Universality, Function Composition, and Algorithm Emulation All In-Context

Anonymous Authors[1]

Abstract

We study the in-context universal approximation and compositional generalization of softmax Transformers. We prove an in-context universality result: a fixed-weight softmax Transformer approximates a broad class of continuous sequence-to-sequence functions. Building on this universality, we establish a composition theorem: by concatenating prompts associated with simple "subprograms," the same fixed Transformer executes their composition, and thereby synthesizes more complex programs on-the-fly. These results support a principled view of prompts as programs and fixed-weight Transformers as program interpreters. Moreover, we provide a concrete mechanism by which GPT-style models both execute and assemble algorithms in context.

1 Introduction

We study the *general-purpose* behavior of foundation models via in-context algorithm emulation and composition in frozen-weight softmax Transformers. We prove two main results: (i) an *in-context universal approximation* theorem for softmax Transformers, and (ii) an *in-context function composition* theorem showing prompt-level program assembly and execution of the composite program in the same frozen model. These results support a principled view of *prompts as programs* and *Transformers as program executors*. They provide a concrete mechanism for foundation models to both execute and assemble algorithms in-context.

Foundation models (large pretrained Transformers) show a striking generality: a single pretrained model does many tasks through prompt changes alone, with no gradient updates (Radford et al., 2019; Brown et al., 2020; Wei et al., 2022; Chowdhery et al., 2023). The community often calls this *in-context learning* (Brown et al., 2020; Min et al., 2022), but the term conflates weight-space learning with inference-time computation. Pretraining learns the model weights. Let $\theta^\star$ denote the frozen weights after pretraining. Inference runs the fixed maps defined by $\theta^\star$ on prompt inputs. Task specification therefore enters through the input context, not through task-specific weight updates. The prompt thus becomes the task interface. There is no explicit learning at all.

To understand better, we present a spectrum of prompt-conditioned computations organized by where the task procedure resides. This spectrum captures the majority of the general-purpose behaviors of foundation models (Figure 1):

- **Parametric recall**[1] sits on one end. The prompt acts as a query. The relevant skill or fact resides in $\theta^\star$. Several statistical analyses formalize this regime as *task identification*: the model matches the prompt to a latent pretrained task and activates its predictor (Wies et al., 2023; Lin and Lee, 2024).
- **Amortized in-context inference**[2] sits in the middle. The prompt supplies examples. The forward pass applies an implicit inference rule encoded in $\theta^\star$. Several analyses formalize this regime as *implicit inference*: the model treats the prompt as a dataset and implements an estimator (e.g., implicit Bayes, least squares, or gradient-style updates) (Xie et al., 2022; Garg et al., 2022; Akyürek et al., 2023; von Oswald et al., 2023; Bai et al., 2023; Li et al., 2023).
- **Prompt-programmed execution**[3] sits on the other end.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]*Practical use cases:* closed-book factual Q&A / knowledge lookup (entities, dates, definitions), commonsense completions, generic completion and editing (paragraph continuation, rewriting, grammar/style fixes), and high-frequency pattern completion (e.g., boilerplate code snippets, common API idioms).

[2]*Practical use cases:* few-shot classification/labeling (sentiment/intent, topic tags, NER), few-shot extraction into a target schema (forms → JSON), learning domain-specific rubric from scored examples (grading/triage), light calibration/regression from demonstrations (ratings, prioritization), and "learn-the-rule-from-examples" reasoning (the prompt defines the mapping implicitly).

[3]*Practical use cases:* multi-step reasoning with explicit scratchpads (math/logic, stepwise verification), planning and decomposition (project plans, study plans, experimental protocols), constrained generation via explicit templates/DSLs (tables, JSON, SQL, checklists), iterative refinement loops (draft → critique → revise), and tool-oriented workflows where the prompt defines an execution protocol (retrieve → compute → check → report).

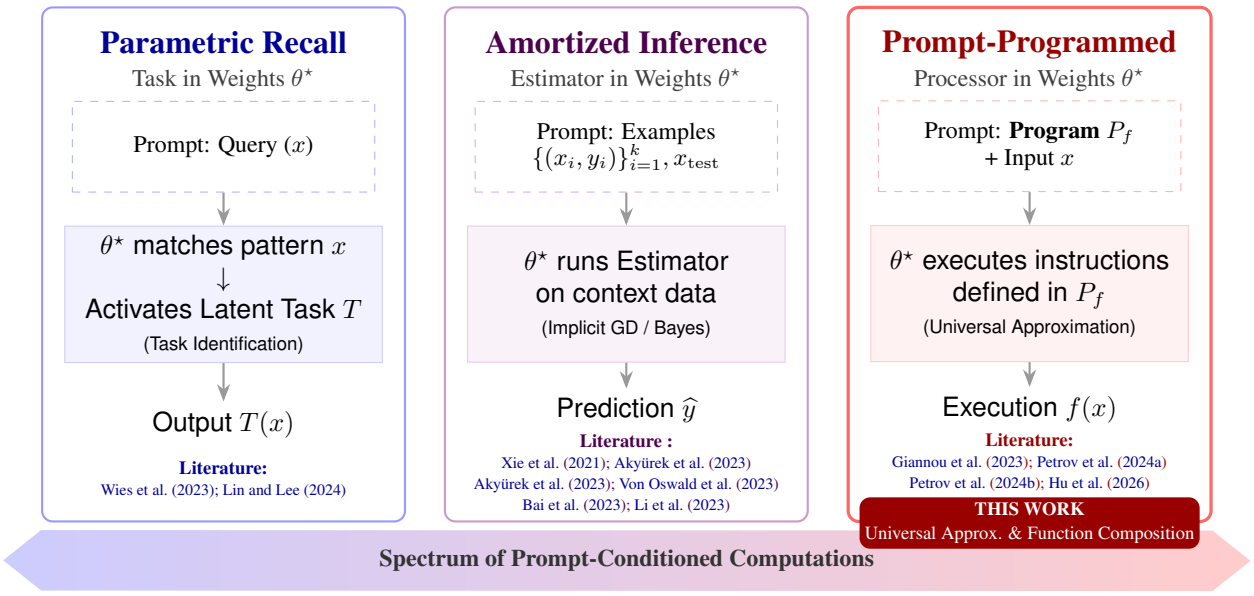

Figure 1. **The Spectrum of Prompt-Conditioned Computations.** We organize general-purpose behaviors by where the task procedure resides. (1) **Parametric Recall** and (2) **Amortized Inference** (blue/purple) rely on fixed tasks or estimators encoded in the frozen weights $\theta^\star$, the prompt serves only as a query or dataset. In contrast, (3) **Prompt-Programmed Execution** (red) encodes the algorithm *in the prompt* (P_f), treating $\theta^\star$ as a universal interpreter. We address the theoretical gap in (3) by providing constructive proofs for in-context universal approximation and function composition.

The prompt specifies a program: a procedure and a context layout for inputs, tokens that store intermediate results, and outputs. Several programmability constructions formalize this regime as *prompt-as-program* (Giannou et al., 2023; Qiu et al., 2025; Hu et al., 2026).

The ultimate goal is to understand the full spectrum and explain the general-purpose behavior of transformer foundation models. However, most existing ICL theory only targets the first two regimes. It explains how a frozen model retrieves pretrained knowledge or applies a fixed inference rule to in-context examples. These regimes yield generality *only within* a task family because they keep the procedure fixed in $\theta^\star$ and vary only the queried content or the in-context data. This leaves the opposite end under-specified. They do not explain prompt changes that specify *new* procedures under one fixed $\theta^\star$, or prompt-level modular assembly of such procedures.

Theory still falls short of explaining this prompt-programmable regime. A satisfactory theory of this end should meet three requirements:

- It should use standard softmax attention.
- It should use one fixed set of model weights across tasks.
- It should provide a constructive and reusable prompt interface that programs procedures and supports composition.

Existing theory satisfies only subsets. Several statistical analyses isolate the *parametric recall* end. They formalize prompting as *task identification* under a task distribution, i.e., similarity-based pattern matching over latent tasks that activates a pretrained predictor (Wies et al., 2023; Lin and Lee, 2024). Other analyses target the *amortized in-context inference* regime. They treat the prompt as examples and characterize the implicit estimator executed by the forward pass, with generalization or stability guarantees within a task family (Xie et al., 2022; Garg et al., 2022; Li et al., 2023; Raventós et al., 2023). Complementary constructions expose specific inference rules inside the forward pass (e.g., least squares or gradient-style updates) (Akyürek et al., 2023; von Oswald et al., 2023; Bai et al., 2023). These results keep the procedure template in $\theta^\star$ and vary the query or the in-context data. They leave the pure *prompt-programmed* end under-specified.

Recent work makes progress toward the prompt-programmed regime and formalizes prompt-as-program behavior (Wang et al., 2023; Giannou et al., 2023; Hu et al., 2024a; Qiu et al., 2025). However, they all take an existential form and leave the constructions implicit[4]. They do not yield a constructive, reusable prompt compiler or modular composition guarantees for one frozen softmax Transformer.

[4]The only exception is Giannou et al. (2023). It gives an explicit construction and establishes a Turing-completeness-style programmability result. Differed from ours, its prompt-programmability comes from building a Turning complete One-Instruction Set Computer (OISC) with Transformer blocks to compute SUBLEQ programs. Such construction require heavy architecture (13 layers Transformer block) and looped operation.

Moreover, a shared limitation cuts across these lines: many proofs replace softmax attention with tractable surrogates (e.g., linear, ReLU or hardmax attention) (von Oswald et al., 2023; Bai et al., 2023; Ahn et al., 2023; Qiu et al., 2025). Collectively, these gaps leave open systematic prompt programming of a frozen softmax Transformer for diverse computations and their modular assembly. We target this gap in this paper.

We now define our scope. Specifically, we study in-context algorithm emulation and prompt-level composition of softmax Transformer models. Let T_θ denote the sequence-to-sequence map implemented by a Transformer with weights θ. We use *function*, *algorithm*, and *program* to refer to sequence-to-sequence procedures. A *program* is a procedure with intermediate results stored and reused across steps.

- We formalize *in-context algorithm emulation* as follows: for a target procedure f, a prompt P_f induces the frozen model $T_{\theta^\star}$ to implement f on the non-prompt tokens, using designated scratchpad tokens for intermediate results.
- We formalize *in-context composition* as a prompt-level construction that maps prompts for subprocedures (e.g., P_f, P_g) to a prompt for the composite procedure (e.g., $P_{g\circ f}$), so that the same frozen model executes the composite program with controlled error.

Contributions. Our contributions are three-fold:

- **In-Context Universal Approximation.** We prove that a frozen softmax Transformer acts as an in-context universal approximator for continuous sequence-to-sequence functions f. Specifically, we show that for any f on a compact domain, there exists a specific prompt P_f that induces the frozen model to approximate f to arbitrary precision (Theorem 3.1 and Theorem 3.2). This establishes that the softmax Transformer is sufficient for a general-purpose computer.
- **General Prompt-Programmable Algorithm Emulation.** We provide an algorithm emulation framework that maps target function parameters into a prompt. Unlike existential arguments, we construct the weight-encoding matrices. These encodings allow the Transformer to instantiate arbitrary feed-forward networks (Lemma 3.1) and multi-head attention (Lemma 3.4) from the context. This grounds the view of the prompt as machine code and the Transformer as an interpreter.
- **In-Context Function Composition.** We establish in-context function composition, proving the frozen model executes sequential subroutines rather than just parallel tasks. We analyze two regimes: *single completion* (Theorem 4.1), which executes m subroutines in one pass (requiring depth linear in m), and *re-prompting* (Theorem 4.2), which supports arbitrary function composition with a fixed-depth model via iterative interaction.

Together, these results extend the scope of provable general-purpose in-context learning. We move beyond the "attention-implementable" algorithms of (Hu et al., 2026) to arbitrary computable functions, and beyond single-task prompting to multi-step task composition. By showing how a frozen Transformer serve as a universal and composable in-context learner, our work takes a step closer to explaining the observed general-purpose ability of foundation models. Ultimately, this line of inquiry lays the theoretical groundwork for understanding foundation models not just as pattern-matching engines, but as versatile in-context computers capable of implementing an unlimited range of algorithms via prompting.

Organizations. Section 2 covers preliminaries. Section 3 demonstrates how the *Prompt-Program Execution* perspective leads to the in-context universality. Section 4 presents two results of function composition. Section 5 summarizes the theoretical results and contextualizes the contributions to in-context learning in more detail. We include the related work in Section A and detailed proofs in Section D.

Notations. We use $\mathbb{N}^+$ to denote the set of positive integers, and for $n \in \mathbb{N}^+$, we let $[n] := \{1, 2, \ldots, n\}$. We use lowercase letters for vectors, and uppercase letters for matrices. For $X \in \mathbb{R}^{d\times n}$, $X_{i,:}$ denotes the i-th row, and $X_{:,i:j}$ denotes the sub-matrix consisting of columns i through j, where $i \le j$ and $i, j \in [n]$. We write $\mathbb{1}_d \in \mathbb{R}^d$ for the all-ones vector. For $X \in \mathbb{R}^{d\times n}$, we define $\|X\|_\infty := \max_{i\in[d],j\in[n]} |X_{ij}|$, and $\|X\|_p$ denotes the entry-wise ℓ_p norm.

2 Preliminaries

In Section 2.1, we lay out the ideas we built on.

2.1 Functions, Algorithms, and Programs

A function f specifies input-output behavior, an algorithm specifies a procedure that realizes that behavior, program provides a finite description of that procedure. In this paper, we focus on the program consist of algorithms and dataset. We state guarantees at the level of functions because our metrics compare outputs. We treat the prompt itself as the program text: after fixing weights $\theta^\star$, we define the semantics of a prompt P by $f_P(X) \triangleq T_{\theta^\star}(P, X)$, i.e., the output of the frozen Transformer on the concatenated sequence (P, X). Our constructions build P_f by compiling a finite description of the target computation into tokens, so the prompt carries parameters and wiring rather than serving as an unconstrained continuous parameter vector. Our composition theorem then supplies a modular assembly rule: it constructs a prompt for $g \circ f$ from reusable pieces P_f and P_g, and it controls both prompt-length growth and error propagation.

2.2 Transformers

This section introduces the Transformer architecture and the distance metric between two sequence-to-sequence maps. Part of our notation follows (Kajitsuka and Sato, 2023).

Definition 2.1 (Attention Layer)**.** *For $x = [x_1, x_2, \ldots, x_d] \in \mathbb{R}^d$, we define $\sigma_\beta(\cdot)$ as the softmax function with inverse temperature β*

$$\sigma_\beta(x) := [\frac{\exp(\beta x_1)}{\sum_{j=1}^d \exp(\beta x_j)}, \cdots, \frac{\exp(\beta x_d)}{\sum_{j=1}^d \exp(\beta x_j)}]. \quad (1)$$

Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. We define an H-head self-attention layer $\mathrm{Attn}(\cdot)$ as

$$\mathrm{Attn}(X) := \sum_{h=1}^{H} W_O^h \cdot W_V^h X \cdot \sigma_\beta((W_K^h X)^\top W_Q^h X),$$

where σ_β applies column-wise softmax as defined in (1), $W_O^h \in \mathbb{R}^{d_O \times d_V}$, $W_V^h \in \mathbb{R}^{d_V \times d}$, and $W_K^h, W_Q^h \in \mathbb{R}^{d_h \times d}$ are the weight matrices. For an l-layer multi-head attention module, we denote it as Attn_l. Finally, we denote the attention layer with a residual connection as $\mathrm{Attn}^{\mathrm{res}}(X) := X + \mathrm{Attn}(X)$.

Definition 2.2 (Feed-Forward Layer)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. Then, we define a two-layer feed-forward neural network as*

$$\mathrm{FF}(X) := W_2 \cdot \mathrm{ReLU}(W_1 X + b_1 \mathbb{1}_n^\top) + b_2 \mathbb{1}_n^\top,$$

where $W_1 \in \mathbb{R}^{r\times d}$, $W_2 \in \mathbb{R}^{d\times r}$ are weight matrices, and $b_1 \in \mathbb{R}^r$ and $b_2 \in \mathbb{R}^d$ are biases. We also define some notations for later convenience. For $j \in [r]$ and $k \in [d]$, define the augmented row vectors $\widetilde{w}_{1,j} := [(W_1)_{j,:}^\top; (b_1)_j] \in \mathbb{R}^{d+1}$ and $\widetilde{w}_{2,k} := [(W_2)_{k,:}^\top; (b_2)_k] \in \mathbb{R}^{r+1}$, and define

$$\mathrm{Enc}(\widetilde{w}) := \begin{bmatrix} 0\cdot\widetilde{w} & 1\cdot\widetilde{w} & \cdots & (n-1)\cdot\widetilde{w} \\ \widetilde{w} & \widetilde{w} & \cdots & \widetilde{w} \end{bmatrix}. \quad (2)$$

Let $\widetilde{W}_{1,j} := \mathrm{Enc}(\widetilde{w}_{1,j})$ and $\widetilde{W}_{2,k} := \mathrm{Enc}(\widetilde{w}_{2,k})$.

We then restate the distance metric between two sequence-to-sequence functions from (Kajitsuka and Sato, 2023).

Definition 2.3 (Distance between Sequence-to-Sequence Functions)**.** *Let $\|\cdot\|_p$ $(1 \le p < \infty)$ be the element-wise ℓ_p matrix norm. We define the distance between two functions $f_1, f_2 : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ by*

$$\mathrm{d}_p(f_1, f_2) := (\int \|f_1(X) - f_2(X)\|_p^p \mathrm{d}X)^{\frac{1}{p}}.$$

2.3 Two Routes of Transformer UAP

In this work, we establish two ways to achieve constructive in-context UAP for Transformers under minimal assumptions on data or target function:

- **Contextual Mapping.** Prior works (Yun et al., 2020a; Kajitsuka and Sato, 2023) establish the theoretical foundation for attention as contextual mappings. Kajitsuka and Sato (2023) shows that one-layer hardmax attention does not realize such mappings, while one-layer softmax attention does.
- **Attention-as-Selector.** In contrast, Hu et al. (2025) proves the softmax-attention UAP using an interpolation view. They treat softmax as a hardmax selector over the interpolation points, allowing the model to approximate functions without feed-forward networks.

Presenting both routes demonstrates that our results do not hinge on a single temperature regime of softmax. There exist several transformer universal approximation results. However, they impose stronger assumptions beyond the continuous target function we consider here (Takakura and Suzuki, 2023; Jiang and Li, 2024). Takakura and Suzuki (2023) further assumes infinite sequences and a shift-invariant input distribution. See Section A for a detailed discussion.

3 In-Context Universal Approximation

This section proves our in-context universality result: a frozen softmax attention module approximates any continuous sequence-to-sequence function by changing the input prompt. We present two complementary constructions. Section 3.1 follows a *contextual mapping perspective*: by invoking the transformer universal approximation ability (Kajitsuka and Sato, 2023), we approximate the two-layer transformer in context; hence, we obtain an in-context approximation of any continuous function. Section 3.2 follows a *selection perspective*: we approximate an attention-only universal approximator (Hu et al., 2025) in context, then obtain an in-context approximation for arbitrary continuous sequence-to-sequence functions.

3.1 Attention-as-Contextual-Mapping Route

To approximate the target function f, we approximate the surrogate Transformer structure $\mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1$ (Kajitsuka and Sato, 2023, Proposition 1, or reference to Theorem C.1). We construct a six-layer attention-only module that processes the data X alongside a parameter encoding of $\mathrm{FF}_1, \mathrm{FF}_2$ and $\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}$.

While recent work show that attention approximate fixed FFNs (Hu et al., 2025), in-context learning requires a more dynamic capability. The model must not merely approximate one function, but must allow the prompt to define which function is applied. In the following lemma, we prove that a frozen attention layer approximates a FFN in context. By encoding the weights W_{FF} directly into the prompt, the attention mechanism instantiates the specific FFN defined by those weights on the fly. We also allow an auxiliary block Z to flow through module unchanged, since later composition requires persistent scratchpad tokens.

Lemma 3.1 (Attention Approximates Feed-Forward Neural Network In Context)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence, and let W_{FF} be the weight of a feed-forward neural network* FF*. We assume $\|X\|_\infty \le B_X$ and $\|W_{\mathrm{FF}}\|_\infty \le B_W$ Let $Z \in \mathbb{R}^{d_Z\times n}$ be a matrix containing an identity matrix I_n with $n \le d_Z$. Assume $\|Z\|_\infty \le B_Z$ for some constant $B_Z > 0$. Then, for every $\epsilon_1 > 0$, there exists a 2-layer self-attention* Attn_2 *such that*

$$\|\mathrm{Attn}_2(\begin{bmatrix} X \\ W_{\mathrm{FF}} \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{FF}(X) \\ Z \end{bmatrix}\|_\infty \le \epsilon_1,$$

where W_{FF} is in the form of $W_{\mathrm{FF}} := [\widetilde{W}_{1,1}^\top, \cdots, \widetilde{W}_{1,r}^\top, \widetilde{W}_{2,1}^\top, \cdots, \widetilde{W}_{2,d}^\top]^\top \in \mathbb{R}^{2(2dr+d+r)\times n}$ with each $\widetilde{W}$ defined in Definition 2.2.

Proof Sketch. To approximate the first hidden layer of the ReLU neural network in context, we use a one-layer multi-head attention model constructed from Corollary C.4.1. Moreover, we introduce an extra attention head to bring the portion of the input that should remain unchanged to the next layer. Please see Section D.1 for a detailed proof. □

Note that in the above lemma we allow an auxiliary block Z to flow through module unchanged, since later composition requires persistent scratchpad tokens. It is straightforward to have another version of lemma without an auxiliary block.

Lemma 3.2 (Attention Approximates Feed-Forward Neural Network In Context)**.** *Under the same setting of Lemma 3.1, for every $\epsilon_2 > 0$, there exists a 2-layer multi-head self-attention layer* Attn_2 *such that*

$$\|\mathrm{Attn}_2(\begin{bmatrix} X \\ W_{\mathrm{FF}} \end{bmatrix}) - \mathrm{FF}(X)\|_\infty \le \epsilon_2.$$

Proof. Please see Section D.2 for a detailed proof. □

Next, we emulate the single-head attention with a residual component $\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}$. We first introduce the input format modified from Hu et al. (2026); see also Definition C.2.

Definition 3.1 (Vectorization)**.** *For any matrix $W \in \mathbb{R}^{p\times s}$, we define $\underline{W} := \mathrm{vec}(W) \in \mathbb{R}^{ps}$ such that $\underline{W}_{(i-1)s+j} = W_{ij}$ for all $i \in [p]$ and $j \in [s]$.*

Definition 3.1 equals stacking the row vectors of a matrix into a single column vector.

Definition 3.2 (Input Prompt for In-Context Emulation of Multi-Head Attention)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. Let $W_K^h, W_Q^h \in \mathbb{R}^{d_h\times d}$ and $W_V^h \in \mathbb{R}^{d_V\times d}$ be the weight matrices of the h-th head in the target H-head attention. We define the concatenation $w_h := [\underline{W}_K^h; \underline{W}_Q^h; \underline{W}_V^h] \in \mathbb{R}^{d(2d_h+d_V)}$ and*

$$W_h := \mathrm{Enc}(w_h) \in \mathbb{R}^{2d(2d_h+d_V)\times n},$$

where $\mathrm{Enc}(\cdot)$ is defined in (2). Then, the input prompt for the in-context emulation of a multi-head attention specified by W_K^h, W_Q^h, W_V^h is

$$X_p^{\mathrm{m}} := \begin{bmatrix} X^\top & W_1^\top & \cdots & W_H^\top & I_n \end{bmatrix}^\top.$$

Now we show that a 2-layer attention emulates single-head attention with a residual connection in-context. We again retain an arbitrary auxiliary block Z unchanged.

Lemma 3.3 (In-Context Emulation of Single-Head Attention with a Residual Connection and a Flow-Through Component)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence and $Z \in \mathbb{R}^{d_Z\times n}$. Let $X_p \in \mathbb{R}^{(d+2d(2d_h+d)+n)\times n}$ be the input prompt from Definition 3.2 with $H = 1$, and set the free parameter $d_V \to d$. Assume $\|X\|_\infty \le B_X, \|Z\|_\infty \le B_Z$ and $\|W_K^h X\|_\infty, \|W_Q^h X\|_\infty, \|W_V^h X\|_\infty \le B_{KQV}$ for some $B_X, B_Z, B_{KQV} > 0$. Then, for any $\epsilon_3 > 0$, there exists a two-layer multi-head attention network* Attn_2 *such that*

$$\|\mathrm{Attn}_2(\begin{bmatrix} X_p \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}(X) \\ Z \end{bmatrix}\|_\infty \le \epsilon_3.$$

Proof Sketch. To propagate Z through the attention module, we add an auxiliary head in each layer and exploit the I_n in X_p. Concretely, we choose the value projection to read Z, and the key/query projections to read I_n. With an appropriate choice of β, the resulting attention weights concentrate on the diagonal, so this head propagates Z. Please see Section D.3 for a detailed proof. □

We now assemble the complete six-layer module by stacking Lemma 3.1, Lemma 3.3, and Lemma 3.2. This yields an attention-only, in-context universal approximator for continuous functions.

Theorem 3.1 (In-Context Universal Approximation by Transformer)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. Let $W_{\mathrm{FF}_1}, W_{\mathrm{FF}_2}$ be the weight encodings for two feed-forward neural networks $\mathrm{FF}_1, \mathrm{FF}_2$ as in Lemma 3.1. Let $W_{\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}}$ be the weight encoding for a single-head self-attention, as in Definition 3.2 with $H = 1$. Define $W := \begin{bmatrix} W_{\mathrm{FF}_1}^\top & W_{\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}}^\top & I_n & W_{\mathrm{FF}_2}^\top \end{bmatrix}^\top$. Let f be any continuous function defined on a compact domain $\mathcal{C} \subset \mathbb{R}^{d\times n}$, for any $\epsilon > 0$, there exists a six-layer attention* Attn_6 *such that*

$$\mathrm{d}_p(\mathrm{Attn}_6(\begin{bmatrix} X \\ W \end{bmatrix}), f(X)) \le \epsilon.$$

Proof. Please see Section D.4 for a detailed proof. □

Remark 3.1 (Explicit Construction of W for Function f)**.** *As in the introduction, a satisfactory theory of the prompt-programmable regime should provide a constructive and reusable prompt interface. Theorem 3.1 gives out the specific structure of the prompt. The parameters of any target*

function is serialized into token embeddings (W) that the frozen attention reads and executes. We further give the explicit construction of W that depends on function f at Remark D.1, based on (Hu et al., 2024b, Theorem G.1, G.2).

3.2 Attention-as-Selector Route

In this section, we refine our strategy by removing the dependence on the feed-forward emulator. We aim to approximate a sequence-to-sequence function f by emulating a multi-head, pure attention network T from Hu et al. (2025, Theorem G.1, or reference to Theorem C.2).

For that purpose, we extend our single-head emulator (Lemma 3.3) to the multi-head setting by encoding a multi-head attention layer's parameters into the prompt.

Lemma 3.4 (In-Context Emulation of Multi-Head Attention)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. Let $\mathrm{Attn} : \mathbb{R}^{d\times n} \to \mathbb{R}^{d_V\times n}$ be an H-head attention specified by $W_K^h, W_Q^h \in \mathbb{R}^{d_h\times d}$ and $W_V^h \in \mathbb{R}^{d_V\times d}$. Assume $\|W_K^h X\|_\infty, \|W_Q^h X\|_\infty, \|W_V^h X\|_\infty \le B_{KQV}$ for some $B_{KQV} > 0$. Then, for any $\epsilon_4 > 0$, there exists a two-layer attention network Attn_2 such that*

$$\|\mathrm{Attn}_2(X_p^{\mathrm{m}}) - \mathrm{Attn}(X)\|_\infty \le \epsilon_4,$$

where $X_p^{\mathrm{m}} = [X^\top, W_1^\top, \cdots, W_H^\top, I_n]^\top$ (Definition 3.2).

Proof. Please see Section D.5 for a detailed proof. □

Corollary 3.1.1 (In-Context Emulation of Multi-Head Attention with Flow-Through Component)**.** *Let $Z \in \mathbb{R}^{d_Z\times n}$ be any matrix satisfying $\|Z\|_\infty \le B_Z$ for some constant $B_Z > 0$. Under the same setting of Lemma 3.4, for any $\epsilon_5 > 0$, there exists a two-layer attention Attn_2 such that*

$$\|\mathrm{Attn}_2(\begin{bmatrix} X_p^{\mathrm{m}} \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}(X) \\ Z \end{bmatrix}\|_\infty \le \epsilon_5.$$

Proof. Please see Section D.6 for a detailed proof. □

We now construct the full emulator. The target surrogate T has four layers. We emulate the first three layers by using Corollary 3.1.1 (to pass weights forward) and the fourth layer by Lemma 3.4.

Definition 3.3 (Input Prompt for In-Context UAP via Selector Route)**.** *Consider the four-layer attention network T, for layer $i \in [4]$, let H_i be the number of heads, and for each head $j \in [H_i]$, let W_j^i be the weight-encoding matrix of the j-th head in the i-th attention layer, constructed as W_h in Definition 3.2. Also, let X_a be the augmented input*

$$X_a := \begin{bmatrix} X & 0_{d\times 1} \\ I_n & 0_{n\times 1} \\ 0_{1\times n} & 1 \end{bmatrix} \in \mathbb{R}^{(d+n+1)\times(n+1)}.$$

Finally, we define the full prompt as the vertical concatenation of X_a and four layer-wise weight blocks:

$$\widetilde{X}_p := \begin{bmatrix} X_{\mathrm{a}}^\top & W_T^{1\top} & W_T^{2\top} & W_T^{3\top} & W_T^{4\top} \end{bmatrix}^\top,$$

wehre $W_T^i := [(W_1^i)^\top, \cdots, (W_{H_i}^i)^\top, I_{n+1}]^\top$.

Finally, the next theorem states that a fixed 8-layer attention module is an in-context universal approximator for any continuous sequence-to-sequence function.

Theorem 3.2 (In-Context Universal Approximation by Attention-Only Transformer)**.** *Let $\mathcal{C} \subset \mathbb{R}^{d\times n}$ be a compact domain of input sequences, and $f : \mathcal{C} \to \mathbb{R}^{d\times n}$ be a continuous sequence-to-sequence function. Let X be the input sequence. Then, for any $\epsilon > 0$, there exists an eight-layer multi-head attention network Attn_8 such that*

$$\mathrm{d}_p(\mathrm{Attn}_8(\widetilde{X}_p)_{:,1:n}, f(X)) < \epsilon,$$

where $\widetilde{X}_p$ follows Definition 3.3.

Proof. Please see Section D.7 for a detailed proof. □

Remark 3.2 (Explicit Construction of W_T^i for Function f)**.** *Again, we recall the reusable and constructive prompt interface proposed in the introduction. Theorem 3.2 provides the reusable prompt format W_T^i that encodes the weights of the attention network T. At Remark D.2, given a target function f, we restate the explicit weight construction provided by Hu et al. (2025).*

3.3 Discussion

Comparison of Two Routes Results. While both routes achieve in-context universality, they offer distinct theoretical insights. The first route relies on the *contextual mapping* view, demonstrating that attention emulate the standard Transformer architecture (FFN + Attention) in-context. The second route adopts the *interpolation* view, proving that even a pure attention architecture without FFNs suffices for in-context universality.

Comparison with Existing Works. Li et al. (2025) study in-context universal approximation in the *Amortized In-Context Inference* spectrum. We study *Prompt-Programmed Execution*. Our setting does not infer a function from examples, and this setting matches the zero-shot mode of LLM: the prompt contains instructions or algorithmic steps but no input-output examples. Secondly, they focus on scalar prediction $\mathbb{R}^d \to \mathbb{R}$, while we study sequence-to-sequence functions. Another line of work (Petrov et al., 2024b) proves that pretrained Transformers approximate smooth continuous functions on a hypersphere via prefix-tuning. In contrast, we only assume continuous functions on compact domains. Furthermore, their prompt is an interpolation table, not a reusable procedure that leads to function composition, as later shown in Section 4.

4 In-Context Function Composition

We extend the in-context universal approximation theory to prove the in-context function composition of attention-only models. Recent works like (Xiong et al., 2024) characterize Transformers as parallel processors executing independent tasks. Algorithmic reasoning, however, demands sequential composition inherently, passing subroutine outputs to subsequent inputs.

We move beyond the parallel regime to establish in-context composition. Specifically, we consider two practical scenarios for the day-to-day use of language models:

- **Single completion:** prompting a language model with multiple sub-tasks in a single prompt, and
- **Re-prompting:** issuing a new prompt after each sub-task, corresponding to a multi-turn interaction with the model.

These results provide clear mechanisms of the fixed weight transformer generalization ability on various complex tasks.

4.1 Single Completion

We first formalize a "flow-through" version of Theorem 3.1, which preserves an arbitrary flow-through block Z we want it to be unchanged. This is the mechanism that preserves the scratchpad tokens.

Lemma 4.1 (In-Context Universal Approximation with Flow-Through Component)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. Let W be the weight encoding for a sequence-to-sequence function f as in Theorem 3.1. Let $Z \in \mathbb{R}^{d_Z\times n}$ be an arbitrary matrix satisfying $\|Z\|_\infty \le B_Z$ for $B_Z > 0$. Then, for any $\epsilon > 0$, there exists a six-layer multi-head attention network Attn_6 such that*

$$\mathrm{d}_p(\mathrm{Attn}_6(\begin{bmatrix} X \\ W \\ Z \end{bmatrix}), \begin{bmatrix} f(X) \\ Z \end{bmatrix}) \le \epsilon$$

Proof. Please see Section D.8 for a detailed proof. □

We now compose m subroutines $f_1, \ldots, f_m$ by stacking the six-layer emulator for m times. Intuitively, for $i \in [m]$, the i-th stage reads W_i from the prompt, applies f_i to the current data tokens, and forwards all remaining tokens for later stages.

Theorem 4.1 (In-Context Composition: Single Completion)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. Let $\mathcal{F}$ be a set of continuous functions defined on a compact domain*

$$\mathcal{F} \coloneqq \{f \mid f : \mathcal{C} \to \mathcal{C},\ \mathcal{C} \subset \mathbb{R}^{d\times n}\}.$$

Consider a compositional function: $f_m \circ f_{m-1} \circ \cdots \circ f_1(X)$, where $m \in \mathbb{N}^+$ and $f_i \in \mathcal{F}$. Let W_i be the weight encoding for f_i as in Theorem 3.1. Then, for any $\epsilon > 0$, there exists a $6m$-layer attention network Attn_{6m} such that

$$\mathrm{d}_p(\mathrm{Attn}_{6m}(\begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}), f_m \circ f_{m-1} \circ \cdots \circ f_1(X)) \le \epsilon.$$

Proof. Please see Section D.9 for a detailed proof. □

The depth grows linearly with the number of composed subroutines ($6m$ layers). This result implies that a fixed weight and depth transformer has limited expressive power for long programs in one shot.

4.2 Re-Prompting

We now keep the module depth fixed and instead re-run the same frozen Attn_6 after each sub-task. This mimics multi-turn interactions with language models. Each call supplies a new subroutine prompt W_i and uses the previous output as the next input.

Theorem 4.2 (In-Context Composition: Re-Prompting)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. Let Attn_6 be the frozen attention module from Theorem 3.1. Let $\mathcal{F}$ be a set of continuous functions defined on a compact domain*

$$\mathcal{F} \coloneqq \{f \mid f : \mathcal{C} \to \mathcal{C},\ \mathcal{C} \subset \mathbb{R}^{d\times n}\}.$$

Consider a function composition of length m: $f_m \circ f_{m-1} \circ \cdots \circ f_1(X)$, where $m \in \mathbb{N}^+$ and $f_i \in \mathcal{F}$. For $i \in [m]$, we define the intermediate steps as

$$c_0(X) \coloneqq X, \quad c_i(X) \coloneqq f_i(c_{i-1}(X)),$$
$$\widehat{c}_0(X) \coloneqq X, \quad \widehat{c}_i(X) \coloneqq \mathrm{Attn}_6(\begin{bmatrix} \widehat{c}_{i-1}(X) \\ W_i \end{bmatrix}),$$

where W_i is the weight encoding for f_i as in Theorem 3.1. Then, for any $\epsilon > 0$, we have

$$\mathrm{d}_p(c_t(X), \widehat{c}_t(X)) \le \epsilon,$$

where $t \in [m] \cup \{0\}$.

Proof. Please see Section D.10 for a detailed proof. □

Theorem 4.2 formalizes a common LLM workflow: solve a hard task by prompting step-by-step. Each round provides a new instruction and uses the previous output as the new context. Our result shows that this multi-turn execution supports arbitrarily long compositions without increasing model depth. Hence, a fixed-depth model has the ability to solve arbitrarily complex function compositions.

Comparison with Existing Works. Xiong et al. (2024) rely on the ReLU-transformer framework from Bai et al. (2023)

and define tasks via input-output datasets, our results apply to the standard softmax Transformer and utilize prompts like rules, code, and algorithmic specifications (our weight encoding is an abstraction of the program) instead of just a dataset. Conceptually, whereas (Xiong et al., 2024) emphasizes parallel multi-task behavior, our results target the composition of sequential tasks.

5 Conclusion and Discussion

Section 3 and Section 4 establish two capabilities of a single frozen softmax Transformer: (i) prompt-programmed universality, that is for any continuous sequence-to-sequence function f, different input prompt P_f makes the same model approximate arbitrary f; and (ii) prompt-level composition, i.e., prompts for subroutines are assembled so the model executes their function composition either in one-shot (with depth growth) or across turns (with fixed depth). We now interpret these theorems through the lens of existing ICL theory and clarify what conceptual gap they close.

> Why existing theory typically satisfies only a subset of desiderata?

To isolate what is missing in current theory, we state three desiderata for a *general-purpose in-context* account:

1. **Standard Softmax Attention (D1).** Theorems and proofs operate on ordinary softmax attention, not on analytically convenient surrogates (e.g., linearized or otherwise simplified attention dynamics).
2. **One Frozen Model Across Tasks (D2).** A single model $\theta^\star$ supports many different computations. Prompts carry all task-specific information while weights remain fixed.
3. **Constructive, Reusable Prompt Mechanism (D3).** An explicit "compiler" maps each target computation f to a prompt P_f, and a modular rule assembles prompts for composite programs.

Under these desiderata, we make precise the statement that "existing work typically satisfies only a subset" by examining the dominant theoretical strands.

Notation. We write T_θ for the sequence-to-sequence map computed by the Transformer with parameters θ. Given a prompt P and an input X, we abbreviate the model output on the concatenated sequence (P, X) as $T_\theta(P, X)$.

Statistical-Learning/Pattern-Matching Analyses. These works study a fixed learned predictor $T_\theta(\cdot)$ and analyze generalization and/or behavior under assumptions on a distribution over tasks, inputs, or prompts, often in stylized ICL settings (Li et al., 2023; Garg et al., 2022; Ahuja and Lopez-Paz, 2023; Bhattamishra et al., 2024). They emphasize distributional generalization of the learned mapping rather than programmable re-targeting across arbitrarily specified functions. As a result, they do not establish (D2)-(D3): they do not exhibit a single $\theta^\star$ that supports prompt-driven re-targeting across a broad family of functions, and they do not supply a reusable compilation procedure $f \mapsto P_f$. Moreover, for analytic tractability, some mechanistic analyses adopt simplified attention models (e.g., linear self-attention (Von Oswald et al., 2023; Ahn et al., 2023)), so they address (D1) only partially.

Task-Specific ICL as Forward-Pass Algorithm Execution. This line formalizes in-context learning as algorithm execution in the forward pass. Typically, these works implement the target algorithm in task-specific weights, while the prompt supplies *instances* (e.g., training examples or datasets) rather than a program description (Bai et al., 2023; Akyürek et al., 2023; Von Oswald et al., 2023).

$$\forall \text{ algorithm } a, \quad \exists\, \theta_a \quad \text{s.t.} \quad T_{\theta_a}(\mathcal{D}, x) \approx a(\mathcal{D}, x),$$

where $\mathcal{D}$ denotes the in-context data/examples. Because θ_a varies with a, this framework violates (D2). It also omits (D3): its prompts serve as data containers, not as program encodings with a reusable compiler interface. Moreover, many proofs again simplify attention dynamics, so they resolve (D1) only partially (Von Oswald et al., 2023; Bai et al., 2023).

Expressivity/Universality via Nonconstructive Existence. Expressivity results often show that Transformers represent rich function classes, but many rely on existential arguments (Pérez et al., 2021; Wang et al., 2023; Hu et al., 2024a; Qiu et al., 2024) or under unrealistic assumption on data or architecture (Petrov et al., 2024b) that do not yield reusable mechanisms. The general-purpose in-context regime requires the reverse quantifier order:

$$\exists\, \theta^\star, \quad \forall\, f, \quad \exists P_f \quad \text{s.t.} \quad T_{\theta^\star}(P_f, X) \approx f(X),$$

with model $\theta^\star$ fixed and prompts expressing task variation. This quantifier swap explains why expressivity alone does not explain prompt programmability with frozen weights. Moreover, nonconstructive proofs omit a compiler $f \mapsto P_f$ and omit a modular assembly rule for composites, so they do not address (D3). When these results also replace softmax attention with surrogates, they leave (D1) open (Pérez et al., 2021; Qiu et al., 2024).

Where Our Results Fit. Our construction meets all three desiderata. We operate with standard softmax attention (D1). We fix a single model $\theta^\star$ and show that prompts steer the model to realize broad function classes (D2). We also provide an explicit prompt-to-program mechanism (D3). In addition, our composition theorem strengthens the bare existence of a prompt for $g \circ f$ into a modular assembly rule: it constructs $P_{g\circ f}$ from reusable components P_f and P_g and controls error propagation.

Impact Statement

By the theoretical nature of this work, we do not anticipate any negative social impact.

References

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Advances in Neural Information Processing Systems*, 2023.

Kshitij Ahuja and David Lopez-Paz. A closer look at in-context learning under distribution shifts. OpenReview preprint, 2023.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *International Conference on Learning Representations*, 2023.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances in Neural Information Processing Systems*, 2023.

Satwik Bhattamishra, Arkil Patel, Phil Blunsom, and Varun Kanade. Understanding in-context learning in transformers and llms by learning to learn discrete functions. In *International Conference on Learning Representations*, 2024.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Low-rank bottleneck in multi-head attention models. In *International conference on machine learning*, pages 864–873. PMLR, 2020.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24 (240):1–113, 2023.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pages 5793–5831. PMLR, 2022.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, 2022.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2023.

Jerry Yao-Chieh Hu, Wei-Po Wang, Ammar Gilani, Chenyang Li, Zhao Song, and Han Liu. Fundamental limits of prompt tuning transformers: Universality, capacity and efficiency. *arXiv preprint arXiv:2411.16525*, 2024a.

Jerry Yao-Chieh Hu, Weimin Wu, Yi-Chen Lee, Yu-Chao Huang, Minshuo Chen, and Han Liu. On statistical rates of conditional diffusion transformers: Approximation, estimation and minimax optimality. *arXiv preprint arXiv:2411.17522*, 2024b.

Jerry Yao-Chieh Hu, Hude Liu, Hong-Yu Chen, Weimin Wu, and Han Liu. Universal approximation with softmax attention. *arXiv preprint arXiv:2504.15956*, 2025.

Jerry Yao-Chieh Hu, Hude Liu, Jennifer Yuntong Zhang, and Han Liu. In-context algorithm emulation in fixed-weight transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Haotian Jiang and Qianxiao Li. Approximation rate of the transformer architecture for sequence modeling. *Advances in Neural Information Processing Systems*, 37: 68926–68955, 2024.

Tokio Kajitsuka and Issei Sato. Are transformers with one layer self-attention using low-rank weight matrices universal approximators? *arXiv preprint arXiv:2307.14023*, 2023.

Anastasis Kratsios. Universal regular conditional distributions via probabilistic transformers. *Constructive Approximation*, 57(3):1145–1212, 2023.

Gen Li, Yuchen Jiao, Yu Huang, Yuting Wei, and Yuxin Chen. Transformers meet in-context learning: A universal approximation theory. *arXiv preprint arXiv:2506.05200*, 2025.

Yingcong Li, M. Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2023.

Valerii Likhosherstov, Krzysztof Choromanski, and Adrian Weller. On the expressive flexibility of self-attention

matrices. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8773–8781, 2023.

Zihan Lin and Jason D. Lee. Dual operating modes of in-context learning. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2024.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Sejun Park, Jaeho Lee, Chulhee Yun, and Jinwoo Shin. Provable memorization via deep neural networks using sub-linear parameters. In *Conference on learning theory*, pages 3627–3661. PMLR, 2021.

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is turing-complete. *Journal of Machine Learning Research*, 22(75):75:1–75:35, 2021.

Aleksandar Petrov, Tom Lamb, Alasdair Paren, Philip Torr, and Adel Bibi. Universal in-context approximation by prompting fully recurrent models. *Advances in Neural Information Processing Systems*, 37:72061–72093, 2024a.

Aleksandar Petrov, Philip HS Torr, and Adel Bibi. Prompting a pretrained transformer can be a universal approximator. *arXiv preprint arXiv:2402.14753*, 2024b.

Ruizhong Qiu, Zhe Xu, Wenxuan Bao, and Hanghang Tong. Ask, and it shall be given: On the turing completeness of prompting. *arXiv preprint arXiv:2411.01992*, 2024.

Ruizhong Qiu, Zhe Xu, Wenxuan Bao, and Hanghang Tong. Ask, and it shall be given: On the turing completeness of prompting. In *International Conference on Learning Representations*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. In *Advances in Neural Information Processing Systems*, 2023.

Maojiang Su, Jerry Yao-Chieh Hu, Yi-Chen Lee, Ning Zhu, Jui-Hui Chung, Shang Wu, Zhao Song, Minshuo Chen, and Han Liu. High-order flow matching: Unified framework and sharp statistical rates. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Shokichi Takakura and Taiji Suzuki. Approximation and estimation ability of transformers for sequence-to-sequence functions with infinite dimensional input. In *International Conference on Machine Learning*, pages 33416–33447. PMLR, 2023.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR, 2023.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2023.

Yihan Wang, Jatin Chauhan, Wei Wang, and Cho-Jui Hsieh. Universality and limitations of prompt tuning. *Advances in Neural Information Processing Systems*, 36:75623–75643, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Noam Wies, Yoav Levine, and Amnon Shashua. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36:36637–36651, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

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

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

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020a.

Chulhee Yun, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. O (n) connections are expressive enough: Universal approximability of sparse transformers. *Advances in Neural Information Processing Systems*, 33:13783–13794, 2020b.

Appendix

A Related Work

In-Context Algorithm Emulation. Recently, Hu et al. (2026) take a first step toward the "one fixed model for many programs" phenomenon by giving a provably correct minimal construction for in-context algorithm emulation with frozen weights. Their key insight is that softmax attention can be forced, via prompt-induced query–key margins, to follow a prescribed computation. This allows a fixed Transformer to emulate a library of attention-implementable algorithms by changing only the prompt. However, that universality is limited to algorithms implementable by a fixed attention head, and it supports only one program at a time. The present work pushes this frontier in two directions: from attention-implementable programs to generic functions (via in-context universal approximation), and from one program at a time to program assembly (via in-context composition).

In-Context Universal Approximation. Prior work by Li et al. (2025) develop a in-context universal approximation theory in the example-conditioned regime. Their approach represents each target function as a linear combination of universal features, then recovers the coefficients by a fixed transformer approximating the Lasso algorithm. This still fall into the "Amortized in-context inference" setting in our introduction. We study the opposite end of the spectrum: *prompt-programmed execution*. We do not infer a function from examples. The prompt encodes a procedure and the frozen model executes it as an interpreter. This framing matches a common zero-shot mode of LLM usage: the prompt contains instructions, constraints, or algorithmic steps but no input-output demonstrations, yet the model still performs well. Our function class and outputs also differ. They focus on scalar prediction $\mathbb{R}^d \to \mathbb{R}$, while we study sequence-to-sequenc function. Finally, the prompt-programmed viewpoint also makes functions/tasks composition natural as describe in Section 4, which match common LLM usage patterns for multi-step tasks. A related line of work studies universality via prefix tuning. Petrov et al. (2024b) prove pretrained transformer approximate smooth continuous function on a hypersphere through prefix tuning. Their construction treats attention as kernel interpolation. The prefix encodes reference inputs and corresponding values $(u_i, f(u_i))$. The model interpolates these values at a query point by similarity. In contrast, we consider continuous functions on general compact domains and use a different prompt interface. Furthermore, their P_f is a interpolation table, not a reusable procedure related to target function that naturally lead to functions/tasks composition as shown in Section 4. Specifically, the need to rebuild the prefix vector containing $(u_i, g \circ f(u_i))$ that doesn't reuse P_f and P_g as modular components as we do. Petrov et al. (2024a) further extend the universal in-context approximation result to recurrent architectures.

In-Context Multi-Tasks/Functions Composition. Prior work by Xiong et al. (2024) study pretrained Transformers under mixed-task demonstrations. One prompt contains input-output examples from several tasks, and the model solve multiple tasks by different head of attention in one forward pass. They use the ReLU-transformer framework by Bai et al. (2023) as

theoretical background. We address a different mechanism that reflects how LLM are used: frozen softmax Transformers is in-context universal approximator when the prompt provides procedure descriptions like rules, code, and algorithmic specifications (our "weight encoding" is a clean abstraction of this program text) instead of just dataset. We then extend to show fixed transformer approximate arbitrary function composition in-context. This also mimics and explains the practical multi-step workflows of LLM. Conceptually, whereas Xiong et al. (2024) emphasizes parallel multi-task behavior elicited by mixing demonstrations in a single prompt, our results target sequential tasks composition.

Why Two In-Context UAP Routes? The distinction matters for how one should think about softmax. The first Transformer UAP results view self-attention as a *contextual- mapping* and use feed-forward networks (FFNs) to map out function value. Yun et al. (2020a) prove that standard Transformers approximate continuous permutation-equivariant sequence-to-sequence functions on compact domains. With positional encodings, they extend universality beyond permutation equivariance. Building on this view, Kajitsuka and Sato (2023) refine the contextual-mapping construction by analyzing softmax (rather than as a hardmax proxy), showing that a single softmax attention layer with low-rank weight matrix suffices for contextual mapping under their design, while one-layer hardmax attention does not. This reduces the depth needed in UAP constructions. In contrast, the softmax-attention UAP by Hu et al. (2025) uses an interpolation view where softmax serves as a near-argmax selector over interpolation points. Presenting both routes demonstrates that our results do not hinge on a single temperature regime of softmax. Section 4 only needs an in-context emulator. Beyond existence-type universality, several works derive quantitative approximation rates by restricting the target class. Takakura and Suzuki (2023) study sequence-to-sequence function on infinite-length inputs and assume a shift-invariant input distribution together with shift-equivariant targets of mixed/anisotropic smoothness. They obtain dimension-free rates under these regularity conditions. Jiang and Li (2024) stablish Jackson-type approximation rates for single-layer, single-head Transformers by assuming finite measured complexity on the target function to enable approximation rates analysis. Several works study Transformer expressiveness theory beyond the standard architecture such as sparse transformer (Yun et al., 2020b) or probabilistic transformer (Kratsios, 2023). Complementary lines study what self-attention can or cannot represent, not limited to generic seq-to-seq function classes (Bhojanapalli et al., 2020; Likhosherstov et al., 2023; Edelman et al., 2022).

B Notation Table

Table 1. Notations and Symbols

Notation	Description
a	Lowercase letters for vectors
A	Uppercase letters for matrices
$A_{i,:}$	i-th row of A
$A_{:,i}$	i-th column of A
$A_{:,i:j}$	Sub-matrix of columns i through j
$\mathbb{1}_d$	d dimensional all-ones vector
$\mathbb{R}$	Set of real numbers
$\mathbb{N}^+$	Set of positive integers
$[n]$	$[n] \coloneqq \{1, 2, \ldots, n\}$
$\mathcal{C}$	Compact domain
d	Token dimension
n	Length of input sequence
H	Number of attention heads
$X = [x_1, \ldots, x_n]$	Matrix of input sequence
x_i	i-th token (column) of X
$\|x\|_2$	ℓ_2 norm of x
$\|X\|_p$	Entry-wise ℓ_p norm of X
$\|X\|_\infty$	$\|X\|_\infty \coloneqq \max_{i,j} \|X_{ij}\|$
$\mathrm{d}_p(f_1, f_2)$	$(\int \|f_1(X) - f_2(X)\|_p^p \mathrm{d}X)^{\frac{1}{p}}$
$\mathbb{1}_{\{\text{condition}\}}$	The indicator
σ_β	Softmax with inverse temperature β
ReLU	ReLU activation
FF	Feed-forward layer
Attn_l	l-layer self-attention
$\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}$	Single head self-attention with a residual connection

C Supplementary Theoretical Backgrounds

C.1 Universal Approximation of Transformer and Attention-Only Model

We state the universal approximation result from (Kajitsuka and Sato, 2023; Hu et al., 2025; Su et al., 2025).

Theorem C.1 (Transformers Universal Approximation, Corollary G.2.1 of (Su et al., 2025))**.** *Let $\mathcal{T}_2$ denote the class of two-layer Transformers with self-attention and positional encoding:*

$$\mathcal{T}_2 \coloneqq \{g : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n} \mid g(X) = \mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1(X + E_p),\ \ E_p \in \mathbb{R}^{d\times n}\},$$

where $\mathrm{FF}_1, \mathrm{FF}_2$ and $\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}$ are feed-forward neural network layers and a single-head self-attention layer as defined in Definitions 2.1 and 2.2, respectively. For $1 \le p < \infty$, let distance $\mathrm{d}_p(\cdot,\cdot)$ be defined as in Definition 2.3. Then, for any continuous function f defined on a compact domain $\mathcal{C} \in \mathbb{R}^{d\times n}$, and $\epsilon_{\mathrm{u}} > 0$, there exists a Transformer $g \in \mathcal{T}_2$ such that

$$\mathrm{d}_p(f, g) < \epsilon_{\mathrm{u}}.$$

Theorem C.2 (Sequence-to-Sequence Universal Approximation of Multi-Head Softmax Attention, Theorem G.1 of (Hu et al., 2025))**.** *Let $1 \le p < \infty$. Let $\mathcal{C} \subset \mathbb{R}^{d\times n}$ be a compact domain of input sequences. Let $f : \mathcal{C} \to \mathbb{R}^{d\times n}$ be a continuous sequence-to-sequence function. Let $X_a \in \mathbb{R}^{(d+n+1)\times(n+1)}$ be the augmented input*

$$X_a \coloneqq \begin{bmatrix} X & 0_{d\times 1} \\ I_n & 0_{n\times 1} \\ 0_{1\times n} & 1 \end{bmatrix} \in \mathbb{R}^{(d+n+1)\times(n+1)}.$$

Then, for any $\epsilon > 0$, there exists a network T composed of four multi-head attention layers such that

$$\mathrm{d}_p(T(X_a)_{:,1:n}, f(X)) < \epsilon.$$

C.2 In-Context Emulation

We state the approximation of a single-head self-attention from (Hu et al., 2026).

Definition C.1 (Vectorization)**.** *For any matrix $W \in \mathbb{R}^{p\times s}$, we define $\underline{W} \coloneqq \mathrm{vec}(W) \in \mathbb{R}^{ps}$ such that $\underline{W}_{(i-1)s+j} = W_{ij}$ for all $i \in [p]$ and $j \in [s]$.*

Definition C.2 (Input Prompt for In-Context Emulation of Attention, Definition 4.1 of (Hu et al., 2026))**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence, and let $W_K, W_Q \in \mathbb{R}^{d_h\times d}$, $W_V \in \mathbb{R}^{d_V\times d}$ be the weight matrices of the target attention head to be emulated. Define the vectorizations*

$$\underline{W}_K \coloneqq \mathrm{vec}(W_K) \in \mathbb{R}^{dd_h},\ \underline{W}_Q \coloneqq \mathrm{vec}(W_Q) \in \mathbb{R}^{dd_h},\ \underline{W}_V \coloneqq \mathrm{vec}(W_V) \in \mathbb{R}^{dd_V},$$

and

$$w \coloneqq [\underline{W}_K; \underline{W}_Q; \underline{W}_V] \in \mathbb{R}^{d(2d_h+d_V)},$$

where w is the concatenation of $\underline{W}_K, \underline{W}_Q, \underline{W}_V$. Finally, define the extended input X_p for the in-context emulation of the attention head specified by W_K, W_Q, W_V as

$$X_p \coloneqq \begin{bmatrix} X \\ W_{\mathrm{in}} \\ I_n \end{bmatrix} \quad \textit{with} \quad W_{\mathrm{in}} \coloneqq \begin{bmatrix} 0\cdot w & 1\cdot w & 2\cdot w & \cdots & (n-1)\cdot w \\ w & w & w & \cdots & w \end{bmatrix} \in \mathbb{R}^{2d(2d_h+d_V)\times n}.$$

Theorem C.3 (In-Context Emulation of Attention, Theorem 4.1 of (Hu et al., 2026))**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence, and let $W_K \in \mathbb{R}^{d_h\times d}$, $W_Q \in \mathbb{R}^{d_h\times d}$, $W_V \in \mathbb{R}^{d_V\times d}$ be the weight matrices of the target attention head we wish to emulate in-context. Assume $\|W_K X\|_\infty, \|W_Q X\|_\infty, \|W_V X\|_\infty \le B_{KQV}$ where $B_{KQV} > 0$. Then, for any $\epsilon_e > 0$, there exists a two-layer attention network — a multi-head attention layer Attn followed by a single-head attention layer $\mathrm{Attn}_{\mathrm{s}}$ — such that*

$$\|\mathrm{Attn}_{\mathrm{s}} \circ \mathrm{Attn}(X_p) - W_V X \cdot \sigma_\beta((W_K X)^\top W_Q X)\|_\infty \le \epsilon_e,$$

where X_p is the prompt defined in Definition C.2.

C.3 In-Context Approximation of Truncated Linear Model

We first define the truncated linear function and the truncated linear model.

Definition C.3 (Truncated Linear Function)**.** *We define the truncated linear function as follows:*

$$\mathrm{Range}_{[a,b]}(x) = \begin{cases} a & x \le a, \\ x & a \le x \le b, \\ b & b \le x. \end{cases}$$

Intuitively, the truncated linear function is a segment of a linear function, with the output value ranging from a to b.

Definition C.4 (Truncated Linear Model)**.** *Define a truncated linear model as* $\mathrm{Range}_{[a,b]}(w^\top x + t)$*, where* $w \in \mathbb{R}^d$ *is a learnable weight and* $t \in \mathbb{R}$ *is a bias.*

Then, we restate the in-context approximation of the truncated linear model from Hu et al. (2026).

Theorem C.4 (Multi-Head Attention Approximates Truncated Linear Models In Context, Theorem B.1 of (Hu et al., 2026))**.** *Let* $X \in \mathbb{R}^{d\times n}$ *be the input. Fix real numbers* $a < b$*, and let the truncation operator* $\mathrm{Range}_{[a,b]}(\cdot)$ *follow Definition C.3. Let* w_s *denote the linear coefficient of the in-context truncated linear model. Define* W_s *as*

$$W_s := \begin{bmatrix} 0\cdot w_s & 1\cdot w_s & \cdots & (n-1)\cdot w_s \\ w_s & w_s & \cdots & w_s \end{bmatrix} \in \mathbb{R}^{2d\times n}.$$

For a precision parameter $p > n$ *with* $\epsilon = O(1/p)$*, number of head* $H = p/(n-2)$ *there exists a single-layer,* H*-head self-attention* Attn^H *with a linear transformation* $A : \mathbb{R}^{d\times n} \to \mathbb{R}^{(3d+n)\times n}$*, such that* $\mathrm{Attn}^H \circ A : \mathbb{R}^{d\times n} \to \mathbb{R}^{d_o\times n}$ *satisfies, for any* $i \in [n]$*,*

$$\|\mathrm{Attn}^H \circ A(\begin{bmatrix} X \\ W_s \end{bmatrix})_{:,i} - \mathrm{Range}_{[a,b]}(w_s^\top x_i) e_{\widetilde{k}_i}\|_\infty \le \underbrace{\max\{|a|,|b|\}\cdot\epsilon_0}_{\textit{finite-}\beta\textit{ softmax error}} + \underbrace{\frac{b-a}{(n-2)H}}_{\textit{interpolation error}}.$$

Here $e_{\widetilde{k}_i}$ *is the one-hot vector with a* 1 *at position* $\widetilde{k}_i$*-th index and* 0 *elsewhere, and*

$$\widetilde{k}_i = G(k_i) \in [d_o], \quad \textit{with} \quad k_i = \operatorname*{argmin}_{k\in\{0,1,\cdots,p-1\}} (-2x_i^\top w_i - 2t_i + \widetilde{L}_0 + \widetilde{L}_k)\cdot k,$$

where $G : [p] \to [d_o]$ *denotes any set-to-constant function sending each selected interpolation index* k_i *into an appropriate integer* $\widetilde{k}_i \in [d_o]$ *for* $i \in [n]$*.*

Proof. For completeness, we restate the necessary step for later modifications. Define $A : \mathbb{R}^{d\times n} \to \mathbb{R}^{(3d+n)\times n}$ for the input sequence X as

$$A(X) := \underbrace{\begin{bmatrix} I_{3d} \\ 0_{n\times 3d} \end{bmatrix}}_{\text{token-wise linear}} \begin{bmatrix} X \\ W_s \end{bmatrix} + \underbrace{\begin{bmatrix} 0_{3d\times n} \\ I_n \end{bmatrix}}_{\text{positional encoding}} = \begin{bmatrix} X \\ W_s \\ I_n \end{bmatrix} \in \mathbb{R}^{(3d+n)\times n}.$$

Thus, A is a token-wise linear layer augmented with positional encoding, as it applies a linear projection to each token and then adds a unique per-token bias. Please refer to the original proof for the remaining details. □

We extend the above theorem as follows for later use.

Corollary C.4.1. *Let* $X \in \mathbb{R}^{d\times n}$ *be the input. Fix real numbers* $a < b$*, and let the truncation operator* $\mathrm{Range}_{[a,b]}(\cdot)$ *follow Definition C.3. For total* l *truncated linear models,let* w_j *denote the linear coefficient of the* j*-th in-context truncated linear model. Define* W_j *as*

$$W_j := \begin{bmatrix} 0\cdot w_j & 1\cdot w_j & \cdots & (n-1)\cdot w_j \\ w_j & w_j & \cdots & w_j \end{bmatrix} \in \mathbb{R}^{2d\times n},$$

where $j \in [l]$. For a precision parameter $p > n$ with $\epsilon = O(1/p)$, number of head $H = p/(n-2)$ there exists a single-layer, H-head self-attention Attn^H *such that* $\mathrm{Attn}^H : \mathbb{R}^{(d(1+2l)+n)\times n} \to \mathbb{R}^{d_o \times n}$ *satisfies, for any $i \in [n]$ and $j \in [l]$,*

$$\|\mathrm{Attn}_j^H(\begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_l \\ I_n \end{bmatrix})_{:,i} - \mathrm{Range}_{[a,b]}(w_j^\top x_i) e_{\widetilde{k}_i}\|_\infty \le \underbrace{\max\{|a|,|b|\}\cdot\epsilon_0}_{\textit{finite-}\beta\textit{ softmax error}} + \underbrace{\frac{b-a}{(n-2)H}}_{\textit{interpolation error}}.$$

Here $e_{\widetilde{k}_i}$ is the one-hot vector with a 1 at position $\widetilde{k}_i$-th index and 0 elsewhere, and

$$\widetilde{k}_i = G(k_i) \in [d_o], \quad \textit{with} \quad k_i = \operatorname*{argmin}_{k\in\{0,1,\cdots,p-1\}} (-2x_i^\top w_i - 2t_i + \widetilde{L}_0 + \widetilde{L}_k)\cdot k,$$

where $G : [p] \to [d_o]$ denotes any set-to-constant function sending each selected interpolation index k_i into an appropriate integer $\widetilde{k}_i \in [d_o]$ for $i \in [n]$.

Proof. Modify the affine linear map A in the original proof to a linear one $A_j : \mathbb{R}^{(d(1+2l)+n)\times n} \to \mathbb{R}^{(3d+n)\times n}$ such that

$$A_j \cdot \begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_l \\ I_n \end{bmatrix} \coloneqq \begin{bmatrix} I_d & 0_{d\times(2dl+n)} \\ 0_{d\times(2d(j-1)+d)} & I_{2d} \quad 0_{d\times(2d(l-j)+n)} \\ 0_{n\times d(2l+1)} & I_n \end{bmatrix} \cdot \begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_l \\ I_n \end{bmatrix} = \begin{bmatrix} X \\ W_j \\ I_n \end{bmatrix} \in \mathbb{R}^{(3d+n)\times n}.$$

That is, A_j picks out W_j.

Note that since the input already contains I_n, the linear transformation A_j is just a linear projection instead of an affine map used to add I_n to the output. This linear transformation A_j is combined with the linear projection of the key, query, and value matrices of attention.

Then, we follow the same proof of Theorem C.4 to complete the proof. □

D Proofs of Main Text

D.1 Proof of Lemma 3.1

Lemma D.1 (Lemma 3.1 Restated: Attention Approximates Feed-Forward Neural Network In Context)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence, and let W_{FF} be the weight of a feed-forward neural network* FF*. We assume $\|X\|_\infty \le B_X$ and $\|W_{\mathrm{FF}}\|_\infty \le B_W$ Let $Z \in \mathbb{R}^{d_Z\times n}$ be a matrix containing an identity matrix I_n with $n \le d_Z$. Assume $\|Z\|_\infty \le B_Z$ for some constant $B_Z > 0$. Then, for every $\epsilon_1 > 0$, there exists a 2-layer self-attention* Attn_2 *such that*

$$\|\mathrm{Attn}_2(\begin{bmatrix} X \\ W_{\mathrm{FF}} \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{FF}(X) \\ Z \end{bmatrix}\|_\infty \le \epsilon_1,$$

where W_{FF} is in the form of $W_{\mathrm{FF}} := [\widetilde{W}_{1,1}^\top, \cdots, \widetilde{W}_{1,r}^\top, \widetilde{W}_{2,1}^\top, \cdots, \widetilde{W}_{2,d}^\top]^\top \in \mathbb{R}^{2(2dr+d+r)\times n}$ with each $\widetilde{W}$ defined in Definition 2.2.

Proof. The proof consists of three steps: (1) We rewrite $W_1x_i + b_1$ as an inner product by augmenting each token, and we fix a truncation range using (B_X, B_w) to obtain ReLU activation. (2) We apply Corollary C.4.1 and sum over $j \in [r]$ to obtain an in-context approximation of the first hidden layer of the ReLU neural network $h_i := \mathrm{ReLU}(W_1x_i+b_1)$. We further use an extra single-head attention to plus the part of matrix we want to flow through this first layer attention ($[\widetilde{W}_{2,1}, \cdots, \widetilde{W}_{2,d}, Z]$). (3) We repeat the same in-context construction for the affine map $W_2h_i + b_2$ (with another augmentation), then we propagate the stage-1 error through W_2.

For the convenience of presentation, we use $\mathrm{Attn}_{2\text{-}2} \circ \mathrm{Attn}_{2\text{-}1}$ to denote the two layers of Attn_2.

Step 1. We first denote the j-th row of W_1 as $w_{1,j} \in \mathbb{R}^d$. For each token $i \in [n]$, define the augmented input

$$\widetilde{x}_i := \begin{bmatrix} x_i \\ 1 \end{bmatrix} \in \mathbb{R}^{d+1}, \qquad \widetilde{X} := \begin{bmatrix} X \\ \mathbb{1}_n^\top \end{bmatrix} \in \mathbb{R}^{(d+1)\times n},$$

and for each hidden row $j \in [r]$, define

$$\widetilde{w}_{1,j} := \begin{bmatrix} w_{1,j} \\ (b_1)_j \end{bmatrix} \in \mathbb{R}^{d+1}.$$

Then $\widetilde{w}_{1,j}^\top \widetilde{x}_i = w_{1,j}^\top x_i + (b_1)_j$.

We use elementwise $\|\cdot\|_\infty$. Hence for $u, v \in \mathbb{R}^m$ we have $|u^\top v| \le m\|u\|_\infty\|v\|_\infty$. With $\|x_i\|_\infty \le B_X$, $\|\widetilde{x}_i\|_\infty \le \widetilde{B}_X := \max\{B_X, 1\}$, and $\|\widetilde{w}_{1,j}\|_\infty \le B_W$, we get for all i, j,

$$|\widetilde{w}_{1,j}^\top \widetilde{x}_i| \le (d+1)\|\widetilde{w}_{1,j}\|_\infty\|\widetilde{x}_i\|_\infty \le (d+1)\widetilde{B}_X B_w.$$

Define

$$B_1 := (d+1)\widetilde{B}_X B_w.$$

Then $\mathrm{ReLU}(\widetilde{w}_{1,j}^\top \widetilde{x}_i) = \mathrm{Range}_{[0,B_1]}(\widetilde{w}_{1,j}^\top \widetilde{x}_i)$ for all i, j.

Step 2: Approximate The Hidden Activation $\mathrm{ReLU}(W_1x_i+b_1)$ **In-Context.** For each $j \in [r]$, define the weight-encoding block (same pattern as Corollary C.4.1, with dimension $d+1$)

$$\widetilde{W}_{1,j} := \begin{bmatrix} 0\cdot\widetilde{w}_{1,j} & 1\cdot\widetilde{w}_{1,j} & \cdots & (n-1)\cdot\widetilde{w}_{1,j} \\ \widetilde{w}_{1,j} & \widetilde{w}_{1,j} & \cdots & \widetilde{w}_{1,j} \end{bmatrix} \in \mathbb{R}^{2(d+1)\times n}.$$

By Corollary C.4.1, With $a = 0$ and $b = B_1$, for each $j \in [r]$ there exists an H_1-head module $\mathrm{Attn}_j^{H_1} \circ A_j$ such that for all

$i \in [n]$,

$$\left\|\mathrm{Attn}_j^{H_1}\left(\begin{bmatrix}\widetilde{X}\\ \widetilde{W}_{1,1}\\ \vdots\\ \widetilde{W}_{1,r}\\ \widetilde{W}_{2,1}\\ \vdots\\ \widetilde{W}_{2,r}\\ Z\end{bmatrix}\right)_{:,i} - \mathrm{Range}_{[0,B_1]}(\widetilde{w}_{1,j}^\top \widetilde{x}_i)\, e_j\right\|_\infty \le B_1\epsilon_0 + \frac{B_1}{(n-2)H_1}.$$

Summing over $j \in [r]$ yields a $(H_1 \cdot r)$-head attention module that satisfies

$$\left\|\sum_{j=1}^{r}\mathrm{Attn}_j^{H_1}\left(\begin{bmatrix}\widetilde{X}\\ \widetilde{W}_{1,1}\\ \vdots\\ \widetilde{W}_{1,r}\\ \widetilde{W}_{2,1}\\ \vdots\\ \widetilde{W}_{2,d}\\ Z\end{bmatrix}\right)_{:,i} - \begin{bmatrix}\mathrm{ReLU}(W_1 x_i + b_1)\\ 0_{(2d(r+1)+d_Z)\times n}\end{bmatrix}\right\|_\infty \le \eta_1 \quad \text{with} \quad \eta_1 := B_1\epsilon_0 + \frac{B_1}{(n-2)H_1}. \tag{3}$$

This completes the in-context approximation of the first hidden layer. However, we still need to build another attention into this layer's multi-head attention module to flow the $[\widetilde{W}_{2,1}, \cdots, \widetilde{W}_{2,d}, Z]$ for later layer use. In the proof of Lemma 3.3 in Section D.3, we know there exist an attention layer exactly copy the identity matrix I_n (see Equations (18) to (25)), and also exist an attention layer approximate $\mathrm{Attn}(Z) = Z$ (see Equations (26) to (29)). Since Z contain I_n, we represent Z as $Z = [Z_1, I_n, Z_2]$. We use singel-head attention to copy I_n, another head to flow $[\widetilde{W}_{2,1}, \cdots, \widetilde{W}_{2,d}, Z_1]$, and another head to flow Z_2.

In summary, there exists a 3-head attention $\mathrm{Attn}^{(3)}$ such that

$$\mathrm{Attn}^{(3)}\left(\begin{bmatrix}\widetilde{X}\\ \widetilde{W}_{1,1}\\ \vdots\\ \widetilde{W}_{1,r}\\ \widetilde{W}_{2,1}\\ \vdots\\ \widetilde{W}_{2,d}\\ Z\end{bmatrix}\right) = \begin{bmatrix}0_{d\times n}\\ \widetilde{W}_{2,1}\sigma_{\beta_2}(I_n)\\ \vdots\\ \widetilde{W}_{2,d}\sigma_{\beta_2}(I_n)\\ Z_1\sigma_{\beta_2}(I_n)\\ I_n\\ Z_2\sigma_{\beta_2}(I_n)\end{bmatrix},$$

and the error is, by (55)

$$\left\|\mathrm{Attn}^{(3)}\left(\begin{bmatrix}\widetilde{X}\\ \widetilde{W}_{1,1}\\ \vdots\\ \widetilde{W}_{1,r}\\ \widetilde{W}_{2,1}\\ \vdots\\ \widetilde{W}_{2,d}\\ Z\end{bmatrix}\right) - \begin{bmatrix}0_{d\times n}\\ \widetilde{W}_{2,1}\\ \vdots\\ \widetilde{W}_{2,d}\\ Z\end{bmatrix}\right\|_\infty \le n\cdot\max(B_Z, B_W)\frac{(n-1)}{e^{\beta_2}+(n-1)} =: \eta_{\text{flow}}. \tag{4}$$

Combining with (3), there exists an $(H_1 \cdot r + 3)$-head attention $\mathrm{Attn}_{2\text{-}1}$ such that

$$\|\mathrm{Attn}_{2\text{-}1}(\begin{bmatrix} X \\ W_{\mathrm{FF}} \\ Z \end{bmatrix})_{:,i} - \begin{bmatrix} \mathrm{ReLU}(W_1 X + b_1) \\ \widetilde{W}_{2,1} \\ \vdots \\ \widetilde{W}_{2,d} \\ Z \end{bmatrix}_{:,i})\|_\infty \le \max(\eta_1, \eta_{\mathrm{flow}}).$$

Step 3: Second-Layer Attention and The Three-Term Error Bound. We write $h_i := \mathrm{ReLU}(W_1 x_i + b_1)$ and keep $\widehat{h}_i \in \mathbb{R}^r$ for the output of the ReLU-approximation heads from Step 2. We construct $\mathrm{Attn}_{2\text{-}2}$ to implement the in-context affine map on input $\widehat{h}_i$.

Denote the output of the first-layer $\mathrm{Attn}_{2\text{-}1}$ as U_{input} and define the ideal input as U_{ideal}

$$U_{\mathrm{input}} := \begin{bmatrix} \widehat{h} \\ \widetilde{W}_{2,1}\sigma_{\beta_2}(I_n) \\ \vdots \\ \widetilde{W}_{2,d}\sigma_{\beta_2}(I_n) \\ Z\sigma_{\beta_2}(I_n) \end{bmatrix}, \qquad U_{\mathrm{ideal}} := \begin{bmatrix} \widehat{h} \\ \widetilde{W}_{2,1} \\ \vdots \\ \widetilde{W}_{2,d} \\ Z \end{bmatrix}.$$

From (4), we have

$$\|U_{\mathrm{input}} - U_{\mathrm{ideal}}\|_\infty \le n \max\{B_W, B_Z\} \frac{n-1}{e^{\beta_2} + (n-1)} =: \eta_{\mathrm{flow}}. \tag{5}$$

Assume $\mathrm{Attn}_{2\text{-}2}$ is L-Lipschitz under element-wise $\|\cdot\|_\infty$: for any two inputs U, V of matching shape,

$$\|\mathrm{Attn}_{2\text{-}2}(U) - \mathrm{Attn}_{2\text{-}2}(V)\|_\infty \le L\|U - V\|_\infty. \tag{6}$$

Apply (6) with $U = U_{\mathrm{input}}$ and $V = U_{\mathrm{ideal}}$ and combine with (5):

$$\|\mathrm{Attn}_{2\text{-}2}(U_{\mathrm{input}}) - \mathrm{Attn}_{2\text{-}2}(U_{\mathrm{ideal}})\|_\infty \le L\,\eta_{\mathrm{flow}}. \tag{7}$$

Again like in step 2, we implement b_2 by augmenting the hidden state with a constant 1. We achieve this through the W_V matrix in the attention to extract constant from Z (the I_n sub-block). Define the augmented hidden vector

$$\widetilde{h}_i := \begin{bmatrix} \widehat{h}_i \\ 1 \end{bmatrix} \in \mathbb{R}^{r+1}.$$

Now we bound the linear model for later choose the a, b in truncated linear model properly. From Step 1 we have $\|h_i\|_\infty \le B_1$, and from (3) we have $\|\widehat{h}_i - h_i\|_\infty \le \eta_1$. Hence

$$\|\widehat{h}_i\|_\infty \le B_1 + \eta_1 \quad \Rightarrow \quad \|\widetilde{h}_i\|_\infty \le \max\{B_1 + \eta_1, 1\} =: \widetilde{B}_1.$$

For each $k \in [d]$, define $\widetilde{w}_{2,k} := [(W_2)^\top_{k,:}; (b_2)_k] \in \mathbb{R}^{r+1}$, so $\widetilde{w}^\top_{2,k}\widetilde{h}_i = (W_2\widehat{h}_i + b_2)_k$. With elementwise $\|\cdot\|_\infty$ and $\|\widetilde{w}_{2,k}\|_\infty \le B_W$,

$$|\widetilde{w}^\top_{2,k}\widetilde{h}_i| \le (r+1)\|\widetilde{w}_{2,k}\|_\infty \|\widetilde{h}_i\|_\infty \le (r+1) B_W \widetilde{B}_1.$$

Set

$$B_2 := (r+1) B_W \widetilde{B}_1.$$

Apply Corollary C.4.1 with dimension $(r+1)$, $l=d$, and $[a,b]=[-B_2,B_2]$ to the input U_{ideal}. We obtain a multi-head attention module $\text{Attn}_{2\text{-}2}$ such that for all $i\in[n]$,

$$\|\text{Attn}_{2\text{-}2}(U_{\text{ideal}})_{:,i}-\begin{bmatrix}(W_2\widehat{h}_i+b_2)\\0_{d_Z\times n}\end{bmatrix}\|_\infty\le\eta_2,\qquad \eta_2:=B_2\epsilon_0+\frac{2B_2}{(n-2)H_2}. \tag{8}$$

For each $i\in[n]$,

$$\|W_2(\widehat{h}_i-h_i)\|_\infty\le r\|W_2\|_\infty\|\widehat{h}_i-h_i\|_\infty\le rB_W\eta_1. \tag{9}$$

Using triangle inequality and (7), (8), (9), we get for all $i\in[n]$,

$$\begin{aligned}&\|\text{Attn}_{2\text{-}2}\circ\text{Attn}_{2\text{-}1}([X;W_{\text{FF}};Z])_{:,i}-\begin{bmatrix}(W_2h_i+b_2)\\0_{d_Z\times n}\end{bmatrix}\|_\infty\\ \le&\|\text{Attn}_{2\text{-}2}(U_{\text{input}})_{:,i}-\text{Attn}_{2\text{-}2}(U_{\text{ideal}})_{:,i}\|_\infty\\ &+\|\text{Attn}_{2\text{-}2}(U_{\text{ideal}})_{:,i}-\begin{bmatrix}(W_2\widehat{h}_i+b_2)\\0_{d_Z\times n}\end{bmatrix}\|_\infty+\|\begin{bmatrix}W_2(\widehat{h}_i-h_i)\\0_{d_Z\times n}\end{bmatrix}\|_\infty\\ \le&\,L\eta_{\text{flow}}+\eta_2+rB_W\eta_1.\end{aligned} \tag{10}$$

Finally, as in Step 2, we add three extra heads in $\text{Attn}_{2\text{-}2}$ that copy the Z block into the last d_Z coordinates of the output. Hence the Z-block after two-layer attention equals

$$Z=\begin{bmatrix}Z_1\sigma_{\beta_1}(I_n)\sigma_{\beta_2}(I_n)\\I_n\\Z_2\sigma_{\beta_1}(I_n)\sigma_{\beta_2}(I_n)\end{bmatrix},$$

where I_n can be exactly copied.

We invoke the two-layer flow analysis (see (52) – (61)). For any $\varepsilon_1>0$, choose β_1,β_2 as in (61). Then

$$\left\|Z_{\{1,2\}}\big(\sigma_{\beta_1}(I_n)\sigma_{\beta_2}(I_n)-I_n\big)\right\|_\infty\le 2\varepsilon_1+\frac{\varepsilon_1^2}{B_Z}. \tag{11}$$

Combined with (10), we finally have

$$\left\|\text{Attn}_{2\text{-}2}\circ\text{Attn}_{2\text{-}1}(\begin{bmatrix}X\\W_{\text{FF}}\\Z\end{bmatrix})-\begin{bmatrix}\text{FF}(X)\\Z\end{bmatrix}\right\|_\infty\le\max\left\{L\eta_{\text{flow}}+\eta_2+rB_W\eta_1,2\varepsilon_1+\frac{\varepsilon_1^2}{B_Z}\right\}.$$

Choose parameters so that the right-hand side is at most ϵ_1. This completes the proof. □

D.2 Proof of Lemma 3.2

Lemma D.2 (Lemma 3.2 Restated: Attention Approximates Feed-Forward Neural Network In Context)**.** *Under the same setting of Lemma 3.1, for every $\epsilon_2>0$, there exists a 2-layer multi-head self-attention layer Attn_2 such that*

$$\|\text{Attn}_2(\begin{bmatrix}X\\W_{\text{FF}}\end{bmatrix})-\text{FF}(X)\|_\infty\le\epsilon_2.$$

Proof. We follow the proof of Lemma 3.1 and keep the same constructions and notation for the two attention layers. The only change is that we remove the auxiliary heads that copy Z.

After this removal, the two-layer attention output contains only the $\text{FF}(X)$ block. Hence, the final error bound contains only the FF approximation error terms and does not include any Z-flow term. Therefore, for any ϵ_2, by choosing proper parameters, we obtain

$$\|\text{Attn}_2(\begin{bmatrix}X\\W_{\text{FF}}\end{bmatrix})-\text{FF}(X)\|_\infty\le L\eta_{\text{flow}}+\eta_2+rB_W\eta_1\le\epsilon_2.$$

This completes the proof. □

D.3 Proof of Lemma 3.3

Lemma D.3 (Lemma 3.3 Restated: In-Context Emulation of Single-Head Attention with a Residual Connection and a Flow-Through Component)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence and $Z \in \mathbb{R}^{d_Z\times n}$. Let $X_p \in \mathbb{R}^{(d+2d(2d_h+d)+n)\times n}$ be the input prompt from Definition 3.2 with $H = 1$, and set the free parameter $d_V \to d$. Assume $\|X\|_\infty \le B_X, \|Z\|_\infty \le B_Z$ and $\|W_K^h X\|_\infty, \|W_Q^h X\|_\infty, \|W_V^h X\|_\infty \le B_{KQV}$ for some $B_X, B_Z, B_{KQV} > 0$. Then, for any $\epsilon_3 > 0$, there exists a two-layer multi-head attention network Attn_2 such that*

$$\|\mathrm{Attn}_2(\begin{bmatrix} X_p \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}(X) \\ Z \end{bmatrix}\|_\infty \le \epsilon_3.$$

Proof. To keep our proof clean, we define $d_p \coloneqq (d + 2d(2d_h + d) + n)$ and express the input dimension as

$$\underbrace{\begin{bmatrix} X_p \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n}.$$

For the convenience of presentation, we use $\mathrm{Attn}_{2\text{-}2} \circ \mathrm{Attn}_{2\text{-}1}$ to denote the two layers of Attn_2.

Step 1. In this part of the proof, our goal is to construct an $\mathrm{Attn}_{2\text{-}1}$ such that

$$\mathrm{Attn}_{2\text{-}1}(\underbrace{\begin{bmatrix} X_p \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n}) \approx \underbrace{\begin{bmatrix} \mathrm{Attn}(X_p) \\ I_n \\ Z \\ X \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n}, \tag{12}$$

where Attn is the first layer emulator from Theorem C.3, and the $(2d_h+d)$ dimension follows from the proof of Theorem C.3 in Hu et al. (2026). Besides, we insert an identity I_n in the middle of the matrix for use by our construction of $\mathrm{Attn}_{2\text{-}2}$ in the second step of the proof.

To achieve (12), we construct $\mathrm{Attn}_{2\text{-}1}$ to be an $(H + 3)$-head attention. The first H heads reconstruct $\mathrm{Attn}(X_p)$. The $(H + 1)$-th head outputs the middle I_n, and the $(H + 2)$-th head outputs the approximation of Z. The last 1 head outputs an approximation of X.

To keep our proof clear, we denote the h-th head from $\mathrm{Attn}_{2\text{-}1}$ as $\mathrm{Attn}_{2\text{-}1}^{(h)}$ and the weight matrices from the h-th head as $W_K^{1,h}, W_Q^{1,h}, W_V^{1,h}$ and $W_O^{1,h}$.

Now we are ready to construct the heads in $\mathrm{Attn}_{2\text{-}1}$.

We construct the weight matrices of the first H heads as follows:

$$W_O^{1,h} \coloneqq \underbrace{\begin{bmatrix} I_{2d_h+d} \\ 0_{n\times(2d_h+d)} \\ 0_{d_Z\times(2d_h+d)} \\ 0_{d\times(2d_h+d)} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times(2d_h+d)} \tag{13}$$

$$W_V^{1,h} \coloneqq \underbrace{\begin{bmatrix} W_V^h & 0_{(2d_h+d)\times d_Z} \end{bmatrix}}_{(2d_h+d)\times(d_p+d_Z)} \tag{14}$$

$$W_K^{1,h} \coloneqq \underbrace{\begin{bmatrix} W_K^h & 0_{(d+1)\times d_Z} \end{bmatrix}}_{(d+1)\times(d_p+d_Z)} \tag{15}$$

$$W_Q^{1,h} \coloneqq \underbrace{\begin{bmatrix} W_Q^h & 0_{(d+1)\times d_Z} \end{bmatrix}}_{(d+1)\times(d_p+d_Z)}, \tag{16}$$

where W_K^h, W_Q^h and W_V^h denote the weight matrices of the first-layer emulator, and their dimensions follow from the proof of Theorem C.3, and we set $d_V \to d$.

The construction provides us with

$$\underbrace{W_V^{1,h}}_{(2d_h+d)\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X_p \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = \underbrace{\begin{bmatrix} W_V^h & 0_{(2d_h+d)\times d_Z} \end{bmatrix}}_{(2d_h+d)\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X_p \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = \underbrace{W_V^h}_{(2d_h+d)\times d_p} \cdot \underbrace{X_p}_{d_p\times n} \quad \text{(By (14))}$$

$$\underbrace{W_K^{1,h}}_{(d+1)\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X_p \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = \underbrace{\begin{bmatrix} W_K^h & 0_{(d+1)\times d_Z} \end{bmatrix}}_{(d+1)\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X_p \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = \underbrace{W_K^h}_{(d+1)\times d_p} \cdot \underbrace{X_p}_{d_p\times n} \quad \text{(By (15))}$$

$$\underbrace{W_Q^{1,h}}_{(d+1)\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X_p \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = \underbrace{\begin{bmatrix} W_Q^h & 0_{(d+1)\times d_Z} \end{bmatrix}}_{(d+1)\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X_p \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = \underbrace{W_Q^h}_{(d+1)\times d_p} \cdot \underbrace{X_p}_{d_p\times n}. \quad \text{(By (16))}$$

Thus, for $h \in [H]$, we have

$$\begin{aligned}
\mathrm{Attn}_{2\text{-}1}^{(h)}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) &= W_O^{1,h} \cdot W_V^{1,h} \begin{bmatrix} X_p \\ Z \end{bmatrix} \cdot \sigma_{\beta_h}((W_K^{1,h} \begin{bmatrix} X_p \\ Z \end{bmatrix})^\top W_Q^{1,h} \begin{bmatrix} X_p \\ Z \end{bmatrix}) \\
&= \underbrace{\begin{bmatrix} I_{2d_h+d} \\ 0_{n\times(2d_h+d)} \\ 0_{d_Z\times(2d_h+d)} \\ 0_{d\times(2d_h+d)} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times(2d_h+d)} \cdot \underbrace{W_V^h X_p}_{(2d_h+d)\times n} \cdot \underbrace{\sigma_{\beta_h}((W_K^h X_p)^\top W_Q^h X_p)}_{n\times n} \quad \text{(By (13))} \\
&= \underbrace{\begin{bmatrix} W_V^h X_p \cdot \sigma_{\beta_h}((W_K^h X_p)^\top W_Q^h X_p) \\ 0_{n\times n} \\ 0_{d_Z\times n} \\ 0_{d\times n} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n}, \quad \text{(By matrix multiplication)}
\end{aligned}$$

where β_h denotes the temperature of the h-th head in the first-layer emulator.

Then, we have

$$\begin{aligned}
&\sum_{h=1}^{H} \mathrm{Attn}_{2\text{-}1}^{(h)}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) \\
&= \begin{bmatrix} \sum_{h=1}^{H} W_V^h X_p \cdot \sigma_{\beta_h}((W_K^h X_p)^\top W_Q^h X_p) \\ 0_{n\times n} \\ 0_{d_Z\times n} \\ 0_{d\times n} \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} \mathrm{Attn}(X_p) \\ 0_{n\times n} \\ 0_{d_Z\times n} \\ 0_{d\times n} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n}. \qquad (17)
\end{aligned}$$

Secondly, we construct the weight matrices of the $(H+1)$-th head of $\mathrm{Attn}_{2\text{-}1}$ as

$$W_K^{1,H+1} := \underbrace{\begin{bmatrix} 0_{n\times d} & 0_{n\times 2d(2d_h+d)} & I_n & 0_{n\times d_Z} \end{bmatrix}}_{n\times(d_p+d_Z)}, \qquad (18)$$

$$W_Q^{1,H+1} := \underbrace{\begin{bmatrix} 0_{n\times d} & 0_{n\times 2d(2d_h+d)} & M_{n\times n} & 0_{n\times d_Z} \end{bmatrix}}_{n\times(d_p+d_Z)}, \qquad (19)$$

$$W_V^{1,H+1} \coloneqq \underbrace{\begin{bmatrix} 0_{n\times d} & 0_{n\times 2d(2d_h+d)} & A^{-1}_{n\times n} & 0_{n\times d_Z}\end{bmatrix}}_{n\times(d_p+d_Z)}, \tag{20}$$

$$W_O^{1,H+1} \coloneqq \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times n} \\ I_n \\ 0_{d_Z\times n} \\ 0_{d\times n}\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n}, \tag{21}$$

where

$$M \coloneqq \frac{1}{\beta_1}\ln(A), \quad \text{for some } \beta_1 > 0, \tag{22}$$

$$A \coloneqq (1-\alpha)I_n + \frac{\alpha}{n}J, \quad \text{for some } 0<\alpha<1, \tag{23}$$

$$J \coloneqq \underbrace{\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}}_{n\times 1}\underbrace{\begin{bmatrix}1 & 1\cdots & 1\end{bmatrix}}_{1\times n} = \underbrace{\begin{bmatrix}1 & 1 & \cdots & 1\\1 & 1 & \cdots & 1\\\vdots & \vdots & \ddots & \vdots\\1 & 1 & \cdots & 1\end{bmatrix}}_{n\times n}, \tag{24}$$

and A^{-1} is

$$A^{-1} = \frac{1}{1-\alpha}I_n - \frac{\alpha}{n(1-\alpha)}J.$$

The construction provides us with

$$W_K^{1,H+1}\begin{bmatrix}X\\W_{\text{in}}\\I_n\\Z\end{bmatrix} = \underbrace{\begin{bmatrix}0_{n\times d} & 0_{n\times 2d(2d_h+d)} & I_n & 0_{n\times d_Z}\end{bmatrix}}_{n\times(d_p+d_Z)}\underbrace{\begin{bmatrix}X\\W_{\text{in}}\\I_n\\Z\end{bmatrix}}_{(d_p+d_Z)\times n} = I_n \qquad \text{(By (18))}$$

$$W_Q^{1,H+1}\begin{bmatrix}X\\W_{\text{in}}\\I_n\\Z\end{bmatrix} = \underbrace{\begin{bmatrix}0_{n\times d} & 0_{n\times 2d(2d_h+d)} & M_{n\times n} & 0_{n\times d_Z}\end{bmatrix}}_{n\times(d_p+d_Z)}\underbrace{\begin{bmatrix}X\\W_{\text{in}}\\I_n\\Z\end{bmatrix}}_{(d_p+d_Z)\times n} = M_{n\times n} \qquad \text{(By (19))}$$

$$W_V^{1,H+1}\begin{bmatrix}X\\W_{\text{in}}\\I_n\\Z\end{bmatrix} = \underbrace{\begin{bmatrix}0_{n\times d} & 0_{n\times 2d(2d_h+d)} & A^{-1}_{n\times n} & 0_{n\times d_Z}\end{bmatrix}}_{n\times(d_p+d_Z)}\underbrace{\begin{bmatrix}X\\W_{\text{in}}\\I_n\\Z\end{bmatrix}}_{(d_p+d_Z)\times n} = A^{-1}_{n\times n}, \qquad \text{(By (20))}$$

such that

$$\begin{aligned}&\sigma_{\beta_1}((W_K^{1,H+1}\begin{bmatrix}X_p\\Z\end{bmatrix})^\top(W_Q^{1,H+1}\begin{bmatrix}X_p\\Z\end{bmatrix}))\\ &= \sigma_{\beta_1}(I_n^\top M)\\ &= \sigma_{\beta_1}(M) && \text{(By } I_n^\top M = I_n M = M\text{)}\end{aligned}$$

and each entry in $\sigma_{\beta_1}(M)$ is

$$\sigma_{\beta_1}(M)_{ii} = \frac{\exp\Big\{\beta_1\cdot\frac{1}{\beta_1}\ln\big(1-\alpha+\frac{\alpha}{n}\big)\Big\}}{\exp\Big\{\beta_1\cdot\frac{1}{\beta_1}\ln(1-\alpha)\Big\}+n\cdot\exp\Big\{\beta_1\cdot\frac{1}{\beta_1}\ln\big(\frac{\alpha}{n}\big)\Big\}} = 1-\alpha+\frac{\alpha}{n}, \qquad \text{(By (1) and (22))}$$

$$\sigma_\beta(M)_{ij} = \frac{\exp\Big\{\beta_1 \cdot \frac{1}{\beta_1}\ln\big(\frac{\alpha}{n}\big)\Big\}}{\exp\Big\{\beta_1 \cdot \frac{1}{\beta_1}\ln(1-\alpha)\Big\} + n \cdot \exp\Big\{\beta_1 \cdot \frac{1}{\beta}\ln\big(\frac{\alpha}{n}\big)\Big\}} = \frac{\alpha}{n}. \qquad \text{(By (1) and (22))}$$

Thus,

$$\sigma_{\beta_1}(M) = (1-\alpha)I_n + \frac{\alpha}{n}J = A,$$

and

$$\begin{aligned} & W_O^{1,H+1} \cdot W_V^{1,H+1}\begin{bmatrix} X_p \\ Z \end{bmatrix} \cdot \sigma_{\beta_1}((W_K^{1,H+1}\begin{bmatrix} X_p \\ Z \end{bmatrix})^\top (W_Q^{1,H+1}\begin{bmatrix} X_p \\ Z \end{bmatrix})) \\ = & \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times n} \\ I_n \\ 0_{d_Z\times n} \\ 0_{d\times n} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} \cdot \underbrace{A^{-1}A}_{n\times n} = \begin{bmatrix} 0_{(2d_h+d)\times n} \\ I_n \\ 0_{d_Z\times n} \\ 0_{d\times n} \end{bmatrix} \cdot I_n = \begin{bmatrix} 0_{(2d_h+d)\times n} \\ I_n \\ 0_{d_Z\times n} \\ 0_{d\times n} \end{bmatrix}, \end{aligned} \tag{25}$$

where the last equality follows from (21).

Thirdly, we construct the weight matrices of the $(H+2)$-th head as follows:

$$W_O^{1,H+2} \coloneqq \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times d_Z} \\ 0_{n\times d_Z} \\ I_{d_Z} \\ 0_{d\times d_Z} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times d_Z} \tag{26}$$

$$W_V^{1,H+2} \coloneqq \underbrace{\begin{bmatrix} 0_{d_Z\times d_p} & I_{d_Z} \end{bmatrix}}_{d_Z\times(d_p+d_Z)} \tag{27}$$

$$W_K^{1,H+2}, W_Q^{1,H+2} \coloneqq \underbrace{\begin{bmatrix} 0_{n\times d} & 0_{n\times 2d(2d_h+d)} & I_n & 0_{n\times d_Z} \end{bmatrix}}_{n\times(d_p+d_Z)}. \tag{28}$$

The construction provides us with

$$W_V^{1,H+2}\begin{bmatrix} X_p \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{d_Z\times d_p} & I_{d_Z} \end{bmatrix}}_{d_Z\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X_p \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = \underbrace{Z}_{d_Z\times n} \qquad \text{(By (27))}$$

$$W_K^{1,H+2}\begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{n\times d} & 0_{n\times 2d(2d_h+d)} & I_n & 0_{n\times d_Z} \end{bmatrix}}_{n\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = I_n \qquad \text{(By (28))}$$

$$W_Q^{1,H+2}\begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{n\times d} & 0_{n\times 2d(2d_h+d)} & I_n & 0_{n\times d_Z} \end{bmatrix}}_{n\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = I_n. \qquad \text{(By (28))}$$

Thus, we have

$$\text{Attn}_{2\text{-}1}^{(H+2)}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) = W_O^{1,H+2} \cdot W_V^{1,H+2}\begin{bmatrix} X_p \\ Z \end{bmatrix} \cdot \sigma_{\beta_2}((W_K^{1,H+2}\begin{bmatrix} X_p \\ Z \end{bmatrix})^\top W_Q^{1,H+2}\begin{bmatrix} X_p \\ Z \end{bmatrix})$$

$$= \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times d_Z} \\ 0_{n\times d_Z} \\ I_{d_Z} \\ 0_{d\times d_Z} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times d_Z} \cdot \underbrace{Z}_{d_Z\times n} \cdot \underbrace{\sigma_{\beta_2}((I_n)^\top I_n)}_{n\times n}$$

$$= \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times n} \\ 0_{n\times n} \\ Z\sigma_{\beta_2}(I_n) \\ 0_{d\times n} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} . \tag{29}$$

Lastly, we construct the weight matrices of the $(H+3)$-th head as follows:

$$W_O^{1,H+3} := \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times d} \\ 0_{n\times d} \\ 0_{d_Z\times d} \\ I_d \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times d} \tag{30}$$

$$W_V^{1,H+3} := \underbrace{\begin{bmatrix} I_d & 0_{d\times 2d(2d_h+d)} & 0_{d\times n} & 0_{d\times d_Z} \end{bmatrix}}_{d\times(d_p+d_Z)} \tag{31}$$

$$W_K^{1,H+3}, W_Q^{1,H+3} := \underbrace{\begin{bmatrix} 0_{n\times d} & 0_{n\times 2d(2d_h+d)} & I_n & 0_{n\times d_Z} \end{bmatrix}}_{n\times(d_p+d_Z)} . \tag{32}$$

The construction provides us with

$$W_V^{1,H+3} \begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} I_d & 0_{d\times 2d(2d_h+d)} & 0_{d\times n} & 0_{d\times d_Z} \end{bmatrix}}_{d\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = \underbrace{X}_{d\times n} \qquad \text{(By (31))}$$

$$W_K^{1,H+3} \begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{n\times d} & 0_{n\times 2d(2d_h+d)} & I_n & 0_{n\times d_Z} \end{bmatrix}}_{n\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = I_n \qquad \text{(By (28))}$$

$$W_Q^{1,H+3} \begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{n\times d} & 0_{n\times 2d(2d_h+d)} & I_n & 0_{n\times d_Z} \end{bmatrix}}_{n\times(d_p+d_Z)} \cdot \underbrace{\begin{bmatrix} X \\ W_{\text{in}} \\ I_n \\ Z \end{bmatrix}}_{(d_p+d_Z)\times n} = I_n . \qquad \text{(By (28))}$$

Thus, we have

$$\begin{aligned} \text{Attn}_{2\text{-}1}^{(H+3)}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) &= W_O^{1,H+3} \cdot W_V^{1,H+3} \begin{bmatrix} X_p \\ Z \end{bmatrix} \cdot \sigma_{\beta_3}((W_K^{1,H+3} \begin{bmatrix} X_p \\ Z \end{bmatrix})^\top W_Q^{1,H+3} \begin{bmatrix} X_p \\ Z \end{bmatrix}) \\ &= \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times d} \\ 0_{n\times d} \\ 0_{d_Z\times d} \\ I_d \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times d} \cdot \underbrace{X}_{d\times n} \cdot \underbrace{\sigma_{\beta_3}(I_n^\top I_n)}_{n\times n} \end{aligned}$$

$$= \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times n} \\ 0_{n\times n} \\ 0_{d_Z\times n} \\ X\sigma_{\beta_3}(I_n) \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} . \tag{33}$$

Finally, all of the heads from $\mathrm{Attn}_{2\text{-}1}$ give us

$$\begin{aligned}
\mathrm{Attn}_{2\text{-}1}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) &= \sum_{h=1}^{H} \mathrm{Attn}_{2\text{-}1}^{(h)}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) + \mathrm{Attn}_{2\text{-}1}^{(H+1)}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) + \mathrm{Attn}_{2\text{-}1}^{(H+2)}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) + \mathrm{Attn}_{2\text{-}1}^{(H+3)}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) \\
&= \underbrace{\begin{bmatrix} \mathrm{Attn}_m(X_p) \\ 0_{n\times n} \\ 0_{d_Z\times n} \\ 0_{d\times n} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} + \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times n} \\ I_n \\ 0_{d_Z\times n} \\ 0_{d\times n} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} + \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times n} \\ 0_{n\times n} \\ Z\sigma_{\beta_2}(I_n) \\ 0_{d\times n} \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} + \underbrace{\begin{bmatrix} 0_{(2d_h+d)\times n} \\ 0_{n\times n} \\ 0_{d_Z\times n} \\ X\sigma_{\beta_3}(I_n) \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} \\
&\qquad \text{(By (17), (25), (29) and (33))} \\
&= \underbrace{\begin{bmatrix} \mathrm{Attn}_m(X_p) \\ I_n \\ Z\sigma_{\beta_2}(I_n) \\ X\sigma_{\beta_3}(I_n) \end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} .
\end{aligned}$$

Step 2. In this part of the proof, our goal is to construct an $\mathrm{Attn}_{2\text{-}2}$ such that

$$\mathrm{Attn}_{2\text{-}2} \circ \mathrm{Attn}_{2\text{-}1}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) \approx \begin{bmatrix} X + \mathrm{Attn}_{\mathrm{s}} \circ \mathrm{Attn}(X_p) \\ Z \end{bmatrix}, \tag{34}$$

where $\mathrm{Attn}_{\mathrm{s}}$ is the second-layer emulator from Theorem C.3.

To achieve (34), we construct $\mathrm{Attn}_{2\text{-}2}$ as a three-head attention layer. The first head reconstructs $\mathrm{Attn}_{\mathrm{s}} \circ \mathrm{Attn}(X_p)$. The second head outputs an approximation of Z. The third head outputs an approximation of X.

Again, to keep our proof clear, we denote the h-th head from $\mathrm{Attn}_{2\text{-}2}$ as $\mathrm{Attn}_{2\text{-}2}^{(h)}$ and the weight matrices from the h-th head as $W_K^{2,h}, W_Q^{2,h}, W_V^{2,h}$ and $W_O^{2,h}$.

We construct the first head of $\mathrm{Attn}_{2\text{-}2}$ as

$$W_O^{2,1} \coloneqq \underbrace{\begin{bmatrix} I_d \\ 0_{d_Z\times d} \end{bmatrix}}_{(d+d_Z)\times d} \tag{35}$$

$$W_V^{2,1} \coloneqq \underbrace{\begin{bmatrix} W_V^{\mathrm{s}} & 0_{d\times n} & 0_{d\times d_Z} & 0_{d\times d} \end{bmatrix}}_{d\times((2d_h+d)+n+d_Z+d)} \tag{36}$$

$$W_K^{2,1} \coloneqq \underbrace{\begin{bmatrix} W_K^{\mathrm{s}} & 0_{d\times n} & 0_{d\times d_Z} & 0_{d\times d} \end{bmatrix}}_{d_h\times((2d_h+d)+n+d_Z+d)} \tag{37}$$

$$W_Q^{2,1} \coloneqq \underbrace{\begin{bmatrix} W_Q^{\mathrm{s}} & 0_{d\times n} & 0_{d\times d_Z} & 0_{d\times d} \end{bmatrix}}_{d_h\times((2d_h+d)+n+d_Z+d)}, \tag{38}$$

where the dimensions of $W_K^{\mathrm{s}}, W_Q^{\mathrm{s}}$ and W_V^{s} follow from the proof of Theorem C.3, and we set $d_V \to d$.

The construction provides us with

$$W_V^{2,1}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix} = \underbrace{\begin{bmatrix}W_V^{\mathrm{s}} & 0_{d\times n} & 0_{d\times d_Z} & 0_{d\times d}\end{bmatrix}}_{d\times((2d_h+d)+n+d_Z+d)} \underbrace{\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} = \underbrace{W_V^{\mathrm{s}}}_{d\times(2d_h+d)}\cdot\underbrace{\mathrm{Attn}(X_p)}_{(2d_h+d)\times n} \quad \text{(By (36))}$$

$$W_K^{2,1}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix} = \underbrace{\begin{bmatrix}W_K^{\mathrm{s}} & 0_{d_h\times n} & 0_{d_h\times d_Z} & 0_{d_h\times d}\end{bmatrix}}_{d_h\times((2d_h+d)+n+d_Z+d)} \underbrace{\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} = \underbrace{W_K^{\mathrm{s}}}_{d_h\times(2d_h+d)}\cdot\underbrace{\mathrm{Attn}(X_p)}_{(2d_h+d)\times n} \quad \text{(By (37))}$$

$$W_Q^{2,1}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix} = \underbrace{\begin{bmatrix}W_Q^{\mathrm{s}} & 0_{d_h\times n} & 0_{d_h\times d_Z} & 0_{d_h\times d}\end{bmatrix}}_{d_h\times((2d_h+d)+n+d_Z+d)} \underbrace{\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} = \underbrace{W_Q^{\mathrm{s}}}_{d_h\times(2d_h+d)}\cdot\underbrace{\mathrm{Attn}(X_p)}_{(2d_h+d)\times n}. \quad \text{(By (38))}$$

Thus, we have

$$\begin{aligned}
&\mathrm{Attn}^{(1)}_{2\text{-}2}(\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})\\
&= W_O^{2,1}\cdot W_V^{2,1}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}\cdot\sigma_\beta((W_K^{2,1}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})^\top W_Q^{2,1}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})\\
&= \underbrace{\begin{bmatrix}I_d\\ 0_{d_Z\times d}\end{bmatrix}}_{(d+d_Z)\times d}\cdot\underbrace{W_V^{\mathrm{s}}\mathrm{Attn}(X_p)}_{d\times n}\cdot\underbrace{\sigma_\beta((W_K^{\mathrm{s}}\mathrm{Attn}(X_p))^\top W_Q^{\mathrm{s}}\mathrm{Attn}(X_p))}_{n\times n} \quad \text{(By (35))}\\
&= \begin{bmatrix}W_V^{s}\mathrm{Attn}(X_p)\cdot\sigma_\beta((W_K^{\mathrm{s}}\mathrm{Attn}(X_p))^\top W_Q^{s}\mathrm{Attn}(X_p))\\ 0_{d_Z\times n}\end{bmatrix}\\
&= \begin{bmatrix}\mathrm{Attn}_{\mathrm{s}}\circ\mathrm{Attn}(X_p)\\ 0_{d_Z\times n}\end{bmatrix},
\end{aligned} \tag{39}$$

where β denotes the softmax temperature of the first layer emulator in Theorem C.3.

Next, we construct the second head of $\mathrm{Attn}_{2\text{-}2}$ as

$$W_O^{2,2} \coloneqq \underbrace{\begin{bmatrix}0_{d\times d_Z}\\ I_{d_Z}\end{bmatrix}}_{(d+d_Z)\times d_Z} \tag{40}$$

$$W_V^{2,2} \coloneqq \underbrace{\begin{bmatrix}0_{d_Z\times(2d_h+d)} & 0_{d_Z\times n} & I_{d_Z} & 0_{d_Z\times d}\end{bmatrix}}_{d_Z\times((2d_h+d)+n+d_Z+d)} \tag{41}$$

$$W_K^{2,2} \coloneqq \underbrace{\begin{bmatrix}0_{n\times(2d_h+d)} & I_n & 0_{n\times d_Z} & 0_{n\times d}\end{bmatrix}}_{n\times((2d_h+d)+n+d_Z+d)} \tag{42}$$

$$W_Q^{2,2} \coloneqq \underbrace{\begin{bmatrix}0_{n\times(2d_h+d)} & I_n & 0_{n\times d_Z} & 0_{n\times d}\end{bmatrix}}_{n\times((2d_h+d)+n+d_Z+d)}, \tag{43}$$

such that

$$W_V^{2,2}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix} = \underbrace{\begin{bmatrix}0_{d_Z\times(2d_h+d)} & 0_{d_Z\times n} & I_{d_Z} & 0_{d_Z\times d}\end{bmatrix}}_{d_Z\times((2d_h+d)+n+d_Z+d)} \underbrace{\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} = \underbrace{Z}_{d_Z\times n}\underbrace{\sigma_{\beta_2}(I_n)}_{n\times n} \quad \text{(By (41))}$$

$$W_K^{2,2}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix} = \underbrace{\begin{bmatrix}0_{n\times(2d_h+d)} & I_n & 0_{n\times d_Z} & 0_{n\times d}\end{bmatrix}}_{n\times((2d_h+d)+n+d_Z+d)} \underbrace{\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} = I_n \quad \text{(By (42))}$$

$$W_Q^{2,2}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix} = \underbrace{\begin{bmatrix}0_{n\times(2d_h+d)} & I_n & 0_{n\times d_Z} & 0_{n\times d}\end{bmatrix}}_{n\times((2d_h+d)+n+d_Z+d)} \underbrace{\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} = I_n. \quad \text{(By (42))}$$

Thus, we have

$$\begin{aligned}
&\mathrm{Attn}^{(2)}_{2\text{-}2}(\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})\\
&= W_O^{2,2}\cdot W_V^{2,2}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}\cdot\sigma_{\beta_4}((W_K^{2,2}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})^\top W_Q^{2,2}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})\\
&= \underbrace{\begin{bmatrix}0_{d\times d_Z}\\ I_{d_Z}\end{bmatrix}}_{(d+d_Z)\times d_Z}\cdot\underbrace{Z\sigma_{\beta_2}(I_n)}_{d_Z\times n}\cdot\underbrace{\sigma_{\beta_4}(I_n^\top I_n)}_{n\times n} \quad \text{(By (40))}\\
&= \underbrace{\begin{bmatrix}0_{d\times n}\\ Z\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n)\end{bmatrix}}_{(d+d_Z)\times n}. \qquad (44)
\end{aligned}$$

Thirdly, we construct the third head of $\mathrm{Attn}_{2\text{-}2}$ as

$$W_O^{2,3} := \underbrace{\begin{bmatrix}I_d\\ 0_{d_Z\times d}\end{bmatrix}}_{(d+d_Z)\times d} \qquad (45)$$

$$W_V^{2,3} := \underbrace{\begin{bmatrix}0_{d\times(2d_h+d)} & 0_{d\times n} & 0_{d\times d_Z} & I_d\end{bmatrix}}_{d\times((2d_h+d)+n+d_Z+d)} \qquad (46)$$

$$W_K^{2,3} := \underbrace{\begin{bmatrix}0_{n\times(2d_h+d)} & I_n & 0_{n\times d_Z} & 0_{n\times d}\end{bmatrix}}_{n\times((2d_h+d)+n+d_Z+d)} \qquad (47)$$

$$W_Q^{2,3} := \underbrace{\begin{bmatrix}0_{n\times(2d_h+d)} & I_n & 0_{n\times d_Z} & 0_{n\times d}\end{bmatrix}}_{n\times((2d_h+d)+n+d_Z+d)}, \qquad (48)$$

such that

$$W_V^{2,3}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix} = \underbrace{\begin{bmatrix}0_{d\times(2d_h+d)} & 0_{d\times n} & 0_{d\times d_Z} & I_d\end{bmatrix}}_{d\times((2d_h+d)+n+d_Z+d)} \underbrace{\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} = \underbrace{X}_{d\times n}\underbrace{\sigma_{\beta_3}(I_n)}_{n\times n} \quad \text{(By (46))}$$

$$W_K^{2,3}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix} = \underbrace{\begin{bmatrix}0_{n\times(2d_h+d)} & I_n & 0_{n\times d_Z} & 0_{n\times d}\end{bmatrix}}_{n\times((2d_h+d)+n+d_Z+d)} \underbrace{\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} = I_n \quad \text{(By (47))}$$

$$W_Q^{2,3}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix} = \underbrace{\begin{bmatrix}0_{n\times(2d_h+d)} & I_n & 0_{n\times d_Z} & 0_{n\times d}\end{bmatrix}}_{n\times((2d_h+d)+n+d_Z+d)} \underbrace{\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}}_{((2d_h+d)+n+d_Z+d)\times n} = I_n. \quad \text{(By (48))}$$

Thus, we have

$$\begin{aligned}
&\mathrm{Attn}^{(3)}_{2\text{-}2}(\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})\\
&= W_O^{2,3}\cdot W_V^{2,3}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix}\cdot\sigma_{\beta_5}((W_K^{2,3}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})^\top W_Q^{2,3}\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})\\
&= \underbrace{\begin{bmatrix}I_d\\ 0_{d_Z\times d}\end{bmatrix}}_{(d+d_Z)\times d}\cdot\underbrace{X\sigma_{\beta_3}(I_n)}_{d\times n}\cdot\underbrace{\sigma_{\beta_5}(I_n^\top I_n)}_{n\times n} \quad \text{(By (45))}\\
&= \underbrace{\begin{bmatrix}X\sigma_{\beta_3}(I_n)\sigma_{\beta_5}(I_n)\\ 0_{d_Z\times n}\end{bmatrix}}_{(d+d_Z)\times n}. \qquad (49)
\end{aligned}$$

Finally, all of the heads from $\mathrm{Attn}_{2\text{-}2}$ give us

$$\begin{aligned}
&\mathrm{Attn}_{2\text{-}2}(\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})\\
&= \sum_{i=1}^{3}\mathrm{Attn}^{(h)}_{2\text{-}2}(\begin{bmatrix}\mathrm{Attn}(X_p)\\ I_n\\ Z\sigma_{\beta_2}(I_n)\\ X\sigma_{\beta_3}(I_n)\end{bmatrix})\\
&= \underbrace{\begin{bmatrix}\mathrm{Attn}_{\mathrm{s}}\circ\mathrm{Attn}(X_p)\\ 0_{d_Z\times n}\end{bmatrix}}_{(d+d_Z)\times n} + \underbrace{\begin{bmatrix}0_{d\times n}\\ Z\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n)\end{bmatrix}}_{(d+d_Z)\times n} + \underbrace{\begin{bmatrix}X\sigma_{\beta_3}(I_n)\sigma_{\beta_5}(I_n)\\ 0_{d_Z\times n}\end{bmatrix}}_{(d+d_Z)\times n} \quad \text{(By (39), (44) and (49))}\\
&= \underbrace{\begin{bmatrix}X\sigma_{\beta_3}(I_n)\sigma_{\beta_5}(I_n)+\mathrm{Attn}_{\mathrm{s}}\circ\mathrm{Attn}(X_p)\\ Z\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n)\end{bmatrix}}_{(d+d_Z)\times n}
\end{aligned}$$

Step 3. In this part of our proof, our goal is to compute the error between the outputs of $\mathrm{Attn}_{2\text{-}2} \circ \mathrm{Attn}_{2\text{-}1}$ and the target. Specifically, we compute

$$\begin{aligned}
&\|\mathrm{Attn}_{2\text{-}2} \circ \mathrm{Attn}_{2\text{-}1}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) - \begin{bmatrix} X + W_V X \cdot \sigma_\beta((W_K X)^\top W_Q X) \\ Z \end{bmatrix} \|_\infty \\
=& \| \begin{bmatrix} X\sigma_{\beta_3}(I_n)\sigma_{\beta_5}(I_n) + \mathrm{Attn}_{\mathrm{s}} \circ \mathrm{Attn}(X_p) \\ Z\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n) \end{bmatrix} - \begin{bmatrix} X + W_V X \cdot \sigma_\beta((W_K X)^\top W_Q X) \\ Z \end{bmatrix} \|_\infty \\
\leq& \| \begin{bmatrix} X\sigma_{\beta_3}(I_n)\sigma_{\beta_5}(I_n) + \mathrm{Attn}_{\mathrm{s}} \circ \mathrm{Attn}(X_p) \\ Z\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n) \end{bmatrix} - \begin{bmatrix} X + \mathrm{Attn}_{\mathrm{s}} \circ \mathrm{Attn}(X_p) \\ Z \end{bmatrix} \|_\infty \\
&+ \| \begin{bmatrix} X + \mathrm{Attn}_{\mathrm{s}} \circ \mathrm{Attn}(X_p) \\ Z \end{bmatrix} - \begin{bmatrix} X + W_V X \cdot \sigma_\beta((W_K X)^\top W_Q X) \\ Z \end{bmatrix} \|_\infty && \text{(By triangle inequality)} \\
=& \| \begin{bmatrix} X\sigma_{\beta_3}(I_n)\sigma_{\beta_5}(I_n) - X \\ Z\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n) - Z \end{bmatrix} \|_\infty + \epsilon_e && \text{(By Theorem C.3)} \\
\leq& \|Z(\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n) - I_n)\|_\infty + \|X(\sigma_{\beta_3}(I_n)\sigma_{\beta_5}(I_n) - I_n)\|_\infty + \epsilon_e, && (50)
\end{aligned}$$

and (50) follows from the triangle inequality.

In the following proof, we treat the two error terms associated with X and Z using identical computations. We analyze the one associated with Z to demonstrate our procedure and provide the result of the one associated with X directly.

To proceed with our proof, we define

$$\begin{aligned}
E_1 &\coloneqq \sigma_{\beta_2}(I_n) - I_n, \\
E_2 &\coloneqq \sigma_{\beta_4}(I_n) - I_n,
\end{aligned}$$

such that

$$\begin{aligned}
&\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n) - I_n \\
=& (E_1 + I_n) \cdot (E_2 + I_n) - I_n \\
=& E_1 \cdot E_2 + E_1 + E_2. && (51)
\end{aligned}$$

Thus, we have

$$\begin{aligned}
&\|Z(\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n) - I_n)\|_\infty \\
=& \|Z(E_1 \cdot E_2 + E_1 + E_2)\|_\infty && \text{(By (51))} \\
\leq& \underbrace{\|ZE_1E_2\|_\infty}_{\coloneqq (I)} + \underbrace{\|ZE_1\|_\infty}_{\coloneqq (II)} + \underbrace{\|ZE_2\|_\infty}_{\coloneqq (III)}, && (52)
\end{aligned}$$

where the last line follows from triangle inequality.

As a result, our goal is to bound each of $(I), (II)$ and (III), and then aggregate the results to obtain the error bound associated with Z. We treat term (II) and term (III) with identical computation, and our analysis of term (I) relies on the results from (II) and (III).

We start with the term (II). To bound the term (II), we analyze each entry (i, j) from (II)

$$\begin{aligned}
&|(ZE_1)_{ij}| \\
\leq& \sum_{k=1}^{n} |Z_{ik}| \cdot |(E_1)_{kj}| && \text{(By triangle inequality)} \\
\leq& \sum_{k=1}^{n} \|Z\|_\infty \|E_1\|_\infty && \text{(By } |Z_{ik}| \leq \|Z\|_\infty \text{ and } |(E_1)_{kj}| \leq \|E_1\|_\infty) \\
=& n\|Z\|_\infty \|E_1\|_\infty \\
\leq& nB_Z \|E_1\|_\infty. && \text{(By } \|Z\|_\infty \leq B_Z)
\end{aligned}$$

Therefore, $nB_Z\|E_1\|_\infty$ bounds every entry from (II) and gives us

$$(II) \leq nB_Z\|E_1\|_\infty. \tag{53}$$

To further analyze (53), we examine each entry from E_1. By definition, E_1 has all diagonal entries equal to the same value, and similarly for off-diagonal entries. Therefore, to examine each entry from E_1, we consider two cases: the diagonal and off-diagonal entries. We start with the diagonal ones.

$$\begin{aligned}
&|(E_1)_{ii}| \\
=&|(\sigma_{\beta_2}(I_n) - I_n)_{ii}| \\
=&|\frac{e^{\beta_2}}{e^{\beta_2}+(n-1)} - 1| && \text{(By the definition of } \sigma_\beta(\cdot)\text{. i.e. (1))} \\
=&|\frac{-(n-1)}{e^{\beta_2}+(n-1)}| \\
=&\frac{(n-1)}{e^{\beta_2}+(n-1)}.
\end{aligned}$$

Concerning off-diagonal terms, we have

$$\begin{aligned}
&|(\sigma_{\beta_2}(I_n) - I_n)_{ij}| \\
=&|\frac{1}{e^{\beta_2}+(n-1)} - 0| && \text{(By the definition of } \sigma_\beta(\cdot)\text{. i.e. (1))} \\
=&|\frac{1}{e^{\beta_2}+(n-1)}| \\
=&\frac{1}{e^{\beta_2}+(n-1)} \\
\leq&\frac{(n-1)}{e^{\beta_2}+(n-1)} \\
=&|(\sigma_{\beta_2}(I_n) - I_n)_{jj}|.
\end{aligned}$$

Therefore, the magnitudes of all E_1 entries are no greater than the diagonal ones, and we obtain

$$\|E_1\|_\infty \leq \frac{(n-1)}{e^{\beta_2}+(n-1)}. \tag{54}$$

As a result,

$$(II) \leq \frac{n(n-1)B_Z}{e^{\beta_2}+(n-1)}, \tag{55}$$

where (55) follows from (53) and (54).

Next, we treat term (III) with the same proof steps as those for (II), and we have

$$(III) = \|ZE_2\|_\infty \leq \frac{n(n-1)B_Z}{e^{\beta_4}+(n-1)}. \tag{56}$$

As for term (I), we analyze each entry (i,j) from (I)

$$\begin{aligned}
&|(ZE_1E_2)_{ij}| \\
\leq&\sum_{k=1}^{n}|Z_{ik}|\cdot|(E_1E_2)_{kj}| && \text{(By triangle inequality)} \\
\leq&\sum_{k=1}^{n}\|Z\|_\infty\|E_1E_2\|_\infty && \text{(By } |Z_{ik}| \leq \|Z\|_\infty \text{ and } |(E_1E_2)_{kj}| \leq \|E_1E_2\|_\infty\text{)}
\end{aligned}$$

$$
\begin{aligned}
&= n\|Z\|_\infty \|E_1E_2\|_\infty \\
&\le nB_Z\|E_1E_2\|_\infty. && \text{(By } \|Z\|_\infty \le B_Z\text{)}
\end{aligned}
$$

Therefore, $nB_Z\|E_1E_2\|_\infty$ bounds every entry from (I) and gives us

$$
(I) \le nB_Z\|E_1E_2\|_\infty. \tag{57}
$$

To further analyze (57), we examine each entry from E_1E_2

$$
\begin{aligned}
&|(E_1E_2)_{ij}| \\
&\le \sum_{k=1}^{n} |(E_1)_{ik}| \cdot |(E_2)_{kj}| && \text{(By triangle inequality)} \\
&\le \sum_{k=1}^{n} \|E_1\|_\infty \|E_2\|_\infty && \text{(By } |(E_1)_{ik}| \le \|E_1\|_\infty \text{ and } |(E_2)_{kj}| \le \|E_2\|_\infty\text{)} \\
&\le n \cdot \frac{(n-1)}{e^{\beta_2} + (n-1)} \cdot \frac{(n-1)}{e^{\beta_4} + (n-1)}. && \text{(By (54))}
\end{aligned}
$$

The above bound on $|(E_1E_2)_{ij}|$ holds for all (i, j). Therefore, we have

$$
\|E_1E_2\|_\infty \le n \cdot \frac{(n-1)}{e^{\beta_2} + (n-1)} \cdot \frac{(n-1)}{e^{\beta_4} + (n-1)}, \tag{58}
$$

and

$$
(I) \le B_Z \cdot \frac{n(n-1)}{e^{\beta_2} + (n-1)} \cdot \frac{n(n-1)}{e^{\beta_4} + (n-1)}, \tag{59}
$$

where (59) follows from (57) and (58).

Combining the analysis results of (I), (II), and (III), we have

$$
\begin{aligned}
&(I) + (II) + (III) \\
&\le \underbrace{B_Z \cdot \frac{n(n-1)}{e^{\beta_2} + (n-1)} \cdot \frac{n(n-1)}{e^{\beta_4} + (n-1)}}_{\text{from } (I)} + \underbrace{\frac{n(n-1)B_Z}{e^{\beta_2} + (n-1)}}_{\text{from } (II)} + \underbrace{\frac{n(n-1)B_Z}{e^{\beta_4} + (n-1)}}_{\text{from } (III)},
\end{aligned} \tag{60}
$$

and (60) follows from (55), (56) and (59).

For any $\widehat{\epsilon}_1 > 0$, we take

$$
\beta_2,\ \beta_4 \ge \ln\left((n-1)(B_Z\frac{n}{\widehat{\epsilon}_1} - 1)\right), \tag{61}
$$

then we have

$$
\begin{aligned}
(II) &\le \widehat{\epsilon}_1, \\
(III) &\le \widehat{\epsilon}_1,
\end{aligned}
$$

and

$$
\begin{aligned}
&(I) \\
&\le B_Z \cdot \frac{n(n-1)}{e^{\beta_2} + (n-1)} \cdot \frac{n(n-1)}{e^{\beta_4} + (n-1)} \\
&\le B_Z \cdot \frac{n(n-1)}{(n-1)(B_Z\frac{n}{\widehat{\epsilon}_1} - 1) + (n-1)} \cdot \frac{n(n-1)}{(n-1)(B_Z\frac{n}{\widehat{\epsilon}_1} - 1) + (n-1)} && \text{(By (61))}
\end{aligned}
$$

$$= B_Z \cdot \frac{\widehat{\epsilon}_1}{B_Z} \cdot \frac{\widehat{\epsilon}_1}{B_Z}$$
$$= \frac{\widehat{\epsilon}_1^2}{B_Z}.$$

Thus,

$$\|Z(\sigma_{\beta_2}(I_n)\sigma_{\beta_4}(I_n) - I)\|_\infty \leq 2\widehat{\epsilon}_1 + \frac{\widehat{\epsilon}_1^2}{B_Z}. \tag{62}$$

As for the error associated with X, we treat the error using the same analysis steps as those for Z. For any $\widehat{\epsilon}_2 > 0$, we take

$$\beta_3,\ \beta_5 \geq \ln\left((n-1)(B_X \frac{n}{\widehat{\epsilon}_2} - 1)\right),$$

and we have

$$\|X(\sigma_{\beta_3}(I_n)\sigma_{\beta_5}(I_n) - I)\|_\infty \leq 2\widehat{\epsilon}_2 + \frac{\widehat{\epsilon}_2^2}{B_X}. \tag{63}$$

Finally, we have

$$\|\mathrm{Attn}_{2\text{-}2} \circ \mathrm{Attn}_{2\text{-}1}(\begin{bmatrix} X_p \\ Z \end{bmatrix}) - \begin{bmatrix} X + W_V X \cdot \sigma_\beta((W_K X)^\top W_Q X) \\ Z \end{bmatrix} \|_\infty \leq 2\widehat{\epsilon}_1 + 2\widehat{\epsilon}_2 + \frac{\widehat{\epsilon}_1^2}{B_Z} + \frac{\widehat{\epsilon}_2^2}{B_X} + \epsilon_e.$$
(By (50), (62) and (63))

Since we are able to make each ϵ arbitrarily small, this completes the proof. □

D.4 Proof of Theorem 3.1

Theorem D.1 (Theorem 3.1 Restated: In-Context Universal Approximation by Transformer)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. Let $W_{\mathrm{FF}_1}, W_{\mathrm{FF}_2}$ be the weight encodings for two feed-forward neural networks $\mathrm{FF}_1, \mathrm{FF}_2$ as in Lemma 3.1. Let $W_{\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}}$ be the weight encoding for a single-head self-attention, as in Definition 3.2 with $H = 1$. Define $W := \begin{bmatrix} W_{\mathrm{FF}_1}^\top & W_{\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}}^\top & I_n & W_{\mathrm{FF}_2}^\top \end{bmatrix}^\top$. Then, for any $\epsilon > 0$, there exists a six-layer attention module Attn_6 such that*

$$\mathrm{d}_p(\mathrm{Attn}_6(\begin{bmatrix} X \\ W \end{bmatrix}), f(X)) \leq \epsilon,$$

where f is a continuous function defined on a compact domain $\mathcal{C} \subset \mathbb{R}^{d\times n}$.

Proof. In the scope of this theorem, we aim to achieve the in-context approximation of a continuous sequence-to-sequence function f. Our strategy is to approximate the surrogate map $\mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1$ from Theorem C.1, and apply Theorem C.1 to approximate f. This strategy rests on a simple rationale. If we obtain an in-context approximation of $\mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1$, then we also obtain an in-context approximation of any continuous sequence-to-sequence function f. To approximate $\mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1$ in context, we design a six-layer multi-head attention module Attn_6. The module Attn_6 receives the weights of FF_1, FF_2, and $\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}$ as part of its input and approximates their combined action in the end. We build Attn_6 by stacking the multi-head attentions from Lemma 3.1, Lemma 3.3, and Lemma 3.2. In more detail, we use Lemma 3.1 to approximate FF_1; we use Lemma 3.3 to approximate $\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}$, and we use Lemma 3.2 to approximate FF_2. Finally, we compose these in-context approximations to approximate the overall map $\mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1$ and apply Theorem C.1 to approximate the target function f at the end.

Step 1. In this part of our proof, we aim to approximate $\mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1(X)$. First, we approximate FF_1 by applying Lemma 3.1. Then we approximate $\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1$ by applying Lemma 3.3. Finally, we approximate $\mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1$ by applying Lemma 3.2.

We start with Lemma 3.1 to approximate FF_1. From Lemma 3.1, we have

$$\|\mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \begin{bmatrix} \mathrm{FF}_1(X) \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}\|_\infty \le \epsilon_1. \tag{64}$$

Here Attn_2^1 denotes the two-layer attention constructed from Lemma 3.1.

Next, we aim to approximate $\mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X)$. Our strategy is to stack the multi-head attentions from Lemma 3.3 and (64). Specifically, we compute

$$\begin{aligned}
&\|\mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X) \\ W_{\mathrm{FF}_2} \end{bmatrix}\|_\infty \\
&\le \|\mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \mathrm{Attn}_2^2(\begin{bmatrix} \mathrm{FF}_1(X) \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix})\|_\infty + \|\mathrm{Attn}_2^2(\begin{bmatrix} \mathrm{FF}_1(X) \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X) \\ W_{\mathrm{FF}_2} \end{bmatrix}\|_\infty,
\end{aligned} \tag{65}$$

where (65) follows from the triangle inequality, and Attn_2^2 denotes the two-layer attention constructed from Lemma 3.3.

To keep our proof clean, we define

$$P_1 \coloneqq \underbrace{\mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix})}_{(d+d_\mathrm{s}+n+d_{\mathrm{FF}_2})\times n}, \quad P_1^* \coloneqq \underbrace{\begin{bmatrix} \mathrm{FF}_1(X) \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}}_{(d+d_\mathrm{s}+n+d_{\mathrm{FF}_2})\times n},$$

where we use d_s and d_{FF_2} to denote the dimensions of $W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}}$ and W_{FF_2} respectively. The actual dimension is at Lemma 3.1 and Lemma 3.3.

Then, (65) becomes

$$\underbrace{\|\mathrm{Attn}_2^2(P_1) - \mathrm{Attn}_2^2(P_1^*)\|_\infty}_{\coloneqq (A)} + \underbrace{\|\mathrm{Attn}_2^2(P_1^*) - \begin{bmatrix} \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X) \\ W_{\mathrm{FF}_2} \end{bmatrix}\|_\infty}_{\coloneqq (B)}. \tag{66}$$

Therefore, to bound the approximation of $\mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1$, we need to bound (A) and (B).

To bound (A), we assume Attn_2^2 to be L_1-Lipschitz

$$\|\mathrm{Attn}_2^2(G) - \mathrm{Attn}_2^2(H)\|_\infty \le L_1\|G - H\|_\infty, \tag{67}$$

where $G, H \in \mathbb{R}^{(d+d_\mathrm{s}+n+d_{\mathrm{FF}_2})\times n}$ denotes any input of matching shape. $L_1 > 0$ is a constant and depends on the parameters of Attn_2^2.

Under the L_1-Lipschitz assumption, we have

$$\begin{aligned}
&(A) \\
&= \|\mathrm{Attn}_2^2(P_1) - \mathrm{Attn}_2^2(P_1^*)\|_\infty && \text{(By the definition of } (A) \text{ from (66))}
\end{aligned}$$

$$\leq L_1 \|P_1 - P_1^*\|_\infty \qquad \text{(By (67))}$$
$$\leq L_1 \epsilon_1, \qquad (68)$$

and (68) follows from (64).

To bound (B), we note that $P_1^* = \left[\mathrm{FF}_1(X);\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}};\ I_n\right]$ has the same form as X_p in Lemma 3.3, so we utilize Lemma 3.3 as follows

$$(B)$$
$$= \|\mathrm{Attn}_2^2(\begin{bmatrix} \mathrm{FF}_1(X) \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X) \\ W_{\mathrm{FF}_2} \end{bmatrix} \|_\infty \qquad \text{(By the definition of } (B) \text{ from (66))}$$
$$\leq \epsilon_3, \qquad (69)$$

where (69) follows from Lemma 3.3.

Combining (68) and (69), we have

$$\|\mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X) \\ W_{\mathrm{FF}_2} \end{bmatrix} \|_\infty \leq L_1\epsilon_1 + \epsilon_3 \qquad (70)$$

Next, we aim to approximate $\mathrm{FF}_2 \circ \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X)$. Our strategy is to stack the multi-head attentions from Lemma 3.2 and (70). Specifically, we compute

$$\|\mathrm{Attn}_2^3 \circ \mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \mathrm{FF}_2 \circ \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X)\|_\infty$$
$$\leq \|\mathrm{Attn}_2^3 \circ \mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \mathrm{Attn}_2^3(\begin{bmatrix} \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X) \\ W_{\mathrm{FF}_2} \end{bmatrix})\|_\infty +$$
$$\|\mathrm{Attn}_2^3(\begin{bmatrix} \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X) \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \mathrm{FF}_2 \circ \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X)\|_\infty, \qquad (71)$$

where (71) follows from the triangle inequality, and Attn_2^3 denotes the two-layer attention from Lemma 3.2.

Again, to keep our proof clean, we define

$$P_2 \coloneqq \mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_\mathrm{s}^\mathrm{res}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}), \quad P_2^* \coloneqq \begin{bmatrix} \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X) \\ W_{\mathrm{FF}_2} \end{bmatrix},$$

such that (71) becomes

$$\underbrace{\|\mathrm{Attn}_2^3(P_2) - \mathrm{Attn}_2^3(P_2^*)\|_\infty}_{\coloneqq (C)} + \underbrace{\|\mathrm{Attn}_2^3(P_2^*) - \mathrm{FF}_2 \circ \mathrm{Attn}_\mathrm{s}^\mathrm{res} \circ \mathrm{FF}_1(X)\|_\infty}_{\coloneqq (D)} \qquad (72)$$

Therefore, to bound the approximation of $\mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1$, we need to bound the terms (C) and (D).

To bound (C), we assume Attn_2^3 to be L_2-Lipschitz

$$\|\mathrm{Attn}_2^3(G) - \mathrm{Attn}_2^3(H)\|_\infty \leq L_2\|G - H\|_\infty, \tag{73}$$

where $G, H \in \mathbb{R}^{(d+d_{\mathrm{FF}_2})\times n}$ denotes any input of matching shape. $L_2 > 0$ is a constant, and it depends on the parameters of Attn_2^3.

Under the L_2-Lipschitz assumption, we have

$$\begin{aligned}
&(C)\\
&= \|\mathrm{Attn}_2^3(P_2) - \mathrm{Attn}_2^3(P_2^*)\|_\infty && \text{(By the definition of } (C) \text{ in (72))}\\
&\leq L_2\|P_2 - P_2^*\|_\infty && \text{(By (73))}\\
&\leq L_2(L_1\epsilon_1 + \epsilon_3), && (74)
\end{aligned}$$

and (74) follows from (70).

To bound (D), we utilize Lemma 3.2, so we have

$$(D) \leq \epsilon_2 \tag{75}$$

Combining (74) and (75), we have

$$\|\mathrm{Attn}_2^3 \circ \mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}) - \mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1(X)\|_\infty \leq L_2(L_1\epsilon_1 + \epsilon_3) + \epsilon_2. \tag{76}$$

Up to this point, we are capable of approximating $\mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1$ arbitrarily closely.

Next, to prepare for the approximation of f, let $E_p \in \mathbb{R}^{d\times n}$ be the positional encoding matrix, and we define

$$\begin{aligned}
h(X) &\coloneqq \mathrm{Attn}_2^3 \circ \mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X + E_p \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}} \\ I_n \\ W_{\mathrm{FF}_2} \end{bmatrix}),\\
g(X) &\coloneqq \mathrm{FF}_2 \circ \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1(X + E_p),
\end{aligned}$$

such that

$$\|h(X) - g(X)\|_\infty \leq L_2(L_1\epsilon_1 + \epsilon_3) + \epsilon_2, \tag{77}$$

where (77) follows from (76).

Step 2. In this part of our proof, we aim to utilize (77) and Theorem C.1 to derive the approximation of f.

Specifically, we compute

$$\begin{aligned}
&\mathrm{d}_p(h, f)\\
&\leq \mathrm{d}_p(h, g) + \mathrm{d}_p(g, f) && \text{(By Minkowski inequality)}\\
&< \mathrm{d}_p(h, g) + \epsilon_{\mathrm{u}}, && (78)
\end{aligned}$$

where (78) follows from Theorem C.1.

As for $\mathrm{d}_p(h, g)$, we have

$$\mathrm{d}_p(h, g) = \left(\int \|h(X) - g(X)\|_p^p \mathrm{d}X\right)^{\frac{1}{p}}. \qquad \text{(By Definition 2.3)}$$

Therefore, to bound $\mathrm{d}_p(h, g)$, we need to analyze the quantity $\|h(X) - g(X)\|_p^p$.

For the simplicity of presentation, we define

$$E \coloneqq h(X) - g(X),$$

so

$$\|h(X) - g(X)\|_p^p = \|E\|_p^p.$$

Then, we have

$$\begin{aligned}
&\|E\|_p^p \\
&= \sum_{i=1}^{d}\sum_{j=1}^{n} |E_{ij}|^p && \text{(By the definition of } \ell_p \text{ norm)} \\
&\le \sum_{i=1}^{d}\sum_{j=1}^{n} \|E\|_\infty^p && \text{(By } |E_{ij}| \le \|E\|_\infty\text{)} \\
&= dn\|E\|_\infty^p \\
&\le dn(L_2(L_1\epsilon_1 + \epsilon_3) + \epsilon_2)^p, && (79)
\end{aligned}$$

and (79) follows from (77).

Thus,

$$\begin{aligned}
&\mathrm{d}_p(h, g) \\
&= (\int \|h(X) - g(X)\|_p^p \mathrm{d}X)^{\frac{1}{p}} \\
&\le (dn(L_2(L_1\epsilon_1 + \epsilon_3) + \epsilon_2)^p \int \mathrm{d}X)^{\frac{1}{p}}. && \text{(By (79))}
\end{aligned}$$

In Theorem C.1, the authors require the function f to take values on a compact domain, so the quantity $\int \mathrm{d}X$ is finite. Then, we have

$$\mathrm{d}_p(h, g) \le d^{\frac{1}{p}} n^{\frac{1}{p}} (L_2(L_1\epsilon_1 + \epsilon_3) + \epsilon_2)C, \tag{80}$$

where $C \coloneqq (\int \mathrm{d}X)^{\frac{1}{p}}$.

Thus, we have

$$\mathrm{d}_p(h, f) \le Cd^{\frac{1}{p}} n^{\frac{1}{p}} (L_2(L_1\epsilon_1 + \epsilon_3) + \epsilon_2) + \epsilon_{\mathrm{u}} \qquad \text{(By (78) and (80))}$$

Since we are capable of making each ϵ arbitrarily small, we complete the proof. □

In the introduction, we propose three requirements that a satisfactory theory of the prompt-programmable regime should meet. The third one requires a constructive and reusable prompt interface. Here, we modify the construction of W given by (Hu et al., 2024b, Theorem G.1, G.2).

Remark D.1 (Explicit Construction of W Given Function f)**.** *Let $\mathcal{C} \coloneqq [-b, b]^{d\times n}$ be the compact domain. They first discretize the compact domain by forming a uniform grid with granularity g*

$$\Gamma_g \coloneqq -b + \frac{1}{g}\{0, 1, \ldots, 2bg\},$$

and

$$\mathcal{G}_g \coloneqq \{C \in \mathbb{R}^{d\times n} \mid \forall\, i \in [d],\, j \in [n],\, C_{ij} \in \Gamma_g\}.$$

That is, the number of elements in $\mathcal{G}_g$ is $(2bg+1)^{dn}$.

Then, for the input sequence $X \in \mathbb{R}^{d\times n}$, they define the following $\mathbb{R}\to\mathbb{R}$ function to map each entry of X to discrete values

$$\mathrm{quant}_g(z) \coloneqq \begin{cases} -b, & z < -b \\ -b+\frac{1}{g}, & -b \le z < -b+\frac{1}{g} \\ \vdots & \vdots \\ b, & b-\frac{1}{g} \le z. \end{cases}$$

They extend quant_g to an entry-wise map $\mathrm{quant}_g^{d\times n}(X):\mathbb{R}^{d\times n}\to\mathbb{R}^{d\times n}$ by

$$\mathrm{quant}_g^{d\times n}(X)_{ij} \coloneqq \mathrm{quant}_g(X_{ij}) \quad \textit{for all } i\in[d],\ j\in[n].$$

That is, the function $\mathrm{quant}_g^{d\times n}$ maps X to the grid points defined by $\mathcal{G}_g$.

Next, for $X \in \mathbb{R}^{d\times n}\setminus[-b,b]^{d\times n}$, they define the following $\mathbb{R}\to\mathbb{R}$ function to indicate out-of-domain inputs

$$\mathrm{penalty}(z) \coloneqq \begin{cases} -2b, & z < -b \\ 0, & z\in[-b,b] \\ -2b, & z > b. \end{cases}$$

Again, they extend the penalty function to an entry-wise map $\mathrm{penalty}^{d\times n}(X):\mathbb{R}^{d\times n}\to\mathbb{R}^{d\times n}$, where they apply $\mathrm{penalty}(z)$ in an entry-wise manner.

Lastly, they let $B\in\mathbb{R}^{d\times n}$ be a matrix with all entries equal to b, and they define $h_1:\mathbb{R}^{d\times n}\to\mathbb{R}^{d\times n}$ by

$$h_1(X) \coloneqq \underbrace{\mathrm{quant}_g^{d\times n}(X)+B}_{(A)} + \underbrace{dn\cdot\mathrm{penalty}^{d\times n}(X)}_{(B)} \tag{81}$$

To implement (81), they design the following functions to approximate (81). In addition, these functions are implementable with a feed-forward layer. Thus, they encode the weights of these functions into the first feed-forward layer to form the explicit construction of FF_1.

For the first term in (A) of (81), they design

$$\widetilde{f}_1(z) \coloneqq -b+\frac{1}{g}\sum_{t=-bg}^{bg-1}(\mathrm{ReLU}(\frac{z}{\delta}-\frac{t}{\delta g})-\mathrm{ReLU}(\frac{z}{\delta}-\frac{t}{\delta g}-1)), \tag{82}$$

where $\delta>0$ determines how quickly each summand goes from 0 to 1, and then they design

$$f_1(z) \coloneqq \widetilde{f}_1(z) - b\cdot(\mathrm{ReLU}(\frac{z}{\delta}-\frac{b}{\delta})-\mathrm{ReLU}(\frac{z}{\delta}-\frac{b}{\delta}-1)) + b\cdot(\mathrm{ReLU}(-\frac{z}{\delta}-\frac{b}{\delta})-\mathrm{ReLU}(-\frac{z}{\delta}-\frac{b}{\delta}-1)). \tag{83}$$

For term (B), they design

$$f_2(z) \coloneqq -2b(\mathrm{ReLU}(\frac{z-b}{\delta})+\mathrm{ReLU}(\frac{z-b}{\delta}-1)) - 2b(\mathrm{ReLU}(\frac{-z-b}{\delta})+\mathrm{ReLU}(\frac{-z-b}{\delta}-1)). \tag{84}$$

Then, they implement (81) by encoding the weights of (83) and (84) into the following feed-forward neural network FF_1

$$\mathrm{FF}_1 \coloneqq W_{1,2}\cdot\mathrm{ReLU}(W_{1,1}\cdot X + b_{1,1}\mathbb{1}_n^\top) + b_{1,2}\mathbb{1}_n^\top,$$

where

$$W_{1,1} \coloneqq \underbrace{\begin{bmatrix} W_{1,1}^{(1)} \\ W_{1,1}^{(2)} \\ \vdots \\ W_{1,1}^{(d)} \end{bmatrix}}_{d(4bg+8)\times d}, \quad b_{1,1} \coloneqq \underbrace{\begin{bmatrix} b_{1,1}^{(1)} \\ b_{1,1}^{(2)} \\ \vdots \\ b_{1,1}^{(d)} \end{bmatrix}}_{d(4bg+8)\times 1}, \quad W_{1,2} \coloneqq \underbrace{\begin{bmatrix} W_{1,2}^{(1)} & W_{1,2}^{(2)} & \cdots & W_{1,2}^{(d)} \end{bmatrix}}_{d\times d(4bg+8)}, \quad b_{1,2} \coloneqq \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{d\times 1},$$

and they set

$$W_{1,1}^{(1)} \coloneqq \underbrace{\begin{bmatrix} \frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ \frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ \frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ \frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ \frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ -\frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ -\frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ \frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ \frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ -\frac{1}{\delta} & 0 & 0 & \cdots & 0 \\ -\frac{1}{\delta} & 0 & 0 & \cdots & 0 \end{bmatrix}}_{(4bg+8)\times d}, \quad W_{1,1}^{(2)} \coloneqq \underbrace{\begin{bmatrix} 0 & \frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & \frac{1}{\delta} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & \frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & \frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & \frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & -\frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & -\frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & \frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & \frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & -\frac{1}{\delta} & 0 & \cdots & 0 \\ 0 & -\frac{1}{\delta} & 0 & \cdots & 0 \end{bmatrix}}_{(4bg+8)\times d}, \quad \cdots, \quad W_{1,1}^{(d)} \coloneqq \underbrace{\begin{bmatrix} 0 & 0 & 0 & \cdots & \frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & \frac{1}{\delta} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & \frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & \frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & \frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & -\frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & -\frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & \frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & \frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & -\frac{1}{\delta} \\ 0 & 0 & 0 & \cdots & -\frac{1}{\delta} \end{bmatrix}}_{(4bg+8)\times d},$$

with

$$b_{1,1}^{(i)} \coloneqq \underbrace{\begin{bmatrix} \frac{b}{\delta} \\ \frac{b}{\delta} - 1 \\ \vdots \\ -\frac{b}{\delta} + \frac{1}{\delta g} \\ -\frac{b}{\delta} - 1 + \frac{1}{\delta g} \\ -\frac{b}{\delta} \\ -\frac{b}{\delta} - 1 \\ -\frac{b}{\delta} \\ -\frac{b}{\delta} - 1 \\ -\frac{b}{\delta} \\ -\frac{b}{\delta} - 1 \\ -\frac{b}{\delta} \\ -\frac{b}{\delta} - 1 \end{bmatrix}}_{(4bg+8)\times 1}.$$

That is, $W_{1,1}^{(i)}$ has all-zero columns except for the i-th column. The entries are $1/\delta$ in the rows corresponding to $\mathrm{ReLU}(\frac{z}{\delta} - \cdot)$ terms in (83) and (84), and $-1/\delta$ in the rows corresponding to $\mathrm{ReLU}(-\frac{z}{\delta} - \cdot)$ terms.

Then, for all $i \in [d]$, they construct $W_{1,2}^{(i)}$ as

$$W_{1,2}^{(i)} \coloneqq \underbrace{e_i^{(d)}}_{d\times 1} \cdot \underbrace{\begin{bmatrix} \frac{1}{g} & -\frac{1}{g} & \cdots & \frac{1}{g} & -\frac{1}{g} & -b & b & b & -b & -2bdn & -2bdn & -2bdn & -2bdn \end{bmatrix}}_{1\times(4bg+8)}.$$

That is, $W_{1,2}^{(i)}$ has only one non-zero row i.

Next, to prepare for the construction of the attention layer, we need to describe the properties of the output from term (A) in (81). For this purpose, we restate two helper definitions from (Hu et al., 2024b).

Definition D.1 (Vocabulary, Definition G.1 of (Hu et al., 2024b))**.** *For each $i \in [N]$, define the i-th vocabulary set by*

$$\mathcal{V}^{(i)} \coloneqq \bigcup_{k\in[n]} X_{:,k}^{(i)} \subset \mathbb{R}^d,$$

and define the whole vocabulary set by

$$\mathcal{V} \coloneqq \bigcup_{i\in[N]} \mathcal{V}^{(i)} \subset \mathbb{R}^d.$$

Definition D.2 (Tokenwise Separateness, Definition G.2 of (Hu et al., 2024b)). *Let $X^{(1)}, \ldots, X^{(N)} \in \mathbb{R}^{d \times n}$ be embeddings. We say that $X^{(1)}, \ldots, X^{(N)}$ are tokenwise $(\gamma_{\min}, \gamma_{\max}, \epsilon)$-separated if the following three conditions hold:*

(i) For any $i \in [N]$ and $k \in [n]$, $\|X^{(i)}_{:,k}\|_2 > \gamma_{\min}$.

(ii) For any $i \in [N]$ and $k \in [n]$, $\|X^{(i)}_{:,k}\|_2 < \gamma_{\max}$.

(iii) For any $i, j \in [N]$ and $k, l \in [n]$, if $X^{(i)}_{:,k} \neq X^{(j)}_{:,l}$ holds, then $\|X^{(i)}_{:,k} - X^{(j)}_{:,l}\|_2 > \epsilon$.

Note that when only conditions (ii) and (iii) hold, we denote this as (γ, ϵ)-separateness. Moreover, if only condition (iii) holds, we denote it as (ϵ)-separateness.

They use $\mathcal{G}_g^\circ$ to denote the set of all possible output sequences from term (A) in (81). Also, let $\widetilde{\mathcal{G}}_g$ be

$$\widetilde{\mathcal{G}}_g \coloneqq \{G \in \mathcal{G}_g^\circ \mid G_{:,i} \neq G_{:,j} \ \text{ for all } \ i, j \in [n] \ \text{ and } \ i \neq j\}.$$

By construction, $\widetilde{\mathcal{G}}_g$ is finite, and the sequences in $\widetilde{\mathcal{G}}_g$ are tokenwise $(1/g, 2b\sqrt{d}, 1/g)$-separated.

Besides, let $\mathcal{V}$ be the set of all tokens from $\widetilde{\mathcal{G}}_g$. Since $\widetilde{\mathcal{G}}_g$ is finite, $\mathcal{V}$ is also finite.

Again, to prepare for the construction of the attention layer, we restate a helper lemma from (Park et al., 2021).

Lemma D.4 (Lemma 13 of (Park et al., 2021)). *For any finite subset $\mathcal{X} \subset \mathbb{R}^d$, there exists at least one unit vector $u \in \mathbb{R}^d$ such that*

$$\frac{1}{|\mathcal{X}|^2}\sqrt{\frac{8}{\pi d}}\,\|x - x'\|_2 \ \leq \ |u^\top (x - x')| \ \leq \ \|x - x'\|_2 \tag{85}$$

for any $x, x' \in \mathcal{X}$.

Since $\mathcal{V}$ is a finite set, by Lemma D.4, there exists at least one unit vector $u \in \mathbb{R}^d$ such that (85) holds for any two tokens from $\mathcal{V}$.

Now, we restate their construction of the attention layer.

1. *We state the dimension first. They let $W_K, W_Q, W_V \in \mathbb{R}^{s \times d}$ and $W_O \in \mathbb{R}^{d \times s}$.*
2. *Then, set the rank of the weight matrices to be ρ. Any choice of ρ that satisfies $1 \leq \rho \leq \min\{d, s\}$ works.*
3. *Next, for each $i \in [\rho]$, pick vectors $a_i, b_i \in \mathbb{R}^s$ such that*

$$|a_i^\top b_i| \ = \ (|\mathcal{V}| + 1)^4 \frac{d\kappa}{\epsilon \gamma_{\min}},$$

   where they set $\kappa = (4 \ln n)/\beta$. Also, $\epsilon = \gamma_{min} = 1/g$ here.
4. *Then, for each $i \in [\rho]$, choose $u_i, w_i \in \mathbb{R}^d$ to be unit vectors. Moreover, they require that there be at least one index $i^\star \in [\rho]$ such that $u_{i^\star} = u$ and $w_{i^\star} = u$. For $[\rho] \setminus \{i^\star\}$, any unit vector in $\mathbb{R}^d$ works.*
5. *Then, they construct the key and query weight matrices as follows*

$$W_K \coloneqq \sum_{i=1}^{\rho} a_i u_i^\top \in \mathbb{R}^{s \times d}, \qquad W_Q \coloneqq \sum_{i=1}^{\rho} b_i w_i^\top \in \mathbb{R}^{s \times d}.$$

6. *Next, for each $i \in [\rho]$, pick an arbitrary non-zero vector $c_i \in \mathbb{R}^s$ and $o_i \in \mathbb{R}^d$, except for o_1, where they set $o_1 = u$. Then, they construct the value weight matrix as*

$$W_V \coloneqq \sum_{i=1}^{\rho} c_i o_i^\top \in \mathbb{R}^{s \times d}.$$

7. Finally, they require W_O to satisfy the following constraint

$$\|W_O c_i\|_2 = \frac{\epsilon}{4\rho\gamma_{\max}}, \tag{86}$$

where $\epsilon = 1/g$ and $\gamma_{\max} = 2b\sqrt{g}$. They provide one explicit way to achieve (86). For each $i \in [\rho]$, they pick a vector $d_i \in \mathbb{R}^d$ such that

$$\|d_i\|_2 = \frac{\epsilon}{4\rho^2\gamma_{\max}\|c_i\|_2^2},$$

with the same $\epsilon = 1/g$ and $\gamma_{\max} = 2b\sqrt{g}$. Then, they construct

$$W_O \coloneqq \sum_{i=1}^{\rho} d_i {c_i}^\top \in \mathbb{R}^{d\times s}.$$

This completes the construction of the attention layer.

Regarding the construction of FF_2, they first consider a fixed $\bar{C} \in \widetilde{\mathcal{G}}_g$, and they define

$$U \coloneqq \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}(\bar{C}),$$

where $\mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}$ is the single-head attention with a residual connection, specified by the weight matrices constructed in the previous steps.

Additionally, they define

$$S \coloneqq \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}} \circ \mathrm{FF}_1(X).$$

For a fixed $\bar{C}$, they define $h_2 : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$, where

$$h_2(S)_{ij} \coloneqq f(\bar{C} - B)_{ij} \cdot \mathbb{1}_{U_{ij}=S_{ij}} \qquad \text{for all } i \in [d], j \in [n],$$

and f is the target function.

Then, they use a bump function $\mathrm{bump}_R : \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ to achieve the indicator in h_2:

$$\mathrm{bump}_R(S)_{ij} \coloneqq \mathrm{ReLU}(R_{\mathrm{FF}}(S_{ij} - U_{ij}) - 1) - 2\mathrm{ReLU}(R_{\mathrm{FF}}(S_{ij} - U_{ij})) + \mathrm{ReLU}(R_{\mathrm{FF}}(S_{ij} - U_{ij}) + 1),$$

and they choose R_{FF} to scale like $R_{\mathrm{FF}} = \mathcal{O}(\exp\big(640\beta(bg)^{4d+2}d^2\ln n\big))$. This bump function is implementable by a feed-forward layer.

Next, for a fixed $\bar{C}$ and $U = \mathrm{Attn}_{\mathrm{s}}^{\mathrm{res}}(\bar{C})$, they use part of FF_2 to implement h_2. Specifically, they construct

$$W_{2,1}^{(i)} \coloneqq \underbrace{\begin{bmatrix} R_{\mathrm{FF}} & 0 & 0 & \cdots & 0 \\ R_{\mathrm{FF}} & 0 & 0 & \cdots & 0 \\ R_{\mathrm{FF}} & 0 & 0 & \cdots & 0 \\ 0 & R_{\mathrm{FF}} & 0 & \cdots & 0 \\ 0 & R_{\mathrm{FF}} & 0 & \cdots & 0 \\ 0 & R_{\mathrm{FF}} & 0 & \cdots & 0 \\ 0 & 0 & R_{\mathrm{FF}} & \cdots & 0 \\ 0 & 0 & R_{\mathrm{FF}} & \cdots & 0 \\ 0 & 0 & R_{\mathrm{FF}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & R_{\mathrm{FF}} \\ 0 & 0 & 0 & \cdots & R_{\mathrm{FF}} \\ 0 & 0 & 0 & \cdots & R_{\mathrm{FF}} \\ \vdots & \vdots & \vdots & & \vdots \\ R_{\mathrm{FF}} & 0 & 0 & \cdots & 0 \\ R_{\mathrm{FF}} & 0 & 0 & \cdots & 0 \\ R_{\mathrm{FF}} & 0 & 0 & \cdots & 0 \\ 0 & R_{\mathrm{FF}} & 0 & \cdots & 0 \\ 0 & R_{\mathrm{FF}} & 0 & \cdots & 0 \\ 0 & R_{\mathrm{FF}} & 0 & \cdots & 0 \\ 0 & 0 & R_{\mathrm{FF}} & \cdots & 0 \\ 0 & 0 & R_{\mathrm{FF}} & \cdots & 0 \\ 0 & 0 & R_{\mathrm{FF}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & R_{\mathrm{FF}} \\ 0 & 0 & 0 & \cdots & R_{\mathrm{FF}} \\ 0 & 0 & 0 & \cdots & R_{\mathrm{FF}} \end{bmatrix}}_{3dn \times d}, \qquad b_{2,1}^{(i)} \coloneqq \underbrace{\begin{bmatrix} -R_{\mathrm{FF}} \cdot U_{11} \\ -R_{\mathrm{FF}} \cdot U_{11} - 1 \\ -R_{\mathrm{FF}} \cdot U_{11} + 1 \\ -R_{\mathrm{FF}} \cdot U_{21} \\ -R_{\mathrm{FF}} \cdot U_{21} - 1 \\ -R_{\mathrm{FF}} \cdot U_{21} + 1 \\ -R_{\mathrm{FF}} \cdot U_{31} \\ -R_{\mathrm{FF}} \cdot U_{31} - 1 \\ -R_{\mathrm{FF}} \cdot U_{31} + 1 \\ \vdots \\ -R_{\mathrm{FF}} \cdot U_{d1} \\ -R_{\mathrm{FF}} \cdot U_{d1} - 1 \\ -R_{\mathrm{FF}} \cdot U_{d1} + 1 \\ \vdots \\ -R_{\mathrm{FF}} \cdot U_{1n} \\ -R_{\mathrm{FF}} \cdot U_{1n} - 1 \\ -R_{\mathrm{FF}} \cdot U_{1n} + 1 \\ -R_{\mathrm{FF}} \cdot U_{2n} \\ -R_{\mathrm{FF}} \cdot U_{2n} - 1 \\ -R_{\mathrm{FF}} \cdot U_{2n} + 1 \\ -R_{\mathrm{FF}} \cdot U_{3n} \\ -R_{\mathrm{FF}} \cdot U_{3n} - 1 \\ -R_{\mathrm{FF}} \cdot U_{3n} + 1 \\ \vdots \\ -R_{\mathrm{FF}} \cdot U_{dn} \\ -R_{\mathrm{FF}} \cdot U_{dn} - 1 \\ -R_{\mathrm{FF}} \cdot U_{dn} + 1 \end{bmatrix}}_{3dn}.$$

That is, they repeat the first $3d$ rows of $W_1^{(i)}$ for n times.

Then, they construct $W_{2,2}^{(i)}$ to be

$$W_{2,2}^{(i)} \coloneqq \underbrace{W_{2,2}^{\prime\prime(i)} \cdot W_{2,2}^{\prime(i)}}_{d \times 3dn},$$

where

$$W_{2,2}^{\prime(i)} \coloneqq \underbrace{\begin{bmatrix} -2f(\bar{C}-B)_{11} & f(\bar{C}-B)_{11} & f(\bar{C}-B)_{11} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -2f(\bar{C}-B)_{dn} & f(\bar{C}-B)_{dn} & f(\bar{C}-B)_{dn} \end{bmatrix}}_{dn \times 3dn},$$

and

$$W_{2,2}^{\prime\prime(i)} \coloneqq \underbrace{\begin{bmatrix} I_d & I_d & \cdots & I_d \end{bmatrix}}_{d \times dn}.$$

Note that the above construction is for a specific $\bar{C}$, so it stacks the construction for all $\bar{C} \in \widetilde{G}_g$ to obtain the final FF_2

$$\mathrm{FF}_2(S) \coloneqq W_{2,2} \cdot \mathrm{ReLU}(W_{2,1} \cdot S + b_{2,1} \mathbb{1}_n^\top),$$

where

$$W_{2,1} \coloneqq \underbrace{\begin{bmatrix} W_{2,1}^1 \\ W_{2,1}^1 \\ \vdots \\ W_{2,1}^{q_2} \end{bmatrix}}_{3dnq_2 \times d}, \quad b_{2,1} \coloneqq \underbrace{\begin{bmatrix} b_{2,1}^1 \\ b_{2,1}^1 \\ \vdots \\ b_{2,1}^{q_2} \end{bmatrix}}_{3dnq_2 \times 1}, \quad W_{2,2} \coloneqq \underbrace{\begin{bmatrix} W_{2,2}^1 & W_{2,2}^1 & \cdots & W_{2,2}^{q_2} \end{bmatrix}}_{d \times 3dnq_2},$$

and they set $q_2 = (2bg)^{dn}/(n!)$. *That is, they set* $b_{2,2}$ *as a zero vector.*

This completes the construction of the entire network.

D.5 Proof of Lemma 3.4

We state a helper lemma from (Hu et al., 2026).

Lemma D.5 (Attention Preceded by a Linear Transformation Is Still an Attention, Lemma B.3 of (Hu et al., 2026))**.** *Let* $W_K, W_Q \in \mathbb{R}^{d_h \times \widetilde{d}}$, $W_V \in \mathbb{R}^{d_V \times \widetilde{d}}$, *and* $W_O \in \mathbb{R}^{d_O \times d_V}$ *be the weight matrices of a single-head attention* Attn_s. *Let* $A : \mathbb{R}^{d \times n} \to \mathbb{R}^{\widetilde{d} \times n}$ *be a linear transformation. Then,* $\mathrm{Attn}_\mathrm{s} \circ A$ *is still an attention.*

Proof. Let $D \in \mathbb{R}^{d \times n}$ denote any input. Then, we have

$$\mathrm{Attn}_\mathrm{s} \circ A(D) = W_O \cdot W_V AD \cdot \sigma_\beta((W_K AD)^\top (W_Q AD)),$$

and this is a new attention mechanism with parameters $W_K A, W_Q A$ and $W_V A$. □

Definition D.3 (Definition 3.2 Restated: Input Prompt for In-Context Emulation of Multi-Head Attention)**.** *Let* $X \in \mathbb{R}^{d \times n}$ *be the input sequence. Let* $W_K^h, W_Q^h \in \mathbb{R}^{d_h \times d}$ *and* $W_V^h \in \mathbb{R}^{d_V \times d}$ *be the weight matrices of the* h*-th head in the target* H*-head attention. We define the concatenation* $w_h \coloneqq [\underline{W}_K^h; \underline{W}_Q^h; \underline{W}_V^h] \in \mathbb{R}^{d(2d_h + d_V)}$ *and*

$$W_h \coloneqq \mathrm{Enc}(w_h) \in \mathbb{R}^{2d(2d_h + d_V) \times n}.$$

Then, the input prompt for the in-context emulation of a multi-head attention specified by W_K^h, W_Q^h, W_V^h *is*

$$X_p^\mathrm{m} \coloneqq \begin{bmatrix} X^\top & W_1^\top & \cdots & W_H^\top & I_n \end{bmatrix}^\top.$$

Lemma D.6 (Lemma 3.4 Restated: In-Context Emulation of Multi-Head Attention)**.** *Let* $X \in \mathbb{R}^{d \times n}$ *be the input sequence. Let* $\mathrm{Attn} : \mathbb{R}^{d \times n} \to \mathbb{R}^{d_V \times n}$ *be an* H*-head attention specified by* $W_K^h, W_Q^h \in \mathbb{R}^{d_h \times d}$ *and* $W_V^h \in \mathbb{R}^{d_V \times d}$. *Assume* $\|W_K^h X\|_\infty, \|W_Q^h X\|_\infty, \|W_V^h X\|_\infty \le B_{KQV}$ *for some* $B_{KQV} > 0$. *Then, for any* $\epsilon_4 > 0$, *there exists a two-layer attention network* Attn_2 *such that*

$$\|\mathrm{Attn}_2(X_p^\mathrm{m}) - \mathrm{Attn}(X)\|_\infty \le \epsilon_4,$$

where $X_p^\mathrm{m} = [X^\top, W_1^\top, \cdots, W_H^\top, I_n]^\top$ *follows Definition 3.2.*

Proof. We modify the proof of the single-head emulation theorem from Hu et al. (2026).

We use $\mathrm{Attn}^{(h)}$ to label the h-th head from the target Attn, and we use W_K^h, W_Q^h, W_V^h for the weight matrices in $\mathrm{Attn}^{(h)}$.

The proof consists of 4 steps. We first write out the details of the weight encoding of target Attn for later use. Then we use the first layer of Attn_2 to reconstruct the key, query, and value matrices of Attn from the input weight encoding. Thirdly, we use the second layer of Attn_2 to assemble the reconstructed matrices into the attention form. Finally, we compute the error.

For the convenience of presentation, we denote Attn_2 as $\mathrm{Attn}_{2\text{-}2} \circ \mathrm{Attn}_{2\text{-}1}$.

Step 1: Weight Encoding. For the convenience of presentation, we define

$$k_i^h := (W_K^h)_{:,i}^\top \in \mathbb{R}^d, \tag{87}$$
$$q_i^h := (W_Q^h)_{:,i}^\top \in \mathbb{R}^d, \tag{88}$$
$$v_i^h := (W_V^h)_{:,i}^\top \in \mathbb{R}^d, \tag{89}$$

and we define the vectorizations of the weight matrices by Definition 3.1

$$\underline{W}_K^h := \mathrm{vec}(W_K^h) = \underbrace{\begin{bmatrix} k_1^h \\ k_2^h \\ \vdots \\ k_{d_h}^h \end{bmatrix}}_{dd_h \times 1}, \quad \underline{W}_Q^h := \mathrm{vec}(W_Q^h) = \underbrace{\begin{bmatrix} q_1^h \\ q_2^h \\ \vdots \\ q_{d_h}^h \end{bmatrix}}_{dd_h \times 1}, \quad \underline{W}_V^h := \mathrm{vec}(W_V^h) = \underbrace{\begin{bmatrix} v_1^h \\ v_2^h \\ \vdots \\ v_{d_V}^h \end{bmatrix}}_{dd_V \times 1}.$$

Also, we define the concatenation

$$w_h := \begin{bmatrix} \underline{W}_K^h \\ \underline{W}_Q^h \\ \underline{W}_V^h \end{bmatrix} = \underbrace{\begin{bmatrix} k_1^h \\ \vdots \\ k_{d_h}^h \\ q_1^h \\ \vdots \\ q_{d_h}^h \\ v_1^h \\ \vdots \\ v_{d_V}^h \end{bmatrix}}_{d(2d_h+d_V)\times 1}. \tag{90}$$

To keep our proof clean, we define $d_{w_h} := d(2d_h + d_V)$.

We define W_h, W_{Attn} to be

$$W_h := \underbrace{\begin{bmatrix} 0 \cdot w_h & 1 \cdot w_h & 2 \cdot w_h & \cdots & (n-1) \cdot w_h \\ w_h & w_h & w_h & \cdots & w_h \end{bmatrix}}_{2d_{w_h} \times n}, \tag{91}$$

$$W_{\mathrm{Attn}} := \underbrace{\begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_H \end{bmatrix}}_{(\sum_{h=1}^H 2d_{w_h}) \times n}. \tag{92}$$

Thus, our input prompt is

$$X_p^{\mathrm{m}} := \underbrace{\begin{bmatrix} X \\ W_{\mathrm{Attn}} \\ I_n \end{bmatrix}}_{(d+\sum_{h=1}^H 2d_{w_h}+n)\times n}. \tag{93}$$

Step 2: Reconstruction of Target K_h, Q_h, V_h. Again, for the convenience of presentation, we define

$$K_h := \underbrace{W_K^h}_{d_h \times d} \cdot \underbrace{X}_{d \times n}, \quad Q_h := \underbrace{W_Q^h}_{d_h \times d} \cdot X, \quad V_h := \underbrace{W_V^h}_{d_V \times d} \cdot X.$$

In this part, our goal is to build approximators for the target K_h, Q_h, V_h from the input prompt via $\mathrm{Attn}_{2\text{-}1}$.

We note that each row of K_h, Q_h, V_h has the inner product form: $(k_i^h)^\top X, (q_i^h)^\top X, (v_i^h)^\top X$. Therefore, we construct $\mathrm{Attn}_{2\text{-}1}$ by applying Theorem C.4 to each row separately, and $\mathrm{Attn}_{2\text{-}1}$ consists of the heads from Theorem C.4.

We state our parameter choices for Theorem C.4 to approximate each row. Firstly, let a_h and b_h be the minimum and maximum among the inner products $(k_i^h)^\top x_r$, $(q_i^h)^\top x_r$ and $(v_m^h)^\top x_r$, over all $i \in [d_h], m \in [d_V]$ and $r \in [n]$

$$a_h \coloneqq \min_{i,m,r}\{(k_i^h)^\top x_r, (q_i^h)^\top x_r, (v_m^h)^\top x_r\} \quad \text{and} \quad b_h \coloneqq \max_{i,m,r}\{(k_i^h)^\top x_r, (q_i^h)^\top x_r, (v_m^h)^\top x_r\}.$$

That is, a_h and b_h denote the minimum and maximum of the inner products in the h-th head of Attn.

Next, for any $\epsilon_0 > 0$, we choose the number of heads $\widetilde{H}_h$ in Theorem C.4 to be

$$\widetilde{H}_h \coloneqq \lceil \frac{2(b_h - a_h)}{\epsilon_0 (n-2)} \rceil,$$

such that the interpolation error in Theorem C.4 is no more than $\frac{\epsilon_0}{2}$. Here $\widetilde{H}_h$ denotes the number of heads we use for each row of K_h, Q_h and V_h. Since K_h, Q_h each has d_h rows and V_h has d_V rows, we need a total of $\widetilde{H}_h(2d_h + d_V)$ heads to approximate the h-th head of the target Attn. As a result, we use $\sum_{h=1}^H \widetilde{H}_h(2d_h + d_V)$ heads in total to approximate all the key, query, and value matrices from Attn.

Thus, we view $\mathrm{Attn}_{2\text{-}1}$ as groups of $\widetilde{H}_h$-head attentions, and we label each group as $\mathrm{Attn}_{2\text{-}1}^{h,j}$. Here $h \in [H]$ still identifies the h-th head of our target Attn, and $j \in [2d_h + d_V]$ identifies the rows in K_h, Q_h and V_h.

Finally, all $\widetilde{H}_h$ heads in Theorem C.4 require a shared linear mapping $A : \mathbb{R}^{d\times n} \to \mathbb{R}^{(3d+n)\times n}$. Here, we construct this common mapping with head-specific ones $A_{\widetilde{h}} : \mathbb{R}^{(d+\sum_{h=1}^H 2d_{w_h}+n)\times n} \to \mathbb{R}^{(3d+n)\times n}$. $\widetilde{h}$ labels the heads in $\mathrm{Attn}_{2\text{-}1}$, and thus $\widetilde{h} \in [\sum_{h=1}^H \widetilde{H}_h(2d_h + d_V)]$. We use $A_{\widetilde{h}}$ to project the input to the required dimension $\mathbb{R}^{(3d+n)\times n}$ and to select the target k_i^h, q_i^h or v_i^h to approximate the desired linear transformation $(k_i^h)^\top X, (q_i^h)^\top X$ or $(v_i^h)^\top X$.

To that end, we construct $A_{\widetilde{h}}$ to be

$$A_{\widetilde{h}} \coloneqq \underbrace{\begin{bmatrix} I_d & 0_{d\times d_{w_\ell}} & 0_{d\times d_{w_\ell}} & 0_{d\times n} \\ 0_{d\times d} & S_{\widetilde{h}} & 0_{d\times d_{w_\ell}} & 0_{d\times n} \\ 0_{d\times d} & 0_{d\times d_{w_\ell}} & S_{\widetilde{h}} & 0_{d\times n} \\ 0_{n\times d} & 0_{n\times d_{w_\ell}} & 0_{n\times d_{w_\ell}} & I_n \end{bmatrix}}_{(3d+n)\times(d+2d_{w_\ell}+n)} \cdot \underbrace{\begin{bmatrix} I_d & 0_{d\times\sum_{h=1}^H 2d_{w_h}} & 0_{d\times n} \\ 0_{2d_{w_\ell}\times d} & F_{\widetilde{h}} & 0_{2d_{w_\ell}\times n} \\ 0_{n\times d} & 0_{n\times\sum_{h=1}^H 2d_{w_h}} & I_n \end{bmatrix}}_{(d+2d_{w_\ell}+n)\times(d+\sum_{h=1}^H 2d_{w_h}+n)}, \tag{94}$$

where

$$\ell(\widetilde{h}) \coloneqq \sum_{h=1}^H h \cdot \mathbb{1}_{\{\lambda_{h-1} < \widetilde{h} \le \lambda_h\}},$$

with

$$\lambda_h \coloneqq \sum_{i=1}^h \widetilde{H}_i(2d_i + d_V), \quad \lambda_0 \coloneqq 0.$$

Thus, $\ell(\widetilde{h}) \in [H]$ identifies the target head in Attn.

We define $F_{\widetilde{h}}$ as

$$F_{\widetilde{h}} \coloneqq \underbrace{\begin{bmatrix} 0_{2d_{w_\ell}\times\zeta_{\ell-1}} & I_{2d_{w_\ell}} & 0_{2d_{w_\ell}\times(\zeta_H-\zeta_\ell)} \end{bmatrix}}_{2d_{w_\ell}\times\zeta_H}, \tag{95}$$

with

$$\zeta_\ell \coloneqq \sum_{h=1}^\ell 2d_{w_h}, \quad \zeta_0 \coloneqq 0.$$

Thus, $F_{\widetilde{h}}$ picks out the W_ℓ from W_{Attn}.

We define $S_{\widetilde{h}}$ as

$$S_{\widetilde{h}} \coloneqq \underbrace{\begin{bmatrix} 0_{d\times d(j-1)} & I_d & 0_{d\times(d_{w_\ell}-d\cdot j)}\end{bmatrix}}_{d\times d_{w_\ell}}, \tag{96}$$

where

$$j(\widetilde{h}) \coloneqq \lceil \frac{\widetilde{h}-\lambda_{\ell-1}}{\widetilde{H}_\ell} \rceil \in [2d_\ell + d_V].$$

Thus, $S_{\widetilde{h}}$ picks out k_j^ℓ, q_j^ℓ or v_j^ℓ from W_ℓ.

Then, we have

$$\begin{aligned}
&A_{\widetilde{h}} \cdot \begin{bmatrix} X \\ W_{\text{Attn}} \\ I_n \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} I_d & 0_{d\times d_{w_\ell}} & 0_{d\times d_{w_\ell}} & 0_{d\times n} \\ 0_{d\times d} & S_{\widetilde{h}} & 0_{d\times d_{w_\ell}} & 0_{d\times n} \\ 0_{d\times d} & 0_{d\times d_{w_\ell}} & S_{\widetilde{h}} & 0_{d\times n} \\ 0_{n\times d} & 0_{n\times d_{w_\ell}} & 0_{n\times d_{w_\ell}} & I_n \end{bmatrix}}_{(3d+n)\times(d+2d_{w_\ell}+n)} \cdot \underbrace{\begin{bmatrix} I_d & 0_{d\times\sum_{h=1}^H 2d_{w_h}} & 0_{d\times n} \\ 0_{2d_{w_\ell}\times d} & F_{\widetilde{h}} & 0_{2d_{w_\ell}\times n} \\ 0_{n\times d} & 0_{n\times\sum_{h=1}^H 2d_{w_h}} & I_n \end{bmatrix}}_{(d+2d_{w_\ell}+n)\times(d+\sum_{h=1}^H 2d_{w_h}+n)} \cdot \underbrace{\begin{bmatrix} X \\ W_{\text{Attn}} \\ I_n \end{bmatrix}}_{(d+\sum_{h=1}^H 2d_{w_h}+n)\times n} && \text{(By (94) and (93))} \\
&= \underbrace{\begin{bmatrix} I_d & 0_{d\times d_{w_\ell}} & 0_{d\times d_{w_\ell}} & 0_{d\times n} \\ 0_{d\times d} & S_{\widetilde{h}} & 0_{d\times d_{w_\ell}} & 0_{d\times n} \\ 0_{d\times d} & 0_{d\times d_{w_\ell}} & S_{\widetilde{h}} & 0_{d\times n} \\ 0_{n\times d} & 0_{n\times d_{w_\ell}} & 0_{n\times d_{w_\ell}} & I_n \end{bmatrix}}_{(3d+n)\times(d+2d_{w_\ell}+n)} \cdot \underbrace{\begin{bmatrix} X \\ F_{\widetilde{h}}\cdot W_{\text{Attn}} \\ I_n \end{bmatrix}}_{(d+2d_{w_\ell}+n)\times n} && \text{(By matrix multiplication)} \\
&= \underbrace{\begin{bmatrix} X \\ \begin{bmatrix} S_{\widetilde{h}} & 0_{d\times d_{w_\ell}}\end{bmatrix}\cdot F_{\widetilde{h}}\cdot W_{\text{Attn}} \\ \begin{bmatrix} 0_{d\times d_{w_\ell}} & S_{\widetilde{h}}\end{bmatrix}\cdot F_{\widetilde{h}}\cdot W_{\text{Attn}} \\ I_n \end{bmatrix}}_{(3d+n)\times n}, && (97)
\end{aligned}$$

and (97) follows from matrix multiplication.

$F_{\widetilde{h}}\cdot W_{\text{Attn}}$ in (97) expands as

$$\begin{aligned}
F_{\widetilde{h}}\cdot W_{\text{Attn}} &= \underbrace{\begin{bmatrix} 0_{2d_{w_\ell}\times\zeta_{\ell-1}} & I_{2d_{w_\ell}} & 0_{2d_{w_\ell}\times(\zeta_H-\zeta_\ell)}\end{bmatrix}}_{2d_{w_\ell}\times\zeta_H} \cdot \underbrace{\begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_H \end{bmatrix}}_{\zeta_H\times n} && \text{(By (95) and (92))} \\
&= \underbrace{W_\ell}_{2d_{w_\ell}\times n}. && (F_{\widetilde{h}} \text{ selects } W_\ell \text{ from } W_{\text{Attn}})
\end{aligned}$$

Then, $\begin{bmatrix} S_{\widetilde{h}} & 0_{d\times d_{w_\ell}}\end{bmatrix}\cdot F_{\widetilde{h}}\cdot W_{\text{Attn}}$ in (97) expands as

$$\begin{aligned}
&\begin{bmatrix} S_{\widetilde{h}} & 0_{d\times d_{w_\ell}}\end{bmatrix}\cdot W_\ell \\
&= \underbrace{\begin{bmatrix} S_{\widetilde{h}} & 0_{d\times d_{w_\ell}}\end{bmatrix}}_{d\times 2d_{w_\ell}} \cdot \underbrace{\begin{bmatrix} 0\cdot w_\ell & 1\cdot w_\ell & 2\cdot w_\ell & \cdots & (n-1)\cdot w_\ell \\ w_\ell & w_\ell & w_\ell & \cdots & w_\ell \end{bmatrix}}_{2d_{w_\ell}\times n} && \text{(By (91))}
\end{aligned}$$

$$= \underbrace{\begin{bmatrix} 0 \cdot S_{\tilde{h}} w_\ell & 1 \cdot S_{\tilde{h}} w_\ell & 2 \cdot S_{\tilde{h}} w_\ell & \cdots & (n-1) \cdot S_{\tilde{h}} w_\ell \end{bmatrix}}_{d \times n}. \qquad \text{(By matrix multiplication)}$$

$S_{\tilde{h}} w_\ell$ expands as

$$S_{\tilde{h}} w_\ell = \underbrace{\begin{bmatrix} 0_{d \times d(j-1)} & I_d & 0_{d \times (d_{w_\ell} - d \cdot j)} \end{bmatrix}}_{d \times d_{w_\ell}} \cdot \underbrace{\begin{bmatrix} k_1^\ell \\ \vdots \\ k_{d_\ell}^\ell \\ q_1^\ell \\ \vdots \\ q_{d_\ell}^\ell \\ v_1^\ell \\ \vdots \\ v_{d_V}^\ell \end{bmatrix}}_{d_{w_\ell} \times 1} \qquad \text{(By (96) and (90))}$$

$$= \begin{cases} k_j^\ell, & \text{when} \quad j \in [d_\ell] \\ q_{j-d_\ell}^\ell, & \text{when} \quad j \in [2d_\ell] \setminus [d_\ell] \\ v_{j-2d_\ell}^\ell, & \text{when} \quad j \in [2d_\ell + d_V] \setminus [2d_\ell] \end{cases}. \qquad (S_{\tilde{h}} \text{ selects } k_j^\ell, q_{j-d_\ell}^\ell \text{ or } v_{j-2d_\ell}^\ell \text{ from } w_\ell)$$

Thus, acting $A_{\tilde{h}}$ on our input yields

$$A_{\tilde{h}} \cdot \begin{bmatrix} X \\ W_{\text{Attn}} \\ I_n \end{bmatrix}$$

$$= \begin{cases} \underbrace{\begin{bmatrix} & & X & \\ 0 \cdot k_j^\ell & 1 \cdot k_j^\ell & \cdots & (n-1) \cdot k_j^\ell \\ k_j^\ell & k_j^\ell & \cdots & k_j^\ell \\ & & I_n & \end{bmatrix}}_{(3d+n) \times n}, & \ell \in [H], \quad j \in [d_\ell] \\ \underbrace{\begin{bmatrix} & & X & \\ 0 \cdot q_{j-d_\ell}^\ell & 1 \cdot q_{j-d_\ell}^\ell & \cdots & (n-1) \cdot q_{j-d_\ell}^\ell \\ q_{j-d_\ell}^\ell & q_{j-d_\ell}^\ell & \cdots & q_{j-d_\ell}^\ell \\ & & I_n & \end{bmatrix}}_{(3d+n) \times n}, & \ell \in [H], \quad j \in [2d_\ell] \setminus [d_\ell] \\ \underbrace{\begin{bmatrix} & & X & \\ 0 \cdot v_{j-2d_\ell}^\ell & 1 \cdot v_{j-2d_\ell}^\ell & \cdots & (n-1) \cdot v_{j-2d_\ell}^\ell \\ v_{j-2d_\ell}^\ell & v_{j-2d_\ell}^\ell & \cdots & v_{j-2d_\ell}^\ell \\ & & I_n & \end{bmatrix}}_{(3d+n) \times n}, & \ell \in [H], \quad j \in [2d_\ell + d_V] \setminus [2d_\ell] \end{cases}.$$

To keep our following proof clean, we define

$$\xi_\ell \coloneqq \sum_{h=1}^{\ell} (2d_h + d_V), \quad \xi_0 \coloneqq 0.$$

Then, for each $\ell \in [H]$ and $j \in [d_\ell]$, by Theorem C.4, there exists an $\widetilde{H}_\ell$-head attention $\mathrm{Attn}_{\ell,j}$ such that

$$\| \underbrace{\mathrm{Attn}_{\ell,j}(A_{\widetilde{h}} \cdot \overbrace{\begin{bmatrix} X \\ W_{\mathrm{Attn}} \\ I_n \end{bmatrix}}^{(3d+n)\times n})_{:,i}}_{\xi_H \times 1} - \underbrace{e^{(\xi_H)}_{\xi_{\ell-1}+j}}_{\xi_H \times 1} \cdot (k^\ell_j)^\top x_i \|_\infty \le \max\{|a_\ell|, |b_\ell|\} \cdot \epsilon_0 + \frac{\epsilon_0}{2}.$$

Thus, we have

$$\| \underbrace{\mathrm{Attn}_{\ell,j}(A_{\widetilde{h}} \cdot \begin{bmatrix} X \\ W_{\mathrm{Attn}} \\ I_n \end{bmatrix})}_{\xi_H \times n} - \underbrace{e^{(\xi_H)}_{\xi_{\ell-1}+j}}_{\xi_H \times 1} \cdot \underbrace{(k^\ell_j)^\top X}_{1\times n} \|_\infty \le \max\{|a_\ell|, |b_\ell|\} \cdot \epsilon_0 + \frac{\epsilon_0}{2},$$

and

$$\| \underbrace{\sum_{j=1}^{d_\ell} \mathrm{Attn}_{\ell,j}(A_{\widetilde{h}} \cdot \begin{bmatrix} X \\ W_{\mathrm{Attn}} \\ I_n \end{bmatrix})}_{\xi_H \times n} - \underbrace{\begin{bmatrix} 0_{d_1 \times n} \\ \vdots \\ K_\ell \\ \vdots \\ 0_{d_V \times n} \end{bmatrix}}_{\xi_H \times n} \|_\infty \le \max\{|a_\ell|, |b_\ell|\} \cdot \epsilon_0 + \frac{\epsilon_0}{2}.$$

Similarly, together with Q_ℓ and V_ℓ, we have

$$\| \underbrace{\sum_{j=1}^{2d_\ell+d_V} \mathrm{Attn}_{\ell,j}(A_{\widetilde{h}} \cdot \begin{bmatrix} X \\ W_{\mathrm{Attn}} \\ I_n \end{bmatrix})}_{\xi_H \times n} - \underbrace{\begin{bmatrix} 0_{d_1 \times n} \\ \vdots \\ K_\ell \\ Q_\ell \\ V_\ell \\ \vdots \\ 0_{d_V \times n} \end{bmatrix}}_{\xi_H \times n} \|_\infty \le \max\{|a_\ell|, |b_\ell|\} \cdot \epsilon_0 + \frac{\epsilon_0}{2}$$

Finally, we have

$$\| \underbrace{\sum_{\ell=1}^{H} \sum_{j=1}^{2d_\ell+d_V} \mathrm{Attn}_{\ell,j}(A_{\widetilde{h}} \cdot \begin{bmatrix} X \\ W_{\mathrm{Attn}} \\ I_n \end{bmatrix})}_{\xi_H \times n} - \underbrace{\begin{bmatrix} K_1 \\ Q_1 \\ V_1 \\ \vdots \\ K_H \\ Q_H \\ V_H \end{bmatrix}}_{\xi_H \times n} \|_\infty \le \epsilon_0 \sum_{\ell=1}^{H} \max\{|a_\ell|, |b_\ell|\} + \frac{\epsilon_0}{2}.$$

By Lemma D.5, $\mathrm{Attn}_{\ell,j} \circ A_{\widetilde{h}}$ is still an attention, so we define

$$\mathrm{Attn}^{\ell,j}_{2\text{-}1} \coloneqq \mathrm{Attn}_{\ell,j} \circ A_{\widetilde{h}}.$$

Then, we have

$$\| \underbrace{\sum_{\ell=1}^{H} \sum_{j=1}^{2d_\ell+d_V} \mathrm{Attn}_{2\text{-}1}^{\ell,j}\left(\begin{bmatrix} X \\ W_{\mathrm{Attn}} \\ I_n \end{bmatrix}\right)}_{\xi_H \times n} - \underbrace{\begin{bmatrix} K_1 \\ Q_1 \\ V_1 \\ \vdots \\ K_H \\ Q_H \\ V_H \end{bmatrix}}_{\xi_H \times n} \|_\infty \le \epsilon_0 \sum_{\ell=1}^{H} \max\{|a_\ell|, |b_\ell|\} + \frac{\epsilon_0}{2}.$$

For the convenience of the following presentation, we define

$$\begin{bmatrix} K_1' \\ Q_1' \\ V_1' \\ \vdots \\ K_H' \\ Q_H' \\ V_H' \end{bmatrix} \coloneqq \sum_{\ell=1}^{H} \sum_{j=1}^{2d_\ell+d_V} \mathrm{Attn}_{2\text{-}1}^{\ell,j}\left(\begin{bmatrix} X \\ W_{\mathrm{Attn}} \\ I_n \end{bmatrix}\right),$$

$$\widetilde{\epsilon} \coloneqq \epsilon_0 \sum_{\ell=1}^{H} \max\{|a_\ell|, |b_\ell|\} + \frac{\epsilon_0}{2}.$$

Step 3: Assemble the Approximated Maps. In this part of our proof, our goal is to reconstruct the attention mechanism $V_h' \cdot \sigma_{\beta_h}((K_h')^\top Q_h')$ from the approximated maps via $\mathrm{Attn}_{2\text{-}2}$.

To that end, we construct $\mathrm{Attn}_{2\text{-}2}$ as an H-head attention, with each head recovering one of the heads $V_h' \cdot \sigma_{\beta_h}((K_h')^\top Q_h')$. We label the h-th head in $\mathrm{Attn}_{2\text{-}2}$ as $\mathrm{Attn}_{2\text{-}2}^{(h)}$, and we use $W_K^{2,h}, W_Q^{2,h}, W_V^{2,h}$ for the weight matrices in $\mathrm{Attn}_{2\text{-}2}^{(h)}$.

Specifically, we construct the h-th head in $\mathrm{Attn}_{2\text{-}2}$ as

$$W_K^{2,h} \coloneqq \underbrace{\begin{bmatrix} 0_{d_h \times d_1} & \cdots & I_{d_h} & 0_{d_h \times d_h} & 0_{d_h \times d_V} & \cdots & 0_{d_h \times d_V} \end{bmatrix}}_{d_h \times \xi_H},$$

$$W_Q^{2,h} \coloneqq \underbrace{\begin{bmatrix} 0_{d_h \times d_1} & \cdots & 0_{d_h \times d_h} & I_{d_h} & 0_{d_h \times d_V} & \cdots & 0_{d_h \times d_V} \end{bmatrix}}_{d_h \times \xi_H},$$

$$W_V^{2,h} \coloneqq \underbrace{\begin{bmatrix} 0_{d_V \times d_1} & \cdots & 0_{d_V \times d_h} & 0_{d_V \times d_h} & I_{d_V} & \cdots & 0_{d_V \times d_V} \end{bmatrix}}_{d_V \times \xi_H}.$$

The construction provides us with

$$W_K^{2,h} \cdot \begin{bmatrix} K_1' \\ \vdots \\ K_h' \\ Q_h' \\ V_h' \\ \vdots \\ V_H' \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{d_h \times d_1} & \cdots & I_{d_h} & 0_{d_h \times d_h} & 0_{d_h \times d_V} & \cdots & 0_{d_h \times d_V} \end{bmatrix}}_{d_h \times \xi_H} \cdot \underbrace{\begin{bmatrix} K_1' \\ \vdots \\ K_h' \\ Q_h' \\ V_h' \\ \vdots \\ V_H' \end{bmatrix}}_{\xi_H \times n} = \underbrace{K_h'}_{d_h \times n},$$

$$W_Q^{2,h} \cdot \begin{bmatrix} K_1' \\ \vdots \\ K_h' \\ Q_h' \\ V_h' \\ \vdots \\ V_H' \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{d_h \times d_1} & \cdots & 0_{d_h \times d_h} & I_{d_h} & 0_{d_h \times d_V} & \cdots & 0_{d_h \times d_V} \end{bmatrix}}_{d_h \times \xi_H} \cdot \underbrace{\begin{bmatrix} K_1' \\ \vdots \\ K_h' \\ Q_h' \\ V_h' \\ \vdots \\ V_H' \end{bmatrix}}_{\xi_H \times n} = \underbrace{Q_h'}_{d_h \times n},$$

$$W_V^{2,h} \cdot \begin{bmatrix} K_1' \\ \vdots \\ K_h' \\ Q_h' \\ V_h' \\ \vdots \\ V_H' \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{d_V \times d_1} & \cdots & 0_{d_V \times d_h} & 0_{d_V \times d_h} & I_{d_V} & \cdots & 0_{d_V \times d_V} \end{bmatrix}}_{d_V \times \xi_H} \cdot \underbrace{\begin{bmatrix} K_1' \\ \vdots \\ K_h' \\ Q_h' \\ V_h' \\ \vdots \\ V_H' \end{bmatrix}}_{\xi_H \times n} = \underbrace{V_h'}_{d_V \times n}.$$

Thus, our h-th head of $\mathrm{Attn}_{2\text{-}2}$ is

$$\mathrm{Attn}_{2\text{-}2}^{(h)} \coloneqq \underbrace{V_h'}_{d_V \times n} \cdot \underbrace{\sigma_{\beta_h}((K_h')^\top Q_h')}_{n \times n}.$$

Then, all of the heads from $\mathrm{Attn}_{2\text{-}2}$ give us

$$\sum_{h=1}^{H} \mathrm{Attn}_{2\text{-}2}^{(h)} = \sum_{h=1}^{H} V_h' \cdot \sigma_{\beta_h}((K_h')^\top Q_h').$$

Step 4: Error Bound. To bound the error, we compute

$$\begin{aligned} &\| \sum_{h=1}^{H} V_h' \cdot \sigma_{\beta_h}((K_h')^\top Q_h') - \sum_{h=1}^{H} V_h \cdot \sigma_{\beta_h}(K_h^\top Q_h) \|_\infty \\ &\leq \sum_{h=1}^{H} \| V_h' \cdot \sigma_{\beta_h}((K_h')^\top Q_h') - V_h \cdot \sigma_{\beta_h}(K_h^\top Q_h) \|_\infty \qquad \text{(By triangle inequality)} \end{aligned}$$

For each $\|V_h' \cdot \sigma_{\beta_h}((K_h')^\top Q_h') - V_h \cdot \sigma_{\beta_h}(K_h^\top Q_h)\|_\infty$, we utilize the error analysis result from Hu et al. (2026), and we have

$$\|V_h' \cdot \sigma_{\beta_h}((K_h')^\top Q_h') - V_h \cdot \sigma_{\beta_h}((K_h)^\top Q_h)\|_\infty < \widetilde{\epsilon} + nB_{KQV}\widehat{\epsilon},$$

where $0 < \widehat{\epsilon} < 2$ and

$$0 < \widetilde{\epsilon} \leq \min\{1, \frac{-\ln(1 - \frac{\widehat{\epsilon}}{2})}{\beta_h d_h (2B_{KQV} + 1)}\}.$$

Thus, we have

$$\sum_{h=1}^{H} \|V_h' \cdot \sigma_{\beta_h}((K_h')^\top Q_h') - V_h \cdot \sigma_{\beta_h}((K_h)^\top Q_h)\|_\infty < \sum_{h=1}^{H} (\widetilde{\epsilon} + nB_{KQV}\widehat{\epsilon}).$$

Since we are able to make $\widetilde{\epsilon}$ and $\widehat{\epsilon}$ arbitrarily small, we complete the proof. □

D.6 Proof of Corollary 3.1.1

Corollary D.1.1 (Corollary 3.1.1 Restated: In-Context Emulation of Multi-Head Attention with Flow-Through Component)**.** *Let $Z \in \mathbb{R}^{d_Z \times n}$ be any matrix satisfying $\|Z\|_\infty \le B_Z$ for some constant $B_Z > 0$. Under the same setting of Lemma 3.4, for any $\epsilon_5 > 0$, there exists a two-layer attention network Attn_2 such that*

$$\|\mathrm{Attn}_2(\begin{bmatrix} X_p^{\mathrm{m}} \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}(X) \\ Z \end{bmatrix}\|_\infty \le \epsilon_5.$$

Proof. Our proof consists of two major steps.

Step 1. We follow the proof of Lemma 3.3. That is, the heads in Attn_2 consist of two parts. In the first part, we incorporate the emulator from Lemma 3.4 into Attn_2. In the second part, we construct auxiliary heads within Attn_2 to copy the matrix Z.

Thus, in this step, the error comes only from the Z-copying behavior, and we have

$$\begin{aligned} &\|\mathrm{Attn}_2(\begin{bmatrix} X_p^{\mathrm{m}} \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}_2'(X_p^{\mathrm{m}}) \\ Z \end{bmatrix}\|_\infty \\ &\le B_Z \cdot \frac{n(n-1)}{e^{\beta_2} + (n-1)} \cdot \frac{n(n-1)}{e^{\beta_4} + (n-1)} + \frac{n(n-1)B_Z}{e^{\beta_2} + (n-1)} + \frac{n(n-1)B_Z}{e^{\beta_4} + (n-1)}, \qquad \text{(By (60))} \end{aligned}$$

where β_2 and β_4 are the Softmax temperatures in the auxiliary heads of Attn_2 (these two temperatures are up to our choice), and Attn_2' denotes the two-layer emulator from Lemma 3.4.

For any $\widehat{\epsilon}_1 > 0$, by choosing β_2, β_4 as in (61), we have

$$\|\mathrm{Attn}_2(\begin{bmatrix} X_p^{\mathrm{m}} \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}_2'(X_p^{\mathrm{m}}) \\ Z \end{bmatrix}\|_\infty \le 2\widehat{\epsilon}_1 + \frac{\widehat{\epsilon}_1^2}{B_Z}. \tag{98}$$

Step 2. In this part of our proof, we use the emulator Attn_2' as a proxy to approximate our target $\mathrm{Attn}(X)$. Specifically, we compute

$$\begin{aligned} &\|\mathrm{Attn}_2(\begin{bmatrix} X_p^{\mathrm{m}} \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}(X) \\ Z \end{bmatrix}\|_\infty \\ &\le \|\mathrm{Attn}_2(\begin{bmatrix} X_p^{\mathrm{m}} \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{Attn}_2'(X_p^{\mathrm{m}}) \\ Z \end{bmatrix}\|_\infty + \|\begin{bmatrix} \mathrm{Attn}_2'(X_p^{\mathrm{m}}) \\ Z \end{bmatrix} - \begin{bmatrix} \mathrm{Attn}(X) \\ Z \end{bmatrix}\|_\infty \qquad \text{(By triangle inequality)} \\ &\le 2\widehat{\epsilon}_1 + \frac{\widehat{\epsilon}_1^2}{B_Z} + \epsilon_4. \qquad \text{(By (98) and Lemma 3.4)} \end{aligned}$$

Since we are able to make each ϵ arbitrarily small, we complete the proof. □

D.7 Proof of Theorem 3.2

Definition D.4 (Definition 3.3 Restated: Input Prompt for In-Context UAP via Selector Route)**.** *Consider the four-layer attention network T, for layer $i \in [4]$, let H_i be the number of heads, and for each head $j \in [H_i]$, let W_j^i be the weight-encoding matrix of the j-th head in the i-th attention layer, constructed as W_h in Definition 3.2. Also, let X_a be the augmented input*

$$X_a := \begin{bmatrix} X & 0_{d\times 1} \\ I_n & 0_{n\times 1} \\ 0_{1\times n} & 1 \end{bmatrix} \in \mathbb{R}^{(d+n+1)\times(n+1)}.$$

Finally, we define the full prompt as the vertical concatenation of X_a and four layer-wise weight blocks:

$$\widetilde{X}_p := \begin{bmatrix} X_{\mathrm{a}}^\top & W_T^{1\top} & W_T^{2\top} & W_T^{3\top} & W_T^{4\top} \end{bmatrix}^\top,$$

wehre $W_T^i := [(W_1^i)^\top, \cdots, (W_{H_i}^i)^\top, I_{n+1}]^\top$.

Theorem D.2 (Theorem 3.2 Restated: In-Context Universal Approximation by Attention-Only Transformer)**.** *Let $\mathcal{C} \subset \mathbb{R}^{d\times n}$ be a compact domain of input sequences, and $f : \mathcal{C} \to \mathbb{R}^{d\times n}$ be a continuous sequence-to-sequence function. Let X be the input sequence. Then, for any $\epsilon > 0$, there exists an eight-layer multi-head attention network Attn_8 such that*

$$\mathrm{d}_p(\mathrm{Attn}_8(\widetilde{X}_p)_{:,1:n}, f(X)) < \epsilon,$$

where $\widetilde{X}_p$ follows Definition 3.3.

Proof. In the scope of this theorem, we aim to achieve the in-context approximation of a continuous sequence-to-sequence function f. Our strategy is to approximate the surrogate map T from Theorem C.2 and apply Theorem C.2 to approximate f. To approximate T in context, we design an eight-layer multi-head attention module Attn_8. The module Attn_8 receives the weights of T as part of its input and approximates its combined action in the end. We build Attn_8 by stacking the attention network from Corollary 3.1.1 three times and the emulator from Lemma 3.4 once. In more detail, we use Corollary 3.1.1 three times to approximate the first three layers of T, and we use Lemma 3.4 to approximate the last layer. Finally, we compose these in-context approximations to approximate the overall map T and apply Theorem C.2 to approximate the target function f at the end.

For the convenience of presentation, we use T_i to denote the i-th layer of T, and we express the map T as $T_4 \circ T_3 \circ T_2 \circ T_1$.

Step 1. In this part of our proof, we aim to approximate $T(X_{\mathrm{a}})$. First, we approximate $T_3 \circ T_2 \circ T_1(X_{\mathrm{a}})$ by applying Corollary 3.1.1 three times, and then we approximate $T_4 \circ T_3 \circ T_2 \circ T_1(X_{\mathrm{a}})$ by applying Lemma 3.4.

We start with Corollary 3.1.1 to approximate $T_1(X_{\mathrm{a}})$. By Corollary 3.1.1, there exists a two-layer attention Attn_2^1 such that

$$\|\mathrm{Attn}_2^1(\begin{bmatrix} X_{\mathrm{a}} \\ W_T^1 \\ W_T^2 \\ W_T^3 \\ W_T^4 \end{bmatrix}) - \begin{bmatrix} T_1(X_{\mathrm{a}}) \\ W_T^2 \\ W_T^3 \\ W_T^4 \end{bmatrix}\|_\infty \le \epsilon_5. \tag{99}$$

Here Attn_2^1 denotes a two-layer attention constructed from Corollary 3.1.1, and the superscript 1 identifies the target T_1 in this step.

To keep our proof clean, we define

$$P_1 \coloneqq \mathrm{Attn}_2^1(\begin{bmatrix} X_{\mathrm{a}} \\ W_T^1 \\ W_T^2 \\ W_T^3 \\ W_T^4 \end{bmatrix}), \quad P_1^* \coloneqq \begin{bmatrix} T_1(X_{\mathrm{a}}) \\ W_T^2 \\ W_T^3 \\ W_T^4 \end{bmatrix}.$$

Next, we aim to approximate $T_2 \circ T_1(X_{\mathrm{a}})$. Our strategy is to stack the multi-head attentions from (99) and Corollary 3.1.1. Specifically, we compute

$$\begin{aligned} &\|\mathrm{Attn}_2^2(P_1) - \begin{bmatrix} T_2 \circ T_1(X_{\mathrm{a}}) \\ W_T^3 \\ W_T^4 \end{bmatrix}\|_\infty \\ &\le \|\mathrm{Attn}_2^2(P_1) - \mathrm{Attn}_2^2(P_1^*)\|_\infty + \|\mathrm{Attn}_2^2(P_1^*) - \begin{bmatrix} T_2 \circ T_1(X_{\mathrm{a}}) \\ W_T^3 \\ W_T^4 \end{bmatrix}\|_\infty, \end{aligned} \tag{100}$$

where (100) follows from the triangle inequality, and Attn_2^2 is another two-layer attention constructed from Corollary 3.1.1. We remind the reader that Attn_2^2 differs from Attn_2^1 since the dimensions to be copied are different.

Then, to bound the approximation of $T_2 \circ T_1(X_{\mathrm{a}})$, we need to bound (100).

To bound the first term in (100), we assume Attn_2^2 to be L_3-Lipschitz in $\|\cdot\|_\infty$

$$\|\mathrm{Attn}_2^2(G) - \mathrm{Attn}_2^2(H)\|_\infty \le L_3\|G - H\|_\infty, \tag{101}$$

where G, H denotes any input of matching shape. $L_3 > 0$ is a constant and depends on the parameters of Attn_2^2.

Under the L_3-Lipschitz assumption, we have

$$\begin{aligned} &\|\mathrm{Attn}_2^2(P_1) - \mathrm{Attn}_2^2(P_1^*)\|_\infty && \text{(The first term in (100))} \\ &\le L_3\|P_1 - P_1^*\|_\infty && \text{(By (101))} \\ &\le L_3\epsilon_5, && (102) \end{aligned}$$

and (102) follows from (99).

To bound the second term in (100), we use Corollary 3.1.1, and we have

$$\|\mathrm{Attn}_2^2(P_1^*) - \begin{bmatrix} T_2 \circ T_1(X_\mathrm{a}) \\ W_T^3 \\ W_T^4 \end{bmatrix}\|_\infty \le \epsilon_5. \tag{103}$$

Combining (102) and (103), we have

$$\|\mathrm{Attn}_2^2(P_1) - \begin{bmatrix} T_2 \circ T_1(X_\mathrm{a}) \\ W_T^3 \\ W_T^4 \end{bmatrix}\|_\infty \le \epsilon_5(L_3 + 1). \tag{104}$$

Again, to keep our proof clean, we define

$$P_2 \coloneqq \mathrm{Attn}_2^2(P_1), \quad P_2^* \coloneqq \begin{bmatrix} T_2 \circ T_1(X_\mathrm{a}) \\ W_T^3 \\ W_T^4 \end{bmatrix}.$$

Thirdly, we aim to approximate $T_3 \circ T_2 \circ T_1(X_\mathrm{a})$. We utilize the same techniques as those for $T_2 \circ T_1(X_\mathrm{a})$. That is, we stack the multi-head attentions from (104) and Corollary 3.1.1, and we utilize the Lipschitz assumption together with Corollary 3.1.1 to derive the following bound

$$\|\mathrm{Attn}_2^3(P_2) - \begin{bmatrix} T_3 \circ T_2 \circ T_1(X_\mathrm{a}) \\ W_T^4 \end{bmatrix}\|_\infty \le \epsilon_5(L_4(L_3 + 1) + 1), \tag{105}$$

where Attn_2^3 is another two-layer attention constructed from Corollary 3.1.1. We remind the reader that Attn_2^3 differs from Attn_2^3 since the dimensions to be copied are different. $L_4 > 0$ is the Lipschitz constant of Attn_2^3 and depends on the parameters of Attn_2^3.

Again, to keep our proof clean, we define

$$P_3 \coloneqq \mathrm{Attn}_2^3(P_2), \quad P_3^* \coloneqq \begin{bmatrix} T_3 \circ T_2 \circ T_1(X_\mathrm{a}) \\ W_T^4 \end{bmatrix}.$$

Finally, we aim to approximate $T_4 \circ T_3 \circ T_2 \circ T_1(X_\mathrm{a})$. Our strategy is to stack the multi-head attentions from (105) and Lemma 3.4. Specifically, we compute

$$\begin{aligned} &\|\mathrm{Attn}_2^4(P_3) - T_4 \circ T_3 \circ T_2 \circ T_1(X_\mathrm{a})\|_\infty \\ &\le \|\mathrm{Attn}_2^4(P_3) - \mathrm{Attn}_2^4(P_3^*)\|_\infty + \|\mathrm{Attn}_2^4(P_3^*) - T_4 \circ T_3 \circ T_2 \circ T_1(X_\mathrm{a})\|_\infty, \end{aligned} \tag{106}$$

where (106) follows from the triangle inequality, and Attn_2^4 denotes the two-layer attention constructed from Lemma 3.4.

Therefore, to bound the approximation of $T_4 \circ T_3 \circ T_2 \circ T_1(X_\mathrm{a})$, we need to bound (106).

To bound the first term in (106), we assume Attn_2^4 to be L_5-Lipschitz

$$\|\mathrm{Attn}_2^4(G) - \mathrm{Attn}_2^4(H)\|_\infty \le L_5\|G - H\|_\infty, \tag{107}$$

where G, H denotes any input of matching shape. $L_5 > 0$ is a constant, and it depends on the parameters of Attn_2^4.

Under the L_5-Lipschitz assumption, we have

$$\begin{aligned} &\|\mathrm{Attn}_2^4(P_3) - \mathrm{Attn}_2^4(P_3^*)\|_\infty && \text{(The first term in (106))} \\ &\le L_5 \|P_3 - P_3^*\|_\infty && \text{(By (107))} \\ &\le L_5 \cdot \epsilon_5 (L_4(L_3+1)+1), && (108) \end{aligned}$$

and (108) follows from (105).

To bound the second term in (106), we utilize Lemma 3.4, so we have

$$\|\mathrm{Attn}_2^4(P_3^*) - T_4 \circ T_3 \circ T_2 \circ T_1(X_{\mathrm{a}})\|_\infty \le \epsilon_4 \tag{109}$$

Combining (108) and (109), we have

$$\|\mathrm{Attn}_2^4(P_3) - T_4 \circ T_3 \circ T_2 \circ T_1(X_{\mathrm{a}})\|_\infty \le \epsilon_5 \cdot L_5(L_4(L_3+1)+1) + \epsilon_4. \tag{110}$$

We note that X_{a} depends on X as in Theorem C.2 (Thus, P_3 also depends on X), so we define the following notation for the convenience of our presentation

$$\begin{aligned} \Phi(X) &\coloneqq \mathrm{Attn}_2^4(P_3)_{:,1:n}, \\ \Psi(X) &\coloneqq T_4 \circ T_3 \circ T_2 \circ T_1(X_{\mathrm{a}})_{:,1:n}, \end{aligned}$$

such that

$$\|\Phi(X) - \Psi(X)\|_\infty \le \epsilon_5 \cdot L_5(L_4(L_3+1)+1) + \epsilon_4, \tag{111}$$

and (111) follows from (110).

Up to this point, we are capable of approximating T arbitrarily closely.

Step 2. In this part of our proof, we aim to utilize (111) and Theorem C.2 to derive the approximation of f.

Specifically, we compute

$$\begin{aligned} \mathrm{d}_p(\Phi, f) &\le \mathrm{d}_p(\Phi, \Psi) + \mathrm{d}_p(\Psi, f) && \text{(By Minkowski inequality)} \\ &< \mathrm{d}_p(\Phi, \Psi) + \epsilon, && (112) \end{aligned}$$

where (112) follows from Theorem C.2.

As for $\mathrm{d}_p(\Phi, \Psi)$, we have

$$\mathrm{d}_p(\Phi, \Psi) = \Big(\int \|\Phi(X) - \Psi(X)\|_p^p \mathrm{d}X\Big)^{\frac{1}{p}}. \qquad \text{(By Definition 2.3)}$$

Therefore, to bound $\mathrm{d}_p(\Phi, \Psi)$, we need to analyze the quantity $\|\Phi(X) - \Psi(X)\|_p^p$.

For the simplicity of presentation, we define

$$E \coloneqq \Phi(X) - \Psi(X),$$

so

$$\|\Phi(X) - \Psi(X)\|_p^p = \|E\|_p^p.$$

Then, we have

$$\|E\|_p^p$$

$$
\begin{aligned}
&= \sum_{i=1}^{d}\sum_{j=1}^{n} |E_{ij}|^p && \text{(By the definition of } \ell_p \text{ norm)} \\
&\le \sum_{i=1}^{d}\sum_{j=1}^{n} \|E\|_\infty^p && \text{(By } |E_{ij}| \le \|E\|_\infty\text{)} \\
&= dn\|E\|_\infty^p \\
&\le dn(\epsilon_5 \cdot L_5(L_4(L_3+1)+1)+\epsilon_4)^p, && (113)
\end{aligned}
$$

and (113) follows from (111).

Thus,

$$
\begin{aligned}
&\mathrm{d}_p(\Phi,\Psi) \\
&= \Big(\int \|\Phi(X)-\Psi(X)\|_p^p \mathrm{d}X\Big)^{\frac{1}{p}} \\
&\le \big(dn(\epsilon_5 \cdot L_5(L_4(L_3+1)+1)+\epsilon_4)^p \int \mathrm{d}X\big)^{\frac{1}{p}}. && \text{(By (113))}
\end{aligned}
$$

In Theorem C.2, the authors require the sequence-to-sequence function f to take values on a compact domain, so the quantity $\int \mathrm{d}X$ is finite. Then we have

$$
\mathrm{d}_p(\Phi,\Psi) \le d^{\frac{1}{p}} n^{\frac{1}{p}} (\epsilon_5 \cdot L_5(L_4(L_3+1)+1)+\epsilon_4) C, \tag{114}
$$

where $C \coloneqq (\int \mathrm{d}X)^{\frac{1}{p}}$.

Thus, we have

$$
\mathrm{d}_p(\Phi,f) \le C d^{\frac{1}{p}} n^{\frac{1}{p}} (\epsilon_5 \cdot L_5(L_4(L_3+1)+1)+\epsilon_4) + \epsilon \qquad \text{(By (112) and (114))}
$$

Since we are capable of making ϵ arbitrarily small, we complete the proof. □

The following remark details how to explicitly construct the weights of a ReLU neural network for a given function f. Since Hu et al. (2025, Theorem G.1) provides an explicit approximation of ReLU networks via attention, we can combine that result with the construction below. This yields the explicit prompt P_f, consisting of the weight blocks W_T^i for $i \in [4]$ required by Theorem 3.2.

This remark is modified from (Hu et al., 2025, Lemma 3.1). We only keep the construction necessary to demonstrate the weight dependency on f. For detailed explanation please refer to the original paper.

Remark D.2 (Example of Explicit ReLU Network Parameters and the Dependence on f)**.** *Let $\mathcal{C} \subset \mathbb{R}^{d\times n}$ be the compact domain. Pick $B > 0$ such that $\mathcal{C} \subset [-B,B]^{d\times n}$. Fix an integer $g \ge 2$ and a margin parameter $\delta \in (0,1)$. Define the grid*

$$
G_D = \left\{ \frac{-B(g-1)}{g}, \frac{-B(g-3)}{g}, \ldots, \frac{B(g-1)}{g} \right\}^N.
$$

Layer 1: Bumps Functions. *For each $v \in G_D$ and each coordinate $i \in [N]$, introduce four ReLU units*

$$
\begin{aligned}
h_{v,i}^{+}(x) &= \mathrm{ReLU}\left(\frac{g}{\delta B}x_i + \frac{1}{\delta}\left(1 - \frac{g}{B}v_i\right)\right), \\
\widetilde{h}_{v,i}^{+}(x) &= \mathrm{ReLU}\left(\frac{g}{\delta B}x_i + \frac{1}{\delta}\left(1 - \delta - \frac{g}{B}v_i\right)\right), \\
h_{v,i}^{-}(x) &= \mathrm{ReLU}\left(-\frac{g}{\delta B}x_i + \frac{1}{\delta}\left(1 + \frac{g}{B}v_i\right)\right), \\
\widetilde{h}_{v,i}^{-}(x) &= \mathrm{ReLU}\left(-\frac{g}{\delta B}x_i + \frac{1}{\delta}\left(1 - \delta + \frac{g}{B}v_i\right)\right).
\end{aligned}
$$

Define

$$\phi_{v,i}(x) = h^+_{v,i}(x) - \widetilde{h}^+_{v,i}(x) + h^-_{v,i}(x) - \widetilde{h}^-_{v,i}(x) - 1, \qquad R_v(x) = \sum_{i=1}^{N} \phi_{v,i}(x).$$

All Layer-1 weights and biases depend only on (B, g, δ) and v.

Layer 2: Grid Gating. *For each $v \in G_D$, introduce one ReLU unit*

$$u_v(x) = \mathrm{ReLU}(R_v(x) - N + 1)\,.$$

Equivalently, since $R_v(x) = \sum_{i=1}^{N} \big(h^+_{v,i} - \widetilde{h}^+_{v,i} + h^-_{v,i} - \widetilde{h}^-_{v,i}\big) - N$, the pre-activation of u_v is an explicit affine map of the Layer-1 outputs with bias $-2N + 1$.

Output Layer (Dependency on f). *Define the scalar-output network*

$$\mathrm{FFN}_{f,g,\delta}(x) = \sum_{v \in G_D} \alpha_v\, u_v(x), \qquad \alpha_v \coloneqq f(v).$$

Hence the dependence on the target function is explicit: f enters the parameters only through the output coefficients $\{\alpha_v\}_{v \in G_D} = \{f(v)\}_{v \in G_D}$. All remaining weights depend only on (B, g, δ) and the fixed grid G_D.

Network Size and Depth. *The construction uses two ReLU hidden layers: Layer 1 has $4N|G_D|$ ReLU units, Layer 2 has $|G_D|$ ReLU units, and the output is linear.*

Vector-Valued Outputs. *For $f : \mathcal{C} \to \mathbb{R}^{d \times n}$, apply the same gates $\{u_v(x)\}_{v \in G_D}$ and set output weights $\alpha_v \in \mathbb{R}^{d \times n}$ with $\alpha_v = f(v)$.*

D.8 Proof of Lemma 4.1

Lemma D.7 (Lemma 4.1 Restated: In-Context Universal Approximation with Flow-Through Component)**.** *Let $X \in \mathbb{R}^{d \times n}$ be the input sequence. Let W be the weight encoding for a sequence-to-sequence function f as in Theorem 3.1. Let $Z \in \mathbb{R}^{d_Z \times n}$ be an arbitrary matrix satisfying $\|Z\|_\infty \le B_Z$ for $B_Z > 0$. Then, for any $\epsilon > 0$, there exists a six-layer multi-head attention network Attn_6 such that*

$$\mathrm{d}_p(\mathrm{Attn}_6(\begin{bmatrix} X \\ W \\ Z \end{bmatrix}), \begin{bmatrix} f(X) \\ Z \end{bmatrix}) \le \epsilon.$$

Proof. We follow the proof strategy of Theorem 3.1 and modify only two places in the proof. We modify the input and the last two layers of attention. We modify our input to

$$\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_s^{\mathrm{res}}} \\ I_n \\ W_{\mathrm{FF}_2} \\ Z \end{bmatrix},$$

and we change the last two layers of attention to the ones in Lemma 3.1. Thus, after the modification, the error is

$$\|\mathrm{Attn}_2^3 \circ \mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_s^{\mathrm{res}}} \\ I_n \\ W_{\mathrm{FF}_2} \\ Z \end{bmatrix}) - \begin{bmatrix} \mathrm{FF}_2 \circ \mathrm{Attn}_s^{\mathrm{res}} \circ \mathrm{FF}_1(X) \\ Z \end{bmatrix}\|_\infty \le L_7(L_6\epsilon_1 + \epsilon_3) + \epsilon_1, \qquad \text{(By (76))}$$

where Attn_2^1 and Attn_2^3 are the two-layer attentions constructed from Lemma 3.1, and Attn_2^2 is the one constructed from Lemma 3.3. Also, L_7, L_6 are the Lipschitz constants of Attn_2^2 and Attn_2^3, respectively.

Next, following the same analysis as in the proof of Theorem 3.1, we have

$$\mathrm{d}_p(\mathrm{Attn}_2^3 \circ \mathrm{Attn}_2^2 \circ \mathrm{Attn}_2^1(\begin{bmatrix} X \\ W_{\mathrm{FF}_1} \\ W_{\mathrm{Attn}_s^{\mathrm{res}}} \\ I_n \\ W_{\mathrm{FF}_2} \\ Z \end{bmatrix}), \begin{bmatrix} f(X) \\ Z \end{bmatrix}) \\ \leq C(d_Z + d)^{\frac{1}{p}} n^{\frac{1}{p}} (L_7(L_6\epsilon_1 + \epsilon_3) + \epsilon_1) + \epsilon_{\mathrm{u}}.$$

Since we are able to make each ϵ arbitrarily small, this completes the proof. □

D.9 Proof of Theorem 4.1

Theorem D.3 (Theorem 4.1 Restated: In-Context Composition: Single Completion)**.** *Let $X \in \mathbb{R}^{d\times n}$ be the input sequence. Let $\mathcal{F}$ be a set of continuous functions defined on a compact domain*

$$\mathcal{F} \coloneqq \{f \mid f : \mathcal{C} \to \mathcal{C},\ \mathcal{C} \subset \mathbb{R}^{d\times n}\}.$$

Consider a function composition of length m: $f_m \circ f_{m-1} \circ \cdots \circ f_1(X)$, where $m \in \mathbb{N}^+$ and $f_i \in \mathcal{F}$. Let W_i be the weight encoding for f_i as in Theorem 3.1. Then, for any $\epsilon > 0$, there exists a $6m$-layer attention network Attn_{6m} such that

$$\mathrm{d}_p(\mathrm{Attn}_{6m}(\begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}), f_m \circ f_{m-1} \circ \cdots \circ f_1(X)) \leq \epsilon.$$

Proof. To approximate a function composition of length m, our strategy is to stack $m-1$ attention modules constructed from Lemma 4.1 for the first $m-1$ functions, and we utilize Theorem 3.1 for the m-th function. The rationale is to approximate each step and, at the same time, pass the weight encoding to the next layer for subsequent functions. For the m-th function, since there is no weight encoding need to be passed, we utilize Theorem 3.1 rather than Lemma 4.1 to approximate the last layer.

In the following proof, we provide the error bound for those $m-1$ intermediate steps as well as the final length m composition.

To that end, for $i \in [m-1]$, we define the intermediate steps as

$$c_0(X) \coloneqq \begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}, \quad c_i(X) \coloneqq \begin{bmatrix} f_i \circ \cdots \circ f_1(X) \\ W_{i+1} \\ \vdots \\ W_m \end{bmatrix},$$

$$\widehat{c}_0(X) \coloneqq \begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}, \quad \widehat{c}_i(X) \coloneqq \mathrm{Attn}_6^i(\widehat{c}_{i-1}(X)),$$

where Attn_6^i is the attention module constructed from Lemma 4.1 for the i-th step.

For $t \in [m-1] \cup \{0\}$, we define the error as

$$e_t \coloneqq \mathrm{d}_p(\widehat{c}_t(X), c_t(X)),$$

and this definition leads to $e_0 = 0$.

To provide an error bound for $t \in [m-1]$, we expand e_t as

$$\begin{aligned} e_t &= \mathrm{d}_p(\mathrm{Attn}_6^t(\widehat{c}_{t-1}(X)), c_t(X)) \\ &\le \mathrm{d}_p(\mathrm{Attn}_6^t(\widehat{c}_{t-1}(X)), \mathrm{Attn}_6^t(c_{t-1}(X))) + \mathrm{d}_p(\mathrm{Attn}_6^t(c_{t-1}(X)), c_t(X)) && \text{(By Minkowski Inequality)} \\ &\le \mathrm{d}_p(\mathrm{Attn}_6^t(\widehat{c}_{t-1}(X)), \mathrm{Attn}_6^t(c_{t-1}(X))) + \epsilon, && (115) \end{aligned}$$

and (115) follows from Lemma 4.1.

For the first term in (115), we assume Attn_6^t to be L_t-Lipschitz in $\|\cdot\|_p$

$$\|\mathrm{Attn}_6^t(G) - \mathrm{Attn}_6^t(H)\|_p \le L_t \|G - H\|_p,$$

where G, H denotes any input of matching shape, such that

$$\begin{aligned} &\mathrm{d}_p(\mathrm{Attn}_6^t(\widehat{c}_{t-1}(X)), \mathrm{Attn}_6^t(c_{t-1}(X))) \\ &= (\int \|\mathrm{Attn}_6^t(\widehat{c}_{t-1}(X)) - \mathrm{Attn}_6^t(c_{t-1}(X))\|_p^p \mathrm{d}X)^{\frac{1}{p}} \\ &\le (L_t^p \int \|\widehat{c}_{t-1}(X) - c_{t-1}(X)\|_p^p \mathrm{d}X)^{\frac{1}{p}} \\ &= L_t \cdot e_{t-1}. && (116) \end{aligned}$$

Combining (115) and (116), we have

$$e_t \le L_t \cdot e_{t-1} + \epsilon.$$

For $t = 1$, we have

$$e_1 \le \epsilon. \qquad \text{(By } e_0 = 0\text{)}$$

For $t = 2$, we have

$$e_2 \le L_2 \epsilon + \epsilon.$$

For $t = 3$, we have

$$e_3 \le L_3 L_2 \epsilon + L_3 \epsilon + \epsilon.$$

By induction, for $t \in [m-1]$, we have

$$e_t \le \epsilon \sum_{s=1}^{t} \prod_{i=s+1}^{t} L_i, \qquad (117)$$

where $\prod_{i=t+1}^{t} L_i = 1$ by convention.

Up to here, we are able to approximate the first $m-1$ steps arbitrarily closely.

For the final m-th function, we utilize Theorem 3.1. Specifically, we compute

$$\mathrm{d}_p(\mathrm{Attn}_6^m \circ \cdots \circ \mathrm{Attn}_6^1(\begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}), f_m \circ \cdots \circ f_1(X))$$

$$\leq \mathrm{d}_p(\mathrm{Attn}_6^m \circ \cdots \circ \mathrm{Attn}_6^1(\begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}), \mathrm{Attn}_6^m(\begin{bmatrix} f_{m-1} \circ \cdots \circ f_1(X) \\ W_m \end{bmatrix}))$$

$$+ \mathrm{d}_p(\mathrm{Attn}_6^m(\begin{bmatrix} f_{m-1} \circ \cdots \circ f_1(X) \\ W_m \end{bmatrix}), f_m \circ \cdots \circ f_1(X)) \qquad \text{(By Minkowski Inequality)}$$

$$\leq \mathrm{d}_p(\mathrm{Attn}_6^m \circ \mathrm{Attn}_6^{m-1} \cdots \circ \mathrm{Attn}_6^1(\begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}), \mathrm{Attn}_6^m(\begin{bmatrix} f_{m-1} \circ \cdots \circ f_1(X) \\ W_m \end{bmatrix})) + \epsilon. \tag{118}$$

where Attn_6^m is the attention module constructed from Theorem 3.1, and (118) follows from Theorem 3.1.

For the first term in (118), we assume Attn_6^m to be L_m-Lipschitz in $\|\cdot\|_p$

$$\|\mathrm{Attn}_6^m(G) - \mathrm{Attn}_6^m(H)\|_p \leq L_m \|G - H\|_p,$$

where G, H denotes any input of matching shape, such that

$$\mathrm{d}_p(\mathrm{Attn}_6^m \circ \mathrm{Attn}_6^{m-1} \cdots \circ \mathrm{Attn}_6^1(\begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}), \mathrm{Attn}_6^m(\begin{bmatrix} f_{m-1} \circ \cdots \circ f_1(X) \\ W_m \end{bmatrix}))$$

$$= (\int \|\mathrm{Attn}_6^m(\widehat{c}_{m-1}(X)) - \mathrm{Attn}_6^m(c_{m-1}(X))\|_p^p \mathrm{d}X)^{\frac{1}{p}}$$

$$\leq (L_m^p \int \|\widehat{c}_{m-1}(X) - c_{m-1}(X)\|_p^p \mathrm{d}X)^{\frac{1}{p}}$$

$$= L_m \cdot e_{m-1}$$

$$\leq L_m \cdot (\epsilon \sum_{s=1}^{m-1} \prod_{i=s+1}^{m-1} L_i), \tag{119}$$

where (119) follows from (117).

Combining (118) and (119), we have

$$\mathrm{d}_p(\mathrm{Attn}_6^m \circ \cdots \circ \mathrm{Attn}_6^1(\begin{bmatrix} X \\ W_1 \\ W_2 \\ \vdots \\ W_m \end{bmatrix}), f_m \circ \cdots \circ f_1(X)) \leq L_m \cdot (\epsilon \sum_{s=1}^{m-1} \prod_{i=s+1}^{m-1} L_i) + \epsilon$$

Since we are able to make ϵ arbitrarily small, this completes the proof. □

D.10 Proof of Theorem 4.2

Theorem D.4 (Theorem 4.2 Restated: In-Context Composition: Re-Prompting)**.** *Let $X \in \mathbb{R}^{d \times n}$ be the input sequence. Let Attn_6 be the frozen attention module from Theorem 3.1. Let $\mathcal{F}$ be a set of continuous functions defined on a compact domain*

$$\mathcal{F} \coloneqq \{f \mid f : \mathcal{C} \to \mathcal{C},\ \mathcal{C} \subset \mathbb{R}^{d \times n}\}.$$

Consider a function composition of length m: $f_m \circ f_{m-1} \circ \cdots \circ f_1(X)$, where $m \in \mathbb{N}^+$ and $f_i \in \mathcal{F}$. For $i \in [m]$, we define the intermediate steps as

$$c_0(X) \coloneqq X, \quad c_i(X) \coloneqq f_i(c_{i-1}(X)),$$
$$\widehat{c}_0(X) \coloneqq X, \quad \widehat{c}_i(X) \coloneqq \mathrm{Attn}_6(\begin{bmatrix} \widehat{c}_{i-1}(X) \\ W_i \end{bmatrix}),$$

where W_i is the weight encoding for f_i as in Theorem 3.1. Then, for any $\epsilon > 0$, we have

$$\mathrm{d}_p(c_t(X), \widehat{c}_t(X)) \leq \epsilon,$$

where $t \in [m] \cup \{0\}$.

Proof. Our goal is to provide an upper bound on the error between $\widehat{c}_t(X)$ and $c_t(X)$.

For $t \in [m] \cup 0$, we define the error after t steps as

$$e_t \coloneqq \mathrm{d}_p(\widehat{c}_t(X), c_t(X)), \tag{120}$$

and expand e_t as follows

$$\begin{aligned}
e_t &= \mathrm{d}_p(\mathrm{Attn}_6(\begin{bmatrix} \widehat{c}_{t-1}(X) \\ W_t \end{bmatrix}), f_t(c_{t-1}(X))) && \text{(By (120))} \\
&\leq \mathrm{d}_p(\mathrm{Attn}_6(\begin{bmatrix} \widehat{c}_{t-1}(X) \\ W_t \end{bmatrix}), \mathrm{Attn}_6(\begin{bmatrix} c_{t-1}(X) \\ W_t \end{bmatrix})) + \mathrm{d}_p(\mathrm{Attn}_6(\begin{bmatrix} c_{t-1}(X) \\ W_t \end{bmatrix}), f_t(c_{t-1}(X))) && \text{(By Minkowski inequality)} \\
&\leq \mathrm{d}_p(\mathrm{Attn}_6(\begin{bmatrix} \widehat{c}_{t-1}(X) \\ W_t \end{bmatrix}), \mathrm{Attn}_6(\begin{bmatrix} c_{t-1}(X) \\ W_t \end{bmatrix})) + \epsilon, && (121)
\end{aligned}$$

where (121) follows from Theorem 3.1.

As for the first term in (121), we assume Attn_6 to be $\widetilde{L}$-Lipschitz in $\|\cdot\|_p$

$$\|\mathrm{Attn}_6(G) - \mathrm{Attn}_6(H)\|_p \leq \widetilde{L}\|G - H\|_p,$$

where G, H denotes any input of matching shape, so we have

$$\begin{aligned}
&\mathrm{d}_p(\mathrm{Attn}_6(\begin{bmatrix} \widehat{c}_{t-1}(X) \\ W_t \end{bmatrix}), \mathrm{Attn}_6(\begin{bmatrix} c_{t-1}(X) \\ W_t \end{bmatrix})) \\
&= (\int \|\mathrm{Attn}_6(\begin{bmatrix} \widehat{c}_{t-1}(X) \\ W_t \end{bmatrix}) - \mathrm{Attn}_6(\begin{bmatrix} c_{t-1}(X) \\ W_t \end{bmatrix})\|_p^p \mathrm{d}X)^{\frac{1}{p}} && \text{(By Definition 2.3)} \\
&\leq L(\int \| \begin{bmatrix} \widehat{c}_{t-1}(X) \\ W_t \end{bmatrix} - \begin{bmatrix} c_{t-1}(X) \\ W_t \end{bmatrix} \|_p^p \mathrm{d}X)^{\frac{1}{p}} \\
&= \widetilde{L} \cdot \mathrm{d}_p(\widehat{c}_{t-1}(X), c_{t-1}(X)) \\
&= \widetilde{L} \cdot e_{t-1}, && (122)
\end{aligned}$$

Combining (121) and (122), we have

$$e_t \leq \widetilde{L} \cdot e_{t-1} + \epsilon.$$

For $t = 1$, we have

$$e_1 \leq \epsilon. \qquad \text{(By } e_0 = 0\text{)}$$

For $t = 2$, we have

$$e_2 \leq \widetilde{L}\epsilon + \epsilon.$$

For $t = 3$, we have

$$e_3 \leq \widetilde{L}^2\epsilon + \widetilde{L}\epsilon + \epsilon.$$

By induction, for $t \in [m]$, we have

$$e_t \leq \epsilon \sum_{i=1}^{t} \widetilde{L}^{i-1}.$$

Since we are able to make ϵ arbitrarily small, this completes the proof. □