query, as the model must also rely on the factual knowledge. See Figure 1 for an illustration of the task, and Section 2 for a formal definition.

1.1. Our Results

We consider fine-tuning a model which already stores factual knowledge in its parameters on sequences from the IC-recall task. Specifically, the model is a transformer with a single self-attention head followed by an MLP layer. We construct an MLP layer that acts as an associative memory (Nichani et al., 2025) storing (subject, relation, answer) triplets (see Section 3), and assumes that the MLP layer remains fixed throughout fine-tuning.

We fine-tune the model to access this relevant knowledge in two steps: it first infers the underlying relation r from the context, and then uses r to predict the answer. This two-step structure naturally aligns with the *chain-of-thought* (CoT) (Wei et al., 2022) paradigm, where an intermediate decoding step outputs r before producing the final answer.

In Section 4, we analyze the supervised fine-tuning dynamics of this one-layer transformer on IC-recall data with two in-context examples. We prove that the transformer successfully performs IC-recall by converging to a *pairwise attention* pattern – the transformer places equal attention weight to the tokens within each pair of in-context examples, but *different* attention weights across pairs. This identifies a mechanism for ICL which is qualitatively different from the induction head (Olsson et al., 2022). We also show that the number of fine-tuning samples required to achieve high test accuracy on the IC-recall task grows polylogarithmically with the number of stored factual associations.

Empirically, we show in Section 5 that as predicted by theory, fine-tuning only requires a very small number of sequences (8 often already achieves high accuracy). We also show that similar pairwise attention patterns extend to the cases with more than two in-context examples, and when the MLP layer is pretrained instead of constructed.

1.2. Related Works

Knowledge localization and associative memories. Prior work suggests that factual knowledge in language models can be stored in either MLP layers (Geva et al., 2021; Meng et al., 2022; Dai et al., 2022), though it may also be encoded in attention layers (Chen et al., 2025; Wei et al., 2024). Recent theory studies the ability of transformers to store factual knowledge via associative memories, showing that their storage capacity scales proportionally with model size (Allen-Zhu & Li, 2025; Nichani et al., 2025; Cabannes et al., 2024a). Ravfogel et al. (2026) further propose a construction in which the required model size depends only logarithmically on the number of subjects. Nichani et al. (2025) analyze the training dynamics of associative memory formation in one-layer linear transformers, and Cabannes et al. (2024b) study the training dynamics for linear layers.

Theoretical understanding of in-context learning. Garg et al. (2022) study the function classes that transformers can learn in context. von Oswald et al. (2023) show that transformers can learn linear functions in context by implicitly simulating gradient descent, and Zhang et al. (2024) later prove convergence in this setting. Olsson et al. (2022) identify induction heads as a mechanism underlying certain forms of in-context learning. Nichani et al. (2024); Edelman et al.

(2024) analyze how induction heads emerge via gradient-based training on in-context Markov chains. Chen et al. (2024) study how transformers learn a generalized induction head from in-context n-gram data. Bu et al. (2025) analyze factual-recall ICL under a data model that admits task-vector arithmetic, where answers are obtained by adding a retrieved task vector to the query representation. Among prior works, Vasudeva et al. (2026) study the most similar data-generation setting, which they call the contextual recall task. Their work assumes disjoint answer sets across relations, providing an additional signal for ruling out irrelevant answers using the in-context examples. In contrast, our setting allows answers to be shared across different relations.

Fine-tuning dynamics. Malladi et al. (2023); Ren & Sutherland (2025); Zeng et al. (2025) analyze fine-tuning dynamics through the neural tangent kernel (NTK) perspective. On related factual recall tasks, Ghosal et al. (2024) study the fine-tuning dynamics of a self-attention layer in which the value matrix W_V stores facts.

Benefit of CoT. Gekhman et al. (2026) empirically discover that chain-of-thought (CoT) improves factual recall. Merrill & Sabharwal (2024); Liu et al. (2024) show that chain-of-thought (CoT) reasoning increases the expressive power of transformers. Abbe et al. (2024) show that CoT can extend the learnable function class of transformers. In addition, Wen et al. (2025); Kim & Suzuki (2025) show that CoT substantially improves the sample efficiency of transformers for k-parity.

2. Preliminary

2.1. Transformer architecture

Vocabulary. Let the vocabulary be $\mathcal{V} = \mathcal{S} \cup \mathcal{R} \cup \mathcal{A} \cup \{u_{\text{EoS}}\}$, where $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{A}$ denote the sets of subjects, relations, and answers, respectively, and u_{EoS} denotes the end-of-sequence (EoS) token. We assume that $|\mathcal{S}| = |\mathcal{A}| =: n$, and that each $r \in \mathcal{R}$ is a bijection from $\mathcal{S}$ to $\mathcal{A}$. For each pair $(s, r) \in \mathcal{S} \times \mathcal{R}$, let $r(s) \in \mathcal{A}$ denote the associated answer. We also denote $\mathcal{R}(s, a)$ the set of relations that map s to a.

Embedding. All elements in vocabulary $u \in \mathcal{V}$ are embedded into $\phi(u) \in \mathbb{R}^d$. For simplicity, we assume that all embeddings are unit vectors and mutually orthogonal, i.e., $\langle \phi(u), \phi(u') \rangle = 0$ for all $u \neq u'$. We also use one-hot positional encodings. Specifically, for the i-th input token, we concatenate its embedding with the one-hot vector $e_i \in \mathbb{R}^{d_P}$, where d_P is the maximum input length. Thus, for an input sequence $\widetilde{Z} = (u_1, \ldots, u_k) \in \mathcal{V}^k$, we define the embedding matrix

$$Z = E(\widetilde{Z}) := \begin{bmatrix} e_1 & e_2 & \cdots & e_k \\ \phi(u_1) & \phi(u_2) & \cdots & \phi(u_k) \end{bmatrix} \in \mathbb{R}^{(d+d_P)\times k}.$$

Architecture. We consider a one-layer transformer $f(Z; \mathcal{W})$. Let $z_{-1} \in \mathbb{R}^{d+d_P}$ denote the last token in the input embedding matrix Z. The model output is given by

$$h = W^P \left(z_{-1} + Z \operatorname{softmax}\left(Z^\top W^{KQ} z_{-1} \right) \right),$$

and

$$f(Z; \mathcal{W}) = f_{\text{MLP}}(h) = V^\top \sigma(Wh),$$

where $W, V \in \mathbb{R}^{d_{\text{MLP}} \times d}$, d_{MLP} is the width of the MLP, $W^P = \begin{pmatrix} 0_{d \times d_P} & I_d \end{pmatrix}$ projects onto the token-embedding component only, and $\mathcal{W} = (W^{KQ}, W, V)$ are the set of parameters. We also assume that the activation in the MLP is the quadratic function $\sigma(x) = x^2$ applied element-wise.

Construction of the MLP associative memory. Transformers have been observed to memorize factual knowledge within the MLP layer. In practice, this memorization process typically occurs during the pretraining stage. However, to enable theoretical analysis, we will assume the MLP layer is pre-constructed to store a given knowledge set. In Section 5, we will observe empirically that pretrained MLPs behave similarly to our constructed MLP on the IC-recall task.

Consider a pretraining sequence $\widetilde{Z}_{\text{PT}} = (u_1, u_2)$ that contains two randomly selected elements u_1, u_2 from a valid fact triplet (s, r, a). We say that the MLP stores this fact triplet if the argmax decoding

$$\arg\max_{u \in \mathcal{V}} \phi(u)^\top f_{\text{MLP}}(\phi(u_1) + \phi(u_2))$$

recovers the remaining element of the triplet. We refer to an MLP layer which satisfies this property for all possible pretraining sequences as an *MLP associative memory*. As shown later in Section 3, such an associative memory admits a simple and natural construction. In the following, we assume that the MLP associative memory used in the *IC-recall task* is given by this construction.

2.2. IC-recall task

We next formally define the IC-recall task.

Definition 2.1 (IC-recall task)**.** Given subject, relation, answer sets $\mathcal{S}, \mathcal{R}, \mathcal{A}$ and a one-layer transformer $f(Z; \mathcal{W})$, whose MLP layer $f_{\mathrm{MLP}}(\cdot)$ is preconstructed as an MLP associative memory, the in-context factual recall task consists of data generated as follows:

1. Sample a relation $r^* \in \mathcal{R}$ uniformly at random.
2. Sample three distinct subjects $s_1, s_2, s_3 \in \mathcal{S}$ uniformly at random, and define $a_1 = r^*(s_1)$, $a_2 = r^*(s_2)$, and $a_3 = r^*(s_3)$.
3. Construct the IC-recall sequence $\widetilde{Z} = (s_1, a_1, s_2, a_2, s_3, u_{\mathrm{EoS}})$, with prediction target r^*, a_3.

We denote P_{IC} the distribution of such generated IC-recall sequence $\widetilde{Z}$.

To make sure it is possible to correctly solve the IC-recall task, we assume that any two subject–answer pairs identifies at most one relation.

Assumption 2.2. For any $s_1, s_2 \in \mathcal{S}$ and $a_1, a_2 \in \mathcal{A}$, $\left|\{\, r \in \mathcal{R} \mid r(s_1) = a_1,\ r(s_2) = a_2 \,\}\right| \le 1$.

The IC-recall task provides the true relation r^* as an intermediate step of the reasoning, which makes the problem easier. Without this form of strong supervision, the partially fine-tuned 1-layer transformer cannot achieve accuracy greater than $1/3$ (see Lemma B.1 in Appendix). Similar strong supervision was also considered in other works analyzing CoT (Wen et al., 2025; Kim & Suzuki, 2025). We also remark that without such intermediate supervision, transformers struggle to learn compositional reasoning tasks (Wang et al., 2025).

2.3. Training loss for the IC-recall task

We now define the loss used for fine-tuning. It is a chain-of-thought (CoT) objective in which the model decodes in two steps to produce the final answer. In the first step, the model predicts the relation token r associated with the input sequence; in the second step, it predicts the answer a_3. For simplicity of analysis, we assume that the "irrelevant" logits are masked out during prediction. That is, in the first decoding step, only the logits corresponding to relation tokens are retained. Likewise, in the second decoding step, only the logits corresponding to answer tokens are retained. Therefore given an IC-recall sequence $\widetilde{Z}$, we can write the prediction of the first decoding step as $p_1(\cdot; \mathcal{W}, \widetilde{Z}) := \mathrm{softmax}\left(A_r f(E(\widetilde{Z}); \mathcal{W})/T\right) \in \mathbb{R}^{|\mathcal{R}|}$ where $A_r \in \mathbb{R}^{|\mathcal{R}| \times d}$ is the embedding matrix of all relations and $T > 0$ is the prediction temperature. Similarly, conditioned on the first decoded relation being r, we write the prediction of the second decoding step as $p_{2,r}(\cdot; \mathcal{W}, \widetilde{Z}) := \mathrm{softmax}\left(A_a f(E([\widetilde{Z}, r]); \mathcal{W})/T\right) \in \mathbb{R}^{|\mathcal{A}|}$ where $A_a \in \mathbb{R}^{|\mathcal{A}| \times d}$ is the embedding matrix of all answers. We sample a fine-tuning dataset D from distribution P_{IC} and let

$$
\begin{aligned}
L_1(\mathcal{W}, D) &= \frac{1}{|D|} \sum_{\widetilde{Z} \in D} \ell_1(\mathcal{W}, \widetilde{Z}) \\
&:= -\frac{1}{|D|} \sum_{\widetilde{Z} \in D} \log\left(p_1(r^*(\widetilde{Z}); \mathcal{W}, \widetilde{Z})\right)
\end{aligned}
$$

denote the cross-entropy loss for the first decoding step, where $r^*(\widetilde{Z})$ is the underlying relation for the sequence $\widetilde{Z}$. The second decoding step uses the cross-entropy loss conditioned on the correct relation $r^*(\widetilde{Z})$.

$$L_2(\mathcal{W}, D) = \frac{1}{|D|}\sum_{\widetilde{Z}\in D} \ell_2(\mathcal{W}, \widetilde{Z})$$
$$:= -\frac{1}{|D|}\sum_{\widetilde{Z}\in D} \log\left(p_{2,r^*(\widetilde{Z})}(a_3(\widetilde{Z}); \mathcal{W}, \widetilde{Z})\right)$$

where $a_3(\widetilde{Z})$ is the correct answer for the sequence $\widetilde{Z}$. We use r^*, a_3 instead of $r^*(\widetilde{Z}), a_3(\widetilde{Z})$ when the sequence is clear from context. The CoT loss is the sum of these two losses

$$L(\mathcal{W}, D) = L_1(\mathcal{W}, D) + L_2(\mathcal{W}, D). \tag{1}$$

Partial fine-tuning. To simplify the theoretical analysis, we only fine-tune the position-position block $W^{KQ}_{1:d_P,\,1:d_P}$ of the attention matrix, while keeping the MLP associative memory and the rest of the entries of W_{KQ} fixed. There are only two vectors inside $W^{KQ}_{1:d_P,\,1:d_P}$ that are relevant for the two decoding steps. Let $\theta := W^{KQ}_{1:6,6} \in \mathbb{R}^6$ and $\omega := W^{KQ}_{1:7,7} \in \mathbb{R}^7$. Then $\mathrm{softmax}(\theta)$ and $\mathrm{softmax}(\omega)$ are the attention scores from the EoS token and the first decoded token respectively; the first decoding step depends solely on θ, and the second depends solely on ω. Therefore we can rewrite the loss as $L(\theta, \omega, D)$. In Section 5, we show empirically that even without this restriction, the transformer finds the same solution as our theory predicts.

3. MLP associative memory

In this section we provide a simple construction of the MLP layer with $d_{\mathrm{MLP}} = O(|\mathcal{S}| \cdot |\mathcal{A}|)$ that achieves 100% accuracy as an MLP associative memory. This serves as the frozen MLP in our theoretical analysis.

Lemma 3.1. *There exists an MLP layer $f_{\mathrm{MLP}}(x) = V^\top \sigma(Wx)$ with width $d_{\mathrm{MLP}} = O(|\mathcal{S}| \cdot |\mathcal{A}|)$, such that for any triplet (s, r, a) where r maps s to a, we have $a = \arg\max_{u\in\mathcal{V}} \phi(u)^\top f_{\mathrm{MLP}}(\phi(s) + \phi(r))$, $r = \arg\max_{u\in\mathcal{V}} \phi(u)^\top f_{\mathrm{MLP}}(\phi(a) + \phi(s))$ and $s = \arg\max_{u\in\mathcal{V}} \phi(u)^\top f_{\mathrm{MLP}}(\phi(a) + \phi(r))$.*

Proof sketch. Let $d_{\mathrm{MLP}} = 3|\mathcal{S}| \cdot |\mathcal{A}|$. Let w_j and v_j be the j-th row of W and V respectively. We assign indices to all (s, a) pairs. For the i-th (s, a) pair, we assign 3 rows in W and V to store the relevant factual knowledge. Specifically, let $w_{3i-2} = \phi(s) + \sum_{r\in\mathcal{R}(s,a)} \phi(r)$ and $v_{3i-2} = \phi(a)$; let $w_{3i-1} = \phi(a) + \phi(s)$ and $v_{3i-1} = \sum_{r\in\mathcal{R}(s,a)} \phi(r)$; let $w_{3i} = \phi(a) + \sum_{r\in\mathcal{R}(s,a)} \phi(r)$ and $v_{3i} = \phi(s)$ (we define $\sum_{r\in\mathcal{R}(s,a)} \phi(r) = 0$ if $\mathcal{R}(s, a)$ is empty). It is easy to examine that such constructed MLP satisfies the properties in Lemma 3.1; see the full proof in Appendix A.

Remark. If non-orthogonal embeddings are allowed, one can obtain a more parameter-efficient construction of MLP associative memory, with a total number of parameters nearly linear in $|\mathcal{S}| \cdot |\mathcal{R}|$, as shown by Nichani et al. (2025). For simplicity in the subsequent dynamics analysis, we stick to the MLP construction in Lemma 3.1, which is based on orthogonal embeddings.

4. Fine-tuning dynamics for IC-Recall

We show in this section that with the constructed MLP associative memory, transformers can achieve perfect accuracy on IC-recall data after fine-tuned by the perturbed gradient descent (PGD) in Algorithm 1. Specifically, we have the following Theorem:

Theorem 4.1. *Assuming that the IC-recall data satisfies Assumptions 2.2 and 4.3, there exist constants $C_1, C_2 > 0$ such that for any $T \le C_1/\log n$, if we set $\eta_1 = \Theta(T\sqrt{T}\log\frac{1}{T}), \eta_2 = \Theta(T^2), t_1 = 1$ and $t_2 \ge \frac{C_2}{T}\log\frac{1}{T}$ in Algorithm 1, with probability at least $1-\delta$ over the sampled data D and the random perturbation, the transformer fine-tuned on $\tilde{O}(\mathrm{poly}(\frac{1}{T}))$ [1] samples will have accuracy at least 0.99 on all IC-recall sequences the for final predicted answers.*

Note that Assumption 2.2 implies that it is possible to infer the hidden relation from only two in-context examples, and Assumption 4.3 (introduced later in Section 4.2) considers a particular difficult setting where the pairwise attention

[1] We use $\tilde{O}$ to hide $\mathrm{poly}\log\frac{1}{\delta}$ factors.

pattern is necessary to achieve good accuracy. This suggests that $O(\mathrm{polylog}(n))$ samples are sufficient for the transformer to learn how to use the stored knowledge to perform IC-recall, far fewer than seeing every possible subjects/answers.

Algorithm 1 Temperature-Scaled Perturbed Gradient Descent

Input: Decoding temperature $T > 0$, failure probability $\delta \in (0, 1)$, learning rates η_1, η_2, number of iterations t_1, t_2, fine-tuning dataset D.
Initialization: $(\theta, \omega) \leftarrow 0$.
Stage 1: Train (θ, ω) using gradient descent with step size η_1 over loss $L(\theta, \omega, D; T)$ for t_1 iterations to obtain $(\tilde{\theta}, \tilde{\omega})$.
Stage 2: Sample a random perturbation ξ from $\mathrm{Uniform}\left(\mathbb{B}\left(0,\ \Theta\left(T^3 \log^{-2}(\frac{1}{\delta})\right)\right)\right)$, and set perturbed initialization $(\theta_0, \omega_0) \leftarrow (\tilde{\theta}, \tilde{\omega}) + \xi$. Run gradient descent from (θ_0, ω_0) with step size η_2 over loss $L(\theta, \omega, D; T)$ for t_2 iterations.

The first decoding step that recovers the underlying relation is harder to analyze. Algorithm 1 is separated into two stages, where stage 1 is a single large gradient step and stage 2 is a perturbation followed by a few smaller gradient steps. We provide intuition on how the two stages work using a simple warm-up example with 3 subjects in Section 4.1. We then discuss the difficulty in extending such an analysis to the case when we do not see all subjects/answers in Section 4.2, and show how this can be overcome by lowering the temperature T in the decoding. Finally we complete the analysis of the second-step decoding in Section 4.3.

4.1. Warm up: a three-subject instance

We first consider the simple case where $|\mathcal{S}| = |\mathcal{A}| = 3$ and $|\mathcal{R}| = 6$. In this case, $\mathcal{R}$ consists all 6 bijections from $\mathcal{S}$ to $\mathcal{A}$. For an IC-recall sequence $(s_1, a_1, s_2, a_2, s_3, u_{\mathrm{EoS}})$, we denote by r_1 the underlying correct relation and $r_2, \ldots, r_6$ the remaining relations (see details in Table 1). The goal is to predict the correct relation r_1. We call the relation r_2 the confusing relation, and $r_3, ..., r_6$ which assign s_3 to one of a_1, a_2 the mismatched relations. It is worth noting that, in this three-subject setting, the gradients satisfy $\nabla_\theta \ell_1(\theta, Z_1) = \nabla_\theta \ell_1(\theta, Z_2)$ for any two IC-recall sequences Z_1 and Z_2. Consequently, the empirical loss is equivalent to the population loss, and it suffices to analyze the dynamics on a single fine-tuning sequence. As we will see later, even in this simple setting, transformers still learn a non-trivial solution and exhibit a two-stage fine-tuning dynamics.

Table 1. The six bijections $r_1, \ldots, r_6$ from $\{s_1, s_2, s_3\}$ to $\{a_1, a_2, a_3\}$.

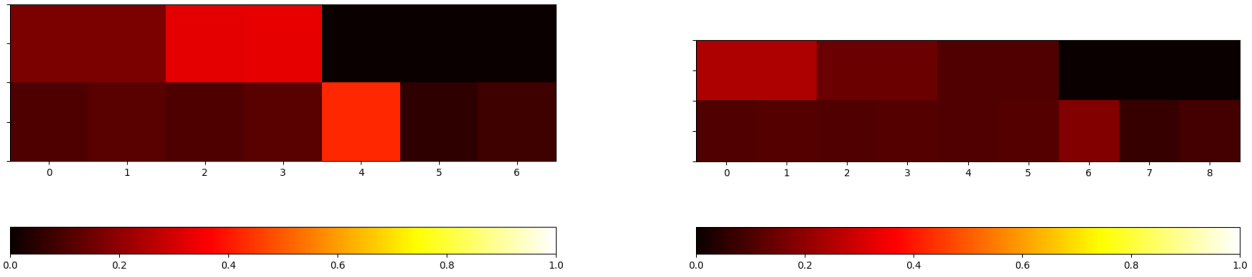
	s_1	s_2	s_3	type
r_1	a_1	a_2	a_3	correct
r_2	a_2	a_1	a_3	confusing
r_3	a_1	a_3	a_2	mismatched
r_4	a_2	a_3	a_1	mismatched
r_5	a_3	a_1	a_2	mismatched
r_6	a_3	a_2	a_1	mismatched

Let $v := \mathrm{softmax}(\theta)$ be the attention vector from the EoS token. We will show that in Algorithm 1, for a sufficiently small constant temperature T, if we set $\eta_1 = \Theta(T^2), \eta_2 = \Theta(T^2), t_1 = t_2 = \infty$ so that Stage 1 and Stage 2 converge, then the fine-tuned transformer will converge to a global optimum where v exhibits a pairwise pattern. Formally, we show the following Theorem.

Theorem 4.2. *There exists constant $T_0(\delta) > 0$ that only depends on δ, such that for any $T < T_0$, if we set $\eta_1 = \Theta(T^2), \eta_2 = \Theta(T^2), t_1 = t_2 = \infty$ in Algorithm 1, then with probability at least $1 - \delta$ the transformer trained by Algorithm 1 will converge to the global optimum θ^* for $L_1(\theta; T)$. At the convergence the attention scores $v^* = (a, a, \frac{1}{2} - a, \frac{1}{2} - a, 0, 0)$ with some $a \neq \frac{1}{4}$.*

The proof of Theorem 4.2 is based on a loss-landscape analysis and is deferred to Appendix B. The convergence happens in two stages, and we provide a sketch of each stage below..

First stage: convergence to a saddle point. In the first stage, the attention scores v converge to the saddle point $\tilde{v} = (1/4, 1/4, 1/4, 1/4, 0, 0)$. The transformer learns that s_3 and the EoS tokens are irrelevant in predicting the



(a) Fine-tuned on $(s_1, a_1, s_2, a_2, s_3, u_{\text{EoS}})$ *(b)* Fine-tuned on $(s_1, a_1, s_2, a_2, s_3, a_3, s_4, u_{\text{EoS}})$

Figure 2. Pairwise attention at convergence for predicting the relation token in the first row. The first row is the attention scores from EoS token (first decoding step), and the second row is the attention scores from the decoded relation r^* (second decoding step). We set $|\mathcal{S}| = |\mathcal{A}| = 8$ and fix MLP as the construction in Lemma 3.1 for both experiments. Left is the 2-ICL-example input with $|\mathcal{R}| = 64, T = 0.05$ and right is 3-ICL-example input with $|\mathcal{R}| = 512, T = 0.01$.

relation. The uniform attention over s_1, a_1, s_2, a_2 arises from the *symmetry* between the two pairs (s_1, a_1) and (s_2, a_2), together with the symmetric initialization. During this stage, v_5 – the attention score on the query subject s_3 – decreases, which makes the prediction logits of mismatched relations $r_3, \ldots, r_6$ significantly smaller than those of r_1 and r_2 (see Table 2). By the end of this stage, the prediction of the transformer is uniform on r_1 (the correct relation) and r_2 (the confusing relation), but it cannot distinguish them, since v remains symmetric across the two subject-answer pairs.

Second stage: escape the saddle. At the beginning of the second stage, a random perturbation is added. This perturbation breaks the symmetry between the two subject-answer pairs, so that after a few steps of gradient descent, the model escapes from the saddle point with high probability (Jin et al., 2017). After escaping the saddle, the transformer begins to assign a higher prediction probability to r_1 than to r_2. Notably, a pairwise attention pattern emerges: the ratios v_1/v_2 and v_3/v_4 converge to 1, while the corresponding limiting attention scores are different. See Figure 2 for an illustration of this pattern. Denote by $l := A_r f(Z; \theta)$ the logit vector of the prediction for the first decoding step, where A_r is the embedding matrix for the the relations and the logit is a quadratic function [2] of the attention scores v, as shown in Table 2 in the Appendix.. The logit gap between r_1 and r_2 satisfies $l(r_1) - l(r_2) \propto (v_1 - v_3)(v_2 - v_4)$. Hence, the transformer ensures that $v_1 - v_3$ and $v_2 - v_4$ have the same sign to make this gap positive. Moreover, by the AM–GM inequality, replacing both v_1, v_2 by their mean $(v_1 + v_2)/2$ and v_3, v_4 by their mean $(v_3 + v_4)/2$ can only increase this gap or leave it unchanged. Therefore v_1 and v_2 will be driven together, and likewise for v_3 and v_4, thus explaining the pairwise attention pattern. Finally, we remark that even though our analysis is for two in-context examples, the pairwise attention emerges empirically when there are more in-context examples (see Figure 2 (b)).

4.2. Analysis on the first decoding step

In this section, we consider the general case of $|\mathcal{S}| = |\mathcal{A}| = n \geq 3$. Now, the empirical loss is no longer equivalent to the population loss. In particular, there may exist two subject–answer pairs that are not jointly matched by any relation, which breaks the symmetry among IC-recall sequences Z. For an input sequence $Z = (s_1, a_1, s_2, a_2, s_3, u_{\text{EoS}})$, we call a relation r a *2-matching relation* if r maps two of the three subjects $\{s_1, s_2, s_3\}$ to the two answers $\{a_1, a_2\}$. We call r a *confusing relation* (similar to r_2 in previous section) for Z if $r(s_1) = a_2$ and $r(s_2) = a_1$. At the saddle point after stage 1, the model cannot distinguish the confusing relation and the correct relation. Similarly, we call a 2-matching relation r a *mismatched relation* for Z if $r(s_3) = a_1$ or $r(s_3) = a_2$. These correspond to the relations r_3, r_4, r_5, r_6 in the simple example in previous subsection. Finally, we call a sequence confusing/mismatched if it has at least one confusing/mismatched relation. Notice that a sequence might be both confusing and mismatched (in fact, all sequences in the simple example are both confusing and mismatched).

Now that we are working with the empirical loss, we will show that fine-tuning the transformer on $O(\text{polylog}(n))$ random samples can still achieve over 0.999 test accuracy on predicting the underlying relation r in the first decoding step. We denote p_{conf} the fraction of confusing sequences and p_{mis} the fraction of mismatched sequences in D. We assume that each of these two sequence types occupies at least a constant fraction of the IC-recall data distribution.

Assumption 4.3. There exists a constant $0 < \zeta \leq 1$, such that the probability a random IC-recall sequence is a

[2] This is because the MLP layer uses the quadratic activation.

confusing sequence is at least ζ, and the probability it is a mismatched sequence is also at least ζ.

Note that if no input sequences are confusing, the problem is easier in the sense that the saddle point that stage 1 converges to will also achieve 100% accuracy. If there are no confusing sequences and no mismatched sequences, then even the initial solution would have 100% accuracy. Therefore this assumption ensures we are considering the hardest case.

The loss-landscape analysis in Section 4.1 relies on the symmetry of the loss in the pairs v_1, v_3 and v_2, v_4. This symmetry is difficult to generalize as the empirical loss is not symmetric. Another difficulty is that the attention pairing in Stage 2 relies on v_5 converging to 0, which is requires many steps and is in fact unnecessary for achieving near-optimal performance. Therefore, in this section we instead analyze the low-temperature dynamics.

With a lower temperature T, any improvement on the prediction logits will be amplified by $1/T$, which accelerates convergence. It therefore suffices to take only a single gradient descent step with a larger step size in Stage 1 and $O\left(\frac{1}{T}\log\frac{1}{T}\right)$ gradient descent steps in Stage 2. We show that this fine-tuning procedure also reaches a near-optimal solution. It is worth noting that, because the temperature is much lower, the near-saddle point reached after Stage 1 is closer to $(\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6})$ than to $(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},0,0)$, and the final solution does not exhibit the exact pairwise attention pattern (yet it still attends more to one pair compared to the other). Nevertheless, the resulting solution remains near-optimal in terms of accuracy. This property allows the training dynamics to be constrained in a local region near its initialization, and avoids difficult boundary cases when the attention to any of the positions gets near 0. Altogether, we show the following theorem, with proof deferred to Appendix C.

Theorem 4.4. *Assuming the IC-recall data satisfies Assumptions 2.2 and 4.3, there exist constants $C_1, C_2 > 0$, such that for any $T \le C_1/\log n$, if we set $\eta_1 = \Theta(T\sqrt{T}\log\frac{1}{T}), \eta_2 = \Theta(T^2), t_1 = 1$ and $t_2 \ge \frac{C_2}{T}\log\frac{1}{T}$ in Algorithm 1, then with probability at least $1-\delta$ over samples D and the random perturbation, transformers fine-tuned by Algorithm 1 on $\tilde{O}\left(\mathrm{poly}\left(\frac{1}{T}\right)\right)$ samples will have accuracy at least 0.999 for the first decoding step on all IC-recall sequences.*

4.3. Analysis on the second decoding step

The second decoding step is easier to analyze than the first decoding step. Intuitively, the transformer only needs to realize that attending to s_3, and combining it with the correct relation from the residual connection, is sufficient to produce the correct answer a_3. We show that this already happens at the end of Stage 1 in Lemma 4.5. The proof is deferred to Appendix D.

Lemma 4.5. *Assume the IC-recall data satisfies Assumptions 2.2 and 4.3. Set T, η_1, η_2 and the sample size $|D|$ same as in Theorem 4.1 for Algorithm 1, after Stage 1 and throughout Stage 2, the transformer has accuracy $1-o\left(\exp\left(-\frac{1}{\sqrt{T}}\right)\right)$ on all IC-recall sequences for the second decoding step.*

Theorem 4.1 follows immediately by combining Theorem 4.4 with Lemma 4.5.

5. Experiments

Experiment setup. Our model is a single-head attention layer followed by an MLP layer. We train the transformer using the Adam optimizer with learning rate 10^{-3}. For all experiments, we fix the size n of subjects $\mathcal{S}$ and answers $\mathcal{A}$, then we generate relations $\mathcal{R}$ as random bijections from $\mathcal{S}$ to $\mathcal{A}$. All subjects, answers, relations and EoS tokens use fixed random orthonormal embeddings.

Figure 3 reports the test accuracies for the first decoding step on the IC-recall data. We can see that transformers can learn the IC-recall task with only 8 samples in the experiment. The second decoding step consistently achieves perfect test accuracy across all hyperparameter settings and random seeds.

Experiments with pretrained MLP. We also conduct experiments with a pretrained MLP. Pretraining data $\widetilde{Z}_{\mathrm{PT}} = (u_1, u_2)$ is generated as follows:

1. Sample a random subject s from $\mathcal{S}$ and a random relation r from $\mathcal{R}$. Let $a = r(s)$ and form a fact triplet (s, r, a).
2. Randomly choose two elements from this triplet and place them at positions u_1 and u_2. The task is to predict the remaining third element from the input sequence $\widetilde{Z}_{\mathrm{PT}}$.

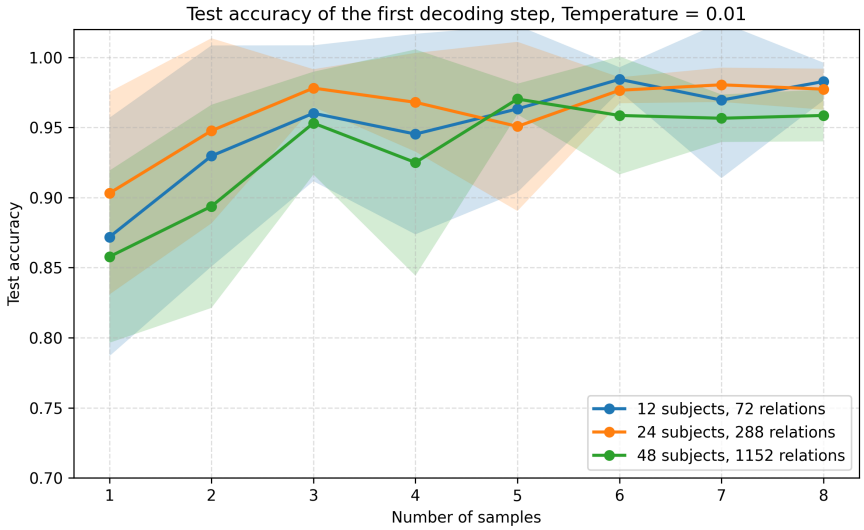
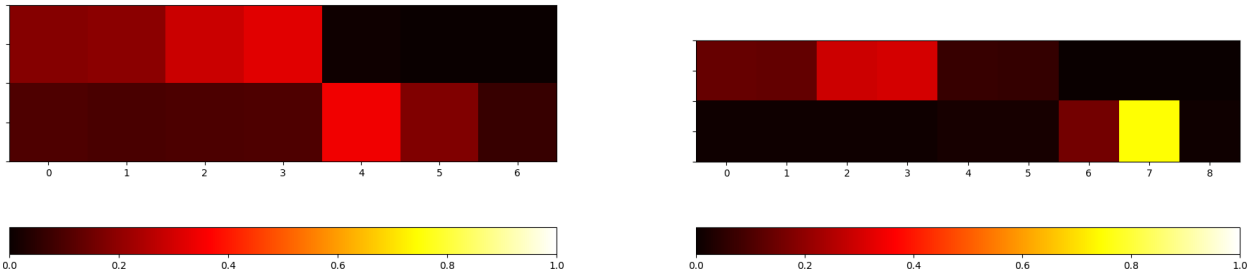


Figure 3. Test accuracies for fine-tuning on the first decoding step. The number of answers is equal to the number of subjects $|\mathcal{S}|$. The MLP is preconstructed and fixed during fine-tuning. We report the mean accuracies and standard deviations of 10 random seeds.



(a) Fine-tuned on $(s_1, a_1, s_2, a_2, s_3, u_{\text{EoS}})$ *(b)* Fine-tuned on $(s_1, a_1, s_2, a_2, s_3, a_3, s_4, u_{\text{EoS}})$

Figure 4. The pairwise attention pattern generalizes to pretraining the MLP. We consider pretraining the whole transformer and fine-tuning the whole attention layer. Left is the 2-ICL-example input with $|\mathcal{S}| = |\mathcal{A}| = 8, |\mathcal{R}| = 64, T = 0.05$ and right is the 3-ICL-example input with $|\mathcal{S}| = |\mathcal{A}| = 8, |\mathcal{R}| = 512, T = 0.01$.

During pretraining, the attention parameters W^{KQ} and the MLP are both unfrozen. Then during fine-tuning, we freeze the MLP layer and train W^{KQ}. Such fine-tuned transformer also exhibits the pairwise attention pattern. See Figure 4 for illustrations of converged attention scores. It is worth noting that attention weight for the second decoding step in Figure 4 is also assigned to the EoS token, which is different from Figure 2. As the EoS token does not contain any information, we believe the transformer uses it as a way to balance the coefficient between the attention to s_3 and the residual connection to r^*.

6. Conclusion

We introduced the IC-recall task to study in-context learning when a transformer must infer a hidden relation from context and then use it to retrieve the correct answer from memory. We constructed an MLP associative memory for fact triplets and analyzed the fine-tuning dynamics of a one-layer transformer with the MLP frozen. We showed that fine-tuning is able to recover the hidden relation by converging to a pairwise attention pattern in the minimal setting and requiring only polylogarithmically many fine-tuning samples in the general setting. The main limitation of our work is that the analysis relies on a specially constructed MLP and a simplified architecture. Extending these results to general pretrained transformers and end-to-end training remains future work.

References

Abbe, E., Bengio, S., Lotfi, A., Sandon, C., and Saremi, O. How far can transformers reason? the globality barrier and inductive scratchpad. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/3107e4bdb658c79053d7ef59cbc804dd-Abstract-Conference.html.

Allen-Zhu, Z. and Li, Y. Physics of language models: Part 3.3, knowledge capacity scaling laws. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=FxNNiUgtfa.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Bu, D., Huang, W., Han, A., Nitanda, A., Zhang, Q., Wong, H., and Suzuki, T. Provable in-context vector arithmetic via retrieving task concepts. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/bu25a.html.

Cabannes, V., Dohmatob, E., and Bietti, A. Scaling laws for associative memories. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=Tzh6xAJSll.

Cabannes, V., Simsek, B., and Bietti, A. Learning associative memories with gradient descent. In Salakhutdinov, R., Kolter, Z., Heller, K. A., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 5114–5134. PMLR / OpenReview.net, 2024b. URL https://proceedings.mlr.press/v235/cabannes24a.html.

Chen, S., Sheen, H., Wang, T., and Yang, Z. Unveiling induction heads: Provable training dynamics and feature learning in transformers. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/7aae9e3ec211249e05bd07271a6b1441-Abstract-Conference.html.

Chen, Y., Cao, P., Chen, Y., Liu, K., and Zhao, J. Knowledge localization: Mission not accomplished? enter query localization! In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=tfyHbvFZ0K.

Cheng, S., Pan, L., Yin, X., Wang, X., and Wang, W. Y. Understanding the interplay between parametric and contextual knowledge for large language models. *CoRR*, abs/2410.08414, 2024. doi: 10.48550/ARXIV.2410.08414. URL https://doi.org/10.48550/arXiv.2410.08414.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. Knowledge neurons in pretrained transformers. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 8493–8502. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.581. URL https://doi.org/10.18653/v1/2022.acl-long.581.

Edelman, E., Tsilivis, N., Edelman, B. L., Malach, E., and Goel, S. The evolution of statistical induction heads: In-context learning markov chains. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/75b0edb869e2cd509d64d0e8ff446bc1-Abstract-Conference.html.

Garg, S., Tsipras, D., Liang, P., and Valiant, G. What can transformers learn in-context? A case study of simple function classes. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/c529dba08a146ea8d6cf715ae8930cbe-Abstract-Conference.html.

Gekhman, Z., Aharoni, R., Ofek, E., Geva, M., Reichart, R., and Herzig, J. Thinking to recall: How reasoning unlocks parametric knowledge in llms. *CoRR*, abs/2603.09906, 2026. doi: 10.48550/ARXIV.2603.09906. URL https://doi.org/10.48550/arXiv.2603.09906.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In Moens, M., Huang, X., Specia, L., and Yih, S. W. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.446. URL https://doi.org/10.18653/v1/2021.emnlp-main.446.

Ghosal, G. R., Hashimoto, T., and Raghunathan, A. Understanding finetuning for factual knowledge extraction. In Salakhutdinov, R., Kolter, Z., Heller, K. A., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 15540–15558. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/ghosal24a.html.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1724–1732. PMLR, 2017. URL http://proceedings.mlr.press/v70/jin17a.html.

Kim, J. and Suzuki, T. Transformers provably solve parity efficiently with chain of thought. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=n2NidsYDop.

Liu, Z., Liu, H., Zhou, D., and Ma, T. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3EWTEy9MTM.

Malladi, S., Wettig, A., Yu, D., Chen, D., and Arora, S. A kernel-based view of language model fine-tuning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pp. 23610–23641. PMLR, 2023. URL https://proceedings.mlr.press/v202/malladi23a.html.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in GPT. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Merrill, W. and Sabharwal, A. The expressive power of transformers with chain of thought. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=NjNGlPh8Wh.

Nichani, E., Damian, A., and Lee, J. D. How transformers learn causal structure with gradient descent. In Salakhutdinov, R., Kolter, Z., Heller, K. A., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 38018–38070. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/nichani24a.html.

Nichani, E., Lee, J. D., and Bietti, A. Understanding factual recall in transformers via associative memories. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=hwSmPOAmhk.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. In-context learning and induction heads. *CoRR*, abs/2209.11895, 2022. doi: 10.48550/ARXIV.2209.11895. URL https://doi.org/10.48550/arXiv.2209.11895.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. H. Language models as knowledge bases? In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2463–2473. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1250. URL https://doi.org/10.18653/v1/D19-1250.

Ravfogel, S., Yehudai, G., Bruna, J., and Bietti, A. Geometric factual recall in transformers, 2026. URL https://arxiv.org/abs/2605.12426.

Ren, Y. and Sutherland, D. J. Learning dynamics of LLM finetuning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=tPNHOoZFl9.

Roberts, A., Raffel, C., and Shazeer, N. How much knowledge can you pack into the parameters of a language model? In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 5418–5426. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.437. URL https://doi.org/10.18653/v1/2020.emnlp-main.437.

Vasudeva, B., Deora, P., Bietti, A., Sharan, V., and Thrampoulidis, C. Understanding contextual recall in transformers: How finetuning enables in-context reasoning over pretraining knowledge. *CoRR*, abs/2603.20969, 2026. doi: 10.48550/ARXIV.2603.20969. URL https://doi.org/10.48550/arXiv.2603.20969.

von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pp. 35151–35174. PMLR, 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

Wang, Z., Nichani, E., Bietti, A., Damian, A., Hsu, D., Lee, J. D., and Wu, D. Learning compositional functions with transformers from easy-to-hard data. In Haghtalab, N. and Moitra, A. (eds.), *The Thirty Eighth Annual Conference on Learning Theory, 30-4 July 2025, Lyon, France*, Proceedings of Machine Learning Research, pp. 5632–5711. PMLR, 2025. URL https://proceedings.mlr.press/v291/wang25a.html.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Wei, Y., Yu, X., Weng, Y., Ma, H., Zhang, Y., Zhao, J., and Liu, K. Does knowledge localization hold true? surprising differences between entity and relation perspectives in language models. In Serra, E. and Spezzano, F. (eds.), *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, pp. 4118–4122. ACM, 2024. doi: 10.1145/3627673.3679900. URL https://doi.org/10.1145/3627673.3679900.

Wen, K., Zhang, H., Lin, H., and Zhang, J. From sparse dependence to sparse attention: Unveiling how chain-of-thought enhances transformer sample efficiency. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=AmEgWDhmTr.

Zeng, X., Wang, H., Lin, J., Wu, J., Cody, T., and Zhou, D. Lensllm: Unveiling fine-tuning dynamics for LLM selection. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/zeng25g.html.

Zhang, R., Frei, S., and Bartlett, P. L. Trained transformers learn linear models in-context. *J. Mach. Learn. Res.*, 25: 49:1–49:55, 2024. URL https://jmlr.org/papers/v25/23-1042.html.

A. Construction of MLP associative memory

We show that there exists a construction of the MLP layer with $d_{\mathrm{MLP}} = O(|\mathcal{S}| \cdot |\mathcal{A}|)$ that achieves 100% accuracy as an MLP associative memory.

Lemma A.1. *There exists an MLP layer $f_{\mathrm{MLP}}(x) = V^\top \sigma(Wx)$ with width $d_{\mathrm{MLP}} = O(|\mathcal{S}| \cdot |\mathcal{A}|)$, such that for any triplet (s, r, a) where r maps s to a, we have $a = \arg\max_{u \in \mathcal{V}} \phi(u)^\top f_{\mathrm{MLP}}(\phi(s) + \phi(r))$, $r = \arg\max_{u \in \mathcal{V}} \phi(u)^\top f_{\mathrm{MLP}}(\phi(a) + \phi(s))$ and $s = \arg\max_{u \in \mathcal{V}} \phi(u)^\top f_{\mathrm{MLP}}(\phi(a) + \phi(r))$.*

Proof. Let $d_{\mathrm{MLP}} = 3|\mathcal{S}| \cdot |\mathcal{A}|$. Let w_j and v_j be the j-th row of W and V respectively. We assign indices to all (s, a) pairs. For i-th (s, a) pair, we assign 3 rows in W and V to store the relevant factual knowledge. Specifically, let $w_{3i-2} = \phi(s) + \sum_{r \in \mathcal{R}(s,a)} \phi(r)$ and $v_{3i-2} = \phi(a)$; let $w_{3i-1} = \phi(a) + \phi(s)$ and $v_{3i-1} = \sum_{r \in \mathcal{R}(s,a)} \phi(r)$; let $w_{3i} = \phi(a) + \sum_{r \in \mathcal{R}(s,a)} \phi(r)$ and $v_{3i} = \phi(s)$. Here we define $\sum_{r \in \mathcal{R}(s,a)} \phi(r) = 0$ if $\mathcal{R}(s, a)$ is empty. Fix a (s, a) pair such that $\mathcal{R}(s, a)$ is non-empty and we examine the properties of MLP associative memory as follows.

For any $r \in \mathcal{R}(s, a)$, since r is an injection, we know s is the only subject that r maps to a. Therefore we can examine that $\phi(a)^\top f_{\mathrm{MLP}}(\phi(s) + \phi(r)) = 4$. For any $a' \neq a$, we know a' can only be mapped from s through relations that are not r. Therefore we have $\phi(a')^\top f_{\mathrm{MLP}}(\phi(s) + \phi(r)) = 1$. For any subject s', we have $\phi(s')^\top f_{\mathrm{MLP}}(\phi(s) + \phi(r)) = 1$. For any relation r', we have $\phi(r')^\top f_{\mathrm{MLP}}(\phi(s) + \phi(r)) = 1$. Therefore the constructed MLP can correctly predict a from $\phi(s) + \phi(r)$ using argmax decoding.

For any $r \in \mathcal{R}(s, a)$, we know a is the only answer that s is mapped to through r. Therefore we have $\phi(s)^\top f_{\mathrm{MLP}}(\phi(a) + \phi(r)) = 4$. For any $s' \neq s$, we know a' can only be mapped from s through relations that are not r. Therefore we have $\phi(s')^\top f_{\mathrm{MLP}}(\phi(a) + \phi(r)) = 1$. For any answer a', we have $\phi(a')^\top f_{\mathrm{MLP}}(\phi(a) + \phi(r)) = 1$ if the preimage of a' under r exists; otherwise $\phi(a')^\top f_{\mathrm{MLP}}(\phi(a) + \phi(r)) = 0$. For any relation r', we have $\phi(r')^\top f_{\mathrm{MLP}}(\phi(a) + \phi(r)) = 1$ if the preimage of a under r' exists; otherwise $\phi(r')^\top f_{\mathrm{MLP}}(\phi(a) + \phi(r)) = 0$. Therefore the constructed MLP can correctly predict s from $\phi(a) + \phi(r)$ using argmax decoding.

For any $r \in \mathcal{R}(s, a)$, we know $\phi(r)^\top f_{\mathrm{MLP}}(\phi(s) + \phi(a)) = 4$. For any $r' \notin \mathcal{R}(s, a)$, we know r' maps s to a different answer a' and the preimage of a is a different subject s' (if any). Therefore we have $\phi(r')^\top f_{\mathrm{MLP}}(\phi(s) + \phi(a)) \leq 2$. For any answer a', we have $\phi(a')^\top f_{\mathrm{MLP}}(\phi(s) + \phi(a)) = 1$. For any subject s', we have $\phi(s')^\top f_{\mathrm{MLP}}(\phi(s) + \phi(a)) = 1$. Therefore the constructed MLP can correctly predict a relation in $\mathcal{R}(s, a)$ from $\phi(s) + \phi(a)$ using argmax decoding. □

B. Proof to the normal temperature convergence for $n = 3$

We first show that under a normal temperature T, with infinite amount of time, transformer trained by PGD can converge to the global optimal solution. For the first decoding step training, we will show Algorithm 1 converges to the global optimal solution θ^* with small enough temperature T. At the convergence, the attention scores $v^* = \mathrm{softmax}(\theta^*) = (a, a, \frac{1}{2} - a, \frac{1}{2} - a, 0, 0)$ with some $a \neq \frac{1}{4}$. We write out the logits w.r.t. the attention scores v in Table 2. We can immediately obtain the following Lemma, which states that the partially fine-tuned transformer cannot predict the answer with one decoding step.

Lemma B.1. *Assume $|\mathcal{S}| = |\mathcal{A}| = 3$ and $\mathcal{R}$ consists of all 6 bijections from $\mathcal{S}$ to $\mathcal{A}$. Asumme the MLP layer f_{MLP} of the transformer $f(\cdot; W^{KQ})$ is fixed to be the construction in Lemma A.1. If we use the first decoding step prediction to predict from answers a_1, a_2, a_3 without the intermediate relation, then the prediction accuracy cannot exceed $1/3$ for any input IC-recall data $\widetilde{Z} = (s_1, a_1, s_2, a_2, s_3, u_{\mathrm{EoS}})$ and any W^{KQ}.*

Proof. By the logits in Table 2, we can see that the logits $l(a_1) = l(a_2) = l(a_3)$ is always true for the first decoding step. Therefore we have $p_1(a_1; W^{KQ}, \widetilde{Z}) = p_1(a_2; W^{KQ}, \widetilde{Z}) = p_1(a_3; W^{KQ}, \widetilde{Z})$ for any IC-recall data $\widetilde{Z} = (s_1, a_1, s_2, a_2, s_3, u_{\mathrm{EoS}})$ and any W^{KQ}. The Lemma follows immediately. □

Before the proof of Theorem 4.2, we first give the smoothness constant of the loss $L_1(\theta, T)$. Importantly through the gradient Lipschitz constant, we know that the learning rate η_1 and η_2 chosen in Theorem 4.2 can be small enough so that the descent lemma holds. The proof is deferred to Appendix E.

Proposition B.2. *The loss $L_1(\theta; T)$ has gradient Lipschitz constant $\Theta(1/T^2)$ and Hessian Lipschitz constant $\Theta(1/T^3)$.*

Now we calculate the gradients. Let $l \in \mathbb{R}^6$ be the vector of prediction logits for relations. Let the prediction $p_1 = \mathrm{softmax}(l) \in \mathbb{R}^6$. The loss $\ell_1(v) = -\log(p_1(r_1))$. Therefore the gradient w.r.t. the logit l is

$$\frac{\partial \ell_1}{\partial l} = \frac{1}{T}(p_1 - e_1).$$

The gradients w.r.t. v are

$$\frac{\partial \ell_1}{\partial v_i} = \sum_{j=1}^{6} \frac{\partial \ell_1}{\partial l_j} \frac{\partial l_j}{\partial v_i}.$$

$l(s_1)$	$v_2^2 + v_4^2$
$l(s_2)$	$v_2^2 + v_4^2$
$l(s_3)$	$v_2^2 + v_4^2$
$l(a_1)$	$v_1^2 + v_3^2 + v_5^2$
$l(a_2)$	$v_1^2 + v_3^2 + v_5^2$
$l(a_3)$	$v_1^2 + v_3^2 + v_5^2$
$l(r_1)$	$(v_1 + v_2)^2 + (v_3 + v_4)^2 + v_5^2$
$l(r_2)$	$(v_1 + v_4)^2 + (v_2 + v_3)^2 + v_5^2$
$l(r_3)$	$(v_5 + v_4)^2 + (v_1 + v_2)^2 + v_3^2$
$l(r_4)$	$(v_5 + v_2)^2 + (v_1 + v_4)^2 + v_3^2$
$l(r_5)$	$(v_5 + v_4)^2 + (v_3 + v_2)^2 + v_1^2$
$l(r_6)$	$(v_5 + v_2)^2 + (v_3 + v_4)^2 + v_1^2$

Table 2. The prediction logits (EoS excluded) of the input $(s_1, a_1, s_2, a_2, s_3, u_{\mathrm{EoS}})$.

Plugging in the logits in Table 2 and noting that it is equivalent to analyze the gradient of a single sequence, we obtain

$$\frac{\partial L_1}{\partial v_1} = \frac{2}{T}\left(\left(p_1\left(r_1\right) + p_1\left(r_3\right)\right) v_2 + \left(p_1\left(r_2\right) + p_1\left(r_4\right)\right) v_4 - v_2\right) \tag{2}$$

$$\frac{\partial L_1}{\partial v_2} = \frac{2}{T}\left(\left(p_1\left(r_1\right) + p_1\left(r_3\right)\right) v_1 + \left(p_1\left(r_2\right) + p_1\left(r_5\right)\right) v_3 + \left(p_1\left(r_4\right) + p_1\left(r_6\right)\right) v_5 - v_1\right) \tag{3}$$

$$\frac{\partial L_1}{\partial v_3} = \frac{2}{T}\left(\left(p_1\left(r_1\right) + p_1\left(r_6\right)\right) v_4 + \left(p_1\left(r_2\right) + p_1\left(r_5\right)\right) v_2 - v_4\right) \tag{4}$$

$$\frac{\partial L_1}{\partial v_4} = \frac{2}{T}\left(\left(p_1\left(r_1\right) + p_1\left(r_6\right)\right) v_3 + \left(p_1\left(r_2\right) + p_1\left(r_4\right)\right) v_1 + \left(p_1\left(r_3\right) + p_1\left(r_5\right)\right) v_5 - v_3\right) \tag{5}$$

$$\frac{\partial L_1}{\partial v_5} = \frac{2}{T}\left(\left(p_1\left(r_3\right) + p_1\left(r_5\right)\right) v_4 + \left(p_1\left(r_4\right) + p_1\left(r_6\right)\right) v_2\right) \tag{6}$$

$$\frac{\partial L_1}{\partial v_6} = 0. \tag{7}$$

The gradient w.r.t. θ is

$$\frac{\partial L_1}{\partial \theta_i} = v_i \left(\frac{\partial L_1}{\partial v_i} - \sum_{j=1}^{6} v_j \frac{\partial L_1}{\partial v_j} \right). \tag{8}$$

Theorem B.3. *There exists constant $T_0(\delta) > 0$ that only depends on δ, such that for any $T < T_0$, if we set $\eta_1 = \Theta(T^2), \eta_2 = \Theta(T^2), t_1 = t_2 = \infty$ in Algorithm 1, then with probability at least $1 - \delta$ the transformer trained by Algorithm 1 will converge to the global optimum θ^* for $L_1(\theta; T)$. At the convergence the attention scores $v^* = (a, a, \frac{1}{2} - a, \frac{1}{2} - a, 0, 0)$ with some $a \neq \frac{1}{4}$.*

Theorem B.3 shows that, if the model is trained to convergence through Stages 1 and 2, a pairwise attention pattern emerges. The behavior in each stage is characterized by Lemma B.4, Lemma B.5, and Lemma B.7. In Stage 1, the model converges to a saddle point.

Lemma B.4. *At the end of Stage 1, θ will converge to a saddle point $\tilde{\theta}$ with $\tilde{v} := \mathcal{S}(\tilde{\theta}) = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 0, 0)$.*

Proof. Some observations:

1. v_1 and v_3 are symmetric; v_2 and v_4 are symmetric along the gradient flow.
2. Initially $\nabla_\theta L_1(0, T) \neq 0$ so the loss will strictly decrease from $L_1(0, T) = \log 6$.
3. $v_5 < v_1$ for any $t > 0$. This is because any v with $v_5 \geq v_1$, $v_1 = v_3$ and $v_2 = v_4$ yields a loss $L_1(v, T) \geq \log 6$. Similar we have that $v_4 = v_2 > 0$ for any time $t > 0$, otherwise all six logits have same value and loss $L_1 = \log 6$, which is contradictory to observation 2.
4. $\frac{\partial L_1}{\partial v_i} < 0$ for $i = 1, 2, 3, 4$ and $\frac{\partial L_1}{\partial v_5} > 0$, $\frac{\partial L_1}{\partial v_6} = 0$ for any time $t > 0$.
5. $p_3 = p_4 = p_5 = p_6$ for any time t.

As $t_1 \to \infty$, at the end of Stage 1, θ will converge to a critical point of the loss. We know that this critical point must satisfy $v_1 = v_3$ and $v_4 = v_5$ and moreover obtain loss $< \log 6$. We first check the critical point condition: for any i, either $v_i = 0$ or $\frac{\partial L_1}{\partial v_i} = \sum_{j=1}^{6} v_j \frac{\partial L_1}{\partial v_j}$. Combined with observation 4 and 3, we have that the critical point along the GD trajectory must satisfy $v_5 = v_6 = 0$, $v_1 = v_3 > 0$ and $v_2 = v_4 > 0$. So we can simplify the gradients to obtain $\frac{\partial L_1}{\partial v_1} = -2(p_5 + p_6)v_2$ and $\frac{\partial L_1}{\partial v_2} = -2(p_4 + p_6)v_1$. Then observation 5 combined with critical point condition implies that $v_1 = v_2 = 1/4$. Therefore $\tilde{v}$ is the only limit point of this gradient flow on the unit simplex, which is a saddle point as $(1, 1, -1, -1, 0, 0)/2$ is a descending direction and $(1, -1, -1, 1, 0, 0)/2$ is an ascending direction.

It remains to prove observations 1,2 and 5.

Proof to observations 1, 2, 5. We prove observations 1 and 5 by induction. Assume that observations 1 and 5 are true at iteration k. Comparing (2) and (4) and plugging them into (8), we can see that the gradients $\frac{\partial L_1}{\partial \theta_1} = \frac{\partial L_1}{\partial \theta_3}$ at iteration k. Similarly we can see that $\frac{\partial L_1}{\partial \theta_2} = \frac{\partial L_1}{\partial \theta_4}$ at iteration k. Therefore after one step of GD update, at iteration $k + 1$, we still have $v_1 = v_3$ and $v_2 = v_4$, which implies $p_1(r_3) = p_1(r_4) = p_1(r_5) = p_1(r_6)$ at iteration $k + 1$ through the logits l in Table 2. It is obvious that the induction hypothesis holds at initialization $\theta = 0$.

At initialization, $v_i = \frac{1}{6}$, $\frac{\partial L_1}{\partial v_1} = \frac{\partial L_1}{\partial v_3} = -\frac{1}{9T}$, $\frac{\partial L_1}{\partial v_2} = \frac{\partial L_1}{\partial v_4} = \frac{\partial L_1}{\partial v_6} = 0$, $\frac{\partial L_1}{\partial v_5} = \frac{2}{9T}$. Therefore $v^\top \nabla_v L_1 = 0$ and $\nabla_\theta L_1 = \frac{1}{6} \nabla_v L_1 \neq 0$, and Observation 2 follows. □

□

We then show that in the late stage of training we have $l(r_1) > \sum_r p(r) l(r)$, which indicates that θ has escaped from the saddle point $\tilde{\theta}$.

Lemma B.5. *If $T < T_{\max}(\delta)$ where $T_{\max}(\delta)$ is some constant only depends on δ, then with probability at least $1 - \delta$, after the perturbation followed by $O(\mathrm{poly}(\frac{1}{T}) \cdot \log \frac{1}{\delta})$ steps of gradient descent, we have the loss $L_1 < \log 2$.*

The proof idea is that as T gets close to 0, $L_1(\tilde{\theta}, T)$ will converge to $\log 2$ from above. The rate of error decreasing is roughly $\exp(-1/T)$. We can calculate to see $\lambda_{\min}(\nabla^2 L_1(\tilde{\theta}, T)) \lesssim -\frac{1}{T}$ and the Hessian Lipschitz constant for L_1 is $O(1/T^3)$. By Lemma 14 in (Jin et al., 2017), the loss will improve by a polynomial amount $L_1(\theta_t) - L_1(\tilde{\theta}) \leq -\Omega(T^3)$. Hence there exists some constant $T_{\max}(\delta)$ such that for any $T < T_{\max}$ we have the improvement surpasses the gap from $\log 2$. For the ease of reference, we restate Lemma 14 in (Jin et al., 2017) here.

Lemma B.6 (restated from Lemma 14 in (Jin et al., 2017))**.** *Suppose $f(x)$ is an α-gradient Lipschitz and ρ-Hessian Lipschitz function where $x \in \mathbb{R}^d$. There exists a universal constant c, for any $\delta \in (0, \frac{d\alpha}{\gamma}]$, suppose $\tilde{x}$ satisfies*

$$\lambda_{\min}(\nabla^2 f(\tilde{x})) \leq -\gamma \text{ and } \|\nabla f(\tilde{x})\| \leq \sqrt{\eta\alpha} \frac{\gamma^2}{\rho} \cdot \log^{-2}(\frac{d\alpha}{\delta\gamma})$$

with some $\gamma > 0$ and $\eta \le c/\alpha$.

Let $x_0 = \tilde{x} + \xi$ where $\xi \sim \mathrm{Uniform}(\mathbb{B}(0, \sqrt{\eta\alpha}\frac{\gamma^2}{\rho\alpha}\log^{-2}(\frac{d\alpha}{\delta\gamma})))$ and let x_k be the k-th iteration of gradient descent starting from x_0 with stepsize η, then with probability at least $1-\delta$, for any $t \ge \frac{\log(\frac{d\alpha}{\delta\gamma})}{c\eta\gamma}$ it holds that

$$f(x_t) - f(\tilde{x}) \le -\eta\alpha\frac{\gamma^3}{\rho^2}\log^{-3}(\frac{d\alpha}{\delta\gamma}).$$

Now we are ready to prove Lemma B.5.

Proof to Lemma B.5. Let $\tilde{\epsilon} = \frac{1}{2}\cdot(1,1,-1,-1,0,0)$ be the potential descending direction of loss. Define $f_{\tilde{\epsilon},\tilde{\theta}}(x) := L_1(\tilde{\theta} + x\cdot\tilde{\epsilon})$ to be the loss along the direction of $\tilde{\epsilon}$ from $\tilde{\theta}$. Then we have $\lambda_{\min}(\nabla^2 L_1(\tilde{\theta})) \le f''_{\tilde{\epsilon},\tilde{\theta}}(0)$. We use $p_1(x)$ to denote $p_1(r_1)$ at parameter $\theta = \tilde{\theta} + x\cdot\tilde{\epsilon}$. So we can write the loss along the direction of $\tilde{\epsilon}$ from $\tilde{\theta}$ as $f_{\tilde{\epsilon},\tilde{\theta}}(x) = -\log p_1(x)$ and hence we can calculate to see that

$$f''_{\tilde{\epsilon},\tilde{\theta}}(0) = \frac{(p_1'(0))^2}{p_1^2(0)} - \frac{p_1''(0)}{p_1(0)}. \tag{9}$$

We first expand

$$p_1'(0) = \sum_i \frac{\partial p_1}{\partial l_i} l_i'(0) = \sum_i \frac{\partial p_1}{\partial l_i}\nabla_v l_i(v(0))^\top v'(0) \tag{10}$$

where $l_i(x)$ is the logit for r_i at parameter $\theta = \tilde{\theta} + x\cdot\tilde{\epsilon}$ and $v(x) = \mathrm{softmax}(\tilde{\theta} + x\cdot\tilde{\epsilon})$ are the attention scores. So we can write

$$v'(0) = J(v(0))\tilde{\epsilon} = \left(\mathrm{Diag}(v(0)) - v(0)v^\top(0)\right)\tilde{\epsilon}$$

where $J(v(0)) = \mathrm{Diag}(v(0)) - v(0)v^\top(0)$ is the Jacobian of softmax. Therefore we obtain

$$v_i'(0) = \tilde{\epsilon}_i v_i(0) - v_i(0)\cdot v(0)^\top\tilde{\epsilon}. \tag{11}$$

Noting that $v(0)^\top\tilde{\epsilon} = 0$, we have

$$v'(0) = \tilde{\epsilon}\odot v = \left(\frac{1}{8},\frac{1}{8},-\frac{1}{8},-\frac{1}{8},0,0\right)^\top.$$

Also by the logits in table 2 we have

$$\nabla_v l_i(v(0)) = (1,1,1,1,0,0)^\top \text{ for } i = 1,2,$$

$$\nabla_v l_3(v(0)) = \left(1,1,\frac{1}{2},\frac{1}{2},0,0\right)^\top,$$

$$\nabla_v l_4(v(0)) = \left(1,\frac{1}{2},\frac{1}{2},1,0,0\right)^\top,$$

$$\nabla_v l_5(v(0)) = \left(\frac{1}{2},1,1,\frac{1}{2},0,0\right)^\top$$

and

$$\nabla_v l_6(v(0)) = \left(\frac{1}{2},\frac{1}{2},1,1,0,0\right)^\top.$$

Therefore we obtain

$$l'(0) = \sum_i \nabla_v l_i^\top v' = (0, 0, \frac{1}{8}, 0, 0, -\frac{1}{8})^\top. \tag{12}$$

Also since $p_1 = \mathrm{softmax}(l/T)_1$, we have that

$$\frac{\partial p_1}{\partial l_i} = \frac{1}{T} p_1(\delta_{i1} - p_i). \tag{13}$$

Plugging (12) and (13) into (10) we have $p_1'(0) = 0$. Therefore we can simplify (9) to obtain $f''_{\tilde{\epsilon},\tilde{\theta}}(0) = -\frac{p_1''(0)}{p_1(0)}$ where

$$p_1''(0) = \underbrace{\sum_{i=1}^{6} \frac{\partial p_1}{\partial l_i} l_i''(0)}_{\text{term I}} + \underbrace{\sum_{i,j=1}^{6} \frac{\partial^2 p_1}{\partial l_i \partial l_j} l_i'(0) l_j'(0)}_{\text{term II}}.$$

We first calculate term I. Since $l_i'(0) = \nabla_v l_i(v(0))^\top v'(0)$, we can differentiate to obtain

$$l_i'' = v'^\top \nabla_v^2 l_i(v) v' + l_i'(v)^\top v''.$$

Differentiating on (11) yields

$$\begin{aligned} v_i''(0) &= \tilde{\epsilon}_i v_i'(0) - v_i'(0) v(0)^\top \tilde{\epsilon} - v_i(0) v'(0)^\top \tilde{\epsilon} \\ &= v_i(0) \left(\tilde{\epsilon}_i^2 - \sum_{j=1}^{6} v_j(0) \tilde{\epsilon}_j^2 \right) \\ &= 0. \end{aligned}$$

Therefore we can simplify l_i'' to obtain

$$l_i''(0) = v'(0)^\top \nabla_v^2 l_i(v(0)) v'(0) \tag{14}$$

where we have

$$\nabla_v^2 l_1(v(0)) = \begin{pmatrix} 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\nabla_v^2 l_2(v(0)) = \begin{pmatrix} 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\nabla_v^2 l_3(v(0)) = \begin{pmatrix} 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\nabla_v^2 l_4(v(0)) = \begin{pmatrix} 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\nabla_v^2 l_5(v(0)) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and

$$\nabla_v^2 l_6(v(0)) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Combining them with $v'(0) = \left(\frac{1}{8}, \frac{1}{8}, -\frac{1}{8}, -\frac{1}{8}, \right)^\top$ and plugging into 14, we have

$$l''(0) = \left(\frac{1}{4}, 0, \frac{3}{16}, \frac{1}{16}, \frac{1}{16}, \frac{3}{16}\right). \tag{15}$$

At saddle point $x = 0$ we have $p_1(r_1) = p_1(r_2) = p_1(0)$ and $p_1(r_3) = p_1(r_4) = p_1(r_5) = p_1(r_6) = \frac{1-2p_1(0)}{4}$. Combining them with (13), (15) and plugging into term I, we obtain

$$\text{term I} = \frac{p_1(0)}{8T}.$$

Then we work with term II. Recall term II $= \sum_{i,j=1}^6 \frac{\partial^2 p_1}{\partial l_i \partial l_j} l_i'(0) l_j'(0)$ and $l'(0) = (0, 0, \frac{1}{8}, 0, 0, -\frac{1}{8})^\top$ in (12). Hence we only need to consider $l_3'(0)$ and $l_6'(0)$ since the rest of l_i' are zero. Specifically we have

$$\frac{\partial^2 p_1}{\partial l_3^2} = \frac{p_1(0)}{T^2}\left(2p_1(r_3)^2 - p_1(r_3)\right), \frac{\partial^2 p_1}{\partial l_6^2} = \frac{p_1(0)}{T^2}\left(2p_1(r_6)^2 - p_1(r_6)\right)$$

and

$$\frac{\partial^2 p_1}{\partial l_3 \partial l_6} = \frac{2p_1(0)p_1(r_3)p_1(r_6)}{T^2}.$$

Plugging them into term II, we obtain term II $= \frac{-p_1(0)p_1(r_3)^2}{16T^2}$. Noting that $p_1(r_3)$ at the saddle is exponentially small, we obtain

$$f''_{\tilde{\epsilon},\tilde{\theta}}(0) = -\frac{1}{8T} + \frac{1}{16T^2\left(4 + 2\exp(\frac{3}{16T})\right)^2} \lesssim -\frac{1}{T}.$$

So we have $\lambda_{\min}(\nabla^2 L_1(\tilde{\theta})) \lesssim -\frac{1}{T}$.

By proposition B.2, we have the Hessian Lipschitz constant is $O(1/T^3)$. Plug the upper bound on $\lambda_{\min}(\nabla^2 L_1(\tilde{\theta}))$ and the Hessian Lipschitz constants into Lemma 14 in Jin et al. (2017), we have the improvement of loss being $\Omega(T^3)$, which exceeds the gap between $L_1(\tilde{\theta})$ and $\log 2$ if T is small enough.

□

Adding perturbation ξ to the saddle point $\tilde{\theta}$ would not change v_5 and v_6 since $\tilde{\theta}_5$ and $\tilde{\theta}_6$ are $-\infty$ at the convergence of Stage 1. So we need only to consider the perturbation on the first 4 coordinates. It turns out that a pairwise attention pattern will emerge along the gradient descent dynamics after escaping the saddle point.

Lemma B.7. *Denote α the ratio $\frac{v_1}{v_2}$ and β the ratio $\frac{v_3}{v_4}$. Define $\psi := \max\{\alpha, \beta, \frac{1}{\alpha}, \frac{1}{\beta}\}$. If $l(r_1) > \sum_r p_1(r)l(r)$ and $v_5 = 0$, then ψ will be decreasing unless $\alpha = \beta = 1$.*

Proof sketch of Lemma B.7. We have that $\frac{d\alpha}{dt} < 0$ is equivalent to $\frac{\partial L_1}{\partial \theta_1} - \frac{\partial L_1}{\partial \theta_2} > 0$. By calculations we have $\sum_j v_j \frac{\partial L_1}{\partial v_j} = \sum_r p_1(r)l(r)$, which implies that

$$
\begin{aligned}
&T\left(\frac{\partial L_1}{\partial \theta_1} - \frac{\partial L_1}{\partial \theta_2}\right) \\
=&(p_1(r_2) + p_1(r_4))v_1v_4 - (p_1(r_2) + p_1(r_5))v_2v_3 - (p_1(r_4) + p_1(r_6))v_2v_5 + (v_1 - v_2)\left(l(r_1) - \sum_r p_1(r)l(r)\right)
\end{aligned}
$$

and

$$
\begin{aligned}
&T\left(\frac{\partial L_1}{\partial \theta_3} - \frac{\partial L_1}{\partial \theta_4}\right) \\
=&(p_1(r_2) + p_1(r_5))v_2v_3 - (p_1(r_2) + p_1(r_4))v_1v_4 - (p_1(r_3) + p_1(r_5))v_4v_5 + (v_3 - v_4)\left(l(r_1) - \sum_r p_1(r)l(r)\right).
\end{aligned}
$$

Note that if $\alpha \geq \beta$, then we have $v_1v_4 \geq v_2v_3$ and $p_1(r_4) \geq p_1(r_5)$. Furthermore if $\alpha \geq 1$, combining it with $v_5 = 0$, we have $\frac{\partial L_1}{\partial \theta_1} - \frac{\partial L_1}{\partial \theta_2} \geq 0$. It is worth noting that the equality $\frac{\partial L_1}{\partial \theta_1} = \frac{\partial L_1}{\partial \theta_2}$ iff $\alpha = \beta = 1$. Therefore we can deduce that $\frac{d\alpha}{dt} \leq 0$ from $\alpha \geq \beta$ and $\alpha \geq 1$.The equality holds iff $\alpha = \beta = 1$.

Now we consider 4 cases:

1. $\alpha \geq \beta \geq 1 \geq 1/\beta \geq 1/\alpha$. In this case, we have $\alpha \geq \beta$ and $\alpha \geq 1$. Therefore we have $\frac{d\alpha}{dt} \leq 0$ and the equality holds iff $\alpha = \beta = 1$.

2. $\alpha \geq 1/\beta \geq 1 \geq \beta \geq 1/\alpha$. In this case, we have $\alpha \geq \beta$ and $\alpha \geq 1$. Therefore we also have $\frac{d\alpha}{dt} \leq 0$ and the equality holds iff $\alpha = \beta = 1$.

3. $1/\beta \geq \alpha \geq 1 \geq 1/\alpha \geq \beta$. In this case, we have $\frac{\alpha}{\beta} = \frac{v_4}{v_3} \cdot \frac{v_1}{v_2} \geq 1$. Hence we have $v_1v_4 \geq v_2v_3$ and thus $p_1(r_4) \geq p_1(r_5)$. We also know $v_4 \geq v_3$ since $1/\beta \geq 1$. Plugging them into $T\left(\frac{\partial L_1}{\partial \theta_3} - \frac{\partial L_1}{\partial \theta_4}\right)$ we have $\frac{\partial L_1}{\partial \theta_3} \leq \frac{\partial L_1}{\partial \theta_4}$, which is equivalent to $\frac{d(1/\beta)}{dt} \leq 0$. It is straightforward to verify that the equality holds iff $v_3 = v_4$ and $v_1 = v_2$, which is equivalent to $\alpha = \beta = 1$.

4. $1/\beta \geq 1/\alpha \geq 1 \geq \alpha \geq \beta$. In this case, by exact same argument in case 3, we have $\frac{d(1/\beta)}{dt} \leq 0$ and the equality holds iff $\alpha = \beta = 1$.

The rest 4 cases are symmetric to these 4 cases (by swapping α and β). Therefore we have $\frac{d\psi}{dt} < 0$ as long as $\psi > 1$.

□

Now we are ready to prove Theorem B.3. Actually it only remains to show the convergent point is the global optimum.

Proof. From Lemma B.7 and Lemma B.5 we know θ will converge to some critical point θ^* such that $v^* = (x, x, \frac{1}{2} - x, \frac{1}{2} - x, 0, 0)$ for some $x \neq 1/4$ and at the convergence $L_1 < \log 2$. It is worth noting that $L_1 < \log 2$ implies

$l(r_1) > l(r)$ for all $r \neq r_1$, hence we have $l(r_1) > \sum_r p_1(r) l(r)$. We can write the loss $L_1(x, T)$ as a univariate function of x. We can write it out explicitly as

$$L_1(x) = \log\left(1 + \sum_{j=2}^{6} \exp\left(d_j(x)/T\right)\right)$$

where $d_j(x) := l_j(x) - l_1(x)$. It is easy to see L_1 is symmetric $L_1(x) = L_1(\frac{1}{4} - x)$. Hence it suffices to show that for small enough T, there is only one critical point of L_1 in $[0, 1/4]$ such that $L_1 < \log 2$. Direct caculations give that $d_2 = -8x^2 + 4x - \frac{1}{2}$, $d_3 = -3x^2 + 3x - \frac{3}{4}$, $d_4 = -7x^2 + 4x - \frac{3}{4}$, $d_5 = -7x^2 + 3x - \frac{1}{2}$ and $d_6 = -3x^2$. The critical point condition is

$$\Phi(x) := \sum_{j=2}^{6} d_j'(x) e^{d_j(x)/T} = 0.$$

We can define the rescaled function $\Psi(x) := e^{-d_6(x)/T}\Phi(x)$ so Ψ and Φ have same roots. We can rewrite

$$\Psi(x) = (4 - 16x)e^{(d_2(x) - d_6(x))/T} - 6x + \sum_{j=3}^{5} d_j'(x) e^{(d_j(x) - d_6(x))/T}.$$

Set $h(x) := d_2(x) - d_6(x) = -5x^2 + 4x^2 - \frac{1}{2}$ and it is easy verify that $h(x)$ is strictly increasing on $(0, 1/4)$ and has a unique root

$$x_0 = \frac{4 - \sqrt{6}}{10}.$$

We can verify that $d_j(x_0) < d_6(x_0)$ for $j = 3, 4, 5$. Hence there exists an surrounding interval $I = (x_0 - \epsilon, x_0 + \epsilon) \subset (0, 1/4)$ such that for all $x \in I$ $d_j(x) - d_6(x) < -\gamma$ for some $\gamma > 0$, which implies that

$$\sum_{j=3}^{5} d_j'(x) e^{(d_j(x) - d_6(x))/T} = O(\exp(-\gamma/T)).$$

Therefore we can define the dominating term as

$$\widetilde{\Psi}(x) := (4 - 16x)e^{(d_2(x) - d_6(x))/T} - 6x$$

and we have $|\Psi(x) - \widetilde{\Psi}(x)| = O(\exp(-\gamma/T))$. It is easy to check the difference of the derivative is also small $|\Psi'(x) - \widetilde{\Psi}'(x)| = O(\frac{1}{T}\exp(-\gamma/T))$. It is worth noting that these two bounds are uniform for $x \in I$. We can claim that $\widetilde{\Psi}(x)$ has a unique root in $(0, 1/4)$ which has a large derivative.

Claim 1. There is a unique $x_T \in I$ such that $\widetilde{\Psi}(x_T) = 0$. Furthermore we have $\widetilde{\Psi}'(x_T) = \Theta(\frac{1}{T})$.

With Claim 1, we can see that $\Psi(x)$ also only have one root in I, since with small enough T we have $|\Psi'(x) - \widetilde{\Psi}'(x)| = o(\widetilde{\Psi}'(x))$. Outside I, if $x < x_0 - \epsilon$, then $h(x) < 0$ uniformly hence $\Psi(x) < 0$. If $x > x_0 + \epsilon$ then $h(x) > 0$ uniformly and hence $\Psi(x) > 0$. Therefore L_1 has exactly one critical point x^* in $(0, 1/4)$ and it is direct to check $L_1(x^*) < \log 2$ if T is small enough. It remains to prove the Claim.

Proof to the Claim 1. We can know $\widetilde{\Psi}(x) = 0$ is equivalent to

$$h(x) = T \log \frac{6x}{4 - 16x}.$$

We can define $F(x) := h(x) - T \log \frac{6x}{4-16x}$ and we have its derivative to be

$$F'(x) = h'(x) - T\left(\frac{1}{x} + \frac{16}{4 - 16x}\right).$$

We can examine that $h'(x_0) > 0$, so for small enough ϵ we have $F'(x) > 0$ in I. Also if T is small enough, we have $F(x_0 - \epsilon) < 0$ since $h(x_0 - \epsilon) < 0$. Similarly, we have $F(x_0 + \epsilon) > 0$. Therefore F and $\widetilde{\Psi}$ have exactly one root $x_T \in I$. We can calculate to see that

$$\widetilde{\Psi}'(x_T) = -16e^{h(x_T)/T} + \frac{6x_T h'(x_T)}{T} - 6.$$

The root condition $\widetilde{\Psi}(x_T) = 0$ implies

$$(4 - 16x_T)\, e^{h(x_T)/T} = 6x_T.$$

Plugging it into the derivative of $\widetilde{\Psi}$ we obtain

$$\widetilde{\Psi}'(x_T) = -16e^{h(x_T)/T} + \frac{6x_T h'(x_T)}{T} - 6,$$

where $-16e^{h(x_T)/T} = -16\frac{6x_T}{4-16x_T} = O(1)$ and $x_T h'(x_T) = x_T(4 - 10x_T) = \Theta(1)$. Therefore we have $\widetilde{\Psi}'(x_T) = \Theta(\frac{1}{T})$. □

□

C. Proof to Theorem 4.4

We first write out the gradient in the general setting. Given an input sequence $Z = (s_1, a_1, s_2, a_2, s_3, u_{\text{EoS}})$, we can write its loss as $\ell_1(v; Z, T)$. For any relation r, the logit of r is

$$l(r) = 2 \sum_{1 \le i \le 3, 1 \le j \le 2} v_{2i-1} v_{2j} \mathbf{1}_{\{r(s_i) = a_j\}} + \sum_{i=1}^{5} v_i^2. \tag{16}$$

Therefore the logit of r contains the quadratic term $(v_{2i-1} + v_{2j})^2$ if r maps s_i to a_j, which is consistent with Table 2 in previous three-subject setting. Then the the formulas for gradient w.r.t. v on sequence Z are as follows.

$$T\frac{\partial \ell_1}{\partial v_1} = 2\left(\sum_{r: r(s_1)=a_1} p(r)\, v_2 \; + \sum_{r: r(s_1)=a_2} p(r)\, v_4 \; - \; v_2\right)$$

$$T\frac{\partial \ell_1}{\partial v_2} = 2\left(\sum_{r: r(s_1)=a_1} p(r)\, v_1 \; + \sum_{r: r(s_2)=a_1} p(r)\, v_3 \; + \sum_{r: r(s_3)=a_1} p(r)\, v_5 - v_1\right)$$

$$T\frac{\partial \ell_1}{\partial v_3} = 2\left(\sum_{r:\, r(s_2)=a_2} p(r)\, v_4 \; + \sum_{r:\, r(s_2)=a_1} p(r)\, v_2 - v_4\right)$$

$$T\frac{\partial \ell_1}{\partial v_4} = 2\left(\sum_{r:\, r(s_2)=a_2} p(r)\, v_3 \; + \sum_{r:\, r(s_1)=a_2} p(r)\, v_1 \; + \sum_{r:\, r(s_3)=a_2} p(r)\, v_5 - v_3\right)$$

$$T\frac{\partial \ell_1}{\partial v_5} = 2\left(\sum_{r:\, r(s_3)=a_1} p(r)\, v_2 \; + \sum_{r:\, r(s_3)=a_2} p(r)\, v_4\right)$$

$$\frac{\partial \ell_1}{\partial v_6} = 0$$

We first show that after Stage 1, the transformer will have near perfect accuracy on non-confusing sequences and near 0.5 accuracy on confusing sequences. This is the area near the saddle point.

Lemma C.1. *After Stage 1, if sample size* $|D| = w\left(\frac{1}{T^3}\cdot\log^2(\frac{1}{T})\cdot\log(\frac{1}{\delta})\right)$ *and* $T < \frac{1}{40\log n}$*, with probability at least* $1-\delta/3$*, the attention scores satisfy that* $\|v - (\frac{1}{6}+\alpha,\ \frac{1}{6},\ \frac{1}{6}+\alpha,\ \frac{1}{6},\ \frac{1}{6}-2\alpha,\ \frac{1}{6})\|_\infty \lesssim \frac{\eta_1^2}{T^2}, |v_1 - v_3| = o(T^2)$ *and* $|v_2 - v_4| = o\left(\exp(-\frac{1}{40T})\right)$ *where* $\alpha = \frac{c\cdot p_{\rm mis}\cdot\eta_1}{T}$ *for some constant* c*. Moreover, for non-confusing sequences, the accuracy* $p_1(r_1) = 1 - o\left(\exp(-\frac{1}{\sqrt{T}})\right)$*; for confusing sequences, the accuracy* $p_1(r_1) = \frac{1}{2} - o\left(\exp(-\frac{1}{\sqrt{T}})\right)$.

Proof. At initialization, fix an input sequence Z. Denote $I(Z)$ the number of 2-matching relations for Z. It is easy to see that all 2-matching relations have logits $2\cdot(\frac{1}{6}+\frac{1}{6})^2+\frac{1}{6^2} = \frac{1}{4}$, all 1-matching relations have logits $(\frac{1}{6}+\frac{1}{6})^2+3\cdot\frac{1}{6^2} = \frac{7}{36}$ and all 0-matching relations have logits $\frac{5}{6^2} = \frac{5}{36}$. Since every two (s,a) pairs correspond to at most one relation, we have the number of 1-matching relations is at most $6(n-3)$ and the number of 0-matching relations is at most $n(n-1)-6(n-2) < n^2$. Then $1 \le I(Z) \le 6$. Note that

$$p_1(r_1) = \frac{1}{I(Z)} - O\left(n\exp\left(-\frac{1}{18T}\right) + n^2\exp\left(-\frac{1}{9T}\right)\right).$$

Therefore if $\frac{1}{T} > 40\log n$, then we have

$$p_1(r_1) = \frac{1}{I(Z)} - O\left(\exp\left(-\frac{1}{36T}\right)\right).$$

If Z is not an mismatched sequence, then

$$0 \le \frac{\partial\ell_1(Z)}{\partial v_5} - \frac{\partial\ell_1(Z)}{\partial v_i} < O\left(\exp\left(-\frac{1}{36T}\right)/T\right)$$

for all $i \ne 5$. If Z is a mismatched sequence, then there is at least one 2-matching relation r that maps s_3 either to a_1 or to a_2, which has prediction probability same as r_1 since r_1 is also a 2-matching relation. Hence we have $\frac{\partial\ell_1}{\partial v_5} \ge \frac{p_1(r_1)}{3T}$. Noting that $\frac{\partial\ell_1(Z)}{\partial v_i} \le 0$ for $i \ne 5$, we have

$$\frac{p_1(r_1)}{3T} \le \frac{\partial\ell_1(Z)}{\partial v_5} - \frac{\partial\ell_1(Z)}{\partial v_i} < \frac{2}{3T}$$

for all $i \ne 5$. Summing over Z, we have

$$\frac{p(r_1)\cdot p_{\rm mis}}{3T} \le \frac{\partial L_1(D)}{\partial v_5} - \frac{\partial L_1(D)}{\partial v_i} < \frac{2\cdot p_{\rm mis}}{3T} + O\left(\exp\left(-\frac{1}{36T}\right)/T\right). \tag{17}$$

Also note that (1) $|\frac{\partial\ell_1}{\partial v_i}| \le O\left(\exp(-\frac{1}{36T})/T\right)$ for $i = 2,4,6$; (2) $|\frac{\partial\ell_1}{\partial v_1} + \frac{\partial\ell_1}{\partial v_3} + \frac{\partial\ell_1}{\partial v_5}| = O\left(\exp\left(-\frac{1}{36T}\right)/T\right)$. Summing over Z, we have

$$\left|\frac{\partial L_1(D)}{\partial v_i}\right| \le O\left(\exp\left(-\frac{1}{36T}\right)/T\right) \text{ for } i = 2,4,6 \tag{18}$$

and

$$\left|\frac{\partial L_1}{\partial v_1} + \frac{\partial L_1}{\partial v_3} + \frac{\partial L_1}{\partial v_5}\right| = O\left(\exp\left(-\frac{1}{36T}\right)/T\right). \tag{19}$$

By (18) and (19), we know at initialization

$$|v_0^\top\nabla_v L_1| = O\left(\exp\left(-\frac{1}{36}\right)/T\right).$$

Here $v_0 = \left(\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6}\right)^\top$ is the attention scores at initialization. Therefore we have

$$\|\nabla_\theta L_1\| = O(\|\nabla_v L_1\|) = \Theta\left(\frac{1}{T}\right).$$

Since $\frac{\partial L_1}{\partial v_i} = v_i\left(\frac{\partial L_1}{\partial v_i} - v^\top \nabla_v L_1\right)$, we have that

$$\|\nabla_\theta L_1(D) - \frac{1}{6}\nabla_v L_1(D)\| = O\left(\exp\left(-\frac{1}{36T}\right)/T\right)$$

at initialization, which also implies that $|v_0^\top \nabla_\theta L_1| = O(\exp(-\frac{1}{36})/T)$. Use Taylor expansion we have after Stage 1 that

$$v - v_0 = -\eta_1 J \nabla_\theta L_1(D) + \epsilon$$

where $J = \mathrm{Diag}(v_0) - v_0 v_0^\top$ is the Jacobian of softmax at initialization and the remainder

$$\|\epsilon\| = O(\eta_1^2 \|\nabla_\theta L_1\|^2) = O(\frac{\eta_1^2}{T^2}).$$

Therefore we can calculate to see that

$$\begin{aligned}\left|v_1 + v_3 + v_5 - \frac{1}{2}\right| &\lesssim \eta_1 \left|\frac{\partial L_1}{\partial v_1} + \frac{\partial L_1}{\partial v_3} + \frac{\partial L_1}{\partial v_5}\right| + \eta_1 |v_0^\top \nabla_\theta L_1| + \|\epsilon\| \\ &\lesssim \frac{\eta_1^2}{T^2}.\end{aligned}$$

Since we have

$$\frac{\partial L_1(D)}{\partial \theta_i} - \frac{\partial L_1(D)}{\partial \theta_j} = \frac{1}{6}\left(\frac{\partial L_1(D)}{\partial v_i} - \frac{\partial L_1(D)}{\partial v_j}\right)$$

for the gradient at initialization for all i, j, applying (17), (18) together with Taylor expansion imply that after first step of GD, we also have

$$|v_5 - v_i| = \Theta(\frac{p_{\mathrm{mis}}\cdot\eta_1}{T}) \text{ and } |v_i - v_j| = O\left(\eta_1 \cdot \exp\left(-\frac{1}{36T}\right)/T\right)$$

for $i, j \in \{2, 4, 6\}$. We still need to show that v_1 and v_3 are close to each other.

Since the gradients $\left|\frac{\partial \ell_1(Z)}{\partial v_1}\right|, \left|\frac{\partial \ell_1(Z)}{\partial v_3}\right|$ for any sample Z are upper bounded by $\frac{2}{T}$ and $\mathbb{E}_Z[\frac{\partial \ell_1(Z)}{\partial v_1}] = \mathbb{E}_Z[\frac{\partial \ell_1(Z)}{\partial v_3}]$ at initialization. By Chernoff bound, if the sample size $|D| = w\left(\frac{1}{T^3}\cdot \log^2(\frac{1}{T})\cdot\log(\frac{1}{\delta})\right)$, then with probability at least $1 - \delta/3$, we have

$$\left|\frac{\partial L_1(D)}{\partial v_1} - \frac{\partial L_1(D)}{\partial v_3}\right| = o\left(\sqrt{T}\log^{-1}(\frac{1}{T})\right),$$

which implies that after Stage 1 we have

$$|v_1 - v_3| = \Theta\left(\eta_1\left|\frac{\partial L_1(D)}{\partial v_1} - \frac{\partial L_1(D)}{\partial v_3}\right|\right) = o(T^2) = o\left(\frac{\eta_1^2}{T^2}\right).$$

Therefore we have

$$\|v - (\tfrac{1}{6}+\alpha,\ \tfrac{1}{6},\ \tfrac{1}{6}+\alpha,\ \tfrac{1}{6},\ \tfrac{1}{6}-2\alpha,\ \tfrac{1}{6})\|_\infty \lesssim \frac{\eta_1^2}{T^2}, |v_1 - v_3| = o(T^2)$$

and

$$|v_2 - v_4| = o\left(\exp\left(-\frac{1}{40T}\right)\right)$$

where $\alpha = \frac{c\cdot p_{\mathrm{mis}}\cdot\eta_1}{T}$ for some constant c. With such v, it is straightforward to see that for non-confusing sequences, the accuracy $p(r_1) = 1 - o\left(\exp(-\frac{1}{\sqrt{T}})\right)$; for confusing sequences, the accuracy $p(r_1) = \frac{1}{2} - o\left(\exp(-\frac{1}{\sqrt{T}})\right)$. □

We then show the perturbation scheme helps the model escape the saddle by reducing the loss on confusing sequences.

Lemma C.2. *If $T < T_{\max}(\delta)$ where $T_{\max}(\delta)$ is some constant only depends on δ, with probability at least $1-\delta/3$, after $O(1/T)$ iterations of GD in Stage 2 in Algorithm 1, we have loss on confusing sequences $\ell_1 \le \log 2 - \Omega(p_{\mathrm{conf}}^3 \cdot T^3)$.*

Proof. To apply Lemma 14 in Jin et al. (2017), we need to examine two properties at $\tilde{\theta}$: (1) the gradient $\|\nabla L_1(\theta, D)\|$ is small, and (2) the most negative eigenvalue of the Hessian $\lambda_{\min}(\nabla^2 L_1(\tilde{\theta}, D))$ is negative enough.

We first examine (1). If Z is a non-confusing sequence, it is easy to see

$$\|\nabla_v \ell_1(v, Z)\| = o\left(-\exp\left(-\frac{1}{\sqrt{T}}\right)/T\right).$$

If Z is a confusing sequence, we have

$$\|\nabla_v \ell_1(v, Z)\|_\infty \lesssim \frac{1}{T}\max\{|v_1 - v_3|, |v_2 - v_4|\}.$$

By Lemma C.1, we know $\|\nabla_v \ell_1(v, Z)\|_\infty = o(T)$. Overall we have

$$\|\nabla_\theta \ell_1(v, Z)\| = O(\|\nabla_v \ell_1(v, Z)\|) = o(T).$$

Now we examine (2). Let $\tilde{\epsilon} = \frac{1}{2}\cdot(1, 1, -1, -1, 0, 0)$ be the potential descending direction of loss. Define $f_{\tilde{\epsilon},\tilde{\theta}}(x, Z) := \ell_1(\tilde{\theta} + x\cdot\tilde{\epsilon}, Z)$ to be the loss along the direction of $\tilde{\epsilon}$ from $\tilde{\theta}$. Then we have $\lambda_{\min}(\nabla^2 \ell_1(\tilde{\theta}, Z)) \le f''_{\tilde{\epsilon},\tilde{\theta}}(0, Z)$. We use $p_1(x)$ to denote $p_1(r_1)$ at parameter $\theta = \tilde{\theta} + x\cdot\tilde{\epsilon}$. Similarly we use $p_i(x)$ for $p_1(r_i)$ in this proof. Direct calculation gives that

$$f''_{\tilde{\epsilon},\tilde{\theta}}(0) = \frac{p_1'(0)^2}{p_1^2(0)} - \frac{p_1''(0)}{p_1(0)}.$$

We first work with the situation where Z is a confusing sequence. We have

$$p_1'(0) = \sum_i \frac{\partial p_1}{\partial l_i}\cdot l_i'(0)$$

where $\frac{\partial p_1}{\partial l_1} = \frac{1}{T}p_1(1 - p_1)$ and $\frac{\partial p_1}{\partial l_i} = -\frac{1}{T}p_1 p_i$ for $i \neq 1$. For all relations r that are neither the underlying true relation r_1 nor the confusing relation r_2, the probability of predicting it $p_1(r) = o\left(\exp(-\frac{1}{\sqrt{T}})\right)$ is exponentially small after Stage 1. It is sufficient to only look at r_1 and r_2. Further we have

$$l_i'(0) = (\nabla_v l_i)^\top v'(0)$$

where $\|v'(0) - \delta\odot v\|_\infty = o(T^2)$ since $v_i'(0) = \delta_i v_i - v_i v^\top\delta$ for each i and $|v^\top\delta| = o(T^2)$ by Lemma C.1. Also we have

$$\nabla_v l_1 = 2\,(v_1 + v_2, v_1 + v_2, v_3 + v_4, v_3 + v_4, v_5, 0)^\top$$

and

$$\nabla_v l_2 = 2\,(v_1 + v_4, v_2 + v_3, v_2 + v_3, v_1 + v_4, v_5, 0)^\top.$$

Hence we have

$$l_1'(0) = (\nabla_v l_1)^\top(\delta\odot v) + o(T^2) = (v_1 + v_2 + v_3 + v_4)(v_1 - v_3 + v_2 - v_4) + o(T^2) = o(T^2).$$

Similarly we have $l_2'(0) = o(T^2)$. Hence we obtain that $p_1'(0) = o(T)$. Therefore we have

$$f''_{\tilde{\epsilon},\tilde{\theta}}(0) = -\frac{p_1''(0)}{p_1(0)} + o(T^2). \tag{20}$$

We have that

$$p_1''(0) = \underbrace{\sum_i \frac{\partial p_1}{\partial l_i} l_i''(0)}_{\text{term I}} + \underbrace{\sum_{i,j}\frac{\partial^2 p_1}{\partial l_i \partial l_j}\cdot l_i'(0)\cdot l_j'(0)}_{\text{term II}}. \tag{21}$$

First calculate term I. We know

$$l_i''(0) = v'(0)^\top \nabla_v^2 l_i(v(0))v'(0) + (\nabla_v l_i)^\top v''(0).$$

Note that

$$v_i''(0) = \delta_i v_i'(0) - v_i' v^\top \delta - v_i (v')^\top \delta$$

and $|v^\top \delta| = o(T^2)$, we have

$$v_i''(0) = v_i \left(\delta_i^2 - \sum_{j=1}^{6} v_j \delta_j^2 \right) + o\left(T^2\right).$$

By Lemma C.1, we have

$$\sum_{j=1}^{6} v_j \delta_j^2 = \frac{1}{6} + O\left(\sqrt{T} \log \frac{1}{T}\right).$$

Hence we have

$$v_i''(0) = \frac{1}{12} v_i - O\left(\sqrt{T} \log \frac{1}{T}\right),$$

which implies that

$$(\nabla_v l_i)^\top v''(0) = \frac{1}{24} - O\left(\sqrt{T} \log \frac{1}{T}\right)$$

for $i = 1, 2$. We also have

$$v'(0)^\top \nabla_v^2 l_1 v'(0) = \frac{1}{12} + O\left(\sqrt{T} \log \frac{1}{T}\right)$$

and

$$v'(0)^\top \nabla_v^2 l_2 v'(0) = O\left(\sqrt{T} \log \frac{1}{T}\right).$$

Therefore we have $l_1''(0) = \frac{1}{8} + O(\sqrt{T} \log \frac{1}{T})$ and $l_2''(0) = \frac{1}{24} - O(\sqrt{T} \log \frac{1}{T})$. So we have

$$\begin{aligned}
\text{term I} &= \frac{p_1}{T} \left(\frac{1}{8}(1 - p_1) - \frac{1}{24} p_2 + O\left(\sqrt{T} \log \frac{1}{T}\right) \right) \\
&= \frac{p_1}{12T} \left(p_2 + O\left(\sqrt{T} \log \frac{1}{T}\right) \right) \\
&= \Theta\left(\frac{1}{T}\right).
\end{aligned} \tag{22}$$

For term II, note that

$$\frac{\partial^2 p_1}{\partial l_i \partial l_j} = \frac{2}{T^2} p_1 p_i p_j, \frac{\partial^2 p_1}{\partial l_i^2} = \frac{1}{T^2} p_1 \left(1 - p_1\right)\left(1 - 2p_1\right), \frac{\partial^2 p_1}{\partial l_1 \partial l_i} = \frac{p_1 p_i}{T^2}\left(2p_1 - 1\right)$$

and $\frac{\partial^2 p_1}{\partial l_i^2} = \frac{p_1 p_i}{T^2}\left(2p_i - 1\right)$ for $i \neq j, i \neq 1, j \neq 1$ and they are all $o\left(\exp(-\frac{1}{\sqrt{T}})\right)$. Therefore term II $= o\left(\exp(-\frac{1}{\sqrt{T}})\right)$. Plugging it with (23) into (21), we have $p_1''(0) = -\Theta(1/T)$ and hence

$$f_{\tilde{\epsilon}, \tilde{\theta}}''(0) \leq -\Omega(1/T)$$

for confusing sequences.

Now we assume that Z is a non-confusing sequence. We then have $|\frac{\partial p_1}{\partial l_i}| = o\left(T^{-1} \exp(-\frac{1}{\sqrt{T}})\right)$ and $|l_i'(0)| = o(T^2)$. Hence $|p_1'(0)| = o\left(T \exp(-\frac{1}{\sqrt{T}})\right)$. So we have for non-confusing Z

$$f_{\tilde{\epsilon}, \tilde{\theta}}''(0) = -\frac{p_1''(0)}{p_1(0)} + o\left(T^2 \exp(-\frac{2}{\sqrt{T}})\right).$$

Similarly we can calculate that for non-confusing Z that

$$\begin{aligned}\text{term I} &= \frac{p_1}{T}\left(\frac{1}{8}(1-p_1) + O\left(\sqrt{T}\log\frac{1}{T}\right)\right)\\ &= O\left(T^{-\frac{1}{2}}\log(\frac{1}{T})\right)\\ &= o\left(\frac{1}{T}\right)\end{aligned} \tag{23}$$

and term II $= o\left(\exp(-\frac{1}{\sqrt{T}})\right)$. Hence for non-confusing Z we have $p_1''(0) = o(\frac{1}{T})$ and $|f''_{\tilde{\epsilon},\tilde{\theta}}(0,Z)| = o(\frac{1}{T})$. Summing over all confusing and non-confusing Z, we have $f''_{\tilde{\epsilon},\tilde{\theta}}(0,D) \lesssim -\Omega(p_{\text{conf}}/T)$ and therefore $\lambda_{\min}(\nabla^2 L_1(\tilde{\theta}, D)) \lesssim -\Omega(p_{\text{conf}}/T)$.

Now we can apply Lemma 14 in Jin et al. (2017) and obtain that after $\tilde{O}(\frac{1}{p_{\text{conf}}\cdot T})$ iterations of GD in Stage 2, with probability at least $1-\delta/3$, we have

$$L_1(\theta, D) \le L_1(\tilde{\theta}, D) - \Omega(p_{\text{conf}}^3 \cdot T^3).$$

We know for non-confusing sequences we have $\ell_1(\tilde{\theta}, Z) = o\left(\exp(-\frac{1}{\sqrt{T}})\right)$, hence denoting $D_{\text{conf}} \subset D$ the set of confusing sequences in D, we must have

$$L_1(\theta, D_{\text{conf}}) \le L_1(\tilde{\theta}, D_{\text{conf}}) - \Omega(p_{\text{conf}}^3 \cdot T^3) = \log 2 - \Omega(p_{\text{conf}}^3 \cdot T^3).$$

For any confusing sequence Z, defining $g := l(r_1) - l(r_2) = 2(v_1 - v_3)(v_2 - v_4)$, it is straightforward to verify that

$$\frac{1}{1+\exp(-g/T) + 4\exp(-\sqrt{\frac{1}{T}}) + n^2\exp(-\frac{1}{18T})} \le p_1(r_1, Z) \le \frac{1}{1+\exp(-g/T)}.$$

Therefore for any two confusing sequences Z_1 and Z_2, we know

$$|p_1(r_1, Z_1) - p_1(r_1, Z_2)| = O(\exp(-\frac{1}{\sqrt{T}}))$$

is exponentially small, which implies that $|\ell_1(\theta, Z_1) - \ell_1(\theta, Z_2)| = O(\exp(-\frac{1}{\sqrt{T}}))$. Combining it with that $L_1(\theta, D_{\text{conf}}) \le \log 2 - \Omega(p_{\text{conf}}^3 \cdot T^3)$, we know

$$\ell_1(\theta, Z) \le \log 2 - \Omega(p_{\text{conf}}^3 \cdot T^3)$$

for any confusing sequence Z.

□

Finally we show that after escaping the saddle, the loss on confusing sequences can decrease fast.

Lemma C.3. *After $O(\frac{T}{\eta_2 \cdot p_{\text{conf}}}\log\frac{1}{T})$ iterations of GD in Stage 2 in Algorithm 1, with probability at least $1-\delta/3$, we have $p(r_1) \ge 0.999$ on any confusing sequence.*

Proof. Assume $Z \in D$ is a confusing sequence. We denote $p_2(Z)$ the prediction probability for r_2 on Z and $p_2(D) := \frac{1}{p_{\text{conf}}\cdot|D|}\sum_{\text{confusing } Z' \in D} p_2(Z')$. After $\tilde{O}(1/T)$ iterations of GD in Stage 2, with probability least $1-\delta/3$, we have

$$\ell_1(\theta, Z) \le \log 2 - \Omega(p_{\text{conf}}^3 \cdot T^3).$$

Hence $p(r_2) \le \frac{1}{2} - \Omega(T^3)$, which implies that

$$(v_1 - v_3)(v_2 - v_4) \gtrsim T^4$$

since $l_1 - l_2 = 2(v_1 - v_3)(v_2 - v_4)$. Without loss of generality, assume $v_1 > v_3$ and $v_2 > v_4$. Let

$$\Delta := \mathrm{softmax}(\theta - \eta_2 \nabla_\theta L_1(\theta)) - \mathrm{softmax}(\theta) \in \mathbb{R}^6$$

be the change of the attention scores after one step of gradient descent. Denote

$$g := (v_1 - v_3)(v_2 - v_4).$$

Then we have

$$g_{k+1} - g_k = (v_1 - v_3)(\Delta_2 - \Delta_4) + (v_2 - v_4)(\Delta_1 - \Delta_3) + (\Delta_1 - \Delta_3)(\Delta_2 - \Delta_4) \tag{24}$$

where all Δ and v_i at RHS are at time k. We also have $g_0 \gtrsim T^4$.

Now we show that $g_{k+1} - g_k \geq \frac{\eta_2 p_{\mathrm{conf}}}{10000T} \cdot g_k$ if $p(r_1) < 0.999$.

As long as $k \leq \frac{1}{T}(\log \frac{1}{T})^{1.5}$, by Lemma E.2 we have

$$\|\theta(k) - \tilde{\theta}\|_2 \lesssim \sqrt{k \cdot T^2} \lesssim T^{1/2}(\log \tfrac{1}{T})^{3/4},$$

which implies $\|v(k) - \tilde{v}\|_2 \lesssim T^{1/2}(\log \frac{1}{T})^{3/4}$ and $p(r_i) = o(\exp^{-1/\sqrt{T}})$ for $i \neq 1, 2$.

Also we have

$$\begin{aligned}
\frac{\partial \ell_1(v, Z)}{\partial v_1} &= \frac{1}{T}\big(p_2(Z)(v_4 - v_2) + o(e^{-1/\sqrt{T}})\big), \\
\frac{\partial \ell_1(v, Z)}{\partial v_2} &= \frac{1}{T}\big(p_2(Z)(v_3 - v_1) + o(e^{-1/\sqrt{T}})\big), \\
\frac{\partial \ell_1(v, Z)}{\partial v_3} &= \frac{1}{T}\big(p_2(Z)(v_2 - v_4) + o(e^{-1/\sqrt{T}})\big), \\
\frac{\partial \ell_1(v, Z)}{\partial v_4} &= \frac{1}{T}\big(p_2(Z)(v_1 - v_3) + o(e^{-1/\sqrt{T}})\big), \\
\frac{\partial \ell_1(v, Z)}{\partial v_5} &= o(e^{-1/\sqrt{T}})
\end{aligned}$$

and

$$\frac{\partial \ell_1(v, Z)}{\partial v_6} = 0.$$

We also know

$$\|\nabla_v \ell_1(v, Z')\|_\infty = o\left(T^{-1} \cdot \exp(-1/\sqrt{T})\right)$$

for non-confusing sequences Z'. Therefore

$$\frac{\partial L_1}{\partial v_i} = p_{\mathrm{conf}} \cdot \frac{\partial \ell_1(v, Z)}{\partial v_i} + o\left(T^{-1} \cdot \exp(-1/\sqrt{T})\right)$$

for all i. Moreover,

$$\frac{\partial L_1}{\partial \theta_i} = v_i \left(\frac{\partial L_1}{\partial v_i} - \sum_{j=1}^{6} v_j \frac{\partial L_1}{\partial v_j} \right)$$

and

$$\sum_{j=1}^{6} v_j \frac{\partial L_1}{\partial v_j} = \frac{4p_2 \cdot p_{\mathrm{conf}}}{T}\big(-g + o(e^{-1/\sqrt{T}})\big).$$

Now we are ready to calculate $\Delta_1 - \Delta_3$ and $\Delta_2 - \Delta_4$ in (24).

By Taylor expansion we have

$$\Delta = -\eta_2 J(v(\theta)) \nabla_\theta L_1 + \epsilon$$

where $J(v(\theta)) = \mathrm{Diag}(v) - vv^\top$ is the Jacobian of softmax and $\|\epsilon\| = O(\eta_2^2 \|\nabla_\theta L_1(\theta)\|^2)$.

Therefore we have

$$\Delta_1 - \Delta_3 = \eta_2 \left((v_1 - v_3) v^\top \nabla_\theta L_1 + v_3 \frac{\partial L_1}{\partial \theta_3} - v_1 \frac{\partial L_1}{\partial \theta_1} \right) + \epsilon_1 - \epsilon_3$$

and

$$\Delta_2 - \Delta_4 = \eta_2 \left((v_2 - v_4) v^\top \nabla_\theta L_1 + v_4 \frac{\partial L_1}{\partial \theta_4} - v_2 \frac{\partial L_1}{\partial \theta_2} \right) + \epsilon_2 - \epsilon_4.$$

Plug them into 24 we have

$$\begin{aligned} g_{k+1} - g_k = \eta_2 \Bigg[& 2(v_1 - v_3)(v_2 - v_4)\, v^\top \nabla_\theta L_1 + \eta_2 (v_1 - v_3)(v_2 - v_4) \left(v^\top \nabla_\theta L_1 \right)^2 \\ & + (v_2 - v_4) \left(v_3 \frac{\partial L_1}{\partial \theta_3} - v_1 \frac{\partial L_1}{\partial \theta_1} \right) + (v_1 - v_3) \left(v_4 \frac{\partial L_1}{\partial \theta_4} - v_2 \frac{\partial L_1}{\partial \theta_2} \right) \\ & + \eta_2 \left(v_3 \frac{\partial L_1}{\partial \theta_3} - v_1 \frac{\partial L_1}{\partial \theta_1} \right) \left(v_4 \frac{\partial L_1}{\partial \theta_4} - v_2 \frac{\partial L_1}{\partial \theta_2} \right) \\ & + \eta_2 (v_1 - v_3) v^\top \nabla_\theta L_1 \left(v_4 \frac{\partial L_1}{\partial \theta_4} - v_2 \frac{\partial L_1}{\partial \theta_2} \right) \\ & + \eta_2 (v_2 - v_4) v^\top \nabla_\theta L_1 \left(v_3 \frac{\partial L_1}{\partial \theta_3} - v_1 \frac{\partial L_1}{\partial \theta_1} \right) \Bigg] + O(\|\epsilon\|) \end{aligned} \tag{25}$$

We first obtain

$$\begin{aligned} v_3 \frac{\partial L_1}{\partial \theta_3} - v_1 \frac{\partial L_1}{\partial \theta_1} &= v_3^2 \frac{\partial L_1}{\partial v_3} - v_1^2 \frac{\partial L_1}{\partial v_1} + \left(v_1^2 - v_3^2 \right) \cdot v^\top \nabla_v L_1 \\ &= \frac{2 p_2(D) p_{\mathrm{conf}}}{T} \left(\left(v_1^2 + v_3^2 \right) (v_2 - v_4) + 2 \left(v_3^2 - v_1^2 \right) g + o\left(e^{-\frac{1}{\sqrt{T}}} / T \right) \right). \end{aligned} \tag{26}$$

Similarly we have

$$v_4 \frac{\partial L_1}{\partial \theta_4} - v_2 \frac{\partial L_1}{\partial \theta_2} = \frac{2 p_2(D) p_{\mathrm{conf}}}{T} \left(\left(v_2^2 + v_4^2 \right) (v_1 - v_3) + 2 \left(v_4^2 - v_2^2 \right) g + o\left(e^{-\frac{1}{\sqrt{T}}} / T \right) \right). \tag{27}$$

Also we have

$$\begin{aligned} v^\top \nabla_\theta L_1 &= \sum_{i=1}^{6} v_i^2 \frac{\partial L_1}{\partial v_i} - \|v\|^2 v^\top \nabla_v L_1 \\ &= \frac{2 p_2(D) p_{\mathrm{conf}}}{T} \left((v_3^2 - v_1^2)(v_2 - v_4) + (v_4^2 - v_2^2)(v_1 - v_3) + 2\|v\|^2 g + o\left(e^{-1/\sqrt{T}} \right) \right) \\ &= \frac{2 p_2(D) p_{\mathrm{conf}}}{T} \left(\left(2\|v\|^2 - (v_1 + v_2 + v_3 + v_4) \right) g + o\left(e^{-1/\sqrt{T}} \right) \right), \end{aligned}$$

which implies that

$$\left| v^\top \nabla_\theta L_1 \right| = O\left(\frac{p_2(D) p_{\mathrm{conf}} g}{T} \right). \tag{28}$$

Plugging (26), (27) and (28) into (25) and assuming that $g = o(1)$ (if $g = \Omega(1)$ then it is easy to see $p_1 \geq 0.999$), we finally obtain that

$$
\begin{aligned}
g_{k+1} - g_k &= \frac{2p_2(D)p_{\text{conf}}\eta_2}{T}\left((v_1^2+v_3^2)(v_2-v_4)^2 + (v_2^2+v_4^2)(v_1-v_3)^2 + o(g)\right) + O(\|\epsilon\|) \\
&\geq \frac{2p_2(D)p_{\text{conf}}\eta_2}{T}\left(\sqrt{(v_1^2+v_3^2)(v_2^2+v_4^2)}g + o(g)\right) \\
&\geq \frac{\eta_2 p_{\text{conf}}}{10000T}\cdot g
\end{aligned} \tag{29}
$$

for sufficiently small T and $p_2(D) > 0.0009$ (If $p_2(D) \leq 0.0009$ then we have $p_1(D) \geq 0.9991$). The first inequality uses the fact that

$$
\begin{aligned}
\|\epsilon\| &\lesssim \eta_2^2\|\nabla_\theta L_1\|^2 \\
&\lesssim \eta_2^2\|\nabla_v L_1\|^2 \\
&\lesssim \frac{p_2^2 p_{\text{conf}}^2\eta_2^2}{T^2}\left((v_1-v_3)^2 + (v_2-v_4)^2\right) \\
&= o\left(\frac{p_2 p_{\text{conf}}\eta_2}{T}\left((v_1^2+v_3^2)(v_2-v_4)^2 + (v_2^2+v_4^2)(v_1-v_3)^2\right)\right)
\end{aligned}
$$

and $(v_1^2+v_3^2)(v_2-v_4)^2 + (v_2^2+v_4^2)(v_1-v_3)^2 \geq 2\sqrt{(v_1^2+v_3^2)(v_2^2+v_4^2)}g$.

Hence $g_{k+1} - g_k \geq \frac{\eta_2 p_{\text{conf}}}{10000T} g_k$ as long as $p_1(D) < 0.9991$. This means $g_{k+1} \geq \left(1 + \frac{\eta_2 p_{\text{conf}}}{10000T}\right) g_k$. Since we have $g_0 \gtrsim T^4$, there exists constant C_2, such that when $k \geq \frac{C_2 T\log(1/T)}{\eta_2 p_{\text{conf}}}$, we have $p_1(D) \geq 0.9991$. It is straightforward to verify that at time k we have

$$
\frac{1}{1+\exp(-g_k/T) + 4\exp(-\sqrt{\frac{1}{T}}) + n^2\exp(-\frac{1}{18T})} \leq p_1(Z') \leq \frac{1}{1+\exp(-g_k/T)}
$$

for any confusing sequence Z'. Therefore for $T < \frac{1}{40\log n}$ and large enough n we have $|p_1(Z') - p_1(Z)| < 0.0001$, which implies $p(r_1) \geq 0.999$ for any confusing sequence. □

Lemma C.3 shows that the test accuracy is high on confusing sequences. To prove Theorem 4.4, it remains to show that the accuracy is also high on non-confusing sequences.

Proof of Theorem 4.4. We first show that after Stage 2, $p(r_1) > 0.99$ for any non-confusing sequence as well. We know that after Stage 2, $\|v(\theta) - v(\tilde{\theta})\| \lesssim \sqrt{T}(\log\frac{1}{T})^{\frac{3}{4}} = o(\alpha) = o(\frac{p_{\text{mis}}\cdot\eta_1}{T})$. Hence by Lemma C.1, we have $p(r_1) > 0.999$ for non-confusing sequences if p_{mis} is a constant. By Lemma C.3, we also need to show that p_{conf} cannot be too small. Apply Chernoff bound to Assumption 4.3, we have that if $|D| \geq \frac{13}{\zeta}\log\frac{1}{\delta}$, then with probability at least $1-\delta/3$, we have $p_{\text{conf}}, p_{\text{mis}} \geq \zeta/2$. Taking a union bound over the randomness here and the randomness in Lemma C.1, Lemma C.3, we finish the proof. □

D. Proof to Lemma 4.5

In this section, we denote $q = \text{softmax}(\omega)$ the attention scores from the first decoding token. We abbreviate $p_{2,r}(a, Z)$ as $p_{2,r}(a)$ if Z can be inferred from the context. We restate Lemma 4.5 here.

Lemma D.1. *Assume the IC-recall data satisfies Assumptions 2.2 and 4.3. Set T, η_1, η_2 and the sample size $|D|$ same as in Theorem 4.1 for Algorithm 1, after Stage 1 and throughout Stage 2, the transformer has accuracy $1 - o\left(\exp\left(-\frac{1}{\sqrt{T}}\right)\right)$ on all IC-recall sequences for the second decoding step.*

Proof. We fix any sampled $\widetilde{Z} = (s_1, a_1, s_2, a_2, s_3, u_{\text{EoS}})$. The prediction logits for the second decoding step conditioned on that the first decoded token being r_1 are

$$
l(a_1) = (q_1 + (q_7+1))^2 + q_3^2 + q_5^2,
$$

$$l(a_2) = (q_3 + (q_7+1))^2 + q_1^2 + q_5^2,$$
$$l(a_3) = (q_5 + (q_7+1))^2 + q_3^2 + q_1^2$$

and $l(a) = q_1^2 + q_3^2 + q_5^2 + (q_7+1)^2$ for any answer $a \notin \{a_1, a_2, a_3\}$. We see that the loss for the second decoding step as a function of q is identical for every sequence, and hence it suffices to analyze a single sequence. We can also see that the gap between $l(a_3)$ and the logits of other answers is lower bounded by $2(q_5 - \max\{q_1, q_3\})$. Our goal is to show this lower bound of the gap becomes large enough after Stage 1, and remains large throughout Stage 2.

We first calculate the gradient of the loss $\ell_2(\omega, \widetilde{Z})$ w.r.t. attention scores q_1, q_3 and q_5 in the following claim.

Claim 2. The gradients w.r.t. q are

$$\frac{\partial \ell_2}{\partial q_1} = \frac{2(1+q_7)}{T} p_{2,r_1}(a_1),$$

$$\frac{\partial \ell_2}{\partial q_3} = \frac{2(1+q_7)}{T} p_{2,r_1}(a_2),$$

$$\frac{\partial \ell_2}{\partial q_5} = \frac{2(1+q_7)}{T} \left(p_{2,r_1}(a_3) - 1\right),$$
$$\frac{\partial \ell_2}{\partial q_7} = \frac{2}{T} \left(p_{2,r_1}(a_1) q_1 + p_{2,r_1}(a_2) q_3 + p_{2,r_1}(a_3) q_5 - q_5\right),$$
$$\frac{\partial \ell_2}{\partial q_i} = 0 \text{ for } i \neq 1, 3, 5, 7.$$

We have $\frac{\partial \ell_2}{\partial \omega_i} = q_i(\frac{\partial \ell_2}{\partial q_i} - \sum_{j=1}^7 q_j \frac{\partial \ell_2}{\partial q_j})$ and at initialization $q_i = \frac{1}{7}$ for all $i \in [7]$, $p_{2,r_1}(a_1) = p_{2,r_1}(a_2) = p_{2,r_1}(a_3)$. Therefore at initialization we have

$$\nabla_q \ell_2 = \frac{1}{7T} \begin{pmatrix} 16p \\ 0 \\ 16p \\ 0 \\ 16(p-1) \\ 0 \\ 2(3p-1) \end{pmatrix} \Longrightarrow \nabla_\omega \ell_2 = \frac{1}{343T} \begin{pmatrix} 58p+18 \\ -54p+18 \\ 58p+18 \\ -54p+18 \\ 58p-94 \\ -54p+18 \\ -12p+4 \end{pmatrix}$$

where $p := p_{2,r_1}(a_1) < \frac{1}{3}$. Therefore after the first GD step, we have

$$\tilde{\omega}_5 - \max_{i \neq 5} \tilde{\omega}_i \geq \frac{32\eta_1}{147T} = \Theta(\sqrt{T}\log(1/T)).$$

Since the perturbation is of radius $\Theta(T^3 \log^{-2}(1/\delta))$, for sufficeintly small T, after the perturbation the parameter ω continues to satisfy $\omega_5 - \max_{i \neq 5} \omega_i = \Theta(\sqrt{T}\log(1/T))$.

Next, we show that if $\omega_5 - \max_{i \neq 5} \omega_i > 0$, then after a GD step this quantity is non-decreasing. It is immediate to see $\frac{\partial \ell_2}{\partial q_i} - \frac{\partial \ell_2}{\partial q_5} \geq 0$ for $i = 1, 2, 3, 4, 6$. Next, since $\omega_5 \geq \omega_i$, $q_5 \geq q_i$ and thus

$$\frac{\partial \ell_2}{\partial q_7} \geq \frac{2q_5}{T}(p_{2,r_1}(a_3) - 1) \geq \frac{2(1+q_7)}{T}(p_{2,r_1}(a_3) - 1) = \frac{\partial \ell_2}{\partial q_5}$$

as well. Finally,

$$\begin{aligned} \frac{\partial \ell_2}{\partial \omega_i} - \frac{\partial \ell_2}{\partial \omega_5} &= q_i \frac{\partial \ell_2}{\partial q_i} - q_5 \frac{\partial \ell_2}{\partial q_5} + (q_5 - q_i) \sum_{j=1}^7 q_j \frac{\partial \ell_2}{\partial q_j} \\ &\geq q_i \frac{\partial \ell_2}{\partial q_i} - q_5 \frac{\partial \ell_2}{\partial q_5} + (q_5 - q_i) \frac{\partial \ell_2}{\partial q_5} \\ &= q_i \left(\frac{\partial \ell_2}{\partial q_i} - \frac{\partial \ell_2}{\partial q_5} \right) \\ &\geq 0. \end{aligned}$$

Hence $\omega_5 - \max_{i\neq 5}\omega_i$ is non-decreasing. Therefore at any point during Stage 2, we have

$$q_5 - q_i = q_5(1 - \frac{q_i}{q_5}) = q_5(1 - \exp(\omega_i - \omega_5)) \geq \Theta(\sqrt{T}\log(1/T))$$

for sufficiently small T since $q_5 \geq 1/7$. This implies the logit gap $l(a_3) - \max\{l(a_1), l(a_2)\} = \Omega(\sqrt{T}\log\frac{1}{T})$, as well as $l(a_3) - l(a) \geq 2q_5 = \Omega(1)$ for any $a \notin \{a_1, a_2, a_3\}$. Therefore for $\frac{1}{T} > 40\log n$, we have

$$p_{2,r_1}(a_3) = \frac{1}{1 + 2\exp\left(-\Omega\left(\frac{1}{\sqrt{T}}\log\frac{1}{T}\right)\right) + (n-3)\exp\left(-\Omega\left(\frac{1}{T}\right)\right)} \geq 1 - o\left(\exp\left(-\frac{1}{\sqrt{T}}\right)\right).$$

Proof to Claim 2. Conditioned on the correct decoded relation r_1, we have the loss for the second decoding step $\ell_2 = -\log\left(p_{2,r_1}(a_3)\right)$ where $p_{2,r_1}(a_3) = \frac{\exp(l(a_3)/T)}{\sum_{a\in\mathcal{A}}\exp(l(a)/T)}$. Therefore we have the gradient w.r.t. the logit vector l is

$$\frac{\partial \ell_2}{\partial l} = \frac{1}{T}(p_{2,r_1} - e_3).$$

Also we know that

$$\begin{aligned} l(a_1) &= (q_1 + (q_7+1))^2 + q_3^2 + q_5^2, \\ l(a_2) &= (q_3 + (q_7+1))^2 + q_1^2 + q_5^2, \\ l(a_3) &= (q_5 + (q_7+1))^2 + q_3^2 + q_1^2 \end{aligned}$$

and $l(a) = q_1^2 + q_3^2 + q_5^2 + (q_7+1)^2$ for any answer $a \notin \{a_1, a_2, a_3\}$. Therefore we can obtain $\frac{\partial l_i}{\partial q_j} = 0$ if $j \neq 1, 3, 5, 7$,

$$\frac{\partial l_i}{\partial q_{2j-1}} = 2q_{2j-1} + 2(q_7+1)\delta_{ij} \text{ for any } i \in [n], j = 1, 2, 3$$

and

$$\frac{\partial l_i}{\partial q_7} = 2(q_7+1) + 2q_{2i-1}\cdot \mathbf{1}_{\{i\leq 3\}} \text{ for any } i \in [n].$$

Since $\sum_i p_{2,r_1}(a_i) = 1$, we have

$$\frac{\partial \ell_2}{\partial q_{2j-1}} = \sum_{i\in[n]} \frac{\partial \ell_2}{\partial l_i}\frac{\partial l_i}{\partial q_{2j-1}} = \frac{2(q_7+1)}{T}\left(p_{2,r_1}(r_1(s_j)) - \delta_{j3}\right)$$

and

$$\frac{\partial \ell_2}{\partial q_7} = \frac{2}{T}(p_{2,r_1}(r_1(s_1))q_1 + p_{2,r_1}(r_1(s_2))q_3 + p_{2,r_1}(r_1(s_3))q_5 - q_5).$$

Plugging into the mapping of r_1, we obtain the desired results. For $j \neq 1, 3, 5, 7$, we have $\frac{\partial \ell_2}{\partial q_j} = 0$. □

□

E. Helper lemmas

Proposition E.1. *There exists constant T_0 such that for any $0 < T < T_0$, the loss $L_1(\theta, T)$ has gradient Lipschitz constant $O(1/T^2)$ and Hessian Lipschitz constant $O(1/T^3)$.*

Proof. We prove the proposition for the general setting where $|\mathcal{S}| = n$ and $|\mathcal{R}| = m$. Recalling that loss $L_1(\theta, T) = \mathbb{E}_Z[\ell_1(\theta, Z, T)]$, it suffices to show for $\ell_1(\theta, Z, T)$ that the desired property holds. Also note that since $v = \mathrm{softmax}(\theta)$, its derivatives w.r.t. θ up to order 3 are uniformly bounded by absolute constants. Therefore by chain rule, it suffices to show for $\ell_1(v, Z, T)$ that its gradient and Hessian Lipschitz constants are $O(1/T^2)$ and $O(1/T^3)$ respectively for any Z. We fix any Z. Then by (16) we know each logit $l(r_j)$ can be written as some quadratic form

$$l(r_j) = v^\top A_j v$$

for some symmetric matrix A_j, and the number of possible quadratic forms is at most 13, which is independent of n. For simplicity, we abbreviate $l(r_j)$ as l_j. Then on the simplex of v, there exists absolute constants $G, H > 0$ such that

$$\|\nabla_v l_j(v)\| \le G, \quad \|\nabla_v^2 l_j(v)\| \le H, \quad \nabla_v^3 l_j(v) = 0 \text{ for all } j \in [m].$$

We know that the loss

$$\ell_1(v, T) = -\log p_1(v) = -\frac{l_1(v)}{T} + \log \sum_{j=1}^{m} \exp\left(\frac{l_j(v)}{T}\right).$$

Define the log-sum-exp term as $\Psi(v) := \log \sum_{j=1}^{m} \exp\left(\frac{l_j(v)}{T}\right)$. For a k-th order tensor $A \in \mathbb{R}^{d\times\cdots\times d}$, we define the associated k-linear form as

$$A[x_1, \ldots, x_k] := \sum_{i_1,\ldots,i_k=1}^{d} A_{i_1,\ldots,i_k} \cdot (x_1)_{i_1} \cdots (x_k)_{i_k}, \quad x_1, \ldots, x_k \in \mathbb{R}^d.$$

For unit vectors a, b we define

$$X_j^a := \nabla_v l_j[a], \quad Y_j^{a,b} := \nabla^2 l_j[a, b].$$

Then we have $|X_j^a| \le G$ and $|Y_j^{a,b}| \le H$. We first derive the upper bound for the gradient.

Gradient Lipschitz constant. for all $j \in [m]$. Differentiating Ψ gives

$$\nabla\Psi(v)[a] = \frac{1}{T} \sum_{j\in[m]} p_j X_j^a$$

and

$$\nabla^2\Psi(v)[a, b] = \frac{1}{T} \sum_{j\in[m]} p_j Y_j^{a,b} + \frac{1}{T^2}\mathrm{Cov}_p(X^a, X^b).$$

Here $p = \mathrm{softmax}(l)$ is the prediction probability vector and

$$\mathrm{Cov}_p(X, Y) := \sum_{j=1}^{m} p_j X_j Y_j - \left(\sum_{j=1}^{m} p_j X_j\right) \cdot \left(\sum_{j=1}^{m} p_j Y_j\right) \text{ for any } X, Y \in \mathbb{R}^m.$$

It is easy to see that

$$|\nabla\Psi(v)[a]| \le \frac{G}{T} = O(1/T)$$

and there exists $T_0 > 0$ such that for any $0 < T < T_0$ we have

$$|\nabla^2\Psi(v)[a, b]| \le \frac{H}{T} + \frac{2G^2}{T^2} = O(1/T^2).$$

Also noting that $|\nabla^2 l_1[a, b]| \le H$, therefore we have

$$|\nabla^2\ell_1[a, b]| = O(1/T^2),$$

which implies that

$$\|\nabla^2\ell_1(v)\| = O(1/T^2).$$

Hessian Lipschitz constant. We can calculate to obtain that for unit vectors a, b, c, we have

$$\begin{aligned}\nabla^3\Psi(v)[a, b, c] = &\frac{1}{T} \sum_{j\in[m]} p_j \nabla^3 l_j[a, b, c] \\ &+ \frac{1}{T^2}\left(\mathrm{Cov}_p(Y^{a,b}, X^c) + \mathrm{Cov}_p(Y^{a,c}, X^b) + \mathrm{Cov}_p(Y^{b,c}, X^a)\right) \\ &+ \frac{1}{T^3}\sum_{j=1}^{m} p_j \left(X_j^a - \mu_a\right)\left(X_j^b - \mu_b\right)\left(X_j^c - \mu_c\right)\end{aligned}$$

where $\mu_a := \sum_{j\in[m]} p_j X_j^a$ and μ_b, μ_c similarly follow. Using $\nabla^3 l_j = 0$, $|X_j^a|, |X_j^b|, |X_j^c| \le G$ and $|Y^{a,b}|, |Y^{b,c}|, |Y^{a,c}| \le H$ for all $j \in [m]$, we obtain

$$|\nabla^3 \Psi(v)[a,b,c]| \le \frac{6HG}{T^2} + \frac{8G^3}{T^3} = O(1/T^3).$$

Therefore combining it with $\nabla^3 l_1 = 0$ we have

$$\left|\nabla^3 \ell_1[a,b,c]\right| = O(1/T^3),$$

which implies that

$$\|\nabla^3 \ell_1(v)\| = O(1/T^3).$$

□

Lemma E.2 (GD movement upper bound)**.** *Let $f(\cdot)$ be ℓ-gradient Lipschitz smooth and $f^\star = \inf_x f(x)$. Consider gradient descent iterates $x_{k+1} = x_k - \eta \nabla f(x_k)$ where $\eta \le \frac{1}{\ell}$. Then for any $t \ge 1$,*

$$\|x_t - x_0\| \le \sqrt{\frac{2t\,(f(x_0) - f^\star)}{\ell}}\,.$$

Proof. By the descent lemma, $\sum_{k=0}^{t-1} \|x_{k+1} - x_k\|^2 = \eta^2 \sum_{k=0}^{t-1} \|\nabla f(x_k)\|^2 \le \frac{2(f(x_0)-f^\star)}{\ell}$. Applying the triangle inequality followed by Cauchy–Schwarz,

$$\|x_t - x_0\| \le \sum_{k=0}^{t-1} \|x_{k+1} - x_k\| \le \sqrt{t \sum_{k=0}^{t-1} \|x_{k+1} - x_k\|^2} \le \sqrt{\frac{2t\,(f(x_0) - f^\star)}{\ell}}.$$

□