Transformers Provably Learn Sparse Token Selection While Fully-Connected Nets Cannot

Zixuan Wang [1] **Stanley Wei** [1] **Daniel Hsu** [2] **Jason D. Lee** [1]

Abstract

The transformer architecture has prevailed in various deep learning settings due to its exceptional capabilities to select and compose structural information. Motivated by these capabilities, Sanford et al. (2023) proposed the *sparse token selection* task, in which transformers excel while fully-connected networks (FCNs) fail in the worst case. Building upon that, we strengthen the FCN lower bound to an average-case setting and establish an algorithmic separation of transformers over FCNs. Specifically, a one-layer transformer trained with gradient descent provably learns the sparse token selection task and, surprisingly, exhibits strong out-of-distribution length generalization. We provide empirical simulations to justify our theoretical findings.

1. Introduction

In modern deep learning, transformer networks have established themselves as a fundamental building block, showcasing their versatility across diverse tasks such as language modeling (OpenAI, 2023), computer vision (Dosovitskiy et al., 2020), and reinforcement learning (Jumper et al., 2021). At the core of transformers is the self-attention layer (Vaswani et al., 2017), a critical component assigning varying attention weights to different segments of the input sequence by discerning relevance between tokens.

The success of transformers is closely tied to their representational capabilities in extracting structural information encoded in token embeddings. Empirical observations reveal that transformers trained with GD-based algorithms exhibit biases towards certain algorithmic solutions in some arithmetic tasks (Edelman et al., 2022; Liu et al., 2022; Yao et al., 2021; Nanda et al., 2023). However, few works have presented rigorous mathematical evidence that substantiates their superiority over alternative architectures.

In recent work, Sanford et al. (2023) introduced a simple task known as the *q-sparse averaging* for an input context sequence, where the target output is an average of a q-subset of the input tokens, specified in the input. When the input context length is T, the q-sparse average was demonstrated to be effectively approximated using a $O(\log T)$-dimensional self-attention layer, whereas any fully-connected network seeking to approximate this task requires a first-layer width of at least $\Omega(T)$ in the worst case. This crucial finding implies that, in theory, the sparse-average task potentially induces an *exponential* separation between transformer models from the fully-connected neural networks (FCNs) with respect to the context length T, further reinforcing existing empirical findings that the inductive bias of transformers merits strength in approximating certain arithmetic tasks.

However, these results are limited to the *expressive power* of transformers and do not inherently guarantee that transformers can be *trained* with standard gradient-based methods to achieve such approximation capability as its expressive power would suggest. Thus, a pertinent and natural question beyond theoretical expressiveness arises:

Q: Does the expressivity separation between transformers and FCNs translate to learnability?

Our results provide an affirmative answer to this question. In our work, we focus on the *sparse token selection task* (STS_q). In this task, every token in the input sequence follows the standard Gaussian distribution, and a token subset of size q is drawn uniformly at random. Importantly, we initiate an exploration into the training dynamics of a one-layer transformer with $\mathrm{softmax}$ attention using GD for the STS_q. We first study the transformer's training dynamics under GD, where the goal is to minimize the expected loss over the input distribution. Notably, we characterize the global convergence of the transformer with the stochastic positional encoding introduced in Shen et al. (2023). Additionally, we delve into the provable length generalization capacity of the transformer learned with GD. We also establish new lower bounds for the capacity of FCNs to approximate the

[1]Department of Electrical and Computer Engineering, Princeton University, NJ, USA [2]Department of Computer Science, Columbia University, NY, USA. Correspondence to: Zixuan Wang <wangzx@princeton.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

sparse token selection task within the data distribution. Empirical simulations in practical settings of our architecture verify our theoretical findings and moreover demonstrate the advantages of stochastic positional encoding over a fixed absolute positional encoding in length generalization ability.

1.1. Our contributions

STS_q is efficiently learnable. We establish a gradient descent convergence guarantee for a one-layer transformer employing a stochastic positional encoding where only $O(d + q \log T)$ width is necessary (Section 3.2). Under mild conditions on the data distribution, initialization, and hyperparameters, we prove that running GD on a one-layer transformer globally converges when both layers are jointly trained with the same learning rate.

FCN cannot express STS_q. Complementing our efficient learnability results, we show a separation between FCNs and one-layer transformer networks: all FCNs (regardless of depth or activation function) that can approximate the task must have $\Omega(Td)$ neurons in the first layer, which is exponentially larger than the $O(d + q \log T)$ width used in the transformer.

Length generalization on STS_q . We investigate the *length generalization* performance of the trained model with stochastic positional encoding, using out-of-distribution data on longer sequences. Based on the global convergence result, we prove that the length generalization loss also converges to zero. Our experiments demonstrate that when the in-distribution training loss converges to zero, the OOD loss also tends to zero for the stochastic positional encoding, but strictly nonzero using fixed positional encoding.

1.2. Related works

Expressiveness of transformers. The transformer architecture (Vaswani et al., 2017) has been long adopted as the fundamental building block in many recent large language models such as GPT (Brown et al., 2020; OpenAI, 2023) and Llama (Touvron et al., 2023a;b). Recent works have begun to study the limitations and strengths of the transformer architecture from a theoretical perspective (Yun et al., 2019; Pérez et al., 2019; Yao et al., 2021; Bhattamishra et al., 2020b; Zhang et al., 2022; Liu et al., 2022; Hahn, 2020; Bhattamishra et al., 2020a). One such direction considers its universal approximation power (Yun et al., 2019; Bhattamishra et al., 2020a;b; Dehghani et al., 2018), similar to the universal approximation results for fully-connected neural networks. More recent works have focused on analyzing its expressive power on certain statistical tasks (Edelman et al., 2022; Elhage et al., 2021; Likhosherstov et al., 2021; Akyürek et al., 2022; Zhao et al., 2023; Yao et al., 2021; Anil et al., 2022; Barak et al., 2022) as well as in-context learning settings (Dong et al., 2022; Huang et al., 2023). In another line of literature, transformer layers are used to represent gradient descent steps for certain learning tasks (Bai et al., 2023; Garg et al., 2022; Von Oswald et al., 2023; Olsson et al., 2022; Akyürek et al., 2022; Panigrahi et al., 2023; Sanford et al., 2023). In a recent work, Sanford et al. (2023) introduced a sparse average task where recurrent neural networks and fully connected neural networks (FCNs) all have memory complexity scaling polynomially in the input sequence length, while a one-layer transformer has a construction for only $\log T$-width on the task. In all of these works, however, the training process of transformers is not considered; rather, the focus was on the representation power of transformers. Building upon the setting of Sanford et al. (2023), our work provides not only a generalized result for the representational power by extending from worst-case to average-case but also investigates the algorithmic aspect of training transformers: we show that the same exponential separation is attainable when using gradient descent on an architecture with stochastic positional encoding.

Training dynamics of transformers. Several works in the literature have studied the learnability of certain transformer models. Jelassi et al. (2022) showed a Vision Transformer (ViT) (Dosovitskiy et al., 2020) trained by GD with positional-embedding attention matrix can learn spatial structure. Li et al. (2023) analyzed the sample complexity required for achieving good generalization on a similar ViT model. However, their results both hinged on a warm start of initialization near the target pattern, which is a practically infeasible assumption. Tarzanagh et al. (2023) established an equivalence between the optimization geometry of self-attention and a hard-margin SVM problem that separates and selects optimal tokens using linear constraints and established global convergence under strong assumptions. Tian et al. (2023a) revealed how the self-attention layer combines input tokens in an SGD-trained transformer, and Tian et al. (2023b) explored the training procedure of multilayer transformers by focusing on the dynamics of MLP layers; however, neither give a provable guarantee for convergence.

Another line of research focuses on the training dynamics of in-context learning. Mahankali et al. (2023) first introduced linear regression as an in-context learning task and showed that a one-layer transformer minimizing the pre-training loss is implementing single-step gradient descent. Zhang et al. (2023) considered a single-layer linear self-attention layer on this linear regression task and proved global convergence for gradient flow. Huang et al. (2023) first proved GD global convergence of a one-layer transformer with *softmax* attention on the linear regression in-context learning task where in-context tokens are drawn from certain distribution. Chen et al. (2024) extended the single-task linear regression task to a multi-task setup and demonstrated the optimal global convergence of a multi-head attention architecture by applying gradient flow to the population loss with a particular

initialization scheme. Nichani et al. (2024) theoretically verifies that a simplified two-layer transformer can learn the induction head and generalize it to some in-context latent causal structures. Li et al. (2023); Tian et al. (2023a); Zhang et al. (2023); Huang et al. (2023); Tarzanagh et al. (2023) and our work share a similar reparameterization technique, combining the key and query matrices $\boldsymbol{K}, \boldsymbol{Q}$ into one matrix $\boldsymbol{W}$ to simplify the dynamics of the training process.

Length generalization and positional encoding. Length generalization is a major challenge for transformers (Newman et al., 2020; Dubois et al., 2019; Anil et al., 2022; Ruis et al., 2020; Hupkes et al., 2020; Kazemnejad et al., 2023; Zhou et al., 2023). Several methods have been proposed to mitigate this issue, including using linear bias (Press et al., 2021), EOS detection (Newman et al., 2020), or scratchpads and chain-of-thought (Nye et al., 2021; Wei et al., 2022; Anil et al., 2022; Liu et al., 2022).

Recently, Kazemnejad et al. (2023) empirically investigated length generalization of different positional encodings. They show that many commonly used schemes such as APE (Vaswani et al., 2017), ALiBi (Press et al., 2021), and rotary (Su et al., 2021) are ill-suited for length generalization, while T5's relative PE (Raffel et al., 2020) works for downstream tasks. In addition, they also showed that transformers without positional encoding (Tsai et al., 2019; Haviv et al., 2022) outperform all explicit positional encoding schemes. Meanwhile, Shen et al. (2023) and Ruoss et al. (2023) both introduced some randomized positional encoding; Ruoss et al. (2023) simulated the positions of longer sequences and randomly sampled a sorted subset to fit the sequence's length; Shen et al. (2023) used some random positional encoding to represent indicators for tokens positions. Zhou et al. (2024) additionally confirmed that randomized positional encoding enhances the length generalization capabilities of transformers in specific effective configurations. All the schemes above significantly improve length generalization, inspiring us in the same spirit to consider a stochastic positional encoding for our theoretical analysis.

1.3. Outline of this paper

The outline of our paper is as follows. In Section 2 we formalize the problem setting, including our STS_q task definition, the positional encoding, and the one-layer transformer architecture. Section 3 contains our main results, consisting of our gradient descent global convergence results, the length generalization theoretical guarantee, and the approximation lower bound for FCNs on expected loss. Section 4 provides simulations for global convergence of GD, interprets the learned parameter pattern, and empirically verifies the length generalization advantage from Section 3.

2. Settings

In this section, we present our notations and problem formulations, including the one-layer transformer architecture and positional encoding definitions of our paper.

Notations: We use $[T]$ to denote the set $\{1, 2, ..., T\}$. Matrices are represented in upper-case bold letters ($\boldsymbol{X}, \boldsymbol{W}$, etc.), and vectors are in lower-case bold letters ($\boldsymbol{x}, \boldsymbol{e}$, etc.). For norm, $\|\cdot\|$ denotes ℓ_2 norm and $\|\cdot\|_F$ denotes the Frobenius norm. For vector $\boldsymbol{v}$, we use $\boldsymbol{v}_k$ to denote the k-th entry of vector $\boldsymbol{v}$. For a matrix $\boldsymbol{W}$, we use $\boldsymbol{W}[:, i]$ to denote its i-th column vector. We use $\mathbb{1}\{\cdot\}$ as the indicator function. We use $\tilde{O}(\cdot)$ to hide logarithmic factors.

2.1. q-sparse token selection task STS_q

We simplify the qSA framework in Sanford et al. (2023) to STS_q: only one query subset y is fed into the model to compute the sparse average instead of T query subsets.[1] The objective is to train the parameterized model using a gradient-based algorithm to approximate the STS_q task given a certain data distribution. Similar to qSA in Sanford et al. (2023), this task is designed to showcase the ability of self-attention units to capture and aggregate dependencies between input tokens, especially the positional information. STS_q highlights two salient features of attention matrices observed in practice: they are sparse, and the sparsity pattern depends on the input (Likhosherstov et al., 2021).

Definition 2.1. For sparsity q, token dimension d, and input dimension $dT + q$, consider the input $(\boldsymbol{X}, y) = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T; y_1, y_2, ..., y_q) \in \mathbb{R}^{dT+q}$ where the input tokens $\boldsymbol{x}_i \in \mathbb{R}^d$ and the query $y \in \binom{[T]}{q}$ is a q-element subset of $[T]$. Define the q-sparse token selection $\mathrm{STS}_q(\cdot)$ as

$$\mathrm{STS}_q(\boldsymbol{X}, y) = \frac{1}{q} \sum_{j=1}^{q} \boldsymbol{x}_{y_j}$$

Data distribution: We consider samples $(\boldsymbol{X}, y) \in \mathbb{R}^{dT+q}$ from the following distribution $\mathcal{D}_{T,q}$ (T is the sequence length, q is the subset size): The input tokens $\boldsymbol{x}_i, i = 1, 2, ..., T$ are sampled from standard Gaussian distribution, and the q-sparse subset y containing all the averaging indices is uniformly sampled from all q-subsets of $[T]$.

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T), \boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d),$$

$$y \sim \mathrm{Unif}\left(\binom{[T]}{q}\right), i \in [T]$$

[1] Under the population loss we are training on, the sparse token selection problem is equivalent to the original problem. For details, see Appendix A.2.

2.2. Positional encodings

One of the key features of transformers is the positional encoding (PE). In our STS_q task, the positional encoding turns out to be necessary for the transformer to maintain the positional information in the input sequence. In this paper, we consider two different PEs: the one-hot and the near-orthogonal positional encoding.

Definition 2.2 (One-hot PE)**.** The one-hot positional encoding for position $i \in [T]$ is:

$$\boldsymbol{e}_i = (\mathbb{1}\{i=1\}, \mathbb{1}\{i=2\}, \cdots, \mathbb{1}\{i=T\})$$

The positional encoding matrix is $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_T] = \boldsymbol{I}_T$, with each column vector $\boldsymbol{e}_i$ as the PE of the i-th token.

Remark. For clarity, we stress that $\boldsymbol{e}_i$ is equal to the i-th elementary basis vector only when using the one-hot PE.

Definition 2.3 (Near-orthogonal PE)**.** A near-orthogonal positional encoding is such that for all positions $i \in [T]$, we have $\boldsymbol{e}_i \in \{\pm 1/\sqrt{d_e}\}^{d_e}$, and moreover, when we denote the positional encoding matrix $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_T] \in \mathbb{R}^{d_e \times T}$, we have, for some $\delta \in (0, 1/2)$, that $d_e = \Theta(q \log T/\delta^2)$ and $|\langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle| \leq \delta$ for any $i \neq j$, with the i-th column vector as the PE of the i-th token.

The existence of such a set of near-orthogonal positional encoding is guaranteed in Lemma A.2 (Lemma 12 in Sanford et al. (2023)) by showing the existence of Rademacher random matrices $\boldsymbol{E}$ satisfying the (q, δ)-*restricted isometry property (RIP).* Related background of the near orthogonality and restricted isometry property refers to Appendix A.

Encoding for subset y: To utilize the positional information, the query subset y needs some encoding $\boldsymbol{e}_y$ based on the positional encoding $\boldsymbol{E}$. For one-hot encoding, we consider $\boldsymbol{e}_y = \sum_{i \in y} \boldsymbol{e}_i$ as the summation of one-hot encoding for all indices in y. The encoding scheme for y can separate the positional encoding vector $\boldsymbol{e}_i$ with index $i \in y$ and those that are not: $\langle \boldsymbol{e}_y, \boldsymbol{e}_i \rangle = 1, \forall i \in y; \langle \boldsymbol{e}_y, \boldsymbol{e}_i \rangle = 0, \forall i \notin y$.

For the near-orthogonal positional encoding when $y = \{y_i\}_{i=1}^q$, we consider the following encoding based on the RIP of $\boldsymbol{E}$:

$$\boldsymbol{e}_y = \boldsymbol{E}_y (\boldsymbol{E}_y^\top \boldsymbol{E}_y)^{-1} \mathbf{1}_q \in \mathbb{R}^{d_e}$$
$$\boldsymbol{E}_y = \left[\boldsymbol{e}_{y_1}, \boldsymbol{e}_{y_2}, \cdots, \boldsymbol{e}_{y_q}\right] \in \mathbb{R}^{d_e \times T}$$

Here, $\boldsymbol{E}_y$ is the concatenation for near-orthogonal PE for indices in y. By Lemma A.3, we know the encoding $\boldsymbol{e}_y$ can separate the column vectors with index in y and all other columns: $\langle \boldsymbol{e}_y, \boldsymbol{e}_i \rangle = 1, \forall i \in y; |\langle \boldsymbol{e}_y, \boldsymbol{e}_i \rangle| \leq \frac{\delta}{1-2\delta}, \forall i \notin y$.

For the input format, we consider the positional encodings $\boldsymbol{E}$ to be concatenated with the input token matrix $\boldsymbol{X}$. We then add the query token $\boldsymbol{z}_{\text{query}} = \left[\boldsymbol{x}_{\text{query}}^\top \;\; \boldsymbol{e}_y^\top\right]^\top$ as an additional query column in the input matrix. The final input matrix for a transformer will be in the following form,

$$[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}] := \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{T-1} & \boldsymbol{x}_T & \boldsymbol{x}_{\text{query}} \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{T-1} & \boldsymbol{e}_T & \boldsymbol{e}_y \end{bmatrix}. \tag{1}$$

which is a $(d + d_e)$ by $(T + 1)$ input matrix.

2.3. One-layer transformer architecture

A simple one-head, one-layer self-attention transformer has the following architecture:

$$f(\boldsymbol{Z}) = \boldsymbol{V}\boldsymbol{Z}\,\mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{K}^\top \boldsymbol{Q}\boldsymbol{Z})$$

where $\boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{m \times d}, \boldsymbol{V} \in \mathbb{R}^{d \times (d+d_e)}$ are the query, key, value matrices, respectively, and m is the width[2] of the self-attention layer parameters $\boldsymbol{Q}, \boldsymbol{K}$. The column-wise softmax operator applied to matrix $\boldsymbol{W} \in \mathbb{R}^{a \times b}$ outputs $\mathrm{softmax}(\boldsymbol{W}) \in \mathbb{R}^{a \times b}$ with:

$$\mathrm{softmax}(\boldsymbol{W})_{i,j} = \frac{\exp(\boldsymbol{W}_{i,j})}{\sum_{i=1}^a \exp(\boldsymbol{W}_{i,j})}$$

Reparameterization: Instead of studying dynamics based on the parameters of key, query, and value matrices $\boldsymbol{K}, \boldsymbol{Q}, \boldsymbol{V}$, we introduce the following reparameterization on the architecture:

1. In the attention layer, we combine $\boldsymbol{Q}, \boldsymbol{K}$ into a single trainable matrix $\boldsymbol{W}^{\boldsymbol{KQ}} \in \mathbb{R}^{(d+d_e) \times (d+d_e)}$.
2. In the attention layer (with parameter $\boldsymbol{W}^{\boldsymbol{KQ}}$), the $(T+1)$-th query token $\boldsymbol{z}_{\text{query}}$ only attends to previous tokens $\boldsymbol{Z}$. The value matrix $\boldsymbol{V}$ also only attends to the $\boldsymbol{Z}$ part.

The first consolidation of the query and key matrices is common in recent theoretical works (Tian et al., 2023a; Huang et al., 2023; Zhang et al., 2023; Jelassi et al., 2022). The second modifications for the $\boldsymbol{W}$ and $\boldsymbol{V}$ are inspired by Huang et al. (2023); Zhang et al. (2023); Chen et al. (2024) to stress the attention between the query token and the input tokens. With those modifications to the architecture, the overall one-layer transformer can be written as follows:

Definition 2.4 (Reparameterization)**.** Define a reparameterized 1-layer self-attention layer with trainable parameter matrix $\boldsymbol{V}, \boldsymbol{W}$ where $\boldsymbol{W} \in \mathbb{R}^{(d+d_e) \times (d+d_e)}, \boldsymbol{V} \in \mathbb{R}^{d \times (d+d_e)}$:

$$\begin{aligned} f_{\boldsymbol{\theta}}(\boldsymbol{X}, y) &= \boldsymbol{V}\boldsymbol{Z}\,\mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{\text{query}}), \\ \boldsymbol{Z} &= [\boldsymbol{X}^\top \;\; \boldsymbol{E}^\top]^\top, \boldsymbol{z}_{\text{query}} = \left[\boldsymbol{x}_{\text{query}}^\top \;\; \boldsymbol{e}_y^\top\right]^\top. \end{aligned} \tag{2}$$

2.4. Stochastic positional encoding

Though a set of fixed near-orthogonal PE may be sufficient for the reparameterized transformer to learn STS_q with

[2]This is called the *embedding dimension* in Sanford et al. (2023).

a specific sequence length and query set size, in practice we would like the trained model to extrapolate beyond the sequence lengths encountered during training. However, experiments in Section 4 show that fixing the positional encoding hinders length generalization. This motivates us to use a stochastic positional encoding module proposed in Shen et al. (2023), in which a stochastic encoding is used for each token, newly generated for each epoch and during testing. They showed that stochastic positional encoding significantly improves length generalization capability on certain arithmetic tasks.

For a one-layer transformer f with freshly sampled random encoding $\boldsymbol{E}$ at each inference, we can combine the stochasticity into the population loss. In expectation, we consider the stochastic architecture $\mathbb{E}[f]$ when training over the population loss. In our setting, we further condition on the event that the positional encoding matrix $\boldsymbol{E}$ has the (q, δ)-restricted isometry and orthogonality property (RIP) for some constant δ, take the conditional expectation over the encoding distribution of the transformer model output, and let this be the stochastic architecture output. For simplicity, we denote the conditional expectation of a random variable $\boldsymbol{\zeta}$ as:

$$\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{\zeta}] = \mathbb{E}_{\boldsymbol{E}}\left[\boldsymbol{\zeta} \middle| \boldsymbol{E} \text{ satisfies } (q,\delta)\text{-RIP}\right]$$

We define a one-layer transformer with stochastic positional encoding as follows.

Definition 2.5 (Transformer with stochastic positional encoding)**.** Define a reparameterized 1-layer self-attention layer with stochastic positional encoding as the following model with trainable parameter matrices $\boldsymbol{V}, \boldsymbol{W}$ where $\boldsymbol{W} \in \mathbb{R}^{(d+d_e)\times(d+d_e)}, \boldsymbol{V} \in \mathbb{R}^{d\times(d+d_e)}$:

$$f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y) = \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{V}\boldsymbol{Z}\,\mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{\text{query}})\right],$$
$$\boldsymbol{Z} = [\boldsymbol{X}^\top \ \boldsymbol{E}^\top]^\top, \boldsymbol{z}_{\text{query}} = \left[\boldsymbol{x}_{\text{query}}^\top \ \ \boldsymbol{e}_y^\top\right]^\top.$$

Training algorithm: To train the neural network on the STS_q task, we minimize the population squared loss for parameterized model $f_{\boldsymbol{\theta}}(\boldsymbol{X})$, similar to Huang et al. (2023) and Zhang et al. (2023):

$$\mathcal{L}_{T,q}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y\sim\mathcal{D}_{T,q}}\left[\|\mathrm{STS}_q(\boldsymbol{X}, y) - f_{\boldsymbol{\theta}}(\boldsymbol{X}, y)\|_2^2\right] \tag{3}$$

The expectation is taken over input tokens $\boldsymbol{x}_i$ and the query subset y. For clarity, we write the expectation as $\mathbb{E}_{\boldsymbol{X},y}[\cdot]$.

For the transformer with stochastic positional encoding, the training objective becomes:

$$\mathcal{L}_{T,q}^{(s)}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y}\left[\|\mathrm{STS}_q(\boldsymbol{X}, y) - f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y)\|_2^2\right]. \tag{4}$$

When it is clear from context, we denote $\mathcal{L}_{T,q}$ and $\mathcal{L}_{T,q}^{(s)}$ as $\mathcal{L}$ and $\mathcal{L}^{(s)}$, respectively. For the learning objective in Equations (3) and (4), we use gradient descent (GD) to train the neural network.

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}(t)) \tag{5}$$

3. Main results

In this section, we present our main results. First, we characterize the convergence of GD on the one-layer architecture introduced in Section 2.3, using one-hot positional encoding as a warm-up example to provide proof intuition (Section 3.1). Then, we consider the architecture with stochastic positional encoding in Section 2.4 and present our main theorem with *exponential* separation (Section 3.2). As a corollary, the transformer with stochastic positional encoding trained with GD provably exhibits *length generalization* capability (Section 3.4). We conclude this section with the expected loss lower bound on FCNs and rigorously establish the claimed exponential separation in expressive power.

3.1. Warm-up: one-hot positional encoding

In this subsection, we study gradient descent convergence on the STS_q with one-hot positional encoding, building up proof intuition for the stochastic positional encoding case. We consider the joint training regime, where GD updates $\boldsymbol{V}$ and $\boldsymbol{W}$ simultaneously with the same learning rate η. With one-hot positional encoding, a one-layer transformer with width $O(T)$ is sufficient, which is a factor of d narrower than the lower bound of $\Omega(Td)$ for FCNs.

We first prove global convergence when $\boldsymbol{V}$ and $\boldsymbol{W}$ are jointly trained with gradient descent. The following theorem characterizes the convergence time of GD. For simplicity, in this subsection we assume $\boldsymbol{x}_{\text{query}} = \boldsymbol{0}_d$. Due to space limits, the proof of Theorem 3.1 is deferred to Appendix D.2.

Theorem 3.1 (Joint training with one-hot positional encoding)**.** *For any $2 \le q < T/4$, $\epsilon \in (0, \frac{dT}{100(T-q)q})$, $\eta \le \frac{1}{20d^2}$, $\boldsymbol{x}_{query} = \boldsymbol{0}_d$, if we run gradient descent on the population loss in Equation (3) with zero initialization $\boldsymbol{W}(0) = \boldsymbol{0}_{(d+T)\times(d+T)}$, $\boldsymbol{V}(0) = \boldsymbol{0}_{d\times(d+T)}$, then after time $t \ge \tilde{O}(\frac{T^2 d}{\eta\epsilon})$, we have $\mathcal{L}(\boldsymbol{\theta}(t)) \le \epsilon$.*

We briefly sketch our proof techniques. We start from deriving the key lemma of the proof, Lemma 3.2. It shows the evolution trajectories of $\boldsymbol{V}(t)$ and $\boldsymbol{W}(t)$ are always along some direction for all time $t \ge 0$.

Lemma 3.2 (Lemma D.5, informal)**.** *Along the gradient descent trajectory, for all $t \ge 0$, there exist some time-dependent scalars $C(t), \alpha(t)$ s.t.:*

$$\boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times T}\\ \boldsymbol{0}_{T\times d} & \left(\boldsymbol{I}_T - \frac{1}{T}\boldsymbol{1}_T\boldsymbol{1}_T^\top\right)\end{bmatrix},$$
$$\boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times T}\end{bmatrix}.$$

The nice property along the gradient descent trajectory is attributed to the symmetry and orthogonality of the one-hot positional encoding. Since y follows uniform distribution over all possible q-subsets of $[T]$ and $\boldsymbol{X}$ follows the Gaussian distribution, we can directly calculate the gradient in each of the entry $\boldsymbol{W}_{ij}(t), \boldsymbol{V}_{ij}(t)$. It can be shown that along the training trajectory on the population objective, except for the position-position block, all other attention blocks are always $\mathbf{0}$. Moreover, the gradient of the position-position block of $\boldsymbol{W}$ always aligns with $\left(\boldsymbol{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^\top\right)$, while the gradient of the token block of $\boldsymbol{V}$ always aligns with $\boldsymbol{I}_d$. Using induction beginning from zero initialization of $\boldsymbol{W}(0) = \mathbf{0}, \boldsymbol{V}(t) = \mathbf{0}$, the lemma holds for all $t \geq 0$.

With Lemma 3.2, the GD dynamics can be approximately reduced to the following ODE on $\alpha(t)$ and $C(t)$:

$$\begin{aligned} \dot{\alpha} &= \eta s_+ \left(1 - \frac{\alpha(Tqs_+^2 - 2qs_+ + 1)}{(T-q)s_+}\right) \\ \dot{C} &= \frac{\eta\alpha d}{T-1} s_+(1-qs_+)\left(1 + \frac{q\alpha}{T-q}(1-Ts_+)\right) \end{aligned} \tag{6}$$

where $s_+ := \mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}(t)\boldsymbol{z}_{\text{query}})_i$. By Lemma 3.2 it is equal to $\mathrm{softmax}(C\boldsymbol{E}^\top \boldsymbol{e}_y) = \frac{1}{q+(T-q)e^{-C}}$ is the attention score of correct positions $i \in y$ for any y. It thus remains to analyze this non-linear ODE.

We first characterize the dynamics for $\alpha(t)$. When s_+ is fixed with a stationary $\alpha^* = \frac{(T-q)s_+}{Tqs_+^2 - 2qs_+ + 1}$, note that $\alpha(t)$ can be seen as a linear ODE. However, the evolution of $C(t)$ leads to a monotonic increment of s_+, making the trajectory of $\alpha(t)$ follow a non-monotonic pattern. Fortunately, we can inductively prove that first, α rapidly grows to a constant c_1 near 1, and subsequently, α can stay above c_1 and below a threshold $\left(1 + c_2\frac{(1-qs_+)}{qs_+(Ts_+-1)}\right)\alpha^*$ with $c_2 < 1$. By the dynamics of $C(t)$, this threshold prevents $C(t)$ from decreasing, and the induction hypothesis enables us to prove that $C(t)$ is monotonically increasing. Note that when s_+ converges to $\frac{1}{q}$ as desired, the threshold naturally converges to 1 and the $\boldsymbol{V}(t)$ converges to the ground-truth $[\boldsymbol{I}_d \;\; \mathbf{0}_{d\times T}]$.

With $\alpha(t)$ bounded, we can analyze the movement of $C(t)$ by lower bounding its increment each iteration. Like in Huang et al. (2023), the increment pattern exhibits a two-stage pattern. In the first stage, $C(t)$ rapidly increases at a linear rate until s_+ reaches $\Theta(\frac{1}{q})$; in the second stage, the growth rate of $C(t)$ gradually decreases while being lower bounded by $\mathrm{poly}(T,d)\cdot\epsilon$ when $\mathcal{L}(t) \geq \epsilon$. Therefore, s_+ converges to $\frac{1}{q}$ at a rate of $O(1/t)$, leading to the final convergence of the model.

3.2. Transformer with stochastic positional encoding

In this subsection, we study the gradient descent convergence on the STS_q problem with stochastic positional encoding. We also consider the joint training case, updating $\boldsymbol{V}(t)$ and $\boldsymbol{W}(t)$ simultaneously with learning rate η.

Though stochastic PE was introduced as an alternative solution due to the failure of fixed absolute PE in length generalization (Shen et al., 2023), it also turns out to help the analysis of the GD training trajectory. The symmetry of the randomized architecture makes it possible to use our intuition from the one-hot encoding case while reducing the transformer width to $\Theta(d + q\log T)$. We now present our main theorem with stochastic PE.

Theorem 3.3 (Joint training with stochastic positional encoding)**.** *Suppose* $q = \Theta(1)$, $\delta < 1/10$, $d_e = \Theta(q\log T/\delta^2)$, $2 \leq q < T/4$, $\epsilon \in (0, \frac{dT}{100(T-q)q})$, $\eta \leq \frac{d_e^2}{40d^2T}$. *If we run GD on the population loss in Equation* (4) *with zero initialization* $\boldsymbol{W}(0) = \boldsymbol{0}_{(d+d_e)\times(d+d_e)}, \boldsymbol{V}(0) = \boldsymbol{0}_{d\times(d+d_e)}$, *then after time* $t \geq \tilde{O}(\frac{T^{\frac{2-2\delta}{1-3\delta}}}{\eta} + \frac{T^2 d}{\eta\epsilon})$, *we have* $\mathcal{L}^{(s)}(\boldsymbol{\theta}(t)) \leq \epsilon$.

The full proof appears in Appendix E.3. The proof idea for this theorem is similar to that of Theorem 3.1. We first use induction to prove a key lemma similar to Lemma 3.2, though the convergence direction is different.

Lemma 3.4 (Consequence of Lemma E.7, informal)**.** *Along the gradient descent trajectory, for all* $t \geq 0$, *there exist some time-dependent scalars* $C(t), \alpha(t)$ *s.t.*

$$\boldsymbol{W}(t) = C(t)\begin{bmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_{d\times d_e} \\ \mathbf{0}_{d_e\times d} & \boldsymbol{I}_{d_e} \end{bmatrix},$$

$$\boldsymbol{V}(t) = \alpha(t)\begin{bmatrix} \boldsymbol{I}_d & \mathbf{0}_{d\times d_e} \end{bmatrix}.$$

With Lemma 3.4, we can use the (q,δ)-RIP of $\boldsymbol{E}$ to estimate the expectation in the dynamics, and moreover derive an approximate ODE similar to Equation (6) that roughly tracks the trajectory. Although we cannot calculate the exact ODE, the upper and lower bounds enable us to formulate an inductive argument, similar to the proof of Theorem 3.1. Eventually, the analysis of our controlled dynamics leads to global convergence.

Note that the model trained by GD with stochastic PE converges exactly to a solution equivalent to the constructed one-layer transformer in Sanford et al. (2023) with the same width $m \sim \Theta(d + q\log T)$. This implies GD learns STS_q using a transformer of near-optimal width, where the theoretical lower bound $\Theta(d+q)$ is only smaller by a logarithmic factor.[3]

Moreover, the learned transformer can perfectly solve STS_q tasks given any fixed positional $\boldsymbol{E}$ satisfying (q,δ)-RIP, and

[3]Remarkably, this resembles the intuition from Zhou et al. (2023): since we learn the smallest transformer in theory, it can length-generalize. This corresponds to our length generalization guarantee in Section 3.4.

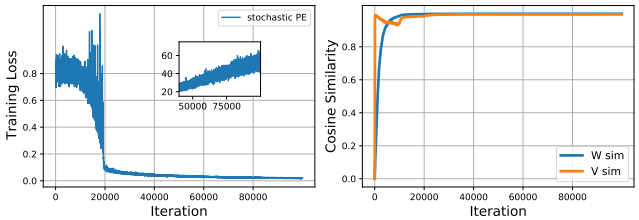

Figure 1. The above figures describe the training trajectory of the one-layer transformer model with attention layer $\boldsymbol{W}$ attending to the full matrix $[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}]$. **Left (Global convergence of the transformer)**: we plot the training loss for the one-layer transformer with stochastic PE, complementing it with the inverse loss plot. The training loss converges to the global minimum of 0, and the inverse loss increases linearly, indicating the $O(1/t)$ convergence rate. **Right (Cosine similarity with the ground-truth)**: we plot the evolution of the cosine similarity between $\boldsymbol{W}$ and $\boldsymbol{W}^\star$ and between $\boldsymbol{V}$ and $\boldsymbol{V}^\star$ throughout training, where $\boldsymbol{W}^\star$ and $\boldsymbol{V}^\star$ are the ground-truth matrices. The two cosine similarity curves gradually converge to 1, indicating the one-layer transformer eventually converges to the desired direction.

we can replace the stochastic architecture with any valid $\boldsymbol{E}$ as a fixed architecture. In practice, this means that once the transformer is successfully trained, we no longer need to sample new positional encodings for evaluation. For more discussion, see Appendix E.6.

3.3. Expressive power separation

We now complement our positive results with a width lower bound for FCNs on the population loss: any FCN without the first layer width $\Omega(Td)$ cannot approximate the STS_q with respect to the expected square loss. In comparison to the results of Sanford et al. (2023), our results lower bound the approximation error on average over the data distribution instead of considering a single worst-case data point[4].

In this section, we consider FCN of the following form:

$$f(\boldsymbol{x}) = \boldsymbol{W}_L \sigma(\boldsymbol{W}_{L-1} \cdots \sigma(\boldsymbol{W}_1 \boldsymbol{x}))$$

In fact, the results can be generalized to any network of the form $f(\boldsymbol{x}) = g(\boldsymbol{W}_1 \boldsymbol{x})$ for some first layer weight matrix $\boldsymbol{W}_1 \in \mathbb{R}^{m \times (Td+q)}$ and arbitrary function $g : \mathbb{R}^m \to \mathbb{R}^{Td}$. For the FCN, the input format should be flattened as a vector $[\boldsymbol{x}_{\text{vec}}, y] = \left(\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, ..., \boldsymbol{x}_T^\top, y_1, y_2, ..., y_q\right)^\top \in \mathbb{R}^{dT+q}$. Here, $\boldsymbol{x}_i \in \mathbb{R}^d$ for $i \in [T]$.

We now consider the expected loss for the STS_q with input tokens $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d)$ and the query subset y sampled uniformly from the set of q-subsets of $[T]$. Our result shows any FCN that can approximate STS_q requires its first hidden layer to have width at least $\Omega(Td)$.

Theorem 3.5. *Let $\mathcal{M} : \mathbb{R}^{dT+q} \to \mathbb{R}^d$ be any FCN employing any activation having first layer width at most $Td - 1$. Then:*

$$\mathbb{E}_{\boldsymbol{X}, y}\left[\|\mathcal{M}([\boldsymbol{x}_{vec}, y]) - \mathrm{STS}_q(\boldsymbol{X}, y)\|_2^2\right] \geq \frac{T-q}{Tq(T-1)}$$

The proof (see Appendix C.1) relies on the observation that when the first hidden layer has fewer than Td neurons, the linear transformation $\boldsymbol{W}_1$ has a non-trivial kernel. This implies there exists some 1-dimensional subspace where the FCN outputs a constant. However, the $\mathrm{STS}_q(\cdot)$ function depends on the total input vector $\boldsymbol{x}_{\text{vec}} \in \mathbb{R}^{Td}$ in all directions. This leads to a non-trivial approximation error.

Combining these results and Section 3.2, we rigorously establish the expressive power separation between transformers and FCNs on this STS_q task in the intrinsic complexity of width: a simple transformer with width $\Theta(d + q \log T)$ can efficiently *learn* STS_q, while any FCNs must have $\Omega(Td)$ width to even *approximate* this task.

3.4. Length generalization

Here, we present our length generalization result. One key motivation for stochastic PE is that it can improve the length generalization of the trained model, which is beyond the range of the training distribution. This inspires us to investigate theoretical guarantees on out-of-distribution data of longer sequence length on the STS_q task.

Suppose our training objective is based on the distribution $\mathcal{D}_{T_1,q}$ with sequence length T_1 and subset size q. Then, the OOD loss with sequence length $T_2 \geq T_1$ becomes:

$$\mathcal{L}_{T_2,q}^{(s)}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X}, y \sim \mathcal{D}_{T_2,q}}\left[\|\mathrm{STS}_q(\boldsymbol{X}, y) - f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y)\|_2^2\right] \tag{7}$$

We can then derive the following corollary of Theorem 3.3, proving the length generalization error goes to 0 as the training error tends to 0, as long as the embedding dimension is large enough for $\boldsymbol{E}$ to satisfy RIP.

Corollary 3.6 (Informal[5])**.** *Suppose $q = \Theta(1), d_e = \Theta(q \log T_{\max}/\delta^2)$. If we apply gradient descent with zero initialization with $T_1 < T_{\max}$ to train the model under same condition in Theorem 3.3, then when the training loss $\mathcal{L}_{T_1,q}^{(s)}(\boldsymbol{\theta}) \leq \epsilon$, it holds that for any $T_2 \in (T_1, T_{\max}]$:*

$$\mathcal{L}_{T_2,q}^{(s)}(\boldsymbol{\theta}) \leq O\left(\frac{T_2^2 \epsilon}{T_1^2}\right)$$

The proof, given in Appendix E.4, is based on the exact global minimizer GD finds after training. With stochastic

[4]The lower bound of average-case loss with respect to some data distribution implies the existence of a lower bound for worst-case loss in (Sanford et al., 2023).

[5]It is a direct corollary of Corollary E.9, which can also generalized to OOD data with unseen query subset size $q' \neq q$.

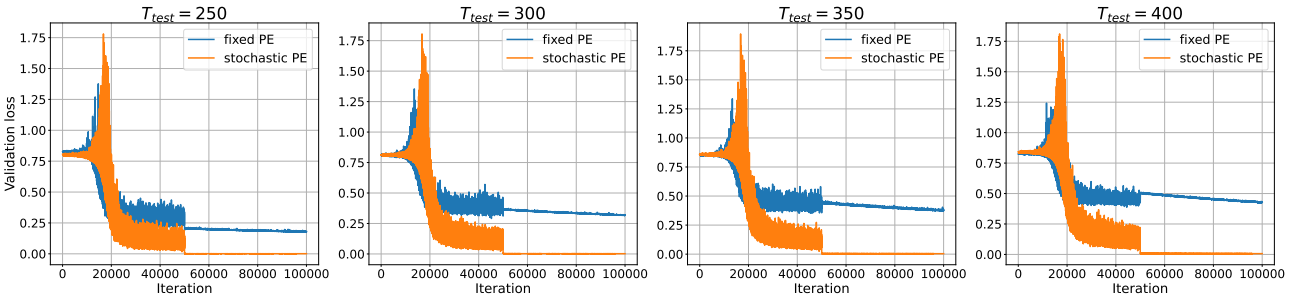

Figure 2. **Length generalization superiority of stochastic PE:** We plot the out-of-distribution error throughout training on each of the four *length generalization* tasks ($T_{\text{test}} = 250, 300, 350, 400$) when $T = 200$ and $q = 3$. Observe that the one-layer transformer with stochastic positional encoding has **a clear advantage** over the fixed positional encoding architecture in all four tasks: stochastic architecture converges after 10k steps, while the length generalization error of the fixed architecture does not go below some constant.

positional encoding, the GD solution can naturally generalize on longer input sequences, leading to this length generalization corollary.

4. Experiments

In this section, we describe our experimental setup on synthetic data, which numerically justifies our theoretical guarantees for convergence. In addition, we devise several length generalization tasks for our model, in which we are able to highlight the benefits of our stochastic architecture.

Synthetic experimental setup. In our experiment, we use the one-layer transformer architecture defined in Section 2.3. Our synthetic data follows the distribution $\mathcal{D}_{T,q}$ in our theoretical analysis: $\boldsymbol{X}$ is sampled from the standard Gaussian, and y is uniformly sampled from all possible q-subsets of $[T]$. In particular, we choose $T = 200$ for our sequence length, $q = 3$, $d = 5$, and $d_e = 170$. In addition, to simulate the population loss training, we train using online stochastic gradient descent (SGD) by resampling a fresh batch of $n = 256$ datapoints $(\boldsymbol{X}, y)$ at each iteration to use for our gradient estimate. When training with a fixed positional encoding, we sample and fix the encoding samples at the start of training; when we simulate the training of the stochastic architecture, we sample a single positional encoding $\boldsymbol{E}$ at each iteration.

Training loss convergence. Our experiments show the convergence of the in-distribution training/population loss (estimated using the fresh batch of data at each iteration) for our stochastic architecture (Figure 1). The convergence process can be verified by observing the cosine similarity of the weight matrices to their ground truth directions in Figure 1. We also experiment on a smaller transformer with $d = 20, d_e = 20$ and plot the heat map of the parameters $\boldsymbol{V}$ and $\boldsymbol{W}$. As we can see in Figure 3, the heatmaps of $\boldsymbol{W}$ and

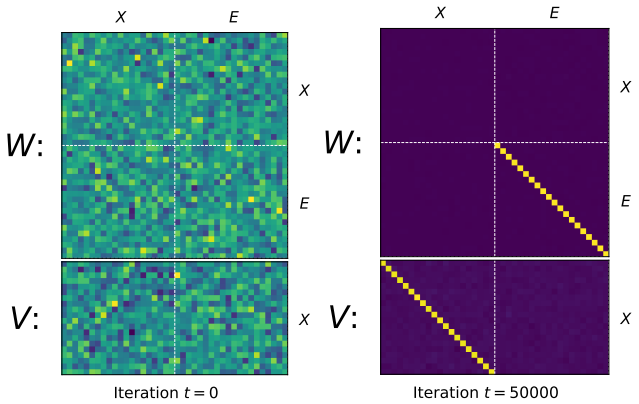

Figure 3. **Interpretable training:** For the full model Equation (2), we present a heat map of the self-attention layer $\boldsymbol{W}$ and the value matrix $\boldsymbol{V}$ at initialization and after convergence. We initialize $\boldsymbol{W}, \boldsymbol{V}$ randomly at $t = 0$. After training, observe that only the sub-block of $\boldsymbol{W}$ that attends to the positional encodings $\boldsymbol{E}$ converges to the identity direction, while all other entries converge to 0; in $\boldsymbol{V}$, only the sub-block that attends to the input tokens $\boldsymbol{X}$ converges to identity direction with all other entries converging to 0.

$\boldsymbol{V}$ coincide with the ground-truth

$$\boldsymbol{W}^* = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \alpha \boldsymbol{I}_{d_e} \end{bmatrix} \quad \boldsymbol{V}^* = \begin{bmatrix} \boldsymbol{I}_d & \boldsymbol{0} \end{bmatrix}.$$

This justifies our reparameterization, confirming that the simplified transformer still captures the essential positional information of the STS_q task.

Length generalization. We propose the following out-of-distribution length generalization tasks for our models. For each task, we fix before training a validation set of $n_{\text{test}} = 128$ out-of-distribution datapoints $(\boldsymbol{X}, y)$ from the corresponding task distribution.[6] Figure 2 confirms that the fixed architecture has drastically worse length generalization compared to the stochastic architecture: transformers

[6]For detailed settings, see Appendix F.

with fixed PE cannot even converge to zero OOD validation loss while the stochastic model extrapolates to data of unseen length. Our empirical findings reveal the benefits of using a stochastic positional encoding architecture over a fixed positional encoding architecture, thereby justifying our theoretical setup and the results in Corollary 3.6.

5. Conclusion

In this paper, we put forward a comprehensive theoretical analysis of gradient descent on the *sparse token selection task* STS_q. We characterize the joint training dynamics of a one-layer transformer with stochastic positional encoding and demonstrate the width separation between transformers and fully-connected networks on STS_q task. The stochasticity of these positional encodings also provably leads to length generalization capabilities beyond what is seen in the input data, highlighting a benefit of our architecture.

There are still many open questions. For instance, can we move beyond population loss and show a sample complexity guarantee? Can we extend the benefits of randomized positional encodings to other tasks? Can we analyze the length generalization of transformers in other practical settings?

Acknowledgement

JDL acknowledges support of the NSF CCF 2002272, NSF IIS 2107304, and NSF CAREER Award 2144994. DH acknowledges support of the NSF grant IIS-2040971.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Anil, C., Wu, Y., Andreassen, A., Lewkowycz, A., Misra, V., Ramasesh, V., Slone, A., Gur-Ari, G., Dyer, E., and Neyshabur, B. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35:38546–38556, 2022.

Bai, Y., Chen, F., Wang, H., Xiong, C., and Mei, S. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*, 2023.

Barak, B., Edelman, B., Goel, S., Kakade, S., Malach, E., and Zhang, C. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

Bhattamishra, S., Ahuja, K., and Goyal, N. On the ability and limitations of transformers to recognize formal languages. *arXiv preprint arXiv:2009.11264*, 2020a.

Bhattamishra, S., Patel, A., and Goyal, N. On the computational power of transformers and its implications in sequence modeling. *arXiv preprint arXiv:2006.09286*, 2020b.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Candes, E. and Tao, T. Decoding by linear programming, 2005.

Chen, S., Sheen, H., Wang, T., and Yang, Z. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *arXiv preprint arXiv:2402.19442*, 2024.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Cohen, J. M., Kaur, S., Li, Y., Kolter, J. Z., and Talwalkar, A. Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*, 2021.

Damian, A., Nichani, E., and Lee, J. D. Self-stabilization: The implicit bias of gradient descent at the edge of stability. *arXiv preprint arXiv:2209.15594*, 2022.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, Ł. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A. P., Caron, M., Geirhos, R., Alabdulmohsin, I., et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pp. 7480–7512. PMLR, 2023.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Dubois, Y., Dagan, G., Hupkes, D., and Bruni, E. Location attention for extrapolation to longer sequences. *arXiv preprint arXiv:1911.03872*, 2019.

Edelman, B. L., Goel, S., Kakade, S., and Zhang, C. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pp. 5793–5831. PMLR, 2022.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1, 2021.

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Giannou, A., Rajput, S., Sohn, J.-y., Lee, K., Lee, J. D., and Papailiopoulos, D. Looped transformers as programmable computers. *arXiv preprint arXiv:2301.13196*, 2023.

Hahn, M. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020.

Haviv, A., Ram, O., Press, O., Izsak, P., and Levy, O. Transformer language models without positional encodings still learn positional information. *arXiv preprint arXiv:2203.16634*, 2022.

Huang, Y., Cheng, Y., and Liang, Y. In-context convergence of transformers. *arXiv preprint arXiv:2310.05249*, 2023.

Hupkes, D., Dankers, V., Mul, M., and Bruni, E. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67: 757–795, 2020.

Jelassi, S., Sander, M., and Li, Y. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 35:37822–37836, 2022.

Jumper, J. M., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D. A., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583 – 589, 2021. URL https://api.semanticscholar.org/CorpusID:235959867.

Kazemnejad, A., Padhi, I., Ramamurthy, K. N., Das, P., and Reddy, S. The impact of positional encoding on length generalization in transformers. *arXiv preprint arXiv:2305.19466*, 2023.

Kim, J. and Suzuki, T. Transformers learn nonlinear features in context: Nonconvex mean-field dynamics on the attention landscape. *arXiv preprint arXiv:2402.01258*, 2024.

Li, H., Wang, M., Liu, S., and Chen, P.-Y. A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity. *arXiv preprint arXiv:2302.06015*, 2023.

Li, Z., Wang, Z., and Li, J. Analyzing sharpness along gd trajectory: Progressive sharpening and edge of stability. *arXiv preprint arXiv:2207.12678*, 2022.

Likhosherstov, V., Choromanski, K., and Weller, A. On the expressive power of self-attention matrices. *arXiv preprint arXiv:2106.03764*, 2021.

Liu, B., Ash, J. T., Goel, S., Krishnamurthy, A., and Zhang, C. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

Mahankali, A., Hashimoto, T. B., and Ma, T. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*, 2023.

Nanda, N., Chan, L., Liberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Newman, B., Hewitt, J., Liang, P., and Manning, C. D. The eos decision and length extrapolation. *arXiv preprint arXiv:2010.07174*, 2020.

Nichani, E., Damian, A., and Lee, J. D. How transformers learn causal structure with gradient descent. *arXiv preprint arXiv:2402.14735*, 2024.

Nye, M., Andreassen, A. J., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

OpenAI. Gpt-4 technical report, 2023.

Panigrahi, A., Malladi, S., Xia, M., and Arora, S. Trainable transformer in transformer. *arXiv preprint arXiv:2307.01189*, 2023.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library, 2019.

Pérez, J., Marinković, J., and Barceló, P. On the turing completeness of modern neural network architectures. *arXiv preprint arXiv:1901.03429*, 2019.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Ruis, L., Andreas, J., Baroni, M., Bouchacourt, D., and Lake, B. M. A benchmark for systematic generalization in grounded language understanding. *Advances in neural information processing systems*, 33:19861–19872, 2020.

Ruoss, A., Delétang, G., Genewein, T., Grau-Moya, J., Csordás, R., Bennani, M., Legg, S., and Veness, J. Randomized positional encodings boost length generalization of transformers. *arXiv preprint arXiv:2305.16843*, 2023.

Sanford, C., Hsu, D., and Telgarsky, M. Representational strengths and limitations of transformers. *arXiv preprint arXiv:2306.02896*, 2023.

Shen, R., Bubeck, S., Eldan, R., Lee, Y. T., Li, Y., and Zhang, Y. Positional description matters for transformers arithmetic. *arXiv preprint arXiv:2311.14737*, 2023.

Su, J., Lu, Y., Pan, S., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *arXiv e-prints*, pp. arXiv–2104, 2021.

Tarzanagh, D. A., Li, Y., Thrampoulidis, C., and Oymak, S. Transformers as support vector machines. *arXiv preprint arXiv:2308.16898*, 2023.

Tian, Y., Wang, Y., Chen, B., and Du, S. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. *arXiv preprint arXiv:2305.16380*, 2023a.

Tian, Y., Wang, Y., Zhang, Z., Chen, B., and Du, S. Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention. *arXiv preprint arXiv:2310.00535*, 2023b.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Tsai, Y.-H. H., Bai, S., Yamada, M., Morency, L.-P., and Salakhutdinov, R. Transformer dissection: a unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.

Wortsman, M., Liu, P. J., Xiao, L., Everett, K. E., Alemi, A. A., Adlam, B., Co-Reyes, J. D., Gur, I., Kumar, A., Novak, R., Pennington, J., Sohl-Dickstein, J., Xu, K., Lee, J., Gilmer, J., and Kornblith, S. Small-scale proxies for large-scale transformer training instabilities. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=d8w0pmvXbZ.

Yao, S., Peng, B., Papadimitriou, C., and Narasimhan, K. Self-attention networks can process bounded hierarchical languages. *arXiv preprint arXiv:2105.11115*, 2021.

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S. J., and Kumar, S. Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*, 2019.

Zhang, R., Frei, S., and Bartlett, P. L. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023.

Zhang, Y., Backurs, A., Bubeck, S., Eldan, R., Gunasekar, S., and Wagner, T. Unveiling transformers with lego: a synthetic reasoning task, 2022. *URL https://arxiv.org/abs/2206.04301*, 2022.

Zhao, H., Panigrahi, A., Ge, R., and Arora, S. Do transformers parse while predicting the masked word? *arXiv preprint arXiv:2303.08117*, 2023.

Zhou, H., Bradley, A., Littwin, E., Razin, N., Saremi, O., Susskind, J., Bengio, S., and Nakkiran, P. What algorithms can transformers learn? a study in length generalization. *arXiv preprint arXiv:2310.16028*, 2023.

Zhou, Y., Alon, U., Chen, X., Wang, X., Agarwal, R., and Zhou, D. Transformers can achieve length generalization but not robustly. *arXiv preprint arXiv:2402.09371*, 2024.

A. Backgrounds and Preliminaries

A.1. Restricted isometry and orthogonality property

We replicate the definition of *restricted isometry and orthogonality* from Sanford et al. (2023) in this section. For $\boldsymbol{v} \in \mathbb{R}^T$, denote $\text{supp}(\boldsymbol{v}) = \{i \in [T] : \boldsymbol{v}_i \neq 0\}$.

Definition A.1. We say a matrix $\boldsymbol{E} \in \mathbb{R}^{d_e \times T}$ satisfies the (q, δ)-restricted isometry and orthogonality property if

$$\|\boldsymbol{E}\boldsymbol{v}\|_2^2 \in [(1-\delta)\|\boldsymbol{v}\|_2^2, (1+\delta)\|\boldsymbol{v}\|_2^2] \quad \text{and} \quad |\langle \boldsymbol{E}\boldsymbol{v}, \boldsymbol{E}\boldsymbol{v}'\rangle| \leq \delta\|\boldsymbol{v}\|_2\|\boldsymbol{v}'\|$$

for all vectors $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^T$ with $|\text{supp}(\boldsymbol{v})| \leq q$, $|\text{supp}(\boldsymbol{v}')| \leq 2q$, and $|\text{supp}(\boldsymbol{v}) \cap \text{supp}(\boldsymbol{v}')| = 0$.

Here we restate the two lemmas in Sanford et al. (2023) for the construction of the transformer approximating STS_q. The first lemma is that the existence of a Rademacher random matrix can satisfy the restricted isometry and orthogonality properties. For simplicity, we just call it RIP.

Lemma A.2 (Lemma 12 from Sanford et al. (2023))**.** *There is an absolute constant $C > 0$ such that the following holds. Fix $\delta \in (0, 1/2)$ and $q \in \mathbb{N}$. Let $\boldsymbol{E}$ denote an $d_e \times T$ matrix of independent Rademacher random variables scaled by $\frac{1}{\sqrt{d_e}}$. If $d_e \geq C(q \log T)/\delta^2$, then with positive probability, $\boldsymbol{E}$ satisfies the (q, δ)-restricted isometry and orthogonality property.*

Lemma A.3 (Lemma 13 from Sanford et al. (2023) and consequence of Lemma 2.1 from Candes & Tao (2005))**.** *Fix $\delta \in (0, 1/2)$ and $q \in \mathbb{N}$. Let $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_T] \in \mathbb{R}^{d_e \times T}$ satisfy the (q, δ)-restricted isometry orthogonality property. For every vector $\boldsymbol{v} \in \{0, 1\}^T$ with $\text{supp}(\boldsymbol{v}) \leq q$, suppose the indices $i_1, i_2, \ldots, i_{\text{supp}(\boldsymbol{v})} \in [T]$ in $\boldsymbol{v}$ have entry 1. Then, the vector $\boldsymbol{e}_{\boldsymbol{v}} = \boldsymbol{E}_{\boldsymbol{v}}(\boldsymbol{E}_{\boldsymbol{v}}^\top \boldsymbol{E}_{\boldsymbol{v}})^{-1}\mathbf{1}_{\text{supp}(\boldsymbol{v})} \in \mathbb{R}^{d_e}$ satisfies the following:*

$$\begin{aligned} \|\boldsymbol{e}_{\boldsymbol{v}}\|_2 &\leq \sqrt{q}/(1-2\delta) \\ \langle \boldsymbol{e}_i, \boldsymbol{e}_{\boldsymbol{v}}\rangle &= 1 \quad \textit{if } v_i = 1 \\ |\langle \boldsymbol{e}_i, \boldsymbol{e}_{\boldsymbol{v}}\rangle| &\leq \delta/(1-2\delta) \quad \textit{if } v_i = 0 \end{aligned}$$

where we define $\boldsymbol{E}_{\boldsymbol{v}} = [\boldsymbol{e}_{i_1}, \ldots, \boldsymbol{e}_{i_{\text{supp}(\boldsymbol{v})}}] \in \mathbb{R}^{d_e \times \text{supp}(v)}$.

Remark. Compared with Sanford et al. (2023), we add the expression of $\boldsymbol{e}_{\boldsymbol{v}} = \boldsymbol{E}_{\boldsymbol{v}}(\boldsymbol{E}_{\boldsymbol{v}}^\top \boldsymbol{E}_{\boldsymbol{v}})^{-1}\mathbf{1}_{\text{supp}(\boldsymbol{v})}$, which we use as the encoding for the query subset y in our one-layer transformer. It is used in Lemma 2.1 of Candes & Tao (2005) as the dual certificate for the specific dual problem of a primal ℓ_1-minimization problem. We borrow the exact form of the dual certificate here as the encoding, replacing the blackbox MLP layer $\phi(\boldsymbol{X})$ used in Sanford et al. (2023) (which can be poly(T, d)-wide, far larger than the width of the transformer). We posit that this positional encoding for y can also be expressed by a multi-layer transformer with width $\Theta(d + q \log T)$, depth $\Theta(\log\log\log T + \log\log \frac{q}{\epsilon})$. It can be constructed using the transformer construction on matrix inverse in Giannou et al. (2023).

A.2. The equivalence between simplified STS_q and qSA in Sanford et al. (2023)

We restate our STS_q task and compare it with the original qSA task used in Sanford et al. (2023).

Definition A.4. For sparsity q, token dimension d, and input dimension $dT + q$, consider the input $(\boldsymbol{X}, y) = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T; y_1, y_2, ..., y_q) \in \mathbb{R}^{dT+q}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y \in \binom{[T]}{q}$ is a q-element subset of $[T]$. Define the q-sparse token selection $\text{STS}_q(\cdot)$ as

$$\text{STS}_q(\boldsymbol{X}, y) = \frac{1}{q}\sum_{j=1}^{q} \boldsymbol{x}_{y_j}$$

This arithmetic task is based on the qSA task, while reduce the number of query subsets y_i (which is T in Sanford et al. (2023)) to 1. As a regression task, we believe it is a natural simplification only to consider one query in one data point.

Moreover, when considering the population ℓ_2 loss Equation (3) we are using (instead of the ℓ_∞ loss used in Sanford et al. (2023)), the two tasks are equivalent. For STS_q, the population loss is:

$$\mathcal{L}_{T,q}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X}, y \sim \mathcal{D}_{T,q}}\left[\|\text{STS}_q(\boldsymbol{X}, y) - f_{\boldsymbol{\theta}}(\boldsymbol{X}, y)\|_2^2\right] = \frac{1}{2}\mathbb{E}_{\boldsymbol{X}, y \sim \mathcal{D}_{T,q}}\left[\left\|\frac{1}{q}\sum_{j=1}^{q}\boldsymbol{x}_{y_j} - f_{\boldsymbol{\theta}}(\boldsymbol{X}, y)\right\|_2^2\right]$$

For the original q**SA**, suppose all y_i follow the uniform distribution, then we have the population loss:

$$\begin{aligned}\mathcal{L}_{T,q}(\boldsymbol{\theta}) &= \frac{1}{2}\,\mathbb{E}_{\boldsymbol{X},y_i\sim\mathcal{D}_{T,q}}\left[\sum_{i=1}^{T}\|q\mathrm{SA}(\boldsymbol{X},y_i)-f_{\boldsymbol{\theta}}(\boldsymbol{X},y_i)\|_2^2\right]\\ &= \frac{T}{2}\,\mathbb{E}_{\boldsymbol{X},y\sim\mathcal{D}_{T,q}}\left[\left\|\frac{1}{q}\sum_{j=1}^{q}\boldsymbol{x}_{y_{i,j}}-f_{\boldsymbol{\theta}}(\boldsymbol{X},y_i)\right\|_2^2\right]\end{aligned}$$

They only have a T-factor difference. It also indicates that the training trajectory is also different in scale with an appropriate learning rate. Therefore, we believe it's more reasonable to consider the optimization problem on our simplified STS_q problem.

B. Limitation and discussion

Beyond the scope of this work, there still exist some limitations and future open problems.

First of all, our analysis is based on population loss rather than empirical risk minimization (ERM) on a finite dataset using (stochastic) gradient descent. The population loss is equivalent to the empirical loss induced by the limit of infinite training samples, which largely simplifies the dynamics. In our case, the stochasticity in the positional encodings also takes advantage of this population loss. It enables us to focus on analyzing the stochastic architecture $\mathbb{E}_E[f]$, where f is the transformer with E as its positional encoding. This population objective is used in almost all recent works analyzing the full-training dynamics on linear/softmax transformer architectures (Huang et al., 2023; Zhang et al., 2023; Nichani et al., 2024; Chen et al., 2024; Tian et al., 2023a; Kim & Suzuki, 2024). It would be an interesting open problem to analyze the SGD dynamics and sample complexity on any of the existing tasks in the literature.

Another limitation of our submitted version of the paper relates to the simplified architecture of our setting. Specifically, we consolidate the query and key matrices $\boldsymbol{Q}, \boldsymbol{K}$ into $\boldsymbol{W}$. Though it may lead to different landscape properties on the training trajectory, it does not inherently change the expressive power of the transformer. In particular, most of the recent works have also adopted (1) to simplify the dynamics (Huang et al., 2023; Zhang et al., 2023; Nichani et al., 2024; Tian et al., 2023a; Kim & Suzuki, 2024).

Finally, we would like to remark that we use stochastic positional encoding in our one-layer transformer instead of a fixed set of near-orthogonal positional encoding; the additional stochasticity enables us to analyze the GD dynamics theoretically. Moreover, it also helps a lot in practice with regard to out-of-distribution length generalization. Empirically, both choices work for in-distribution convergence, but only stochastic positional encoding can also achieve out-of-distribution length generalization. The length generalization superiority of randomized PE is also justified in several recent works with extensive experiments (Shen et al., 2023; Ruoss et al., 2023; Zhou et al., 2024). Nevertheless, analyzing the dynamics with a set of fixed positional encodings on the in-distribution loss can be an interesting open problem.

C. Approximation results on STS_q

As Sanford et al. (2023) proved the worst-case width lower bound for FCNs to approximate the original qSA problem, we can also prove similar results for the new formulation of STS_q. In this paper, we go beyond the worst-case analysis and prove the width lower bound in expectation under certain data distribution. Also for completeness, we present the approximation results with a one-layer transformer, showing the capability of the transformer to represent STS_q.

C.1. The average-case lower bounds for FCNs on STS_q

We first present our negative results FCNs for STS_q in the expected loss. For an FCN, we vectorize the input to $[\boldsymbol{x}_{\text{vec}}, y] = \left(\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, ..., \boldsymbol{x}_T^\top, y_1, y_2, ..., y_q\right)^\top \in \mathbb{R}^{dT+q}$. Here $\boldsymbol{x}_i \in \mathbb{R}^d$ for $i = 1, 2, ..., T$. With this theorem, we rigorously establish the *exponential* expressivity separation between FCNs and one-layer transformers in the complexity metric of width/embedding dimension.

Theorem C.1. *Let $\mathcal{M} : \mathbb{R}^{dT+q} \to \mathbb{R}^d$ be any FCN employing any activation having first layer width at most $Td - 1$, then*

$$\mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d), y \sim \text{Unif}\left(\binom{[T]}{q}\right)} \left[\|\mathcal{M}(\boldsymbol{x}_{vec}, y) - \text{STS}_q(\boldsymbol{X}, y)\|_2^2 \right] \geq \frac{T-q}{Tq(T-1)}$$

Proof. Consider the first layer weight matrix $\boldsymbol{W} \in \mathbb{R}^{k \times (dT+q)}$. Since $k \leq Td - 1$, $\text{rank}(\boldsymbol{W}) \leq Td - 1$. Therefore, we have the submatrix of the first Td rows $W_s = \boldsymbol{W}[:, Td] \in \mathbb{R}^{k \times Td}$ has its nullspace $\ker(\boldsymbol{W}_s)$. Denote the vector $\boldsymbol{v} \in \ker(\boldsymbol{W}_s)$ s.t. $\|\boldsymbol{v}\| = 1$ and w.l.o.g. $\|\boldsymbol{v}\|_\infty = v_1$, which is the first entry of the vector. We then find the set of orthonormal basis $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_{Td}\}$ in $\mathbb{R}^{Td}$ with $\boldsymbol{v}_1 := \boldsymbol{v}$ as the first basis vector.

Now consider the decomposition of $\boldsymbol{x}_{\text{vec}}$ in the basis $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_{Td}\}$. Since it's sampled from normal distribution, $\boldsymbol{x}_{\text{vec}}$ can be rewritten to the reparametrized vector:

$$\boldsymbol{x}_{\text{vec}} = \epsilon_1 \boldsymbol{v}_1 + \epsilon_2 \boldsymbol{v}_2 + \cdots + \epsilon_{Td} \boldsymbol{v}_{Td},\ \epsilon_i \sim \mathcal{N}(0, 1),\ i = 1, 2, ..., Td$$

Denote $\boldsymbol{\epsilon} := (\epsilon_1, ..., \epsilon_{Td})$. Then we can calculate the expected MSE loss given some sparse set y:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d)} \left[\|\mathcal{M}(\boldsymbol{x}_{\text{vec}}, y) - \text{STS}_q(\boldsymbol{X}, y)\|_2^2 \right] \\
&= \int_{\boldsymbol{x}_{\text{vec}}} \|\mathcal{M}(\boldsymbol{x}_{\text{vec}}, y) - \text{STS}_q(\boldsymbol{x}_{\text{vec}}, y)\|^2 \mathrm{d}p(\boldsymbol{x}) \\
&= \int_{\boldsymbol{\epsilon}} \left\| \mathcal{M}\left(\sum_{i=1}^{Td} \epsilon_i \boldsymbol{v}_i, y_1, ..., y_q \right) - \text{STS}_q(\boldsymbol{x}_{\text{vec}}, y) \right\|^2 \mathrm{d}p(\boldsymbol{\epsilon})
\end{aligned}$$

Note that $\text{STS}_q(\boldsymbol{x}_{\text{vec}}, y) = \text{STS}_q\left(\sum_{i=1}^{Td} \epsilon_i \boldsymbol{v}_i, y_1, ..., y_q\right) = \sum_{i=1}^{Td} \epsilon_i \, \text{STS}_q\left(\boldsymbol{v}_i, y_1, ..., y_q\right)$. And since $\boldsymbol{v}_1 = \boldsymbol{v}$ is in the kernel of the first layer,

$$\mathcal{M}\left(\sum_{i=1}^{Td} \epsilon_i \boldsymbol{v}_i, y_1, ..., y_q \right) = \mathcal{M}\left(\sum_{i=2}^{Td} \epsilon_i \boldsymbol{v}_i, y_1, ..., y_q \right)$$

which is constant with respect to ϵ_1. Therefore we have

$$\begin{aligned}
&\int_{\mathbb{R}} \left\| \mathcal{M}\left(\sum_{i=1}^{Td} \epsilon_i \boldsymbol{v}_i, y_1, ..., y_q \right) - \text{STS}_q(\boldsymbol{x}_{\text{vec}}, y) \right\|^2 p(\epsilon_1) \mathrm{d}\epsilon_1 \\
&= \int_{\mathbb{R}} \left\| \mathcal{M}\left(\sum_{i=2}^{Td} \epsilon_i \boldsymbol{v}_i, y_1, ..., y_q \right) - \sum_{i=1}^{Td} \epsilon_i \, \text{STS}_q\left(\boldsymbol{v}_i, y_1, ..., y_q\right)) \right\|^2 p(\epsilon_1) \mathrm{d}\epsilon_1 \\
&= \int_{\mathbb{R}} \left\| \mathcal{M}\left(\sum_{i=2}^{Td} \epsilon_i \boldsymbol{v}_i, y_1, ..., y_q \right) - \epsilon_1 \, \text{STS}_q\left(\boldsymbol{v}_1, y_1, ..., y_q\right)) \right. \\
&\quad \left. - \sum_{i=2}^{Td} \epsilon_i \, \text{STS}_q\left(\boldsymbol{v}_i, y_1, ..., y_q\right)) \right\|^2 p(\epsilon_1) \mathrm{d}\epsilon_1 \\
&\geq \int_{\mathbb{R}} \|\text{STS}_q\left(\boldsymbol{v}_1, y_1, ..., y_q\right))\|^2 \epsilon_1^2 p(\epsilon_1) \mathrm{d}\epsilon_1 = \|\text{STS}_q\left(\boldsymbol{v}_1, y_1, ..., y_q\right))\|^2
\end{aligned}$$

The last inequality is because after expanding the squared norm we have (1) ϵ_1^0-terms are always non-negative and (2) the integral for ϵ_1-terms is 0.

Now we have the lower bound for the expected loss:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d), y \sim \text{Unif}\left(\binom{[T]}{q}\right)} \left[\|\mathcal{M}(\boldsymbol{x}_{\text{vec}}, y) - \text{STS}_q(\boldsymbol{X}, y)\|_2^2 \right] \\
&= \mathbb{E}_{y \sim \text{Unif}\left(\binom{[T]}{q}\right)} \int_{\boldsymbol{\epsilon}} \left\| \mathcal{M}\left(\sum_{i=1}^{Td} \epsilon_i \boldsymbol{v}_i, y_1, ..., y_q \right) - \text{STS}_q(\boldsymbol{x}_{\text{vec}}, y) \right\|^2 \mathrm{d}p(\boldsymbol{\epsilon})
\end{aligned}$$

$$
\begin{aligned}
&\geq \mathbb{E}_{y\sim \mathrm{Unif}\left(\binom{[T]}{q}\right)} \int_{\boldsymbol{\epsilon}_{2:Td}} \left\|\mathrm{STS}_q\left(\boldsymbol{v}_1, y_1, ..., y_q\right)\right\|^2 \mathrm{d}p(\boldsymbol{\epsilon}_{2:Td}) \\
&= \mathbb{E}_{y\sim \mathrm{Unif}\left(\binom{[T]}{q}\right)} \left\|\mathrm{STS}_q\left(\boldsymbol{v}, y_1, ..., y_q\right)\right\|^2 \\
&= \mathbb{E}_{y\sim \mathrm{Unif}\left(\binom{[T]}{q}\right)} \left\|\sum_{i\in y} \boldsymbol{v}^{(i)}\right\|^2 / q^2 \\
&= \mathbb{E}_{y\sim \mathrm{Unif}\left(\binom{[T]}{q}\right)} \left[\sum_{i\in y} \|\boldsymbol{v}^{(i)}\|^2/q^2\right] + \mathbb{E}_{y\sim \mathrm{Unif}\left(\binom{[T]}{q}\right)} \left[\frac{1}{q^2} \sum_{i,j\in y, i\neq j} \boldsymbol{v}^{(i)\top} \boldsymbol{v}^{(j)}\right] \\
&= \frac{q}{T} \sum_{i=1}^{T} \frac{1}{q^2} \|\boldsymbol{v}^{(i)}\|^2 + \frac{q(q-1)}{T(T-1)} \sum_{i\neq j} \frac{1}{q^2} \boldsymbol{v}^{(i)\top} \boldsymbol{v}^{(j)} \\
&\geq \frac{T-q}{Tq(T-1)} \|\boldsymbol{v}\|^2 + \frac{(q-1)}{Tq(T-1)} \left\|\sum_{i=1}^{T} \boldsymbol{v}^{(i)}\right\|^2 \geq \frac{T-q}{Tq(T-1)}.
\end{aligned}
$$

□

C.2. The worst-case lower bounds for FCNs on STS_q

For completeness, we present the negative results FCNs to approximate the newly formulated STS_q problem. It is using exactly the same technique in Theorem 10 of Sanford et al. (2023).

Theorem C.2. *Let $\mathcal{M} : \mathbb{R}^{dT+q} \to \mathbb{R}^d$ be any FCN employing any activation having first layer width at most $(T-q+1)d-1$, then there exists some input $(\boldsymbol{x}_{vec}, y) \in \mathbb{R}^{dT+q}$ s.t.*

$$
\|\mathcal{M}(\boldsymbol{x}_{vec}, y) - \mathrm{STS}_q(\boldsymbol{X}, y)\|_2^2 \geq \frac{1}{2q}
$$

Proof. Let $\mathcal{M}(\boldsymbol{x}) = f(\boldsymbol{W}\boldsymbol{x})$, where $\boldsymbol{W}$ is the first layer matrix satisfying $\boldsymbol{W} \in \mathbb{R}^{m\times(dT+q)}$, $m \leq (T-q+1)d-1$, and $f : \mathbb{R}^m \to \mathbb{R}^d$ is an arbitrary function representing the subsequent layers of the FCN. $\boldsymbol{W}$ can be partitioned as

$$
\left[\boldsymbol{V}_1; ...; \boldsymbol{V}_T; \tilde{\boldsymbol{W}}\right],
$$

where $\boldsymbol{V}_1, ..., \boldsymbol{V}_T \in \mathbb{R}^{m\times d}$, $\tilde{\boldsymbol{W}} \in \mathbb{R}^{m\times q}$. Due to our restriction on the width of the first layer, we have

$$
\mathrm{rank}\left([\boldsymbol{V}_q; \cdots; \boldsymbol{V}_T]\right) \leq m \leq (T-q+1)d-1 < (T-q+1)d.
$$

This implies that $[\boldsymbol{V}_q; \cdots; \boldsymbol{V}_T]$ has a nontrivial null space containing a nonzero vector $\boldsymbol{u} = (\boldsymbol{u}_q, ..., \boldsymbol{u}_T) \in \mathbb{R}^{(T-q+1)d}$. Let

$$
\boldsymbol{\xi} = \frac{1}{2\max_{j\in\{q,...,T\}} \|\boldsymbol{u}_j\|_2} \boldsymbol{u},
$$

and (for simplicity, we concatenate $\boldsymbol{x}_{\text{vec}}$ and y and call it $\boldsymbol{x}_{\text{vec}}$ in the following proof.)

$$
\begin{aligned}
\boldsymbol{x}_{\text{vec}} &= (\underbrace{\boldsymbol{0}, ..., \boldsymbol{0}}_{q-1}, \boldsymbol{\xi}_q, ..., \boldsymbol{\xi}_T, y_1, y_2, ..., y_q) \\
\boldsymbol{x}'_{\text{vec}} &= (\underbrace{\boldsymbol{0}, ..., \boldsymbol{0}}_{q-1}, -\boldsymbol{\xi}_q, ..., -\boldsymbol{\xi}_T, y_1, y_2, ..., y_q).
\end{aligned}
$$

Then

$$
\boldsymbol{W}\boldsymbol{x}_{\text{vec}} = \boldsymbol{V}_1\boldsymbol{0} + ... + \boldsymbol{V}_{q-1}\boldsymbol{0} + [\boldsymbol{V}_q; \cdots; \boldsymbol{V}_T]\,\boldsymbol{\xi} + \tilde{\boldsymbol{W}}\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}'_{\text{vec}},
$$

and for some $j^* \in \{q, ..., T\}$, $\|\boldsymbol{\xi}_{j^*}\| = 1/2$. Consider

$$
\boldsymbol{y} = (1, ..., q-1, j)
$$

$$\boldsymbol{x}_{\text{vec},j} = (\mathbf{0}, ..., \mathbf{0}, \boldsymbol{\xi}_q, ..., \boldsymbol{\xi}_T, \boldsymbol{y})$$
$$\boldsymbol{x}'_{\text{vec},j} = (\mathbf{0}, ..., \mathbf{0}, -\boldsymbol{\xi}_q, ..., -\boldsymbol{\xi}_T, \boldsymbol{y})$$

for each $j \in \{q, ..., T\}$. We observe that

$$\mathrm{STS}_q(\boldsymbol{x}_{\text{vec},j}) = \frac{1}{q}\boldsymbol{\xi}_j \quad \text{and} \quad \mathrm{STS}_q(\boldsymbol{x}'_{vec,j}) = -\frac{1}{q}\boldsymbol{\xi}_j.$$

It follows that

$$\left\|\mathrm{STS}_q(\boldsymbol{x}_{\text{vec},j^*}) - \mathrm{STS}_q(\boldsymbol{x}'_{\text{vec},j^*})\right\| = \frac{1}{q}.$$

Because we have shown that $\mathcal{M}(\boldsymbol{x}_{\text{vec},j}) = f(\boldsymbol{W}\boldsymbol{x}_{\text{vec},j}) = f(\boldsymbol{W}\boldsymbol{x}'_{\text{vec},j}) = \mathcal{M}(\boldsymbol{x}'_{\text{vec},j})$,

$$\max\left(\left\|\mathcal{M}(\boldsymbol{x}_{\text{vec},j}) - \mathrm{STS}_q(\boldsymbol{x}_{\text{vec},j})\right\|, \left\|\mathcal{M}(\boldsymbol{x}'_{\text{vec},j}) - \mathrm{STS}_q(\boldsymbol{x}'_{\text{vec},j})\right\|\right) \geq \frac{1}{2q}.$$

So in the worst case, $\mathcal{M}$ can approximate STS_q with a loss no better than $1/2q$. □

C.3. Self-attention can approximate STS_q

We exhibit the expressivity result for the q-sparse token selection task for completeness. Due to the equivalence between our STS_q and the original qSA, the proof uses the same method in the proof in Sanford et al. (2023). In this method, it is required to reform the input matrix Equation (8). In expressivity, it is equivalent to our model.

Theorem C.3 (Consequence of Theorem 2 in Sanford et al. (2023))**.** *For any T, ϵ, any $m := d + 2d_e \geq \Omega(d + q\log T)$, there exists some near-orthogonal positional encoding $\boldsymbol{E} \in \mathbb{R}^{d_e \times T}$ and the corresponding q-sparse subset encoding $\boldsymbol{e}_y \in \mathbb{R}^{d_e}$ for $y \in \binom{[T]}{q}$, s.t. there exists some 1-layer self-attention unit with width m that ϵ-approximates STS_q.*

Proof. First, we choose the near-orthogonal positional encodings to enable the efficient representation of set y and position i. We consider some $d_e = C\frac{q\log T}{\delta^2}$ for some constant C (We can pick $\delta = 1/4$), and generate Rademacher random vectors. According to Lemma 12 and 13 in Sanford et al. (2023), there exist T positional encoding vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_T \in \mathbb{R}^{d_e}$ satisfying

$$\langle \boldsymbol{e}_i, \boldsymbol{e}_i \rangle = 1$$
$$|\langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle| \leq \delta, \qquad i \neq j$$

Denote $\boldsymbol{e}_y = h(y) = \boldsymbol{E}_y(\boldsymbol{E}_y^\top \boldsymbol{E}_y)^{-1}\mathbf{1}_q$, then we will have (this is the explicit form of Sanford et al. (2023) Lemma 13)

$$\langle \boldsymbol{e}_i, \boldsymbol{e}_y \rangle = 1 \text{ for all } i \in y$$
$$|\langle \boldsymbol{e}_i, \boldsymbol{e}_y \rangle| \leq \frac{\delta}{1 - 2\delta} \text{ for all } i \notin y$$

Then we describe the transformer weights. Now we have the input in the following form:

$$\boldsymbol{X}_{\text{input}} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{T-1} & \boldsymbol{x}_T \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{e}_y \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{T-1} & \boldsymbol{e}_T \end{bmatrix} \tag{8}$$

With positional encodings, we can construct sparse linear operators $\boldsymbol{Q} = [\mathbf{0}_{d\times d}, \boldsymbol{I}_{d_e}, \mathbf{0}_{d_e \times d_e}]$, $\boldsymbol{K} = [\mathbf{0}_{d\times d}, \mathbf{0}_{d_e \times d_e}, \boldsymbol{I}_{d_e}]$, $\boldsymbol{V}$ to represent the v_{y_i}, e_i, and $\boldsymbol{x}_i$ with $\Theta(d + qd_e)$ width:

$$\boldsymbol{QX} = (0, 0, ...0, , \alpha\boldsymbol{e}_y) \in \mathbb{R}^{d_e \times T}$$
$$\boldsymbol{KX} = (\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_T) \in \mathbb{R}^{d_e \times T}$$
$$\boldsymbol{VX} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T) \in \mathbb{R}^{d \times T}$$

We pick $\alpha := \lceil 2\log(2T)/\epsilon \rceil$ to help the softmax layer to do the average.

Pass the input through the transformer f, and we only take the output for the last token with $\boldsymbol{e}_y$, we have

$$\begin{aligned} f(\boldsymbol{X})_T &= \boldsymbol{V}\boldsymbol{X}\text{softmax}(\boldsymbol{X}^\top \boldsymbol{K}^\top \boldsymbol{Q}\boldsymbol{X})[:, T] \\ &= \sum_{i=1}^{T} \text{softmax}(\boldsymbol{X}^\top \boldsymbol{K}^\top \boldsymbol{Q}\boldsymbol{X})[i, T]\boldsymbol{x}_i \end{aligned}$$

Analyze the output of the softmax. If $i \in y$, we first consider the upper bound:

$$\text{softmax}(\boldsymbol{X}^\top \boldsymbol{K}^\top \boldsymbol{Q}\boldsymbol{X})[i, T] \leq \frac{e^\alpha}{qe^\alpha} = \frac{1}{q}$$

And then the lower bound

$$\text{softmax}(\boldsymbol{X}^\top \boldsymbol{K}^\top \boldsymbol{Q}\boldsymbol{X})[i, T] \geq \frac{e^\alpha}{qe^\alpha + Ne^{\frac{\alpha}{2}}} \geq \frac{1}{q} - \frac{T}{q^2}e^{-\alpha} \geq \frac{1}{q} - \frac{\epsilon}{2q}.$$

That means

$$\text{softmax}(\boldsymbol{X}^\top \boldsymbol{K}^\top \boldsymbol{Q}\boldsymbol{X})[i, T] \in \left[\frac{1}{q}, \frac{1}{q} + \frac{\epsilon}{2q}\right] \text{ for all } i \in y_i.$$

If $i \notin y_i$, then the upper bound of the softmax is

$$\text{softmax}(\boldsymbol{X}^\top \boldsymbol{K}^\top \boldsymbol{Q}\boldsymbol{X})[i, T] \leq \frac{e^{\alpha/2}}{qe^\alpha} \leq \frac{\epsilon}{2T}.$$

Finally, we conclude the above bounds:

$$\begin{aligned} &\|f(\boldsymbol{X}, y)_T - \text{STS}_q(\boldsymbol{X}, y)_T\|_2 \\ \leq &\left\| \sum_{i \in y} \left(\frac{1}{q} - \text{softmax}(\boldsymbol{X}^\top \boldsymbol{K}^\top \boldsymbol{Q}\boldsymbol{X})[i, T] \right) \boldsymbol{x}_i - \sum_{i \notin y} \left(\text{softmax}(\boldsymbol{X}^\top \boldsymbol{K}^\top \boldsymbol{Q}\boldsymbol{X})[i, T] \right) \boldsymbol{x}_i \right\|_2 \\ \leq &\, q \cdot \frac{\epsilon}{2q} + (T - q) \cdot \frac{\epsilon}{2T} \leq \epsilon \end{aligned}$$

So we construct a 1-layer transformer that can ϵ-approximate STS_q. □

D. Proof details in Section 3.1

In this subsection, we study the gradient descent convergence on the q-sparse token selection problem with one-hot positional encoding. We consider updating $\boldsymbol{V}$ and $\boldsymbol{W}$ simultaneously with the same learning rate η. For gradient descent, the update dynamics for $\boldsymbol{W}(t)$ and $\boldsymbol{V}(t)$ should be

$$\begin{aligned} \boldsymbol{W}(t+1) &= \boldsymbol{W}(t) - \eta \nabla_{\boldsymbol{W}} \mathcal{L}(t) \\ \boldsymbol{V}(t+1) &= \boldsymbol{V}(t) - \eta \nabla_{\boldsymbol{V}} \mathcal{L}(t) \end{aligned}$$

D.1. GD dynamics and preliminaries

Based on the reparameterization and the objective in Equation (3), the following lemma shows the gradients of $\boldsymbol{W}$ and $\boldsymbol{V}$. Recall that the input matrix is in the following form

$$[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}] := \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{T-1} & \boldsymbol{x}_T & \boldsymbol{x}_{\text{query}} \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{T-1} & \boldsymbol{e}_T & \boldsymbol{e}_y \end{bmatrix} \in \mathbb{R}^{(d+T)\times(T+1)}. \tag{9}$$

where we separate the input tokens $\boldsymbol{Z}$ and the query token $\boldsymbol{z}_{\text{query}}$:

$$\boldsymbol{Z} := \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{E} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{T-1} & \boldsymbol{x}_T \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{T-1} & \boldsymbol{e}_T \end{bmatrix} \in \mathbb{R}^{(d+T)\times T}, \boldsymbol{z}_{\text{query}} = \begin{bmatrix} \boldsymbol{x}_{\text{query}} \\ \boldsymbol{e}_y \end{bmatrix} \tag{10}$$

Lemma D.1. *Denote $\boldsymbol{S}_y := \mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{query}) \in \mathbb{R}^T$ at time t for certain q-sparse set y. Also, we define the q-hot vector $\boldsymbol{Y} = (\mathbb{1}\{1 \in y\}, \mathbb{1}\{2 \in y\}, ..., \mathbb{1}\{T \in y\})$ for the subset $y \in \binom{[T]}{q}$. The gradient dynamics of $\boldsymbol{W}$ with input token matrix $\boldsymbol{X}$ is:*

$$\nabla_{\boldsymbol{W}} \mathcal{L} = -\mathbb{E}_{\boldsymbol{X},y}\left(\boldsymbol{Z}(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top)\boldsymbol{Z}^\top\boldsymbol{V}^\top(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y)\boldsymbol{z}_{query}^\top\right)$$
$$\nabla_{\boldsymbol{V}} \mathcal{L} = -\mathbb{E}_{\boldsymbol{X},y}\left(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)(\boldsymbol{Z}\boldsymbol{S}_y)^\top\right)$$

Proof. The loss function is as follows according to Equation (3)

$$\mathcal{L}(\boldsymbol{\theta}(t)) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y}\left[\|\mathrm{STS}_q(\boldsymbol{X}, y) - f_{\boldsymbol{\theta}}(\boldsymbol{X}, y)\|_2^2\right].$$

Take matrix differentials and we have

$$\begin{aligned}\mathrm{d}\mathcal{L} =& \mathbb{E}_{\boldsymbol{X},y}\left[(f(\boldsymbol{X}) - \mathrm{STS}_q(\boldsymbol{X}))^\top\boldsymbol{V}\boldsymbol{X}\mathrm{d}(\mathrm{softmax}(\boldsymbol{Z}^\top\boldsymbol{W}\boldsymbol{z}_{\text{query}})\right]\\ &+ \mathbb{E}_{\boldsymbol{X},y}\left[(f(\boldsymbol{X}) - \mathrm{STS}_q(\boldsymbol{X}))^\top\mathrm{d}\boldsymbol{V}(\boldsymbol{Z}\boldsymbol{S}_y)\right]\end{aligned}$$

To the softmax function, we have $\mathrm{d}\,\mathrm{softmax}(\boldsymbol{v}) = (\mathrm{diag}(\boldsymbol{v}) - \boldsymbol{v}\boldsymbol{v}^\top)\mathrm{d}\boldsymbol{v}$. Therefore we have

$$\begin{aligned}\mathrm{d}\mathcal{L} =& \mathbb{E}_{\boldsymbol{X},y}\left(f(\boldsymbol{X}) - \mathrm{STS}_q(\boldsymbol{X})\right)^\top\boldsymbol{V}\boldsymbol{Z}\mathrm{d}(\mathrm{softmax}(\boldsymbol{Z}^\top\boldsymbol{W}\boldsymbol{z}_{\text{query}}))\\ &+ \mathbb{E}_{\boldsymbol{X},y}\left[(f(\boldsymbol{X}) - \mathrm{STS}_q(\boldsymbol{X}))^\top\mathrm{d}\boldsymbol{V}(\boldsymbol{Z}\boldsymbol{S}_y)\right]\\ =& -\mathbb{E}_{\boldsymbol{X},y}\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)^\top\boldsymbol{V}\boldsymbol{Z}\mathrm{d}(\mathrm{softmax}(\boldsymbol{Z}^\top\boldsymbol{W}\boldsymbol{z}_{\text{query}})) - \mathbb{E}_{\boldsymbol{X},y}\left[\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)^\top\mathrm{d}\boldsymbol{V}(\boldsymbol{Z}\boldsymbol{S}_y)\right]\\ =& -\mathbb{E}_{\boldsymbol{X},y}\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)^\top\boldsymbol{V}\boldsymbol{Z}(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top)\boldsymbol{Z}^\top\mathrm{d}\boldsymbol{W}\boldsymbol{z}_{\text{query}} - \mathbb{E}_{\boldsymbol{X},y}\left[\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)^\top\mathrm{d}\boldsymbol{V}(\boldsymbol{Z}\boldsymbol{S}_y)\right]\end{aligned}$$

We have the gradients for the two parameters:

$$\nabla_{\boldsymbol{W}} \mathcal{L} = -\mathbb{E}_{\boldsymbol{X},y}\left(\boldsymbol{Z}(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top)\boldsymbol{Z}^\top\boldsymbol{V}^\top(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y)\boldsymbol{z}_{\text{query}}^\top\right)$$
$$\nabla_{\boldsymbol{V}} \mathcal{L} = -\mathbb{E}_{\boldsymbol{X},y}\left(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)(\boldsymbol{Z}\boldsymbol{S}_y)^\top\right)$$

Thus we complete the proof. □

Along the gradient trajectory, if the value matrix can be aligned with the ground-truth $\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times T}\end{bmatrix}$, we can have the following nice form for the loss function.

Lemma D.2. *Denote $\boldsymbol{S}_y := \mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{query}) \in \mathbb{R}^T$ for certain q-sparse set y. Also, we define the q-hot vector $\boldsymbol{Y} = (\mathbb{1}\{1 \in y\}, \mathbb{1}\{2 \in y\}, ..., \mathbb{1}\{T \in y\})^\top$ for the subset $y \in \binom{[T]}{q}$. If $\boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times T}\end{bmatrix}$, the loss function can be represented as the following form:*

$$\mathcal{L}(\boldsymbol{\theta}(t)) = \frac{d}{2}\mathbb{E}_y\left[\left\|\frac{1}{q}\boldsymbol{Y} - \alpha(t)\boldsymbol{S}_y\right\|^2\right]$$

Proof. We have the architecture $f_{\boldsymbol{\theta}} = \boldsymbol{V}\boldsymbol{Z}\,\mathrm{softmax}(\boldsymbol{Z}^\top\boldsymbol{W}\boldsymbol{z}_{\text{query}})$ and the loss function in Equation (3):

$$\mathcal{L}(\boldsymbol{\theta}(t)) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y}\left[\|\mathrm{STS}_q(\boldsymbol{X}, y) - f_{\boldsymbol{\theta}}(\boldsymbol{X}, y)\|_2^2\right].$$

Then we can have the following expression

$$\mathcal{L}(\boldsymbol{\theta}(t)) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y}\left[\left\|\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}(t)\boldsymbol{Z}\boldsymbol{S}_y\right\|^2\right]$$

$$= \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y}\left[\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \alpha(t)\boldsymbol{X}\boldsymbol{S}_y\right)^\top\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \alpha(t)\boldsymbol{X}\boldsymbol{S}_y\right)\right]$$
$$= \frac{d}{2}\mathbb{E}_y\left[\left\|\frac{1}{q}\boldsymbol{Y} - \alpha(t)\boldsymbol{S}_y\right\|^2\right]$$

The last identity is due because the expectation of the covariance matrix $\boldsymbol{X}^\top\boldsymbol{X}$ is $d\boldsymbol{I}_T$. □

In the rest of this section, we characterize the convergence result training with GD for the settings above. Specifically, we consider the one-hot positional encoding. Here we define the encoding for all positions $i \in [T]$ and the encoding for any q-subset y.

Definition D.3 (One-hot Positional encoding)**.** The one-hot positional encoding is an orthogonal matrix $\boldsymbol{E} = \boldsymbol{I}_T$, with each column unit one-hot vector $\boldsymbol{e}_i$ as the positional encoding of the i-th token. Thus, for any set $y \in \binom{[T]}{q}$, it holds that $\boldsymbol{e}_y = \sum_{j\in y}\boldsymbol{e}_j$.

Remark. With one-hot positional encoding, the one-layer transformer can achieve $O(T)$ width cost, which is polynomially smaller than the memory lower bound $\Omega(Td)$ for FCNs. However, it cannot achieve *exponential* separation due to the inefficiency of one-hot encodings.

D.2. Joint training of value and attention matrix

Now, we analyze the dynamics of training the value matrix $\boldsymbol{V}$ and attention matrix $\boldsymbol{W}$ simultaneously with the same learning rate η. Instead of continuous gradient descent training, we directly train the transformer with gradient descent on the population loss. The following theorem characterizes the convergence of GD when training both layers simultaneously.

Theorem D.4 (Joint training with one-hot positional encoding)**.** *Suppose $2 \le q < T/4$. For any $\epsilon \in (0, \frac{dT}{100(T-q)q})$, $\eta \le \frac{1}{20d^2}$, $\boldsymbol{x}_{query} = \boldsymbol{0}_d$, if we apply gradient descent on the population loss in Equation (3) with zero initialization $\boldsymbol{W}(0) = \boldsymbol{0}_{(d+T)\times(d+T)}$, $\boldsymbol{V}(0) = \boldsymbol{0}_{d\times(d+T)}$, then after time $t \ge \tilde{O}(\frac{T^2 d}{\eta\epsilon})$, we have*

$$\mathcal{L}(\boldsymbol{\theta}(t)) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y}\left[\|\operatorname{STS}_q(\boldsymbol{X},y) - f_{\boldsymbol{\theta}}(\boldsymbol{X},y)\|_2^2\right] \le \epsilon.$$

We have the following lemma that will hold throughout the gradient descent training trajectory. We inductively prove that $\boldsymbol{V}$ and $\boldsymbol{W}$ are always along the ground-truth direction, respectively. For convenience, we consider all functions of $\boldsymbol{W}, \boldsymbol{V}$ including $\mathcal{L}, \boldsymbol{S}_y$ as a function of t.

Lemma D.5 (Induction Hypothesis)**.** *Given the initialization of $\boldsymbol{W}(0) = \boldsymbol{0}$ and $\boldsymbol{V}(t) = \boldsymbol{0}$, $\boldsymbol{x}_{query} = \boldsymbol{0}$, then along the gradient descent trajectory, for all $t \ge 0$, we have $\forall y, y' \in \binom{[T]}{q}$:*

1. *For all $t \ge 0$, there exists some time-dependent scalar $C(t)$ s.t.*

$$\boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times T}\\ \boldsymbol{0}_{T\times d} & \left(\boldsymbol{I}_T - \frac{1}{T}\boldsymbol{1}_T\boldsymbol{1}_T^\top\right)\end{bmatrix}.$$

2. *For all $t \ge 0$, there exists some time-dependent scalar $\alpha(t)$ s.t.*

$$\boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times T}\end{bmatrix}.$$

Proof. First, observe that the base case when $t = 0$ is clearly true. Thus, given that these statements hold up to time t, it suffices to prove these for next iteration $t + 1$, and the result will follow by induction.

Note that since the position-position block (right-bottom block) of $\boldsymbol{W}$ is always in direction of $\boldsymbol{I}_T - \frac{1}{T}\boldsymbol{1}_T\boldsymbol{1}_T^\top$ and other entries are 0, so we know

$$\boldsymbol{S}_y^{(t)}(i) = s_+(t) > s_-(t) = \boldsymbol{S}_y^{(t)}(j), \forall i \in y, j \notin y.$$

Once again, since we are only considering the dynamics at an arbitrary time t throughout the proof, we will abuse the notation as $s_+ := s_+(t)$, $s_- := s_-(t)$, $\alpha := \alpha(t)$, and $\boldsymbol{S}_y(i) := \boldsymbol{S}_y^{(t)}(i)$ which is the i-th entry of the softmax output.

First, suppose that the properties 1 and 2 hold. Recall that

$$\nabla_{\boldsymbol{V}}\mathcal{L} = \mathbb{E}_{\boldsymbol{X},y}\left(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)(\boldsymbol{Z}\boldsymbol{S}_y)^\top\right)$$

Now we consider the token-token block in $\boldsymbol{V}$. We have that the (i,j)-entry ($i \in [d], j \in [d]$) of $\nabla_{\boldsymbol{V}}\mathcal{L}(t)$ is

$$\begin{aligned}
\boldsymbol{e}_i^\top \nabla_{\boldsymbol{V}}\mathcal{L}(t)\boldsymbol{e}_j &= \mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{e}_i^\top\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)(\boldsymbol{Z}\boldsymbol{S}_y)^\top\boldsymbol{e}_j\right] && (\boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times T}\end{bmatrix}) \\
&= \mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{e}_i^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)(\boldsymbol{X}\boldsymbol{S}_y)^\top\boldsymbol{e}_j\right] \\
&= \mathbb{E}_{\boldsymbol{X},y}\left[\left(\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\right)_i(\boldsymbol{X}\boldsymbol{S}_y)_j\right] \\
&= \mathbb{E}_{\boldsymbol{X},y}\left[\left(\sum_{k=1}^T \boldsymbol{X}_{i,k}\left(\frac{\mathbb{1}\{k\in y\}}{q} - \alpha\boldsymbol{S}_y(k)\right)\right)\left(\sum_{k=1}^T \boldsymbol{X}_{j,k}\boldsymbol{S}_y(k)\right)\right]
\end{aligned}$$

We will case these entries based on whether they are on or off-diagonal.

Case I. Diagonal entries ($i = j$)

$$\begin{aligned}
\boldsymbol{e}_i^\top \nabla_{\boldsymbol{V}}\mathcal{L}(t)\boldsymbol{e}_j &= \mathbb{E}_{\boldsymbol{X},y}\left[\left(\sum_{k=1}^T \boldsymbol{X}_{i,k}\left(\frac{\mathbb{1}\{k\in y\}}{q} - \alpha\boldsymbol{S}_y(k)\right)\right)\left(\sum_{k=1}^T \boldsymbol{X}_{j,k}\boldsymbol{S}_y(k)\right)\right] \\
&= \mathbb{E}_{\boldsymbol{X},y}\left[\sum_{k=1}^T \boldsymbol{X}_{i,k}^2\left(\frac{\mathbb{1}\{k\in y\}}{q} - \alpha\boldsymbol{S}_y(k)\right)\boldsymbol{S}_y(k)\right] \\
&= \mathbb{E}_{y}\left[\sum_{k=1}^T \left(\frac{\mathbb{1}\{k\in y\}}{q} - \alpha\boldsymbol{S}_y(k)\right)\boldsymbol{S}_y(k)\right] \\
&= \mathbb{E}_{y}\left[q\left(\frac{1}{q} - \alpha s_+\right)s_+ - (T-q)\alpha s_-^2\right] \\
&= (1 - \alpha q s_+)s_+ - (T-q)\alpha s_-^2
\end{aligned}$$

Case II. Off-diagonal entries ($i \neq j$)

$$\boldsymbol{e}_i^\top \nabla_{\boldsymbol{V}}\mathcal{L}(t)\boldsymbol{e}_j = \mathbb{E}_{\boldsymbol{X},y}\left[\left(\sum_{k=1}^T \boldsymbol{X}_{i,k}\left(\frac{\mathbb{1}\{k\in y\}}{q} - \alpha\boldsymbol{S}_y(k)\right)\right)\left(\sum_{k=1}^T \boldsymbol{X}_{j,k}\boldsymbol{S}_y(k)\right)\right] = 0$$

This follows from the fact that each cross term of this sum is a product of two mean zero independent Gaussians, and thus the entire expression is 0. That means the gradient of $\boldsymbol{V}$ has its token-token block aligned with identity.

Then, we consider the position part. We have that the $(i, d+j)$-entry ($i \in [d], j \in [T]$) of $\nabla_{\boldsymbol{V}}\mathcal{L}(t)$ is

$$\begin{aligned}
\boldsymbol{e}_i^\top \nabla_{\boldsymbol{V}}\mathcal{L}(t)\boldsymbol{e}_j &= \mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{e}_i^\top\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)(\boldsymbol{Z}\boldsymbol{S}_y)^\top\boldsymbol{e}_{j+d}\right] && (\boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times T}\end{bmatrix}) \\
&= \mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{e}_i^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)(\boldsymbol{Z}\boldsymbol{S}_y)^\top\boldsymbol{e}_{j+d}\right] \\
&= \mathbb{E}_{\boldsymbol{X},y}\left[\left(\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\right)_i(\boldsymbol{S}_y)_j\right] \\
&= \mathbb{E}_{\boldsymbol{X},y}\left[\left(\sum_{k=1}^T \boldsymbol{X}_{i,k}\left(\frac{\mathbb{1}\{k\in y\}}{q} - \alpha\boldsymbol{S}_y(k)\right)\right)\boldsymbol{S}_y(j)\right]
\end{aligned}$$

Using induction hypothesis, since

$$\boldsymbol{W}(t) = C(t)\begin{bmatrix}\mathbf{0}_{d\times d} & \mathbf{0}_{d\times T}\\ \mathbf{0}_{T\times d} & \left(\boldsymbol{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^\top\right)\end{bmatrix}.$$

$\boldsymbol{S}_y(j)$ is independent of $\boldsymbol{X}$, so this expectation of the token-position block is all zero:

$$\begin{aligned}\boldsymbol{e}_i^\top \nabla_{\boldsymbol{V}}\mathcal{L}(t)\boldsymbol{e}_j &= \mathbb{E}_{\boldsymbol{X},y}\left[\left(\sum_{k=1}^{T}\boldsymbol{X}_{i,k}\left(\frac{\mathbb{1}\{k\in y\}}{q} - \alpha\boldsymbol{S}_y(k)\right)\right)\boldsymbol{S}_y(j)\right]\\ &= \mathbb{E}_y\left[\left(\sum_{k=1}^{T}\mathbb{E}_{\boldsymbol{X}}\left[\boldsymbol{X}_{i,k}\right]\left(\frac{\mathbb{1}\{k\in y\}}{q} - \alpha\boldsymbol{S}_y(k)\right)\right)\boldsymbol{S}_y(j)\right] = 0\end{aligned}$$

Combining the two blocks, we have that $\nabla_{\boldsymbol{V}}\mathcal{L}(t) = \left((1-\alpha q s_+)s_+ - (T-q)\alpha s_-^2\right)\begin{bmatrix}\boldsymbol{I}_d & \mathbf{0}_{d\times T}\end{bmatrix}$, which implies that for time $t+1$, $\boldsymbol{V}(t)$ will also be in the direction of $\begin{bmatrix}\boldsymbol{I}_d & \mathbf{0}_{d\times T}\end{bmatrix}$, hence proving the fourth property.

We now proceed to prove the third property with a similar analysis. We also just need to prove that the gradient can be expressed as:

$$\nabla_{\boldsymbol{W}}\mathcal{L}(t) = C_1(t)\begin{bmatrix}\mathbf{0}_{d\times d} & \mathbf{0}_{d\times T}\\ \mathbf{0}_{T\times d} & \left(\boldsymbol{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^\top\right)\end{bmatrix}$$

for some scalar $C_1(t)$, and then it indicates the next iteration $\boldsymbol{W}(t+1)$ will keep in the same direction. Recall that

$$\nabla_{\boldsymbol{W}}\mathcal{L}(t) = \mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{Z}\left(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\right)\boldsymbol{Z}^\top\boldsymbol{V}^\top\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{S}_y\right)\boldsymbol{z}_{\text{query}}^\top\right]$$

Since $\boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \mathbf{0}_{d\times T}\end{bmatrix}$ by the induction hypothesis, we have that this is equal to

$$\nabla_{\boldsymbol{W}}\mathcal{L}(t) = \alpha\,\mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{Z}\left(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\right)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\boldsymbol{z}_{\text{query}}^\top\right]$$

We first consider the first d columns of the gradient, which is

$$\begin{aligned}\nabla_{\boldsymbol{W}}\mathcal{L}(t)_{:,\le d} &= \alpha\,\mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{Z}\left(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\right)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right){\boldsymbol{x}_{\text{query}}}^\top\right]\\ &= \mathbf{0}_{(d+T)\times d}.\end{aligned}$$

Because $\boldsymbol{x}_{\text{query}} = \mathbf{0}_d$.

Next, we consider the top-right block of the gradient $\nabla_{\boldsymbol{W}}\mathcal{L}(t)_{\le d, d:T+d}$. By induction hypothesis, $\boldsymbol{S}_y = \mathrm{softmax}(\boldsymbol{Z}^\top\boldsymbol{W}(t)\boldsymbol{z}_{\text{query}}) = \mathrm{softmax}(C(t)(\boldsymbol{I} - \frac{1}{T}\mathbf{1}\mathbf{1}^\top)\boldsymbol{e}_y)$ at time t, which is independent of $\boldsymbol{X}$. Therefore, by symmetry the gradient of this block is

$$\begin{aligned}\nabla_{\boldsymbol{W}}\mathcal{L}(t)_{\le d, d:T+d} &= \alpha\,\mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{X}\left(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\right)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right){\boldsymbol{e}_y}^\top\right]\\ &= \frac{1}{2}\alpha\,\mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{X}\left(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\right)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right){\boldsymbol{e}_y}^\top\right]\\ &\quad - \frac{1}{2}\alpha\,\mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{X}\left(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\right)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right){\boldsymbol{e}_y}^\top\right] = \mathbf{0}\end{aligned}$$

Finally, we consider the position-position block. In particular, since $\mathbb{E}\,\boldsymbol{X}^\top\boldsymbol{X} = d\boldsymbol{I}_d$ we have that for $i, j \in [T]$,

$$\nabla_{\boldsymbol{W}}\mathcal{L}(t)_{(i+d),(j+d)} = \alpha\,\mathbb{E}_{\boldsymbol{X},y}\left[\boldsymbol{e}_i^\top\left(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\right)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\boldsymbol{e}_y^\top\boldsymbol{e}_j\right]$$

(Here $\boldsymbol{e}_i$ are one-hot vectors in $\mathbb{R}^T$.)

$$
\begin{aligned}
&= \alpha d\, \mathbb{E}_y \left[\left(\boldsymbol{S}_y(i) \boldsymbol{e}_i^\top - \boldsymbol{S}_y(i) \boldsymbol{S}_y^\top \right) \left(\frac{1}{q} \boldsymbol{Y} - \alpha \boldsymbol{S}_y \right) \mathbb{1}\{j \in y\} \right] \\
&= \frac{\alpha d}{\binom{T}{q}} \sum_{y: j \in y} \left(\boldsymbol{S}_y(i) \boldsymbol{e}_i^\top - \boldsymbol{S}_y(i) \boldsymbol{S}_y^\top \right) \left(\frac{1}{q} \boldsymbol{Y} - \alpha \boldsymbol{S}_y \right) \\
&= \frac{\alpha d}{\binom{T}{q}} \sum_{y: j \in y} \left(\boldsymbol{S}_y(i) \mathbb{1}\{i \in y\}/q - \alpha \boldsymbol{S}_y(i)^2 - \boldsymbol{S}_y(i) s_+ + \alpha \boldsymbol{S}_y(i) \|\boldsymbol{S}_y\|^2 \right)
\end{aligned}
$$

We now case this between on and off diagonals again.

Case I. Diagonal entries $(i = j)$

$$
\begin{aligned}
\boldsymbol{e}_i^\top \nabla_{\boldsymbol{W}} \mathcal{L}(t) \boldsymbol{e}_j &= \frac{\alpha d}{\binom{T}{q}} \sum_{y: j \in y} \left(\boldsymbol{S}_y(i) \mathbb{1}\{i \in y\}/q - \alpha \boldsymbol{S}_y(i)^2 - \boldsymbol{S}_y(i) s_+ + \alpha \boldsymbol{S}_y(i) \|\boldsymbol{S}_y\|^2 \right) \\
&= \frac{\alpha d \binom{T-1}{q-1}}{\binom{T}{q}} s_+ \left(\frac{1}{q} - \alpha s_+ - s_+ + \alpha \|\boldsymbol{S}_y\|^2 \right) \\
&= \frac{\alpha d q}{T} s_+ \left(\frac{1}{q} - \alpha s_+ - s_+ + \alpha q s_+^2 + \frac{\alpha (1 - q s_+)^2}{T - q} \right) \\
&= \frac{\alpha d}{T} s_+ \left(1 - \alpha q s_+ - q s_+ + \alpha q^2 s_+^2 + \frac{\alpha q (1 - q s_+)^2}{T - q} \right) \\
&= \frac{\alpha d}{T} s_+ \left((1 - \alpha)(1 - q s_+) + \frac{T \alpha}{T - q} (1 - q s_+)^2 \right)
\end{aligned}
$$

Case II. Off-diagonal entries $(i \neq j)$

$$
\begin{aligned}
\boldsymbol{e}_i^\top \nabla_{\boldsymbol{W}} \mathcal{L}(t) \boldsymbol{e}_j &= \frac{\alpha d}{\binom{T}{q}} \sum_{y: j \in y} \left(\boldsymbol{S}_y(i) \mathbb{1}\{i \in y\}/q - \alpha \boldsymbol{S}_y(i)^2 - \boldsymbol{S}_y(i) s_+ + \alpha \boldsymbol{S}_y(i) \|\boldsymbol{S}_y\|^2 \right) \\
&= \frac{\alpha d}{\binom{T}{q}} \sum_{y: j \in y, i \in y} \left(\boldsymbol{S}_y(i) \mathbb{1}\{i \in y\}/q - \alpha \boldsymbol{S}_y(i)^2 - \boldsymbol{S}_y(i) s_+ + \alpha \boldsymbol{S}_y(i) \|\boldsymbol{S}_y\|^2 \right) \\
&+ \frac{\alpha d}{\binom{T}{q}} \sum_{y: j \in y, i \notin y} \left(\boldsymbol{S}_y(i) \mathbb{1}\{i \in y\}/q - \alpha \boldsymbol{S}_y(i)^2 - \boldsymbol{S}_y(i) s_+ + \alpha \boldsymbol{S}_y(i) \|\boldsymbol{S}_y\|^2 \right)
\end{aligned}
$$

The first of these terms is similar in structure to the diagonal entry case:

$$
\begin{aligned}
&\frac{\alpha d}{\binom{T}{q}} \sum_{y: j \in y, i \in y} \left(\boldsymbol{S}_y(i) \mathbb{1}\{i \in y\}/q - \alpha \boldsymbol{S}_y(i)^2 - \boldsymbol{S}_y(i) s_+ + \alpha \boldsymbol{S}_y(i) \|\boldsymbol{S}_y\|^2 \right) \\
&= \frac{\alpha d \binom{T-2}{q-2}}{\binom{T}{q}} s_+ \left(\frac{1}{q} - \alpha s_+ - s_+ + \alpha \|\boldsymbol{S}_y\|^2 \right) \\
&= \frac{\alpha d (q-1) q}{T(T-1)} s_+ \left(\frac{1}{q} - \alpha s_+ - s_+ + \alpha \|\boldsymbol{S}_y\|^2 \right) \\
&= \frac{\alpha d (q-1)}{T(T-1)} s_+ \left((1 - \alpha)(1 - q s_+) + \frac{T \alpha}{T - q} (1 - q s_+)^2 \right)
\end{aligned}
$$

where the last line follows from the calculation in the diagonal entry case.

Looking at the second term, we have:

$$
\frac{\alpha d}{\binom{T}{q}} \sum_{y: j \in y, i \notin y} \left(\boldsymbol{S}_y(i) \mathbb{1}\{i \in y\}/q - \alpha \boldsymbol{S}_y(i)^2 - \boldsymbol{S}_y(i) s_+ + \alpha \boldsymbol{S}_y(i) \|\boldsymbol{S}_y\|^2 \right)
$$

$$
\begin{aligned}
&= \frac{\alpha d\binom{T-2}{q-1}}{\binom{T}{q}} s_-\big(-\alpha s_- - s_+ + \alpha\|\boldsymbol{S}_y\|^2\big) \\
&= \frac{\alpha dq(T-q)}{T(T-1)}\frac{1-qs_+}{T-q}\bigg(-\frac{\alpha(1-qs_+)}{T-q} - s_+ + \alpha q s_+^2 + \frac{\alpha(1-qs_+)^2}{T-q}\bigg) \\
&= \frac{\alpha dq}{T(T-1)}(1-qs_+)\bigg(-s_+ + \alpha q s_+^2 - \frac{qs_+\alpha(1-qs_+)}{T-q}\bigg)
\end{aligned}
$$

From this, we get that $\boldsymbol{e}_i^\top \nabla_{\boldsymbol{W}}\mathcal{L}(t)\boldsymbol{e}_j$ is the following expression multiplied by $\frac{\alpha d}{T(T-1)}$

$$
\begin{aligned}
&(q-1)s_+\bigg((1-\alpha)(1-qs_+) + \frac{T\alpha}{T-q}(1-qs_+)^2\bigg) + q(1-qs_+)\bigg(-s_+ + \alpha q s_+^2 - \frac{qs_+\alpha(1-qs_+)}{T-q}\bigg) \\
&= (q-1)\bigg(s_+ - s_+\alpha - qs_+^2 + \alpha q s_+^2 + \frac{T\alpha s_+}{T-q}(1-qs_+)^2\bigg) \\
&\quad + q\bigg(-s_+ + qs_+^2 + \alpha q s_+^2 - \alpha q^2 s_+^3 - \frac{qs_+\alpha(1-qs_+)^2}{T-q}\bigg) \\
&= -\bigg(s_+ - s_+\alpha - qs_+^2 + \alpha q s_+^2 + \frac{T\alpha s_+}{T-q}(1-qs_+)^2\bigg) + q\big(-s_+\alpha + 2\alpha q s_+^2 - \alpha q^2 s_+^3 + \alpha s_+(1-qs_+)^2\big) \\
&= -s_+ + s_+\alpha + qs_+^2 - \alpha q s_+^2 - \frac{T\alpha s_+}{T-q}(1-qs_+)^2 - qs_+\alpha + 2\alpha q^2 s_+^2 - \alpha q^3 s_+^3 + \alpha q s_+(1-qs_+)^2 \\
&= -s_+\bigg((1-\alpha)(1-qs_+) + \frac{T\alpha}{T-q}(1-qs_+)^2\bigg)
\end{aligned}
$$

Thus, we have that

$$
\boldsymbol{e}_i^\top \nabla_{\boldsymbol{W}}\mathcal{L}(t)\boldsymbol{e}_j = \frac{\alpha d}{T(T-1)}\bigg((1-\alpha)(1-qs_+) + \frac{T\alpha}{T-q}(1-qs_+)^2\bigg)(-s_+)
$$

for off-diagonal entries.

Combining our on and off-diagonal entry calculations, we obtain that

$$
\nabla_{\boldsymbol{W}}\mathcal{L}(t) = \bigg(\frac{\alpha d}{T-1}s_+\bigg((1-\alpha)(1-qs_+) + \frac{T\alpha}{T-q}(1-qs_+)^2\bigg)\bigg)\bigg(\boldsymbol{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^\top\bigg)
$$

Combine all the four blocks, we know the gradient of $\boldsymbol{W}$ is along the $\begin{bmatrix}\mathbf{0}_{d\times d} & \mathbf{0}_{d\times T} \\ \mathbf{0}_{T\times d} & (\boldsymbol{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^\top)\end{bmatrix}$. This proves property 1 for iteration $t+1$ as desired. □

Remark. After proving the induction lemma about the evolving direction of $\boldsymbol{W}(t)$ and $\boldsymbol{V}(t)$, the optimization problem can be reduced to analyzing the two variable dynamics of $C(t)$ and $\alpha(t)$.

We can now proceed to prove the main theorem for the joint training algorithm by analyzing the $C(t)$ and $\alpha(t)$ dynamics. One can refer to the main paper for proof ideas.

Proof of Theorem D.4. After Lemma D.5 shows that $\boldsymbol{V}$ and $\boldsymbol{W}$ are always along the ground-truth direction: $\boldsymbol{V}(t) = \alpha(t)\boldsymbol{I}_d, \boldsymbol{W}(t) = C(t)(\boldsymbol{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^\top)$, the dynamics of the parameter matrices then can be characterized by two scalar variable $\alpha(t)$ and $C(t)$. Our update rules become

$$
\begin{aligned}
\alpha(t+1) &= \alpha(t) + \eta\bigg((1-\alpha q s_+)s_+ - \frac{\alpha(1-qs_+)^2}{T-q}\bigg) \\
&= \alpha(t) + \eta s_+\big(1 - \frac{\alpha(t)(Tqs_+^2 - 2qs_+ + 1)}{(T-q)s_+}\big)
\end{aligned}
$$

$$C(t+1) = C(t) + \eta\frac{\alpha d}{T-1}s_+\left((1-\alpha)(1-qs_+) + \frac{T\alpha}{T-q}(1-qs_+)^2\right)$$
$$= C(t) + \eta\frac{\alpha d}{T-1}s_+(1-qs_+)\left(1 + \frac{q\alpha}{T-q}(1-Ts_+)\right)$$

Here, $s_+(t) := \boldsymbol{S}_y^{(t)}(i), i \in y$ is the correct position softmax probability value at time t. We omit the t here for clarity since at each iteration only $s_+(t)$ is related. Along the trajectory, $s_+ \leq \frac{1}{q}$ by its definition: since $\boldsymbol{W}$ is along $(\boldsymbol{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^\top)$, all $i \in y$ has the same softmax probability $\boldsymbol{S}_y^{(t)}(i)$, and thus they cannot exceed the upper bound $1/q$.

Now we prove that these update rules take $\alpha \to 1, C(t) \to +\infty$ when $t \to +\infty$. Note that when s_+ is fixed, $\alpha(t)$ has a stationary point $\alpha^*(t) = \frac{(T-q)s_+}{Tqs_+^2 - 2qs_+ + 1}$. That means we can write the α dynamics into:

$$\alpha(t+1) = \alpha(t) + \eta s_+\left(1 - \frac{\alpha(t)}{\alpha^*(t)}\right).$$

One can easily check that when $s_+ \in (\frac{1}{T}, \frac{1}{q}), \alpha^*(t) \geq 1$, and it achieves maximum at $s_+ = \frac{1}{\sqrt{Tq}}$.

To characterize this limit above when $t \to \infty$, we need to prove the following two arguments:

1. $C(t)$ is non-decreasing for all $t \geq 0$.
2. $\alpha(t)$ goes through 2 phases:

   Phase I. α monotonically grows to $1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$ at some time t_1.

   Phase II. α stays within an interval whose upper bound is close to α^* after time t_1:

$$\alpha(t) \in \left[1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}, \left(1 + \frac{(8d-1)(1-qs_+)}{8dqs_+(Ts_+-1)}\right)\alpha^*(t)\right] \tag{IH1}$$

Phase I. We inductively prove that both $\alpha(t), C(t)$ are non-decreasing. For $t = 0$, by zero initialization, $C(0) = C(1) = 0, \alpha(1) = \frac{\eta}{T} > 0$ so it holds for $t = 0$. Suppose it holds for some $t < t_1$ before α hit $1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$. Then we know for $\alpha(t+1)$, we have the update rule:

$$\alpha(t+1) = \alpha(t) + \eta s_+\left(1 - \frac{\alpha(t)}{\alpha^*(t)}\right) \geq \alpha(t) + \eta\frac{1}{T}(1 - \frac{\alpha(t)}{\alpha^*(t)}) \qquad \text{(Induction Hyp. } s_+ \geq \tfrac{1}{T})$$
$$\geq \alpha(t) + \eta\frac{1}{T}(1-\alpha(t)) \qquad (\alpha^* \geq 1)$$
$$\geq \alpha(t) + 0.1\eta\sqrt{\frac{q(T-q)\epsilon}{dT^3}}. \qquad (\alpha \leq 1 - 0.1\sqrt{\tfrac{q(T-q)\epsilon}{dT}})$$

So α is non-decreasing. Meanwhile, for $C(t+1)$:

$$C(t+1) = C(t) + \eta\frac{\alpha d}{T-1}s_+(1-qs_+)\left(1 + \frac{q\alpha}{T-q}(1-Ts_+)\right)$$
$$\geq C(t) + \eta\frac{\alpha d}{T-1}s_+(1-qs_+)\left(1 + \frac{q}{T-q}(1-Ts_+)\right) \qquad (\alpha < 1, (1-Ts_+) < 0)$$
$$= C(t) + \eta\frac{\alpha d}{T-1}s_+(1-qs_+)\frac{T-q+q-Tqs_+}{T-q}$$
$$= C(t) + \eta\frac{\alpha dT}{(T-1)(T-q)}s_+(1-qs_+)^2 \geq C(t)$$

so they are both non-decreasing. Then we need to upper bound the time t_1 for $\alpha(t)$ to reach $1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$: by the update above we have

$$\alpha(t+1) = \alpha(t) + \eta s_+\left(1 - \frac{\alpha(t)}{\alpha^*(t)}\right) \geq \alpha(t) + \eta\frac{1}{T}(1 - \frac{\alpha(t)}{\alpha^*(t)}) \qquad \text{(Induction Hyp. } s_+ \geq \tfrac{1}{T})$$

$$\geq \alpha(t) + \eta\frac{1}{T}(1-\alpha(t)) \qquad (\alpha^* \geq 1)$$
$$\Rightarrow 1-\alpha(t+1) \leq (1-\eta/T)(1-\alpha(t)) \leq ... \leq (1-\eta/T)^t(1-\alpha(0)).$$

Thus for $\alpha(t+1) \geq 1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$, it takes at most $O(\frac{T\log\frac{d}{\epsilon}}{\eta})$ iterations.

Phase II. In this phase, we first consider $s_+(t) < \frac{1}{\sqrt{Tq}}$. In this case, $\alpha^*(t+1) > \alpha^*(t)$, and inductively

$$\alpha^*(t) - \alpha(t+1) = \alpha^*(t) - \alpha(t) - \frac{\eta s_+(t)}{\alpha^*(t)}(\alpha^*(t)-\alpha(t)) = (1-\frac{\eta s_+(t)}{\alpha^*(t)})(\alpha^*(t)-\alpha(t)) > 0,$$

$\alpha(t+1) < \alpha^*(t+1)$ always holds, and the induction hypothesis holds for $t+1$.

All arguments below is based on $s_+(t) \geq \frac{1}{\sqrt{Tq}}$. We first verify that within the induction hypothesis range, $C(t+1) \geq C(t)$.

$$\begin{aligned}
C(t+1) &= C(t) + \eta\frac{\alpha d}{T-1}s_+(1-qs_+)\left(1+\frac{q\alpha}{T-q}(1-Ts_+)\right) \\
&\geq C(t) + \eta\frac{\alpha d}{T-1}s_+(1-qs_+)\left(1+\left(1+\frac{(d-1)(1-qs_+)}{dqs_+(Ts_+-1)}\right)\alpha^*(t)\frac{q}{T-q}(1-Ts_+)\right) \\
&\qquad\qquad (\alpha \leq \left(1+\tfrac{(d-1)(1-qs_+)}{dqs_+(Ts_+-1)}\right)\alpha^*(t), (1-Ts_+) < 0) \\
&= C(t) + \eta\frac{\alpha d}{T-1}s_+(1-qs_+)\cdot\frac{1}{d}\frac{1-qs_+}{Tqs_+^2-2qs_++1} \\
&= C(t) + \eta\frac{\alpha}{(T-1)(Tqs_+^2-2qs_++1)}s_+(1-qs_+)^2 \geq C(t)
\end{aligned}$$

Next, we first divide $\alpha(t)$'s possible range into two parts: $\alpha(t) \leq \left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)$ and $\alpha(t) \in \left[\left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t), \left(1+\frac{(8d-1)(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)\right]$.

For the first part, we prove the following statement **(S1)**:

$$\text{If } \alpha(t) \leq \left(1+\tfrac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t), \text{ the next step}$$
$$\alpha(t+1) \leq \left(1+\frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\alpha^*(t+1).$$

If **(S1)** is true, then we know once $\alpha(t) \leq \left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)$, $\alpha(t+1)$ satisfy the induction hypothesis. After proving **(S1)**, the only part left is when

$$\alpha(t) \in \left(\left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t), \left(1+\frac{(8d-1)(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)\right].$$

We prove statement **(S1)** by proving

$$\left(1+\frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\alpha^*(t+1) \geq \left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t) \qquad \textbf{(S2)}$$

When the inequality **(S2)** above is proved, then **(S1)** is proved. This is because: if $\alpha(t) < \alpha^*(t)$, then by update rule we have

$$\alpha^*(t) - \alpha(t+1) = \alpha^*(t) - \alpha(t) - \frac{\eta s_+(t)}{\alpha^*(t)}(\alpha^*(t)-\alpha(t)) = (1-\frac{\eta s_+(t)}{\alpha^*(t)})(\alpha^*(t)-\alpha(t)) > 0,$$

$$\alpha(t+1) < \alpha^*(t) \leq \left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)$$

$$\leq \left(1 + \frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\alpha^*(t+1)$$

If $\alpha(t) \geq \alpha^*(t)$, then $\alpha(t+1) \leq \alpha(t) \leq \left(1 + \frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)$, and therefore smaller than $\left(1 + \frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\alpha^*(t+1)$.

Now we prove **(S2)** by expanding the $s_+(t+1)$ using the update rule of $C(t)$. Denote $\Delta C(t) := C(t+1) - C(t)$. Since $\eta \leq \frac{1}{20d^2}$, $\Delta C(t) < \frac{1}{5}$. Then we have

$$\begin{aligned}
s_+(t+1) &= \frac{1}{q + (T-q)e^{-C(t)-\Delta C(t)}} \leq \frac{1}{q + (T-q)e^{-C(t)}(1-\Delta C(t))} \\
&= \frac{1}{q + (T-q)e^{-C(t)} - (T-q)e^{-C(t)}\Delta C(t)} \\
&\leq \frac{1}{q + (T-q)e^{-C(t)}} + \frac{5}{4}\left(\frac{1}{q + (T-q)e^{-C(t)}}\right)^2 (T-q)e^{-C(t)}\Delta C(t)
\end{aligned}$$

The last inequality is due to $\Delta C(t) < \frac{1}{5}$. Then we have

$$\begin{aligned}
s_+(t+1) &\leq \frac{1}{q + (T-q)e^{-C(t)}} + \frac{5}{4}\left(\frac{1}{q + (T-q)e^{-C(t)}}\right)^2 (T-q)e^{-C(t)}\Delta C(t) \\
&= s_+(t) + \frac{5}{4}s_+^2(t)\left(\frac{1}{s_+} - q\right)\Delta C(t) \\
&= s_+(t) + \frac{5}{4}s_+(1-qs_+)\Delta C(t)
\end{aligned}$$

Then we consider the decrement of $\alpha^*(t+1)$ and $\frac{(1-qs_+(t+1))}{qs_+(t+1)(Ts_+(t+1)-1)}$.

$$\begin{aligned}
\alpha^*(t+1) &= \frac{(T-q)s_+(t+1)}{Tqs_+(t+1)^2 - 2qs_+(t+1) + 1} = \frac{(T-q)}{Tqs_+(t+1) - 2q + 1/s_+(t+1)} \\
&\geq \frac{(T-q)}{Tqs_+(t) - 2q + 1/s_+(t) + \frac{5}{4}Tqs_+(t)(1-qs_+(t))\Delta C(t)} \\
&\geq \frac{(T-q)}{Tqs_+(t) - 2q + 1/s_+(t)} - \frac{\frac{5}{4}(T-q)Tqs_+(t)(1-qs_+(t))\Delta C(t)}{(Tqs_+(t) - 2q + 1/s_+(t))^2} \\
&= \alpha^*(t) - \frac{5(T-q)Tqs_+^3(t)(1-qs_+(t))\Delta C(t)}{4(Tqs_+^2(t) - 2qs_+(t) + 1)^2} = \alpha^*(t) - \frac{5Tqs_+^2(t)(1-qs_+(t))\Delta C(t)}{4(Tqs_+^2(t) - 2qs_+(t) + 1)}\alpha^*(t)
\end{aligned}$$

$$\begin{aligned}
&\frac{(1-qs_+(t+1))}{qs_+(t+1)(Ts_+(t+1)-1)} \\
&= \frac{1}{qs_+(t+1)(Ts_+(t+1)-1)} - \frac{1}{(Ts_+(t+1)-1)} \\
&\geq \frac{1}{qs_+(t+1)(Ts_+(t+1)-1)} - \frac{1}{(Ts_+(t)-1)} && (s_+(t+1) \geq s_+(t)) \\
&= \frac{1}{qT} \cdot \frac{1}{s_+(t+1)} \cdot \frac{1}{s_+(t+1) - \frac{1}{T}} - \frac{1}{(Ts_+(t)-1)} \\
&\geq \frac{1}{qT}\left(\frac{1}{s_+(t)} - \frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)}{s_+^2(t)}\right)\left(\frac{1}{s_+(t) - \frac{1}{T}} - \frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)}{(s_+ - \frac{1}{T})^2}\right) - \frac{1}{(Ts_+(t)-1)} \\
&\geq \frac{(1-qs_+(t))}{qs_+(t)(Ts_+(t)-1)} - \frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)}{qs_+(t)^2(Ts_+-1)} - \frac{\frac{5}{4}s_+(1-qs_+)T\Delta C(t)}{qs_+(t)(Ts_+-1)^2} \\
&= \frac{(1-qs_+(t))}{qs_+(t)(Ts_+(t)-1)} - \frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)(2Ts_+-1)}{qs_+(t)^2(Ts_+-1)^2}
\end{aligned}$$

Then plug in the original term, we have the lower bound for

$$\begin{aligned}
&\left(1+\frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\alpha^*(t+1)\\
\geq&\left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)+\frac{(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\alpha^*(t)\\
&-\frac{5s_+(1-qs_+)\Delta C(t)(2Ts_+-1)}{4qs_+(t)^2(Ts_+-1)^2}\alpha^*(t)\\
&-\left(1+\frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\frac{5Tqs_+^2(t)(1-qs_+(t))\Delta C(t)}{4(Tqs_+^2(t)-2qs_+(t)+1)}\alpha^*(t)
\end{aligned}$$

Since $\frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\leq\frac{8d-1}{8d}\leq 1$ when $s_+\geq\frac{1}{\sqrt{Tq}}$, we have

$$\begin{aligned}
&\left(1+\frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\alpha^*(t+1)\\
\geq&\left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)+\frac{(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\alpha^*(t)\\
&-2\cdot\frac{5Tqs_+^2(t)(1-qs_+(t))\Delta C(t)}{4(Tqs_+^2(t)-2qs_+(t)+1)}\alpha^*(t)-\frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)(2Ts_+-1)}{qs_+(t)^2(Ts_+-1)^2}\alpha^*(t) \qquad \textbf{(*)}
\end{aligned}$$

Then we need to prove that (here $s_+:=s_+(t)$) to show **(S2)**.

$$\frac{(1-qs_+)}{8dqs_+(Ts_+-1)}\geq\frac{5Tqs_+^2(1-qs_+)\Delta C(t)}{2(Tqs_+^2-2qs_++1)}+\frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)(2Ts_+-1)}{qs_+^2(Ts_+-1)^2}$$

We have that the right hand side has the following upper bound ($T\geq 4q$):

$$\begin{aligned}
&\frac{5Tqs_+^2(1-qs_+)\Delta C(t)}{2(Tqs_+^2-2qs_++1)}+\frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)(2Ts_+-1)}{qs_+^2(Ts_+-1)^2}\\
\leq&\frac{5Tqs_+^2(1-qs_+)\Delta C(t)}{2qs_+(Ts_+-1)}+\frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)(2Ts_+-1)}{qs_+^2(Ts_+-1)^2} \qquad (s_+\leq 1/q.)\\
\leq&\frac{5Tqs_+^2(1-qs_+)\Delta C(t)}{2qs_+(Ts_+-1)}+\frac{\frac{15}{4}s_+(1-qs_+)\Delta C(t)}{qs_+^2(Ts_+-1)} \qquad \textbf{(**)}
\end{aligned}$$

Let $\alpha=(1+\gamma\frac{1-qs_+}{qs_+(Ts_+-1)})\alpha^*$, then we have the upper bound for the update

$$\begin{aligned}
\Delta C(t)&=\eta(1-\gamma)(1+\gamma\frac{1-qs_+}{qs_+(Ts_+-1)})\alpha^*\frac{d}{(T-1)(Tqs_+^2-2qs_++1)}s_+(1-qs_+)^2\\
&\leq\eta(1-\gamma^2)\frac{d(T-q)s_+^2(1-qs_+)^2}{(T-1)(Tqs_+^2-2qs_++1)^2}\\
&\leq\eta(1-\gamma^2)\frac{d(T-q)(1-qs_+)^2}{(T-1)(Tqs_+-2q+1/s_+)^2}\\
&\leq\eta(1-\gamma^2)\frac{d(T-q)(1-\frac{\sqrt{q}}{\sqrt{T}})^2}{(T-1)(2\sqrt{Tq}-2q)^2}\leq\frac{\eta d}{4qT}. \qquad (s_+\geq 1/\sqrt{Tq}.)
\end{aligned}$$

Since $\eta\leq\frac{1}{20d^2}$, plug the upper bound for $\Delta C(t)$ back to the two terms in **(**)** respectively and we proved the inequality. And therefore, **(S2)** is proved, which also leads to **(S1)**.

Finally, we consider $\alpha(t)\in\left[\left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t),\left(1+\frac{(8d-1)(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)\right]$. Now, since by update rule,

$$\alpha(t+1)=\alpha(t)+\eta s_+\left(1-\frac{\alpha(t)}{\alpha^*(t)}\right)\leq\left(1+\frac{(8d-1)(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)-\frac{(4d-1)\eta(1-qs_+(t))}{4dq(Ts_+(t)-1)}$$

We just need to prove that

$$\left(1+\frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\alpha^*(t+1)$$
$$\geq\left(1+\frac{(8d-1)(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)-\frac{(4d-1)\eta(1-qs_+(t))}{4dq(Ts_+(t)-1)}$$

Note Equation (*) gives the lower bound for the left hand side:

$$\begin{aligned}&\left(1+\frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\alpha^*(t+1)\\ \geq&\left(1+\frac{(8d-1)(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)-\frac{5Tqs_+^2(t)(1-qs_+(t))\Delta C(t)}{2(Tqs_+^2(t)-2qs_+(t)+1)}\alpha^*(t)\\ &-\frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)(2Ts_+-1)}{qs_+(t)^2(Ts_+-1)^2}\alpha^*(t)\end{aligned}\tag{\#}$$

Yet when $\alpha(t)\geq\left(1+\frac{(4d-1)(1-qs_+(t))}{4dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t)$, we have a better upper bound for $\Delta C(t)$:

$$\begin{aligned}\Delta C(t)&=\eta(1-\gamma)(1+\gamma\frac{1-qs_+}{qs_+(Ts_+-1)})\alpha^*\frac{d}{(T-1)(Tqs_+^2-2qs_++1)}s_+(1-qs_+)^2\\ &\leq\eta(1-\gamma^2)\frac{d(T-q)s_+^2(1-qs_+)^2}{(T-1)(Tqs_+^2-2qs_++1)^2}\\ &\leq\eta\left(1-\left(\frac{4d-1}{4d}\right)^2\right)\frac{d(T-q)s_+^2(1-qs_+)^2}{(T-1)(Tqs_+^2-2qs_++1)^2}\\ &\leq\frac{\eta}{2}\frac{(T-q)s_+^2(1-qs_+)^2}{(T-1)(Tqs_+^2-2qs_++1)^2}\end{aligned}$$

Then we need to bound both terms in Equation (#) (for simplicity denote s_+ as $s_+(t)$):

$$\begin{aligned}&\frac{5Tqs_+^2(t)(1-qs_+(t))\Delta C(t)}{2(Tqs_+^2(t)-2qs_+(t)+1)}\alpha^*(t)\leq\frac{5\eta Tqs_+^5(1-qs_+)^3(T-q)^2}{4(T-1)(Tqs_+^2-2qs_++1)^4}\\ =&\frac{5\eta}{4}\cdot\frac{T(T-q)^2}{T-1}\cdot\frac{(1-qs_+)}{Tqs_+^2-2qs_++1}\cdot\frac{qs_+^5(1-qs_+)^2}{(Tqs_+^2-2qs_++1)^3}\\ \leq&\frac{5\eta}{4}\cdot\frac{T(T-q)^2}{T-1}\cdot\frac{(1-qs_+)}{qs_+(Ts_+-1)}\cdot\frac{qs_+^5(1-qs_+)^2}{(Tqs_+^2-2qs_++1)^3}&(s_+\leq\tfrac{1}{q})\\ =&\frac{5\eta}{4}\cdot\frac{T(T-q)^2}{T-1}\cdot\frac{(1-qs_+)}{q(Ts_+-1)}\cdot\frac{q(1-qs_+)^2}{(Tqs_+^2-2qs_++1)(Tq-2q/s_++1/s_+^2)^2}\\ \leq&\frac{5\eta}{4}\cdot\frac{T(T-q)^2}{T-1}\cdot\frac{(1-qs_+)}{q(Ts_+-1)}\cdot\frac{q}{2(Tq-q^2)^2}&(s_+\geq\tfrac{1}{\sqrt{Tq}}.)\\ =&\frac{5\eta}{8q}\cdot\frac{T}{T-1}\cdot\frac{(1-qs_+)}{q(Ts_+-1)}\leq\frac{5\eta}{14}\frac{(1-qs_+)}{q(Ts_+-1)}&(q\geq2,T\geq4q)\end{aligned}$$

$$\begin{aligned}&\frac{5s_+(1-qs_+)\Delta C(t)(2Ts_+-1)}{4qs_+(t)^2(Ts_+-1)^2}\alpha^*(t)\\ \leq&\frac{5(1-qs_+)(2Ts_+-1)}{4qs_+(Ts_+-1)^2}\cdot\frac{\eta(T-q)^2s_+^3(1-qs_+)^2}{2(T-1)(Tqs_+^2-2qs_++1)^3}&(\text{Plug in }\Delta C(t))\\ =&\frac{5\eta(1-qs_+)(2Ts_+-1)(T-q)^2s_+^2(1-qs_+)^2}{8q(Ts_+-1)^2(T-1)(Tqs_+^2-2qs_++1)^3}\end{aligned}$$

$$\leq \frac{5\eta(1-qs_+)}{8q(Ts_+-1)} \cdot \frac{2Ts_+-1}{Ts_+-1} \cdot \frac{(T-q)^2}{T-1} \cdot \frac{1}{Tq-2q/s_+ + 1/s_+^2} \cdot \frac{(1-qs_+)^2}{(Tqs_+^2-2qs_++1)^2}$$
$$\leq \frac{5\eta(1-qs_+)}{8q(Ts_+-1)} \cdot 3 \cdot \frac{(T-q)^2}{T-1} \frac{1}{Tq-q^2} \frac{1}{4} \leq \frac{15\eta(1-qs_+)}{32q^2(Ts_+-1)} \leq \frac{15\eta(1-qs_+)}{64q(Ts_+-1)}.$$

Therefore, by Equation (#) we have

$$\begin{aligned}
&\left(1+\frac{(8d-1)(1-qs_+(t+1))}{8dqs_+(t+1)(Ts_+(t+1)-1)}\right)\alpha^*(t+1) \\
&\geq \left(1+\frac{(8d-1)(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t) - \frac{5Tqs_+^2(t)(1-qs_+(t))\Delta C(t)}{2(Tqs_+^2(t)-2qs_+(t)+1)}\alpha^*(t) \\
&\quad - \frac{\frac{5}{4}s_+(1-qs_+)\Delta C(t)(2Ts_+-1)}{qs_+(t)^2(Ts_+-1)^2}\alpha^*(t) \\
&\geq \left(1+\frac{(8d-1)(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t) - \left(\frac{5}{14}+\frac{15}{64}\right)\frac{\eta(1-qs_+)}{q(Ts_+-1)} \\
&\geq \left(1+\frac{(8d-1)(1-qs_+(t))}{8dqs_+(t)(Ts_+(t)-1)}\right)\alpha^*(t) - \frac{4d-1}{4d}\frac{\eta(1-qs_+)}{q(Ts_+-1)} \qquad (d \geq 1.) \\
&\geq \alpha(t+1).
\end{aligned}$$

Therefore, we finish the induction for (IH1).

With the induction hypothesis, we can analyze the upper bound of convergence time. For **Phase I**, we have the lower bound for $\Delta C(t)$ (note that $\alpha < 1$, $\gamma < 0$):

$$\begin{aligned}
\Delta C(t) &\geq \eta(1-\gamma)(1-0.1\sqrt{\frac{q(T-q)\epsilon}{dT}})\frac{d}{(T-1)(Tqs_+^2-2qs_++1)}s_+(1-qs_+)^2 \\
&\geq \frac{\eta d s_+(1-qs_+)^2}{2(T-1)(Tqs_+^2-2qs_++1)}
\end{aligned}$$

And for **Phase II**, we have the lower bound since $\gamma \leq \frac{8d-1}{8d}$:

$$\begin{aligned}
\Delta C(t) &= \eta(1-\gamma)(1+\gamma\frac{1-qs_+}{qs_+(Ts_+-1)})\alpha^*\frac{d}{(T-1)(Tqs_+^2-2qs_++1)}s_+(1-qs_+)^2 \\
&\geq \frac{\eta}{8}\frac{(T-q)s_+^2(1-qs_+)^2}{(T-1)(Tqs_+^2-2qs_++1)^2} \geq \frac{\eta}{8}\frac{(1-qs_+)^2}{q(T-1)(Tqs_+^2-2qs_++1)}
\end{aligned}$$

Then $\Delta C(t) \geq \frac{\eta}{8}\frac{(1-qs_+)^2}{q(T-1)(Tqs_+^2-2qs_++1)}$ for all t. Then we divide the training trajectory into two stages as in Huang et al. (2023): in the first stage, s_+ grows to $1/2q$. In the second one, $C(t)$ grows large enough s.t. $s_+ \geq \frac{1}{q} - \frac{1}{q}\sqrt{\frac{q(T-q)\epsilon}{dT}}$.

For the first stage, it's necessary that $C(t) \geq \log\frac{T-q}{q}$. While since $s_+ < \frac{1}{2q}$, we have

$$\Delta C(t) \geq \frac{\eta \cdot \frac{1}{4}}{8q(T-1)(Tqs_+^2-2qs_++1)} \geq \frac{\eta}{8T^2}$$

It takes at most $O(\frac{T^2\log T}{\eta})$ iteration for $C(t)$ to reach this value.

For the second stage, we need $C(t) \geq \log\left(\frac{1}{q}\sqrt{\frac{dT}{q(T-q)\epsilon}}\right)$. Since $s_+ \leq \frac{1}{q} - \frac{1}{q}\sqrt{\frac{q(T-q)\epsilon}{dT}}$ during this period, we can lower bound the increment:

$$\Delta C(t) \geq \frac{\eta}{8}\frac{(1-qs_+)^2}{q(T-1)(Tqs_+^2-2qs_++1)} \geq \frac{\eta q\epsilon}{8d(T-1)T}.$$

So it takes at most $O(\frac{T^2 d\log\left(\frac{dT}{\epsilon}\right)}{\epsilon\eta})$ iterations.

Finally, we check that if $s_+ \geq \frac{1}{q} - \frac{1}{q}\sqrt{\frac{q(T-q)\epsilon}{dT}}$, then the loss is smaller than ϵ.

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}(t)) &= \frac{1}{2}\mathbb{E}\left[\left\|\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}(t)\boldsymbol{Z}\boldsymbol{S}_y\right\|^2\right] \\
&= \frac{1}{2}\mathbb{E}\left[\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \alpha(t)\boldsymbol{X}\boldsymbol{S}_y\right)^\top\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \alpha(t)\boldsymbol{X}\boldsymbol{S}_y\right)\right] \\
&= \frac{1}{2}\mathbb{E}\left[d\left\|\frac{1}{q}\boldsymbol{Y} - \alpha(t)\boldsymbol{S}_y\right\|^2\right] \\
&= \frac{d}{2(T-q)}\left((T-q)q\left(\alpha(t)s_+ - \frac{1}{q}\right)^2 + \alpha(t)^2(1-qs_+)^2\right)
\end{aligned}$$

While $\alpha(t) \in \left[1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}, \left(1 + \frac{(8d-1)(1-qs_+)}{8dqs_+(Ts_+-1)}\right)\alpha^*(t)\right] := [\alpha_1, \alpha_2]$, the loss value is upper bounded by

$$\max_{j\in\{1,2\}} \frac{d}{2(T-q)}\left((T-q)q\left(\alpha_j s_+ - \frac{1}{q}\right)^2 + \alpha_j^2(1-qs_+)^2\right)$$

For $\alpha_1 = 1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$, we have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}(t)) &= \frac{d}{2(T-q)}\left((T-q)q\left(\alpha(t)s_+ - \frac{1}{q}\right)^2 + \alpha(t)^2(1-qs_+)^2\right) \\
&\leq \frac{d}{2(T-q)}\left(\frac{T-q}{q}(q\alpha(t)s_+ - 1)^2 + (1-qs_+)^2\right) \\
&\leq \frac{d}{2(T-q)}\left(\frac{T-q}{q}\left(qs_+ - 1 + 0.1qs_+\sqrt{\frac{q(T-q)\epsilon}{dT}}\right)^2 + (1-qs_+)^2\right) \\
&\leq \frac{d}{2(T-q)}\left(\frac{T-q}{q}\cdot\left((1-qs_+)^2 + 0.2\sqrt{\frac{q(T-q)\epsilon}{dT}}(1-qs_+) + 0.01\cdot\frac{q(T-q)\epsilon}{dT}\right) + (1-qs_+)^2\right) \\
&\leq \frac{d}{2(T-q)}\left(\frac{T-q}{q}\cdot\left((1-qs_+)^2 + 0.2\sqrt{\frac{q(T-q)\epsilon}{dT}}\cdot\sqrt{\frac{q(T-q)\epsilon}{dT}} + 0.01\cdot\frac{q(T-q)\epsilon}{dT}\right)\right) \\
&\quad + \frac{d}{2(T-q)}(1-qs_+)^2 \\
&\leq \frac{d}{2(T-q)}\frac{T}{q}(1-qs_+)^2 + 0.21\epsilon \leq \epsilon.
\end{aligned}$$

For $\alpha_2 = \left(1 + \frac{(8d-1)(1-qs_+)}{8dqs_+(Ts_+-1)}\right)\alpha^*(t)$, denote $\Delta\alpha = \alpha_2 - \alpha^*$. Then $\Delta\alpha s_+ \leq \frac{2\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^*$. Also, for α^* we have a upper bound (using $\epsilon \leq \frac{dT}{100(T-q)q}$)):

$$\alpha^* = \frac{T-q}{Tqs_+ - 2q + \frac{1}{s_+}} \leq \frac{1}{qs_+} \leq \frac{1}{1-\sqrt{\frac{q(T-q)\epsilon}{dT}}} \leq 1 + \frac{6}{5}\sqrt{\frac{q(T-q)\epsilon}{dT}} \leq \frac{28}{25}.$$

Then the loss can be upper bounded:

$$\mathcal{L}(\boldsymbol{\theta}(t)) = \frac{d}{2(T-q)}\left((T-q)q\left(\alpha_2 s_+ - \frac{1}{q}\right)^2 + \alpha_2^2(1-qs_+)^2\right)$$

$$
\begin{aligned}
&= \frac{d}{2(T-q)}\left((T-q)q\left(\alpha^* s_+ - \frac{1}{q}\right)^2 + \alpha^{*2}(1-qs_+)^2\right) \\
&+ \frac{d}{2(T-q)}\left(-(T-q)\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^*(1-qs_+\alpha^*) + \frac{4q^2(T-q)^2\epsilon}{dT^3}\alpha^{*2}\right) \\
&+ \frac{d}{2(T-q)}\left(\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^{*2}(1-qs_+)^2 + \frac{4q(T-q)\epsilon}{dT^3}\alpha^{*2}(1-qs_+)^2\right) \\
&= \frac{d}{2(T-q)}\left((T-q)q\left(\alpha^* s_+ - \frac{1}{q}\right)^2 + \alpha^{*2}(1-qs_+)^2\right) \\
&+ \frac{d}{2(T-q)}\left(-(T-q)\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T(Tqs_+^2 - 2qs_+ + 1)}\alpha^*(1-qs_+)^2 + \frac{4q^2(T-q)^2\epsilon}{dT^3}\alpha^{*2}\right) \\
&+ \frac{d}{2(T-q)}\left(\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^{*2}(1-qs_+)^2 + \frac{4q(T-q)\epsilon}{dT^3}\alpha^{*2}(1-qs_+)^2\right) \\
&= \frac{dT}{2q(T-q)}(qs_+(t)-1)^2 + \frac{d}{2(T-q)}\left(-\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{Ts_+}\alpha^{*2}(1-qs_+)^2 + \frac{4q^2(T-q)^2\epsilon}{dT^3}\alpha^{*2}\right) \\
&+ \frac{d}{2(T-q)}\left(\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^{*2}(1-qs_+)^2 + \frac{4q(T-q)\epsilon}{dT^3}\alpha^{*2}(1-qs_+)^2\right) \\
&\le \frac{\epsilon}{2} + \frac{d}{2(T-q)}\left(\frac{4q^2(T-q)^2\epsilon}{dT^3}\alpha^{*2} + \frac{4q(T-q)\epsilon}{dT^3}\alpha^{*2}(1-qs_+)^2\right) \le \epsilon.
\end{aligned}
$$

when $T \ge 4q$.

In conclusion, after **Phase I** (at most $O(\frac{T\log\frac{d}{\epsilon}}{\eta})$ iterations) and **Phase II** (takes at most $O(\frac{T^2\log T}{\eta} + \frac{T^2 d\log(\frac{dT}{\epsilon})}{\eta\epsilon})$ iterations), the population loss $\mathcal{L}(\boldsymbol{\theta}(t)) \le \epsilon$ after time $\tilde{O}(\frac{T^2 d}{\eta\epsilon})$. □

E. Proof details in Section 3.2

Though Sanford et al. (2023) demonstrated the success of the representational power of one-layer transformers on the original qSA task, it does not guarantee gradient descent can converge to the constructed solution for $\boldsymbol{W}$. For our simplified task STS_q, the problem remains. Experiments (see Section 4) show that if the positional encoding is fixed during training, even though the trained model can express STS_q when it has the original positional encoding, the performance can be drastically bad after switching to another set of valid positional encoding. This motivates us to consider the population loss with the resampling of positional encoding.

To be specific, we further consider resampling from all possible, yet valid positional encodings to get a transformer with a stochastic positional encoding module. In the construction of C.3, as long as the positional encoding matrix $\boldsymbol{E}$ satisfy the (q,δ)-restricted isometry property (RIP) and subset encoding $\boldsymbol{e}_y = \boldsymbol{E}_y(\boldsymbol{E}_y^\top \boldsymbol{E}_y)^{-1}\mathbf{1}_q$ is used, $\boldsymbol{W} = \alpha \boldsymbol{I}_d$ with infinite large α can approximate STS_q. Therefore, we condition on that the random Rademacher matrix has the (q,δ)-restricted isometry and orthogonality property for some δ, and take the expectation of the model output as a transformer with a stochastic positional encoding. After adding the randomized architecture, the model becomes:

Definition E.1 (Transformer with Stochastic Positional Encoding)**.** Define a reparameterized 1-layer self-attention layer with stochastic positional encoding as the following model with trainable parameter matrix $\boldsymbol{V}, \boldsymbol{W}$ where $\boldsymbol{W} \in \mathbb{R}^{d_e \times d_e}, \boldsymbol{V} \in \mathbb{R}^{d\times d}$:

$$
f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y) = \mathbb{E}_{\boldsymbol{E}}\left[\boldsymbol{V}\boldsymbol{Z}\,\mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{\text{query}})\middle|\boldsymbol{E}\text{ satisfies }(q,\delta)\text{-RIP}\right]
$$

Recall the definition of $\boldsymbol{Z}$ and $\boldsymbol{z}_{\text{query}}$:

$$\boldsymbol{Z} := \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{E} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{T-1} & \boldsymbol{x}_T \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{T-1} & \boldsymbol{e}_T \end{bmatrix} \in \mathbb{R}^{(d+T)\times T}, \boldsymbol{z}_{\text{query}} = \begin{bmatrix} \boldsymbol{x}_{\text{query}} \\ \boldsymbol{e}_y \end{bmatrix} \tag{11}$$

For simplicity, we denote the conditional expectation of a random variable $\boldsymbol{x}$ as:

$$\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{x}] = \mathbb{E}_{\boldsymbol{E}}\left[\boldsymbol{x} \middle| \boldsymbol{E} \text{ satisfies } (q,\delta)\text{-RIP}\right]$$

The training objective becomes:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y}\left[\|\operatorname{STS}_q(\boldsymbol{X},y) - f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X},y)\|_2^2\right]. \tag{12}$$

E.1. Notations

In this section, we introduce the notations used in the proof of Theorem E.6 for simplicity. One can refer to this section once some undefined notations are found.

We apply the same notation $\boldsymbol{S}_y$ for simplicity: denote the attention score

$$\boldsymbol{S}_y^{(t)} := \operatorname{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}(t)\boldsymbol{z}_{\text{query}}) \in \mathbb{R}^T$$

for certain q-sparse set $y \in \binom{[T]}{q}$. We denote the i-th entry for the attention score as

$$\boldsymbol{S}_y^{(t)}(i) := \operatorname{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}(t)\boldsymbol{z}_{\text{query}})_i \in \mathbb{R}$$

For clarity, we ignore the timestamp and overload the notation as $\boldsymbol{S}_y$ for all time t during gradient calculation if all variables that occur in the expression are at time t. For GD updates, we will specify the difference between $\boldsymbol{S}_y^{(t)}$ and $\boldsymbol{S}_y^{(t+1)}$.

For $\boldsymbol{W}(t) = C(t)\begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e} \\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e} \end{bmatrix}$, we know before softmax layer the pre-attention score $C(t)$ is the same for any $i,j \in y$ since $\boldsymbol{e}_i^\top \boldsymbol{e}_y = \boldsymbol{e}_j^\top \boldsymbol{e}_y = 1$. That means $\boldsymbol{S}_y(i) = \boldsymbol{S}_y(j)$. During the proof, we denote $s_+ = \boldsymbol{S}_y(i)$ with $i \in y$ when $\boldsymbol{W}(t) = C(t)\begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e} \\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e} \end{bmatrix}$ for some scalar (which is always true in the proof of the two theorems).

After introducing the stochastic architecture, we have $\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]$ in our dynamics. For simplicity, we define $\overline{\boldsymbol{S}_y} = \mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]$. By Lemma E.14, we know for $\boldsymbol{W}(t) = C(t)\begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e} \\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e} \end{bmatrix}$ (which is always true in the two theorems), there exists $\overline{s_+}(t), \overline{s_-}(t)$ s.t.

$$\overline{\boldsymbol{S}_y} = \overline{s_+}\boldsymbol{Y} + \overline{s_-}(\boldsymbol{1}_T - \boldsymbol{Y}).$$

And here $\overline{s_+} := \mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y(i)], i \in y, \overline{s_-} := \mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y(i)], i \notin y$.

In the following subsections, we will separate the gradient matrices into blocks. For matrices in the same shape of $\boldsymbol{W} \in \mathbb{R}^{(d+d_e)\times(d+d_e)}$, we denote $\boldsymbol{W}_{:,\le d} \in \mathbb{R}^{(d+d_e)\times d}$ as the submatrix formed by the first d columns. Then, we define $\boldsymbol{W}_{:,>d}$ as the submatrix formed by the last d_e columns. Similarly, we denote $\boldsymbol{W}_{\le d,\le d}, \boldsymbol{W}_{>d,\le d}, \boldsymbol{W}_{\le d,>d}, \boldsymbol{W}_{>d,>d}$ as the four different block submatrices, respectively. Similar notations also apply in the case of $\boldsymbol{V}$.

E.2. GD dynamics for stochastic PE

Before introducing the main theorem in this section, we first provide the expression of the gradient dynamics. Recall that the input matrix is in the following form

$$[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}] := \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{T-1} & \boldsymbol{x}_T & \boldsymbol{x}_{\text{query}} \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{T-1} & \boldsymbol{e}_T & \boldsymbol{e}_y \end{bmatrix} \in \mathbb{R}^{(d+d_e)\times(T+1)}. \tag{13}$$

where we separate the input tokens $\boldsymbol{Z}$ and the query token $\boldsymbol{z}_{\text{query}}$:

$$\boldsymbol{Z} := \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{E} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{T-1} & \boldsymbol{x}_T \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{T-1} & \boldsymbol{e}_T \end{bmatrix} \in \mathbb{R}^{(d+d_e)\times T}, \boldsymbol{z}_{\text{query}} = \begin{bmatrix} \boldsymbol{x}_{\text{query}} \\ \boldsymbol{e}_y \end{bmatrix} \tag{14}$$

With the stochastic positional encoding introduced, we have the following lemma that shows the dynamics of $\boldsymbol{W}, \boldsymbol{V}$:

Lemma E.2. *Denote $\boldsymbol{S}_y := \mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{query}) \in \mathbb{R}^T$ for certain q-sparse set y. Also, we define the q-hot vector $\boldsymbol{Y} = (\mathbb{1}\{1 \in y\}, \mathbb{1}\{2 \in y\}, ..., \mathbb{1}\{T \in y\})$ for the subset $y \in \binom{[T]}{q}$. The gradient dynamics of $\boldsymbol{W}$ with input $\boldsymbol{X}$ is:*

$$\nabla_{\boldsymbol{W}} \mathcal{L} = -\mathbb{E}_{\boldsymbol{X},y} \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(\boldsymbol{Z}(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y \boldsymbol{S}_y^\top)\boldsymbol{Z}^\top \boldsymbol{V}^\top (\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])\boldsymbol{z}_{query}^\top \right)$$

$$\nabla_{\boldsymbol{V}} \mathcal{L} = -\mathbb{E}_{\boldsymbol{X},y} \left(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y] \right) (\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])^\top \right)$$

Proof. The loss function is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y}\left[\| \mathrm{STS}_q(\boldsymbol{X}, y) - f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y) \|_2^2 \right].$$

Take matrix differentials and we have

$$\begin{aligned} \mathrm{d}\mathcal{L} =& \mathbb{E}_{\boldsymbol{X},y}\left[(f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y) - \mathrm{STS}_q(\boldsymbol{X}, y))^\top \boldsymbol{V}\boldsymbol{Z}\mathrm{d}(\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{\text{query}})\right] \right] \\ &+ \mathbb{E}_{\boldsymbol{X},y}\left[(f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y) - \mathrm{STS}_q(\boldsymbol{X}, y))^\top \mathrm{d}\boldsymbol{V}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(\boldsymbol{Z}\boldsymbol{S}_y) \right] \end{aligned}$$

To the softmax function, we have $\mathrm{d}\,\mathrm{softmax}(\boldsymbol{v}) = (\mathrm{diag}(\boldsymbol{v}) - \boldsymbol{v}\boldsymbol{v}^\top)\mathrm{d}\boldsymbol{v}$. Therefore we have

$$\begin{aligned} \mathrm{d}\mathcal{L} =& \mathbb{E}_{\boldsymbol{X},y}\left[(f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y) - \mathrm{STS}_q(\boldsymbol{X}, y))^\top \boldsymbol{V}\boldsymbol{Z}\mathrm{d}(\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{\text{query}})\right] \right] \\ &+ \mathbb{E}_{\boldsymbol{X},y}\left[(f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y) - \mathrm{STS}_q(\boldsymbol{X}, y))^\top \mathrm{d}\boldsymbol{V}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(\boldsymbol{Z}\boldsymbol{S}_y) \right] \\ =& -\mathbb{E}_{\boldsymbol{X},y}\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y] \right)^\top \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\,\mathrm{d}(\mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{\text{query}})) \\ &- \mathbb{E}_{\boldsymbol{X},y}\left[\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y] \right)^\top \mathrm{d}\boldsymbol{V}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(\boldsymbol{Z}\boldsymbol{S}_y) \right] \\ =& -\mathbb{E}_{\boldsymbol{X},y}(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y] \right)^\top \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top)\boldsymbol{Z}^\top \mathrm{d}\boldsymbol{W}\boldsymbol{z}_{\text{query}} \right] \\ &- \mathbb{E}_{\boldsymbol{X},y}\left[\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y] \right)^\top \mathrm{d}\boldsymbol{V}(\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]) \right] \end{aligned}$$

Since $\frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t} = -\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}$, we have

$$\nabla_{\boldsymbol{W}} \mathcal{L} = -\mathbb{E}_{\boldsymbol{X},y} \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(\boldsymbol{Z}(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y \boldsymbol{S}_y^\top)\boldsymbol{Z}^\top \boldsymbol{V}^\top (\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])\boldsymbol{z}_{\text{query}}^\top \right)$$

$$\nabla_{\boldsymbol{V}} \mathcal{L} = -\mathbb{E}_{\boldsymbol{X},y} \left(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y] \right) (\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])^\top \right)$$

□

Along the gradient trajectory, if the value matrix $\boldsymbol{V}$ can be aligned with the ground-truth $\begin{bmatrix} \boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e} \end{bmatrix}$ and $\boldsymbol{W} = C \cdot \begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e} \\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e} \end{bmatrix}$, we can have the following nice form for the loss function.

Lemma E.3. *Denote $\boldsymbol{S}_y := \mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{query}) \in \mathbb{R}^T$ for certain q-sparse set y. Also, we define the q-hot vector $\boldsymbol{Y} = (\mathbb{1}\{1 \in y\}, \mathbb{1}\{2 \in y\}, ..., \mathbb{1}\{T \in y\})^\top$ for the subset $y \in \binom{[T]}{q}$. If $\boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$ and $\boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$, the loss function can be represented as the following form:*

$$\mathcal{L}(\boldsymbol{\theta}(t)) = \frac{d}{2}\,\mathbb{E}_y\left[\left\|\frac{1}{q}\boldsymbol{Y} - \alpha(t)\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right\|^2\right]$$

Proof. We have the loss function

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\,\mathbb{E}_{\boldsymbol{X}}\left[\|\,\mathrm{STS}_q(\boldsymbol{X}, y) - f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X}, y)\|_2^2\right]$$

and the architecture $f_{\boldsymbol{\theta}} = \mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{V}\boldsymbol{Z}\,\mathrm{softmax}(\boldsymbol{Z}^\top\boldsymbol{W}\boldsymbol{z}_{\text{query}})]$.

Then we have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}(t)) &= \frac{1}{2}\,\mathbb{E}_{\boldsymbol{X},y}\left[\left\|\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}(t)\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right\|^2\right]\\
&= \frac{1}{2}\,\mathbb{E}_{\boldsymbol{X},y}\left[\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \alpha(t)\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)^\top\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \alpha(t)\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)\right]\\
&= \frac{d}{2}\,\mathbb{E}_y\left[\left\|\frac{1}{q}\boldsymbol{Y} - \alpha(t)\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right\|^2\right]
\end{aligned}$$

The last identity is due to that the expectation of the covariance matrix $\boldsymbol{X}^\top\boldsymbol{X}$ is $d\boldsymbol{I}_T$, and $\boldsymbol{S}_y$ is independent of $\boldsymbol{X}$. The independence is because $\boldsymbol{S}_y = \mathrm{softmax}(\boldsymbol{Z}^\top\boldsymbol{W}\boldsymbol{z}_{\text{query}}) = \mathrm{softmax}(C(t)\boldsymbol{E}^\top\boldsymbol{e}_y)$ according to the condition that $\boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$. □

Then, we will prove that during joint training dynamics, $\boldsymbol{V}$ can always evolve along the direction of $\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$.

Lemma E.4. *If $\boldsymbol{V}(t) = \alpha(t)\boldsymbol{I}_d$, $\boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$ for some scalar $\alpha(t), C(t)$, then the gradient for $\boldsymbol{V}(t)$ is*

$$\nabla_{\boldsymbol{V}}\mathcal{L}(t) = \left(-\,\mathbb{E}_y\left[\sum_{i\in y}\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y(i)]/q\right] + \alpha\,\mathbb{E}_y\left[\left\|\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right\|^2\right]\right)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$$

Proof. Consider the gradient of $\boldsymbol{V}$:

$$\nabla_{\boldsymbol{V}}\mathcal{L} = -\,\mathbb{E}_{\boldsymbol{X},y}\left(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)(\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])^\top\right)$$

Since $\boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$, we know $\boldsymbol{S}_y$ is independent of the randomness of $\boldsymbol{X}$.

We first consider the token block in $\boldsymbol{V}$, which is the first d columns.

$$\begin{aligned}
\nabla_{\boldsymbol{V}}\mathcal{L}_{:,\le d} &= -\,\mathbb{E}_{\boldsymbol{X},y}\left(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)(\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])_{\le d}^\top\right)\\
&= -\,\mathbb{E}\left[\frac{1}{q}\boldsymbol{X}\boldsymbol{Y}\left(\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)^\top\right] + \mathbb{E}_{\boldsymbol{X},y}\left(\boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y](\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])^\top\right)
\end{aligned}$$

We consider the two matrices separately:

$$\mathbb{E}\left[\frac{1}{q}\boldsymbol{X}\boldsymbol{Y}\left(\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)^\top\right] \text{ and } \mathbb{E}_{\boldsymbol{X},y}\left(\boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y](\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])^\top\right)$$

For the first matrix, consider the (n, m)-entry of the matrix:

$$\begin{aligned}\boldsymbol{e}_n^\top \mathbb{E}\left[\frac{1}{q}\boldsymbol{X}\boldsymbol{Y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]^\top \boldsymbol{X}^\top\right]\boldsymbol{e}_m &= \frac{1}{q}\,\mathbb{E}\left[\boldsymbol{X}_{n:}\boldsymbol{Y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]^\top \boldsymbol{X}_{m:}^\top\right] \\ &= \frac{1}{q}\,\mathbb{E}\left[\sum_{i\in y} x_{ni}\sum_{i=1}^{T}\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y(i)]x_{mi}\right]\end{aligned}$$

When $m \neq n$, the expectation should be 0 since x_{ni} is sampled from standard Gaussian distribution. When $m = n$, then the expectation is $\mathbb{E}_y\left[\sum_{i\in y}\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y(i)]/q\right]$. Therefore, we have the whole matrix

$$\mathbb{E}\left[\frac{1}{q}\boldsymbol{X}\boldsymbol{Y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]^\top \boldsymbol{X}^\top\right] = \mathbb{E}_y\left[\sum_{i\in y}\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y(i)]/q\right]\boldsymbol{I}_d$$

$$\begin{aligned}\boldsymbol{e}_n^\top \mathbb{E}\left[\boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y](\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])^\top\right]\boldsymbol{e}_m &= \alpha\,\mathbb{E}\left[\boldsymbol{e}_n^\top\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]^\top\boldsymbol{X}^\top\boldsymbol{e}_m\right] \\ &= \alpha\,\mathbb{E}\left[\boldsymbol{X}_{n:}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]^\top\boldsymbol{X}_{m:}^\top\right] \\ &= \alpha\delta_{nm}\,\mathbb{E}\left[\left(\sum_{i=1}^{T}\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y(i)]x_{mi}\right)^2\right] \\ &= \alpha\delta_{nm}\,\mathbb{E}_y\left[\left\|\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right\|^2\right]\end{aligned}$$

Thus the matrix should be in the following form:

$$\mathbb{E}\left[\boldsymbol{V}\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y](\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])^\top\right] = \alpha\,\mathbb{E}_y\left[\left\|\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right\|^2\right]\boldsymbol{I}_d$$

So we get the token block of the gradient

$$\begin{aligned}\nabla_{\boldsymbol{V}}\mathcal{L}_{:,\leq d} &= -\,\mathbb{E}\left[\frac{1}{q}\boldsymbol{X}\boldsymbol{Y}\left(\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)^\top\right] + \mathbb{E}_{\boldsymbol{X},y}\left(\boldsymbol{V}\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y](\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])^\top\right) \\ &= \left(-\,\mathbb{E}_y\left[\sum_{i\in y}\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y(i)]/q\right] + \alpha\,\mathbb{E}_y\left[\left\|\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right\|^2\right]\right)\boldsymbol{I}_d\end{aligned}$$

Then, we consider the position block of the gradient:

$$\begin{aligned}\nabla_{\boldsymbol{V}}\mathcal{L}_{:,d+1:d+d_e} &= -\,\mathbb{E}_{\boldsymbol{X},y}\left(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)(\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])_{d+1:d+d_e}^\top\right) \\ &= -\,\mathbb{E}_{\boldsymbol{X},y}\left(\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y} - \alpha\boldsymbol{X}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)(\boldsymbol{E}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y])^\top\right) = \boldsymbol{0}_{d\times d_e}\end{aligned}$$

Since $\boldsymbol{X} \sim \mathcal{N}(0, \boldsymbol{I}_d)$ and $\boldsymbol{S}_y$ is independent of $\boldsymbol{X}$. Combine the two parts, and the proof is completed. □

Finally, along the gradient descent trajectory, if $\boldsymbol{V} = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$ and $\boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$ for all t, we can further simplify the gradient expression.

Lemma E.5. *If $\boldsymbol{V} = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$ and $\boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$ for all t along the training trajectory and $\boldsymbol{E}$ is stochastic positional encoding, the position-position block matrix of the gradient of $\boldsymbol{W}$, i.e. $\nabla_{\boldsymbol{W}}\mathcal{L}_{>d,>d}$, with input tokens $\boldsymbol{X}$ is:*

$$\nabla_{\boldsymbol{W}}\mathcal{L}_{>d,>d} = -d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\boldsymbol{E}(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top)\left(\frac{1}{q}\boldsymbol{Y} - \alpha(t)\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)\boldsymbol{e}_y^\top\right)$$

Proof. By Lemma E.2, we have the gradient dynamics of $\boldsymbol{W}$ with input token matrix $\boldsymbol{X}$ is:

$$\begin{aligned}
\nabla_{\boldsymbol{W}}\mathcal{L}_{>d,>d} &= -\mathbb{E}_{\boldsymbol{X},y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\boldsymbol{E}(\mathrm{diag}(\boldsymbol{S}_y)-\boldsymbol{S}_y\boldsymbol{S}_y^\top)\boldsymbol{Z}^\top\boldsymbol{V}^\top\left(\frac{1}{q}\boldsymbol{X}\boldsymbol{Y}-\boldsymbol{V}\boldsymbol{Z}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)\boldsymbol{e}_y^\top\right)\\
&= -\alpha(t)\,\mathbb{E}_{\boldsymbol{X},y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\boldsymbol{E}(\mathrm{diag}(\boldsymbol{S}_y)-\boldsymbol{S}_y\boldsymbol{S}_y^\top)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y}-\alpha(t)\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)\boldsymbol{e}_y^\top\right)\\
&\qquad\qquad\text{(Using } \boldsymbol{V}=\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}\text{)}\\
&= -\alpha(t)\,\mathbb{E}_{y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\boldsymbol{E}(\mathrm{diag}(\boldsymbol{S}_y)-\boldsymbol{S}_y\boldsymbol{S}_y^\top)\,\mathbb{E}_{x_{ij}\sim\mathcal{N}(0,1)}[\boldsymbol{X}^\top\boldsymbol{X}]\left(\frac{1}{q}\boldsymbol{Y}-\alpha(t)\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)\boldsymbol{e}_y^\top\right)\\
&\qquad\qquad\text{(Using } \boldsymbol{W}=\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}\Rightarrow \boldsymbol{S}_y \text{ is independent of } \boldsymbol{X}\text{)}\\
&= -d\alpha(t)\,\mathbb{E}_{y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\boldsymbol{E}(\mathrm{diag}(\boldsymbol{S}_y)-\boldsymbol{S}_y\boldsymbol{S}_y^\top)\left(\frac{1}{q}\boldsymbol{Y}-\alpha(t)\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)\boldsymbol{e}_y^\top\right)\qquad(\mathbb{E}\left[\boldsymbol{X}^\top\boldsymbol{X}\right]=d\boldsymbol{I})
\end{aligned}$$

□

In this section, we consider the gradient descent dynamics. The update rules are in the following form

$$\begin{aligned}
\boldsymbol{W}(t+1) &= \boldsymbol{W}(t)-\eta\nabla_{\boldsymbol{W}}\mathcal{L}(t)\\
\boldsymbol{V}(t+1) &= \boldsymbol{V}(t)-\eta\nabla_{\boldsymbol{V}}\mathcal{L}(t)
\end{aligned}$$

E.3. Joint Training

Now, we analyze the dynamics of training the value matrix $\boldsymbol{V}$ and attention matrix $\boldsymbol{W}$ simultaneously with the same learning rate η with the stochastic positional encoding. The following theorem characterizes the convergence of GD when training both layers simultaneously with stochastic positional encoding.

Theorem E.6 (Joint training with stochastic positional encoding)**.** *Suppose* $2\le q<T/4, q,\delta=\Theta(1), d_e=\Theta(q\log T/\delta^2), \delta<1/10$. *For any* $\epsilon\in(0,\frac{dT}{100(T-q)q}), \eta\le\frac{d_e^2}{40d^2T}$, *if we apply gradient descent on the population loss in Equation* (4) *with zero initialization* $\boldsymbol{W}(0)=\boldsymbol{0}_{d_e\times d_e}, \boldsymbol{V}(0)=\boldsymbol{0}_{d\times d}$, *then after time* $t\ge\tilde{O}(\frac{T^{\frac{2-2\delta}{1-3\delta}}}{\eta}+\frac{T^2d}{\eta\epsilon})$, *we have*

$$\mathcal{L}(\boldsymbol{\theta}(t))=\frac{1}{2}\,\mathbb{E}_{\boldsymbol{X},y}\left[\|\,\mathrm{STS}_q(\boldsymbol{X},y)-f_{\boldsymbol{\theta}}^{(s)}(\boldsymbol{X},y)\|_2^2\right]\le\epsilon.$$

The proof idea is similar to the joint training scenario with one-hot PE. We can still simplify the dynamics of $\boldsymbol{W}$ and $\boldsymbol{V}$ using symmetry, proving convergence along the global optimal direction. Then the two variable dynamics are considered inductively so that $\boldsymbol{W}$ and $\boldsymbol{V}$ can converge to the global minimum.

We have the following lemma that characterizes the evolution speed along the converging direction of $\boldsymbol{W}$ and $\boldsymbol{V}$: they always pointing to the ground-truth direction that $\boldsymbol{W}(t)=C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$, $\boldsymbol{V}(t)=\alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$.

Lemma E.7 (Induction Hypothesis for Stochastic PE, Joint Training)**.** *Suppose all conditions in Theorem E.6 holds, then along the gradient descent trajectory, for all* $t\ge0$, *there exist some time-dependent scalars* $C(t),\alpha(t)$ *s.t.* $\boldsymbol{W}(t)=C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$,$\boldsymbol{V}(t)=\alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$, *and* $C(t),\alpha(t)$ *satisfies:*

$$\begin{aligned}
C(t+1) &\ge C(t)+\eta\frac{1-3\delta}{1-2\delta}\cdot\frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)\left(1-\frac{q\alpha(t)(T\overline{s_+}-1)}{T-q}\right)s_+\\
C(t+1) &\le C(t)+\eta\frac{1-\delta}{1-2\delta}\cdot\frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)\left(1-\frac{q\alpha(t)(T\overline{s_+}-1)}{T-q}\right)s_+\\
\alpha(t+1) &= \alpha(t)+\eta\overline{s_+}\left(1-q\alpha\overline{s_+}+\frac{\alpha(1-q\overline{s_+})^2}{(T-q)\overline{s_+}}\right)
\end{aligned}$$

Proof. We prove $\boldsymbol{W}(t) = C(t)\begin{bmatrix}\mathbf{0}_{d\times d} & \mathbf{0}_{d\times d_e}\\ \mathbf{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}, \boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \mathbf{0}_{d\times d_e}\end{bmatrix}$. by induction. First, check the initialization $\boldsymbol{W}(t) = 0 = 0\cdot\begin{bmatrix}\mathbf{0}_{d\times d} & \mathbf{0}_{d\times d_e}\\ \mathbf{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}, \boldsymbol{V}(0) = 0 = 0\cdot\begin{bmatrix}\boldsymbol{I}_d & \mathbf{0}_{d\times d_e}\end{bmatrix}$. Here the scalar $C(0) = \alpha(0) = 0$. For GD dynamics, by Lemma E.4 and Lemma E.5 we have $C(1) = C(0), \alpha(1) = \alpha(0) + \eta/T$. Therefore, the induction hypothesis holds for $t = 0$.

Then we prove this argument inductively: if Lemma E.7 holds for iteration t, it is enough to prove that for iteration $t+1$, this argument still holds. Then by induction, we can conclude that it holds for all time $t \geq 0$.

Note that since the position-position block (right-bottom block) of $\boldsymbol{W}$ is always in direction of $\begin{bmatrix}\mathbf{0}_{d\times d} & \mathbf{0}_{d\times d_e}\\ \mathbf{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$ and other entries are 0, $\boldsymbol{S}_y = \mathrm{softmax}(\boldsymbol{Z}^\top\boldsymbol{W}\boldsymbol{z}_{\text{query}}) = \mathrm{softmax}(C(t)\,\mathbb{E}^\top\,\boldsymbol{e}_y)$ when y is given. Therefore, the softmax vector is always independent of $\boldsymbol{X}$ for iterations $\leq t$.

Now we suppose these two properties hold for iteration t. We first consider **the token-token, token-position and position token** submatrices of $\boldsymbol{W}(t)$. We prove that those gradient blocks should always be $\mathbf{0}$.

For the first d rows, we have (here within $\boldsymbol{z}_{\text{query}}$, $\boldsymbol{x}_{\text{query}}$ can be any fixed token vector):

$$\begin{aligned}
\nabla_{\boldsymbol{W}}\mathcal{L}(t)_{\leq d,:} &= \alpha\,\mathbb{E}_{\boldsymbol{X},y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{X}\big(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\big)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\boldsymbol{z}_{\text{query}}{}^\top\right]\\
&= \frac{1}{2}\alpha\,\mathbb{E}_{\boldsymbol{X},y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{X}\big(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\big)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\boldsymbol{z}_{\text{query}}{}^\top\right]\\
&+ \frac{1}{2}\alpha\,\mathbb{E}_{\boldsymbol{X},y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[-\boldsymbol{X}\big(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\big)(-\boldsymbol{X}^\top)(-\boldsymbol{X})\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\boldsymbol{z}_{\text{query}}{}^\top\right]\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(By symmetry and independence between } \boldsymbol{X} \text{ and } \boldsymbol{S}_y)\\
&= \mathbf{0}_{d\times(d+d_e)}.
\end{aligned}$$

Then we consider the position-token block $\nabla_{\boldsymbol{W}}\mathcal{L}(t)_{>d,\leq d}$:

$$\begin{aligned}
\nabla_{\boldsymbol{W}}\mathcal{L}(t)_{>d,\leq d} &= \alpha\,\mathbb{E}_{\boldsymbol{X},y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{E}\big(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\big)\boldsymbol{X}^\top\boldsymbol{X}\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\boldsymbol{x}_{\text{query}}{}^\top\right]\\
&= d\alpha\,\mathbb{E}_{\boldsymbol{X},y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{E}\big(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\big)\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\boldsymbol{x}_{\text{query}}{}^\top\right] \quad \text{(Independence between } \boldsymbol{X} \text{ and } \boldsymbol{S}_y)\\
&= d\alpha\,\mathbb{E}_{\boldsymbol{X},y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{E}\big(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\big)\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\boldsymbol{x}_{\text{query}}{}^\top\right]\\
&+ d\alpha\,\mathbb{E}_{\boldsymbol{X},y}\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[-\boldsymbol{E}\big(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top\big)\left(\frac{1}{q}\boldsymbol{Y} - \alpha\boldsymbol{S}_y\right)\boldsymbol{x}_{\text{query}}{}^\top\right] = \mathbf{0}_{d_e\times d}.\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(For } \boldsymbol{E} \text{ and } -\boldsymbol{E}, \boldsymbol{S}_y \text{ are the same.)}
\end{aligned}$$

Finally, we consider the update for **the position-position submatrix** $\boldsymbol{W}(t)_{>d,>d}$ in its (k,j)-entry $\boldsymbol{W}_{kj}(t)$, $k,j \in [d_e]$. With the update rule by Lemma E.5,

$$\Delta\boldsymbol{W}(t)_{>d,>d} = \eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\boldsymbol{E}(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top)\left(\frac{1}{q}\boldsymbol{Y} - \alpha(t)\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)\boldsymbol{e}_y^\top\right)$$

Since $\boldsymbol{W}(t)_{>d,>d} = C(t)\boldsymbol{I}_{d_e}$, we only need to prove that the gradient term is along the direction of $\boldsymbol{I}_{d_e}$. Recall the notation $\overline{\boldsymbol{S}_y} = \mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]$. To prove this statement, we first expand the gradient term:

$$\begin{aligned}
\Delta\boldsymbol{W}(t)_{>d,>d} &= \eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\boldsymbol{E}(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top)\left(\frac{1}{q}\boldsymbol{Y} - \alpha(t)\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right)\boldsymbol{e}_y^\top\right)\\
&= \eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\sum_{i=1}^{T}\boldsymbol{e}_i\left[(\mathrm{diag}(\boldsymbol{S}_y) - \boldsymbol{S}_y\boldsymbol{S}_y^\top)\left(\frac{1}{q}\boldsymbol{Y} - \alpha(t)\overline{\boldsymbol{S}_y}\right)\right]_i\boldsymbol{e}_y^\top\right)
\end{aligned}$$

$$= \eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\sum_{i\in y}\boldsymbol{e}_i\left[(\operatorname{diag}(\boldsymbol{S}_y)-\boldsymbol{S}_y\boldsymbol{S}_y^\top)\left(\frac{1}{q}\boldsymbol{Y}-\alpha(t)\overline{\boldsymbol{S}_y}\right)\right]_i\boldsymbol{e}_y^\top\right)$$

$$+\eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\sum_{i\notin y}\boldsymbol{e}_i\left[(\operatorname{diag}(\boldsymbol{S}_y)-\boldsymbol{S}_y\boldsymbol{S}_y^\top)\left(\frac{1}{q}\boldsymbol{Y}-\alpha(t)\overline{\boldsymbol{S}_y}\right)\right]_i\boldsymbol{e}_y^\top\right)$$

$$=\eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\sum_{i\in y}\boldsymbol{e}_i\left[\boldsymbol{S}_y(i)\left(\frac{1}{q}-\alpha(t)\overline{\boldsymbol{S}_y}(i)\right)+\boldsymbol{S}_y(i)\left[\alpha(t)\boldsymbol{S}_y^\top\overline{\boldsymbol{S}_y}-\sum_{i\in y}\boldsymbol{S}_y(i)/q\right]\right]\boldsymbol{e}_y^\top\right) \tag{Term 1'}$$

$$+\eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\sum_{i\notin y}\boldsymbol{e}_i\left[-\alpha(t)\boldsymbol{S}_y(i)\overline{\boldsymbol{S}_y}(i)+\alpha(t)\boldsymbol{S}_y(i)(\boldsymbol{S}_y^\top\overline{\boldsymbol{S}_y})-\frac{1}{q}\boldsymbol{S}_y(i)\sum_{j\in y}\boldsymbol{S}_y(j)\right]\boldsymbol{e}_y^\top\right) \tag{Term 2'}$$

By Lemma E.14, we know for $\boldsymbol{W}(t)=C(t)\boldsymbol{I}_{d_e}$, there exists $\overline{s_+}(t),\overline{s_-}(t)$ s.t.

$$\overline{\boldsymbol{S}_y}=\overline{s_+}\boldsymbol{Y}+\overline{s_-}(\boldsymbol{1}_T-\boldsymbol{Y}).$$

By Lemma E.12, any off-diagonal entry of both terms in the gradient should be 0 according to symmetry. Now we only need to consider the diagonal entries. By Lemma E.13, all the diagonal entries of each term in the gradient have the same value. Therefore, the gradient of the submatrix is aligned with $\boldsymbol{I}_{d_e}$. Combine the four blocks, the whole gradient of $\boldsymbol{W}$ is aligned with $\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$.

For $\boldsymbol{V}(t)$, by Lemma E.4 we know its update is always in the $\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$ direction. Therefore,

$$\boldsymbol{V}(t+1)=\boldsymbol{V}(t)+\left(\mathbb{E}_y\left[\sum_{i\in y}\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y(i)]/q\right]-\alpha\,\mathbb{E}_y\left[\left\|\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right\|^2\right]\right)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$$

aligns with $\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$. By induction, we complete the proof for the direction property.

After proving the direction property, we calculate $C(t)$'s dynamics by considering two terms separately. Now we only consider the position-position blocks' dynamics. Since the symmetry among all the diagonal entries, the trace of each term is considered and each diagonal entry will be $1/d_e$ of the conditional expectation of the trace.

Term 1: Terms with $i\in y$.

$$\eta d\alpha(t)\operatorname{tr}\left(\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\sum_{i\in y}\boldsymbol{e}_i\left[\boldsymbol{S}_y(i)\left(\frac{1}{q}-\alpha(t)\overline{\boldsymbol{S}_y}(i)\right)+\boldsymbol{S}_y(i)\left[\alpha(t)\boldsymbol{S}_y^\top\overline{\boldsymbol{S}_y}-\sum_{i\in y}\boldsymbol{S}_y(i)/q\right]\right]\boldsymbol{e}_y^\top\right)\right)$$

$$=\eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\sum_{i\in y}\left[\boldsymbol{S}_y(i)\left(\frac{1}{q}-\alpha(t)\overline{\boldsymbol{S}_y}(i)\right)+\boldsymbol{S}_y(i)\left[\alpha(t)\boldsymbol{S}_y^\top\overline{\boldsymbol{S}_y}-\sum_{i\in y}\boldsymbol{S}_y(i)/q\right]\right]\operatorname{tr}\left(\boldsymbol{e}_i\boldsymbol{e}_y^\top\right)\right)$$

(Linearity of trace and expectation)

$$=\eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\sum_{i\in y}\left[\boldsymbol{S}_y(i)\left(\frac{1}{q}-\alpha(t)\overline{\boldsymbol{S}_y}(i)\right)+\boldsymbol{S}_y(i)\left[\alpha(t)\boldsymbol{S}_y^\top\overline{\boldsymbol{S}_y}-\sum_{i\in y}\boldsymbol{S}_y(i)/q\right]\right]\boldsymbol{e}_i^\top\boldsymbol{e}_y\right)$$

$$=\eta d\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\sum_{i\in y}\left[\boldsymbol{S}_y(i)\left(\frac{1}{q}-\alpha(t)\overline{\boldsymbol{S}_y}(i)\right)+\boldsymbol{S}_y(i)\left[\alpha(t)\boldsymbol{S}_y^\top\overline{\boldsymbol{S}_y}-\sum_{i\in y}\boldsymbol{S}_y(i)/q\right]\right]\right)\qquad(\boldsymbol{e}_i^\top\boldsymbol{e}_y=1\text{ for }i\in y)$$

$$=\eta dq\alpha(t)\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left(\left[\boldsymbol{S}_y(i)\left(\frac{1}{q}-\alpha(t)\overline{\boldsymbol{S}_y}(i)\right)+\boldsymbol{S}_y(i)\left[\alpha(t)\boldsymbol{S}_y^\top\overline{\boldsymbol{S}_y}-\sum_{i\in y}\boldsymbol{S}_y(i)/q\right]\right]\right)\qquad(\text{Symmetry among }i\in y)$$

Expand $\boldsymbol{S}_y^\top \overline{\boldsymbol{S}_y}$ and we have:

$$\boldsymbol{S}_y^\top \overline{\boldsymbol{S}_y} = \sum_{i\in y} \boldsymbol{S}_y(i)\overline{s_+} + \sum_{i\notin y} \boldsymbol{S}_y(i)\overline{s_-}$$

Plug it back into Term 1:

$$\begin{aligned}
&\eta d q \alpha(t)\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(\left[\boldsymbol{S}_y(i)\left(\frac{1}{q} - \alpha(t)\overline{\boldsymbol{S}_y}(i)\right) + \boldsymbol{S}_y(i)\left[\alpha(t)\boldsymbol{S}_y^\top \overline{\boldsymbol{S}_y} - \sum_{i\in y} \boldsymbol{S}_y(i)/q\right]\right]\right) \\
&= \eta d q \alpha(t)\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(\left[\boldsymbol{S}_y(i)\left(\frac{1}{q} - \alpha(t)\overline{s_+}\right) + \boldsymbol{S}_y(i)\left(\sum_{i\in y} \boldsymbol{S}_y(i)\left(\alpha(t)\overline{s_+} - \frac{1}{q}\right) + \alpha(t)\sum_{i\notin y} \boldsymbol{S}_y(i)\overline{s_-}\right)\right]\right) \\
&= \eta d q\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(\left[\boldsymbol{S}_y(i)(1 - q\boldsymbol{S}_y(i))\left(\frac{1}{q} - \alpha(t)\overline{s_+}\right) + \alpha(t)\boldsymbol{S}_y(i)\left(\sum_{i\notin y} \boldsymbol{S}_y(i)\overline{s_-}\right)\right]\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\boldsymbol{S}_y(i) = s_+ \text{ are equal for } i \in y) \\
&= \eta d q\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(\left[\boldsymbol{S}_y(i)(1 - q\boldsymbol{S}_y(i))\left(\frac{1}{q} - \alpha(t)\overline{s_+}\right) + \alpha(t)\boldsymbol{S}_y(i)\left(\sum_{i\notin y} \boldsymbol{S}_y(i)\frac{(1-q\overline{s_+})}{T-q}\right)\right]\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (q\overline{s_+} + (T-q)s_+ = 1) \\
&= \eta d q\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(\left[s_+(1 - qs_+)\left(\frac{1}{q} - \alpha(t)\overline{s_+}\right) + \alpha(t)s_+(1-qs_+)\frac{(1-q\overline{s_+})}{T-q}\right]\right) \qquad (qs_+ + \textstyle\sum_{j\notin y} \boldsymbol{S}_y(j) = 1) \\
&= \eta d q\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(s_+(1 - qs_+)\left(\frac{1}{q} - \alpha(t)\frac{T\overline{s_+} - 1}{T-q}\right)\right)
\end{aligned}$$

Therefore,

$$\text{Term 1} = \frac{\eta d}{d_e}\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(s_+(1 - qs_+)\left(1 - q\alpha(t)\frac{T\overline{s_+} - 1}{T-q}\right)\right)\boldsymbol{I}_{d_e}$$

Term 2: Terms with $i \notin y$. We use a similar technique from above.

$$\begin{aligned}
&\eta d \alpha(t)\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \left(\sum_{i\notin y} \boldsymbol{e}_i \left[-\alpha(t)\boldsymbol{S}_y(i)\overline{\boldsymbol{S}_y}(i) + \alpha(t)\boldsymbol{S}_y(i)(\boldsymbol{S}_y^\top \overline{\boldsymbol{S}_y}) - \frac{1}{q}\boldsymbol{S}_y(i)\sum_{j\in y} \boldsymbol{S}_y(j)\right] \boldsymbol{e}_y^\top \right) \\
&= \eta d \alpha(t)\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \sum_{i\notin y} \left(-\left[\boldsymbol{S}_y(i)\left(\alpha(t)\overline{s_-} - \alpha(t)\sum_{i\in y} \boldsymbol{S}_y(i)\overline{s_+} - \alpha(t)\sum_{i\notin y} \boldsymbol{S}_y(i)\overline{s_-} + \frac{1}{q}\sum_{j\in y} \boldsymbol{S}_y(j)\right)\right] \boldsymbol{e}_i^\top \boldsymbol{e}_y \right) \\
&= \eta d \alpha(t)\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \sum_{i\notin y} \left(-\left[\boldsymbol{S}_y(i)\left(\alpha(t)\frac{(1-q\overline{s_+})}{T-q} - \alpha(t)qs_+\overline{s_+} - \alpha(t)(1-qs_+)\frac{(1-q\overline{s_+})}{T-q} + s_+\right)\right] \boldsymbol{e}_i^\top \boldsymbol{e}_y \right) \\
&= \eta d \alpha(t)\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \sum_{i\notin y} \left(-\left[\boldsymbol{S}_y(i)s_+\left(1 - \frac{q\alpha(t)(T\overline{s_+} - 1)}{T-q}\right)\right] \boldsymbol{e}_i^\top \boldsymbol{e}_y \right) \\
&\geq -\eta\alpha(t)d \cdot \frac{\delta}{1-2\delta}\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} \sum_{i\notin y} \left[\boldsymbol{S}_y(i)s_+\left(1 - \frac{q\alpha(t)(T\overline{s_+} - 1)}{T-q}\right)\right] \\
&= -\eta d\alpha(t)\frac{\delta}{1-2\delta}\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} (1 - qs_+)\left(1 - \frac{q\alpha(t)(T\overline{s_+} - 1)}{T-q}\right)s_+
\end{aligned}$$

The last line is due to $\sum_{i\notin y} \boldsymbol{S}_y(i) = 1 - \sum_{i\in y} \boldsymbol{S}_y(i) = 1 - qs_+$. Thus we have

$$\text{Term 2} \succeq -\frac{\eta d\alpha(t)\delta}{(1-2\delta)d_e}\, \mathbb{E}_y\, \mathbb{E}_{\boldsymbol{E}}^{(R)} (1 - qs_+)\left(1 - \frac{q\alpha(t)(T\overline{s_+} - 1)}{T-q}\right)s_+\boldsymbol{I}_{d_e}$$

Similarly, we can also have the upper bound for Term 2.

$$\text{Term 2} \preceq \frac{\eta d\alpha(t)\delta}{(1-2\delta)d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)\left(1-\frac{q\alpha(t)(T\overline{s_+}-1)}{T-q}\right)s_+\boldsymbol{I}_{d_e}$$

Combine two terms, we have the lower bound for $C(t)$'s dynamics:

$$C(t+1) \geq C(t) + \eta\frac{1-3\delta}{1-2\delta}\cdot\frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)\left(1-\frac{q\alpha(t)(T\overline{s_+}-1)}{T-q}\right)s_+$$

$$C(t+1) \leq C(t) + \eta\frac{1-\delta}{1-2\delta}\cdot\frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)\left(1-\frac{q\alpha(t)(T\overline{s_+}-1)}{T-q}\right)s_+$$

By induction, we complete the proof. □

Remark. Similar to the one-hot case, after proving the induction lemma about the evolving direction of $\boldsymbol{W}(t)$ and $\boldsymbol{V}(t)$, the optimization problem can be reduced to analyzing the two variable dynamics of $C(t)$ and $\alpha(t)$:

$$C(t+1) \approx C(t) + \frac{\eta d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)\left(1-\frac{q\alpha(t)(T\overline{s_+}-1)}{T-q}\right)s_+$$
$$\alpha(t+1) = \alpha(t) + \eta\overline{s_+}\left(1-q\alpha\overline{s_+}+\frac{\alpha(1-q\overline{s_+})^2}{(T-q)\overline{s_+}}\right)$$

Then the rest of the proof is to analyze the dynamics of $C(t)$ and $\alpha(t)$ and prove that $C(t)\to+\infty$ and $\alpha(t)\to 1$ eventually, and calculate the convergence time by estimating the trajectory of the two variable dynamical systems. One can refer to the main paper for proof ideas.

Proof of Theorem E.6. After Lemma E.7 shows that $\boldsymbol{V}$ and $\boldsymbol{W}$ are always along the ground-truth direction: $\boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}, \boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$, the dynamics of the parameter matrices then can be characterized by two scalar variable $\alpha(t)$ and $C(t)$. Our update rules becomes

$$C(t+1) \geq C(t) + \eta\frac{1-3\delta}{1-2\delta}\cdot\frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)\left(1-\frac{q\alpha(t)(T\overline{s_+}-1)}{T-q}\right)s_+$$
$$C(t+1) \leq C(t) + \eta\frac{1-\delta}{1-2\delta}\cdot\frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)\left(1-\frac{q\alpha(t)(T\overline{s_+}-1)}{T-q}\right)s_+$$
$$\alpha(t+1) = \alpha(t) + \eta\overline{s_+}\left(1-q\alpha\overline{s_+}+\frac{\alpha(1-q\overline{s_+})^2}{(T-q)\overline{s_+}}\right)$$

Along the trajectory, $s_+ \leq \frac{1}{q}$ by its definition: since $\boldsymbol{W}$ is along $\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$, all $i\in y$ has the same softmax probability $\boldsymbol{S}_y^{(t)}(i)$, and thus they cannot exceed the upper bound $1/q$.

Note that when s_+ is fixed, $\alpha(t)$ has a stationary point $\alpha^*(t) = \frac{(T-q)\overline{s_+}}{Tq\overline{s_+}^2-2q\overline{s_+}+1}$. With some calculation, we can rewrite the α dynamics into:

$$\alpha(t+1) = \alpha(t) + \eta\overline{s_+}\left(1-\frac{\alpha(t)}{\alpha^*(t)}\right).$$

Observe that when $\overline{s_+}\in(\frac{1}{T},\frac{1}{q}), \alpha^*(t)\geq 1$, and it achieves maximum at $\overline{s_+} = \frac{1}{\sqrt{Tq}}$.

To characterize this limit above when $t\to\infty$, we need to prove the following two arguments:

1. $C(t)$ is non-decreasing for all $t\geq 0$.

2. $\alpha(t)$ goes through 2 phases:

   Phase I. α monotonically grows to $1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$ at some time t_1.

   Phase II. α stays within an interval whose upper bound is close to α^* after time t_1:

$$\alpha(t) \in \left[1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}, \left(1 + \frac{(8Td - d_e)(1 - q\overline{s_+})}{8Tdq\overline{s_+}(T\overline{s_+} - 1)}\right)\alpha^*(t)\right] \tag{IH2}$$

Phase I. In this phase, we inductively prove that both $\alpha(t), C(t)$ are non-decreasing. For $t = 0$, by zero initialization, $C(0) = C(1) = 0, \alpha(1) = \frac{\eta}{T} > 0$.

Suppose it holds for some $t < t_1$ before α reaches $1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$. Then for $\alpha(t+1)$, we have the update rule:

$$\begin{aligned}
\alpha(t+1) &= \alpha(t) + \eta\overline{s_+}\left(1 - \frac{\alpha(t)}{\alpha^*(t)}\right) \\
&\geq \alpha(t) + \eta\frac{1}{T}(1 - \frac{\alpha(t)}{\alpha^*(t)}) && \text{(Induction Hyp. } \overline{s_+} \geq \tfrac{1}{T}) \\
&\geq \alpha(t) + \eta\frac{1}{T}(1 - \alpha(t)) && (\alpha^* \geq 1) \\
&\geq \alpha(t) + 0.1\eta\sqrt{\frac{q(T-q)\epsilon}{dT^3}}. && (\alpha \leq 1 - 0.1\sqrt{\tfrac{q(T-q)\epsilon}{dT}})
\end{aligned}$$

So α is non-decreasing. Meanwhile, for $C(t+1)$:

$$\begin{aligned}
C(t+1) &\geq C(t) + \eta\frac{1-3\delta}{1-2\delta} \cdot \frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)\left(1 - \frac{q\alpha(t)(T\overline{s_+} - 1)}{T-q}\right)s_+ \\
&\geq C(t) + \frac{7\eta}{8} \cdot \frac{\alpha d}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)\left(1 - \frac{q(T\overline{s_+} - 1)}{T-q}\right)s_+ && (\alpha < 1, (1 - Ts_+) < 0) \\
&= C(t) + \frac{7\eta}{8} \cdot \frac{\alpha d}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)\frac{T - q + q - Tq\overline{s_+}}{T-q}s_+ \\
&= C(t) + \frac{7\eta}{8} \cdot \frac{\alpha dT}{d_e(T-q)}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)(1 - q\overline{s_+})s_+ \geq C(t)
\end{aligned}$$

so they are both non-decreasing. Then we need to upper bound the time t_1 for $\alpha(t)$ to reach $1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$: by the update above we have

$$\begin{aligned}
\alpha(t+1) &= \alpha(t) + \eta\overline{s_+}\left(1 - \frac{\alpha(t)}{\alpha^*(t)}\right) \\
&\geq \alpha(t) + \eta\frac{1}{T}(1 - \frac{\alpha(t)}{\alpha^*(t)}) && \text{(Induction Hyp. } \overline{s_+} \geq \tfrac{1}{T}) \\
&\geq \alpha(t) + \eta\frac{1}{T}(1 - \alpha(t)) && (\alpha^* \geq 1) \\
\Rightarrow 1 - \alpha(t+1) &\leq (1 - \eta/T)(1 - \alpha(t)) \leq ... \leq (1 - \eta/T)^t(1 - \alpha(0)).
\end{aligned}$$

Thus for $\alpha(t+1) \geq 1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$, it takes at most $O(\frac{T\log\frac{d}{\epsilon}}{\eta})$ iterations.

Phase II. In this phase, we first consider the easiest case: $\overline{s_+}(t) < \frac{1}{\sqrt{Tq}}$. Under this condition, $\alpha^*(t+1) > \alpha^*(t)$, and using induction hypothesis

$$\alpha^*(t) - \alpha(t+1) = \alpha^*(t) - \alpha(t) - \frac{\eta\overline{s_+}(t)}{\alpha^*(t)}(\alpha^*(t) - \alpha(t)) = (1 - \frac{\eta\overline{s_+}(t)}{\alpha^*(t)})(\alpha^*(t) - \alpha(t)) > 0,$$

$\alpha(t+1) < \alpha^*(t+1)$ always holds, and the induction hypothesis holds for $t+1$.

Then we only need to consider the case with $\overline{s_+}(t) \geq \frac{1}{\sqrt{Tq}}$. First, we check that within the induction hypothesis range, $C(t+1) \geq C(t)$. We know the following term in the gradient update of $C(t)$ should be greater than 0:

$$\left(1 - \frac{q(T\overline{s_+} - 1)}{T-q}\right) \geq 1 - \left(1 + \frac{(8Td - d_e)(1 - q\overline{s_+})}{8Tdq\overline{s_+}(T\overline{s_+} - 1)}\right)\alpha^*(t)\frac{q}{T-q}(T\overline{s_+} - 1)$$

$$(\alpha \leq \left(1 + \tfrac{(d-1)(1-qs_+)}{dqs_+(Ts_+-1)}\right)\alpha^*(t), (1 - Ts_+) < 0)$$

$$\geq \frac{d_e}{8Td}\frac{1 - q\overline{s_+}}{Tq\overline{s_+}^2 - 2qs_+ + 1} \geq 0. \qquad (\overline{s_+} \leq \tfrac{1}{q}.)$$

Therefore by the update rule:

$$C(t+1) \geq C(t) + \eta\frac{1-3\delta}{1-2\delta} \cdot \frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)\left(1 - \frac{q\alpha(t)(T\overline{s_+} - 1)}{T-q}\right)s_+$$

$$\geq C(t) + \eta\frac{7d\alpha(t)}{8d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)s_+ \cdot \frac{d_e(1 - q\overline{s_+})}{8Td(Tq\overline{s_+}^2 - 2q\overline{s_+} + 1)} \geq C(t)$$

Next, we first divide $\alpha(t)$'s possible range into two parts: $\alpha(t) \leq \left(1 + \frac{(4Td-d_e)(1-q\overline{s_+}(t))}{4dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t)$ and $\alpha(t) \in \left[\left(1 + \frac{(4dT-d_e)(1-q\overline{s_+}(t))}{4dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t), \left(1 + \frac{(8dT-d_e)(1-q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t)\right]$. For the first part, we prove the following statement **(S3)**:

$$\text{If } \alpha(t) \leq \left(1 + \tfrac{(4dT-d_e)(1-q\overline{s_+}(t))}{4dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t), \text{ the next step}$$

$$\alpha(t+1) \leq \left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t+1))}{8dTq\overline{s_+}(t+1)(T\overline{s_+}(t+1) - 1)}\right)\alpha^*(t+1).$$

If **(S3)** is true, then we know once $\alpha(t) \leq \left(1 + \frac{(4dT-d_e)(1-q\overline{s_+}(t))}{4dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t)$, $\alpha(t+1)$ satisfy the induction hypothesis. After proving **(S3)**, the only part left is when

$$\alpha(t) \in \left(\left(1 + \frac{(4dT - d_e)(1 - q\overline{s_+}(t))}{4dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t), \left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t)\right].$$

We prove statement **(S3)** by proving

$$\left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t+1))}{8dTq\overline{s_+}(t+1)(T\overline{s_+}(t+1) - 1)}\right)\alpha^*(t+1) \geq \left(1 + \frac{(4dT - d_e)(1 - q\overline{s_+}(t))}{4dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t) \qquad \textbf{(S4)}$$

When the inequality **(S4)** above is proved, then **(S3)** is proved. This is because: if $\alpha(t) < \alpha^*(t)$, then by update rule we have

$$\alpha^*(t) - \alpha(t+1) = \alpha^*(t) - \alpha(t) - \frac{\eta\overline{s_+}(t)}{\alpha^*(t)}(\alpha^*(t) - \alpha(t)) = (1 - \frac{\eta\overline{s_+}(t)}{\alpha^*(t)})(\alpha^*(t) - \alpha(t)) > 0,$$

$$\alpha(t+1) < \alpha^*(t) \leq \left(1 + \frac{(4dT - d_e)(1 - q\overline{s_+}(t))}{4dqT\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t)$$

$$\leq \left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t+1))}{8dqT\overline{s_+}(t+1)(T\overline{s_+}(t+1) - 1)}\right)\alpha^*(t+1)$$

If $\alpha(t) \geq \alpha^*(t)$, then $\alpha(t+1) \leq \alpha(t) \leq \left(1 + \frac{(4dT-d_e)(1-q\overline{s_+}(t))}{4dqT\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t)$, and therefore smaller than $\left(1 + \frac{(8dT-d_e)(1-q\overline{s_+}(t+1))}{8dqT\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)}\right)\alpha^*(t+1)$.

Now we prove **(S4)** by expanding the $\overline{s_+}(t+1)$ using the update rule of $C(t)$. Denote $\Delta C(t) := C(t+1) - C(t)$. Since $\eta \leq \frac{d_e^2}{40d^2T}$, $\Delta C(t) < \frac{1}{5}$. Then we have

$$\overline{s_+}(t+1) = \mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\,\frac{1}{q + \sum_{i \notin y} e^{-(C(t)+\Delta C(t))(1-\boldsymbol{e}_i^\top \boldsymbol{e}_y)}}$$

$$
\begin{aligned}
&\leq \mathbb{E}_y \, \mathbb{E}_{\boldsymbol{E}}^{(R)} \frac{1}{q + \sum_{i \notin y} e^{-C(t)(1-\boldsymbol{e}_i^\top \boldsymbol{e}_y)} e^{\Delta C(t)(\boldsymbol{e}_i^\top \boldsymbol{e}_y - 1)}} \\
&\leq \mathbb{E}_y \, \mathbb{E}_{\boldsymbol{E}}^{(R)} \frac{1}{q + \sum_{i \notin y} e^{-C(t)(1-\boldsymbol{e}_i^\top \boldsymbol{e}_y)} (1 - \Delta C(t)(1 - \boldsymbol{e}_i^\top \boldsymbol{e}_y))} \\
&\leq \mathbb{E}_y \, \mathbb{E}_{\boldsymbol{E}}^{(R)} \Bigg(\frac{1}{q + \sum_{i \notin y} e^{-C(t)(1-\boldsymbol{e}_i^\top \boldsymbol{e}_y)}} \\
&\qquad + \frac{5}{4} \left(\frac{1}{q + \sum_{i \notin y} e^{-C(t)(1-\boldsymbol{e}_i^\top \boldsymbol{e}_y)}} \right)^2 \sum_{i \notin y} e^{C(t)(\boldsymbol{e}_i^\top \boldsymbol{e}_y - 1)} (1 - \boldsymbol{e}_i^\top \boldsymbol{e}_y) \Delta C(t) \Bigg) \\
&\qquad\qquad (\textstyle\sum_{i \notin y} e^{C(t)(\boldsymbol{e}_i^\top \boldsymbol{e}_y - 1)} (1 - \boldsymbol{e}_i^\top \boldsymbol{e}_y) \Delta C(t) < \frac{1}{5}, \text{ due to } \eta \leq \frac{d_e^2}{40 d^2 T}) \\
&\leq \overline{s_+} + \frac{5}{4} \frac{1-\delta}{1-2\delta} \mathbb{E}_y \, \mathbb{E}_{\boldsymbol{E}}^{(R)} \, s_+^2 \left(\frac{1}{s_+} - q \right) \Delta C(t) \qquad (\textstyle\sum_{i \notin y} e^{C(t)(\boldsymbol{e}_i^\top \boldsymbol{e}_y - 1)} = \frac{1}{s_+} - q) \\
&= \overline{s_+} + \frac{45}{32} \mathbb{E}_y \, \mathbb{E}_{\boldsymbol{E}}^{(R)} \, s_+ (1 - q s_+) \Delta C(t) \qquad (\delta < 0.1) \\
&\leq \overline{s_+} + \frac{45}{32} \overline{s_+} (1 - q \overline{s_+}) \Delta C(t) \qquad (\mathbb{E}[{s_+}^2] \geq \mathbb{E}[s_+]^2)
\end{aligned}
$$

Then we lower bound the decrement of $\alpha^*(t+1)$ and $\frac{(1-q\overline{s_+}(t+1))}{q\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)}$, respectively. Denote $\Delta s = \overline{s_+}(t+1) - \overline{s_+}(t)$.

$$
\begin{aligned}
\alpha^*(t+1) &= \frac{(T-q)\overline{s_+}(t+1)}{Tq\overline{s_+}(t+1)^2 - 2q\overline{s_+}(t+1) + 1} \\
&= \frac{(T-q)}{Tq\overline{s_+}(t+1) - 2q + 1/\overline{s_+}(t+1)} \\
&\geq \frac{(T-q)}{Tq\overline{s_+}(t+1) - 2q + 1/\overline{s_+}(t+1)} \\
&\geq \frac{(T-q)}{Tq\overline{s_+}(t) - 2q + 1/\overline{s_+}(t) + Tq\Delta s} \\
&\geq \frac{(T-q)}{Tq\overline{s_+}(t) - 2q + 1/\overline{s_+}(t)} - \frac{(T-q)Tq\Delta s}{(Tq\overline{s_+}(t) - 2q + 1/\overline{s_+}(t))^2} \\
&= \alpha^*(t) - \frac{(T-q)Tq\Delta s}{(Tq\overline{s_+}(t) - 2q + 1/\overline{s_+}(t))^2} \\
&= \alpha^*(t) - \frac{Tq\overline{s_+}\Delta s}{(Tq\overline{s_+}^2(t) - 2q\overline{s_+}(t) + 1)} \alpha^*(t)
\end{aligned}
$$

$$
\begin{aligned}
\frac{(1 - q\overline{s_+}(t+1))}{q\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)} &= \frac{1}{q\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)} - \frac{1}{(T\overline{s_+}(t+1)-1)} \\
&\geq \frac{1}{q\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)} - \frac{1}{(T\overline{s_+}(t)-1)} \qquad (\overline{s_+}(t+1) \geq \overline{s_+}(t)) \\
&= \frac{1}{qT} \cdot \frac{1}{\overline{s_+}(t+1)} \cdot \frac{1}{\overline{s_+}(t+1) - \frac{1}{T}} - \frac{1}{(T\overline{s_+}(t)-1)} \\
&\geq \frac{1}{qT} \left(\frac{1}{\overline{s_+}(t)} - \frac{\Delta s}{\overline{s_+}^2(t)} \right) \cdot \\
&\quad \left(\frac{1}{\overline{s_+}(t) - \frac{1}{T}} - \frac{\Delta s}{(\overline{s_+}(t) - \frac{1}{T})^2} \right) - \frac{1}{(T\overline{s_+}(t)-1)} \\
&\geq \frac{(1 - q\overline{s_+}(t))}{q\overline{s_+}(t)(T\overline{s_+}(t)-1)} - \frac{\Delta s}{q s_+(t)^2 (T s_+ - 1)} - \frac{\Delta s}{q s_+(t)(T s_+ - 1)^2} \\
&= \frac{(1 - q s_+(t))}{q\overline{s_+}(t)(T\overline{s_+}(t)-1)} - \frac{\Delta s (2T s_+ - 1)}{q s_+(t)^2 (T s_+ - 1)^2}
\end{aligned}
$$

Then plug in the original term, we have the lower bound for

$$\begin{aligned}
&\left(1+\frac{(8dT-d_e)(1-q\overline{s_+}(t+1))}{8dTq\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)}\right)\alpha^*(t+1)\\
\geq&\left(1+\frac{(4dT-d_e)(1-q\overline{s_+}(t))}{4dq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t)+\frac{d_e(1-q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\alpha^*(t)\\
&-\left(1+\frac{(8dT-d_e)(1-q\overline{s_+}(t+1))}{8dTq\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)}\right)\frac{Tq\overline{s_+}\Delta s}{(Tq\overline{s_+}^2(t)-2q\overline{s_+}(t)+1)}\alpha^*(t)\\
&-\frac{\Delta s(2T\overline{s_+}-1)}{q\overline{s_+}(t)^2(T\overline{s_+}-1)^2}\alpha^*(t)
\end{aligned}$$

Since $\frac{(8dT-d_e)(1-q\overline{s_+}(t+1))}{8dTq\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)}\leq\frac{8dT-d_e}{8dT}\leq 1$ when $s_+\geq\frac{1}{\sqrt{Tq}}$, we have

$$\begin{aligned}
&\left(1+\frac{(8dT-d_e)(1-q\overline{s_+}(t+1))}{8dTq\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)}\right)\alpha^*(t+1)\\
\geq&\left(1+\frac{(4dT-d_e)(1-q\overline{s_+}(t))}{4dq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t)+\frac{d_e(1-q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\alpha^*(t)\\
&-\frac{2Tq\overline{s_+}\Delta s}{(Tq\overline{s_+}^2(t)-2q\overline{s_+}(t)+1)}\alpha^*(t)-\frac{\Delta s(2T\overline{s_+}-1)}{q\overline{s_+}(t)^2(T\overline{s_+}-1)^2}\alpha^*(t)
\end{aligned}\tag{***}$$

Then we need to prove that (here $\overline{s_+}:=\overline{s_+}(t)$) to show **(S4)**.

$$\begin{aligned}
&\frac{d_e(1-q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\geq\frac{2Tq\overline{s_+}\Delta s}{(Tq\overline{s_+}^2(t)-2q\overline{s_+}(t)+1)}+\frac{\Delta s(2T\overline{s_+}-1)}{q\overline{s_+}(t)^2(T\overline{s_+}-1)^2}\\
\iff&\frac{d_e(1-q\overline{s_+})}{8dTq(T\overline{s_+}-1)}\geq\frac{2Tq\overline{s_+}^2\Delta s}{(Tq\overline{s_+}^2-2q\overline{s_+}+1)}+\frac{\Delta s(2T\overline{s_+}-1)}{q\overline{s_+}(T\overline{s_+}-1)^2}
\end{aligned}$$

We have that the right hand side has the following upper bound ($T\gg q$):

$$\begin{aligned}
&\frac{2Tq\overline{s_+}^2\Delta s}{(Tq\overline{s_+}^2-2q\overline{s_+}+1)}+\frac{\Delta s(2T\overline{s_+}-1)}{q\overline{s_+}(T\overline{s_+}-1)^2}\\
\leq&\frac{45Tq\overline{s_+}^3(1-q\overline{s_+})\Delta C(t)}{16(Tq\overline{s_+}^2-2q\overline{s_+}+1)}+\frac{45(2T\overline{s_+}-1)\overline{s_+}(1-q\overline{s_+})\Delta C(t)}{32q\overline{s_+}(T\overline{s_+}-1)^2} && (\Delta s\leq\tfrac{45}{32}\overline{s_+}(1-q\overline{s_+})\Delta C(t))\\
\leq&\frac{45Tq\overline{s_+}^3(1-q\overline{s_+})\Delta C(t)}{16q\overline{s_+}(T\overline{s_+}-1)}+\frac{135\overline{s_+}(1-q\overline{s_+})\Delta C(t)}{32q\overline{s_+}(T\overline{s_+}-1)} && (s_+\leq 1/q, T\geq 4q)\\
\leq&\frac{45Tq\overline{s_+}^2(1-q\overline{s_+})\Delta C(t)}{16q(T\overline{s_+}-1)}+\frac{135(1-q\overline{s_+})\Delta C(t)}{32q(T\overline{s_+}-1)}
\end{aligned}$$

Let $\alpha=(1+\gamma\frac{1-q\overline{s_+}}{q\overline{s_+}(T\overline{s_+}-1)})\alpha^*$, then we have the upper bound for the update

$$\begin{aligned}
\Delta C(t)\leq&\,\eta\frac{1-\delta}{1-2\delta}\cdot\frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)\left(1-\frac{q\alpha(t)(T\overline{s_+}-1)}{T-q}\right)s_+\\
\leq&\,\frac{9}{8}\eta(1-\gamma)(1+\gamma\frac{1-q\overline{s_+}}{q\overline{s_+}(T\overline{s_+}-1)})\frac{d\alpha^*}{d_e(Tq\overline{s_+}^2-2q\overline{s_+}+1)}\overline{s_+}(1-q\overline{s_+})^2 && (\mathbb{E}[s_+{}^2]\geq\mathbb{E}[s_+]^2,\delta<0.1)\\
\leq&\,\eta(1-\gamma^2)\frac{9d(T-q)\overline{s_+}^2(1-q\overline{s_+})^2}{8d_e(Tq\overline{s_+}^2-2q\overline{s_+}+1)^2} && \text{(UP1)}\\
\leq&\,\eta(1-\gamma^2)\frac{9d(T-q)(1-q\overline{s_+})^2}{8d_e(Tq\overline{s_+}-2q+1/\overline{s_+})^2}\\
\leq&\,\eta(1-\gamma^2)\frac{9d(T-q)(1-\frac{\sqrt{q}}{\sqrt{T}})^2}{8d_e(2\sqrt{Tq}-2q)^2} && (s_+\geq 1/\sqrt{Tq}.)
\end{aligned}$$

$$\leq \frac{9\eta d}{32qd_e}. \tag{UP2}$$

Since $\eta \leq \frac{d_e}{40d^2T}$, plug the upper bounds (UP1), (UP2) for $\Delta C(t)$ back to the two terms in **(***)** respectively:

$$\begin{aligned}
&\frac{2Tq\overline{s_+}^2\Delta s}{(Tq\overline{s_+}^2 - 2q\overline{s_+} + 1)} + \frac{\Delta s(2T\overline{s_+} - 1)}{q\overline{s_+}(T\overline{s_+} - 1)^2} \\
\leq\ & \frac{45Tq\overline{s_+}^2(1 - q\overline{s_+})\Delta C(t)}{16q(T\overline{s_+} - 1)} + \frac{135(1 - q\overline{s_+})\Delta C(t)}{32q(T\overline{s_+} - 1)} \\
\leq\ & \frac{405dT(T - q)\overline{s_+}^4(1 - q\overline{s_+})^3}{128(T\overline{s_+} - 1)d_e(Tq\overline{s_+}^2 - 2q\overline{s_+} + 1)^2} + \frac{135(1 - q\overline{s_+})\Delta C(t)}{32q(T\overline{s_+} - 1)} \cdot \frac{9\eta d}{32q} \\
\leq\ & \frac{405\eta dT(1 - q\overline{s_+})}{128q(T\overline{s_+} - 1)(T - q)qd_e} + \frac{1215(1 - q\overline{s_+})}{1024q(T\overline{s_+} - 1)} \cdot \frac{\eta d}{qd_e} \\
\leq\ & \frac{405\eta d(1 - q\overline{s_+})}{128q(T\overline{s_+} - 1)d_e} + \frac{1215(1 - q\overline{s_+})}{1024q(T\overline{s_+} - 1)} \cdot \frac{\eta d}{qd_e} && (q \geq 2, T \geq 4q) \\
\leq\ & \frac{d_e(1 - q\overline{s_+})}{8dTq(T\overline{s_+} - 1)}.
\end{aligned}$$

Therefore, we proved the inequality. And therefore, **(S2)** is proved, which also leads to **(S1)**.

Finally, we consider $\alpha(t) \in \left(\left(1 + \frac{(4dT - d_e)(1 - q\overline{s_+}(t))}{4dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t), \left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t)\right]$. Now, since by update rule,

$$\alpha(t+1) = \alpha(t) + \eta\overline{s_+}\left(1 - \frac{\alpha(t)}{\alpha^*(t)}\right) \leq \left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t) - \frac{\eta(4dT - d_e)(1 - q\overline{s_+}(t))}{4dTq(T\overline{s_+}(t) - 1)}$$

Therefore, it's sufficient to prove that

$$\begin{aligned}
&\left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t+1))}{8dTq\overline{s_+}(t+1)(T\overline{s_+}(t+1) - 1)}\right)\alpha^*(t+1) \\
\geq\ & \left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t) - \frac{\eta(4dT - d_e)(1 - q\overline{s_+}(t))}{4dTq(T\overline{s_+}(t) - 1)}
\end{aligned}$$

Note Equation (***) gives the lower bound for the left hand side:

$$\begin{aligned}
&\left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t+1))}{8dTq\overline{s_+}(t+1)(T\overline{s_+}(t+1) - 1)}\right)\alpha^*(t+1) \\
\geq\ & \left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t) \\
&- \frac{2Tq\overline{s_+}\Delta s}{(Tq\overline{s_+}^2(t) - 2q\overline{s_+}(t) + 1)}\alpha^*(t) - \frac{\Delta s(2T\overline{s_+} - 1)}{q\overline{s_+}(t)^2(T\overline{s_+} - 1)^2}\alpha^*(t) && \textbf{(\#\#)}
\end{aligned}$$

Yet when $\alpha(t) \geq \left(1 + \frac{(4dT - d_e)(1 - q\overline{s_+}(t))}{4dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t)$, we have a better upper bound for $\Delta C(t)$:

$$\begin{aligned}
\Delta C(t) &\leq \eta\frac{1 - \delta}{1 - 2\delta} \cdot \frac{d\alpha(t)}{d_e}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)\left(1 - \frac{q\alpha(t)(T\overline{s_+} - 1)}{T - q}\right)s_+ \\
&\leq \frac{9}{8}\eta(1 - \gamma)(1 + \gamma\frac{1 - q\overline{s_+}}{q\overline{s_+}(T\overline{s_+} - 1)})\frac{d\alpha^*}{d_e(Tq\overline{s_+}^2 - 2q\overline{s_+} + 1)}\overline{s_+}(1 - q\overline{s_+})^2 && (\mathbb{E}[s_+{}^2] \geq \mathbb{E}[s_+]^2, \delta < 0.1) \\
&\leq \eta(1 - \gamma^2)\frac{9d(T - q)\overline{s_+}^2(1 - q\overline{s_+})^2}{8d_e(Tq\overline{s_+}^2 - 2q\overline{s_+} + 1)^2} \\
&\leq \eta\left(1 - \left(\frac{4dT - d_e}{4dT}\right)^2\right)\frac{9d(T - q)\overline{s_+}^2(1 - q\overline{s_+})^2}{8d_e(Tq\overline{s_+}^2 - 2q\overline{s_+} + 1)^2}
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{\eta}{2dT/d_e} \cdot \frac{9d(T-q)\overline{s_+}^2(1-q\overline{s_+})^2}{8d_e(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)^2} \\
&= \frac{9\eta(T-q)\overline{s_+}^2(1-q\overline{s_+})^2}{16T(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)^2}
\end{aligned}
$$

Then we need to bound both terms in Equation (##) (for simplicity denote s_+ as $s_+(t)$):

$$
\begin{aligned}
&\frac{2Tq\overline{s_+}\Delta s}{(Tq\overline{s_+}^2(t) - 2q\overline{s_+}(t)+1)}\alpha^*(t) = \frac{2Tq(T-q)\overline{s_+}^2\Delta s}{(Tq\overline{s_+}^2(t) - 2q\overline{s_+}(t)+1)^2} \\
&\leq \frac{45Tq(T-q)\overline{s_+}^3(1-q\overline{s_+})\Delta C(t)}{16(Tq\overline{s_+}^2(t) - 2q\overline{s_+}(t)+1)^2} \\
&\leq \frac{405\eta q(T-q)^2\overline{s_+}^5(1-q\overline{s_+})^3}{256(Tq\overline{s_+}^2(t) - 2q\overline{s_+}(t)+1)^4} && \text{(Plug in } \Delta C(t) \text{ bound.)} \\
&\leq \frac{405\eta}{256}\cdot(T-q)^2\cdot\frac{(1-q\overline{s_+})}{q\overline{s_+}(T\overline{s_+}-1)}\cdot\frac{q\overline{s_+}^5(1-q\overline{s_+})^2}{(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)^3} && (s_+ \leq \tfrac{1}{q}) \\
&= \frac{405\eta}{256}\cdot(T-q)^2\cdot\frac{(1-q\overline{s_+})}{q(T\overline{s_+}-1)}\cdot\frac{q(1-q\overline{s_+})^2}{(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)(Tq - 2q/\overline{s_+} + 1/\overline{s_+}^2)^2} \\
&\leq \frac{405\eta}{256}\cdot(T-q)^2\cdot\frac{(1-q\overline{s_+})}{q(T\overline{s_+}-1)}\cdot\frac{q(1-q\overline{s_+})^2}{(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)(Tq - q^2)^2} && (s_+ \leq \tfrac{1}{q}) \\
&\leq \frac{405\eta}{256}\cdot\frac{(1-q\overline{s_+})}{q(T\overline{s_+}-1)}\cdot\frac{(1-q\overline{s_+})^2}{(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)q} \\
&\leq \frac{405\eta}{256}\cdot\frac{(1-q\overline{s_+})}{q(T\overline{s_+}-1)}\cdot\frac{1}{2q} && (s_+ \geq \tfrac{1}{\sqrt{Tq}}) \\
&\leq \frac{405\eta}{1024}\cdot\frac{(1-q\overline{s_+})}{q(T\overline{s_+}-1)} && (s_+ \geq \tfrac{1}{\sqrt{Tq}})
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\Delta s(2T\overline{s_+}-1)}{q\overline{s_+}(t)^2(T\overline{s_+}-1)^2}\alpha^*(t) \\
&\leq \frac{(2T\overline{s_+}-1)}{q\overline{s_+}(t)^2(T\overline{s_+}-1)^2}\alpha^*(t)\cdot\frac{45}{32}\overline{s_+}(1-q\overline{s_+})\Delta C(t) && \text{(Plug in } \Delta s) \\
&\leq \frac{135(T-q)(1-q\overline{s_+})}{32q(T\overline{s_+}-1)(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)}\cdot\frac{9\eta(T-q)\overline{s_+}^2(1-q\overline{s_+})^2}{16T(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)^2} && \text{(Plug in } \Delta C(t)) \\
&= \frac{1215\eta(T-q)^2\overline{s_+}^2(1-q\overline{s_+})^3}{512qT(T\overline{s_+}-1)(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)^3} \\
&\leq \frac{1215\eta}{512}\cdot\frac{(1-q\overline{s_+})}{q(T\overline{s_+}-1)}\frac{(1-q\overline{s_+})^2}{(Tq\overline{s_+}^2 - 2q\overline{s_+}+1)^2(Tq - 2q/\overline{s_+} + 1/\overline{s_+}^2)} \\
&\leq \frac{1215\eta}{4096}\cdot\frac{(1-q\overline{s_+})}{q(T\overline{s_+}-1)}.
\end{aligned}
$$

Therefore, by Equation (##) we have

$$
\begin{aligned}
&\left(1 + \frac{(8dT-d_e)(1-q\overline{s_+}(t+1))}{8dTq\overline{s_+}(t+1)(T\overline{s_+}(t+1)-1)}\right)\alpha^*(t+1) \\
&\geq \left(1 + \frac{(8dT-d_e)(1-q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t) \\
&\quad - \frac{2Tq\overline{s_+}\Delta s}{(Tq\overline{s_+}^2(t) - 2q\overline{s_+}(t)+1)}\alpha^*(t) - \frac{\Delta s(2T\overline{s_+}-1)}{q\overline{s_+}(t)^2(T\overline{s_+}-1)^2}\alpha^*(t) \\
&\geq \left(1 + \frac{(8dT-d_e)(1-q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t)-1)}\right)\alpha^*(t) - \left(\frac{405}{1024} + \frac{1215}{4096}\right)\frac{\eta(1-qs_+)}{q(Ts_+-1)}
\end{aligned}
$$

$$\geq \left(1 + \frac{(8dT - d_e)(1 - q\overline{s_+}(t))}{8dTq\overline{s_+}(t)(T\overline{s_+}(t) - 1)}\right)\alpha^*(t) - \frac{\eta(4dT - d_e)(1 - q\overline{s_+}(t))}{4dTq(T\overline{s_+}(t) - 1)} \qquad (dT \geq d_e.)$$
$$\geq \alpha(t+1).$$

Therefore, we finish the induction proof for (IH2).

With the induction hypothesis, we can analyze the upper bound of convergence time. For **Phase I**, we have the lower bound for $\Delta C(t)$ (note that $1/2 < \alpha < 1$):

$$\begin{aligned}
\Delta C(t+1) &\geq \eta \frac{1-3\delta}{1-2\delta} \cdot \frac{d\alpha(t)}{d_e} \mathbb{E}_y \mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)\left(1 - \frac{q\alpha(t)(T\overline{s_+} - 1)}{T - q}\right)s_+ \\
&= \frac{7\eta}{8} \cdot \frac{\alpha d}{d_e} \mathbb{E}_y \mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)\frac{T - q + q - Tq\overline{s_+}}{T - q}s_+ \\
&= \frac{7\eta dT}{16d_e(T-q)} \mathbb{E}_y \mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)(1 - q\overline{s_+})s_+
\end{aligned}$$

And for **Phase II**, we have the lower bound since $\alpha \leq \left(1 + \frac{(8Td - d_e)(1 - q\overline{s_+})}{8Td q\overline{s_+}(T\overline{s_+} - 1)}\right)\alpha^*(t)$:

$$\begin{aligned}
\Delta C(t) &\geq \eta(1 - \frac{8Td - d_e}{8Td})\left(1 + \frac{(8Td - d_e)(1 - q\overline{s_+})}{8Td q\overline{s_+}(T\overline{s_+} - 1)}\right)\alpha^*(t) \cdot \frac{7d\,\mathbb{E}_y \mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)(1 - q\overline{s_+})s_+}{8d_e(Tq\overline{s_+}^2 - 2q\overline{s_+} + 1)} \\
&\geq \frac{\eta d_e}{8dT} \cdot \frac{7d(T-q)\overline{s_+}\,\mathbb{E}_y \mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)(1 - q\overline{s_+})s_+}{8d_e(Tq\overline{s_+}^2 - 2q\overline{s_+} + 1)^2} \\
&\geq \frac{7\eta(T-q)\overline{s_+}\,\mathbb{E}_y \mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)(1 - q\overline{s_+})s_+}{64T(Tq\overline{s_+}^2 - 2q\overline{s_+} + 1)^2}
\end{aligned}$$

Then we also divide the training trajectory into two stages as in Huang et al. (2023): in the first stage from $(0, t_1]$, all possible s_+ grow to $1/2q$. In the second stage $(t_1, t_2]$, $C(t)$ grows large enough s.t. $\overline{s_+} \geq \frac{1}{q} - \frac{1}{q}\sqrt{\frac{q(T-q)\epsilon}{dT}}$.

For the first stage, it's necessary that $C(t) \geq \frac{1-2\delta}{1-3\delta}\log\frac{T-q}{q}$. This is because by Lemma E.15, we have:

$$\frac{1}{q + (T-q)e^{-\frac{1-3\delta}{1-2\delta}C}} \leq s_+ \leq \frac{1}{q + (T-q)e^{-\frac{1-\delta}{1-2\delta}C}},$$

To make any $s_+(\boldsymbol{E})$ reaches this threshold $1/2q$, $C(t_1) \geq \frac{1-2\delta}{1-3\delta}\log\frac{T-q}{q}$ is necessary. At this time t_1, all $s_+ \leq 1/2q$, which means the expectation $\overline{s_+} \leq 1/2q$. At this iteration, we can also upper bound all the attention score:

$$s_+ \leq \frac{1}{q + (T-q)e^{-\frac{1-\delta}{1-2\delta}C(t)}} \leq \frac{1}{q + (T-q)^{-\frac{2\delta}{1-3\delta}}q^{\frac{1-\delta}{1-3\delta}}}$$

That means

$$\begin{aligned}
(1 - qs_+) &\geq \left(1 - q \cdot \frac{1}{q + (T-q)^{-\frac{2\delta}{1-3\delta}}q^{\frac{1-\delta}{1-3\delta}}}\right) \\
&= \left(\frac{q^{\frac{2\delta}{1-3\delta}}}{(T-q)^{\frac{2\delta}{1-3\delta}} + q^{\frac{2\delta}{1-3\delta}}}\right) \geq \left(\frac{q}{T}\right)^{\frac{2\delta}{1-3\delta}}
\end{aligned}$$

Since all s_+ satisfy the lower bound, the expectation $\overline{s_+}$ also has this lower bound. Therefore during this stage, by **Phase I** lower bound for $\Delta C(t)$ we have:

$$\begin{aligned}
\Delta C(t) &\geq \frac{7\eta dT}{16d_e(T-q)} \mathbb{E}_y \mathbb{E}_{\boldsymbol{E}}^{(R)}(1 - qs_+)(1 - q\overline{s_+})s_+ \\
&\geq \frac{7\eta d}{16d_e(T-q)} \cdot \left(\frac{q}{T}\right)^{\frac{4\delta}{1-3\delta}} \qquad (s_+ \geq \tfrac{1}{T}, (1 - qs_+) \geq \left(\tfrac{q}{T}\right)^{\frac{2\delta}{1-3\delta}})
\end{aligned}$$

And for the **Phase II** lower bound, we have:

$$\begin{aligned}\Delta C(t) &\geq \frac{7\eta(T-q)\overline{s_+}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)(1-q\overline{s_+})s_+}{64T(T\overline{s_+}^2-2q\overline{s_+}+1)^2}\\ &\geq \frac{7\eta(T-q)\overline{s_+}^2\left(\frac{q}{T}\right)^{\frac{4\delta}{1-3\delta}}}{64T(T\overline{s_+}^2-2q\overline{s_+}+1)^2}\\ &\geq \frac{7\eta(T-q)\left(\frac{q}{T}\right)^{\frac{4\delta}{1-3\delta}}}{64T(Tq-2q/\overline{s_+}+1/\overline{s_+}^2)(T/q-1)}\\ &\geq \frac{7\eta\left(\frac{q}{T}\right)^{\frac{4\delta}{1-3\delta}}}{64T(T-q)}\end{aligned}$$

Since $T \gg d_e = \Theta(q\log T)$, $\Delta C(t) \geq \frac{7\eta\left(\frac{q}{T}\right)^{\frac{4\delta}{1-3\delta}}}{64T(T-q)}$ through all the first stage. Therefore, for $C(t)$ to reach $\Theta(\log\frac{T-q}{q})$, it takes at most $O\left(\log\frac{T-q}{q}/\frac{7\eta\left(\frac{q}{T}\right)^{\frac{4\delta}{1-3\delta}}}{64T(T-q)}\right) = \tilde{O}(T^{\frac{2-2\delta}{1-3\delta}})$ iterations.

For the second stage, we need $C(t_2) \geq \frac{1-2\delta}{1-3\delta}\log\left(\frac{T-q}{q}\sqrt{\frac{Td}{(T-q)q\epsilon}}\right)$ so that $\overline{s_+} \geq \frac{1}{q} - \frac{1}{q}\sqrt{\frac{q(T-q)\epsilon}{dT}}$. Since all $s_+ \geq \frac{1}{2q}$ during this period, we can lower bound the increment:

$$\begin{aligned}\Delta C(t) &\geq \frac{7\eta(T-q)\overline{s_+}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)(1-q\overline{s_+})s_+}{64T(T\overline{s_+}^2-2q\overline{s_+}+1)^2}\\ &\geq \frac{7\eta(T-q)\cdot\frac{1}{2q}\,\mathbb{E}_y\,\mathbb{E}_{\boldsymbol{E}}^{(R)}(1-qs_+)(1-q\overline{s_+})\cdot\frac{1}{2q}}{64T(\frac{T}{q}-1)^2}\\ &\geq \frac{7\eta}{128T(T-q)}\,\mathbb{E}_y(1-q\overline{s_+})^2\\ &\geq \frac{7\eta q\epsilon}{128dT^2} \qquad \text{(Before } t_2,\ \overline{s_+} \leq \tfrac{1}{q} - \tfrac{1}{q}\sqrt{\tfrac{q(T-q)\epsilon}{dT}}\text{)}\end{aligned}$$

Hence it takes at most $t_2 \leq O(\frac{1-2\delta}{1-3\delta}\log\left(\frac{T-q}{q}\sqrt{\frac{Td}{2(T-q)q\epsilon}}\right)/\frac{7\eta q\epsilon}{128dT^2}) = \tilde{O}(\frac{dT^2}{\eta\epsilon})$ iterations to converge.

Finally, we check that if $\overline{s_+} \geq \frac{1}{q} - \frac{1}{q}\sqrt{\frac{q(T-q)\epsilon}{dT}}$, then the loss is smaller than ϵ. This part of proof is the similar to Theorem D.4[7]. By Lemma E.3:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}(t)) &= \frac{1}{2}\,\mathbb{E}\left[d\left\|\frac{1}{q}\boldsymbol{Y} - \alpha(t)\,\mathbb{E}_{\boldsymbol{E}}^{(R)}[\boldsymbol{S}_y]\right\|^2\right]\\ &= \frac{d}{2(T-q)}\left((T-q)q\left(\alpha(t)\overline{s_+} - \frac{1}{q}\right)^2 + \alpha(t)^2(1-q\overline{s_+})^2\right)\end{aligned}$$

While $\alpha(t) \in \left[1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}, \left(1 + \frac{(8Td-d_e)(1-q\overline{s_+})}{8Tdq\overline{s_+}(T\overline{s_+}-1)}\right)\alpha^*(t)\right] := [\alpha_1, \alpha_2]$, the loss value is upper bounded by

$$\max_{j\in\{1,2\}}\frac{d}{2(T-q)}\left((T-q)q\left(\alpha_j\overline{s_+} - \frac{1}{q}\right)^2 + \alpha_j^2(1-q\overline{s_+})^2\right)$$

For $\alpha_1 = 1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}$, we have

$$\mathcal{L}(\boldsymbol{\theta}(t)) = \frac{d}{2(T-q)}\left((T-q)q\left(\alpha(t)\overline{s_+} - \frac{1}{q}\right)^2 + \alpha(t)^2(1-q\overline{s_+})^2\right)$$

[7] By 'similar', one can replace all the s_+ in the proof of Theorem D.4 with $\overline{s_+}$ here.

$$
\begin{aligned}
&\leq \frac{d}{2(T-q)}\left(\frac{T-q}{q}(q\alpha(t)\overline{s_+}-1)^2+(1-q\overline{s_+})^2\right)\\
&\leq \frac{d}{2(T-q)}\left(\frac{T-q}{q}\left(q\overline{s_+}-1+0.1q\overline{s_+}\sqrt{\frac{q(T-q)\epsilon}{dT}}\right)^2+(1-q\overline{s_+})^2\right)\\
&\leq \frac{d}{2(T-q)}\left(\frac{T-q}{q}\cdot\left((1-q\overline{s_+})^2+0.2\sqrt{\frac{q(T-q)\epsilon}{dT}}(1-q\overline{s_+})+0.01\cdot\frac{q(T-q)\epsilon}{dT}\right)+(1-q\overline{s_+})^2\right)\\
&\leq \frac{d}{2(T-q)}\left(\frac{T-q}{q}\cdot\left((1-q\overline{s_+})^2+0.2\sqrt{\frac{q(T-q)\epsilon}{dT}}\cdot\sqrt{\frac{q(T-q)\epsilon}{dT}}+0.01\cdot\frac{q(T-q)\epsilon}{dT}\right)\right)\\
&\quad+\frac{d}{2(T-q)}(1-q\overline{s_+})^2\\
&\leq \frac{d}{2(T-q)}\frac{T}{q}(1-q\overline{s_+})^2+0.21\epsilon\leq\epsilon.
\end{aligned}
$$

For $\alpha_2=\left(1+\frac{(8Td-d_e)(1-q\overline{s_+})}{8Tdq\overline{s_+}(T\overline{s_+}-1)}\right)\alpha^*(t)$, denote $\Delta\alpha=\alpha_2-\alpha^*$. Then $\Delta\alpha\overline{s_+}\leq\frac{2\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^*$. Also, for α^* we have a upper bound (using $\epsilon\leq\frac{dT}{100(T-q)q}$)):

$$
\alpha^*=\frac{T-q}{Tq\overline{s_+}-2q+\frac{1}{\overline{s_+}}}\leq\frac{1}{q\overline{s_+}}\leq\frac{1}{1-\sqrt{\frac{q(T-q)\epsilon}{dT}}}\leq 1+\frac{6}{5}\sqrt{\frac{q(T-q)\epsilon}{dT}}\leq\frac{28}{25}.
$$

Therefore, the loss can be bounded by

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}(t))&=\frac{d}{2(T-q)}\left((T-q)q\left(\alpha_2\overline{s_+}-\frac{1}{q}\right)^2+\alpha_2^2(1-q\overline{s_+})^2\right)\\
&=\frac{d}{2(T-q)}\left((T-q)q\left(\alpha^*\overline{s_+}-\frac{1}{q}\right)^2+\alpha^{*2}(1-q\overline{s_+})^2\right)\\
&\quad+\frac{d}{2(T-q)}\left(-(T-q)\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^*(1-q\overline{s_+}\alpha^*)+\frac{4q^2(T-q)^2\epsilon}{dT^3}\alpha^{*2}\right)\\
&\quad+\frac{d}{2(T-q)}\left(\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^{*2}(1-q\overline{s_+})^2+\frac{4q(T-q)\epsilon}{dT^3}\alpha^{*2}(1-q\overline{s_+})^2\right)\\
&=\frac{d}{2(T-q)}\left((T-q)q\left(\alpha^*\overline{s_+}-\frac{1}{q}\right)^2+\alpha^{*2}(1-q\overline{s_+})^2\right)\\
&\quad+\frac{d}{2(T-q)}\left(-(T-q)\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T(T\overline{q\overline{s_+}}^2-2q\overline{s_+}+1)}\alpha^*(1-q\overline{s_+})^2+\frac{4q^2(T-q)^2\epsilon}{dT^3}\alpha^{*2}\right)\\
&\quad+\frac{d}{2(T-q)}\left(\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^{*2}(1-q\overline{s_+})^2+\frac{4q(T-q)\epsilon}{dT^3}\alpha^{*2}(1-q\overline{s_+})^2\right)\\
&=\frac{dT}{2q(T-q)}(q\overline{s_+}(t)-1)^2+\frac{d}{2(T-q)}\left(-\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T\overline{s_+}}\alpha^{*2}(1-q\overline{s_+})^2+\frac{4q^2(T-q)^2\epsilon}{dT^3}\alpha^{*2}\right)\\
&\quad+\frac{d}{2(T-q)}\left(\frac{4\sqrt{\frac{q(T-q)\epsilon}{dT}}}{T}\alpha^{*2}(1-q\overline{s_+})^2+\frac{4q(T-q)\epsilon}{dT^3}\alpha^{*2}(1-q\overline{s_+})^2\right)
\end{aligned}
$$

$$\leq \frac{\epsilon}{2} + \frac{d}{2(T-q)}\left(\frac{4q^2(T-q)^2\epsilon}{dT^3}\alpha^{*2} + \frac{4q(T-q)\epsilon}{dT^3}\alpha^{*2}(1-q\overline{s_+})^2\right)$$
$$\leq \frac{\epsilon}{2} + \frac{d}{2(T-q)}\left(\frac{4q^2(T-q)^2\epsilon}{dT^3}\left(\frac{28}{25}\right)^2 + \frac{4q(T-q)\epsilon}{dT^3}(\frac{28}{25})^2\right) \leq \epsilon.$$

when $T \geq 4q$.

In conclusion, after **Phase I** (at most $O(\frac{T\log\frac{d}{\epsilon}}{\eta})$ iterations) and **Phase II** (takes at most $\tilde{O}(\frac{T^{\frac{2-2\delta}{1-3\delta}}}{\eta} + \frac{dT^2}{\eta\epsilon})$ iterations), the population loss $\mathcal{L}(\boldsymbol{\theta}(t)) \leq \epsilon$ after time $\tilde{O}(\frac{T^{\frac{2-2\delta}{1-3\delta}}}{\eta} + \frac{T^2 d}{\eta\epsilon})$. □

After proving the convergence to the global minimizer, we can directly have the following corollary on the parameter $\boldsymbol{V}(t)$ and $\boldsymbol{W}(t)$. By the dynamics proved in the theorem, $\boldsymbol{W}(t)$ is always along the $\boldsymbol{I}_{d_e}$ direction, and $\boldsymbol{V}(t)$ should converge to $\boldsymbol{I}_d$. It coincides with the construction in Sanford et al. (2023), showing the constructed one-layer transformer can be learned with GD.

Corollary E.8. *Under the condition of Theorem E.6, after time* $t \geq \tilde{O}(\frac{T^{\frac{2-2\delta}{1-3\delta}}}{\eta} + \frac{T^2 d}{\eta\epsilon})$, *we have* $\boldsymbol{W}(t) = C(t)\boldsymbol{I}_{d_e}, \boldsymbol{V}(t) = \alpha(t)\boldsymbol{I}_d$, *and* $C(t) \geq \frac{1-2\delta}{1-3\delta}\log\left(\frac{T-q}{q}\sqrt{\frac{Td}{(T-q)q\epsilon}}\right)$, $\alpha(t) \in \left[1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}, 1 + \frac{8}{5}\sqrt{\frac{q(T-q)\epsilon}{dT}}\right]$.

Proof. The lower bound for $C(t)$ is proved in the Theorem E.6.

By Equation (IH2), $\alpha^*(t) = \frac{(T-q)\overline{s_+}}{Tq\overline{s_+}^2 - 2q\overline{s_+} + 1}$ and $C(t) \geq \frac{1-2\delta}{1-3\delta}\log\left(\frac{T-q}{q}\sqrt{\frac{Td}{(T-q)q\epsilon}}\right)$, we have

$$\alpha(t) \in \left[1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}, \left(1 + \frac{(8Td - d_e)(1-q\overline{s_+})}{8Tdq\overline{s_+}(T\overline{s_+}-1)}\right)\alpha^*(t)\right], \overline{s_+} \geq \frac{1}{q} - \frac{1}{q}\sqrt{\frac{q(T-q)\epsilon}{dT}}.$$

and

$$\alpha^*(t) = \frac{T-q}{Tq\overline{s_+} - 2q + \frac{1}{\overline{s_+}}} \leq \frac{1}{q\overline{s_+}} \leq \frac{1}{1 - \sqrt{\frac{q(T-q)\epsilon}{dT}}} \leq 1 + \frac{6}{5}\sqrt{\frac{q(T-q)\epsilon}{dT}}.$$

Therefore for $\left(1 + \frac{(8Td-d_e)(1-q\overline{s_+})}{8Tdq\overline{s_+}(T\overline{s_+}-1)}\right)\alpha^*(t)$ we also have an upper bound.

$$\left(1 + \frac{(8Td - d_e)(1-q\overline{s_+})}{8Tdq\overline{s_+}(T\overline{s_+}-1)}\right)\alpha^*(t) \leq \left(1 + \frac{6}{5}\sqrt{\frac{q(T-q)\epsilon}{dT}}\right)\left(1 + \frac{q}{T-q}\sqrt{\frac{q(T-q)\epsilon}{dT}}\right)$$
$$\leq 1 + \frac{8}{5}\sqrt{\frac{q(T-q)\epsilon}{dT}}$$

The last inequality uses $T \geq 4q, \epsilon \leq \frac{dT}{100q(T-q)}$. □

E.4. Length generalization and out-of-distribution guarantee

In this subsection, we present the strength of stochastic positional encoding, which is the strong out-of-distribution guarantee, including the *length generalization* performance mentioned in the main paper. Empirically, in Section 4 we observe stochastic positional encoding has superiority over a fixed set of near-orthogonal positional encodings, especially in length generalization tasks. Here, based on the global minimizer found by gradient descent, we can also present a theoretical guarantee for an out-of-distribution guarantee.

Recall our data model is: The input tokens $\boldsymbol{x}_i, i = 1, 2, ..., T$ are sampled from standard Gaussian distribution, and the q-sparse subset y containing all the averaging indices is uniformly sampled from all q-subsets of the set $[T]$.

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T), \boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d),$$

$$y \sim \text{Unif}\left(\binom{[T]}{q}\right), i \in [T]$$

Suppose our training objective is based on the population distribution with sequence length T_1 and subset size q_1. By "out-of-distribution", we include two different tasks: (1) generalize on unseen data points with q_2-subsets where $q_2 > q$. (2) generalize on longer sequences with $T_2 > T_1$. We know the out-of-distribution loss with sequence length T_2 and q_2 as

$$\mathcal{L}^{(s)}_{T_2,q_2}(\boldsymbol{\theta}) = \frac{1}{2}\,\mathbb{E}_{\boldsymbol{X},y\sim\mathcal{D}_{T_2,q_2}}\left[\|\,\text{STS}_{q_2}(\boldsymbol{X},y) - f^{(s)}_{\boldsymbol{\theta}}(\boldsymbol{X},y)\|_2^2\right] \tag{15}$$

With the constructed transformer in C.3, as long as $d_e = \Theta(q_2 \log T_2)$ where the positional encoding matrix $\boldsymbol{E}' \in \mathbb{R}^{d_e \times T_2}$ with sequence length T_2 can satisfy RIP, the one-layer transformer can express STS_q with any input sequence length $T \le T_2$ and $q \le q_2$. In this paper, we present the following corollary that demonstrates the GD-trained transformer with stochastic PE can also achieve good OOD performance in both subset size $q_2 \ge q_1$ and $T_2 \ge T_1$ under the condition that $\boldsymbol{E}$ can satisfy RIP with maximal sequence length $T_{\max}$ and maximal $q_{\max}$.

Corollary E.9. *Suppose $q_{\max} = \Theta(1), d_e = \Theta(q_{\max}\log T_{\max}/\delta^2), \delta < 1/10, \eta \le \frac{d_e^2}{40d^2T}$. For any $\epsilon \in (0, \frac{dT}{100(T-q)q})$, and we apply gradient descent with zero initialization with $q_1 < q_{\max}, T_1 < T_{\max}$ to train the model. Then we have the following out-of-distribution loss generalization guarantee with $q_2 \in (q_1, q_{\max}], T_2 \in (T_1, T_{\max})$: if the training time $t \ge \tilde{O}(\frac{T^{\frac{2-2\delta}{1-3\delta}}}{\eta} + \frac{T^2 d}{\eta\epsilon})$, we have:*

$$\mathcal{L}^{(s)}_{T_2,q_2}(\boldsymbol{\theta}) \le O\left(\frac{T_2^2\epsilon}{T_1^2}\right)$$

Proof idea. The intuition behind the corollary is that training the transformer with the stochastic architecture on arbitrary T and q where the RIP condition holds can lead to exact the same global minima in the construction equivalent to (Sanford et al., 2023). That means it naturally satisfies all possible T' and q', as long as the RIP condition holds.

Proof. By Corollary E.8, we have $\boldsymbol{W}(t) = C(t)\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}, \boldsymbol{V}(t) = \alpha(t)\begin{bmatrix}\boldsymbol{I}_d & \boldsymbol{0}_{d\times d_e}\end{bmatrix}$, and $C(t) \ge \frac{1-2\delta}{1-3\delta}\log\left(\frac{T-q}{q}\sqrt{\frac{Td}{(T-q)q\epsilon}}\right), \alpha(t) \in \left[1 - 0.1\sqrt{\frac{q(T-q)\epsilon}{dT}}, 1 + \frac{8}{5}\sqrt{\frac{q(T-q)\epsilon}{dT}}\right]$.

Denote $\Delta\alpha = \alpha - 1$. Similarly, we can have the following bounds for the s'_+:

$$1 - \mathbb{E}_{\boldsymbol{E}'}[\boldsymbol{S}_{y'}(i)]q_2 = 1 - q_2\overline{s_+}' \le \frac{\frac{(T_2-q_2)q_1}{T_1-q_1}\sqrt{\frac{(T_1-q_1)q_1\epsilon}{T_1 d}}}{q_2 + \frac{(T_2-q_2)q_1}{T_1-q_1}\sqrt{\frac{(T_1-q_1)q_1\epsilon}{T_1 d}}} \le \frac{\frac{(T_2-q_2)q_1}{T_1-q_1}\sqrt{\frac{(T_1-q_1)q_1\epsilon}{T_1 d}}}{q_2}$$

By Lemma E.3, we have

$$\begin{aligned}
\mathcal{L}^{(s)}_{T_2,q_2}(\boldsymbol{\theta}) &= \frac{d}{2}\,\mathbb{E}_{y'}\left[\left\|\frac{1}{q_2}\boldsymbol{Y}_{T_2} - \alpha(t)\,\mathbb{E}_{\boldsymbol{E}'}[\boldsymbol{S}_{y'}]\right\|_2^2\right]\\
&= \frac{d}{2}\left(\frac{1}{q_2}(1 - q_2\alpha(t)\overline{s_+}')^2 + \alpha(t)^2\frac{(1-q\overline{s_+}')^2}{T_2 - q_2}\right)\\
&\le \frac{d}{2}\cdot\frac{T_2}{(T_2-q_2)q_2}\cdot(1-q\overline{s_+}')^2 - \frac{d}{q_2}\Delta\alpha q_2\overline{s_+}'(1-q\overline{s_+}')\\
&\quad + \frac{d}{q_2}\Delta\alpha^2(q\overline{s_+}')^2 + \frac{d}{2}\cdot\frac{2\Delta\alpha + (\Delta\alpha)^2}{T_2-q_2}(1-q\overline{s_+}')^2 \qquad \text{(Plug in } \overline{s_+}', \Delta\alpha \text{ bounds)}\\
&\le \frac{T_2(T_2-q_2)q_1^3}{2q_2^3T_1(T_1-q_1)} + \frac{(T_2-q_2)q_1^2\epsilon}{10q_2^2T_1} + \frac{32q_1(T_1-q_1)\epsilon}{25q_2T_1} + \frac{3(T_2-q_2)q_1^3\epsilon}{2q_2^2T_1(T_1-q_1)} \le O\left(\frac{T_2^2\epsilon}{T_1^2}\right).
\end{aligned}$$

Therefore, we complete the proof. □

E.5. Supplementary lemmas on conditional expectations

Lemma E.10 (Flipping rows doesn't change the softmax)**.** *Given a q-subset y, denote the softmax output with input $\boldsymbol{E}$ as $\boldsymbol{S}_y(\boldsymbol{E}) = \mathrm{softmax}(C\boldsymbol{E}^\top \boldsymbol{e}_y)$, we have for any i:*

$$\boldsymbol{S}_y(\boldsymbol{E}) = \boldsymbol{S}_y(\boldsymbol{E}'), \boldsymbol{E}_y^\top \boldsymbol{E}_y = \boldsymbol{E}_y'^\top \boldsymbol{E}_y'$$

where $\boldsymbol{E}' = \boldsymbol{F}_i\boldsymbol{E}$, $\boldsymbol{F}_i = \mathrm{diag}(1, ..., 1, -1, 1, ..., 1) \in \mathbb{R}^{d_e \times d_e}$ with the i-th entry being -1.[8]

Proof. Notice that

$$\boldsymbol{E}^\top \boldsymbol{e}_y = \boldsymbol{E}^\top \boldsymbol{E}_y(\boldsymbol{E}_y^\top \boldsymbol{E}_y)^{-1}\mathbf{1}_q.$$

while

$$\boldsymbol{E}'^\top \boldsymbol{e}_y' = \boldsymbol{E}^\top \boldsymbol{F}_i^\top \boldsymbol{F}_i\boldsymbol{E}_y(\boldsymbol{E}_y^\top \boldsymbol{F}_i^\top \boldsymbol{F}_i\boldsymbol{E}_y)^{-1}\mathbf{1}_q = \boldsymbol{E}^\top \boldsymbol{E}_y(\boldsymbol{E}_y^\top \boldsymbol{E}_y)^{-1}\mathbf{1}_q.$$

The second identity is due to $\boldsymbol{F}_i^\top \boldsymbol{F}_i = \boldsymbol{I}_{d_e}$. So the vector inside $\mathrm{softmax}$ is the same, so the outputs are the same. For $\boldsymbol{E}_y^\top \boldsymbol{E}_y = \boldsymbol{E}_y'^\top \boldsymbol{E}_y'$, it's similar:

$$\boldsymbol{E}_y^\top \boldsymbol{E}_y = \boldsymbol{E}_y^\top \boldsymbol{F}_i^\top \boldsymbol{F}_i\boldsymbol{E}_y = \boldsymbol{E}_y'^\top \boldsymbol{E}_y'$$

□

Lemma E.11 (Switching rows doesn't change conditional expectation)**.** *Given a q-subset y, denote the softmax output with input $\boldsymbol{E}$ as $\boldsymbol{S}_y(\boldsymbol{E}) = \mathrm{softmax}(C\boldsymbol{E}^\top \boldsymbol{e}_y)$, we have for any i:*

$$\boldsymbol{S}_y(\boldsymbol{E}) = \boldsymbol{S}_y(\boldsymbol{E}'), \boldsymbol{E}_y^\top \boldsymbol{E}_y = \boldsymbol{E}_y'^\top \boldsymbol{E}_y'$$

where $\boldsymbol{E}' = \boldsymbol{R}_{ij}\boldsymbol{E}$, $\boldsymbol{R}_{ij} \in \mathbb{R}^{d_e \times d_e}$ where $\boldsymbol{R}_{ij}$ is the elementary matrix to switch the i-th row and j-th row of the positional encoding matrix $\boldsymbol{E}$.

Proof. Similar to Lemma E.10, notice that

$$\boldsymbol{E}^\top \boldsymbol{e}_y = \boldsymbol{E}^\top \boldsymbol{E}_y(\boldsymbol{E}_y^\top \boldsymbol{E}_y)^{-1}\mathbf{1}_q.$$

while

$$\boldsymbol{E}'^\top \boldsymbol{e}_y' = \boldsymbol{E}^\top \boldsymbol{R}_{ij}^\top \boldsymbol{R}_{ij}\boldsymbol{E}_y(\boldsymbol{E}_y^\top \boldsymbol{R}_{ij}^\top \boldsymbol{R}_{ij}\boldsymbol{E}_y)^{-1}\mathbf{1}_q = \boldsymbol{E}^\top \boldsymbol{E}_y(\boldsymbol{E}_y^\top \boldsymbol{E}_y)^{-1}\mathbf{1}_q.$$

The second identity is due to $\boldsymbol{R}_{ij}^\top \boldsymbol{R}_{ij} = \boldsymbol{I}_{d_e}$. So the vector inside $\mathrm{softmax}$ is the same, so the outputs are the same.

For $\boldsymbol{E}_y^\top \boldsymbol{E}_y = \boldsymbol{E}_y'^\top \boldsymbol{E}_y'$, it's similar:

$$\boldsymbol{E}_y^\top \boldsymbol{E}_y = \boldsymbol{E}_y^\top \boldsymbol{R}_{ij}^\top \boldsymbol{R}_{ij}\boldsymbol{E}_y = \boldsymbol{E}_y'^\top \boldsymbol{E}_y'$$

□

Lemma E.12 (Off-diagonal entries of the gradient have 0 expectation)**.** *Given a q-subset y, for any function $f(\boldsymbol{S}_y(\boldsymbol{E})) : \mathbb{R}^{d_e \times T} \to \mathbb{R}$, we have for any $i \in [T]$:*

$$u_k^\top \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_i\boldsymbol{e}_y^\top\right]u_j^\top = 0, j \neq k.$$

where u_i is the one-hot vector $(\mathbb{1}\{i = 1\}, \mathbb{1}\{i = 2\}, ..., \mathbb{1}\{i = d_e\})$.

Proof. First, the exact expression the (k, j)-th entry in the matrix is:

$$u_k^\top \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_i\boldsymbol{e}_y^\top\right]u_j = \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top \boldsymbol{E}_y)^{-1}\begin{bmatrix}\boldsymbol{e}_{y_1,j}\\ \boldsymbol{e}_{y_2,j}\\ \vdots\\ \boldsymbol{e}_{y_q,j}\end{bmatrix}\right]$$

[8] $\boldsymbol{F}_i$ is to flip the i-th row of the positional encoding matrix $\boldsymbol{E}$.

where $\boldsymbol{e}_{i,j}$ denote the j-th entry of position encoding vector $\boldsymbol{e}_i$, $\left[\boldsymbol{e}_{y_1,j}, \boldsymbol{e}_{y_2,j}, \cdots, \boldsymbol{e}_{y_q,j}\right]^\top$ as $\boldsymbol{E}_y[j]$, and y_i is the i-th index in the subset y. We expand the conditional expectation (Here $\Pr^{(R)}(\cdot) := \Pr(\cdot|\boldsymbol{E}$ has (q,δ)-RIP)):

$$
\begin{aligned}
& u_k^\top \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_i\boldsymbol{e}_y^\top\right]u_j \\
= & \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[j]\right] \\
= & \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[j]\middle|\boldsymbol{e}_{ik}=\frac{1}{\sqrt{d_e}}\right]\Pr^{(R)}(\boldsymbol{e}_{ik}=\frac{1}{\sqrt{d_e}}) \\
& +\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[j]\middle|\boldsymbol{e}_{ik}=-\frac{1}{\sqrt{d_e}}\right]\Pr^{(R)}(\boldsymbol{e}_{ik}=-\frac{1}{\sqrt{d_e}})
\end{aligned}
$$

Note that by expansion of condition expectation:

$$
\begin{aligned}
& \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[j]\middle|\boldsymbol{e}_{ik}=\frac{1}{\sqrt{d_e}}\right]\Pr^{(R)}(\boldsymbol{e}_{ik}=\frac{1}{\sqrt{d_e}}) \\
= & \sum_{\xi_p,p\neq i}\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[j]\middle|\boldsymbol{e}_{ik}=\frac{1}{\sqrt{d_e}},\boldsymbol{e}_{pk}=\xi_p,p\neq i\right] \\
& \cdot\Pr^{(R)}\left(\boldsymbol{e}_{pk}=\xi_p,p\neq i\middle|\boldsymbol{e}_{ik}=\frac{1}{\sqrt{d_e}}\right)\Pr^{(R)}(\boldsymbol{e}_{ik}=\frac{1}{\sqrt{d_e}}) \qquad (\xi_p \text{ is selected in } \pm\tfrac{1}{\sqrt{d_e}}) \\
= & -\sum_{\xi_p,p\neq i}\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[j]\middle|\boldsymbol{e}_{ik}=-\frac{1}{\sqrt{d_e}},\boldsymbol{e}_{pk}=-\xi_p,p\neq i\right] \\
& \cdot\Pr^{(R)}\left(\boldsymbol{e}_{pk}=\xi_p,p\neq i,\boldsymbol{e}_{ik}=\frac{1}{\sqrt{d_e}}\right) \\
& \qquad (\text{Lemma E.10, } f(\boldsymbol{S}_y(\boldsymbol{E})), \boldsymbol{E}_y^\top\boldsymbol{E}_y \text{ are not changed, and } \boldsymbol{E}_y[j] \text{ is unrelated to } \boldsymbol{e}_{pk}) \\
= & -\sum_{\xi_p,p\neq i}\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[j]\middle|\boldsymbol{e}_{ik}=-\frac{1}{\sqrt{d_e}},\boldsymbol{e}_{pk}=-\xi_p,p\neq i\right] \\
& \cdot\Pr^{(R)}\left(\boldsymbol{e}_{pk}=-\xi_p,p\neq i,\boldsymbol{e}_{ik}=-\frac{1}{\sqrt{d_e}}\right) \qquad (\text{Flipped row have same probability}) \\
= & -\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[j]\middle|\boldsymbol{e}_{ik}=-\frac{1}{\sqrt{d_e}}\right]\Pr^{(R)}(\boldsymbol{e}_{ik}=-\frac{1}{\sqrt{d_e}})
\end{aligned}
$$

Therefore they cancel out and the expectation is 0. □

Lemma E.13 (Diagonal entries of the gradient are the same)**.** *Given a q-subset y, for any function $f(\boldsymbol{S}_y(\boldsymbol{E})) : \mathbb{R}^{d_e\times T}\to\mathbb{R}$, we have for any $i\in[T], j,k\in[d_e]$:*

$$
u_k^\top \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_i\boldsymbol{e}_y^\top\right]u_k^\top = u_j^\top \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_i\boldsymbol{e}_y^\top\right]u_j^\top.
$$

where u_i is the one-hot vector $(\mathbb{1}\{i=1\},\mathbb{1}\{i=2\},...,\mathbb{1}\{i=d_e\})$.

Proof. First, the exact expression of the (k,k)-th entry in the matrix (which is diagonal) is:

$$
u_k^\top \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_i\boldsymbol{e}_y^\top\right]u_k = \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[k]\right]
$$

Consider another PE matrix $\boldsymbol{E}' = \boldsymbol{R}_{kj}\boldsymbol{E}$, which also satisfy the (q,δ)-RIP and have same conditional probability with $\boldsymbol{E}$. Then we have

$$
\begin{aligned}
u_k^\top \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_i\boldsymbol{e}_y^\top\right]u_k &= \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[k]\right] \\
&= \mathbb{E}_{\boldsymbol{E}'}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}'))\boldsymbol{e}'_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y'^\top\boldsymbol{E}'_y)^{-1}\boldsymbol{E}'_y[k]\right] \qquad (\text{Change of variable}) \\
&= \mathbb{E}_{\boldsymbol{E}'}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}'_{ik}\mathbf{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}'_y[k]\right] \qquad (\text{Lemma E.11})
\end{aligned}
$$

$$= \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_{ij}\boldsymbol{1}_q^\top(\boldsymbol{E}_y^\top\boldsymbol{E}_y)^{-1}\boldsymbol{E}_y[j]\right] \qquad \text{(Change back the variable)}$$
$$= u_j^\top\,\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[f(\boldsymbol{S}_y(\boldsymbol{E}))\boldsymbol{e}_i\boldsymbol{e}_y^\top\right]u_j^\top.$$

□

Lemma E.14 (Conditional expectation of the softmax vector)**.** *Given a q-subset y and $\boldsymbol{E}$ satisfies (q,δ)-RIP, for the softmax probability vector $\boldsymbol{S}_y(\boldsymbol{E}) = \mathrm{softmax}(C\boldsymbol{E}^\top\boldsymbol{e}_y)$, we have for any $i,j\in y$ or $i,j\notin y$,*

$$\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{S}_y(i)\right] = \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{S}_y(j)\right]$$

Proof. For $i,j\in y$, we have $\boldsymbol{e}_i^\top\boldsymbol{e}_y = \boldsymbol{e}_j^\top\boldsymbol{e}_y = 1$, so

$$\boldsymbol{S}_y(i) = \frac{e^C}{qe^C + \sum_{i\notin y} e^{C\boldsymbol{e}_i^\top\boldsymbol{e}_y}} = \boldsymbol{S}_y(j)$$

and thus their expectations are the same. For $i,j\notin y$, considered a switched PE matrix $\boldsymbol{E}' = \boldsymbol{R}_{ij}\boldsymbol{E}$. By Lemma E.11, and the probability for $\mathbb{E}$ and $\mathbb{E}'$ are the same:

$$\mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{S}_y(i)\right] = \mathbb{E}_{\boldsymbol{E}}^{(R)}\frac{e^{C\boldsymbol{e}_i^\top\boldsymbol{e}_y}}{qe^C + \sum_{k\notin y} e^{C\boldsymbol{e}_k^\top\boldsymbol{e}_y}}$$
$$= \mathbb{E}_{\boldsymbol{E}}^{(R)}\frac{e^{C\boldsymbol{e}_j^\top\boldsymbol{e}_y}}{qe^C + \sum_{k\notin y} e^{C\boldsymbol{e}_k^\top\boldsymbol{e}_y}} = \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{S}_y(j)\right]$$

□

Lemma E.15 (Estimation for softmax vector with $\boldsymbol{W} = C\begin{bmatrix}\boldsymbol{0}_{d\times d} & \boldsymbol{0}_{d\times d_e}\\ \boldsymbol{0}_{d_e\times d} & \boldsymbol{I}_{d_e}\end{bmatrix}$)**.** *Given a q-subset y and $\boldsymbol{E}$ satisfies (q,δ)-RIP, for the softmax probability vector $\boldsymbol{S}_y(\boldsymbol{E}) = \mathrm{softmax}(C\boldsymbol{E}^\top\boldsymbol{e}_y)$, we have for any $i\in y$:*

$$\frac{1}{q+(T-q)e^{-\frac{1-3\delta}{1-2\delta}C}} \le \boldsymbol{S}_y(i) \le \frac{1}{q+(T-q)e^{-\frac{1-\delta}{1-2\delta}C}},$$
$$\frac{1}{q+(T-q)e^{-\frac{1-3\delta}{1-2\delta}C}} \le \mathbb{E}_{\boldsymbol{E}}^{(R)}\left[\boldsymbol{S}_y(i)\right] \le \frac{1}{q+(T-q)e^{-\frac{1-\delta}{1-2\delta}C}}.$$

Proof. By Lemma A.3, we know $\boldsymbol{e}_i^\top\boldsymbol{e}_y = 1$ for $i\in y$, and $\left|\boldsymbol{e}_i^\top\boldsymbol{e}_y\right| \le \frac{\delta}{1-2\delta}$. Then we know for any C, we have

$$\frac{e^C}{qe^C + (T-q)e^{\frac{\delta}{1-2\delta}C}} \le \boldsymbol{S}_y(i) \le \frac{e^C}{q+(T-q)e^{-\frac{\delta}{1-2\delta}C}},$$
$$\Leftrightarrow \frac{1}{q+(T-q)e^{-\frac{1-3\delta}{1-2\delta}C}} \le \boldsymbol{S}_y(i) \le \frac{1}{q+(T-q)e^{-\frac{1-\delta}{1-2\delta}C}}$$

Since all individual value is bounded within $\left[\frac{1}{q+(T-q)e^{-\frac{1-3\delta}{1-2\delta}C}}, \frac{1}{q+(T-q)e^{-\frac{1-\delta}{1-2\delta}C}}\right]$, the expectation has the same bound. Thus, we complete the proof. □

E.6. From stochastic to fixed: after training

Here, we continue the discussion at the end of Section 3.2: our one-layer transformer with stochastic positional encoding can be 'derandomized' after the two layers $\boldsymbol{W},\boldsymbol{V}$ are already trained. That is, when in the training phase, we use the stochastic positional encoding in the model and run gradient descent till convergence. But after training, we can keep all the parameters unchanged, sample only one near-orthogonal $\boldsymbol{E}$ satisfying (q,δ)-RIP, fix it as the set of PE, and use that model as our trained model. The following corollary of Theorem E.6 can characterize the perfect interpolation ability of the trained transformer.

Corollary E.16. *Under the condition of Theorem E.6, after time* $t \geq \tilde{O}(\frac{T^{\frac{2-2\delta}{1-3\delta}}}{\eta} + \frac{T^2 d}{\eta\epsilon})$*, we keep*

$$\boldsymbol{W}(t) = C(t)\begin{bmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_{d\times d_e} \\ \mathbf{0}_{d_e\times d} & \boldsymbol{I}_{d_e} \end{bmatrix}, \boldsymbol{V}(t) = \alpha(t)\begin{bmatrix} \boldsymbol{I}_d & \mathbf{0}_{d\times d_e} \end{bmatrix}$$

unchanged. Then, consider any arbitrary $\boldsymbol{E} \in \mathbb{R}^{d_e\times T}$ *satisfying* (q,δ)*-RIP, consider the model*

$$f_{\boldsymbol{\theta}}(\boldsymbol{X}, y) = \boldsymbol{V}\boldsymbol{Z}\,\mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{query})$$

we have the non-stochastic training objective Equation (3):

$$\mathcal{L}_{T,q}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{X},y\sim\mathcal{D}_{T,q}}\left[\|\,\mathrm{STS}_q(\boldsymbol{X}, y) - f_{\boldsymbol{\theta}}(\boldsymbol{X}, y)\|_2^2\right] \leq \epsilon.$$

Proof. Use Corollary E.8, we have for any y and $\boldsymbol{E}$ satisfying (q,δ)-RIP be bounded by Lemma E.15:

$$\frac{1}{q + (T-q)e^{-\frac{1-3\delta}{1-2\delta}C}} \leq \boldsymbol{S}_y(i) \leq \frac{1}{q + (T-q)e^{-\frac{1-\delta}{1-2\delta}C}}, \forall i \in y.$$

Since $\boldsymbol{W} = C(t)\begin{bmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_{d\times d_e} \\ \mathbf{0}_{d_e\times d} & \boldsymbol{I}_{d_e} \end{bmatrix}$, all $\boldsymbol{S}_y(i)$ are the same. Then, we denote $s_+ = \boldsymbol{S}_y(i)$ and use the same argument in the proof of Theorem E.6 when verifying $\mathcal{L}(t) \leq \epsilon$ to complete the proof. □

This corollary implies that in practice, once we use the fresh sampled randomized PE in Shen et al. (2023) or Ruoss et al. (2023) on our tasks, it's only necessary to sample the $\boldsymbol{E}$ during training. At the time for evaluation, we can fix one valid PE, preventing us from do more sampling. This avoids 'memorizing' all the random number within the model, and therefore does not violate the memory lower bound condition.

F. Experiment details

In this section, we describe in detail our experimental setup, as well as present additional plots that were not highlighted in the main text. For all of our experiments, we use PyTorch (Paszke et al., 2019), run on NVIDIA RTX A6000s.

Setup. For all of our experiments, we choose $T = 200$ for our context length, and $q = 3$, $d = 5$, and $d_e = 170$. The trainable weight matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ have two initialization case: zero-initialization at $\mathbf{0}$, or randomly initialize with the standard deviation $\sigma^2 = \frac{1}{d+d_e}$.

At each iteration, we sample a batch of $n = 128$ training data points $(\boldsymbol{X}, y)$, and for each OOD task, we fix a set of $n_{\text{test}} = 128$ data points $(\boldsymbol{X}, y)$. For our length generalization tasks, we look at $T_{\text{test}} = \{250, 300, 350, 400\}$, and for our q-generalization tasks, we look at $q_{\text{test}} = \{5, 6, 7, 8\}$. We use the reparameterized model Equation (2):

$$f(\boldsymbol{X}, y) = \boldsymbol{V}\boldsymbol{Z}\,\mathrm{softmax}(\boldsymbol{Z}^\top \boldsymbol{W}\boldsymbol{z}_{\text{query}}),\ \boldsymbol{Z} = [\boldsymbol{X}^\top\ \boldsymbol{E}^\top]^\top.$$

where we use the whole input matrix and the query:

$$[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}] := \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{T-1} & \boldsymbol{x}_T & \boldsymbol{x}_{\text{query}} \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{T-1} & \boldsymbol{e}_T & \boldsymbol{e}_y \end{bmatrix} \in \mathbb{R}^{(d+d_e)\times(T+1)}. \tag{16}$$

Positional encoding sampling. We also enforce a weaker version of the RIP assumption by sampling positional encodings so that they are pairwise near-orthogonal, as defined by some dot product threshold hyperparameter; it turns out that such a choice of positional encodings will satisfy the restricted isometry and orthogonality property, as we would expect for a near-orthogonal matrix. In practice, this is implemented using rejection sampling.

For the experiments where we run with a fixed architecture, the train and test positional encodings are fixed beforehand. For the experiments where stochastic architecture is used, this is simulated at each iteration by sampling T positional encodings for training at each iteration and to be used for the entire $(\boldsymbol{X}, y)$ batch at that iteration, and for validation, a single set of T_{test} positional encodings is sampled for each validation set at each iteration.

Training details. We run simulations on three different settings:

(1) Attending the entire input matrix $[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}]$ as described in Equation (2) and training with SGD (with zero initialization/random initialization of $\boldsymbol{W}$ and $\boldsymbol{V}$).

(2) Attending the entire input matrix $[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}]$ as described in Equation (2) and training with Adam (with zero initialization/random initialization of $\boldsymbol{W}$ and $\boldsymbol{V}$).

(3) An additional experiment run on smaller d, d_e for illustration and training with small random initialization and SGD with annealing.

Fixed vs. stochastic architecture. For the fixed architecture, we sample $T_{\max} = 400$ positional encodings at the start of training. For each iteration of training, we use the first 200 for the fixed architecture, and for each validation task, we use the respective prefix of positional encodings ($T = 250, 300, 350, 400$). For the stochastic architecture, we sample positional encodings at every step for training, as well as for validation. When sampled, we use rejection sampling to ensure the positional encoding matrix satisfy near-orthogonality.

Our experiments demonstrate that even though both fixed and stochastic PE architectures lead to in-distribution population loss of 0, the out-of-distribution validation performances are different, both for length generalization and generalization on unseen q_{test}-subsets. The following sections describe the experiments that were run.

F.1. SGD from zero initialization

When we attend $[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}]$ and train with GD, we run with $\eta = 1$, then annealing to $\eta = 1/3$ at iteration 50000. We run until iteration 100000. The learning rate schedule is to prevent potential instability, for instance via the edge-of-stability phenomenon (Cohen et al., 2021; Li et al., 2022; Damian et al., 2022) or loss spikes in transformer training (Chowdhery et al., 2023; Dehghani et al., 2023; Wortsman et al., 2024).

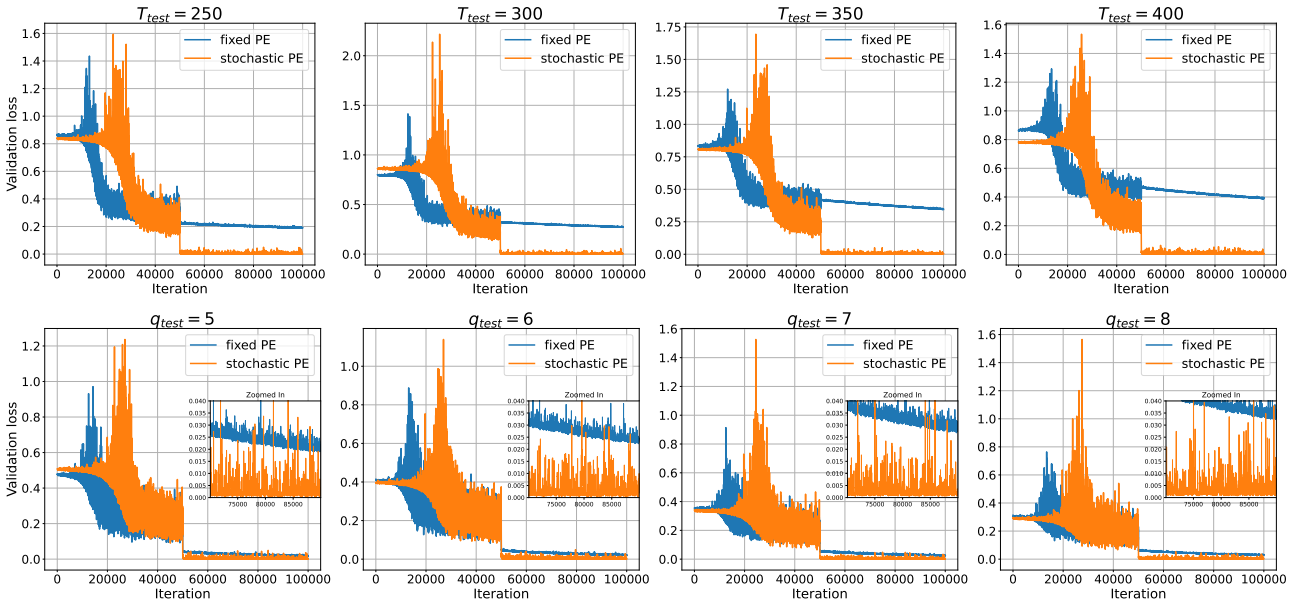

Figure 4. The **length generalization** performance and OOD performance on **unseen q_{test}-subsets**. **Top: Length generalization.** Note that stochastic PE converges to 0 validation loss, whereas a fixed PE is unable to do so; all of the fixed PE end up with validation loss at least 0.15. **Bottom: Generalization to unseen q_{test}-subsets.** Note that while both stochastic and fixed PE can converge to 0 validation loss in the long run, stochastic PE converges slightly quicker, as seen by the zoomed in versions of the plots near the end of training. Additionally, the fixed PE's validation performance gets worse as q_{test} increases.

F.2. Adam from zero initialization

Here, we attend $[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}]$ and train with default Adam settings until iteration 100000.

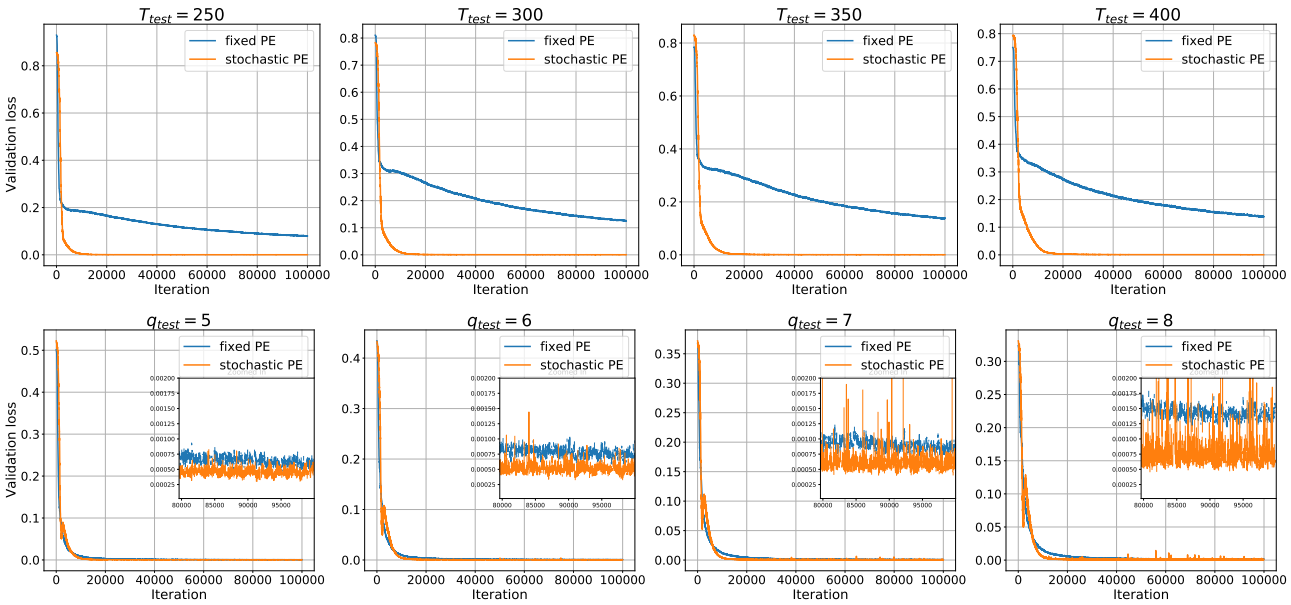

Figure 5. The training trajectory of Adam. The length generalization advantage for stochastic positional encoding is similar to the description of Figure 4. While Adam may allow the validation loss for the q_{test} to converge to 0 in the long run, for all practical purposes related to early stopping, stochastic PE dominates in such OOD performance.

F.3. SGD from random initialization

Similar to Appendix F.1, we attend $[\boldsymbol{Z}, \boldsymbol{z}_{\text{query}}]$ and train with GD, we run with $\eta = 1$, then annealing to $\eta = 1/3$ for iteration 50000 to 100000. We observe similar results as in Appendix F.1.

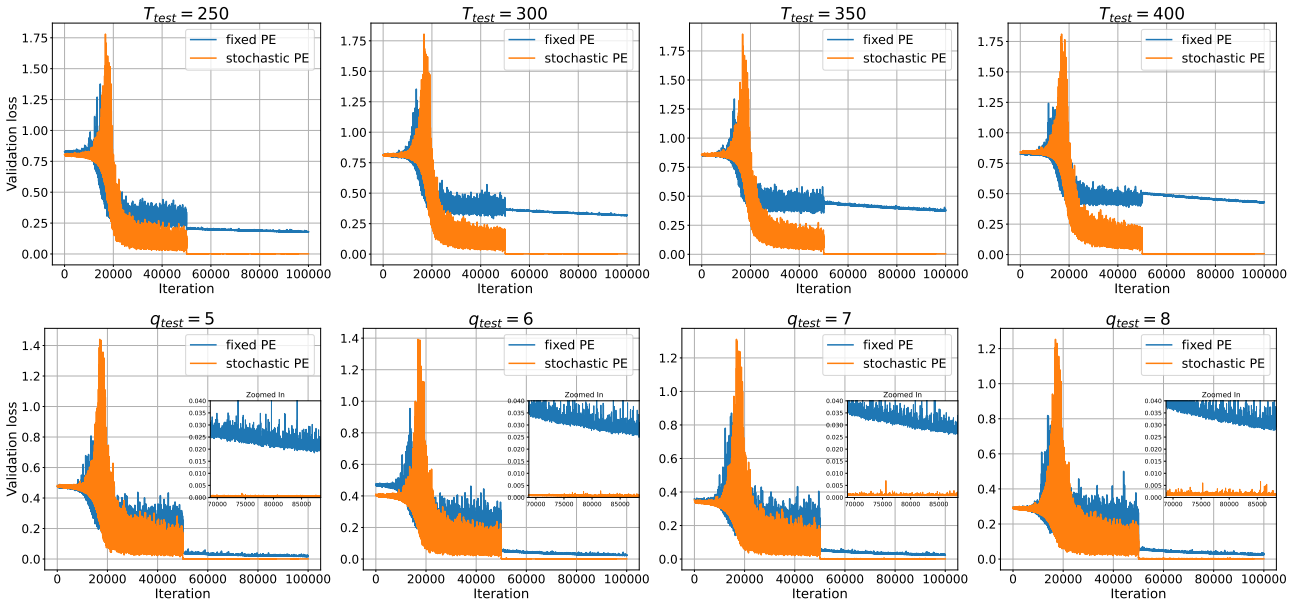

Figure 6. The training trajectory of SGD with random initialization. See description of Figure 4. Note that the overall dynamics are similar to the zero initialization case; as before, the length generalization advantage of stochastic positional encoding is evident. Moreover, stochastic positional encoding achieves better out-of-distribution loss compared to fixed positional encoding.

F.4. Adam from random initialization

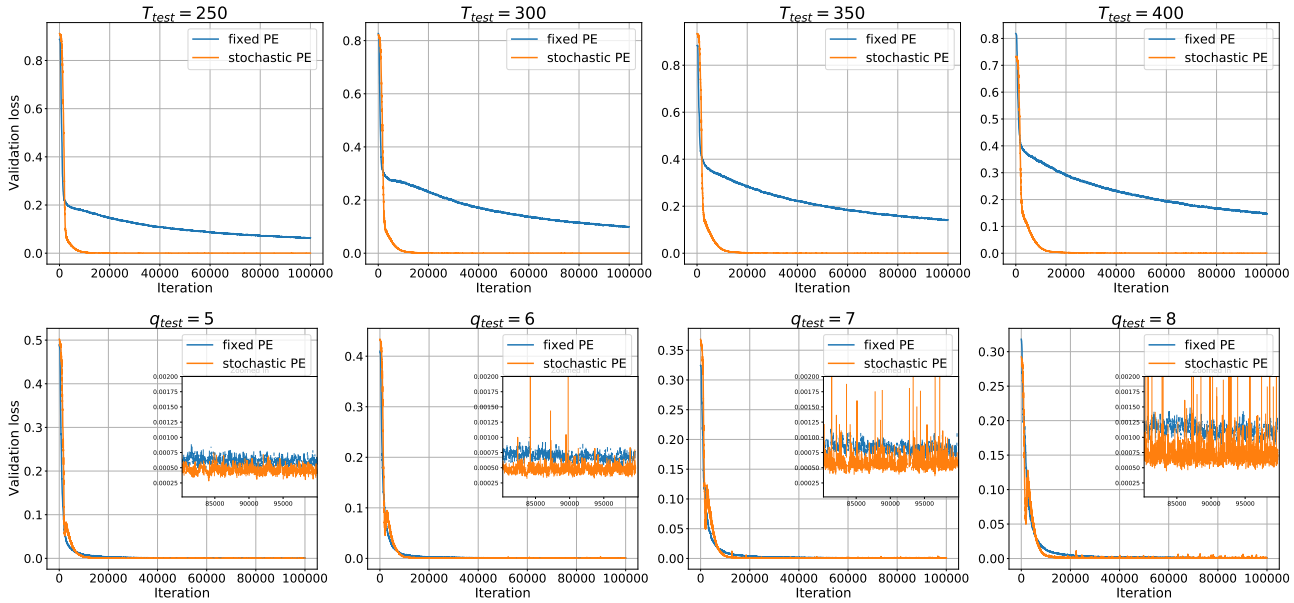

Figure 7. The training trajectory of Adam with random initialization is similar to the zero initialization case. See description of Figure 4. While Adam may allow the validation loss for the q_{test} to converge to 0 in the long run, for all practical purposes related to early stopping, stochastic PE dominates in such OOD performance.

F.5. Additional heat maps in the setting of Figure 3

Here, for the sake of illustrations, we train with $T = 10$, $d = 20$, and $d_e = 20$. In addition, we use small Gaussian initialization with an entrywise standard deviation of $1/\sqrt{d_e}$. The heat maps can be seen as follows; GD eventually converges to the ground truth directions of $\boldsymbol{W}^\star$ and $\boldsymbol{V}^\star$.

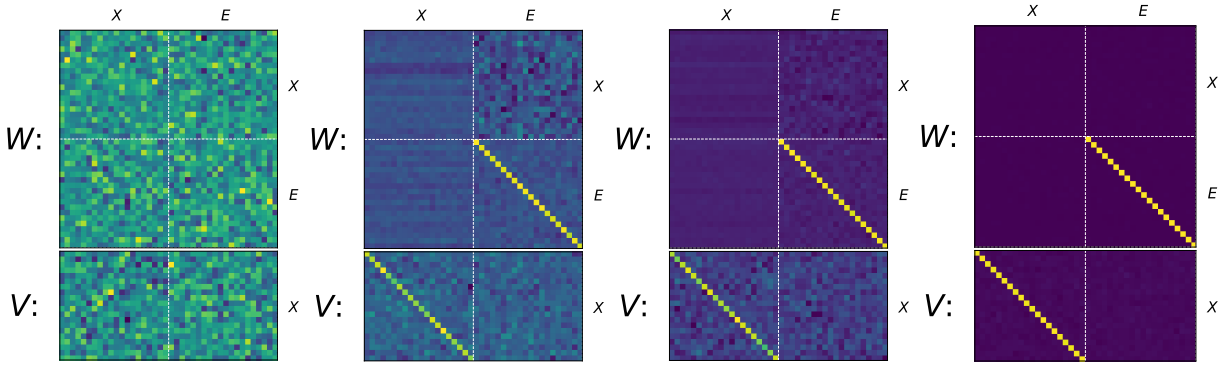

Figure 8. **More figures in interpretable training (iterations $t = 0, 100, 500, 50000$):** For the practical model Equation (2), we present the heat map of the self-attention layer $\boldsymbol{W}$ and the value matrix $\boldsymbol{V}$ after convergence. We initialize $\boldsymbol{W}, \boldsymbol{V}$ randomly at $t = 0$. We can observe that during training, in $\boldsymbol{W}$ only the **sub-block that attends to the positional encodings $\boldsymbol{E}$** gradually converges to identity $\boldsymbol{I}_{d_e}$ direction, while all other entries gradually converge to near 0. Similar phenomenon happens in $\boldsymbol{V}$, only the **sub-block that attends to the input tokens $\boldsymbol{X}$** gradually converges to $\boldsymbol{I}_d$ direction with all other converging to near 0.