

Constructing Efficient Fact-Storing MLPs for Transformers

Owen Dugan^{1 * †} Roberto Garcia^{2 *} Ronny Junkins^{1 *} Jerry Liu^{2 *}
Dylan Zinsley³ Sabri Eyuboglu¹ Atri Rudra⁴ Chris Ré¹

¹Computer Science Department, Stanford University

²Institute for Computational & Mathematical Engineering, Stanford University

³Computer Science Department, University of Wisconsin–Madison

⁴Computer Science and Engineering Department, University at Buffalo

December 2, 2025

Abstract

The success of large language models (LLMs) can be attributed in part to their ability to efficiently store factual knowledge as key-value mappings within their MLP parameters. Recent work has proposed explicit weight constructions to build such fact-storing MLPs, providing an improved understanding of LLM fact storage mechanisms. In this paper, we introduce an MLP construction framework that improves over previous constructions in three areas: it 1) works for all but a measure zero set of feasible input-output pairs, 2) achieves asymptotically optimal parameter efficiency matching information-theoretic bounds for some embeddings, and 3) maintains usability within Transformers for factual recall. Through our improvements, we 1) discover a metric on value embeddings that characterizes facts-per-parameter scaling for both constructed and gradient-descent-trained MLPs, 2) identify a simple encoder-decoder mechanism that empirically matches gradient-descent MLP facts-per-parameter asymptotics across all the inputs and outputs we test, and 3) uncover a fundamental tradeoff between an MLP’s fact-storage capacity and its usability within Transformers. Finally, we demonstrate a proof-of-concept application of fact-storing MLPs: modular fact editing on one-layer Transformers by *replacing entire MLPs at once*.

1 Introduction

Large language models (LLMs) achieve remarkable performance across domains such as mathematics, science, and law (Google DeepMind, 2024; Guha et al., 2023; Saab et al., 2024), in part because of their ability to store vast amounts of knowledge within their parameters (Petroni et al., 2019; Meng et al., 2023a). As a result, there has been considerable interest in understanding the mechanism by which LLMs store knowledge.

A body of prior work seeks to understand how and where LLMs store knowledge by probing pretrained LLMs. These works observed that knowledge is often stored within Multi-Layer Perceptrons (MLPs) via key-value mappings (*facts*) (Geva et al., 2021; Dai et al., 2022) and have explored LLM fact-editing by modifying MLP parameters (Geva et al., 2022; Meng et al., 2023a; Nanda et al., 2023). Another line of work measures the empirical fact storage capacity of LLMs (Allen-Zhu & Li, 2024; Zucchet et al., 2025; Morris et al., 2025), observing that their facts-per-parameter scaling is asymptotically optimal. More recently, Nichani et al. (2024) further the understanding of MLP fact storage by introducing the first construction for fact-storing MLPs that provably comes within a polylog factor of matching the empirical facts-per-parameter scaling of LLMs.

Despite progress from recent constructions, particularly Nichani et al. (2024), several key questions remain unanswered about the mechanics and properties of MLPs as fact-storage devices:

Q1: How do MLP input and output geometries affect fact-storage capacity? Existing fact-storing MLP constructions (Nichani et al., 2024) assume that inputs and outputs are uniformly distributed, even though MLPs in the wild have uncentered and non-uniform inputs and outputs (Section 4).

*Equal first author

†Corresponding author: odugan@stanford.edu

‡Preprint, working draft version.

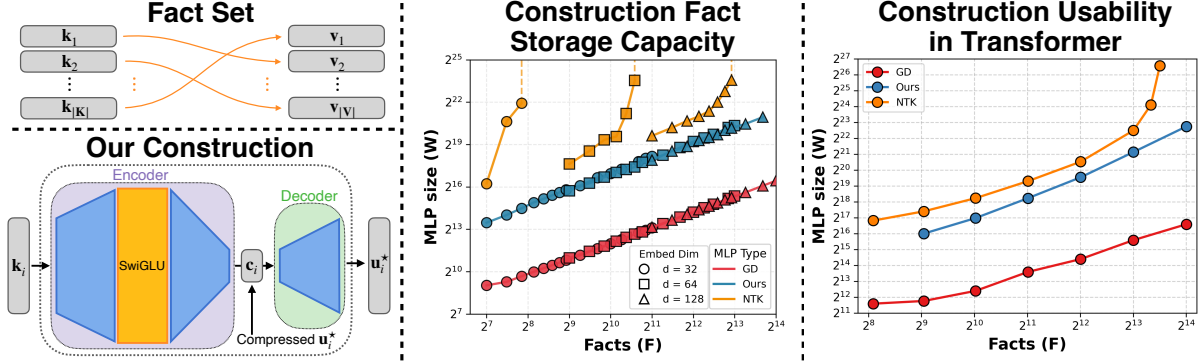


Figure 1: **(Left) Top:** We formalize factual knowledge as discrete maps between key and value embeddings. **Bottom:** Our construction consists of an *encoder* MLP that exactly maps keys to compressed intermediate values, and a *decoder* linear layer that linearly decompresses the intermediate values. **(Center)** We compare how the number of parameters (y -axis) needed to represent a fact set scales with the number of facts (x -axis). Our construction matches gradient-descent trained (GD) MLP asymptotics and requires 5–150 $\times$ fewer parameters than prior constructions. **(Right)** We compare how the number of parameters (y -axis) needed for an MLP to represent a fact set in a way that is *usable within a transformer* scales with the number of facts (x -axis). Our constructed MLPs exhibit similar asymptotic scaling to GD MLPs, unlike NTK MLPs. *Note:* NTK refers to the construction from Nichani et al. (2024).

Q2: How do MLPs achieve parameter-efficient fact-storage? Existing constructions still fall short of explaining the fact-storage efficiency observed in practice. For instance, the theoretical guarantees in Nichani et al. (2024) suggest that their construction stores $O(\log^{11} F)$ fewer facts per parameter than the information-theoretic optimal for a fact set of size F .

Q3: How do fact-storing MLPs interface with the rest of the Transformer stack? Prior work focuses on MLP constructions in isolation (Bubeck et al., 2020; Nichani et al., 2024) or the capacity of a full Transformer stack at once (Allen-Zhu & Li, 2024). However, we still lack a clear understanding of how a transformer might learn to perform recall tasks using a fact-storing MLP.

We address each of the above questions by improving over existing constructed fact-storing MLPs in a way that uncovers new insights into fact-storing MLPs more broadly. Together, our improvements form an MLP construction framework which produces MLPs that 1) work on all but a measure-zero set of feasible MLP inputs and outputs, 2) match asymptotic information theoretic lower bounds on parameter count for some embeddings, and 3) can be directly used by transformers for factual recall. These improvements allow us to 1) discover a metric on value embeddings that is predictive of MLP facts-per-parameter scaling for both our constructed MLPs and gradient-descent-trained MLPs (GD MLPs), 2) identify a simple encoder-decoder mechanism which is sufficient to empirically match GD MLP facts-per-parameter asymptotics across all of inputs and outputs we test, and 3) identify a fundamental capacity-usability tradeoff for MLPs inside transformers.

Q1: In Section 3, we study the effect of desired output geometry on MLP capacity. We improve the construction from Nichani et al. (2024), improving facts-per-parameter scaling by 2–4 $\times$ and extending it to anisotropic output distributions through an output-whitening procedure. These improvements provide an insight into MLP scaling: we propose a measure, the *decodability*, which predicts fact-storage capacity for both constructed and GD MLPs with an R^2 greater than 97%.

Q2: In Section 4, we improve over existing constructions by providing an MLP construction framework requiring asymptotically fewer parameters than the lowest proven bounds for existing constructions, while also generalizing to nearly all feasible input and output distributions. Our closed-form constructed MLPs match the information-theoretic lower bound for some embeddings, empirically require 5–150 $\times$ fewer parameters than NTK MLPs, and are the first constructed MLPs to match GD MLP asymptotics regardless of input/output dimension. This construction leads to a key insight about fact-storing MLPs: a simple encoder-decoder MLP framework using dimensionality reduction on the desired MLP outputs (e.g., Johnson et al. (1984)) can asymptotically match information-theoretically optimal facts-per-parameter scaling.

Q3: In Section 5, we improve existing constructions by identifying a set of modifications to the transformer architecture that enable training a transformer block to use fact storing MLPs for factual

recall. We find that our transformer block can use our constructed MLPs, storing an amount of facts per parameter comparable to the information-theoretically optimal, unlike previous constructions. Additionally, we gain insight into fact-storing MLPs interactions with transformers by identifying a fundamental tradeoff between their capacity and usability in transformers.

Finally, in Section 5.4, inspired by our results on MLP usability within transformers, we demonstrate modular fact editing in 1-layer transformers as an application of fact-storing MLPs. If, given a transformer block, we modularly swap its fact-storing MLP with another one storing new facts, the transformer outputs the new facts accurately and only increases the cross-entropy loss of non-fact-related tokens by $\sim 3\%$ *without any additional training*. Further, our modular MLP-swapping approach to fact editing doubles the *fact-editing score* (defined in Section 5.4) of SoTA fact-editing weight updates (e.g. MEMIT Meng et al. (2023c), Alpha-Edit Fang et al. (2025), and ROME Meng et al. (2023b)) when editing 10% of the fact set.

In summary, we present a construction that a) supports a broader class of embeddings than prior constructions, b) produces MLPs with asymptotically fewer parameters than the bounds proven for alternative constructions, and c) produces MLPs that are usable within transformers for factual recall. We use this construction to gain insights into 1) MLP fact-storage capacity’s dependence on output geometry, 2) mechanisms behind MLP facts-per-parameter scaling, and 3) the tradeoff between MLP capacity and usability in transformers. By directly constructing MLPs to store facts, we provide a theoretical framework for studying fact storage and a path toward more robust fact manipulation in LLMs.

2 Preliminaries

2.1 Definitions

We first formalize our notion of factual knowledge, which matches the definitions of Nichani et al. (2024).

Formalizing Factual Knowledge. Inspired by prior work (Nichani et al., 2024; Arora et al., 2023; Allen-Zhu & Li, 2024), we define a *fact set* as a discrete mapping between integers. In particular, given a list of keys K and a list of values V , a fact set is a function $f : [|K|] \rightarrow [|V|]$. For example, given $K = [\text{“France”}, \text{“USA”}]$ and $V = [\text{“Washington, D.C.”}, \text{“Paris”}]$, the fact set mapping countries to capitals would be $f(1) = 2, f(2) = 1$.

Although we use human-interpretable examples of key-value maps above, our definition of fact sets applies broadly to transformer tasks. In particular, a language model specifies a fixed vocabulary and encodes maps between tokens as maps between integers, which is also representable in this framework.

Transformers interface with tokens through embedding tables. Motivated by this, we consider *key embeddings* $\mathbf{K} \in \mathbb{R}^{|K| \times d}$ and *value embeddings* $\mathbf{V} \in \mathbb{R}^{|V| \times d}$, which map keys and values, respectively, to vectors. We define $|\mathbf{K}|$ and $|\mathbf{V}|$ as the number of key and value embeddings, respectively, and we denote the i th key and value embedding as $\mathbf{k}_i$ and $\mathbf{v}_i$, respectively. In the case of MLPs within transformers, key and value embeddings come from the internal representations of the surrounding transformer.

Storing a fact set. We say that a model $\mathbf{g}_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ stores a fact set $f : [|K|] \rightarrow [|V|]$ given embeddings $\mathbf{K}$ and $\mathbf{V}$ if, for all $i \in [|K|]$, and all $j \neq f(i) \in [|V|]$,

$$\langle \mathbf{g}_\theta(\mathbf{k}_i), \mathbf{v}_{f(i)} \rangle > \langle \mathbf{g}_\theta(\mathbf{k}_i), \mathbf{v}_j \rangle, \quad (1)$$

or, equivalently, $\langle \mathbf{g}_\theta(\mathbf{k}_i), \mathbf{v}_{f(i)} - \mathbf{v}_j \rangle > 0$. In the context of language modeling, this definition is equivalent to outputting the correct value token for each input key token under softmax decoding (see Section B.2). For an MLP output $\mathbf{o}$, we refer to $\langle \mathbf{o}, \mathbf{v}_i \rangle$ as the *score* of $\mathbf{o}$ with respect to the i th value.

We define the *fact-storage cost* of key/value embeddings $\mathbf{K}$ and $\mathbf{V}$ given a model class $\mathbf{g}$ as the smallest number of model parameters needed to store *all possible fact sets* over those embeddings:

$$W(\mathbf{g}; \mathbf{K}, \mathbf{V}) = \min \left\{ \#(\theta) \left| \begin{array}{l} \forall f : [|K|] \rightarrow [|V|], \\ \exists \theta \text{ s.t. } \mathbf{g}_\theta \text{ stores } f \end{array} \right. \right\}. \quad (2)$$

A standard information-theoretic lower bound for fact storage cost (Allen-Zhu & Li, 2024), which we prove for completeness in Section B.2, is the following:

Proposition 2.1.1. *Assuming a constant number of bits per parameter, the fact-storage cost of embeddings $\mathbf{K}$ and $\mathbf{V}$ for any model family $\mathbf{g}$ satisfies $W(\mathbf{g}; \mathbf{K}, \mathbf{V}) = \Omega(|\mathbf{K}| \log |\mathbf{V}|)$.*

Following prior work (Allen-Zhu & Li, 2024; Zucchet et al., 2025), we define the *fact-storage capacity* of a model as the maximum number of facts it can store for a given number of parameters. See Section B.2 for a formal definition.

2.2 Related Work

A first body of prior work has attempted to understand and manipulate LLM knowledge storage by probing pretrained LLMs. Geva et al. (2021, 2022) observed that knowledge is often stored within MLPs via key-value mappings. This discovery sparked a number of studies which attempt to reverse engineer the facts found in MLPs (Dai et al., 2022; Nanda et al., 2023).

After identifying the facts stored by individual LLM MLPs, researchers naturally turned to editing this knowledge. Works such as Dai et al. (2022); Meng et al. (2023a,c); Gupta et al. (2024); Gu et al. (2024); Fang et al. (2025); Sun et al. (2025) have developed increasingly more accurate, general, and targeted methods for editing of specific facts within LLM MLPs.

Building on the insights from probing LLMs, a second body of work attempts to formalize factual knowledge, often focusing on its scaling. Typically, these works treat knowledge as a key-value store and study the scaling of factual knowledge through associative recall synthetics (Allen-Zhu & Li, 2024; Zucchet et al., 2025), design choices which we also follow. Remarkably, these works consistently find empirically that trained LLMs store facts at the asymptotically optimal rate provided in Theorem 2.1.1 (Allen-Zhu & Li, 2024; Zucchet et al., 2025; Morris et al., 2025).

The discovery that trained MLPs store facts at the asymptotically optimal rate raises the question of how MLPs achieve such a scaling. In an attempt to answer this question, Elhage et al. (2022) have explored the geometric properties and learning dynamics of MLPs that store a large number of facts. Recently, Nichani et al. (2024) have taken an additional step toward uncovering the mechanisms underlying MLP fact storage; they propose a construction for fact-storing MLPs that comes within a (large) polylog factor of matching the asymptotic fact-scaling of LLM MLPs.

In this work, we improve upon the results of Nichani et al. (2024) by a) improving MLP fact-storage cost asymptotics, b) handling more general input and output embeddings, and c) enabling constructed MLPs to be usable within transformers. We use insights from our construction to gain insight into fact-storing MLPs.

3 Embedding Geometry and Fact-Storage Cost

In this section, we investigate how the fact-storage cost of an MLP depends on the geometry of a fact set’s value embeddings. We first gain insight into fact-storing MLPs by developing a metric on the value embeddings which is predictive of MLP fact-storage cost, achieving an $R^2 > 97\%$ (Section 3.1). Further, we use this insight to improve the NTK construction from Nichani et al. (2024), by generalizing it to non-isotropic embeddings with an embedding by using an embedding whitening procedure. Moreover, we enhance gradient-descent-trained MLPs (GD MLPs), reducing its fact-storage cost for non-isotropic embeddings (Section 3.3) using the same procedure.

3.1 A Metric $\rho(\mathbf{V})$ that Predicts Fact-Storage Cost

First, we introduce $\rho(\mathbf{V})$ to measure the *decodability* of value embeddings $\mathbf{V}$. Intuitively, $\rho(\mathbf{V})$ is the minimum normalized margin between the margin-optimal MLP outputs $\mathbf{U}^* \in \mathbb{R}^{n,d}$ and the value embeddings $\mathbf{V} \in \mathbb{R}^{n,d}$.

Definition 3.1.1. *The decodability $\rho(\mathbf{V})$ of embeddings $\mathbf{V}$ is*

$$\rho(\mathbf{V}) = \max_{\mathbf{u}_i \in \mathbb{R}^d} \left[\min_{i \neq j} \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle}{\|\mathbf{u}_i\|_2 \|\mathbf{v}_i - \mathbf{v}_j\|_2} \right]. \quad (3)$$

Given the margin-optimal output embeddings $\mathbf{u}_i$, $\rho(\mathbf{V})$ measures the minimum margin $\langle \mathbf{u}_i, \mathbf{v}_i \rangle - \langle \mathbf{u}_i, \mathbf{v}_j \rangle$ normalized by $\|\mathbf{u}_i\|_2$ and $\|\mathbf{v}_i - \mathbf{v}_j\|_2$ ¹. Such a normalization ensures that arbitrary scalings of $\mathbf{u}_i$ or $\mathbf{v}_i$ do

¹A related notion is the *coherence* of the value embeddings, defined as $\mu(\mathbf{V}) = \max_{i \neq j} \frac{|\langle \mathbf{v}_i, \mathbf{v}_j \rangle|}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|}$. When all $\mathbf{v}_i$ have unit norm, one can show that $\rho(\mathbf{V}) \geq \sqrt{[1 - \mu(\mathbf{V})]/2}$. However, no corresponding *upper* bound on $\rho(\mathbf{V})$ in terms of $\mu(\mathbf{V})$ exists in general (Appendix B.5.1). Empirically, coherence is not as predictive of fact-storage cost as $\rho(\mathbf{V})$ is for either our constructed MLPs ($R^2 \approx 0.44$) or GD MLPs ($R^2 \approx 0.10$): see Figure 6. This helps motivate the use of $\rho(\mathbf{V})$ rather than coherence as the relevant geometric predictor of decoding difficulty.

not affect the decoding difficulty of $\mathbf{V}$, as one would expect. Notably, the quantity $\rho(\mathbf{V})$ also appears naturally in our decoder construction in Section 4.2.

$\rho(\mathbf{V})$ predicts fact storage capacity. In Figure 2a, we find empirically that fact-storage cost scales inversely with ρ for both our constructed MLPs (presented in Section 4) and GD MLPs. We show that ρ is predictive of fact set difficulty ($R^2 > 97\%$), as measured by the size of MLP required to store a fact set, for both our constructed MLPs and GD MLPs. This ability to predict capacity for multiple types of fact-storing MLPs suggests that ρ is not a construction-dependent quantity, and that it is instead a property of *near-optimal* fact-storing MLPs.

3.2 Defining Optimal MLP Outputs

Interestingly, using $\mathbf{u}_i = \mathbf{v}_i$ is generally suboptimal for decoding to index i of $\mathbf{V}$.

As an extreme case, consider the embeddings $\mathbf{v}_1 = \mathbf{e}_1$ and $\mathbf{v}_2 = 2\mathbf{e}_1$. If we wish to select an output that decodes to index 1, outputting $\mathbf{v}_1 = \mathbf{e}_1$ is incorrect and will instead decode to index 2. In fact, outputting $-\mathbf{e}_1$ is optimal, in the sense that it is the unit vector that maximizes the gap between its score with respect to $\mathbf{v}_1$ ($\text{score}_1 = \langle -\mathbf{e}_1, \mathbf{v}_1 \rangle = -1$) and its score with respect to $\mathbf{v}_2$ ($\text{score}_2 = \langle -\mathbf{e}_1, \mathbf{v}_2 \rangle = -2$).

Instead, we can define the margin-optimal output embeddings as the unit $\mathbf{u}_i$ that achieve the maximum value in the definition of $\rho(\mathbf{V})$:

Definition 3.2.1. The margin-optimal output embeddings (*optimal output embeddings for short*) $\mathbf{U}^* \in \mathbb{R}^{|\mathbf{V}| \times d}$ for value embeddings $\mathbf{V}$ is

$$\mathbf{u}_i^*(\mathbf{V}) = \arg \max_{\mathbf{u} \in \mathbb{S}^{d-1}} \left[\min_j \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u} \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\|_2} \right]. \quad (4)$$

We can obtain $\mathbf{u}_i^*$ as the solution to a convex program by relaxing the domain to $\|\mathbf{u}_i\|_2 \leq 1$ (See Appendix B).

Interestingly, $\mathbf{u}_i^*$ is the spherical Chebyshev center (Vrahatis, 2024) of the set $S_i = \{\mathbf{v}_i - \mathbf{v}_j \mid j \neq i\}$. Similarly, $\rho(\mathbf{V})$ is the maximum of the spherical Chebyshev radii of the S_i . We explore the resulting bounds on $\rho(\mathbf{V})$ in Appendix B.

3.3 Embedding Whitening

Interestingly, the *decodability* ρ is *not* invariant to affine transformations of the value embeddings, but MLPs *are* equivariant to such transformations. If the MLP $\mathbf{g}(x) = \mathbf{B} \text{ReLU}(\mathbf{A}\mathbf{x} + \mathbf{b})$ stores a fact given the value embeddings $\{\mathbf{v}_i\}$, then for any invertible affine transformation of the value embeddings² $T(\mathbf{v}) = \mathbf{M}\mathbf{v} + \mathbf{c}$ for $\mathbf{M} \in \text{GL}(d)$, $\mathbf{c} \in \mathbb{R}^d$, the reparameterized MLP $\tilde{\mathbf{g}}(\mathbf{x}) = \tilde{\mathbf{B}} \text{ReLU}(\mathbf{A}\mathbf{x} + \mathbf{b})$ stores the fact set given value embeddings $\{T(\mathbf{v}_i)\}$, where $\tilde{\mathbf{B}} = \mathbf{M}^{-1}\mathbf{B}$.³

This motivates the following procedure for improving the fact-storage cost of MLPs. Given embeddings $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\} \subset \mathbb{R}^d$, we search for an invertible affine transform $T(\mathbf{v})$ that maximizes the decodability of the transformed set:

$$\max_{\mathbf{M} \in \text{GL}(d), \mathbf{c} \in \mathbb{R}^d} \rho(\{T(\mathbf{v}_i)\}_{i=1}^n). \quad (5)$$

Let $\tilde{\mathbf{V}} = \{T(\mathbf{v}_i)\}_{i=1}^{|\mathbf{V}|}$ denote the resulting embeddings, so that $\rho(\tilde{\mathbf{V}}) \geq \rho(\mathbf{V})$. We then train or construct the MLP on $\tilde{\mathbf{V}}$, then fold the affine transformation into the network parameters.

We find that a simple heuristic choice of transformation, where $\mathbf{M}$ is the whitening transform of the empirical covariance of $\mathbf{V}$ and $\mathbf{c}$ is the negative of the mean of $\mathbf{V}$, often improves the decodability: see Section B.7 for formal bounds. We refer to this procedure as *embedding whitening*, and we refer to MLPs trained or constructed with and without embedding whitening as *whitened* and *non-whitened* MLPs, respectively.

Embedding whitening improves fact storage capacity. In Figure 2a, we find that embedding whitening improves constructed MLP fact-storage cost⁴ for embeddings with low ρ by up to $32\times$. However,

²Here $\text{GL}(d)$ is the usual set of $d \times d$ real valued matrices with non-zero determinant.

³We prove this for completeness in Theorem B.7.1.

⁴For Figure 2a, to obtain embeddings with small ρ , we use embeddings which are sampled uniformly from a unit sphere and then multiplied by an ill-conditioned transformation matrix. For this choice of embeddings, whitening exactly removes the dependence on ρ , but for other embeddings a dependence on ρ may remain (See Appendix A).

Algorithm 1 Fact-Storing MLP Framework

Require: $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$, $\mathbf{V} \in \mathbb{R}^{|\mathbf{V}| \times d}$, $f: [|\mathbf{K}|] \rightarrow [|\mathbf{V}|]$

Require: Hidden size h , compressed dim. m , activation σ

- 1: $(\mathbf{C} \in \mathbb{R}^{|\mathbf{V}| \times m}, \mathbf{D} \in \mathbb{R}^{d \times m}) \leftarrow \text{DEC}(\mathbf{V}, m)$
 - 2: $(\mathbf{A}, \mathbf{G} \in \mathbb{R}^{h \times d}, \mathbf{E} \in \mathbb{R}^{m \times h}) \leftarrow \text{ENC}(\mathbf{K}, \mathbf{C}, f, h, \sigma)$
 - 3: $\text{MLP}(\mathbf{x}) := \text{DE}(\sigma(\mathbf{G}\mathbf{x}) \odot (\mathbf{A}\mathbf{x}))$
 - 4: **return** MLP
-

as we will show in Section 5, whitening the embeddings results in MLPs with large Lipschitz constant that are harder to use within transformers.

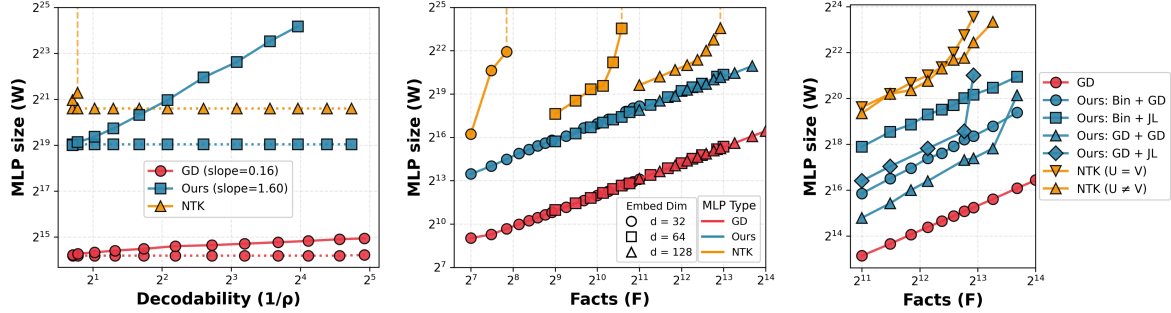


Figure 2: (a) For both GD and our constructed MLPs, ρ is predictive ($R^2 > 0.97$) of MLP size for a fixed number of facts. Embedding whitening reduces our constructed MLPs’ fact-storage cost by up to $32\times$ and allows NTK MLPs to generalize to highly anisotropic embeddings. (b) GD MLPs and our constructed MLPs exhibit consistent facts-per-parameter scaling as embedding dimension and number of facts vary jointly, whereas NTK MLPs exhibit asymptotically worse scaling as more facts are squeezed into a fixed embedding dimension (pictured for spherical embeddings). Our constructed MLPs have between $5\text{--}150\times$ lower fact-storage cost than NTK MLPs, while GD MLPs have $\sim 20\times$ lower fact-storage cost than ours. (c) When training the encoder and decoder with gradient descent, the fact-storage cost gap to GD MLPs narrows from $\sim 20\times$ to $\sim 4\times$.

4 MLP Constructions

We now present our framework for fact-storing MLPs (Algorithm 1). The core insight of our framework is to define *compressed output embeddings* $\mathbf{C} \in \mathbb{R}^{|\mathbf{V}| \times m}$ and to decompose the MLP into an *encoder*, which maps keys $\mathbf{k}_i$ to compressed outputs $\mathbf{c}_{f(i)}$, and a *decoder*, which decompresses $\mathbf{c}_{f(i)}$ into an output in $\mathbb{R}^d$ which decodes to $\mathbf{v}_{f(i)} \in \mathbb{R}^d$. This encoder-decoding framework is sufficient to match the asymptotic scaling of GD MLPs’ fact-storage cost across a range of embeddings.

In Section 4.1 and Section 4.2, we present the details of the encoder and decoder portions of our frameworks, respectively. For each, we 1) present the encoder/decoder structure and objective, 2) demonstrate how an encoder/decoder can be obtained through gradient descent, and 3) present explicit, closed-form weight constructions with asymptotic analysis.

In Section 4.3 we present the full construction and show that it provides tighter asymptotic fact-storage cost than has been proven for prior constructions, even matching the information-theoretic lower bounds in some cases. Finally, in Section 4.4 we demonstrate empirically that 1) our construction has a lower fact-storage cost than prior constructions and 2) unlike prior constructions, our construction’s *fact-storage cost* scaling matches that of GD MLPs even when varying the number of facts or input-output dimensions independently.

4.1 The Encoder

Our encoder is a single-hidden layer MLP mapping key embeddings to compressed output embeddings.

Algorithm 2 Encoder Construction (ENC)

Require: $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$, $\mathbf{C} \in \mathbb{R}^{|\mathbf{V}| \times m}$, $f : [|\mathbf{K}|] \rightarrow [|\mathbf{V}|]$

Require: Hidden size h , activation σ

- 1: $\tilde{h} := h/m$
- 2: **for** $j = 1$ **to** m **do**
- 3: $\mathbf{o}^{(j)} := [\mathbf{C}_{f(1),j}, \dots, \mathbf{C}_{f(|\mathbf{K}|),j}] \in \mathbb{R}^{|\mathbf{K}|}$
- 4: $(\mathbf{A}^{(j)}, \mathbf{G}^{(j)}) \in \mathbb{R}^{\tilde{h} \times d} \leftarrow \text{ENCGAD}(\mathbf{K}, \mathbf{o}^{(j)}, \tilde{h}, \sigma)$
- 5: **end for**
- 6: Stack encoder gadgets $\mathbf{A}, \mathbf{G} \in \mathbb{R}^{m \times d}$:

$$\mathbf{A} := \begin{bmatrix} \mathbf{A}^{(1)} \\ \vdots \\ \mathbf{A}^{(m)} \end{bmatrix}, \quad \mathbf{G} := \begin{bmatrix} \mathbf{G}^{(1)} \\ \vdots \\ \mathbf{G}^{(m)} \end{bmatrix}$$

$$7: \mathbf{E} := \begin{bmatrix} \mathbf{1}_{1 \times \tilde{h}} & \mathbf{0}_{1 \times \tilde{h}} & \cdots & \mathbf{0}_{1 \times \tilde{h}} \\ \mathbf{0}_{1 \times \tilde{h}} & \mathbf{1}_{1 \times \tilde{h}} & \cdots & \mathbf{0}_{1 \times \tilde{h}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1 \times \tilde{h}} & \mathbf{0}_{1 \times \tilde{h}} & \cdots & \mathbf{1}_{1 \times \tilde{h}} \end{bmatrix} \in \mathbb{R}^{m \times h}$$

8: **return** $(\mathbf{A}, \mathbf{G}, \mathbf{E})$

Encoder Structure Our encoder is a gated MLP⁵

$$\mathbf{enc}(\mathbf{x}) = \mathbf{E}(\sigma(\mathbf{G}\mathbf{x}) \odot (\mathbf{A}\mathbf{x}))$$

where $\mathbf{A}, \mathbf{G} \in \mathbb{R}^{h \times d}$, $\mathbf{E} \in \mathbb{R}^{m \times h}$, $\mathbf{x} \in \mathbb{R}^d$, and $\sigma : \mathbb{R}^h \rightarrow \mathbb{R}^h$ is an activation function.

Gated MLPs simplify our analysis and are now popular across frontier models (Yang et al., 2025b; Dubey et al., 2024). In Section B, we extend to non-gated MLPs.

Encoder Framework Objective Given key embeddings $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$, compressed output embeddings $\mathbf{C} \in \mathbb{R}^{|\mathbf{V}| \times m}$, and a mapping f , the objective of our encoder framework is to produce an MLP $\mathbf{enc}$ with a minimal number of parameters such that $\mathbf{enc}(\mathbf{k}_i) = \mathbf{c}_{f(i)}$ for all $i \in [|\mathbf{K}|]$.

Gradient-Descent Construction One strategy to build an encoder MLP is to use gradient descent (a *GD Encoder*) by optimizing for $\mathbf{enc}$ in the Mean-Squared Error (MSE) objective

$$\mathcal{L}(\mathbf{K}, \mathbf{C}; \mathbf{enc}) = \sum_{i \in [|\mathbf{K}|]} \|\mathbf{enc}(\mathbf{k}_i) - \mathbf{c}_{f(i)}\|^2.$$

Closed-Form Weight Construction Alternatively, we can construct an encoder via a closed-form weight construction. Our constructed encoder builds m encoder gadgets⁶

$$\mathbf{enc}_j(\mathbf{x}) = \mathbf{1}_{\tilde{h}}^\top [\sigma(\mathbf{G}\mathbf{x}) \odot (\mathbf{A}\mathbf{x})], \quad \mathbf{G}, \mathbf{A} \in \mathbb{R}^{\tilde{h} \times d},$$

that map $\mathbf{k}_i$ to $\mathbf{c}_{f(i)}[j] \in \mathbb{R}$, respectively, where $\tilde{h} = h/m$. We will demonstrate that these gadgets require only $O(|\mathbf{K}|)$ parameters. By stacking all m gadgets together, one for each target dimension j , we can construct $\mathbf{c}_{f(i)}$ with a total of $O(m|\mathbf{K}|)$ parameters, as shown in Algorithm 2.

Simple Two-Hot Encoder Gadget: For clarity, we first present the encoder gadget in a simplified setting (Construction 4.1), where the key embeddings are *two-hot*, i.e., $\mathbf{K} = \{\mathbf{e}_i - \mathbf{e}_j \in \mathbb{R}^d \mid i \neq j \in [d]\}$, with $|\mathbf{K}| = d(d-1)$.

Intuitively, Construction 4.1 involves two sequential steps: 1) pick a gating term that selects different portions of the input for different hidden neurons (in the case below, $\text{ReLU}(\mathbf{I}_d \mathbf{x})$) and 2) find the $\mathbf{A}$ that fits the data. These two steps underlie our generalization of Construction 4.1 to arbitrary gating functions and embeddings.

⁵For the rest of Section 4, we drop biases for notational simplicity.

⁶We can set the down projection to $\mathbf{1}^\top$ without loss of generality by replacing $\mathbf{A}$ with $\text{diag}(\mathbf{E})\mathbf{A}$.

Construction 4.1 (Encoder, Two-Hot). *Let*

$$h : \{(i, j) \mid i \neq j \in [d]\} \rightarrow \mathbb{R}$$

be a function mapping each pair (i, j) to the desired output for key embedding $\mathbf{e}_i - \mathbf{e}_j$. Define $\mathbf{enc}(\mathbf{x}) = \mathbf{1}_d^\top [\text{ReLU}(\mathbf{I}_d \mathbf{x}) \odot (\mathbf{A} \mathbf{x})]$, where $\mathbf{A} \in \mathbb{R}^{d \times d}$ with

$$\mathbf{A}[p, q] = \begin{cases} 0 & \text{if } p = q \\ -h(p, q) & \text{if } p \neq q. \end{cases}$$

Then $\mathbf{enc}(\mathbf{e}_i - \mathbf{e}_j) = h(i, j)$ for all $i \neq j \in [d]$. This encoder has $2|\mathbf{K}| + O(d)$ parameters.⁷

Proof:

$$\begin{aligned} & \text{ReLU}(\mathbf{I}_d(\mathbf{e}_i - \mathbf{e}_j)) \odot (\mathbf{A}(\mathbf{e}_i - \mathbf{e}_j)) \\ &= \mathbf{e}_i \odot (\mathbf{A}(\mathbf{e}_i - \mathbf{e}_j)) \\ &= (\mathbf{A}[i, i] - \mathbf{A}[i, j]) \mathbf{e}_i \\ &= h(i, j) \mathbf{e}_i. \end{aligned}$$

Finally, multiplying by $\mathbf{1}_d^\top$ extracts $h(i, j)$. □

A Generalized Gated Encoder Gadget: Following the two-hot example, our generalized gated encoder gadget will follow two simple steps: 1) pick $\mathbf{G}$, and 2) solve the resulting linear system for $\mathbf{A}$. The rest of this section will be dedicated to defining the linear system for $\mathbf{A}$ and providing conditions for a solution to exist.

Define

$$\begin{aligned} \Sigma &= \sigma(\mathbf{G}\mathbf{K}^\top) \in \mathbb{R}^{h \times |\mathbf{K}|} \\ \mathbf{o} &= [\mathbf{c}_{f(1)}[j], \dots, \mathbf{c}_{f(|\mathbf{K}|)}[j]]^\top \\ \mathbf{M}(\Sigma, \mathbf{K}) &= [\text{diag}(\Sigma_1)\mathbf{K}, \dots, \text{diag}(\Sigma_h)\mathbf{K}] \in \mathbb{R}^{|\mathbf{K}| \times dh}. \end{aligned}$$

The $\mathbf{A}$ matrices such that $\mathbf{enc}(\mathbf{k}_i) = \mathbf{c}_{f(i)}[j]$ for all $i \in |\mathbf{K}|$ are exactly the solutions to the linear system⁸:

$$\mathbf{M}(\Sigma, \mathbf{K}) \text{vec}(\mathbf{A}) = \mathbf{o}$$

To obtain a construction, we need to choose Σ such that the system is solvable for every choice of $\mathbf{o}$, which is true if and only if $\mathbf{M}(\Sigma, \mathbf{K})$ has full row-rank. Interestingly, this is true for generic $\mathbf{K}$ provided a simple rank condition on Σ :

Lemma 4.1.1. *The matrix $\mathbf{M}(\Sigma, \mathbf{K})$ has full row-rank for generic⁹ $\mathbf{K}$ if and only if*

$$d \cdot \text{rank}(\Sigma[:, S]) \geq |S| \quad \forall S \subseteq [|\mathbf{K}|]. \quad (6)$$

Further, for analytic σ , such a Σ is easy to find:

Lemma 4.1.2. *Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be a non-polynomial analytic activation. As long as $dh \geq |\mathbf{K}|$, for generic $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$ and $\mathbf{G} \in \mathbb{R}^{h \times d}$, we have that $\Sigma = \sigma(\mathbf{G}\mathbf{K}^\top)$ satisfies Equation 6.*

Putting these results together gives the more general construction in Algorithm 3, proven in Appendix B.4 along with generalizations to other activations functions σ such as ReLU.

⁷By a simple degrees-of-freedom argument, any MLP that can map $d^2 - d = |\mathbf{K}|$ inputs each to an arbitrary real requires at least $|\mathbf{K}|$ parameters. Hence, the construction is asymptotically optimal in parameter count.

⁸We define $\text{vec}(\mathbf{A}) = [\mathbf{a}_1, \dots, \mathbf{a}_h]^\top \in \mathbb{R}^{dh}$.

⁹I.e., for all $\mathbf{K}$ in a Zariski open set. The set of $\mathbf{K}$ not satisfying this condition is measure 0.

Algorithm 3 Encoder Gadget Construction (ENC_{GAD})

Require: $\mathbf{o} \in \mathbb{R}^{|\mathbf{K}|}$, generic $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$

Require: Hidden size h with $dh \geq |\mathbf{K}|$, analytic σ

1: Sample generic $\mathbf{G} \in \mathbb{R}^{h \times d}$ (e.g. i.i.d. Gaussian)

2: $\Sigma := \sigma(\mathbf{G}\mathbf{K}^\top) \in \mathbb{R}^{h \times |\mathbf{K}|}$

3: $\mathbf{M} := [\text{diag}(\Sigma_1)\mathbf{K}, \dots, \text{diag}(\Sigma_h)\mathbf{K}] \in \mathbb{R}^{|\mathbf{K}| \times (dh)}$

4: Solve for $\mathbf{v} \in \mathbb{R}^{dh}$ in $\mathbf{M}\mathbf{v} = \mathbf{o}$

5: $\mathbf{A} := \begin{bmatrix} \mathbf{v}[1:d] \\ \mathbf{v}[d+1:2d] \\ \vdots \\ \mathbf{v}[(h-1)d+1:hd] \end{bmatrix} \in \mathbb{R}^{h \times d}$

6: **return** $(\mathbf{A}, \mathbf{G})$

Asymptotic Analysis When m copies of the generalized encoder gadget from Algorithm 3 are stacked to produce full output vectors, the full encoder contains $2m|\mathbf{K}| + O(md) + O(mh)$ parameters, which for $d, h = o(|\mathbf{K}|)$ is within a factor of two of the degrees-of-freedom lower bound of $m|\mathbf{K}|$ (up to lower order terms).

To our knowledge, our generalized encoder gadget is the first demonstration that gated MLPs can exactly memorize N generic datapoints with $O(N)$ parameters, asymptotically matching the degrees-of-freedom lower bound.

In Appendix B.4, we show that our results extend to non-gated MLPs (up to an arbitrarily small ϵ error) by implementing a neural tangent kernel approximation similar to Nichani et al. (2024). Interestingly, when this generalization is applied to ReLU MLPs, we obtain a construction which generalizes that from Bubeck et al. (2020).

Naively, if we allow $m = d$, the encoder alone could output the target embeddings exactly. However, this construction would yield an MLP with $\Theta(d|\mathbf{K}|)$ parameters, which does not match the information-theoretic limit of $\Omega(|\mathbf{K}| \log |\mathbf{V}|)$ from Theorem 2.1.1. As we explore in the next subsection, we can obtain a $\Theta(|\mathbf{K}| \log |\mathbf{V}|)$ construction by instead setting $m < d$ and picking *compressed output embeddings* that can be approximately decoded into the optimal output embeddings.

4.2 The Decoder and ρ

We next describe our decoder framework.

Decoder Structure The decoder consists of a single linear layer $\text{dec}(\mathbf{x}) = \mathbf{D}\mathbf{x}$, where $\mathbf{D} \in \mathbb{R}^{d \times m}$ and $\mathbf{x} \in \mathbb{R}^m$.

Decoder Framework Objective Given value embeddings $\mathbf{V} \in \mathbb{R}^{|\mathbf{V}| \times d}$, the objective of our decoder framework is to produce 1) compressed output embeddings $\mathbf{C} \in \mathbb{R}^{|\mathbf{V}| \times m}$ and 2) a decoder dec such that

$$\langle \mathbf{v}_i, \text{dec}(\mathbf{c}_i) \rangle > \langle \mathbf{v}_j, \text{dec}(\mathbf{c}_i) \rangle, \quad \forall i \neq j \in [|\mathbf{V}|], \quad (7)$$

for a minimal value of m . We seek to minimize m because the overall MLP parameter count is proportional to m .

Gradient Descent Construction We can easily construct such a pair of compressed output embeddings and a decoder linear layer using gradient descent (a *GD Decoder*) by optimizing for $\mathbf{C}$ and $\mathbf{D}$ in the objective

$$\mathcal{L}(\mathbf{C}, \mathbf{D}, \mathbf{K}) = \sum_{i \neq j \in [|\mathbf{V}|]} \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{D}\mathbf{c}_i \rangle.$$

Closed-Form Weight Construction We will now provide a closed-form construction for such a decoder framework where $m = O(\log |\mathbf{V}|)$ with high probability for most embedding common embeddings

Algorithm 4 Decoder Construction (DEC)

Require: $\mathbf{V} \in \mathbb{R}^{|\mathbf{V}| \times d}$, compressed dimension m

- 1: $\mathbf{U}^* \in \mathbb{R}^{|\mathbf{V}| \times d} \leftarrow \text{OPTIMALOUT}(\mathbf{V})$
 - 2: Sample an i.i.d. Gaussian matrix $\mathbf{D} \in \mathbb{R}^{d \times m}$
 - 3: $\mathbf{C} := \mathbf{U}^* \mathbf{D} \in \mathbb{R}^{|\mathbf{V}| \times m}$
 - 4: **return** $(\mathbf{C}, \mathbf{D})$
-

Table 1: Comparison of construction fact storage costs and assumptions. Nichani et al. (2024) assumes $|\mathbf{K}| = |\mathbf{V}|$. The naïve construction is detailed in Section B.3.1.

	Parameters	Hidden Sizes	Assumptions on $\mathbf{K}$	Assumptions on $\mathbf{V}$
Info-Theory Bound	$ \mathbf{K} \log \mathbf{V} $	$d^{-1} \mathbf{K} \log \mathbf{V} $	None	None
Naïve	$d \mathbf{K} $	$ \mathbf{K} $	General Position	$\rho(\mathbf{V}) > 0$
Nichani et al. (2024)	$ \mathbf{K} \log^{12} \mathbf{V} $	$d^{-1} \mathbf{K} \log^{12} \mathbf{V} $	Uniform on S^{d-1}	Uniform on S^{d-1}
Ours	$[\rho(\mathbf{V})]^{-2} \mathbf{K} \log \mathbf{V} $	$d^{-1} [\rho(\mathbf{V})]^{-2} \mathbf{K} \log \mathbf{V} $	General Position	$\rho(\mathbf{V}) > 0$

distributions (e.g., normal, spherical, etc.). This gives $O(|\mathbf{K}| \log |\mathbf{V}|)$ parameters¹⁰ for the full encoder-decoder MLP.

Construction 4.2 (Decoder Construction). *Sample an i.i.d. random Gaussian matrix $\mathbf{D} \in \mathbb{R}^{d \times m}$. Then, define $\mathbf{c}_i = \mathbf{D}^\top \mathbf{u}_i^*(\mathbf{V})$. For $m = O([\rho(\mathbf{V})]^{-2} \log |\mathbf{V}|)$, Equation 7 holds with probability $> 2/3$. Thus, $\text{dec}(\mathbf{x}) = \mathbf{D}\mathbf{x}$ is a valid decoder construction with probability greater than $2/3$.*

Proof Sketch. $\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{D}\mathbf{c}_i \rangle = \langle \mathbf{D}^\top (\mathbf{v}_i - \mathbf{v}_j), \mathbf{D}^\top \mathbf{u}_i^* \rangle$. By Johnson-Lindenstrauss (Johnson et al., 1984), for $m = \Omega([\rho(\mathbf{V})]^{-2} \ln |\mathbf{V}|)$ and for all $i, j \in [|\mathbf{V}|]$,

$$\text{sign}(\langle \mathbf{D}^\top (\mathbf{v}_i - \mathbf{v}_j), \mathbf{D}^\top \mathbf{u}_i^* \rangle) = \text{sign}(\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i^* \rangle)$$

with probability $> 2/3$. See Theorem B.5.3 for a full proof. $\square$

The decodability $\rho(\mathbf{V})$ (Equation (4)) quantifies how large m needs to be as a function of how tightly clustered the value embeddings are. Notably, our construction applies to all feasible embeddings ($\rho(\mathbf{V}) > 0$).

4.3 Full MLP Construction

Finally, we put the encoder and decoder together and describe our full fact MLP construction.

Theorem 4.3.1 (Full Construction). *For any fact set f , generic key embeddings $\mathbf{K}$, and value embeddings $\mathbf{V}$ with $\rho(\mathbf{V}) > 0$, construct enc as described in Section 4.1 and construct dec as described in Section 4.2. Our constructed fact MLP*

$$\mathbf{g}(\mathbf{x}) = \text{dec}(\text{enc}(\mathbf{x})) = \mathbf{D}\mathbf{E}(\sigma(\mathbf{G}\mathbf{x}) \odot (\mathbf{A}\mathbf{x}))$$

stores f given $\mathbf{K}$ and $\mathbf{V}$. Our constructed fact MLP has fact-storage cost $\Theta([\rho(\mathbf{V})]^{-2} |\mathbf{K}| \log |\mathbf{V}|)$.

We compare our construction to other fact-storing MLP constructions in Table 1. For value embeddings with $\rho(\mathbf{V}) = \Omega(1)$, our construction is the first to match the asymptotic parameter count predicted by the information-theory lower bound (Theorem 2.1.1) and requires a $\log^{11} |\mathbf{V}|$ factor fewer parameters than Nichani et al. (2024). Additionally, in the case of two-hot key and value embeddings (using Construction 4.1 for the encoder), our construction matches the information-theory lower bound (Theorem 2.1.1) in terms of bits.

¹⁰We describe this in detail in Appendix B.8.

4.4 Constructed and GD fact MLPs Empirical Scaling

In Figure 2 we show the fact-storage cost of our constructed MLPs, the constructed MLPs from Nichani et al. (2024) (NTK MLPs), and MLPs trained with gradient descent (GD MLPs) across a range of embeddings.

In Figure 2a, we demonstrate that our constructed MLP fact-storage cost scales inversely with ρ at a rate matching the prediction from Construction B.6.1.

In Figure 2b, we show that for embeddings sampled from an i.i.d. uniform spherical distribution (*spherical embeddings*), our MLPs empirically match the asymptotic fact-storage cost of GD MLPs unlike NTK MLPs.

Additionally, we ablate the effect of using gradient descent for the encoder and decoder of our construction: replacing our encoder construction with a gradient-descent-trained encoder ($GD + JL$) increases our construction fact-storage capacity by $\sim 3\times$, replacing our decoder construction with a gradient-descent-trained decoder ($Bin + GD$) increases our construction fact-storage capacity by $\sim 4\times$, and replacing both our encoder and decoder constructions with gradient-descent-trained counterparts ($GD + GD$) increases our construction fact-storage capacity by $\sim 8\times$.

In Figure 2c, we show the fact-storage cost on spherical embeddings for $d \in \{32, 64, 128\}$ and variable $F = |\mathbf{K}| = |\mathbf{V}|$, specifically by setting $F = \alpha d^2$ for various α . We see that like GD MLPs, our construction exhibits the same scaling regardless of the choice of d . On the other hand, for each choice of d , NTK MLPs diverge for sufficiently large α and F , indicating that NTK MLPs do not mimic the ability of fact MLPs to store large fact sets with small input-output dimension.

5 Integrating fact-storing MLPs into Transformers

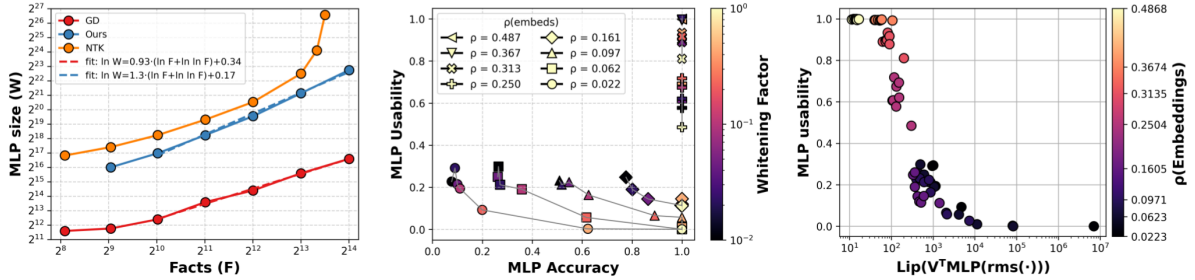


Figure 3: (a) **MLP size vs. fact-set size for MLPs with $\geq 99\%$ usability within Transformer.** We find that fact-storing MLPs are usable within 1-layer Transformers and that our constructed MLPs and GD MLPs exhibit similar $\geq 99\%$ usability scaling. (b) **MLP usability within Transformer v.s. MLP storage capacity.** We observe a tradeoff between MLP usability within a Transformer and the MLP’s fact-storage capacity. (c) **MLP usability within Transformer v.s. its Lipschitz constant.** We observe that the measured Lipschitz constant is predictive of an MLP’s usability within Transformers.

We now investigate the extent to which fact-storing MLPs can be used by a transformer for factual recall. In Section 5.1, we introduce the *Synthetic Sequential Factual Recall* (SSFR) task, which formalizes the notion of transformer factual recall. We then find a small set of architectural modifications that enable vanilla transformers to use constructed MLPs for factual recall. Under this setup, we show that the number of MLP parameters required for a transformer to properly use the for factual recall grows at a comparable rate to the information-theoretically optimal one.

In Section 5.2, we uncover a tradeoff between the capacity of an MLP to store facts and its usability for factual recall within transformers. We demonstrate that this tradeoff can be navigated through embedding whitening. In Section 5.3, we further show that an MLP’s Lipschitz constant serves as an indicator of its usability for factual recall by transformers.

Finally, in Section 5.4, we explore using fact-storing MLPs within 1-layer transformers on a synthetic language-modeling (LM) task. We find that fact-storing MLPs within transformers can be swapped by MLPs storing entirely different fact sets, incurring only a $\sim 3\%$ cross-entropy increase on non-fact tokens while enabling the transformer to produce the new facts. Moreover, our MLP-swapping method outperforms prior fact-editing MLP updates, doubling their fact-editing score when editing 10% of the fact set.

5.1 Transformers can use fact-storing MLPs for factual recall

We first demonstrate that fact-storing MLPs can be used for factual recall within a transformer. Further, we show that, together with GD MLPs, our construction is the first to be usable within a transformer while storing an amount of facts per parameter comparable to the information-theory optimal one.

Task. We introduce an associative-recall-style task (Arora et al., 2023; Nichani et al., 2024), which we term *Synthetic Sequential Factual Recall* (SSFR), to test whether fact-storing MLPs can be used by transformers for factual recall. In SSFR, a transformer processes a sequence of “junk” tokens containing a single *key* token and must predict the corresponding *value* token at the end of the sequence. For example,

$$\underbrace{* \% \& \# \$}_{\text{junk prefix}} \underbrace{A}_{\text{key}} \underbrace{\& \% * \$ \#}_{\text{junk suffix}} \rightarrow \underbrace{B}_{\text{value}}.$$

This mirrors how, in a sentence such as “The capital of France is Paris,” the key and value (“capital of France” and “Paris”) are separated by an unrelated prefix and suffix (“The” and “is”). See Appendix A.2.1 for details.

Training setup. Our goal is to evaluate to what extent fact-storing MLPs can be used by transformers on an SSFR task. To test this, we create a fact-storing MLP that stores the SSFR key-value mapping. We then freeze the fact-storing MLP and insert it into a single-layer transformer. Finally, we train the transformer to output the correct value for each SSFR sequence.

Metrics. To evaluate whether a transformer is actually using its fact-storing MLP for factual recall, as opposed to memorizing the facts in its attention weights, we define the *fact-adaptive accuracy*. We take a transformer trained on SSFR and replace its fact-storing MLP with a new MLP storing a different fact set. We define the transformer’s *fact-adaptive accuracy* as the modified transformer’s accuracy on the SSFR task corresponding to the fact set *of the new MLP*. Intuitively, if a transformer has high fact-adaptive accuracy, it is using its fact-storing MLP for factual recall.

Fact-Storing MLPs are usable within transformers. We find that a simple set of modifications to the vanilla transformer architecture are sufficient for transformers to use both constructed and GD-trained MLPs for factual recall, achieving $> 99\%$ fact-adaptive accuracy, while approximately using an information-theoretically optimal amount of parameters. Figure 3a shows the minimum fact-storing MLP parameters required for a transformer using it to reach 99% fact-adaptive accuracy as a function of fact-set size. Strikingly, our constructed and GD MLPs both exhibit empirical scaling similar to the theoretical optimum $\log W \approx \log F + \log \log F$, in contrast to NTK MLPs, whose fact-adaptive accuracy explodes for large fact sets. We attribute such a deterioration in fact-adaptive accuracy of NTK MLPs to their sharp decline in fact-storage capacity on large fact sets, as shown in Figure 2b. See Appendix A.2.3 for experimental details.

Concretely, we empirically find that i) tying transformer and MLP embeddings, ii) removing residual connections, iii) freezing the pre-MLP RMSNorm layer, and iv) freezing the *value* and *out-project* matrices of the attention layer to the identity matrix are sufficient for transformers to use fact-storing MLPs for factual recall.

Further, as observed in Figure 7, we find that the minimum MLP size needed to achieve $> 99\%$ fact-adaptive accuracy for GD gated and non-gated MLPs is almost identical, suggesting that fact-storage within a transformer doesn’t depend on the specific MLP architecture, but instead on its number of parameters.

5.2 Tradeoff Between Capacity and Usability of an MLP

We uncover a tradeoff between an fact-storing MLP’s *storage capacity*, the fraction of facts of a fact set that it can successfully store, and *usability*, the fraction of those stored facts that a transformer using the fact-storing can correctly retrieve, as can be seen in Figure 3b and Figure 8a. Formally, we define:

$$\begin{aligned} \text{capacity} &= \frac{\# \text{ facts MLP stores}}{\text{total } \# \text{ facts}} \\ \text{usability} &= \frac{\text{transformer fact-adaptive accuracy}}{\text{capacity}}. \end{aligned}$$

To study this capacity-usability tradeoff, we use our *embedding whitening* technique from Section 3 but vary the strength $\alpha \in [0, 1]$ of the empirical covariance whitening transform $T(\mathbf{x}) = \mathbf{M}^\alpha \mathbf{x} + \mathbf{b}$. For a fixed pair of transformer key and value embeddings, characterized by $\rho(\mathbf{K}) = \rho(\mathbf{V})$, we apply different whitening strengths α , train an MLP to store a fact set using the corresponding MLP embeddings, and then train a Transformer to use that whitened MLP in SSFR.

We find that adjusting the whitening degree allows us to explore the tradeoff between usability and capacity. MLPs trained on less-whitened embeddings store fewer facts but are more usable by transformers, whereas MLPs trained on highly whitened embeddings store more facts but are harder for transformers to use. See Appendix A.2.4 for experimental details.

5.3 MLP Usability Depends on Lipschitz Constant

In Section 5.2 we observe that whitened MLPs, with high fact storage capacity, tend to be less usable by transformers. Here, we find that the Lipschitz constant of an MLP serves as an indicator of its usability within a transformer. Concretely, given an MLP trained to represent a fact-set mapping from transformer key embeddings $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$ to value embeddings $\mathbf{V} \in \mathbb{R}^{|\mathbf{V}| \times d}$, we look at:

$$\text{Lip}(\mathbf{V}^T \text{MLP}(\text{rms}(\cdot))) \approx \max_i \sigma_1(\mathbf{J}(\mathbf{k}_i)), \quad (8)$$

where

$$\mathbf{J}(\mathbf{x}_i) = \frac{\partial \mathbf{V}^T \text{MLP}(\text{RMSNorm}(\mathbf{x}_i))}{\partial \mathbf{x}_i}.$$

As seen in Figure 3c and Figure 8a, increased MLP Lipschitz constant correlates with reduced MLP usability for factual recall. Intuitively, we believe this relationship arises due to optimization dynamics, similar to how training convergence under first-order optimizers depends on the largest Hessian singular value (Mohammadi et al., 2022). We note there likely exist other MLP conditioning related metrics that can also capture this relationship. See Appendix A.2.5 for experimental details.

5.4 Language Modeling and Fact Editing with fact-storing MLPs

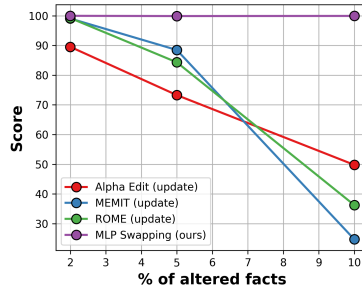


Figure 4: **Fact editing score as number of altered facts increases.** Fact editing via MLP swapping outperforms prior weight updates as the number of altered facts increase. The fact-editing score is computed as the geometric mean of the efficacy, specificity and paraphrase accuracies.

Finally, we explore whether fact-storing MLPs can be used by transformers for language modeling. On a synthetic task involving sentences about author-book relations (see Appendix A.3.1), we demonstrate that 1-layer transformers can use fact-storing MLPs for factual recall (Figure 9a). Remarkably, when we swap a transformer’s MLP for an entirely new fact-storing MLP, the transformer outputs the new facts with $> 99\%$ accuracy while incurring less than a $\sim 3\%$ increase in cross-entropy on non-fact tokens (Appendix 9b). See Appendix A.3.3 for experimental details.

Under the same setup, we show that transformers equipped with fact-storing MLPs can be modularly fact-edited. As shown in Figure 4, our modular fact-editing procedure (MLP Swapping) consistently outperforms prior fact editing updates, including those of MEMIT (Meng et al., 2023c), ROME (Meng et al., 2023b), and Alpha Edit (Fang et al., 2025), doubling their fact-editing scores (defined in Figure 4) on our 1-layer transformers when editing as little as 10% of the facts stored in its MLP (see Appendix A.3.4). These results suggest a path toward more robust and modular fact manipulation in LLMs.

6 Discussion

We have presented a construction that produces fact-storing MLPs with asymptotically fewer parameters than prior approaches, supports a broader class of embeddings, and can be used by transformers for factual recall. Using this construction, we characterized how output geometry affects fact-storage capacity, identified a simple encoder–decoder mechanism that matches information-theoretic facts-per-parameter scaling, and uncovered a capacity–usability tradeoff for fact-storing MLPs within transformers. These results offer a coherent framework for understanding how MLPs store and expose knowledge within transformers.

More broadly, our work outlines a constructive path forward for studying LLMs. Rather than relying solely on descriptive analyses of pretrained models, we show that explicitly building MLPs with interpretable, provable mechanisms can reveal principles that are otherwise difficult to extract from their learned weights. This constructive approach suggests several promising directions such as designing modular and robust memory systems, developing more parameter-efficient training and inference pipelines, and exploring whether similar constructions can shed light over LLM behaviors beyond factual recall.

In summary, by directly constructing MLPs that store and expose facts, we provide both a theoretical foundation and practical tools for understanding knowledge storage in transformers, as well as a path toward more interpretable and controllable mechanisms in large language models.

Acknowledgements

The authors thank Neel Guha, Yasa Baig, Catherine Deng, Kelly Buchanan, Sam Buchanan, Avanika Narayan, Andy Dimnaku, Mayee Chen, Hermann Kumbong, Francois Chaubard, Jon Saad-Falcon, Stuart Sul, Alex Waitz, Dan Biderman, Ben Spector, Simran Arora and Michael Zhang for their helpful feedback and discussion.

The authors gratefully acknowledge the support of NIH under No. U54EB020405 (Mobilize), NSF under Nos. CCF2247015 (Hardware-Aware), CCF1763315 (Beyond Sparsity), CCF1563078 (Volume to Velocity), and 1937301 (RTML); US DEVCOM ARL under Nos. W911NF-23-2-0184 (Long-context) and W911NF-21-2-0251 (Interactive Human-AI Teaming); ONR under Nos. N000142312633 (Deep Signal Processing); Stanford HAI under No. 247183; NXP, Xilinx, LETI-CEA, Intel, IBM, Microsoft, NEC, Toshiba, TSMC, ARM, Hitachi, BASF, Accenture, Ericsson, Qualcomm, Analog Devices, Google Cloud, Salesforce, Total, the HAI-GCP Cloud Credits for Research program, the Stanford Data Science Initiative (SDSI), and members of the Stanford DAWN project: Meta, Google, and VMWare. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, policies, or endorsements, either expressed or implied, of NIH, ONR, or the U.S. Government. OD is supported by the Hertz Foundation Fellowship, the Stanford Knight-Hennessy Scholarship, and the NSF GRFP. JL is supported by the Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0023112. AR’s research is supported by NSF grant CCF#2247014.

References

- Allen-Zhu, Z. and Li, Y. Physics of language models: Part 3.3, knowledge capacity scaling laws, 2024. URL <https://arxiv.org/abs/2404.05405>.
- Arora, S., Eyuboglu, S., Timalsina, A., Johnson, I., Poli, M., Zou, J., Rudra, A., and Ré, C. Zoology: Measuring and improving recall in efficient language models, 2023. URL <https://arxiv.org/abs/2312.04927>.
- Aubrun, G., Jenkinson, J., and Szarek, S. J. Optimal constants in concentration inequalities on the sphere and in the gauss space, 2024. URL <https://arxiv.org/abs/2406.13581>.
- Bubeck, S., Eldan, R., Lee, Y. T., and Mikulincer, D. Network size and weights size for memorization with two-layers neural networks, 2020. URL <https://arxiv.org/abs/2006.02855>.
- Chafaï, D., Guédon, O., Lecué, G., and Pajor, A. *Interactions between compressed sensing random matrices and high dimensional geometry*, volume 37, chapter 1, pp. 26–27. Société Mathématique de France Paris, 2012.

- Chirka, E. M. Complex analytic sets, 1997. URL https://doi.org/10.1007/978-3-642-61525-2_3.
- Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. Knowledge neurons in pretrained transformers, 2022. URL <https://arxiv.org/abs/2104.08696>.
- Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv-2407, 2024.
- Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy models of superposition, 2022. URL <https://arxiv.org/abs/2209.10652>.
- Fang, J., Jiang, H., Wang, K., Ma, Y., Jie, S., Wang, X., He, X., and seng Chua, T. Alphaedit: Null-space constrained knowledge editing for language models, 2025. URL <https://arxiv.org/abs/2410.02355>.
- Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories, 2021. URL <https://arxiv.org/abs/2012.14913>.
- Geva, M., Caciularu, A., Wang, K. R., and Goldberg, Y. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space, 2022. URL <https://arxiv.org/abs/2203.14680>.
- Google DeepMind. Ai achieves silver-medal standard solving international mathematical olympiad problems with alphaproof. Blog post, Jul 2024. URL <https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/>. “AlphaProof is a system that trains itself to prove mathematical statements in the formal language Lean ...”, solves 3 IMO problems etc.
- Gu, J.-C., Xu, H.-X., Ma, J.-Y., Lu, P., Ling, Z.-H., Chang, K.-W., and Peng, N. Model editing harms general abilities of large language models: Regularization to the rescue, 2024. URL <https://arxiv.org/abs/2401.04700>.
- Guha, N., Nyarko, J., Ho, D., Ré, C., Chilton, A., Chohlas-Wood, A., Peters, A., Waldon, B., Rockmore, D., Zambrano, D., et al. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36:44123–44279, 2023.
- Gupta, A., Rao, A., and Anumanchipalli, G. Model editing at scale leads to gradual and catastrophic forgetting, 2024. URL <https://arxiv.org/abs/2401.07453>.
- Johnson, W. B., Lindenstrauss, J., et al. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.
- Kalavasis, A., Karbasi, A., Larsen, K. G., Velegkas, G., and Zhou, F. Replicable learning of large-margin halfspaces, 2024. URL <https://arxiv.org/abs/2402.13857>.
- Karpathy, A. NanoGPT. <https://github.com/karpathy/nanoGPT>, 2022.
- Kotelenez, P., Leitman, M. J., and Jr, J. A. M. On the depletion effect in colloids: Correlated brownian motions, 2008. URL <https://arxiv.org/abs/0807.5054>.
- Leskelä, L. and Zhukov, M. Sharp constants relating the sub-gaussian norm and the sub-gaussian parameter, 2025. URL <https://arxiv.org/abs/2507.05928>.
- Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt, 2023a. URL <https://arxiv.org/abs/2202.05262>.
- Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt, 2023b. URL <https://arxiv.org/abs/2202.05262>.
- Meng, K., Sharma, A. S., Andonian, A., Belinkov, Y., and Bau, D. Mass-editing memory in a transformer, 2023c. URL <https://arxiv.org/abs/2210.07229>.
- Mohammadi, H., Razaviyayn, M., and Jovanović, M. R. Tradeoffs between convergence rate and noise amplification for momentum-based accelerated optimization algorithms. *arXiv e-prints*, art. arXiv:2209.11920, September 2022. doi: 10.48550/arXiv.2209.11920.

- Morris, J. X., Sitawarin, C., Guo, C., Kokhlikyan, N., Suh, G. E., Rush, A. M., Chaudhuri, K., and Mahlouljifar, S. How much do language models memorize?, 2025. URL <https://arxiv.org/abs/2505.24832>.
- Nanda, N., Rajamanoharan, S., Kramár, J., and Shah, R. Fact finding: Attempting to reverse-engineer factual recall on the neuron level, Dec 2023. URL <https://www.alignmentforum.org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall>.
- Nichani, E., Lee, J. D., and Bietti, A. Understanding factual recall in transformers via associative memories, 2024. URL <https://arxiv.org/abs/2412.06538>.
- Oxley, J. *Matroid Theory*. Number 21 in Oxford Graduate Texts in Mathematics. Oxford University Press, Oxford, 2 edition, 2011.
- Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Saab, K., Tu, T., Weng, W.-H., Tanno, R., Stutz, D., Wulczyn, E., Zhang, F., Strother, T., Park, C., Vedadi, E., et al. Capabilities of gemini models in medicine. *arXiv preprint arXiv:2404.18416*, 2024.
- Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding, 2023. URL <https://arxiv.org/abs/2104.09864>.
- Sun, W., Qu, T., Li, M., Davis, J., and Moens, M.-F. Mitigating negative interference in multilingual sequential knowledge editing through null-space constraints, 2025. URL <https://arxiv.org/abs/2506.10800>.
- Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47, chapter 3, pp. 67–68. Cambridge university press, 2018.
- Vrahatis, M. N. Towards the mathematical foundation of the minimum enclosing ball and related problems. *arXiv e-prints*, art. arXiv:2402.06629, January 2024. doi: 10.48550/arXiv.2402.06629.
- Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48, chapter 2, pp. 24–25. Cambridge university press, 2019.
- Wan, M. and McAuley, J. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys ’18, pp. 86–94, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359016. doi: 10.1145/3240323.3240369. URL <https://doi.org/10.1145/3240323.3240369>.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025a. URL <https://arxiv.org/abs/2505.09388>.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025b.
- Zucchet, N., Bornschein, J., Chan, S., Lampinen, A., Pascanu, R., and De, S. How do language models learn facts? dynamics, curricula and hallucinations, 2025. URL <https://arxiv.org/abs/2503.21676>.

A Experiments

A.1 MLP Experiments

Here we describe the experimental setup used for the MLP fact-storage capacity results in Sections 3 and 4.

A.1.1 Task Setup

Fact sets. Following the definition of the synthetic fact-storage task (Equation (1)), we index facts by $i \in [F]$. Although fact-storage cost is defined as the smallest number of parameters needed to represent *all possible fact sets* (Equation (2)), in our experiments we approximate fact-storage cost as the smallest number of parameters needed to represent *randomly sampled* bijective key-value maps $f : [F] \rightarrow [F]$.

Facts vs. embedding dimension. In our experiments, for each embedding dimension d_{model} , we set the number of facts to $F = \beta d_{\text{model}}^2$, where the multiplier $\beta = 0.25$ unless otherwise specified.

Empirically, we find that the choice of β does not affect the fact-storage capacity of gradient-descent-trained MLPs or our constructed MLPs. However, interestingly, larger values of β significantly *decrease* the fact-storage capacity of the MLP construction of Nichani et al. (2024): see Section A.1.4.

Embeddings. Following prior work (Nichani et al., 2024), key and value embeddings $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{F \times d}$ are uniformly sampled from the unit sphere. Mirroring how word embeddings in LLMs work, our experiments *tie keys and values*, i.e. $\mathbf{K} = \mathbf{V}$.

Anisotropic value embeddings. To vary the condition number of the value embeddings while preserving their geometric structure, we modify only the singular values of the embeddings matrix. We keep the left and right singular vectors fixed and apply a log-affine rescaling to the singular values so that the largest one is preserved and the smallest one is set to achieve a desired condition number κ .

Approximating MLP fact-storage cost via binary search. For each choice of $(d, F, \kappa, \text{MLP family})$, we determine the minimum number of parameters needed to perfectly store a randomly-sampled fact set given randomly-sampled embeddings. To do so, we perform a one-dimensional binary search over a single scalar hyperparameter characterizing the “size” of the MLP. The hyperparameter we sweep over depends on the family of MLPs we evaluate:

- For gradient-descent-trained (GD) and NTK MLPs (Nichani et al., 2024), we search over the hidden dimension h .
- For our constructed MLPs, we either search over the decoder dimension m or the *encoder width multiplier*.

See Section A.1.3 for details about each of the MLP variants we evaluate.

A.1.2 Metrics

Accuracy-based success criterion. We evaluate models using the same dot-product scoring rule used in the definition of fact storage (Equation (1)), which we restate here for convenience. Given a trained model $\mathbf{g}_\theta$ and embeddings $(\mathbf{K}, \mathbf{V})$, the predicted value index for a key $i \in [F]$ is

$$\hat{f}(i) = \arg \max_{j \in [F]} \langle \mathbf{g}_\theta(\mathbf{k}_i), \mathbf{v}_j \rangle,$$

i.e. the index achieving the highest score with respect to the MLP output.

The fact-storage *accuracy* of $\mathbf{g}_\theta$ on a fact set $f : [F] \rightarrow [F]$ is then

$$\text{Acc} = \frac{1}{F} \sum_{i \in [F]} \mathbf{1}[\hat{f}(i) = f(i)].$$

Within our binary searches, we declare that a model successfully stores a fact set if it achieves an accuracy of at least $1 - \varepsilon_{\text{acc}}$. For our MLP fact-storage capacity experiments, we set $\varepsilon_{\text{acc}} = 0$ unless otherwise stated.

When multiple random seeds are used for a given binary search experiment (e.g. where the randomness is over the choice of fact set and embeddings), we aggregate by taking the *minimum accuracy across seeds* before comparing to this threshold. The binary search then returns the smallest number of parameters for which the aggregated accuracy is at least $1 - \varepsilon_{\text{acc}}$.

A.1.3 MLP architectures and variants

Here we summarize all MLP variants evaluated in the capacity sweeps, corresponding to the methods compared in Figure 2c and described formally in Section 4. Each configuration consists of (i) a choice of MLP variant (gradient-descent-trained, our explicit construction, or the NTK construction of Nichani et al. (2024)), (ii) variant-specific configuration details, including optional use of margin-optimal outputs for NTK MLPs and encoder-decoder settings for our construction models, and (iii) optional embedding whitening.

We start by describing each MLP variant and variant-specific configuration details:

- **Gradient-descent-trained (GD) MLPs.** GD MLPs use the standard two-layer gated MLP (SwiGLU-style) architecture described in Section 4.1, with an “up” projection $\mathbb{R}^d \rightarrow \mathbb{R}^h$ followed by a “down” projection $\mathbb{R}^h \rightarrow \mathbb{R}^d$. Given an input $\mathbf{x} \in \mathbb{R}^d$, the block computes

$$\mathbf{g}_\theta(\mathbf{x}) = W_{\text{down}}(\sigma(W_{\text{gate}}\mathbf{x} + \mathbf{b}_{\text{gate}}) \odot (W_{\text{up}}\mathbf{x} + \mathbf{b}_{\text{up}})) + \mathbf{b}_{\text{down}},$$

where $W_{\text{up}}, W_{\text{gate}} \in \mathbb{R}^{h \times d}$, $W_{\text{down}} \in \mathbb{R}^{d \times h}$, σ is Swish, and $\odot$ denotes element-wise multiplication.

Models are trained with full-batch gradient descent using Adam and a cosine-annealed learning rate schedule (initial rate 10^{-3} , final rate 10^{-6}) for up to 20,000 epochs with early stopping. We use the cross-entropy objective formed from dot-product logits $\mathbf{g}_\theta(\mathbf{K})\mathbf{V}^\top$, matching the decoding rule of Equation (1).

In the sweeps, the hidden dimension h is the sole capacity parameter, which means binary search identifies the smallest h for which the trained GD MLP achieves perfect fact-storage accuracy.

- **Our constructed MLPs.** Our construction decomposes the fact-storing MLP into an *encoder* and a *decoder*, each of which admits both an explicit construction and a learnable gradient-descent-based alternative. For completeness, we summarize all variants evaluated in the sweeps.

Encoder variants.

- **Binning / explicit (Bin) encoder.** This is the encoder defined in Section 4.1 and Algorithm 2, built by stacking m closed-form encoder gadgets (Algorithm 3). Each gadget solves a linear system to map keys to the j th coordinate of the compressed code $\mathbf{C}$; the full encoder has the gated form

$$\mathbf{enc}(\mathbf{x}) = \mathbf{E}(\sigma(\mathbf{G}\mathbf{x}) \odot (\mathbf{A}\mathbf{x})).$$

This encoder is fully explicit and requires no training.

- **Gradient-descent-trained (GD) encoder.** Instead of constructing $(\mathbf{A}, \mathbf{G}, \mathbf{E})$ analytically, we train a gated encoder $g_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^m$ via full-batch gradient descent to fit the compressed codes $\mathbf{C}$. Given keys $\mathbf{K}$ and targets $\mathbf{C}$ permuted by f , we minimize

$$\mathcal{L}_{\text{enc}} = \frac{1}{F} \sum_{i=1}^F \|g_\theta(\mathbf{k}_i + \eta_i) - \mathbf{c}_{f(i)}\|_2^2, \quad \eta_i \sim \mathcal{N}(0, \varepsilon_{\text{key}}^2 I_d),$$

with $\varepsilon_{\text{key}} = 10^{-7}$. The encoder uses the same gated MLP architecture as the explicit encoder, but with hidden dimension

$$h = \lceil m(F/d) \cdot \text{encoder_width_multiplier} \rceil$$

(where the encoder width multiplier = 1 by default), and is trained for 1000 Adam updates with learning rate 10^{-2} . After training, g_θ is used as the encoder and produces the hidden codes used by the decoder.

Decoder variants.

- **Johnson-Lindenstrauss (JL) decoder.** This is the explicit decoder of Section 4.2 and Algorithm 4. We sample a Gaussian matrix $\mathbf{D} \in \mathbb{R}^{d \times m}$ and set compressed codes $\mathbf{C} = \mathbf{U}^* \mathbf{D}$, where $\mathbf{U}^*$ is the *margin-optimal output embeddings* (Theorem 3.2.1). For $m = \Theta(\rho(\mathbf{V})^{-2} \log |\mathbf{V}|)$, the JL decoder satisfies the decoding inequalities with high probability.

- **Gradient-descent-trained (GD) decoder.** We replace the random projection with learnable compressed codes $\mathbf{C} \in \mathbb{R}^{F \times m}$ and a learnable decoding matrix $\mathbf{M} \in \mathbb{R}^{m \times d}$. Predicted values are $\hat{\mathbf{V}} = \mathbf{C}\mathbf{M}$ with dot-product scores $S = \hat{\mathbf{V}}\mathbf{V}^\top$. We train $(\mathbf{C}, \mathbf{M})$ using full-batch Adam (with a learning rate of 1, cosine decay to 0.01, and 1000 steps) with cross-entropy loss over the scores:

$$\mathcal{L}_{\text{dec}} = \text{CE}(S, f).$$

After training, we normalize the rows of $\mathbf{C}$ and $\mathbf{M}$ for numerical stability, and $(\mathbf{C}, \mathbf{M})$ replaces the analytic JL decoder in the full construction.

Each constructed MLP is uniquely identified by its encoder/decoder pair (Bin+JL, GD+JL, Bin+GD, GD+GD).

In the sweeps, the decoder width m_{dec} is the capacity parameter for the Bin+JL and Bin+GD construction variants. For the GD+JL and GD+GD variants, we use a two-step procedure. First, we sweep over the decoder width m , obtaining the smallest value $\hat{m}$ for which the constructed MLP achieves perfect fact-storage accuracy. Next, we fix $m = \hat{m}$ and further sweep over the *encoder width multiplier* to find the smallest value in the range $[0, 2]$ for which the MLP achieves perfect accuracy.

- **NTK MLPs.** We also evaluate the Hermite-feature construction of Nichani et al. (2024), which we refer to throughout as “NTK MLPs”.

Given key embeddings $\mathbf{K} \in \mathbb{R}^{F \times d}$, value embeddings $\mathbf{V} \in \mathbb{R}^{F \times d}$, and a mapping $f : [F] \rightarrow [F]$, the NTK MLP of width h is constructed as in Algorithm 5.

- We first (optionally) replace $\mathbf{V}$ by the *minimum-margin output embeddings* $\mathbf{U}^*$: in our ablations, we find this improves fact-storage capacity by 2-4 $\times$ (Figure 5).
- We then apply the construction from Nichani et al. (2024). Crucially, although Nichani et al. (2024)’s Theorem 2 describes a *non-gated* MLP construction, in fact their work first defines a *gated* MLP, then uses an NTK argument to show that a non-gated MLP can be used to approximate the gated MLP by rescaling the magnitudes of the MLP weights. In our experiments, we find the non-gated MLP exhibits large Lipschitz constant, making it impractical to use within a Transformer; as such, we directly implement their gated MLP without the NTK approximation.

The resulting gated MLP has the form

$$\mathbf{g}_{\text{NTK}}(\mathbf{x}) = \mathbf{P}\left(\sigma(\mathbf{W}_{\text{gate}}\mathbf{x}) \odot (\mathbf{W}_{\text{up}}\mathbf{x})\right),$$

with σ equal to the chosen activation. In our experiments, mirroring the GD and our constructed MLPs, we use $\sigma = \text{Swish}$.

In the sweeps, the hidden dimension h is the sole capacity parameter for NTK MLPs, and we perform binary search over h exactly as for GD MLPs.

Note that Nichani et al. (2024) proposes their construction for uniformly spherically distributed key and value embeddings that are *not tied*; in our experiments, we evaluate how well the NTK MLP construction can generalize to more realistic settings, such as tied + anisotropic embeddings.

Computing margin-optimal output embeddings. For both our constructed MLPs and the NTK baseline, we optionally replace the original value embeddings $\mathbf{V} \in \mathbb{R}^{F \times d}$ by a new set $\mathbf{U}^*$ obtained by maximizing the dot-product decoding margin (as in Theorem 3.2.1). Specifically, for each i we solve the convex optimization problem

$$\max_{\|u\|_2 \leq 1} \min_{j \neq i} \frac{\langle \mathbf{v}_i - \mathbf{v}_j, u \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\|_2},$$

and denote the optimizer by u_i^* . We solve these problems using ADMM.

Algorithm 5 NTK MLP Construction

Require: Keys $\mathbf{K} \in \mathbb{R}^{F \times d}$, values $\mathbf{V} \in \mathbb{R}^{F \times d}$, mapping $f : [F] \rightarrow [F]$

Require: Hidden width h , activation choice σ , Hermite degree k , finite-difference step ε (for plain MLP)

Require: Flag `margin_optimal` (whether to use $\mathbf{U}^*$)

- 1: **if** `margin_optimal` is True **then**
- 2: $\mathbf{V} \leftarrow \mathbf{U}^*$ {margin-optimal output embeddings}
- 3: **end if**
- 4: Sample gate weights $\mathbf{W}_{\text{gate}} \sim \mathcal{N}(0, 1)^{h \times d}$
- 5: Sample $\mathbf{P}_{\text{raw}} \sim \mathcal{N}(0, 1)^{d \times h}$ and normalize each column to unit norm to obtain $\mathbf{P}$
- 6: $\mathbf{Z} \leftarrow \mathbf{K} \mathbf{W}_{\text{gate}}^\top \in \mathbb{R}^{F \times h}$ {project inputs}
- 7: Choose Hermite degree k (from activation or configuration)
- 8: $\mathbf{H} \leftarrow \hat{\mathbf{H}}_k(\mathbf{Z}) \in \mathbb{R}^{F \times h}$ {degree- k normalized Hermite features}
- 9: $\mathbf{Y} \leftarrow [\mathbf{V}_{f(0)}; \dots; \mathbf{V}_{f(F-1)}] \in \mathbb{R}^{F \times d}$ {reorder values by f }
- 10: $\mathbf{A} \leftarrow \mathbf{Y} \mathbf{P} \in \mathbb{R}^{F \times h}$ {feature coefficients}
- 11: $\mathbf{W}_{\text{up}} \leftarrow \frac{1}{h} (\mathbf{H} \odot \mathbf{A})^\top \mathbf{K} \in \mathbb{R}^{h \times d}$

return the gated MLP:

$$\mathbf{g}(\mathbf{x}) = \mathbf{P} \left(\sigma(\mathbf{W}_{\text{gate}} \mathbf{x}) \odot (\mathbf{W}_{\text{up}} \mathbf{x}) \right)$$

Embedding whitening. For anisotropic value embeddings, we optionally apply a ZCA whitening preconditioning step prior to training or construction. Given an embedding matrix $E \in \mathbb{R}^{F \times d}$ (keys or values), we estimate its second-moment matrix

$$\Sigma = \frac{1}{F} E^\top E, \quad \tilde{\Sigma} = \Sigma + \varepsilon I_d$$

with a small ridge $\varepsilon \approx 10^{-6}$ to ensure invertibility. Let $\tilde{\Sigma} = Q \Lambda Q^\top$ be the eigendecomposition, where Q is orthonormal and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$ with $\lambda_i > 0$. Full ZCA whitening corresponds to the transform

$$W_{\text{zca}} = Q \Lambda^{-1/2} Q^\top.$$

We also investigate *interpolating* between no whitening and full whitening using a strength parameter $\alpha \in [0, 1]$:

$$W_\alpha = W_{\text{zca}}^\alpha.$$

Before training or construction, we replace E by the whitened embeddings $E_{\text{white}} = E W_\alpha$. The inverse transform W_α^{-1} is then folded into the final linear block of the resulting MLP, so that the MLP output remains in the original embedding basis.

A.1.4 Ablations

Effect of margin-optimal output embeddings on NTK MLPs. Figure 2 shows that NTK MLPs fail to achieve perfect fact storage once the value embeddings become sufficiently anisotropic. Here, we investigate whether applying the NTK construction to the *margin-optimal output embeddings* $\mathbf{U}^*$ improves its robustness. As shown in Figure 5, although replacing the raw value embeddings by $\mathbf{U}^*$ improves fact-storage capacity by a factor of 2-4 $\times$, the NTK construction still breaks down once the condition number exceeds a moderate threshold. In contrast, both GD MLPs and our constructed MLPs maintain consistent scaling across a broad range of anisotropic embeddings.

Coherence exhibits weak predictive power for fact-storage capacity. Figure 6 compares fact-storage capacity against the coherence of the embedding matrix, a commonly used measure of geometric spread. Unlike our decodability statistic $\rho(\mathbf{V})$, coherence does not strongly correlate with the number of parameters needed to store a fixed number of facts; this is true for both GD MLPs ($R^2 = 0.10$) and our constructed MLPs ($R^2 = 0.44$). This supports our use of ρ , rather than coherence or related spectral heuristics, as a natural predictor of separability for the decoder and, ultimately, of fact-storage capacity.

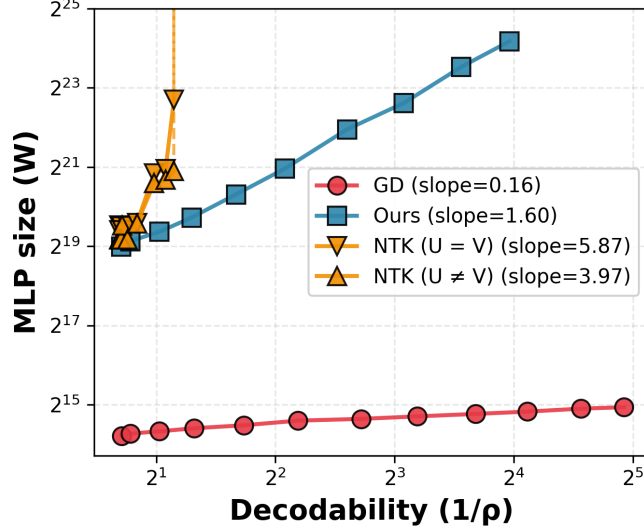


Figure 5: NTK MLPs fail to achieve perfect fact storage for sufficiently anisotropic output embeddings. Using the margin-optimal output embeddings for the NTK construction improves fact-storage capacity by up to 4×, but does not improve robustness to anisotropic embeddings.

A.2 SSFR Experiments

A.2.1 SSFR Task

We introduce the SSFR task to evaluate a model’s ability to retrieve facts stored in its weights. In this task, the model is presented with a sequence containing a single key token surrounded by “junk” tokens and is required to output the corresponding value token according to the task’s *fact set*.

Formally, let $f : \mathcal{S}_k \rightarrow \mathcal{S}_v$ be a fact set over tokens $\mathcal{S}_k \cup \mathcal{S}_v$. Let $\mathcal{J} = \{(j_1^{\text{prefix}}, j_1^{\text{suffix}}), (j_2^{\text{prefix}}, j_2^{\text{suffix}}), \dots\}$ be the set containing junk prefixes and suffixes tuples. The SSFR task is then defined as the set of sequences:

$$\mathcal{S}_{SSFR}[f] = \{\text{concat}(j_{\text{prefix}}, k, j_{\text{suffix}}, f(k)) \mid k \in \mathcal{S}_k, (j^{\text{prefix}}, j^{\text{suffix}}) \in \mathcal{J}\}.$$

The model’s task, given a sequence from $\mathcal{S}_{SSFR}[f]$, is then to predict $f(k)$ as the final token of the sequence. For example, given the sequence

$$\underbrace{* \% \& \# \$}_{\text{junk prefix}} \underbrace{A}_{\text{key}} \underbrace{* \% \& \# \$}_{\text{junk suffix}} \underbrace{B}_{\text{value}}$$

from $\mathcal{S}_{SSFR}[f]$, the model’s task is to predict the final token $B = f(A)$.

In practice, across all of our experiments, the junk prefix and junk suffixes have a length between 8 and 16. Further, the amount of junk prefixes and suffixes tuples we use, i.e. $|\mathcal{J}|$, is 16. Finally, we reserve 16 additional tokens (to those representing the keys and values of the fact-set), as the junk tokens.

A.2.2 Training Setup

The setup we use to train transformers using fact-storing MLPs in all SSFR experiments is as follows:

1. Randomly sample the transformer embeddings for the key, value and junk tokens from a standard normal distribution. We optionally ill-condition the embeddings, as in the MLP fact-storage capacity experiments (Appendix A.1.1). We do not ill-condition embeddings unless stated otherwise.
2. Randomly sample a fact set.
3. Compute the MLP embeddings. To obtain the MLP key embeddings, we just project all the transformer key embeddings to the unit sphere (since the transformer stack forwards them through a normalization layer before feeding them to the MLP). The MLP value embeddings stay the same as the transformer value embeddings.
4. Construct or train with gradient-descent a fact-storing MLP that stores the fact set under the MLP embeddings.

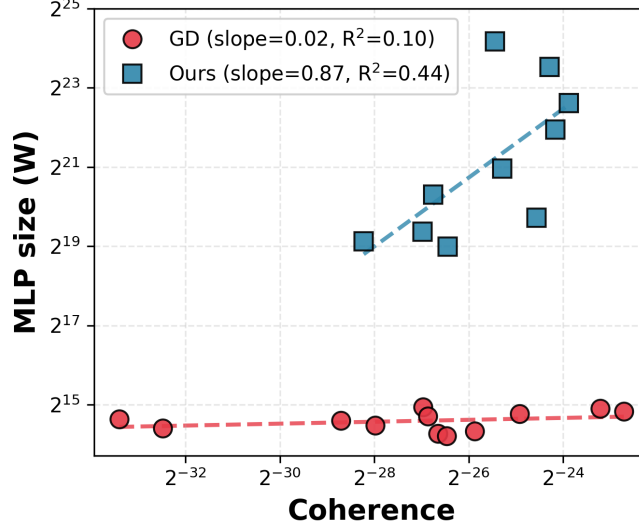


Figure 6: Unlike our decodability metric, ρ , coherence is not strongly predictive of fact-storage capacity for GD nor our constructed MLPs.

5. Train the modified transformer, as outlined in Section 5.1, with frozen key and value transformer embeddings, in the SSFR task corresponding to the fact set we sampled.

Constructed / GD MLPs Setup. Across our SSFR experiments, we use constructed and GD fact-storing MLPs as outlined in Appendix A.1.3.

Transformer Setup. Across all our SSFR experiments we use a modified 1-layer GPT2 transformer (Radford et al., 2019; Karpathy, 2022) with RoPE (Su et al., 2023) positional embeddings, frozen key and value transformer embeddings, RMSNorm normalization layers, single-head attention. Moreover, as outlined in Section 5.1, we tie the transformer and MLP embeddings, remove residual connections, freeze the RMSNorm before the MLP (so that it just projects to the unit sphere) and freeze the *value* and *out-project* matrices of the attention layer to the identity matrix. Across all experiments, we train transformers on a total of 4.8M sequences randomly sampled from the SSFR task, or until convergence, using an AdamW optimizer, with a learning rate of 2×10^{-4} unless stated otherwise.

A.2.3 MLP Size v.s. Facts

In our MLP size (W) v.s. Facts (F) scaling experiments, presented in Section 5.1 and observed in Figure 3.a and Figure 7, we seek to find the smallest MLP size such that the MLP is usable for factual recall by a transformer. We determine whether an MLP is usable by a transformer by testing whether its *fact-adaptive accuracy* is $> 99\%$. To this end, we take a transformer using a fact-storing MLP with embedding-dimension $d = 128$ and run a binary search to find the minimum hidden size h needed to store every fact-set size $F \in \{2^8, \dots, 2^{14}\}$. In this binary search, to reduce noise, we run each experiment corresponding to an MLP size with 4 seeds and take the maximum *fact-adaptive accuracy* out of them. We then report the total MLP size v.s. # of Facts curve outlined by our binary search results.

A.2.4 MLP Usability v.s. Capacity

In our MLP Usability v.s. Accuracy experiments, we study the effect of embedding whitening on the usability v.s. accuracy tradeoff of GD fact MLPs (trained with Cross-Entropy loss), as outlined in Section 5.2. Concretely, we look at transformers using SwiGLU and ReLU fact MLPs, with $d = 128$ and hidden size $m = 1.1h^*$, where h^* is the hidden dimension size found in our scaling experiments from Figure 7.

Concretely, for SwiGLU MLP’s we study ill-conditioned transformer embeddings with $\kappa(\mathbf{K}_t) = \kappa(\mathbf{V}_t) \in \{1.1 \times 10^0, 1.0 \times 10^1, 2.5 \times 10^1, 5.0 \times 10^1, 2.5 \times 10^2, 1.0 \times 10^3, 1.0 \times 10^4, 1.0 \times 10^6\}$, yielding a varied spectrum of ρ values, as observed in Figure 3.b.

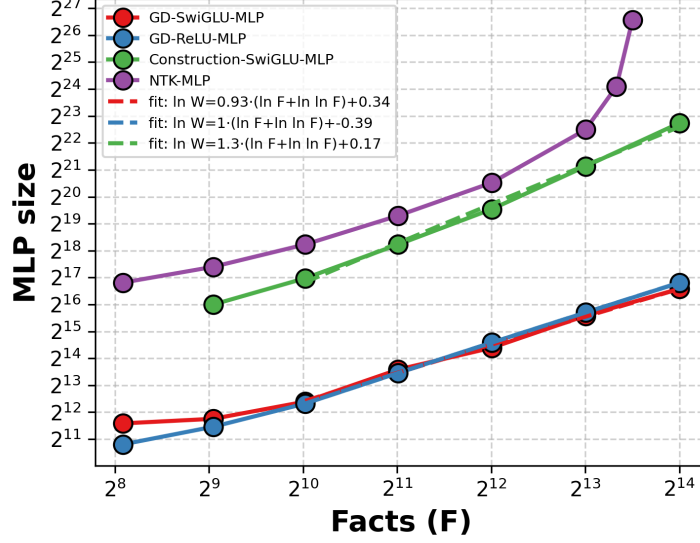


Figure 7: MLP size vs. fact-set size for MLPs with $\geq 99\%$ usability within a Transformer, including ReLU MLPs.

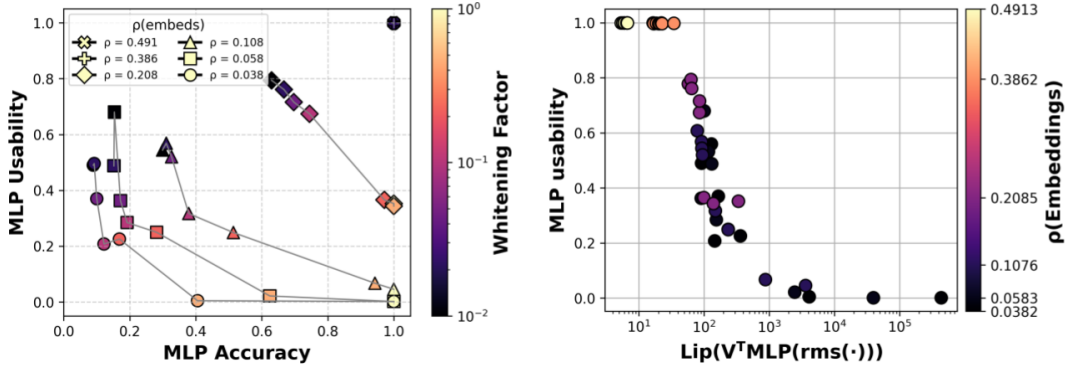


Figure 8: (a) MLP usability within Transformer v.s. MLP storage capacity for a ReLU MLP. We observe a tradeoff between MLP usability within a Transformer and the MLP’s fact-storage capacity. (b) MLP usability within Transformer v.s. its Lipschitz constant for a ReLU MLP. We observe that the measured Lipschitz constant is predictive of an MLP’s usability within Transformers.

In addition, for ReLU MLPs, we look at transformer embeddings with $\kappa(\mathbf{K}_t) = \kappa(\mathbf{V}_t) \in \{1.1 \times 10^0, 1.0 \times 10^1, 1.0 \times 10^2, 1.0 \times 10^3, 1.0 \times 10^4, 1.0 \times 10^5\}$, yielding a varied spectrum of ρ values, as observed in Figure 8.a.

Further, for every ρ , we study the whitening degrees $\alpha \in \{0.0, 0.01, 0.022, 0.046, 0.1, 0.22, 0.46, 1.0\}$. To reduce noise, for every combination of α, ρ , we run experiments for the learning rates $lr \in \{2 \times 10^{-6}, 2 \times 10^{-5}, 2 \times 10^{-4}, 2 \times 10^{-3}, 2 \times 10^{-2}\}$ with 4 seeds each, keeping the transformer with the largest *fact-adaptive* accuracy.

A.2.5 MLP Usability v.s. Lipschitz constant

In our MLP Usability v.s. Lipschitz constant experiments, we study the variation of MLP Usability v.s. an approximation of the Lipschitz constant, as outlined in Section 5.3 and observed in Figure 3.c and Figure 8.b. Concretely, for every transformer obtained in our MLP Usability v.s. Accuracy experiments Section 5.2, we approximate its fact-storing MLP’s Lipschitz constant as the maximum out of 100 random $\mathbf{k}_i$ samples of Equation (8).

A.3 Language Modeling Experiments

A.3.1 Authors and Books Dataset

We introduce a simple language modeling (LM) task to evaluate a transformer’s ability to perform next-token prediction while recalling factual information. In this task, the model is presented with a natural-language sentence expressing a (*book*, *author*) relation and is required to predict each subsequent token in the sequence. Notably, we curate this dataset using author-books relations from the Goodreads Book Graph Dataset (Wan & McAuley, 2018).

Formally, let $f : S_k \rightarrow S_v$ be the authors *fact set*, where $S_k = \{\text{“It”}, \text{“1984”}, \text{“And Then There Were None”}, \dots\}$ is the set of book titles (keys) and $S_v = \{\text{“Stephen King”}, \text{“George Orwell”}, \text{“Agatha Christie”}, \dots\}$ is the set of corresponding authors (values). To simplify analysis, we select exactly one book per author. Let $J = \{(\text{“The author of”}, \text{“is”}), (\text{“Who is the author of”}, \text{“? It is”}), \dots\}$ denote the set of natural-language template prefix-suffix pairs. The LM task given f can then be defined as:

$$\mathcal{S}_{LM}[f] = \{\text{concat}(t_{\text{prefix}}, k, t_{\text{suffix}}, f(k)) \mid (t_{\text{prefix}}, t_{\text{suffix}}) \in J, k \in S_k\}.$$

For example, given the sequence:

$$\underbrace{\text{The author of}}_{\text{template prefix}} \underbrace{1984}_{\text{key}} \underbrace{\text{is}}_{\text{template suffix}} \underbrace{\text{George Orwell}}_{\text{value}}$$

from $\mathcal{S}_{LM}[f]$, the model’s task is to perform next-token prediction *at every position* in the sentence. This LM task allows us to study factual recall in a more natural language modeling setting, complementing the SSFR setup.

A.3.2 Training Setup

The setup we use to train transformers using fact-storing MLPs in the Language Modeling experiments is the same as that outlined in Section A.2.2. However, instead of using a random fact set, we use the authors and books fact-set and use uniformly sampled embeddings.

GD MLP Setup. Notably, in our LM experiments, we only use GD trained fact-storing MLPs, which are trained in a MSE objective (as opposed to a Cross-Entropy objective) to store the fact set under arg-max decoding. Concretely, these MLPs are trained to minimize $L_{MLP}(\mathbf{K}, \mathbf{V}, f) \propto \sum_{i=1}^{|\mathbf{K}|} \|MLP(\mathbf{k}_i) - \mathbf{v}_{f(i)}\|_2^2$.

Transformer Setup In our LM experiments, we use a similar setup as that outlined in Section A.2.2, with some additional modifications we find empirically helpful:

- Replace the state-mixer of the transformer with a Mixture-of-Experts (MoE) module with 2 experts and an MLP router. Concretely, we use a *fact-expert*, which is the frozen fact-storing MLP and a *language-expert*, which is a trainable low-rank linear layer. Intuitively, this MoE setup enables the transformer to selectively use the fact-storing MLP only for factual recall.
- Parametrize the *query* and *key* projections in the attention module with MLPs.

A.3.3 MLP Size v.s. Facts

Similar to Section A.2.3, we perform MLP size (W) v.s. Facts (F) scaling experiments for our transformers, equipped with GD fact MLPs, in the LM task. Concretely, we take transformers equipped with SwiGLU fact MLPs with $d = 256$ and use a binary search with 4 seeds per experiment to determine to find the smallest MLP size W s.t. a transformer can use such an MLP for factual recall on a fact set of size F. As can be observed in Figure 9.a, our transformers can use fact-storing MLPs for factual recall with reasonable scaling in facts per parameter. Furthermore, each of these transformers only suffer a small decay of $\sim 3\%$ in average Cross-Entropy loss for the non-fact tokens of the LM task (e.g. “The”, “author” “of”, etc.) when their MLP is swapped by another one storing a different fact-set (i.e. a different mapping from books to authors).

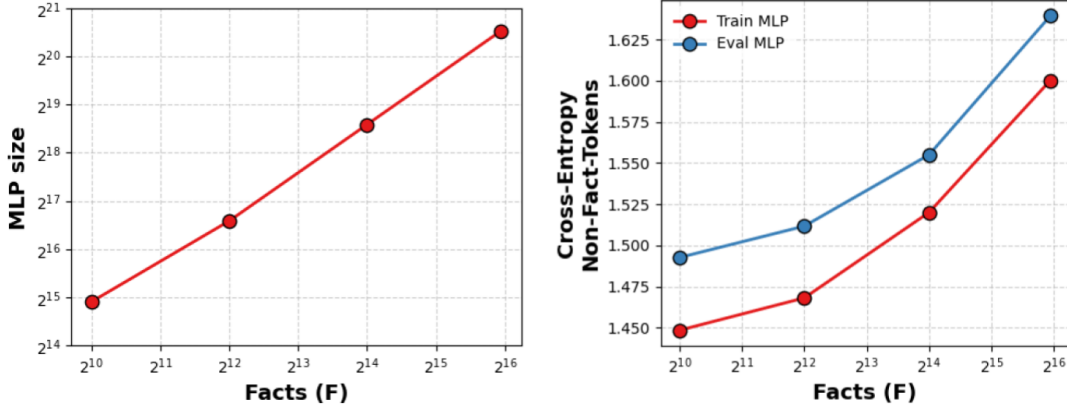


Figure 9: (a) **MLP size vs. fact-set size for MLPs with $\geq 99\%$ usability in LM task within a transformer.** Notably, fact MLPs are usable within transformers for Language Modeling. (b) **CE Loss of on non-fact tokens on a LM task for the transformers in Figure 9.a after swapping their fact-storing MLP for different one.** Notably, the CE Loss of the transformers decays minimally ($\sim 3\%$) when replacing the original MLP (train) with another one storing a different fact-set (eval).

A.3.4 Fact Editing

We evaluate fact-editing methods in the same setting used for our Language Modeling experiments. Concretely, we use the model obtained in those experiments storing 16,000 author-book facts, each represented by 16 rephrases.

To study how different fact-editing approaches behave, we divide the fact set into two subsets: a *preserved fact set*, whose facts the editor should maintain, and an *altered fact set*, whose facts the editor should modify. We run experiments using several combinations of preserved/altered fact set sizes: $\{(6554, 1638), (3277, 819), (1311, 327)\}$, which are subsets of the original fact set of 16,000 facts.

We evaluate each editing method using three standard metrics. *Specificity* measures accuracy on the altered-fact set, indicating how well the method performs the intended edits. *Efficacy* measures accuracy on the preserved-fact set, capturing whether the method avoids unintended side effects. *Paraphrase* evaluates the accuracy on paraphrases of the altered facts, measuring how well edits generalize beyond the training prompts. We also report a *Score*, defined as the harmonic mean of these three metrics.

We compare four editing methods. Our method, *MLP swapping*, trains an MLP to store the full altered-fact set and swaps it into the transformer in place of the original fact-storing MLP. The remaining three methods: MEMIT (Meng et al., 2023c), AlphaEdit (Fang et al., 2025), and ROME (Meng et al., 2023b), are existing weight-update-based editors, which are set up to alter the *altered fact set* and preserve the *preserved fact set*. Because these methods are designed for large language models and real-world text, we adapt them to our simplified 1-layer transformer setup. For each, we perform a grid search over its hyperparameters and report the accuracies corresponding to the configuration achieving the best overall score.

- **MEMIT:** We search over `train_steps` $\in \{10, 25, 100\}$, `lr` $\in \{0.005, 0.05, 0.5\}$, `$\lambda$` $\in \{1.5 \times 10^4, 1.5 \times 10^3, 1.5 \times 10^2, 1\}$, and `clip_norm` $\in \{0.5, 0.75, 1\}$.
- **AlphaEdit:** We search over `train_steps` $\in \{10, 25, 100\}$, `lr` $\in \{0.005, 0.05, 0.5\}$, `clip_norm` $\in \{0.5, 0.75, \text{None}\}$, and `singular_value_tolerance` $\in \{10^{-2}, 1, 10\}$.
- **ROME:** We search over `train_steps` $\in \{10, 25, 100\}$, `lr` $\in \{0.005, 0.05, 0.5\}$, `wd` $\in \{1.5 \times 10^{-3}, 1.5 \times 10^{-4}, 0\}$, and `early_stopping_loss` $\in \{5 \times 10^{-2}, \text{None}\}$.

For these methods, we apply residual updates to the output of the MLP inside the MoE module on the final token of the input prompt. We find this appropriate since our transformer has a single layer, so the fact-storing MLP directly precedes the logits without any intervening attention layers. Moreover, we do not introduce random token prefixes when computing residual vectors. Instead, we use a single templated prompt per fact. In addition, for ROME, we omit the KL-divergence term from the residual computation given the simplicity of our dataset, where each subject (author) appears in only one relation, mapping uniquely to a book.

B Theoretical Results

This section is organized as follows:

1. In Section B.1 we discuss notation and external results that will be useful throughout the appendix.
2. In Section B.2 we provide additional preliminary information on softmax decoding and fact storage capacity in support of Section 2.1.
3. In Section B.3 we detail our encoding construction in support of Section 4.1.
4. In Section B.5 we prove bounds on ρ , and detail our decoding construction in support of Section 4.2.
5. In Section B.6 we prove our full construction in support of Section 4.3.
6. In Section B.7 we explore the interaction between ρ and transformations on embeddings in support of Section 3.
7. In Section B.8 we prove that our construction has bounded bit complexity.
8. In Section B.9 we prove bounds on the spherical Chebyshev value.
9. In Section B.10 we collect deferred proofs from the previous sections.

B.1 Notation and External Results

All vectors are denoted by bold lower case letters (*e.g.*, $\mathbf{x}$), and matrices by bold uppercase letters (*e.g.*, $\mathbf{V}$). All vectors are assumed to be in column form and indices will start from 1. We denote $\mathbb{S}^{d-1}$ to be the unit sphere in $\mathbb{R}^d$.

For a set $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N^\top]$ with $\mathbf{u}_i \in \mathbb{S}^{d-1}$, set

$$\rho(\mathbf{U}; \mathbf{V}) := \min_{i \neq j} \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\|_2}, \quad \rho(\mathbf{V}) := \max_{\mathbf{U}} \rho(\mathbf{U}; \mathbf{V}).$$

We use the former definition of ρ in several sections of the appendix as it is somewhat easier to work with.

We generally abbreviate $\|x\|_2$ to $\|x\|$; other norms are explicitly marked. We occasionally use $|\cdot|$ to denote the number of rows in a matrix (ie. $|\mathbf{K}| = \#$ of rows in $\mathbf{K}$). Additionally, note that $O(d)$ is the set of $d \times d$ orthonormal matrices and is distinguishable from Big-O notation by the type of its elements (eg. $\mathbf{U} \in O(d)$).

A random vector $x \in \mathbb{R}^d$ is *rotationally invariant* if

$$Vx \sim x \quad \forall V \in O(d),$$

i.e., its distribution depends only on $\|x\|_2$ and not on its direction (e.g. $x \sim \mathcal{N}(0, I_d)$). When we say the keys are rotationally invariant, we mean they are i.i.d. draws from such a distribution.

B.1.1 The Bubeck Result

Fix some dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i \in [n]} \subset (\mathbb{R}^d \times \mathbb{R})^n$. Let $\mathcal{F}_k$ be the set of functions of the form

$$f(\mathbf{x}) = \mathbf{a}^\top \text{ReLU}(\mathbf{W}\mathbf{x} + \mathbf{b})$$

where $\mathbf{a} = (a_1, \dots, a_k)^\top \in \mathbb{R}^k$, $\mathbf{b} = (b_1, \dots, b_k)^\top \in \mathbb{R}^k$, and $\mathbf{W} \in \mathbb{R}^{k \times d}$ with rows $\mathbf{w}_1^\top, \dots, \mathbf{w}_k^\top$. Denote $\mathbf{y} = (y_1, \dots, y_n)$ and $\mathbf{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))$ with $f \in \mathcal{F}_k$, $f: \mathbb{R}^d \rightarrow \mathbb{R}$. Note that this is equivalent to the definition in (Bubeck et al., 2020).

We will use the following result from (Bubeck et al., 2020):

Theorem B.1.1. *Let $(\mathbf{x}_i)_{i \in [n]}$ be in general position in $\mathbb{R}^d$ (i.e., any hyperplane contains at most d points). Then there exists $f \in \mathcal{F}_{4 \lceil \frac{n}{d} \rceil}$ such that $\mathbf{f} = \mathbf{y}$.*

We now provide a proof sketch of the result to provide intuition. For a full proof, see Proposition 4 of (Bubeck et al., 2020).

Proof. Split the n samples into $r = \lceil n/d \rceil$ disjoint sets of indices $S_1, \dots, S_r$ of size d (last may be smaller). By general position, for each block S there is a hyperplane $H_S = \{\mathbf{x} : \mathbf{z}_S \cdot \mathbf{x} = b_S\}$ that contains exactly $\{\mathbf{x}_i : i \in S\}$.

Define the function, for small enough $\delta > 0$:

$$g_{\mathbf{z}, \mathbf{v}, b, \delta}(\mathbf{x}) := \frac{\text{ReLU}((\mathbf{z} + \delta \mathbf{v}) \cdot \mathbf{x} - b) - \text{ReLU}(\mathbf{z} \cdot \mathbf{x} - b)}{\delta}.$$

If δ preserves the signs of $\mathbf{z} \cdot \mathbf{x}_i - b$ for all data (i.e., if no input crosses the ReLU boundary), then

$$g_{\mathbf{z}, \mathbf{v}, b, \delta}(\mathbf{x}_i) = \begin{cases} \mathbf{v} \cdot \mathbf{x}_i, & \mathbf{z} \cdot \mathbf{x}_i > b, \\ 0, & \mathbf{z} \cdot \mathbf{x}_i < b. \end{cases}$$

Set

$$h_{\mathbf{z}, \mathbf{v}, b, \delta}(\mathbf{x}) := g_{\mathbf{z}, \mathbf{v}, b - \tau, \delta}(\mathbf{x}) - g_{\mathbf{z}, \mathbf{v}, b + \tau, \delta}(\mathbf{x}),$$

for small enough $\tau > 0$. We then have that $h_{\mathbf{z}_S, \mathbf{v}_S, b_S, \tau, \delta}(\mathbf{x}_i) = \mathbf{v}_S \cdot \mathbf{x}_i$ for $\mathbf{x}_i \in S$ and 0 otherwise. Choices which always work are $0 < \tau < \frac{1}{2} \min_{i \notin S} |\mathbf{u}_S \cdot \mathbf{x}_i - b_S|$ and $\delta \leq \frac{1}{2} \min_{i \in [n]} \min_{\sigma \in \{-1, 1\}} \frac{|\mathbf{z}_S \cdot \mathbf{x}_i - (b_S + \sigma \tau)|}{|\mathbf{v}_S \cdot \mathbf{x}_i|}$.

Pick S_i such that X_{S_i} (the matrix collecting all $\mathbf{x}_j \in S_i$), by general position of $\mathbf{v}_{iS}$, has full rank for all i . For each block S , solve $X_S \mathbf{v}_S = \mathbf{y}_S$ and define $f_S(\mathbf{x}) := h_{\mathbf{z}_S, \mathbf{v}_S, b_S, \tau, \delta}(\mathbf{x})$. Then $f_S(\mathbf{x}_i) = y_i$ for $i \in S$ and 0 for $i \notin S$.

Finally,

$$f(\mathbf{x}) := \sum_{t=1}^r f_{S_t}(\mathbf{x}) \in \mathcal{F}_{4r} = \mathcal{F}_{4\lceil n/d \rceil} \quad \text{and} \quad f(\mathbf{x}_i) = y_i \quad \forall i \in [n].$$

□

B.1.2 Johnson-Lindenstrauss Inner Product Preservation

We will use the following result from (Kalavasis et al., 2024).

We say that a random matrix $\mathbf{A} \in \mathbb{R}^{k \times d}$ is a *JL-matrix* if either $\mathbf{A}_{i,j} \sim_{i.i.d} \mathcal{N}(0, 1/k)$ or $\mathbf{A}_{i,j} \sim_{i.i.d} U\{-1/\sqrt{k}, 1/\sqrt{k}\}$.

Corollary B.1.2. Fix $\epsilon, \delta_{JL} \in (0, 1)$. Let $\mathbf{A} \in \mathbb{R}^{k \times d}$ be a JL-matrix for $k = \Omega(\epsilon^{-2} \log(\frac{1}{\delta_{JL}}))$. Then for any $x, z \in \mathbb{R}^d$,

$$\Pr_{\mathbf{A}}[|\mathbf{z}^\top \mathbf{x} - (\mathbf{A}\mathbf{z})^\top \mathbf{A}\mathbf{x}| > \epsilon \|\mathbf{z}\| \cdot \|\mathbf{x}\|] \leq \delta_{JL}.$$

B.1.3 Sub-gaussian rows

We will use the following result from (Vershynin, 2018).

Theorem B.1.3. Let $\mathbf{A}$ be an $N \times n$ matrix whose rows $\mathbf{A}_i$ are independent sub-gaussian isotropic random vectors in $\mathbb{R}^n$. Then for every $t \geq 0$, with probability at least $1 - 2\exp(-ct^2)$ one has

$$\sqrt{N} - C\sqrt{n} - t \leq s_{\min}(\mathbf{A}) \leq \sqrt{N} + C\sqrt{n} + t.$$

Here $C = C_K, c = c_K > 0$ depend only on the subgaussian norm $K = \max_i \|\mathbf{A}_i\|_{\psi_2}$ of the rows.

B.2 Additional Details on Section 2.1

In Section 2.1 we define what it means for a model to store a fact set. Here, we describe why this is equivalent to outputting the correct value token under softmax decoding, and for completeness provide a proof of Theorem 2.1.1. We use the definition of softmax decodability as follows.

Definition B.2.1. Let $\mathbf{H} \in \mathbb{R}^{|\mathbf{K}| \times d}$. A family of output embeddings $\{\mathbf{v}_i\}_{i=1}^{|\mathbf{K}|} \subset \mathbb{R}^d$ is softmax-decodable if there exists a matrix $\mathbf{M} \in \mathbb{R}^{d \times m}$ such that for all i ,

$$\|\text{softmax}_j(\langle \mathbf{M} \cdot \mathbf{H}[i], \mathbf{v}_j \rangle) - \mathbf{e}_i\|_\infty < \alpha. \quad (9)$$

¹¹ for some $\frac{1}{2} > \alpha > 0$.

¹¹As a reminder, here the $\text{softmax}_j(\langle \mathbf{z}, \mathbf{y}_j \rangle)$ notation means a vector of length n where the ℓ -th coordinate is given, for some arbitrary $\mathbf{z} \in \mathbb{R}^d$, by:

$$(\text{softmax}_j(\langle \mathbf{z}, \mathbf{y}_j \rangle))_\ell = \frac{\exp(\langle \mathbf{z}, \mathbf{v}_\ell \rangle)}{\sum_{k=1}^n \exp(\langle \mathbf{z}, \mathbf{v}_k \rangle)}$$

In the notation of Section 4.2, we have $\mathbf{H}[i] := \mathbf{D}\mathbf{u}_i$. The following lemma shows that this is equivalent to the provided “dot-product” version.

Lemma B.2.2. *A set of output embeddings $\{\mathbf{v}_i\}$ is softmax-decodable if and only if there exists an $\mathbf{M}$ such that, for every $i \neq j$, $\langle \mathbf{M} \cdot \mathbf{H}[i], \mathbf{v}_i \rangle > \langle \mathbf{M} \cdot \mathbf{H}[i], \mathbf{v}_j \rangle$.*

Proof. See Section B.10.6 □

The following theorem is a formalized version of Theorem 2.1.1.

Theorem B.2.3 (Information-theoretic capacity bounds). *Let an MLP have W trainable real weights, each stored with a fixed precision of p bits; write $B = pW = \Theta(W)$ for the total number of bits that can be set by training. Let F be the number of (key,value) pairs (“facts”) we wish to memorize.*

1. **Multi-valued facts.** *If every key may take any of the F values— i.e. the fact set is a function $f : [F] \rightarrow [F]$ —then any such table representable by the network satisfies*

$$F = O\left(\frac{W}{\log W}\right).$$

2. **Binary facts.** *If every key is mapped to a bit ($f : [F] \rightarrow \{0,1\}$) the capacity bound tightens to*

$$F = O(W).$$

Proof. Let $\mathcal{H}$ be the set of hypothesis functions the parameterised family can express. Because each of the $B = \Theta(W)$ bits can be chosen independently,

$$|\mathcal{H}| \leq 2^B = 2^{\Theta(W)}.$$

In the case of multi-valued facts, there are F^F distinct functions $[F] \rightarrow [F]$. Representability of all such maps demands

$$2^{\Theta(W)} \geq F^F.$$

Taking $\log_2$ and rearranging:

$$F \log_2 F = O(W) \implies F = O\left(\frac{W}{\log_2 W}\right),$$

since $\log_2 F = \Theta(\log_2 W)$ whenever F grows at most polynomially in W .

For binary facts there are only 2^F possibilities, so the same counting gives

$$2^{\Theta(W)} \geq 2^F \implies F = O(W).$$

□

B.3 Additional Details for Section 4.1

B.3.1 A Naïve Construction

We briefly describe a naïve construction, which we compare to ours in Table 1. Let $\mathbf{K} = \{\mathbf{k}_i\}_{i=1}^{|\mathbf{K}|} \subset \mathbb{R}^d$ and stack input embeddings as columns $\tilde{\mathbf{K}} = [\mathbf{k}_1 \cdots \mathbf{k}_{|\mathbf{K}|}] \in \mathbb{R}^{d \times |\mathbf{K}|}$. Consider

$$g(\mathbf{x}) = \mathbf{V} \text{ReLU}(\tilde{\mathbf{K}}^\top \mathbf{x} - \mathbf{b}), \quad \mathbf{V} \in \mathbb{R}^{d \times |\mathbf{K}|}, \quad \mathbf{b} \in \mathbb{R}^{|\mathbf{K}|}.$$

For each j , define $\alpha_j := \langle \mathbf{k}_j, \mathbf{k}_j \rangle$ and $\beta_j := \max_{i \neq j} \langle \mathbf{k}_j, \mathbf{k}_i \rangle$, and assume $\alpha_j > \beta_j$. Choose any $b_j \in (\beta_j, \alpha_j)$ and set $a_i := \alpha_i - b_i > 0$. Then

$$\text{ReLU}(\tilde{\mathbf{K}}^\top \mathbf{k}_i - \mathbf{b}) = a_i \mathbf{e}_i,$$

so taking

$$\mathbf{V} = [\mathbf{v}_{f(1)}/a_1 \quad \mathbf{v}_{f(2)}/a_2 \quad \cdots \quad \mathbf{v}_{f(H)}/a_H]$$

gives exact retrieval $g(\mathbf{k}_i) = \mathbf{v}_{f(i)}$. However, the hidden size is $|\mathbf{K}|$, and the parameter count is $\Theta(d|\mathbf{K}|)$ which is much too large.

B.3.2 Two-hot Construction

Construction B.1 (Encoder Construction, Two-Hot). Fix a dimension $d \geq 2$ and let $\{\mathbf{e}_1, \dots, \mathbf{e}_d\} \subset \mathbb{R}^d$ be the standard basis. Define the key set

$$\mathbf{K} := \{\mathbf{k}_{i,j} = \mathbf{e}_i - \mathbf{e}_j : i \neq j, i, j \in [d]\}, \quad |\mathbf{K}| = d(d-1).$$

Let $h : \{(i, j) \mid i \neq j, i, j \in [d]\} \rightarrow [0, 1]$ prescribe a target scalar for each key $\mathbf{k}_{i,j}$. Define the (one-hidden-layer) encoder $\mathbf{enc} : \mathbb{R}^d \rightarrow \mathbb{R}$ by

$$\mathbf{enc}(\mathbf{x}) = \mathbf{1}^\top \text{ReLU}(\mathbf{A} \mathbf{x} - \mathbf{1}),$$

where $\mathbf{1} \in \mathbb{R}^d$ is the all-ones vector, ReLU acts elementwise, and the weight matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ is

$$\mathbf{A}[p, q] = \begin{cases} 1 & \text{if } p = q, \\ -h(p, q) & \text{if } p \neq q. \end{cases}$$

Then, for every $i \neq j \in [d]$,

$$\mathbf{enc}(\mathbf{k}_{i,j}) = h(i, j).$$

Proof. Fix $i \neq j$ and consider $\mathbf{k}_{i,j} = \mathbf{e}_i - \mathbf{e}_j$. For each coordinate $p \in [d]$,

$$\begin{aligned} (\mathbf{A} \mathbf{k}_{i,j} - \mathbf{1})[p] &= \mathbf{A}[p, i] - \mathbf{A}[p, j] - 1 \\ &= \begin{cases} 1 - (-h(i, j)) - 1 = h(i, j), & p = i, \\ (-h(j, i)) - 1 - 1 = -h(j, i) - 2, & p = j, \\ (-h(p, i)) - (-h(p, j)) - 1 = h(p, j) - h(p, i) - 1, & p \notin \{i, j\}. \end{cases} \end{aligned}$$

Since $h(\cdot, \cdot) \in [0, 1]$, we have: (i) the i -th coordinate equals $h(i, j) \geq 0$; (ii) the j -th coordinate is ≤ -2 and thus strictly negative; and (iii) for $p \notin \{i, j\}$, $h(p, j) - h(p, i) - 1 \leq 1 - 0 - 1 = 0$, hence these coordinates are nonpositive. Applying ReLU elementwise zeroes out all nonpositive coordinates and preserves the i -th coordinate, yielding

$$\text{ReLU}(\mathbf{A} \mathbf{k}_{i,j} - \mathbf{1})[p] = \begin{cases} h(i, j), & p = i, \\ 0, & p \neq i. \end{cases}$$

Finally, summing with $\mathbf{1}^\top$ gives $\mathbf{enc}(\mathbf{k}_{i,j}) = \mathbf{1}^\top \text{ReLU}(\mathbf{A} \mathbf{k}_{i,j} - \mathbf{1}) = h(i, j)$, as claimed. $\square$

Remark In the above proof, we say that h outputs values in $[0, 1]$ without loss of generality. Because the domain of h is finite, let $a := \min_{i \neq j} h(i, j)$ and $b := \max_{i \neq j} h(i, j)$. Set $\Delta := b - a$ (take $\Delta = 1$ if $a = b$) and define the normalized function

$$\tilde{h}(i, j) = \frac{h(i, j) - a}{\Delta} \in [0, 1].$$

Build the encoder above for $\tilde{h}$, yielding $\widetilde{\mathbf{enc}}(\mathbf{k}_{i,j}) = \tilde{h}(i, j)$. Recover h exactly with the 1D transform:

$$\mathbf{enc}_h(\mathbf{x}) = a + \Delta \cdot \widetilde{\mathbf{enc}}(\mathbf{x}).$$

This post-composition changes only $O(1)$ top-layer parameters and does not affect the gating argument, so we may assume $\text{range}(h) \subset [0, 1]$ without loss of generality.

B.3.3 Discussion of Nichani et al.'s polylog factor

Throughout the paper, we compare our construction with that given by Nichani et al. (2024). Here, we discuss why the number of parameters of the Nichani et al. (2024) construction is at least $\Omega(|\mathbf{K}| \log^{12} |\mathbf{V}|)$. For comparability, we use some notation such as m, d from Nichani et al. (2024).

Nichani et al. (2024)'s result for a one-layer MLP with non-linear activation is presented in their Theorem 9 in Appendix B. Their theorem statement is as follows for $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{m \times d}$.

Assumption 3. σ is a polynomial of degree q . Furthermore, if $\sigma(z) = \sum_{k=0}^q c_k h_k(z)$ is the Hermite decomposition of σ , then $c_k \neq 0$ for all $0 \leq k \leq q$.

Theorem 9 (Nichani et al., 2024). Let $\epsilon \in (0, 1)$ be a fixed constant. Assume that $d \geq N^\epsilon$ and $N \geq C_1(\epsilon)$, where $C_1(\epsilon)$ is a constant depending only on ϵ . Assume that q in Assumption 3 satisfies $q = \frac{C_2}{\epsilon}$ for some $C_2 > 2$. Then, if

$$md \gtrsim N(C_3 \log(MN/\delta))^{C_4/\epsilon},$$

with probability $1 - \delta$ over the draw of the embeddings, there exists $\mathbf{V}, \mathbf{W}$ such that

$$\arg \max_{y \in [M]} \mathbf{u}_y^\top \mathbf{V}^\top \sigma(\mathbf{W} \mathbf{e}_x) = f^*(x) \quad (20)$$

for all $x \in [N]$.

Mapping their notation to ours, we have $N := |\mathbf{K}|$ and $M := 32$. In Theorem 9, they require $md \gtrsim N C_3^q \log^{4q+4}\left(\frac{1}{\delta'}\right)$ where $\delta' = \frac{\delta}{MN}$. This gives $\frac{md}{N} \gtrsim C_3^q (\log(MN/\delta))^{4q+4}$ and for $\delta = N^{-c}$ for a constant $c > 0$,

$$\log\left(\frac{MN}{\delta}\right) = \Theta(\log N) \implies \frac{md}{N} \gtrsim C_3^q (\log N)^{4q+4}.$$

Using their dimensional regime $d \geq N^\epsilon$ gives $\log N = \Theta(\log d)$. In addition, they assume that $|\mathbf{K}| = |\mathbf{V}|$, so

$$\log N \asymp \log |\mathbf{V}| \implies md \gtrsim C_3^q |\mathbf{K}| (\log |\mathbf{V}|)^{4q+4}.$$

Since $q = \frac{C_2}{\epsilon} > 2$ implies $4q + 4 \geq 12$, we have

$$\#\text{Parameters} \simeq md \gtrsim |\mathbf{K}| \log^{12} |\mathbf{V}|.$$

B.4 Additional Details for Section 4.1

This section is divided into three parts:

1. In Section B.4.1, we provide an overview of our encoder architecture, desiderata, and more. We describe how we break the encoder into gated or non-gated encoder gadgets, each of which output one component of the final result.
2. In Section B.4.2, we describe the gated encoder gadget in more detail and prove that it works for asymptotically optimal parameter counts.
3. In Section B.4.3, we describe the non-gated encoder gadget in more detail. We show how we can construct the non-gated encoder gadget using the gated encoder gadget algorithm, and we illustrate how, in the special case of a ReLU encoder, we obtain a generalization of the Baum network from Bubeck et al. (2020).

B.4.1 Overview of the Encoder

Our encoder is a single-hidden layer MLP mapping key embeddings to compressed output embeddings.

Encoder Structure Our encoder is either a gated MLP

$$\mathbf{enc}(\mathbf{x}) = \mathbf{E}(\sigma(\mathbf{G}\mathbf{x} + \mathbf{b}_G) \odot (\mathbf{A}\mathbf{x} + \mathbf{b}_A)) + \mathbf{b}_E,$$

or a non-gated MLP

$$\mathbf{enc}(\mathbf{x}) = \mathbf{E}\sigma(\mathbf{A}\mathbf{x} + \mathbf{b}_A) + \mathbf{b}_E$$

with $\mathbf{A}, \mathbf{G} \in \mathbb{R}^{h \times d}$, $\mathbf{E} \in \mathbb{R}^{m \times h}$, $\mathbf{b}_A, \mathbf{b}_G \in \mathbb{R}^h$, $\mathbf{b}_E \in \mathbb{R}^m$, $\mathbf{x} \in \mathbb{R}^d$, and $\sigma : \mathbb{R} \rightarrow \mathbb{R}$.

Gated MLPs simplify our analysis and are now popular across frontier models (Yang et al., 2025a; Dubey et al., 2024). In Section B.4.3, we extend our arguments to non-gated encoders.

Encoder Framework Objective Given key embeddings $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$, compressed output embeddings $\mathbf{C} \in \mathbb{R}^{|\mathbf{V}| \times m}$, and a mapping f , the objective of our encoder framework is to produce an MLP $\mathbf{enc}$ with a minimal number of parameters such that $\mathbf{enc}(\mathbf{k}_i) = \mathbf{c}_{f(i)}$ for all $i \in [|\mathbf{K}|]$.

Construction Our constructed encoder builds m encoder gated or non-gated gadgets, for each $j \in [m]$:

$$\mathbf{enc}_j(\mathbf{x}) = \mathbf{1}_{\tilde{h}}^\top \left[\sigma(\mathbf{G}^{(j)}\mathbf{x} + \mathbf{b}_G^{(j)}) \odot (\mathbf{A}^{(j)}\mathbf{x} + \mathbf{b}_A^{(j)}) \right] + b_E^{(j)};$$

or alternatively,

$$\mathbf{enc}_j(\mathbf{x}) = \mathbf{E}^{(j)}\sigma(\mathbf{A}^{(j)}\mathbf{x} + \mathbf{b}_A^{(j)}) + b_E^{(j)}$$

$$\text{with } \mathbf{G}^{(j)}, \mathbf{A}^{(j)} \in \mathbb{R}^{\tilde{h} \times d}, \quad \mathbf{E}^{(j)} \in \mathbb{R}^{1 \times \tilde{h}}, \quad \mathbf{b}_G^{(j)}, \mathbf{b}_A^{(j)} \in \mathbb{R}^{\tilde{h}}, \quad b_E^{(j)} \in \mathbb{R}$$

that map $\mathbf{k}_i$ to $\mathbf{c}_{f(i)}[j] \in \mathbb{R}$, respectively, where $\tilde{h} = h/m$. We can set the down projection to $\mathbf{1}^\top$ in the gated encoder gadget without loss of generality by replacing $\mathbf{A}^{(j)}$ with $\text{diag}(\mathbf{E}^{(j)})\mathbf{A}^{(j)}$ and $\mathbf{b}_A^{(j)}$ with $\text{diag}(\mathbf{E}^{(j)})\mathbf{b}_A^{(j)}$. We will apply a similar technique in the case of the non-gated encoder gadget, but it is more involved.

We will demonstrate that these gadgets require only $O(|\mathbf{K}|)$ parameters. By stacking all m gadgets together, one for each target dimension j , we can construct $\mathbf{c}_{f(i)}$ with a total of $O(m|\mathbf{K}|)$ parameters, as shown in Algorithm 6.

We will describe the gated and non-gated encoder gadgets in Appendix B.4.2 and B.4.3, respectively. We will drop the j indexing everywhere for notational simplicity.

Algorithm 6 Encoder Construction (ENCODER)

Require: Key embeddings $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$, Compressed output embeddings $\mathbf{C} \in \mathbb{R}^{|\mathbf{V}| \times m}$, Fact-mapping $f : [|\mathbf{K}|] \rightarrow [|\mathbf{V}|]$

Require: Hidden size h , activation σ , gated MLP flag GATED, bias flag BIAS, tolerance δ

```

1:  $\tilde{h} := h/m$ 

2: for  $j = 1$  to  $m$  do
3:    $\mathbf{o}^{(j)} := [\mathbf{C}_{f(1),j}, \dots, \mathbf{C}_{f(|\mathbf{K}|),j}] \in \mathbb{R}^{|\mathbf{K}|}$ 
4:   if GATED:
5:      $\mathbf{enc}_j(\mathbf{x}) := \mathbf{E}^{(j)} \left( \sigma(\mathbf{G}^{(j)}\mathbf{x} + \mathbf{b}_G^{(j)}) \odot (\mathbf{A}^{(j)}\mathbf{x} + \mathbf{b}_A^{(j)}) \right) + \mathbf{b}_E^{(j)} \leftarrow$ 
       GATEDENCODERGADGET( $\mathbf{K}, \mathbf{o}^{(j)}, \tilde{h}, \sigma, \text{BIAS}$ )
6:   else:
7:      $\mathbf{enc}_j(\mathbf{x}) := \mathbf{E}^{(j)} \sigma(\mathbf{A}^{(j)}\mathbf{x} + \mathbf{b}_A^{(j)}) + \mathbf{b}_E^{(j)} \leftarrow \text{ENCODERGADGET}(\mathbf{K}, \mathbf{o}^{(j)}, \tilde{h}, \sigma, \text{BIAS}, \delta)$ 
8:   end for

9: Stack  $\mathbf{A} := \begin{bmatrix} \mathbf{A}^{(1)} \\ \vdots \\ \mathbf{A}^{(m)} \end{bmatrix} \in \mathbb{R}^{h \times d}$ ,  $\mathbf{b}_A := \begin{bmatrix} \mathbf{b}_A^{(1)} \\ \vdots \\ \mathbf{b}_A^{(m)} \end{bmatrix} \in \mathbb{R}^h$ , and  $\mathbf{b}_E := \begin{bmatrix} \mathbf{b}_E^{(1)} \\ \vdots \\ \mathbf{b}_E^{(m)} \end{bmatrix} \in \mathbb{R}^m$ 

10:  $\mathbf{E} := \begin{bmatrix} \mathbf{E}^{(1)} & \mathbf{0}_{1 \times \tilde{h}} & \dots & \mathbf{0}_{1 \times \tilde{h}} \\ \mathbf{0}_{1 \times \tilde{h}} & \mathbf{E}^{(2)} & \dots & \mathbf{0}_{1 \times \tilde{h}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1 \times \tilde{h}} & \mathbf{0}_{1 \times \tilde{h}} & \dots & \mathbf{E}^{(m)} \end{bmatrix} \in \mathbb{R}^{m \times h}$ 

11: if GATED:
12:   Stack  $\mathbf{G} := \begin{bmatrix} \mathbf{G}^{(1)} \\ \vdots \\ \mathbf{G}^{(m)} \end{bmatrix} \in \mathbb{R}^{h \times d}$  and  $\mathbf{b}_G := \begin{bmatrix} \mathbf{b}_G^{(1)} \\ \vdots \\ \mathbf{b}_G^{(m)} \end{bmatrix} \in \mathbb{R}^h$ 

13: if GATED:
14:    $\mathbf{enc}(\mathbf{x}) := \mathbf{E}(\sigma(\mathbf{G}\mathbf{x} + \mathbf{b}_G) \odot (\mathbf{A}\mathbf{x} + \mathbf{b}_A)) + \mathbf{b}_E$ 
15: else
16:    $\mathbf{enc}(\mathbf{x}) := \mathbf{E}\sigma(\mathbf{A}\mathbf{x} + \mathbf{b}_A) + \mathbf{b}_E$ 
17: return enc

```

B.4.2 Gated Encoder Theory

Our gated encoder gadget will follow two simple steps: 1) pick $\mathbf{G}$, and 2) solve the resulting linear system for $\mathbf{A}$. The rest of this section will be dedicated to defining the linear system for $\mathbf{A}$ and providing conditions for a solution to exist.

Define

$$\begin{aligned} \mathbf{\Sigma} &= \sigma(\mathbf{G}\mathbf{K}^\top + \mathbf{b}_G \mathbf{1}_{|\mathbf{K}|}^\top) \in \mathbb{R}^{h \times |\mathbf{K}|} \\ \mathbf{o} &= [\mathbf{c}_{f(1)}[j], \dots, \mathbf{c}_{f(|\mathbf{K}|)}[j]]^\top \end{aligned}$$

where $\mathbf{b}_G = \mathbf{0}$ if **enc** has no biases.

If **enc** has no biases, further define

$$\begin{aligned} \mathbf{M}(\mathbf{\Sigma}, \mathbf{K}) &= [\text{diag}(\mathbf{\Sigma}_1)\mathbf{K}, \dots, \text{diag}(\mathbf{\Sigma}_h)\mathbf{K}] \in \mathbb{R}^{|\mathbf{K}| \times dh} \\ \text{vec}(\mathbf{A}) &= [\mathbf{a}_1, \dots, \mathbf{a}_h]^\top \in \mathbb{R}^{dh}. \end{aligned}$$

The $\mathbf{A}$ matrices such that $\mathbf{enc}(\mathbf{k}_i) = \mathbf{c}_{f(i)}[j]$ for all $i \in [|\mathbf{K}|]$ are exactly the solutions to the linear system

$$\mathbf{M}(\mathbf{\Sigma}, \mathbf{K}) \text{vec}(\mathbf{A}) = \mathbf{o}.$$

The above holds since once $\mathbf{\Sigma}$ entries are fixed, the encoder output is linear in the entries of $\mathbf{A}$.

If instead **enc** does have biases, define

$$\begin{aligned}\tilde{d} &= d + 1 \\ D &= h\tilde{d} + 1 \\ \tilde{\mathbf{K}} &= [\mathbf{K}, \mathbf{1}_{|\mathbf{K}|}] \in \mathbb{R}^{|\mathbf{K}| \times \tilde{d}} \\ \tilde{\mathbf{M}}(\boldsymbol{\Sigma}, \mathbf{K}) &= [\text{diag}(\boldsymbol{\Sigma}_1)\tilde{\mathbf{K}}, \dots, \text{diag}(\boldsymbol{\Sigma}_h)\tilde{\mathbf{K}}, \mathbf{1}_{|\mathbf{K}|}] \in \mathbb{R}^{|\mathbf{K}| \times D} \\ \text{vec}(\mathbf{A}, \mathbf{b}_A, b_E) &= [\mathbf{a}_1, \mathbf{b}_A[1], \dots, \mathbf{a}_h, \mathbf{b}_A[h], b_E]^\top \in \mathbb{R}^D.\end{aligned}$$

The $\mathbf{A}$, $\mathbf{b}_A$, and b_E such that $\mathbf{enc}(\mathbf{k}_i) = \mathbf{c}_{f(i)}[j]$ for all $i \in |\mathbf{K}|$ are exactly the solutions to the linear system

$$\tilde{\mathbf{M}}(\boldsymbol{\Sigma}, \mathbf{K}) \text{vec}(\mathbf{A}, \mathbf{b}_A, b_E) = \mathbf{o}.$$

To obtain a construction, it is sufficient to choose $\boldsymbol{\Sigma}$ such that the system is solvable for every choice of $\mathbf{o}$, which is true if and only if $\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})$ or $\tilde{\mathbf{M}}(\boldsymbol{\Sigma}, \mathbf{K})$ has full row-rank. Since $\tilde{\mathbf{M}}(\boldsymbol{\Sigma}, \mathbf{K})$ always has full row rank if $\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})$ does (because $\tilde{\mathbf{M}}(\boldsymbol{\Sigma}, \mathbf{K})$ is a submatrix of $\tilde{\mathbf{M}}(\boldsymbol{\Sigma}, \mathbf{K})$ with the same number of rows), we focus below on proving $\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})$ has full row rank. Tighter bounds can be obtained for the bias case by considering $\tilde{\mathbf{M}}(\boldsymbol{\Sigma}, \mathbf{K})$ directly, but they do not affect parameter-count asymptotics (or even constant multipliers).

Rank condition on $\boldsymbol{\Sigma}$ Interestingly, the above is true for generic $\mathbf{K}$ provided a simple rank condition on $\boldsymbol{\Sigma}$. We start with the following definitions.

Definition B.4.1. Given a set S , define a d -partition of S as a tuple of sets $\mathcal{I} = (I_1, \dots, I_d)$ with $I_1, \dots, I_d \subseteq [|S|]$ satisfying $I_i \cap I_j = \emptyset$ for all $i \neq j \in [d]$. Define a complete d -partition of S as a d partition also satisfying $\bigcup_{i \in [d]} I_i = S$.

Definition B.4.2. Let $I_1, \dots, I_d$ be a d -partition of $[|\mathbf{K}|]$ and let $\mathbf{a} \in \mathbb{R}^{|\mathbf{K}|}$. Define $\mathbf{K}(\mathbf{a}, I_1, \dots, I_d) \in \mathbb{R}^{|\mathbf{K}| \times d}$ according to the rule

$$\mathbf{K}(\mathbf{a}, I_1, \dots, I_d)[i, j] = \mathbf{a}[i] \mathbb{1}\{i \in I_j\}.$$

We abbreviate $\mathbf{K}(I_1, \dots, I_d) \equiv \mathbf{K}(\mathbf{1}_{|\mathbf{K}|}, I_1, \dots, I_d)$.

Next, we provide the following lemmas characterizing the rank of $\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})$ and $\tilde{\mathbf{M}}(\boldsymbol{\Sigma}, \mathbf{K})$.

Lemma B.4.3. Let $I_1, \dots, I_d$ be a d -partition of $[|\mathbf{K}|]$, pick any $\boldsymbol{\Sigma} \in \mathbb{R}^{h \times |\mathbf{K}|}$, and pick any $\mathbf{a} \in \mathbb{R}^{|\mathbf{K}|}$ with $\mathbf{a}[i] \neq 0$ for all $i \in [|\mathbf{K}|]$. Then

$$\text{rank}(\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K}(\alpha, I_1, \dots, I_d))) = \sum_{j=1}^d \text{rank}(\boldsymbol{\Sigma}[:, I_j]).$$

Proof. We define $\mathbf{K} := \mathbf{K}(\alpha, I_1, \dots, I_d)$ for notational simplicity.

The columns of $\mathbf{M}$ can be re-grouped to form d blocks of size $|\mathbf{K}| \times h$. Let $\mathbf{M}_j$ be the j -th new block, $j \in [d]$. This block contains all columns from $\mathbf{M}$ that were constructed using $\mathbf{K}[:, j]$ and can be written as $\mathbf{M}_j = \text{diag}(\mathbf{K}[:, j])\boldsymbol{\Sigma}^\top$.

The matrix $\text{diag}(\mathbf{K}[:, j])$ acts as a row-selector. It zeroes out all rows of $\boldsymbol{\Sigma}^\top$ except for those with indices in I_j . Thus, $\text{col}(\mathbf{M}_i) \perp \text{col}(\mathbf{M}_j)$ for all $i, j \in [d]$, so

$$\dim(\text{col}(\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K}))) = \dim\left(\bigoplus_{j=1}^d \text{col}(\mathbf{M}_j)\right) = \sum_{j=1}^d \text{rank}(\mathbf{M}_j).$$

Furthermore,

$$\begin{aligned}\text{rank}(\mathbf{M}_j) &= \text{rank}(\text{diag}(\mathbf{K}[:, j])\boldsymbol{\Sigma}^\top) \\ &= \text{rank}(\text{diag}(\mathbf{K}[I_j, j])\boldsymbol{\Sigma}^\top[I_j, :]) \\ &= \text{rank}(\boldsymbol{\Sigma}^\top[I_j, :]) \\ &= \text{rank}(\boldsymbol{\Sigma}[:, I_j]).\end{aligned}$$

Thus

$$\text{rank}(\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})) = \sum_{j=1}^d \text{rank}(\mathbf{M}_j) = \sum_{j=1}^d \text{rank}(\boldsymbol{\Sigma}[:, I_j]),$$

as desired. $\square$

Lemma B.4.4. *For generic $\mathbf{K}$, we have that*

$$\text{rank}(\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})) = \min_{S \subseteq [\mathbf{K}]} \left[|\mathbf{K}| - |S| + d \cdot \text{rank}(\boldsymbol{\Sigma}[:, S]) \right] \equiv R(\boldsymbol{\Sigma}). \quad (10)$$

More specifically, the set $\mathcal{K} = \{\mathbf{K} \mid \text{rank}(\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})) = R(\boldsymbol{\Sigma})\}$ is a non-empty Zariski open set (i.e. its complement is an algebraic set) and hence has full measure.

Proof. For the full proof, see Section B.10.1. A sketch of the proof is as follows.

We first show that $\mathcal{K}$ is a Zariski open set. We show this by demonstrating that the $\mathbf{K}$ contained in $\mathcal{K}$ are exactly those for which not all $R(\boldsymbol{\Sigma})$ th order minors of $\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})$ are 0.

Thus, we simply need to show that $\mathcal{K}$ is non-empty. Fortunately, by noting that Equation 10 matches the form of the Matroid Union Theorem (Oxley, 2011), we can use the Matroid Union Theorem to construct an explicit $\mathbf{K}$ contained in $\mathcal{K}$, thus completing the proof. $\square$

Lemma B.4.5. *The set $\mathcal{K} = \{\mathbf{K} \mid \text{rank}(\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})) = |\mathbf{K}|\}$ is a non-empty Zariski open set (and hence has full measure) if and only if*

$$d \cdot \text{rank}(\boldsymbol{\Sigma}[:, S]) \geq |S| \quad \forall S \subseteq [\mathbf{K}]. \quad (11)$$

Proof. ($\implies$) Follows immediately from Lemma B.4.4.

($\impliedby$) Conversely, suppose there exists a subset $S \subseteq [\mathbf{K}]$ such that

$$d \cdot \text{rank}(\boldsymbol{\Sigma}[:, S]) < |S|.$$

Then

$$R(\boldsymbol{\Sigma}) = \min_{T \subseteq [\mathbf{K}]} \left[|\mathbf{K}| - |T| + d \cdot \text{rank}(\boldsymbol{\Sigma}[:, T]) \right] \leq |\mathbf{K}| - |S| + d \cdot \text{rank}(\boldsymbol{\Sigma}[:, S]) < |\mathbf{K}|.$$

By Lemma B.4.4, there exists a non-empty Zariski open set $\mathcal{K}_0$ such that for all $\mathbf{K} \in \mathcal{K}_0$,

$$\text{rank}(\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})) = R(\boldsymbol{\Sigma}) < |\mathbf{K}|.$$

Therefore the full-rank locus

$$\mathcal{K}_{\text{full}} := \{\mathbf{K} : \text{rank}(\mathbf{M}(\boldsymbol{\Sigma}, \mathbf{K})) = |\mathbf{K}|\}$$

is contained in the complement of $\mathcal{K}_0$, which is a proper Zariski closed set. Hence $\mathcal{K}_{\text{full}}$ cannot be a non-empty Zariski open set. $\square$

Further, for analytic σ , such a $\boldsymbol{\Sigma}$ is easy to find. To show this, we first start with the following standard lemmas (proofs given for completeness):

Lemma B.4.6. *Let $f_1, \dots, f_r$ be linearly independent real-valued functions on some set S . Then there exist points $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(r)} \in S$ such that the $r \times r$ matrix $\mathbf{M} = (f_i(\mathbf{a}^{(j)}))_{1 \leq i, j \leq r}$ has rank r (equivalently, is invertible).*

Proof. See Section B.10.2. $\square$

Lemma B.4.7. *Let σ be a non-polynomial analytic function and define $f_\lambda(t) = \sigma(\lambda t)$. Further, define $\mathcal{S} = \text{span}\{f_\lambda \mid \lambda \in \mathbb{R}\}$. The dimension of $\mathcal{S}$ is infinite.*

Proof. See Section B.10.3. $\square$

Lemma B.4.8. *Given a non-polynomial analytic function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$, for generic $\mathbf{x} \in \mathbb{R}^{d_1}$ and $\mathbf{y} \in \mathbb{R}^{d_2}$, we have that*

$$\text{rank}(\sigma(\mathbf{xy}^\top)) = \min\{d_1, d_2\}. \quad (12)$$

More specifically, the set

$$\mathcal{S} = \left\{ (\mathbf{x}, \mathbf{y}) \mid \text{rank}(\sigma(\mathbf{xy}^\top)) = \min\{d_1, d_2\} \right\}$$

is the complement of a proper analytic subvariety of $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$.

Proof. We first show that the set $\mathcal{S}$ is the complement of an algebraic subvariety in a similar approach to the proof of Theorem B.4.4. Thus, all that remains is to show that $\mathcal{S}$ is non-empty.

Case 1, $d_1 \geq d_2$: By Theorem B.4.7 there exists a choice of $\mathbf{x} \in \mathbb{R}^{d_1}$ such that $\{\sigma(\mathbf{x}[i] \cdot y)\}_{i=1}^{d_1}$ are independent functions of y . Thus, by Theorem B.4.6, we can choose $\mathbf{y} \in \mathbb{R}^{d_2}$ such that the matrix $\sigma(\mathbf{xy}^\top)$ has rank $\min\{d_1, d_2\}$.

Case 2, $d_1 < d_2$: By Theorem B.4.7 there exists a choice of $\mathbf{y} \in \mathbb{R}^{d_2}$ such that $\{\sigma(x \cdot \mathbf{y}[i])\}_{i=1}^{d_2}$ are independent functions of x . Thus, by Theorem B.4.6, we can choose $\mathbf{x} \in \mathbb{R}^{d_1}$ such that the matrix $\sigma(\mathbf{xy}^\top)$ has rank $\min\{d_1, d_2\}$.

This demonstrates that $\mathcal{S}$ is nonempty, completing the proof. $\square$

The above lemma can be naturally generalized:

Lemma B.4.9. *Given a non-polynomial analytic function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$, for generic $\mathbf{x} \in \mathbb{R}^{d_1}$ and $\mathbf{y} \in \mathbb{R}^{d_2}$ we have that*

$$\text{rank}(\sigma(\mathbf{xy}^\top)[S_1, S_2]) = \min\{|S_1|, |S_2|\} \quad \forall S_1 \subseteq [d_1], S_2 \subseteq [d_2]. \quad (13)$$

More specifically, the set

$$\mathcal{S} = \left\{ (\mathbf{x}, \mathbf{y}) \mid \text{rank}(\sigma(\mathbf{xy}^\top)[S_1, S_2]) = \min\{|S_1|, |S_2|\} \quad \forall S_1 \subseteq [d_1], S_2 \subseteq [d_2] \right\}$$

is the complement of a proper analytic subvariety of $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$.

Proof. See Section B.10.4. $\square$

Finally, we combine Theorem B.4.3 and Theorem B.4.9 to obtain the following characterization for when $\mathbf{M}$ has full row rank.

Lemma B.4.10 (Full-row-rank condition for non-polynomial analytic activations). *Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be a non-polynomial analytic function. If $dh \geq |\mathbf{K}|$, then for generic $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$ and $\mathbf{G} \in \mathbb{R}^{h \times d}$, the matrix*

$$\mathbf{M}(\sigma(\mathbf{GK}^\top), \mathbf{K}) \in \mathbb{R}^{|\mathbf{K}| \times (dh)}$$

has full row rank $|\mathbf{K}|$. The tuples for which full row rank fails form a proper analytic subvariety of the ambient parameter space.

Proof. A more careful combination of the proofs of Theorems B.4.4, B.4.5 and B.4.9. Full proof given in Section B.10.5. $\square$

Theorem B.4.10 is the last piece we need to prove the full encoder gadget theorem:

Theorem B.4.11. *Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be a non-polynomial analytic activation. If $dh \geq |\mathbf{K}|$ and $\text{rank}[\sigma] \geq h$, then following Algorithm 7 with BIAS either TRUE or FALSE produces an MLP $\text{enc}(\mathbf{x}) := \mathbf{1}_h(\sigma(\mathbf{Gx}) \odot (\mathbf{Ax}))$ which satisfies $\text{enc}(\mathbf{k}_i) = o_i$ for all $i \in [|\mathbf{K}|]$.*

Proof. By Lemma B.4.10, under the stated conditions (no-bias or biased case) and for generic draws of $\mathbf{G}$ (setting $\mathbf{b}_G = \mathbf{0}_h$), the corresponding matrix $\mathbf{M}(\Sigma, \mathbf{K})$ or $\widetilde{\mathbf{M}}(\Sigma, \mathbf{K})$ have full row rank. Hence, for any target vector $\mathbf{o}$, the linear system in $\text{vec}(\mathbf{A})$ (or $\text{vec}(\mathbf{A}, \mathbf{b}_A, \mathbf{b}_E)$) is solvable, and the parameters returned by Algorithm 7 satisfy $\text{enc}(\mathbf{k}_i) = o_i$ for all $i \in [|\mathbf{K}|]$. $\square$

B.4.3 Non-Gated Encoders Reduce to Gated Encoders

In Appendix B.4, it is shown that these results extend to non-gated MLPs (up to an arbitrarily small δ error) by implementing a neural tangent kernel (NTK) approximation similar to Nichani et al. (2024). Interestingly, when this generalization is applied to ReLU MLPs, a construction is obtained which generalizes that from Bubeck et al. (2020) while utilizing up to $4 \times$ fewer parameters¹². Additionally, while it is possible to use the encoder construction from Bubeck et al. (2020) directly in the full fact-storing construction, we found that the resulting MLPs are not usable by transformers, whereas the MLPs constructed herein are.

The construction, detailed in Algorithm 8, approximates a gated MLP that uses the activation's derivative, σ' , with a standard non-gated MLP that uses σ . This is achieved in three steps:

¹²In fact, this generalization of Bubeck et al. (2020) matches the degrees-of-freedom-based parameter count lower bound up to lower order terms.

1. **Construct a “Derivative” Gadget:** First, Algorithm 8 (Line 1) calls Algorithm 7 to find the parameters of an intermediate gated gadget. This call uses a hidden size of $h/2$ (where h is the hidden size required by Algorithm 8) and replaces the activation σ with its derivative, $\frac{d\sigma}{dx}$. Let the parameters returned by this call be $(\mathbf{G}_{\text{deriv}}, \mathbf{b}_{G,\text{deriv}}, \mathbf{A}_{\text{deriv}}, \mathbf{b}_{A,\text{deriv}}, b_E)$ where $\mathbf{G}_{\text{deriv}}, \mathbf{A}_{\text{deriv}} \in \mathbb{R}^{(h/2) \times d}$ and $\mathbf{b}_{G,\text{deriv}}, \mathbf{b}_{A,\text{deriv}} \in \mathbb{R}^{h/2}$. The resulting encoder (which Algorithm 8 temporarily calls $\mathbf{enc}(\mathbf{x})$ on Line 1) is:

$$\mathbf{enc}_{\text{deriv}}(\mathbf{x}) = \mathbf{1}_{h/2}^\top (\sigma'(\mathbf{G}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{G,\text{deriv}}) \odot (\mathbf{A}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{A,\text{deriv}})) + b_E$$

This $\mathbf{enc}_{\text{deriv}}$ is constructed to map $\mathbf{k}_i$ to the target output o_i for all $i \in [|\mathbf{K}|]$.

2. **Find Approximation Parameter ϵ :** Second (Lines 3-6), the algorithm finds a small vector $\epsilon \in \mathbb{R}^{h/2}$. This ϵ is chosen such that a central difference approximation of $\mathbf{enc}_{\text{deriv}}$ (using σ) is within a tolerance δ of the target values $o_i \approx \mathbf{enc}_{\text{deriv}}(\mathbf{k}_i)$ for all keys $\mathbf{k}_i$.
3. **Construct Final Non-Gated Gadget:** Finally (Lines 8-12), the algorithm uses the intermediate parameters and ϵ to define the parameters of the *final* non-gated MLP, which has the target hidden size $h = 2 \times (h/2)$. The parameters for the returned $\mathbf{enc}(\mathbf{x})$ are:

$$\begin{aligned} \mathbf{A} &:= \begin{bmatrix} \mathbf{G}_{\text{deriv}} + \text{diag}(\epsilon)\mathbf{A}_{\text{deriv}} \\ \mathbf{G}_{\text{deriv}} - \text{diag}(\epsilon)\mathbf{A}_{\text{deriv}} \end{bmatrix} \in \mathbb{R}^{h \times d} \\ \mathbf{b}_A &:= \begin{bmatrix} \mathbf{b}_{G,\text{deriv}} + \epsilon \odot \mathbf{b}_{A,\text{deriv}} \\ \mathbf{b}_{G,\text{deriv}} - \epsilon \odot \mathbf{b}_{A,\text{deriv}} \end{bmatrix} \in \mathbb{R}^h \\ \mathbf{E} &:= \begin{bmatrix} \frac{1}{2}\epsilon^{-1} & -\frac{1}{2}\epsilon^{-1} \end{bmatrix} \in \mathbb{R}^{1 \times h} \end{aligned}$$

The final returned encoder is $\mathbf{enc}(\mathbf{x}) := \mathbf{E}\sigma(\mathbf{A}\mathbf{x} + \mathbf{b}_A) + b_E$, which by construction approximates the target outputs $\mathbf{o}$.

Intuitively, the final non-gated gadget implements a finite-difference approximation of the “derivative” gadget. Plugging in the definitions of $\mathbf{A}, \mathbf{b}_A, \mathbf{E}$, we obtain for any $\mathbf{x}$:

$$\mathbf{enc}(\mathbf{x}) = \sum_{r=1}^{h/2} \frac{1}{2\epsilon_r} \left[\sigma(g_r(\mathbf{x}) + \epsilon_r a_r(\mathbf{x})) - \sigma(g_r(\mathbf{x}) - \epsilon_r a_r(\mathbf{x})) \right] + b_E,$$

where $g_r(\mathbf{x})$ and $a_r(\mathbf{x})$ are the r -th coordinates of $\mathbf{G}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{G,\text{deriv}}$ and $\mathbf{A}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{A,\text{deriv}}$, respectively. By Taylor expansion (or the mean value theorem), each bracket implements

$$\frac{\sigma(g_r + \epsilon_r a_r) - \sigma(g_r - \epsilon_r a_r)}{2\epsilon_r} \approx \sigma'(g_r) a_r,$$

so $\mathbf{enc}(\mathbf{x})$ approximates

$$\mathbf{enc}_{\text{deriv}}(\mathbf{x}) = \sum_{r=1}^{h/2} \sigma'(g_r(\mathbf{x})) a_r(\mathbf{x}) + b_E.$$

By construction of $\epsilon \in \bigcap_i S_i$, this approximation error is at most δ on all keys $\mathbf{k}_i$, so the returned non-gated encoder matches the desired targets up to tolerance δ .

Special Case: ReLU Activation Here, we show the generality of our framework by showing that (Bubeck et al., 2020) is a special case. In the special case where the activation function is the ReLU function, the derivative $\sigma'(\mathbf{x}) = \mathbf{1}_{\{\mathbf{x} > 0\}}$ is used to construct the intermediate gadget. The final encoder returned by Algorithm 8 (Line 12) implements the central difference approximation:

$$\mathbf{enc}(\mathbf{x}) = \begin{bmatrix} \frac{1}{2}\epsilon^{-1} & -\frac{1}{2}\epsilon^{-1} \end{bmatrix} \text{ReLU} \left(\begin{bmatrix} \mathbf{G}_{\text{deriv}} + \text{diag}(\epsilon)\mathbf{A}_{\text{deriv}} \\ \mathbf{G}_{\text{deriv}} - \text{diag}(\epsilon)\mathbf{A}_{\text{deriv}} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{b}_{G,\text{deriv}} + \epsilon \odot \mathbf{b}_{A,\text{deriv}} \\ \mathbf{b}_{G,\text{deriv}} - \epsilon \odot \mathbf{b}_{A,\text{deriv}} \end{bmatrix} \right) + b_E.$$

If a forward difference approximation were used instead (as in Bubeck et al. (2020)), the form would be:

$$\text{MLP}(\mathbf{x}) = \mathbf{1}_{h/2}^\top \left(\text{diag}(\epsilon)^{-1} \left(\text{ReLU}(\mathbf{G}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{G,\text{deriv}} + \text{diag}(\epsilon)(\mathbf{A}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{A,\text{deriv}})) - \text{ReLU}(\mathbf{G}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{G,\text{deriv}}) \right) \right) + b_E.$$

The portion inside the outer brackets is the derivative neuron from Bubeck et al. (2020).

Note that one can also pull the $\text{diag}(\epsilon)^{-1}$ term inside the brackets and define λ such that $\epsilon \odot \lambda = \mathbf{1}$ (element-wise) to get a “Lagrangian formulation”:

$$\text{MLP}(\mathbf{x}) = \mathbf{1}_{h/2}^\top \left(\text{ReLU}(\text{diag}(\lambda)(\mathbf{G}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{G,\text{deriv}}) + (\mathbf{A}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{A,\text{deriv}})) - \text{diag}(\lambda) \text{ReLU}(\mathbf{G}_{\text{deriv}}\mathbf{x} + \mathbf{b}_{G,\text{deriv}}) \right) + b_E.$$

The ReLU case possesses the property that this forward difference approximation is exactly equal to the corresponding gated MLP on a set of points $\mathbf{x}_i$ as long as $\lambda \geq -\min_i \frac{\mathbf{A}_{\text{deriv}}\mathbf{x}_i + \mathbf{b}_{A,\text{deriv}}}{\mathbf{G}_{\text{deriv}}\mathbf{x}_i + \mathbf{b}_{G,\text{deriv}}}$ (element-wise). In particular, if $\min_i \frac{\mathbf{A}_{\text{deriv}}\mathbf{x}_i + \mathbf{b}_{A,\text{deriv}}}{\mathbf{G}_{\text{deriv}}\mathbf{x}_i + \mathbf{b}_{G,\text{deriv}}} \geq 0$, then $\lambda = \mathbf{0}$ can be set to achieve the exact result, which avoids extra neurons. In contrast, the Bubeck et al. (2020) derivative neuron formulation would diverge in this case.

B.5 Additional Details for Section 4.2

We prove lower bounds on ρ and detail our decoding construction. We use a slightly more practical definition of ρ as follows when doing computations. However, since $\rho \geq \rho_{\min}$ by definition, similar statements hold for ρ .

Definition B.5.1. For vectors $\mathbf{v}_1, \dots, \mathbf{v}_{|\mathbf{K}|} \in \mathbb{R}^d$ and $\mathbf{u}_1, \dots, \mathbf{u}_{|\mathbf{K}|} \in \mathbb{R}^d$, we define $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_{|\mathbf{K}|}]^\top \in \mathbb{R}^{|\mathbf{K}| \times d}$ and $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{|\mathbf{K}|}]^\top \in \mathbb{R}^{|\mathbf{K}| \times d}$. Let

$$\rho_{\min}(\mathbf{V}, \mathbf{U}) = \min_i \min_{j \neq i} \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|}$$

For ease of notation, we often write $\rho_{\min} := \rho_{\min}(\mathbf{V}, \mathbf{U})$. Occasionally, we refer to the set $\{\mathbf{v}_i\}_{i=1}^{|\mathbf{K}|}$ as our set of output embeddings, and the set $\{\mathbf{u}_i\}_{i=1}^{|\mathbf{K}|}$ as our set of auxiliary directions.

We now prove our full construction. In this case, we have that $\rho(\mathbf{V})$ as defined in Section 4.2 satisfies $\rho(\mathbf{V}) \geq \rho_{\min}(\mathbf{V}, \mathbf{U})$.

Theorem B.5.2. Assume $\mathbf{v}_1, \dots, \mathbf{v}_{|\mathbf{K}|} \stackrel{i.i.d.}{\sim} \text{Unif}(\mathbb{S}^{d-1})$ with $d \geq 2$ and for simplicity set¹³ $\mathbf{u}_i = \mathbf{v}_i$ for all i . Then, with probability at least $1 - \delta$,

$$\rho_{\min} \geq \sqrt{\frac{1 - \sqrt{\frac{2}{d} \ln \frac{\binom{|\mathbf{K}|}{2}}{\delta}}}{2}}.$$

Proof. See Section B.10.9 □

Theorem B.5.3. Let $\mathbf{D} \in \mathbb{R}^{m \times d}$ have i.i.d $\mathcal{N}(0, 1)$ entries. Set $\mathbf{M} := \frac{1}{m} \mathbf{D}^\top$ and, for each $i \in [|\mathbf{K}|]$, define $\mathbf{H}[i] := \mathbf{D} \mathbf{u}_i \in \mathbb{R}^m$. Let $\rho_{\min} = \rho_{\min}(\mathbf{V}, \mathbf{U})$ be as in Theorem B.5.1, and fix a failure probability $\delta \in (0, 1)$. If

$$m \geq \frac{32}{\rho_{\min}^2} \ln \frac{4|\mathbf{K}|(|\mathbf{K}| - 1)}{\delta},$$

and $\rho_{\min} > 0$, then with probability at least $1 - \delta$ the following holds simultaneously for all $i \neq j$:

$$\langle \mathbf{v}_i, \mathbf{M} \mathbf{H}[i] \rangle - \langle \mathbf{v}_j, \mathbf{M} \mathbf{H}[i] \rangle \geq \frac{\rho_{\min}}{2} \|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\| > 0$$

Proof. See Section B.10.7 □

Corollary B.5.4. For $\delta = \frac{1}{\text{poly } d}$, $|\mathbf{K}| = \text{poly}(d)$, large enough d , and for output embeddings $\{\mathbf{v}_i\}_{i=1}^{|\mathbf{K}|}$ as in Theorem B.5.2, the set of output embeddings are softmax decodable with probability $1 - \delta$ as long as the conditions in Theorem B.5.3 on m hold.

Proof. By Theorem B.5.2, $\rho_{\min} \geq \gamma$ for some γ with $\gamma \rightarrow \frac{1}{\sqrt{2}}$ as $d \rightarrow \infty$. Hence, for all large enough d , there exists an absolute positive constant γ^* such that $\rho_{\min} \geq \gamma^*$ with probability $1 - \delta$. Thus, we apply Theorem B.2.2 and Theorem B.5.3 to decode the embeddings. □

¹³One may wonder why we can set $\mathbf{u}_i = \mathbf{v}_i$ in this step. The reason is that it simplifies the proof and shows existence of a lower bound on $\rho_{\min}$. However, there may be a better choice of $\mathbf{u}_i$ which yields a tighter bound.

In the following theorem, we will need the sub-gaussian norm $\|\cdot\|_{\psi_2}$:

$$\|\mathbf{X}\|_{\psi_2} := \inf\{t > 0 : \mathbb{E}[\exp(\mathbf{X}^2/t^2)] \leq 2\}$$

Theorem B.5.5. *Let $\mathbf{v}_i = (\xi_{i1}, \dots, \xi_{id}) \in \mathbb{R}^d$ for $i = 1, \dots, |\mathbf{K}|$, where the coordinates are i.i.d. sub-gaussian with*

$$\mathbb{E}[\xi_{ik}] = 0, \quad \mathbb{E}[\xi_{ik}^2] = \frac{1}{d}, \quad \|\xi_{ik}\|_{\psi_2} \leq \frac{K}{\sqrt{d}}.$$

Set $\mathbf{u}_i := \mathbf{v}_i / \|\mathbf{v}_i\|$ and let $c_B = \frac{1}{2(2e-1)}$. Then for every $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\rho_{\min} \geq \frac{1 - \varepsilon_{|\mathbf{K}|} - t_{|\mathbf{K}|}}{2(1 + \varepsilon_{|\mathbf{K}|})},$$

where

$$\varepsilon_{|\mathbf{K}|} := (K^2 + \frac{1}{\ln 2}) \max \left(\sqrt{\frac{1}{c_B d} \ln \frac{4|\mathbf{K}|}{\delta}}, \frac{1}{c_B d} \ln \frac{4|\mathbf{K}|}{\delta} \right)$$

$$t_{|\mathbf{K}|} := K \sqrt{\frac{2 \ln 2}{d} \ln \frac{4|\mathbf{K}|(|\mathbf{K}| - 1)}{\delta}}.$$

Proof. See Section B.10.10 □

Corollary B.5.6. *For $\delta = \frac{1}{\text{poly } d}$, $|\mathbf{K}| = \text{poly}(d)$, large enough d , and for output embeddings $\{\mathbf{v}_i\}_{i=1}^{|\mathbf{K}|}$ as in Theorem B.5.5, the set of output embeddings are softmax decodable with probability $1 - \delta$ as long as the conditions in Theorem B.5.3 on m hold.*

Proof. By Theorem B.5.5, $\rho_{\min} \geq \gamma$ for some γ with $\gamma \rightarrow 1/2$ as $d \rightarrow \infty$. Hence, for all large enough d , there exists an absolute positive constant γ^* such that $\rho_{\min} \geq \gamma^*$ with probability $1 - \delta$. Thus, we apply Section B.10.9 to decode the embeddings. □

B.5.1 Relation of ρ to Coherence

Throughout this section, we define coherence in the traditional sense as follows.

Definition B.5.7 (Coherence). *For unit-norm row vectors $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_{|\mathbf{K}|}]^\top \in \mathbb{R}^{|\mathbf{K}| \times d}$,*

$$\mu(\mathbf{V}) := \max_{i \neq j} |\langle \mathbf{v}_i, \mathbf{v}_j \rangle|.$$

Given the definition of $\rho(\mathbf{V}, \mathbf{U})$, which doesn't have similar absolute values around the inner product term, we could have defined the coherence as $\mu(\mathbf{V}) = \max_{i \neq j} \langle \mathbf{v}_i, \mathbf{v}_j \rangle$. The results of this section hold using either definition of $\mu(\mathbf{V})$.

Lemma B.5.8 (Lower bound via absolute coherence). *Let $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_{|\mathbf{K}|}]^\top \in \mathbb{R}^{|\mathbf{K}| \times d}$ with $\|\mathbf{v}_i\|_2 = 1$ for all i . By Theorem B.5.7, then*

$$\rho(\mathbf{V}) \geq \frac{1}{\sqrt{2}} \sqrt{1 - \mu(\mathbf{V})}.$$

Proof. Fix i and set $\mathbf{u}_i := \mathbf{v}_i$. For any $j \neq i$,

$$\frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\|_2} = \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{v}_i \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\|_2} = \frac{1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle}{\sqrt{\|\mathbf{v}_i\|_2^2 + \|\mathbf{v}_j\|_2^2 - 2\langle \mathbf{v}_i, \mathbf{v}_j \rangle}} = \frac{1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle}{\sqrt{2 - 2\langle \mathbf{v}_i, \mathbf{v}_j \rangle}} = \frac{1}{\sqrt{2}} \sqrt{1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle}.$$

Taking the minimum over $j \neq i$ and then over i yields

$$\rho(\mathbf{V}) \geq \frac{1}{\sqrt{2}} \min_{i \neq j} \sqrt{1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle}.$$

Since for every $i \neq j$ we have $\langle \mathbf{v}_i, \mathbf{v}_j \rangle \leq |\langle \mathbf{v}_i, \mathbf{v}_j \rangle| \leq \mu(\mathbf{V})$ and $a \mapsto \sqrt{1 - a}$ is decreasing on $(-\infty, 1]$, it follows that

$$\min_{i \neq j} \sqrt{1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle} \geq \sqrt{1 - \mu(\mathbf{V})}.$$

Therefore $\rho(\mathbf{V}) \geq \frac{1}{\sqrt{2}} \sqrt{1 - \mu(\mathbf{V})}$, as claimed. □

Given this lower bound on $\rho(\mathbf{U}, \mathbf{V})$ in terms of $1 - \mu(\mathbf{V})$, one might wonder if there exists a similar upper bound. Specifically, does there exist some constant $\beta > 0$ such that

$$\rho(\mathbf{V}) \leq O((1 - \mu(\mathbf{V}))^\beta)$$

In the following proposition, we provide a counter example which shows that this is false. Hence, $\rho(\mathbf{V})$ and $1 - \mu(\mathbf{V})$ are fundamentally different quantities.

Lemma B.5.9. *Fix a constant integer $p \geq 2$. Then, for large enough d , there exist unit-norm row vectors $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_{|\mathbf{K}|}]^\top \in \mathbb{R}^{|\mathbf{K}| \times d}$ such that*

$$\mu(\mathbf{V}) = 1 - o(1) \quad \text{but} \quad \rho(\mathbf{V}) \geq \sqrt{\frac{1/p}{2}} > 0.$$

Proof. Choose a dimension $d_0 = o(d)$ and construct $\mathbf{V}_0 = [\mathbf{v}_1^{(0)}, \dots, \mathbf{v}_{|\mathbf{K}|}^{(0)}]^\top \in \mathbb{R}^{|\mathbf{K}| \times d_0}$ as follows. Choose each row $\mathbf{v}_i^{(0)}$ to be the the p -hot encoding of the row index. Thus each row has exactly p non-zero entries, each equal to $1/\sqrt{p}$ and pairwise the non-zero entries overlap in at most $p - 1$ coordinates. Then for $i \neq j$,

$$|\langle \mathbf{v}_i^{(0)}, \mathbf{v}_j^{(0)} \rangle| \leq 1 - \frac{1}{p} \quad \implies \quad \mu(\mathbf{V}_0) \leq 1 - \frac{1}{p} < 1.$$

Let $\mathbf{u}_i^{(0)} := \mathbf{v}_i^{(0)}$. Then

$$\left\langle \frac{\mathbf{v}_i^{(0)} - \mathbf{v}_j^{(0)}}{\|\mathbf{v}_i^{(0)} - \mathbf{v}_j^{(0)}\|_2}, \mathbf{u}_i^{(0)} \right\rangle = \frac{1 - \langle \mathbf{v}_i^{(0)}, \mathbf{v}_j^{(0)} \rangle}{\sqrt{2 - 2\langle \mathbf{v}_i^{(0)}, \mathbf{v}_j^{(0)} \rangle}} = \sqrt{\frac{1 - \langle \mathbf{v}_i^{(0)}, \mathbf{v}_j^{(0)} \rangle}{2}} \geq \sqrt{\frac{1 - 1/p}{2}}.$$

Minimizing over all $i \neq j$ shows

$$\rho(\mathbf{V}_0) \geq \gamma_0 := \sqrt{\frac{1/p}{2}} > 0.$$

We now pad each vector with ones. Let $t := d - d_0$ and define

$$\hat{\mathbf{v}}_i := (\mathbf{v}_i^{(0)}, \mathbf{1}_t) \in \mathbb{R}^d, \quad \mathbf{v}_i := \frac{\hat{\mathbf{v}}_i}{\|\hat{\mathbf{v}}_i\|_2} = \frac{(\mathbf{v}_i^{(0)}, \mathbf{1}_t)}{\sqrt{1 + t}}.$$

where here $(\mathbf{v}_i^{(0)}, \mathbf{1}_t)$ denotes the lengthwise concatenation of $\mathbf{v}_i^{(0)}$ and $\mathbf{1}_t$ where $\mathbf{1}_t$ is a vector of length t of ones. Then for $i \neq j$,

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \frac{\langle \mathbf{v}_i^{(0)}, \mathbf{v}_j^{(0)} \rangle + t}{1 + t} = 1 - \frac{1 - \langle \mathbf{v}_i^{(0)}, \mathbf{v}_j^{(0)} \rangle}{1 + t} \geq 1 - \frac{1}{1 + t} \geq 0,$$

hence

$$\mu(\mathbf{V}) = \max_{i \neq j} |\langle \mathbf{v}_i, \mathbf{v}_j \rangle| \geq 1 - \frac{1}{1 + t} = 1 - o(1),$$

where the final equality holds since $t \rightarrow \infty$ increases $\frac{1}{1+t} \rightarrow 0$.

On the other hand, if we set $\mathbf{u}_i = (\mathbf{u}_i^{(0)}, \mathbf{0}_t)$, where $\mathbf{u}_i^{(0)}$ are picked such that $\rho(\mathbf{V}^{(0)}, \mathbf{U}^{(0)}) = \rho(\mathbf{V}^{(0)})$ and $\mathbf{0}_t$ is a vector of length t of all zeros, for any $i \neq j$,

$$\rho(\mathbf{V}, \mathbf{U}) = \left\langle \frac{\mathbf{v}_i - \mathbf{v}_j}{\|\mathbf{v}_i - \mathbf{v}_j\|_2}, (\mathbf{u}_i^{(0)}, \mathbf{0}_t) \right\rangle = \left\langle \frac{(\mathbf{v}_i^{(0)}, \mathbf{1}_t) - (\mathbf{v}_j^{(0)}, \mathbf{1}_t)}{\|(\mathbf{v}_i^{(0)}, \mathbf{1}_t) - (\mathbf{v}_j^{(0)}, \mathbf{1}_t)\|_2}, (\mathbf{u}_i^{(0)}, \mathbf{0}_t) \right\rangle = \left\langle \frac{\mathbf{v}_i^{(0)} - \mathbf{v}_j^{(0)}}{\|\mathbf{v}_i^{(0)} - \mathbf{v}_j^{(0)}\|_2}, \mathbf{u}_i^{(0)} \right\rangle = \rho(\mathbf{V}_0).$$

Combining the bounds yields $\mu(\mathbf{V}) = 1 - o(1)$ while $\rho(\mathbf{V}) \geq \sqrt{1/2p} > 0$, completing the proof. $\square$

B.6 Additional Details for Section 4.3

Theorem B.6.1 (Full Construction). *For any fact set f , generic key embeddings $\mathbf{K}$, and value embeddings $\mathbf{V}$ with $\rho(\mathbf{V}) > 0$, construct $\mathbf{enc}$ as in Section 4.1 and construct $\mathbf{dec}$ as in Section 4.2. Then the fact MLP*

$$\mathbf{g}(\mathbf{x}) = \mathbf{dec}(\mathbf{enc}(\mathbf{x})) = \mathbf{D}\mathbf{E}(\sigma(\mathbf{G}\mathbf{x}) \odot (\mathbf{A}\mathbf{x}))$$

stores f given $\mathbf{K}$ and $\mathbf{V}$, and has fact-storage cost

$$\Theta([\rho(\mathbf{V})]^{-2} |\mathbf{K}| \log |\mathbf{V}|).$$

Proof. By Theorem B.5.3, for any $\rho(\mathbf{V}) > 0$ there exist a compressed dimension

$$m = \Theta([\rho(\mathbf{V})]^{-2} \log |\mathbf{V}|)$$

and a linear decoder $\mathbf{dec}(\mathbf{x}) = \mathbf{D}\mathbf{x}$ together with compressed codes $\mathbf{C} = \{\mathbf{c}_i\}_{i=1}^{|\mathbf{V}|}$ such that the dot-product decoding condition

$$\langle \mathbf{v}_i, \mathbf{dec}(\mathbf{c}_i) \rangle > \langle \mathbf{v}_j, \mathbf{dec}(\mathbf{c}_i) \rangle \quad \forall i \neq j$$

holds. Fix such a $(\mathbf{C}, \mathbf{D})$.

Given these compressed codes, apply Theorem B.4.11 coordinate-wise: for each $j \in [m]$, with $|\mathbf{K}|$ generic inputs and targets $\{\mathbf{c}_{f(i),j}\}_{i=1}^{|\mathbf{K}|}$, the theorem guarantees a scalar-output gated encoder gadget that fits these values exactly. Stacking the m gadgets as in the encoder construction yields $\mathbf{enc}$ with

$$\mathbf{enc}(\mathbf{k}_i) = \mathbf{c}_{f(i)} \quad \forall i,$$

and total encoder parameter count $\Theta(m|\mathbf{K}|)$.

The composed MLP $\mathbf{g} = \mathbf{dec} \circ \mathbf{enc}$ thus satisfies

$$\mathbf{g}(\mathbf{k}_i) = \mathbf{dec}(\mathbf{enc}(\mathbf{k}_i)) = \mathbf{dec}(\mathbf{c}_{f(i)}),$$

which decodes (under dot products with $\mathbf{V}$) to $\mathbf{v}_{f(i)}$ by the property of $\mathbf{dec}$ and $\mathbf{C}$. Hence $\mathbf{g}$ stores f . Its parameter count is

$$\Theta(m|\mathbf{K}|) = \Theta([\rho(\mathbf{V})]^{-2} |\mathbf{K}| \log |\mathbf{V}|),$$

as claimed. $\square$

As it turns out, we may also prove a similar theorem using the result from Bubeck et al. (2020) as follows:

Theorem B.6.2 (Full construction). *Let $\mathbf{K} = \{\mathbf{k}_i\}_{i=1}^{|\mathbf{K}|} \subset \mathbb{R}^d$ be generic. Let $\mathbf{V} = \{\mathbf{v}_j\}_{j=1}^{|\mathbf{V}|} \subset \mathbb{R}^d$ with $\rho(\mathbf{V}) > 0$, and fix $f : [|\mathbf{K}|] \rightarrow [|\mathbf{V}|]$ and $\delta \in (0, 1)$. Let $\mathbf{U} = \{\mathbf{u}_j\}_{j=1}^{|\mathbf{V}|} \subset \mathbb{R}^d$. Additionally, set*

$$m \geq \frac{32}{\rho_{\min}(\mathbf{V}, \mathbf{U})^2} \ln \frac{4|\mathbf{V}|(|\mathbf{V}| - 1)}{\delta}, \quad \mathbf{G} \sim \mathcal{N}(0, 1)^{m \times d}, \quad \mathbf{M} := \frac{1}{m} \mathbf{G}^\top \in \mathbb{R}^{d \times m}.$$

where each coordinate $\mathbf{G}_{\ell,k}$ is sampled i.i.d from $\mathcal{N}(0, 1)$. Then, with probability at least $1 - \delta$ over $\mathbf{G}$, there exist $\mathbf{A} \in \mathbb{R}^{\tilde{m} \times d}$ and $\mathbf{b} \in \mathbb{R}^{\tilde{m}}$ with $\tilde{m} = 4m \lceil |\mathbf{K}|/d \rceil$ such that the one-hidden-layer ReLU network

$$\mathbf{V}^\top \mathbf{M} \text{ReLU}(\mathbf{A}\mathbf{x} + \mathbf{b}) \in \mathbb{R}^{|\mathbf{V}|}$$

achieves for all i, j such that $j \neq f(i)$:

$$\left\langle \mathbf{v}_{f(i)}, \mathbf{M} \text{ReLU}(\mathbf{A}\mathbf{k}_i + \mathbf{b}) \right\rangle - \left\langle \mathbf{v}_j, \mathbf{M} \text{ReLU}(\mathbf{A}\mathbf{k}_i + \mathbf{b}) \right\rangle \geq \frac{\rho_{\min}(\mathbf{V}, \mathbf{U})}{2} \|\mathbf{v}_{f(i)} - \mathbf{v}_j\| \|\mathbf{u}_{f(i)}\|$$

The number of trainable parameters that scale with $|\mathbf{K}|$ (the fact-storage cost) is $\Theta(m|\mathbf{K}|) = \Theta(\rho_{\min}(\mathbf{V}, \mathbf{U})^{-2} |\mathbf{K}| \log |\mathbf{V}|)$.

Proof. Define the m -dimensional codes $\mathbf{c}_j := \mathbf{G} \mathbf{u}_j \in \mathbb{R}^m$ for $j \in [|\mathbf{V}|]$. By Theorem B.5.3, the stated lower bound on m ensures that, with probability at least $1 - \delta$, for all i and all $j \neq i$,

$$\langle \mathbf{v}_i, \mathbf{M} \mathbf{c}_i \rangle - \langle \mathbf{v}_j, \mathbf{M} \mathbf{c}_i \rangle \geq \frac{\rho_{\min}}{2} \|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\| > 0. \quad (14)$$

Note that in the above, $\mathbf{c}_i$ are defined exactly as $\mathbf{H}[i]$ in Theorem B.5.3.

Apply Theorem B.1.1 *coordinatewise* to the dataset $\{(\mathbf{k}_i, (\mathbf{c}_{f(i)})_t)\}_i$ for each $t \in [m]$: stacking the m constructions produced by Theorem B.1.1 yields a ReLU map with width $\tilde{m} = 4m \lceil |\mathbf{K}|/d \rceil$ and parameters $\mathbf{A} \in \mathbb{R}^{\tilde{m} \times d}$, $\mathbf{b} \in \mathbb{R}^{\tilde{m}}$, together with a fixed matrix $\mathbf{E} \in \mathbb{R}^{m \times \tilde{m}}$, such that

$$\mathbf{E} \text{ReLU}(\mathbf{A}\mathbf{k}_i + \mathbf{b}) = \mathbf{c}_{f(i)} \quad \text{for all } i.$$

Now set

$$g(\mathbf{x}) := \mathbf{M} \mathbf{E} \text{ReLU}(\mathbf{A}\mathbf{x} + \mathbf{b}).$$

For each $\mathbf{k}_i$ we have $\mathbf{M} \mathbf{E} \text{ReLU}(\mathbf{A}\mathbf{k}_i + \mathbf{b}) = \mathbf{M} \mathbf{c}_{f(i)}$, so the margin at $\mathbf{k}_i$ equals the left-hand side of equation 14 with $i \mapsto f(i)$ (i.e., g stores f). Finally, only $(\mathbf{A}, \mathbf{b})$ scale with $|\mathbf{K}|$, giving the claimed $\Theta(m|\mathbf{K}|)$ fact-storage cost; substituting the bound on m finishes the proof. $\square$

B.7 Additional Details for Section 3

We provide theoretical results on embeddings and decodability.

Theorem B.7.1 (Affine invariance for 1-hidden-layer MLP with keys/values). *Consider a fact set $f : [F] \rightarrow [F]$, key embeddings $\mathbf{K} = \{\mathbf{k}_i\}_{i=1}^F \subset \mathbb{R}^d$, and value embeddings $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^F \subset \mathbb{R}^d$. Assume there exist $\mathbf{A} \in \mathbb{R}^{m \times d}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{B} \in \mathbb{R}^{d \times m}$ such that*

$$\langle \mathbf{v}_{f(i)} - \mathbf{v}_j, \mathbf{B} \text{ReLU}(\mathbf{A}\mathbf{k}_i + \mathbf{b}) \rangle > 0 \quad \text{for all } i \in [F], j \neq f(i). \quad (15)$$

Then for any affine transformation¹⁴ of the key and value embeddings:

$$\tilde{\mathbf{k}}_i = \mathbf{T}_k \mathbf{k}_i + \mathbf{c}_k, \quad \mathbf{T}_k \in \text{GL}(d), \mathbf{c}_k \in \mathbb{R}^d, \quad \tilde{\mathbf{v}}_i = \mathbf{T}_v \mathbf{v}_i + \mathbf{c}_v, \quad \mathbf{T}_v \in \text{GL}(d), \mathbf{c}_v \in \mathbb{R}^d,$$

there exist $\mathbf{A}' \in \mathbb{R}^{m \times d}$, $\mathbf{b}' \in \mathbb{R}^m$, $\mathbf{B}' \in \mathbb{R}^{d \times m}$ such that

$$\langle \tilde{\mathbf{v}}_{f(i)} - \tilde{\mathbf{v}}_j, \mathbf{B}' \text{ReLU}(\mathbf{A}'\tilde{\mathbf{k}}_i + \mathbf{b}') \rangle > 0 \quad \text{for all } i \in [F], j \neq f(i).$$

Proof. Define

$$\mathbf{A}' := \mathbf{A} \mathbf{T}_k^{-1}, \quad \mathbf{b}' := \mathbf{b} - \mathbf{A} \mathbf{T}_k^{-1} \mathbf{c}_k, \quad \mathbf{B}' := (\mathbf{T}_v^\top)^{-1} \mathbf{B}.$$

Then for each i ,

$$\text{ReLU}(\mathbf{A}'\tilde{\mathbf{k}}_i + \mathbf{b}') = \text{ReLU}(\mathbf{A} \mathbf{T}_k^{-1}(\mathbf{T}_k \mathbf{k}_i + \mathbf{c}_k) + \mathbf{b} - \mathbf{A} \mathbf{T}_k^{-1} \mathbf{c}_k) = \text{ReLU}(\mathbf{A}\mathbf{k}_i + \mathbf{b}).$$

Thus for any i and $j \neq f(i)$,

$$\begin{aligned} \langle \tilde{\mathbf{v}}_{f(i)} - \tilde{\mathbf{v}}_j, \mathbf{B}' \text{ReLU}(\mathbf{A}'\tilde{\mathbf{k}}_i + \mathbf{b}') \rangle &= \langle \mathbf{T}_v(\mathbf{v}_{f(i)} - \mathbf{v}_j), (\mathbf{T}_v^\top)^{-1} \mathbf{B} \text{ReLU}(\mathbf{A}\mathbf{k}_i + \mathbf{b}) \rangle \\ &= \langle \mathbf{v}_{f(i)} - \mathbf{v}_j, \mathbf{B} \text{ReLU}(\mathbf{A}\mathbf{k}_i + \mathbf{b}) \rangle > 0, \end{aligned}$$

using Equation (15). □

B.7.1 Decodability and affine transformations on embeddings

We study how the *decodability* of embeddings changes after affine transformations. Starting from the definition from Theorem B.5.1, we take the maximum over all decoder inputs:

$$\rho(\mathbf{V}) := \max_{\mathbf{U}} \min_{i \neq j} \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|}, \quad \mathbf{V} = \{\mathbf{v}_i\}_{i=1}^F \subset \mathbb{R}^d, \mathbf{U} = \{\mathbf{u}_i\}_{i=1}^F \subset \mathbb{R}^d \setminus \{0\}.$$

Given $\mathbf{V}$, consider new embeddings $\tilde{\mathbf{V}}$ via the affine map $\tilde{\mathbf{v}}_i = \mathbf{T}\mathbf{v}_i + \mathbf{c}$ with $\mathbf{T} \in \text{GL}(d)$, $\mathbf{c} \in \mathbb{R}^d$.

Lemma B.7.2 (Translation, scaling, and orthogonal invariance). *For any $\mathbf{c} \in \mathbb{R}^d$, $\alpha > 0$, and any orthogonal $\mathbf{R} \in \text{GL}(d)$,*

$$\rho(\mathbf{V} + \{\mathbf{c}\}) = \rho(\mathbf{V}), \quad \rho(\alpha \mathbf{V}) = \rho(\mathbf{V}), \quad \rho(\mathbf{R}\mathbf{V}) = \rho(\mathbf{V}).$$

Proof. Each claim follows by the invariance of the objective: (i) translation leaves all differences $\mathbf{v}_i - \mathbf{v}_j$ unchanged; (ii) positive scaling multiplies both the numerator and the $\|\mathbf{v}_i - \mathbf{v}_j\|$ factor by α ; (iii) taking $\tilde{\mathbf{u}}_i = \mathbf{R}\mathbf{u}_i$, orthogonality preserves inner products and norms, hence each cosine is unchanged. Taking min and then max preserves equality. □

Lemma B.7.3 (Linear conditioning bound). *Let $\mathbf{T} \in \text{GL}(d)$ with condition number $\kappa(\mathbf{T}) = \|\mathbf{T}\|_2 \|\mathbf{T}^{-1}\|_2 = \sigma_{\max}(\mathbf{T})/\sigma_{\min}(\mathbf{T})$. Then*

$$\frac{1}{\kappa(\mathbf{T})} \rho(\mathbf{V}) \leq \rho(\mathbf{T}\mathbf{V}) \leq \kappa(\mathbf{T}) \rho(\mathbf{V}).$$

Proof. Lower bound. Let $\mathbf{U}^* = \{\mathbf{u}_i^*\}$ attain $\rho(\mathbf{V})$. We compute the cosine similarity term for $\tilde{\mathbf{u}}_i := \mathbf{T}^{-\top} \mathbf{u}_i^*$ given transformed embeddings $\mathbf{T}\mathbf{V}$:

$$\frac{\langle \mathbf{T}(\mathbf{v}_i - \mathbf{v}_j), \tilde{\mathbf{u}}_i \rangle}{\|\mathbf{T}(\mathbf{v}_i - \mathbf{v}_j)\| \|\tilde{\mathbf{u}}_i\|} = \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i^* \rangle}{\|\mathbf{T}(\mathbf{v}_i - \mathbf{v}_j)\| \|\mathbf{T}^{-\top} \mathbf{u}_i^*\|} \geq \frac{1}{\kappa(\mathbf{T})} \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i^* \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i^*\|}.$$

¹⁴ $\text{GL}(d)$ is the group of invertible $d \times d$ (real) matrices.

Taking $\min_{j \neq i}$ and then max over $\tilde{\mathbf{U}}$ gives the left inequality.

Upper bound. Apply the lower bound from above to $\mathbf{V} = \mathbf{T}^{-1}(\mathbf{TV})$:

$$\rho(\mathbf{V}) \geq \frac{1}{\kappa(\mathbf{T}^{-1})} \rho(\mathbf{TV}) = \frac{1}{\kappa(\mathbf{T})} \rho(\mathbf{TV}),$$

so $\rho(\mathbf{TV}) \leq \kappa(\mathbf{T}) \rho(\mathbf{V})$. □

Remark 1 (Embedding-aware bound). Let $\mathbf{C} = \mathbf{T}^\top \mathbf{T} \succ 0$ and define

$$\kappa_{\text{eff}}(\mathbf{T}; \mathbf{V}, \mathbf{U}) := \max_i \max_{j \neq i} \sqrt{\frac{(\mathbf{v}_i - \mathbf{v}_j)^\top \mathbf{C} (\mathbf{v}_i - \mathbf{v}_j)}{\|\mathbf{v}_i - \mathbf{v}_j\|^2}} \cdot \frac{\mathbf{u}_i^\top \mathbf{C}^{-1} \mathbf{u}_i}{\|\mathbf{u}_i\|^2}.$$

Intuitively, $\kappa_{\text{eff}}(\mathbf{T}; \mathbf{V}, \mathbf{U})$ captures the worst-case conditioning of $\mathbf{T}$, when its action is restricted to the subspaces $\text{span}(\{\mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i\})$ for all $i \neq j$. Then computing the cosine similarity term for $\tilde{\mathbf{u}}_i = \mathbf{T}^{-\top} \mathbf{u}_i$ yields

$$\rho(\mathbf{TV}) \geq \frac{1}{\kappa_{\text{eff}}(\mathbf{T}; \mathbf{V}, \mathbf{U})} \min_{i \neq j} \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|}.$$

In particular, with $\mathbf{U} = \mathbf{U}^*$ that attains $\rho(\mathbf{V})$,

$$\rho(\mathbf{TV}) \geq \frac{\rho(\mathbf{V})}{\kappa_{\text{eff}}(\mathbf{T}; \mathbf{V}, \mathbf{U}^*)}, \quad \kappa_{\text{eff}}(\mathbf{T}; \mathbf{V}, \mathbf{U}^*) \leq \kappa(\mathbf{T}).$$

Remark 2 (Tightness). The $1/\kappa(\mathbf{T})$ lower bound is tight in general.

As a concrete example for $d = 2$, consider $\mathbf{v}_1 = (0, 0)$, $\mathbf{v}_2 = (1, 0)$, $\mathbf{v}_3 = (1, -\varepsilon)$. For $i = 1$, the tightest cosine margin is between $\mathbf{e}_1$ and $\mathbf{e}_1 - \varepsilon \mathbf{e}_2$. The optimal $\mathbf{u}_1^*$ then lies in the direction of their angle bisector, giving $\rho(\mathbf{V}) = \Theta(\varepsilon)$ as $\varepsilon \rightarrow 0$. Then, consider $\mathbf{T} = \text{diag}(\sigma_{\max}, \sigma_{\min})$, for which $\kappa(\mathbf{T}) = \sigma_{\max}/\sigma_{\min}$. A direct calculation with $\tilde{\mathbf{u}}_1 = \mathbf{T}^{-\top} \mathbf{u}_1^*$ shows $\rho(\mathbf{TV}) \approx \rho(\mathbf{V})/\kappa(\mathbf{T})$ as $\varepsilon \rightarrow 0$, showing the lower bound factor $1/\kappa_2(\mathbf{T})$ is tight.

B.8 Bit Complexity

Theorem B.8.1. Let $F = |K|$. Suppose that $h, d, m = O(\text{poly } F)$, that σ is an L^2 continuously differentiable function, that $\mathbf{G}$ is such that all its rows are i.i.d. $\mathbf{G}[i] \sim \text{Normal}(0, \mathbf{I}_d)$, that for all $\mathbf{k}_i \in K$, $\mathbf{k}_i$ is sampled from a rotationally invariant distribution with $\|\mathbf{k}_i\| \leq O(\text{poly } F)$, that the targets $\|\mathbf{o}_i\| \leq O(\text{poly } F)$, that $F \geq C_0 d h$ for some sufficiently large universal constant C_0 , that $\mathbb{E}[\sigma(\mathbf{G}[1]^\top \mathbf{k}_i) \mid \mathbf{k}_i] = 0$ for all i , and that $\rho \geq O(\frac{1}{\text{poly } F})$. Then with high probability (depending on F), the encoder / decoder construction described in Theorem B.6.1 requires $O(\log F)$ bits per parameter to store, of which there are $O(\text{poly } F)$.

Proof. See Section B.10.11. □

B.8.1 Noisy Decoding

Theorem B.8.2 (Noisy decoding via JL, Rademacher case). Let $\mathbf{D} \in \{-1, +1\}^{m \times d}$ have i.i.d. Rademacher entries ($\Pr(\mathbf{D}_{kl} = 1) = \Pr(\mathbf{D}_{kl} = -1) = \frac{1}{2}$) and set $\mathbf{M} := \frac{1}{m} \mathbf{D}^\top$. For each $i \in [N]$, let $\mathbf{v}_i, \mathbf{u}_i \in \mathbb{R}^d$ and define

$$\rho := \min_{i \neq j} \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|} > 0.$$

Let the noisy codes be

$$\mathbf{H}[i] := (\mathbf{D} \mathbf{u}_i) \odot (1 + \nu_i), \quad \nu_i \in [-\varepsilon, \varepsilon]^m, \quad \varepsilon \in [0, 1),$$

and define scores $s_{ij} := \langle \mathbf{v}_j, \mathbf{M} \mathbf{H}[i] \rangle$. Then there is a universal constant $C > 0$ such that if

$$m \geq \frac{C}{\rho^2} \ln \frac{4N(N-1)}{\delta},$$

then with probability at least $1 - \delta$ over $\mathbf{D}$, we have, simultaneously for all $i \neq j$,

$$s_{ii} - s_{ij} \geq \left(\frac{\rho}{2} - 4\varepsilon \right) \|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|.$$

Proof. See Section B.10.12. □

B.8.2 Bounding The Magnitudes

Lemma B.8.3. Let $\mathbf{k}_1, \dots, \mathbf{k}_F \in \mathbb{R}^d$ be i.i.d. random vectors with $\mathbf{k}_i \sim \mathcal{N}(0, \mathbf{I}_d)$. Then for every $c > 0$ there exists a constant $C = C(c) > 0$ such that

$$\Pr \left[\max_{1 \leq i \leq F} \|\mathbf{k}_i\|_2 \leq C(\sqrt{d} + \sqrt{\log F}) \right] \geq 1 - F^{-c}.$$

Proof. See Section B.10.13. □

Lemma B.8.4 (Row covariance is well-conditioned under rotationally invariant model). Fix $d, h \in \mathbb{N}$ and let

$$\mathbf{k} \in \mathbb{R}^d \quad \text{and} \quad \mathbf{G}[1], \dots, \mathbf{G}[h] \in \mathbb{R}^d$$

be random vectors such that:

- (i) $\mathbf{k}$ has a rotationally invariant distribution
- (ii) $\mathbf{G}[1], \dots, \mathbf{G}[h]$ are i.i.d. rotationally invariant.
- (iii) $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is a non-constant measurable function with $\mathbb{E}[\sigma(\mathbf{G}[1]^\top \mathbf{k})^2] < \infty$ and $\mathbb{E}[\sigma(\mathbf{G}[1]^\top \mathbf{k}) \mid \mathbf{k}] = 0$ a.s.

Define the random row vector $\mathbf{r}^\top \in \mathbb{R}^{dh}$ by

$$\mathbf{r}^\top(\mathbf{k}, \mathbf{G}[1], \dots, \mathbf{G}[h]) := (\sigma(\mathbf{G}[1]^\top \mathbf{k})\mathbf{k}^\top, \dots, \sigma(\mathbf{G}[h]^\top \mathbf{k})\mathbf{k}^\top),$$

and let

$$\Sigma_{\text{row}} := \mathbb{E}[\mathbf{r}\mathbf{r}^\top] \in \mathbb{R}^{dh \times dh}.$$

Then there exists a constant $c > 0$, depending only on the distributions of $\mathbf{k}$, $\mathbf{G}[\ell]$, and σ (but independent of F), such that

$$\lambda_{\min}(\Sigma_{\text{row}}) = \lambda_{\max}(\Sigma_{\text{row}}) = c.$$

In particular, $\lambda_{\min}(\Sigma_{\text{row}}) \geq F^{-C_1}$ and $\lambda_{\max}(\Sigma_{\text{row}}) \leq F^{C_2}$ for some fixed exponents C_1, C_2 and all F (i.e., the lower bound is $\frac{1}{\text{poly}(F)}$).

Proof. See Section B.10.14. □

Equipped with Theorem B.8.4 (which gives us assumption ii) in the theorem below) we may now finish the prove that the parameter magnitudes are bounded.

Theorem B.8.5 (Encoder weight norm bound). Fix an output coordinate j and consider the linear system

$$\mathbf{M}\mathbf{a} = \mathbf{o},$$

where $\mathbf{M} \in \mathbb{R}^{F \times dh}$ and $\mathbf{a} = \text{vec}(\mathbf{A}) \in \mathbb{R}^{dh}$. Assume:

- (i) The i -th row of $\mathbf{M}$ is

$$\mathbf{r}_i^\top = (\sigma(\mathbf{G}[1]^\top \mathbf{k}_i)\mathbf{k}_i^\top, \dots, \sigma(\mathbf{G}[h]^\top \mathbf{k}_i)\mathbf{k}_i^\top),$$

where $\{\mathbf{k}_i\}_{i=1}^F$ and $\{\mathbf{G}[\ell]\}_{\ell=1}^h$ are independent, rotationally invariant subgaussian random vectors in $\mathbb{R}^d$, and σ is continuously differentiable and non-constant.

- (ii) The covariance $\Sigma_{\text{row}} := \mathbb{E}[\mathbf{r}_i\mathbf{r}_i^\top]$ satisfies $\lambda_{\min}(\Sigma_{\text{row}}) \geq \lambda_0 > 0$ and $\lambda_{\max}(\Sigma_{\text{row}}) \leq \Lambda_0 < \infty$, with λ_0, Λ_0 independent of F .

- (iii) The targets $\mathbf{o} \in \mathbb{R}^F$ obey $|\mathbf{o}_i| \leq B(F)$ for all i , where $B(F) \leq \text{poly}(F)$.

- (iv) $F \geq C_0 dh$ for a sufficiently large absolute constant C_0 .

Let $\mathbf{a}_\star$ be the minimum- ℓ_2 -norm solution of $\mathbf{M}\mathbf{a} = \mathbf{o}$ (i.e. $\mathbf{a}_\star = \mathbf{M}^\dagger \mathbf{o}$). Then with probability at least $1 - e^{-cF}$, $c > 0$ we have

$$\|\mathbf{a}_\star\|_2 \leq \text{poly}(F).$$

Proof. See section Theorem B.8.5. □

B.8.3 Precision Bound

Lemma B.8.6 (Encoder is Lipschitz in the parameters). *Fix a number of facts F and keys $\{\mathbf{k}_i\}_{i=1}^F \subset \mathbb{R}^d$. Consider the scalar-output gated encoder*

$$\mathbf{enc}_\theta(\mathbf{x}) = \mathbf{1}_h^\top [\sigma(\mathbf{G}\mathbf{x}) \odot (\mathbf{A}\mathbf{x})] = \sum_{r=1}^h \sigma(\langle \mathbf{g}_r, \mathbf{x} \rangle) \langle \mathbf{a}_r, \mathbf{x} \rangle,$$

where $\mathbf{A}, \mathbf{G} \in \mathbb{R}^{h \times d}$ have rows $\mathbf{a}_r^\top, \mathbf{G}[r]^\top$, and $\theta \in \mathbb{R}^P$ is the vector of all entries of $\mathbf{A}, \mathbf{G}$.

Assume:

1. $\|\mathbf{k}_i\|_2 \leq R_{\mathbf{x}}(F)$ for all i , with $R_{\mathbf{x}}(F) \leq \text{poly}(F)$.
2. $\|\theta\|_2 \leq R_\theta(F)$, with $R_\theta(F) \leq \text{poly}(F)$.
3. The width and input dimension satisfy $h, d \leq \text{poly}(F)$, so that $P = 2hd \leq \text{poly}(F)$.
4. The activation $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is continuously differentiable and on the interval $[-B(F), B(F)]$ with $B(F) := R_\theta(F)R_{\mathbf{x}}(F)$ we have

$$|\sigma(t)| \leq C_\sigma, \quad |\sigma'(t)| \leq C'_\sigma \quad \forall t \in [-B(F), B(F)],$$

for some constants C_σ, C'_σ independent of F .¹⁵

Then for each key $\mathbf{k}_i$ there exists a constant $L(F) \leq \text{poly}(F)$ such that for all parameter vectors θ, θ' with $\|\theta\|_2, \|\theta'\|_2 \leq R_\theta(F)$,

$$\|\mathbf{enc}_\theta(\mathbf{k}_i) - \mathbf{enc}_{\theta'}(\mathbf{k}_i)\| \leq L(F) \|\theta - \theta'\|_2.$$

In particular, $\mathbf{enc}_\theta(\mathbf{k}_i)$ is Lipschitz in θ with Lipschitz constant at most polynomial in F .

Proof. See Section B.10.16. □

Theorem B.8.7 (Polynomial precision for encoder parameters). *Let F be the number of facts, and assume the noisy decoding theorem above holds for some choice of m (so that, for any codes whose noise is at most a fixed constant multiple of ρ , decoding is still correct).*

Assume the following polynomial bounds:

- (i) (Margin) $\rho \geq 1/\text{poly}(F)$.
- (ii) (Lipschitz in parameters) For each key $\mathbf{k}_i$ and all encoder parameter vectors θ, θ' ,
$$\|\mathbf{enc}_\theta(\mathbf{k}_i) - \mathbf{enc}_{\theta'}(\mathbf{k}_i)\| \leq L(F) \|\theta - \theta'\| \quad \text{with } L(F) \leq \text{poly}(F).$$
- (iii) (Parameter count) The number of encoder parameters satisfies $P \leq \text{poly}(F)$.
- (iv) (Magnitude) There is an encoder $\theta_\star$ such that $H_\star[i] := \mathbf{enc}_{\theta_\star}(\mathbf{k}_i) = \mathbf{D}\mathbf{u}_i$ and $\|\theta_\star\|_\infty \leq \text{poly}(F)$.

Then there exists a constant $c > 0$ such that if we quantize each coordinate of $\theta_\star$ to the grid $F^{-c}\mathbb{Z}$, obtaining $\tilde{\theta}$, the corresponding codes $\tilde{H}[i] := \mathbf{enc}_{\tilde{\theta}}(\mathbf{k}_i)$ still satisfy the conditions of the noisy decoding theorem and hence decode all F facts correctly. In particular, each encoder parameter requires only $O(\log F)$ bits of precision.

Proof. See Section B.10.17. □

¹⁵Since $\sigma \in C^1$ and all preactivations satisfy $|\langle \mathbf{g}_r, \mathbf{x} \rangle| \leq R_\theta(F)R_{\mathbf{x}}(F)$, they lie in the compact interval $[-B(F), B(F)]$. By continuity, σ and σ' are bounded on this interval, yielding constants $C_\sigma, C'_\sigma < \infty$. This ensures $\mathbf{enc}_\theta$ is Lipschitz in θ , with constants growing at most polynomially in F .

B.9 Spherical Chebyshev Bounds with a Fixed Anchor

We derive explicit lower and upper bounds on the spherical Chebyshev value ρ^* of the star $\{\mathbf{x}_{aj}\}_{j \neq a}$. We show (i) general bounds with no assumptions, (ii) simplifications under unit-norm embeddings, and (iii) coarse coherence-based corollaries.

Let $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^d$ and define, for any ordered pair (i, j) with $i \neq j$,

$$\mathbf{x}_{ij} := \frac{\mathbf{v}_i - \mathbf{v}_j}{\|\mathbf{v}_i - \mathbf{v}_j\|}.$$

We *always* assume a **fixed anchor** index a and consider only the star

$$\{\mathbf{x}_{aj} : j \neq a\}.$$

We are then interested in the following quantity:

Definition B.9.1. Define the Spherical Chebyshev value as

$$\rho^* := \max_{\|\mathbf{c}\|=1} \min_{j \neq a} \mathbf{c}^\top \mathbf{x}_{aj}$$

the cosine of the smallest spherical cap covering the star induced by anchor a .

B.9.1 General bounds (no norm assumptions on $\mathbf{v}_i$)

For notational simplicity, define

$$m_{\text{edge}} := \min_{\substack{j \neq k \\ j \neq a, k \neq a}} \mathbf{x}_{aj}^\top \mathbf{x}_{ak},$$

Then we have the following result.

Lemma B.9.2 (Spherical Chebyshev sandwich for a star). *For the spherical Chebyshev value ρ^* as defined above we have*

$$m_{\text{edge}} \leq \rho^* \leq \sqrt{\frac{1 + m_{\text{edge}}}{2}}.$$

Proof. For the lower bound, fix $j_0 \neq a$ and take $\mathbf{c} = \mathbf{x}_{aj_0}$. Then $\|\mathbf{c}\| = 1$ and

$$\min_{j \neq a} \mathbf{c}^\top \mathbf{x}_{aj} = \min_{j \neq a} \mathbf{x}_{aj_0}^\top \mathbf{x}_{aj} = \min \left(1, \min_{\substack{j \neq a \\ j \neq j_0}} \mathbf{x}_{aj_0}^\top \mathbf{x}_{aj} \right) \geq m_{\text{edge}},$$

so $\rho^* \geq m_{\text{edge}}$.

For the upper bound, pick j, k with $j \neq k$, $j \neq a$, $k \neq a$ such that $\mathbf{x}_{aj}^\top \mathbf{x}_{ak} = m_{\text{edge}}$. For any unit $\mathbf{c}$,

$$\min_{i \neq a} \mathbf{c}^\top \mathbf{x}_{ai} \leq \min(\mathbf{c}^\top \mathbf{x}_{aj}, \mathbf{c}^\top \mathbf{x}_{ak}),$$

hence

$$\rho^* \leq \sup_{\|\mathbf{c}\|=1} \min(\mathbf{c}^\top \mathbf{x}_{aj}, \mathbf{c}^\top \mathbf{x}_{ak}).$$

Let $P := \text{span}\{\mathbf{x}_{aj}, \mathbf{x}_{ak}\}$. Orthogonal projection onto P cannot decrease both inner products simultaneously, so the supremum is attained by some unit $\mathbf{c} \in P$. In an orthonormal basis of P , write

$$\mathbf{x}_{aj} = (1, 0), \quad \mathbf{x}_{ak} = (\cos \theta, \sin \theta), \quad \mathbf{c} = (\cos \varphi, \sin \varphi),$$

where $\theta := \arccos(\mathbf{x}_{aj}^\top \mathbf{x}_{ak})$ so $\cos \theta = m_{\text{edge}}$. Then

$$\mathbf{c}^\top \mathbf{x}_{aj} = \cos \varphi, \quad \mathbf{c}^\top \mathbf{x}_{ak} = \cos(\theta - \varphi),$$

and we must maximize

$$f(\varphi) := \min(\cos \varphi, \cos(\theta - \varphi)).$$

On $[0, \pi]$, $\cos$ is strictly decreasing, so f is maximized when $\cos \varphi = \cos(\theta - \varphi)$, i.e. $\varphi = \theta/2$, giving

$$\sup_{\|\mathbf{c}\|=1} \min(\mathbf{c}^\top \mathbf{x}_{aj}, \mathbf{c}^\top \mathbf{x}_{ak}) = \cos(\theta/2).$$

Therefore $\rho^* \leq \cos(\theta/2)$. Using $\cos^2(\theta/2) = \frac{1 + \cos \theta}{2}$ and $\cos \theta = m_{\text{edge}}$, we obtain

$$\rho^* \leq \sqrt{\frac{1 + \cos \theta}{2}} = \sqrt{\frac{1 + m_{\text{edge}}}{2}}.$$

Combining both bounds yields the claim. $\square$

B.9.2 Unit-norm specialization.

For notational simplicity, define

$$s_a := \max_{j \neq a} \mathbf{v}_a^\top \mathbf{v}_j$$

Lemma B.9.3 (Spherical Chebyshev bounds for a star: unit-norm case). *In the setting of Theorem B.9.2, assume in addition that $\|\mathbf{v}_i\| = 1$ for all $i \in [n]$. Then*

$$\sqrt{\frac{1-s_a}{2}} \leq \rho^* \leq \sqrt{\frac{1+m_{\text{edge}}}{2}}.$$

Proof. The upper bound follows directly from Theorem B.9.2. When $\|\mathbf{v}_i\| = 1$ for all i ,

$$\|\mathbf{v}_a - \mathbf{v}_j\| = \sqrt{2 - 2\mathbf{v}_a^\top \mathbf{v}_j}.$$

By direct calculation,

$$\mathbf{v}_a^\top \mathbf{x}_{aj} = \frac{1 - \mathbf{v}_a^\top \mathbf{v}_j}{\sqrt{2 - 2\mathbf{v}_a^\top \mathbf{v}_j}} = \sqrt{\frac{1 - \mathbf{v}_a^\top \mathbf{v}_j}{2}},$$

so

$$\rho^* \geq \min_{j \neq a} \sqrt{\frac{1 - \mathbf{v}_a^\top \mathbf{v}_j}{2}}. \quad (16)$$

Writing $s_a := \max_{j \neq a} \mathbf{v}_a^\top \mathbf{v}_j$ (the anchor's nearest neighbor in cosine),

$$\rho^* \geq \sqrt{\frac{1-s_a}{2}}. \quad (17)$$

□

To obtain bounds that depend only on a single global parameter, we now suppose the vectors satisfy a standard coherence condition.

B.9.3 Coherence-style corollaries (unit-norm)

Lemma B.9.4 (Coherence-style bounds for a fixed-anchor star). *In the setting of Theorem B.9.2, assume in addition that $|\mathbf{v}_i^\top \mathbf{v}_j| \leq \mu$ for all $i \neq j$, with $\mu \in [0, 1)$ and $\|\mathbf{v}_i\| = 1$ for all $i \in [n]$. Then the spherical Chebyshev value ρ^* satisfies*

$$\sqrt{\frac{1-\mu}{2}} \leq \rho^* \leq \sqrt{\frac{1}{2} \left(1 + \frac{1+3\mu}{2-2\mu} \right)}.$$

Proof. The coherence bound implies, for the anchor a ,

$$s_a := \max_{j \neq a} \mathbf{v}_a^\top \mathbf{v}_j \leq \mu.$$

By equation 17 from the unit-norm specialization,

$$\rho^* \geq \sqrt{\frac{1-s_a}{2}} \geq \sqrt{\frac{1-\mu}{2}}.$$

For any $j \neq k$ by direct computation,

$$\mathbf{x}_{aj}^\top \mathbf{x}_{ak} = \frac{1 - \mathbf{v}_a^\top \mathbf{v}_j - \mathbf{v}_a^\top \mathbf{v}_k + \mathbf{v}_j^\top \mathbf{v}_k}{\sqrt{(2 - 2\mathbf{v}_a^\top \mathbf{v}_j)(2 - 2\mathbf{v}_a^\top \mathbf{v}_k)}}.$$

Write $a_j := \mathbf{v}_a^\top \mathbf{v}_j$, $a_k := \mathbf{v}_a^\top \mathbf{v}_k$, $b_{jk} := \mathbf{v}_j^\top \mathbf{v}_k$. Then $|a_j|, |a_k|, |b_{jk}| \leq \mu$, so

$$1 - a_j - a_k + b_{jk} \leq 1 + |a_j| + |a_k| + |b_{jk}| \leq 1 + 3\mu,$$

and since $a_j, a_k \leq \mu$,

$$2 - 2a_j \geq 2 - 2\mu, \quad 2 - 2a_k \geq 2 - 2\mu,$$

hence

$$\sqrt{(2 - 2a_j)(2 - 2a_k)} \geq 2 - 2\mu.$$

Therefore, for all $j \neq k$,

$$\mathbf{x}_{aj}^\top \mathbf{x}_{ak} \leq \frac{1 + 3\mu}{2 - 2\mu},$$

and taking the minimum over $j \neq k$ yields

$$m_{\text{edge}} := \min_{j \neq k} \mathbf{x}_{aj}^\top \mathbf{x}_{ak} \leq \frac{1 + 3\mu}{2 - 2\mu}.$$

By Theorem B.9.2,

$$\rho^* \leq \sqrt{\frac{1 + m_{\text{edge}}}{2}} \leq \sqrt{\frac{1}{2} \left(1 + \frac{1 + 3\mu}{2 - 2\mu} \right)}.$$

Combining with the lower bound completes the proof. $\square$

B.10 Deferred proofs

B.10.1 Proof of Theorem B.4.4

Proof. We proceed in three steps:

1. Proof a Matroid Union Theorem sublemma which we use in Part 4.
2. Establish the rank upper bound from linear algebra principles.
3. Show that the set of $\mathbf{K}$ achieving this bound is Zariski open.
4. Show that this set is non-empty by constructing a $\mathbf{K}$ that achieves the bound.

Part 1: Matroid Union Theorem Sublemma

Lemma B.10.1. *The rank $R(\Sigma)$ is also given by:*

$$R(\Sigma) = \max_{\substack{\mathbf{I}_1, \dots, \mathbf{I}_d \subseteq [\mathbf{K}] \\ \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \ \forall i \neq j \\ \bigcup_{i=1}^d \mathbf{I}_i = [\mathbf{K}]}} \left[\sum_{i=1}^d \text{rank}(\Sigma[:, \mathbf{I}_i]) \right].$$

Proof. Define $R_k(\Sigma, S) = \min_{S' \subseteq S} [|S| - |S'| + k \cdot \text{rank}(\Sigma[:, S'])]$.

We first prove by induction on d that $R_k(\Sigma, S)$ is the rank of S in the matroid union of d copies of the matroid of Σ .

The base case is $d = 1$. In this case $R_1(\Sigma, S) = \min_{S' \subseteq S} [|S| - |S'| + \text{rank}(\Sigma[:, S'])]$ is minimized for $S = S'$, so $R_1(\Sigma, S) = \text{rank}(\Sigma[:, S])$, which is exactly the rank of S in the matroid union of 1 copy of the matroid of Σ (just the matroid of Σ).

Now, for the inductive step, suppose that the inductive hypothesis is true for $d - 1$. By the Matroid Union Theorem¹⁶ between the matroid of Σ and the matroid union of $d - 1$ copies of Σ , the rank of S under the matroid union of d copies of the matroid of Σ is given by

$$\begin{aligned} & \min_{S' \subseteq S} [|S| - |S'| + \text{rank}(\Sigma[:, S']) + R_{d-1}(\Sigma, S')] \\ &= \min_{S' \subseteq S} [|S| - |S'| + \text{rank}(\Sigma[:, S']) + \min_{S'' \subseteq S'} [|S'| - |S''| + (d-1)\text{rank}(\Sigma[:, S''])]] \\ &= \min_{S'' \subseteq S' \subseteq S} [|S| - |S''| + \text{rank}(\Sigma[:, S']) + (d-1)\text{rank}(\Sigma[:, S''])] \\ &= \min_{S'' \subseteq S} [|S| - |S''| + d \cdot \text{rank}(\Sigma[:, S''])] \\ &= R_d(\Sigma, S), \end{aligned}$$

as desired.

¹⁶See Theorem 11.3.1 of Oxley (2011).

Now, we prove that

$$R(\Sigma) = R_d(\Sigma, [\mathbf{K}]) = \max_{\substack{\mathbf{I}_1, \dots, \mathbf{I}_d \subseteq [\mathbf{K}] \\ \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \ \forall i \neq j}} \left[\sum_{i=1}^d \text{rank}(\Sigma[:, \mathbf{I}_i]) \right].$$

First, note that by the definition of the matroid union,

$$\begin{aligned} R(\Sigma) &= \max \left\{ \left| \bigcup_{i=1}^d \mathbf{I}_i \right| \mid \forall i \in [d], \text{rank}(\Sigma[:, \mathbf{I}_i]) = |\mathbf{I}_i| \right\} \\ &= \max \left\{ \left| \bigcup_{i=1}^d \mathbf{I}_i \right| \mid \forall i \in [d], \text{rank}(\Sigma[:, \mathbf{I}_i]) = |\mathbf{I}_i|, \quad \forall i \neq j \in [d], \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \right\} \\ &= \max \left\{ \sum_{i=1}^d |\mathbf{I}_i| \mid \forall i \in [d], \text{rank}(\Sigma[:, \mathbf{I}_i]) = |\mathbf{I}_i|, \quad \forall i \neq j \in [d], \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \right\} \\ &= \max \left\{ \sum_{i=1}^d \text{rank}(\Sigma[:, \mathbf{I}_i]) \mid \forall i \in [d], \text{rank}(\Sigma[:, \mathbf{I}_i]) = |\mathbf{I}_i|, \quad \forall i \neq j \in [d], \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \right\} \\ &= \max \left\{ \sum_{i=1}^d \text{rank}(\Sigma[:, \mathbf{I}_i]) \mid \forall i \neq j \in [d], \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \right\} \\ &= \max_{\substack{\mathbf{I}_1, \dots, \mathbf{I}_d \subseteq [\mathbf{K}] \\ \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \ \forall i \neq j}} \left[\sum_{i=1}^d \text{rank}(\Sigma[:, \mathbf{I}_i]) \right]. \end{aligned}$$

This completes our proof. $\square$

Part 2: Rank Upper Bound

We first derive the upper bound for $\mathbf{M}(\Sigma, \mathbf{K})$. The matrix $\mathbf{M} \equiv \mathbf{M}(\Sigma, \mathbf{K})$ is a $|\mathbf{K}| \times (dh)$ matrix. The definition $\mathbf{M} = [\text{diag}(\Sigma_1)\mathbf{K}, \dots, \text{diag}(\Sigma_h)\mathbf{K}]$ concatenates by h blocks of size $|\mathbf{K}| \times d$.

The columns of $\mathbf{M}$ can be re-grouped to form d blocks of size $|\mathbf{K}| \times h$. Let $\mathbf{M}_j$ be the j -th new block, $j \in [d]$. This block contains all columns from $\mathbf{M}$ that were constructed using the j -th column of $\mathbf{K}$, $\mathbf{K}[:, j]$. This block can be written as:

$$\mathbf{M}_j = \text{diag}(\mathbf{K}[:, j])\Sigma^\top$$

Here, $\text{diag}(\mathbf{K}[:, j])$ is $|\mathbf{K}| \times |\mathbf{K}|$ and $\Sigma^\top$ is $|\mathbf{K}| \times h$, so $\mathbf{M}_j$ is $|\mathbf{K}| \times h$. The full matrix $\mathbf{M}$ is a column-permutation of the concatenation $[\mathbf{M}_1, \dots, \mathbf{M}_d]$. The column space of $\mathbf{M}$ is the sum of the column spaces of these submatrices:

$$\text{col}(\mathbf{M}) = \sum_{j=1}^d \text{col}(\mathbf{M}_j).$$

By the subadditivity of rank over sums of subspaces, the rank is bounded by:

$$\begin{aligned} \text{rank}(\mathbf{M}) &\leq \min_{S \subseteq [\mathbf{K}]} (\text{rank}(\mathbf{M}[\neg S, :]) + \text{rank}(\mathbf{M}[S, :])) \\ &\leq \min_{S \subseteq [\mathbf{K}]} \left(\text{rank}(\mathbf{M}[\neg S, :]) + \sum_{j=1}^d \text{rank}(\mathbf{M}_j[S, :]) \right) \\ &\leq \min_{S \subseteq [\mathbf{K}]} \left(|\neg S| + \sum_{j=1}^d \text{rank}(\mathbf{M}_j[S, :]) \right) \end{aligned}$$

where S is a set of *row* indices, $\neg S$ is its complement ($|\neg S| = |\mathbf{K}| - |S|$), and $\mathbf{M}_j[S, :]$ is the submatrix of $\mathbf{M}_j$ with rows from S .

We now analyze $\text{rank}(\mathbf{M}_j[S, :])$:

$$\mathbf{M}_j[S, :] = (\text{diag}(\mathbf{K}[:, j])\Sigma^\top)[S, :] = \text{diag}(\mathbf{K}[S, j]) \cdot (\Sigma^\top[S, :]).$$

Note that $\Sigma^\top[S, :] = (\Sigma[:, S])^\top$. For any rectangular matrices $\mathbf{A}$ and $\mathbf{B}$ we have¹⁷ $\text{rank}(\mathbf{AB}) \leq \text{rank}(\mathbf{B})$. Thus:

$$\text{rank}(\mathbf{M}_j[S, :]) \leq \text{rank}((\Sigma[:, S])^\top) = \text{rank}(\Sigma[:, S]).$$

Substituting this back into our rank bound for $\mathbf{M}$:

$$\begin{aligned} \text{rank}(\mathbf{M}) &\leq \min_{S \subseteq [\mathbf{K}]} \left((|\mathbf{K}| - |S|) + \sum_{j=1}^d \text{rank}(\Sigma[:, S]) \right) \\ \text{rank}(\mathbf{M}(\Sigma, \mathbf{K})) &\leq \min_{S \subseteq [\mathbf{K}]} \left[|\mathbf{K}| - |S| + d \cdot \text{rank}(\Sigma[:, S]) \right] \equiv R(\Sigma). \end{aligned}$$

This establishes $R(\Sigma)$ as the maximum possible rank.

Part 3: a Zariski open set

Let $R = R(\Sigma)$. From Part 2, the rank cannot exceed R . The set of $\mathbf{K}$ for which the rank is *sub-maximal* is $\mathcal{K}^c = \{\mathbf{K} \mid \text{rank}(\mathbf{M}(\Sigma, \mathbf{K})) < R\}$.

This condition $\text{rank}(\mathbf{M}(\Sigma, \mathbf{K})) < R$ holds if and only if every $R \times R$ submatrix of $\mathbf{M}(\Sigma, \mathbf{K})$ has a determinant equal to 0.

The entries of $\mathbf{M}(\Sigma, \mathbf{K})$ are polynomial functions of the entries of Σ and $\mathbf{K}$. Since Σ is fixed, the determinant of any $R \times R$ submatrix is a polynomial in the entries (components) of $\mathbf{K}$. Let this finite set of polynomials be $\mathcal{P} = \{p_j(\mathbf{K})\}_j$.

The set $\mathcal{K}^c$ is the set of $\mathbf{K}$ that are common zeros of all polynomials in $\mathcal{P}$. By definition, this set $\mathcal{K}^c$ is an algebraic variety (a Zariski closed set). The set $\mathcal{K} = \{\mathbf{K} \mid \text{rank}(\mathbf{M}(\Sigma, \mathbf{K})) = R\}$ is the complement of $\mathcal{K}^c$. As the complement of a Zariski closed set, $\mathcal{K}$ is, by definition, a Zariski open set.

An algebraic variety over $\mathbb{R}$ or $\mathbb{C}$ is either the entire space or a set of measure zero. To show $\mathcal{K}$ has full measure, it suffices to show it is non-empty (proving $\mathcal{K}^c$ is not the entire space). We construct an explicit $\mathbf{K}$ that achieves the maximum rank $R(\Sigma)$.

Part 4: An explicit example

By the Matroid Union Theorem¹⁸, the rank $R(\Sigma)$ is also given by:

$$R(\Sigma) = \max_{\substack{\mathbf{I}_1, \dots, \mathbf{I}_d \subseteq [\mathbf{K}] \\ \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \ \forall i \neq j}} \left[\sum_{i=1}^d \text{rank}(\Sigma[:, \mathbf{I}_i]) \right].$$

Let $\mathbf{I}_1^*, \dots, \mathbf{I}_d^*$ be an optimal partition, defined as:

$$(\mathbf{I}_1^*, \dots, \mathbf{I}_d^*) = \underset{\substack{\mathbf{I}_1, \dots, \mathbf{I}_d \subseteq [\mathbf{K}] \\ \mathbf{I}_i \cap \mathbf{I}_j = \emptyset \ \forall i \neq j}}{\text{argmax}} \left[\sum_{i=1}^d \text{rank}(\Sigma[:, \mathbf{I}_i]) \right].$$

Construct $\mathbf{K}(\mathbf{I}_1^*, \dots, \mathbf{I}_d^*) \in \mathbb{R}^{|\mathbf{K}| \times d}$ as in Theorem B.4.2. Then, by Theorem B.4.3,

$$\text{rank}(\mathbf{M}(\Sigma, \mathbf{K}(\mathbf{I}_1^*, \dots, \mathbf{I}_d^*))) = \sum_{j=1}^d \text{rank}(\Sigma[:, \mathbf{I}_j^*]).$$

This is exactly the maximal value $R(\Sigma)$. Since one $\mathbf{K}$ has been found for which the rank $R(\Sigma)$ is achieved, the set $\mathcal{K}$ is non-empty. □

B.10.2 Proof of Theorem B.4.6

Proof. Define a map

$$F : S \rightarrow \mathbb{R}^r, \quad F(\mathbf{a}) := [f_1(\mathbf{a}), \dots, f_r(\mathbf{a})].$$

Then, for any choice of points $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(r)} \in S$, the j -th column of the matrix $M = (f_i(\mathbf{a}^{(j)}))_{1 \leq i, j \leq r}$ is exactly the vector $F(\mathbf{a}^{(j)}) \in \mathbb{R}^r$. Thus, it suffices to show that there exist points $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(r)} \in S$ such that the vectors $F(\mathbf{a}^{(1)}), \dots, F(\mathbf{a}^{(r)})$ are linearly independent in $\mathbb{R}^r$.

¹⁷This follows by basic properties of linear maps. Let $B : \mathbb{R}^p \rightarrow \mathbb{R}^n$, $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $AB : \mathbb{R}^p \rightarrow \mathbb{R}^m$. Then $\text{Im}(AB) = A(\text{Im}(B))$, so $\text{rank}(AB) = \dim(\text{Im}(AB)) = \dim(A(\text{Im}(B))) \leq \dim(\text{Im}(B)) = \text{rank}(B)$.

¹⁸See Theorem 11.3.1 of Oxley (2011).

We construct such points inductively.

Base step. Since $f_1, \dots, f_r$ are linearly independent as functions on S , not all of them are identically zero. Hence, there exists some $\mathbf{a}^{(1)} \in S$ such that

$$F(\mathbf{a}^{(1)}) = [f_1(\mathbf{a}^{(1)}), \dots, f_r(\mathbf{a}^{(1)})] \neq 0.$$

Thus the single vector $F(\mathbf{a}^{(1)})$ is linearly independent (as a set of size one).

Inductive step. Assume that for some k with $1 \leq k < r$ we have already chosen points $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(k)} \in S$ such that

$$F(\mathbf{a}^{(1)}), \dots, F(\mathbf{a}^{(k)})$$

are linearly independent in $\mathbb{R}^r$. Let

$$W := \text{span}\{F(\mathbf{a}^{(1)}), \dots, F(\mathbf{a}^{(k)})\} \subset \mathbb{R}^r.$$

Then $\dim W = k < r$, so W is a proper subspace of $\mathbb{R}^r$.

We claim there exists $\mathbf{a}^{(k+1)} \in S$ such that $F(\mathbf{a}^{(k+1)}) \notin W$. Suppose, for contradiction, that $F(\mathbf{a}) \in W$ for all $\mathbf{a} \in S$. Since W is a proper subspace of $\mathbb{R}^r$, there exists a nonzero linear functional $\ell : \mathbb{R}^r \rightarrow \mathbb{R}$ such that $\ell(v) = 0$ for all $v \in W$. Equivalently, there exists a nonzero vector $\lambda = (\lambda_1, \dots, \lambda_r) \in \mathbb{R}^r$ such that

$$\lambda \cdot v = 0 \quad \text{for all } v \in W.$$

In particular, for every $\mathbf{a} \in S$ we have $F(\mathbf{a}) \in W$, hence

$$0 = \lambda \cdot F(\mathbf{a}) = \sum_{i=1}^r \lambda_i f_i(\mathbf{a}).$$

Therefore the function

$$g := \sum_{i=1}^r \lambda_i f_i$$

is identically zero on S , i.e.,

$$g(\mathbf{a}) = 0 \quad \text{for all } \mathbf{a} \in S.$$

Since $\lambda \neq 0$, this is a nontrivial linear relation among the functions $f_1, \dots, f_r$, contradicting the assumption that they are linearly independent.

Hence our supposition was false, and there exists some $\mathbf{a}^{(k+1)} \in S$ with $F(\mathbf{a}^{(k+1)}) \notin W$. Then

$$F(\mathbf{a}^{(1)}), \dots, F(\mathbf{a}^{(k)}), F(\mathbf{a}^{(k+1)})$$

are linearly independent in $\mathbb{R}^r$, completing the inductive step.

By induction, we can choose points $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(r)} \in S$ so that the vectors $F(\mathbf{a}^{(1)}), \dots, F(\mathbf{a}^{(r)})$ are linearly independent in $\mathbb{R}^r$. Equivalently, the $r \times r$ matrix

$$M = (f_i(\mathbf{a}^{(j)}))_{1 \leq i, j \leq r}$$

has r linearly independent columns, so $\text{rank}(M) = r$, and M is invertible. $\square$

B.10.3 Proof of Theorem B.4.7

Proof. Since σ is real-analytic and not a polynomial, its Taylor series at any point has infinitely many nonzero coefficients.

(1) **The family** $\{\sigma(\lambda t)\}$. Expand σ at 0:

$$\sigma(t) = \sum_{k=0}^{\infty} c_k t^k$$

with infinitely many $c_k \neq 0$. For $n \in \mathbb{N}$, define

$$f_n(t) := \sigma(nt).$$

We show that $\{f_n\}_{n \geq 1}$ is linearly independent.

Suppose, for some $N \geq 1$, there exist real numbers $\beta_1, \dots, \beta_N$ such that

$$\sum_{n=1}^N \beta_n f_n(t) \equiv 0 \quad \text{as a function of } t.$$

Expand using the Taylor series:

$$0 = \sum_{n=1}^N \beta_n \sigma(nt) = \sum_{n=1}^N \beta_n \sum_{k=0}^{\infty} c_k (nt)^k = \sum_{k=0}^{\infty} c_k \left(\sum_{n=1}^N \beta_n n^k \right) t^k.$$

Since two power series are equal if and only if all their coefficients are equal, we obtain

$$c_k \left(\sum_{n=1}^N \beta_n n^k \right) = 0 \quad \text{for all } k \geq 0.$$

For each k with $c_k \neq 0$, this implies

$$\sum_{n=1}^N \beta_n n^k = 0. \quad (*)$$

Because there are infinitely many k with $c_k \neq 0$, we have infinitely many equations $(*)$. Let $n_{\max}$ be the largest index with $\beta_{n_{\max}} \neq 0$. Define

$$S(k) := \sum_{n=1}^N \beta_n n^k.$$

Then for each such k ,

$$S(k) = 0.$$

Now divide by $n_{\max}^k$:

$$\frac{S(k)}{n_{\max}^k} = \beta_{n_{\max}} + \sum_{n=1}^{N-1} \beta_n \left(\frac{n}{n_{\max}} \right)^k.$$

Since $n < n_{\max}$, we have $\left| \frac{n}{n_{\max}} \right| < 1$, and so

$$\sum_{n=1}^{N-1} \beta_n \left(\frac{n}{n_{\max}} \right)^k \xrightarrow{k \rightarrow \infty} 0.$$

Thus

$$\frac{S(k)}{n_{\max}^k} \xrightarrow{k \rightarrow \infty} \beta_{n_{\max}}.$$

On the other hand, $S(k) = 0$ for infinitely many k (all those with $c_k \neq 0$), and these k tend to infinity. Along that subsequence k_j , we have

$$0 = \frac{S(k_j)}{n_{\max}^{k_j}} \xrightarrow{j \rightarrow \infty} \beta_{n_{\max}},$$

so $\beta_{n_{\max}} = 0$, contradicting the definition of $n_{\max}$. Therefore all β_n must be zero, and $\{f_n\}_{n \geq 1}$ is linearly independent. Hence the span of $\{\sigma(\lambda t)\}$ is infinite-dimensional. $\square$

B.10.4 Proof of Theorem B.4.9

Proof. Note that if $|S_1| = 0$ or $|S_2| = 0$, then the submatrix $\sigma(\mathbf{xy}^\top)[S_1, S_2]$ has rank 0, which agrees with $\min\{|S_1|, |S_2|\}$. Thus such subsets impose no nontrivial constraints, and we may freely ignore them in the argument below.

Define the row-restricted vectors

$$\mathbf{x}_{S_1} := \mathbf{x}[S_1] \in \mathbb{R}^{|S_1|}, \quad \mathbf{y}_{S_2} := \mathbf{y}[S_2] \in \mathbb{R}^{|S_2|}.$$

Then $\sigma(\mathbf{xy}^\top)[S_1, S_2] = \sigma((\mathbf{xy}^\top)[S_1, S_2]) = \sigma((\mathbf{x}_{S_1}\mathbf{y}_{S_2}^\top))$.

Now, for arbitrary nonempty subsets $S_1 \in [d_1]$ and $S_2 \in [d_2]$, define the map

$$\pi_{S_1, S_2} : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \rightarrow \mathbb{R}^{|S_1|} \times \mathbb{R}^{|S_2|}, \quad \pi_{S_1, S_2}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}_{S_1}, \mathbf{y}_{S_2}).$$

This map is analytic and surjective.

By Lemma B.4.8, the set

$$\mathcal{S}_{S_1, S_2}^{\text{part}} := \left\{ (\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in \mathbb{R}^{|S_1|}, \quad \mathbf{y} \in \mathbb{R}^{|S_2|}, \quad \text{rank}(\sigma(\mathbf{xy}^\top)) = \min\{|S_1|, |S_2|\} \right\}$$

is the complement of a proper analytic subvariety of $\mathbb{R}^{|S_1|} \times \mathbb{R}^{|S_2|}$.

Define the corresponding full-parameter set

$$\mathcal{S}^{(S_1, S_2)} := \pi_{S_1, S_2}^{-1}(\mathcal{S}_{S_1, S_2}^{\text{part}}) \subseteq \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}.$$

Let

$$\mathcal{V}_{S_1, S_2}^{\text{part}} := (\mathcal{S}_{S_1, S_2}^{\text{part}})^c$$

denote the “bad” set in the smaller space (a proper analytic subvariety by Lemma B.4.8) and define

$$\mathcal{V}_{S_1, S_2} := (\mathcal{S}^{(S_1, S_2)})^c = \pi_{S_1, S_2}^{-1}(\mathcal{V}_{S_1, S_2}^{\text{part}}).$$

Since π_{S_1, S_2} is analytic, the preimage of an analytic subvariety is again an analytic subvariety, so $\mathcal{V}_{S_1, S_2}$ is an analytic subvariety of $\mathbb{R}^{n \times d} \times \mathbb{R}^{h \times d}$. It is *proper* because $\mathcal{V}_{S_1, S_2}^{\text{part}}$ is a proper subset and π_{S_1, S_2} is surjective: there are points $(\mathbf{x}, \mathbf{y})$ in $\mathcal{S}_{S_1, S_2}^{\text{part}}$, and any lift of such a point is not in $\mathcal{V}_{S_1, S_2}$.

Now define the global no-bias set

$$\mathcal{S} := \bigcap_{\substack{S_1 \subseteq [h] \\ S_2 \subseteq [n]}} \mathcal{S}^{(S_1, S_2)}.$$

The complement of $\mathcal{S}$ is

$$\mathcal{S}^c = \bigcup_{\substack{S_1 \subseteq [h] \\ S_2 \subseteq [n]}} \mathcal{V}_{S_1, S_2}.$$

This is a finite union of analytic subvarieties, hence itself an analytic subvariety (see e.g., 1.2 of Chirka (1997)).

Finally, to see that $\mathcal{S}^c$ is *proper*, it suffices to note that each $\mathcal{V}_{S_1, S_2}$ is a proper analytic subvariety, hence has empty interior (a nontrivial real analytic function cannot vanish on a nonempty open set). Because the union is finite, the union also has empty interior, and so its complement $\mathcal{S}$ is nonempty and dense. Thus $\mathcal{S}$ is the complement of a proper analytic subvariety of $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$, and it is full measure, completing the proof. $\square$

B.10.5 Proof of Theorem B.4.10

Proof. Throughout, $N := |\mathbf{K}|$ and we assume $d \geq h$.

Define

$$F : (\mathbf{K}, \mathbf{G}) \mapsto \mathbf{M}(\sigma(\mathbf{GK}^\top), \mathbf{K}) \in \mathbb{R}^{N \times (dh)}.$$

Each entry of $\mathbf{GK}^\top$ is a polynomial in the entries of $(\mathbf{K}, \mathbf{G})$. Since σ is analytic, each entry of $\sigma(\mathbf{GK}^\top)$ is an analytic function of $(\mathbf{K}, \mathbf{G})$. Multiplying by $\mathbf{K}$ and taking diagonals are polynomial operations, hence every entry of $\mathbf{M}(\sigma(\mathbf{GK}^\top), \mathbf{K})$ is analytic in $(\mathbf{K}, \mathbf{G})$.

Therefore, every $N \times N$ minor of $\mathbf{M}(\sigma(\mathbf{GK}^\top), \mathbf{K})$ is an analytic function of $(\mathbf{K}, \mathbf{G})$. The set

$$\mathcal{B} := \left\{ (\mathbf{K}, \mathbf{G}) : \text{rank } \mathbf{M}(\sigma(\mathbf{GK}^\top), \mathbf{K}) < N \right\}$$

is exactly the common zero set of all these minors, hence an *analytic subvariety* of $\mathbb{R}^{N \times d} \times \mathbb{R}^{h \times d}$.

If we can find *one* parameter choice for which the corresponding matrix has full row rank N , then not all $N \times N$ minors vanish identically, and the “bad” set is a *proper* analytic subvariety. Its complement is then a nonempty Zariski open set, proving the desired generic statement.

Thus, the rest of the proof is devoted to constructing such a full-row-rank example.

Define $\mathbf{I}_i = \{j \mid j \in [|\mathbf{K}|], (i-1)h < j \leq ih\}$ for all $i \in [d]$. Fix pairwise distinct nonzero scalars $\{\alpha_t\}_{t=1}^N$. Also, define $\vec{\alpha} = [\alpha_1, \dots, \alpha_N]$.

Finally, define $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$ such that $\mathbf{K}[i, j] = \alpha_i \mathbb{1}\{i \in \mathbf{I}_j\}$. Note that each α_i occurs exactly once in $\mathbf{K}$.

We keep this $\mathbf{K}$ fixed from now on. We will choose $\mathbf{G}$ and $\vec{\alpha}$ to make the resulting $\mathbf{M}$ full row rank. By Theorem B.4.3, we have

$$\text{rank}(\mathbf{M}(\Sigma, \mathbf{K})) = \sum_{j=1}^d \text{rank}(\Sigma[:, \mathbf{I}_j]),$$

so we must simply choose $\mathbf{G}$ and α such that $\text{rank}(\Sigma[:, \mathbf{I}_j]) = |\mathbf{I}_j|$ for all $j \in [d]$.

Now,

$$\begin{aligned} \Sigma[:, \mathbf{I}_j] &= \sigma(\mathbf{G}\mathbf{K}^\top)[:, \mathbf{I}_j] \\ &= \sigma(\mathbf{G}\mathbf{K}^\top[:, \mathbf{I}_j]) \\ &= \sigma(\mathbf{G}(\mathbf{K}[\mathbf{I}_j, :])^\top) \\ &= \sigma(\mathbf{G}[:, j](\vec{\alpha}[\mathbf{I}_j])^\top) \in \mathbb{R}^{h \times |\mathbf{I}_j|}. \end{aligned}$$

Now, $\text{rank}[\sigma] \geq h$, by Theorem B.4.8, $\sigma(\mathbf{G}[:, j](\vec{\alpha}[\mathbf{I}_j])^\top)$ has rank $|\mathbf{I}_j|$ for generic $\mathbf{G}[:, j]$ and $\vec{\alpha}[\mathbf{I}_j]$.

Thus there exists $\mathbf{G}$ and $\vec{\alpha}$ such that $\text{rank}(\Sigma[:, \mathbf{I}_j]) = |\mathbf{I}_j|$ for all $j \in [d]$.

This completes the proof. $\square$

B.10.6 Proof of Theorem B.2.2

Proof. We first assume our code to be *softmax-decodable* as defined in Theorem B.2.1 to prove the forward direction. For the sake of contradiction, assume there exists some $\mathbf{H}[i]$, $i, j \neq i$ such that

$$\langle \mathbf{M}\mathbf{H}[i], \tilde{\mathbf{v}}_j \rangle \geq \langle \mathbf{M}\mathbf{H}[i], \tilde{\mathbf{v}}_i \rangle \quad (18)$$

For ease of notation, define

$$\begin{aligned} w &= \exp(\langle \mathbf{M}\mathbf{H}[i], \tilde{\mathbf{v}}_i \rangle), \\ z &= \exp(\langle \mathbf{M}\mathbf{H}[i], \tilde{\mathbf{v}}_j \rangle), \\ S &= \sum_{k=1}^n \exp(\langle \mathbf{M}\mathbf{H}[i], \tilde{\mathbf{v}}_k \rangle). \end{aligned}$$

Theorem B.2.1 gives

$$\left| \frac{w}{S} - 1 \right| < \alpha, \quad \frac{z}{S} < \alpha. \quad (19)$$

Since Theorem B.2.1 holds for all $\frac{1}{2} > \alpha > 0$, fix some $\alpha < 1/2$. From the first inequality,

$$\frac{w}{S} > 1 - \alpha \implies S < \frac{w}{1 - \alpha}. \quad (20)$$

Substituting this into the second part of (2) yields

$$z < \alpha S < \frac{\alpha w}{1 - \alpha}. \quad (21)$$

Inequality (4) and our assumption Equation (18) implies that

$$w < \frac{\alpha w}{1 - \alpha} \implies 1 < \frac{\alpha}{1 - \alpha} \implies \alpha > \frac{1}{2},$$

contradicting $\alpha < \frac{1}{2}$. Therefore

$$\langle \mathbf{M}\mathbf{H}[i], \tilde{\mathbf{v}}_i \rangle > \langle \mathbf{M}\mathbf{H}[i], \tilde{\mathbf{v}}_j \rangle$$

for every $j \neq i$. We now prove the backwards direction.

Assume that for every index i

$$\langle \mathbf{MH}[i], \mathbf{y}_i \rangle > \langle \mathbf{MH}[i], \mathbf{y}_j \rangle \quad \text{for all } j \neq i. \quad (22)$$

Then we show that we can handle any tolerance by scaling $\mathbf{M}$. For any $\mathbf{H}$ and i and for ease of notation define

$$\begin{aligned} \mathbf{z}_k &= \langle \mathbf{MH}[i], \mathbf{y}_k \rangle, \\ g &= \min_{j \neq i} (\mathbf{z}_i - \mathbf{z}_j). \end{aligned}$$

Choose $\lambda > 0$ and set $\mathbf{M}_\lambda = \lambda \mathbf{M}$. Define

$$\begin{aligned} \tilde{\mathbf{z}}_k(\lambda) &= \lambda \mathbf{z}_k, \\ p_k(\lambda) &= \frac{\exp(\tilde{\mathbf{z}}_k(\lambda))}{\sum_{\ell} \exp(\tilde{\mathbf{z}}_{\ell}(\lambda))}. \end{aligned}$$

Because $\mathbf{z}_i - \mathbf{z}_j \geq g$ for every $j \neq i$,

$$p_i(\lambda) = \frac{1}{1 + \sum_{j \neq i} \exp(\lambda(\mathbf{z}_j - \mathbf{z}_i))} \geq \frac{1}{1 + (n-1) \exp(-\lambda g)}, \quad (23)$$

$$p_j(\lambda) = \frac{\exp(\lambda \mathbf{z}_j)}{\exp(\lambda \mathbf{z}_i) + \sum_{\ell \neq i} \exp(\lambda \mathbf{z}_{\ell})} = \frac{\exp(-\lambda(\mathbf{z}_i - \mathbf{z}_j))}{1 + \sum_{\ell \neq i} \exp(-\lambda(\mathbf{z}_i - \mathbf{z}_{\ell}))} \leq \exp(-\lambda g). \quad (24)$$

Given any $\alpha \in (0, 1/2)$ pick

$$\lambda > \frac{1}{g} \ln((n-1)/\alpha). \quad (25)$$

Then $(n-1) \exp(-\lambda g) < \alpha$ and $\exp(-\lambda g) < \alpha$, so Equation (23)–Equation (25) give

$$p_i(\lambda) > 1 - \alpha, \quad p_j(\lambda) < \alpha \text{ for } j \neq i.$$

Also note that since $\exp$ has positive range and addition is monotonic over $\mathbb{Z}^+$, for all i, j, λ :

$$p_i(\lambda) \leq 1, \quad p_j(\lambda) \geq 0.$$

Hence

$$\left\| \text{softmax}_k(\langle \mathbf{M}_\lambda \mathbf{H}[i], \mathbf{y}_k \rangle) - e_i \right\|_{\infty} < \alpha.$$

Since α was arbitrary, the softmax condition holds for every tolerance after scaling $\mathbf{M}$ by a suitable λ . $\square$

B.10.7 Proof of Theorem B.5.3

Fix a finite $\mathcal{P} \subset \mathbb{S}^{d-1} \times \mathbb{S}^{d-1}$ and define

$$\mathcal{S}_{\pm} := \{\mathbf{x} \pm \mathbf{y} : (\mathbf{x}, \mathbf{y}) \in \mathcal{P}\}.$$

Going forward, for convenience we use the notation

$$\mathbf{a}_{ij} := \mathbf{v}_i - \mathbf{v}_j, \quad \mathbf{b}_i := \mathbf{u}_i,$$

define

$$\hat{\mathbf{a}}_{ij} = \mathbf{a}_{ij} / \|\mathbf{a}_{ij}\|, \quad \hat{\mathbf{b}}_i = \mathbf{b}_i / \|\mathbf{b}_i\|.$$

We first show the following intermediate result.

Lemma B.10.2. *Let $\Phi = \frac{1}{\sqrt{m}} \mathbf{D}$ with $\mathbf{D} \in \mathbb{R}^{m \times d}$ having i.i.d. $\mathcal{N}(0, 1)$ entries.*

Then for any $\varepsilon \in (0, 1)$,

$$\Pr \left[\forall (\mathbf{x}, \mathbf{y}) \in \mathcal{P} : |\langle \Phi \mathbf{x}, \Phi \mathbf{y} \rangle - \langle \mathbf{x}, \mathbf{y} \rangle| \leq \varepsilon \right] \geq 1 - 2|\mathcal{S}_{\pm}| \exp\left(-\frac{\varepsilon^2}{8} m\right).$$

Equivalently, it suffices that

$$m \geq \frac{8}{\varepsilon^2} \ln\left(\frac{2|\mathcal{S}_{\pm}|}{\delta}\right) \quad (26)$$

to ensure the event above holds with probability at least $1 - \delta$.

Proof. See Section B.10.8 □

Corollary B.10.3. Let $\mathbf{E} := \Phi^\top \Phi - \mathbf{I}$ with $\Phi = \frac{1}{\sqrt{m}} \mathbf{D}$ and $\mathbf{D}$ i.i.d. standard Gaussian. If Equation (26) holds, then for

$$\mathcal{P} = \{(\hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_i) : i \in [|\mathbf{K}|], j \neq i\}$$

it follows that

$$\mathcal{S}_\pm = \{\hat{\mathbf{a}}_{ij} \pm \hat{\mathbf{b}}_i\}, \quad |\mathcal{S}_\pm| \leq 2|\mathbf{K}|(|\mathbf{K}| - 1),$$

we have, simultaneously for all $i \neq j$,

$$|\mathbf{a}_{ij}^\top \mathbf{E} \mathbf{b}_i| = \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| \cdot |\langle \Phi \hat{\mathbf{a}}_{ij}, \Phi \hat{\mathbf{b}}_i \rangle - \langle \hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_i \rangle| \leq \varepsilon \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\|.$$

Proof. This follows directly from Theorem B.10.2.

Equipped with these results, the proof of the theorem is relatively concise.

Define $\mathbf{s}_{ij} = \langle \mathbf{v}_j, \mathbf{M}\mathbf{H}[i] \rangle = \langle \mathbf{v}_j, \frac{1}{m} \mathbf{D}^\top \mathbf{D} \mathbf{u}_i \rangle$. Apply Theorem B.10.3 with $\varepsilon = \rho_{\min}/2$ to the family $\mathcal{P} = \{(\hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_i)\}$. By Theorem B.10.3, $|\mathbf{a}_{ij}^\top \mathbf{E} \mathbf{b}_i| \leq (\rho_{\min}/2) \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\|$, where $\mathbf{E}$ is the same as in Theorem B.10.3. We then have $\mathbf{s}_{ii} - \mathbf{s}_{ij} = \langle \mathbf{v}_i - \mathbf{v}_j, \frac{1}{m} \mathbf{D}^\top \mathbf{D} \mathbf{u}_i \rangle = \langle \mathbf{a}_{ij}, (\mathbf{I} + \mathbf{E}) \mathbf{b}_i \rangle = \langle \mathbf{a}_{ij}, \mathbf{b}_i \rangle + \mathbf{a}_{ij}^\top \mathbf{E} \mathbf{b}_i$. By definition of $\rho_{\min}$, $\langle \mathbf{a}_{ij}, \mathbf{b}_i \rangle \geq \rho_{\min} \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\|$. Therefore each gap satisfies

$$\mathbf{s}_{ii} - \mathbf{s}_{ij} = \langle \mathbf{a}_{ij}, \mathbf{b}_i \rangle + \mathbf{a}_{ij}^\top \mathbf{E} \mathbf{b}_i \geq \rho_{\min} \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| - (\rho_{\min}/2) \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| = (\rho_{\min}/2) \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| > 0,$$

simultaneously for all $i \neq j$ on the high-probability event. To make this event have probability at least $1 - \delta$, Lemma B.10.2 requires $m \geq \frac{8}{(\rho_{\min}/2)^2} \ln(2|\mathcal{S}_\pm|/\delta)$. Substituting in $|\mathcal{S}_\pm| \leq 2|\mathbf{K}|(|\mathbf{K}| - 1)$, which follows from the number of elements in $\mathcal{P}$, provides the stated condition. □

B.10.8 Proof of Theorem B.10.2

Proof. For any fixed $\mathbf{z} \in \mathbb{R}^d$ we have

$$\|\Phi \mathbf{z}\|_2^2 = \frac{1}{m} \|\mathbf{D} \mathbf{z}\|_2^2 \sim \|\mathbf{z}\|_2^2 \cdot \frac{\chi_m^2}{m}.$$

This fact and the following χ^2 tail bound are well known results. For instance, see Example 2.12 of (Wainwright, 2019). Remember that $\chi_m^2 \sim \text{Gamma}(\alpha = \frac{m}{2}, \theta = 2)$. We then have from a classic χ^2 tail bound for any $0 < \varepsilon < 1$ and any fixed $\mathbf{z} \neq \mathbf{0}$,

$$\Pr \left[\left| \frac{\|\Phi \mathbf{z}\|_2^2}{\|\mathbf{z}\|_2^2} - 1 \right| \geq \varepsilon \right] \leq 2 \exp \left(-\frac{\varepsilon^2}{8} m \right).$$

Equivalently,

$$\Pr \left[\left| \|\Phi \mathbf{z}\|_2^2 - \|\mathbf{z}\|_2^2 \right| > \varepsilon \|\mathbf{z}\|_2^2 \right] \leq 2 \exp \left(-\frac{\varepsilon^2}{8} m \right).$$

Then for any $(\mathbf{x}, \mathbf{y}) \in \mathbb{S}^{d-1} \times \mathbb{S}^{d-1}$,

$$\langle \Phi \mathbf{x}, \Phi \mathbf{y} \rangle - \langle \mathbf{x}, \mathbf{y} \rangle = \frac{1}{4} \left(\|\Phi(\mathbf{x} + \mathbf{y})\|_2^2 - \|\mathbf{x} + \mathbf{y}\|_2^2 \right) - \frac{1}{4} \left(\|\Phi(\mathbf{x} - \mathbf{y})\|_2^2 - \|\mathbf{x} - \mathbf{y}\|_2^2 \right).$$

If simultaneously

$$\left| \|\Phi(\mathbf{x} + \mathbf{y})\|_2^2 - \|\mathbf{x} + \mathbf{y}\|_2^2 \right| \leq \varepsilon \|\mathbf{x} + \mathbf{y}\|_2^2, \quad \left| \|\Phi(\mathbf{x} - \mathbf{y})\|_2^2 - \|\mathbf{x} - \mathbf{y}\|_2^2 \right| \leq \varepsilon \|\mathbf{x} - \mathbf{y}\|_2^2,$$

then, using $\|\mathbf{x}\| = \|\mathbf{y}\| = 1$,

$$|\langle \Phi \mathbf{x}, \Phi \mathbf{y} \rangle - \langle \mathbf{x}, \mathbf{y} \rangle| \leq \frac{\varepsilon}{4} (\|\mathbf{x} + \mathbf{y}\|_2^2 + \|\mathbf{x} - \mathbf{y}\|_2^2) = \frac{\varepsilon}{4} (2\|\mathbf{x}\|_2^2 + 2\|\mathbf{y}\|_2^2) = \varepsilon.$$

Let A_z denote the event that $\left| \|\Phi \mathbf{z}\|_2^2 - \|\mathbf{z}\|_2^2 \right| > \varepsilon \|\mathbf{z}\|_2^2$ for a fixed $\mathbf{z} \in \mathcal{S}_\pm$. Then $\Pr[A_z] \leq 2e^{-(\varepsilon^2/8)m}$. If none of the events $\{A_z\}_{z \in \mathcal{S}_\pm}$ occur, the bound in the previous step holds for all $(\mathbf{x}, \mathbf{y}) \in \mathcal{P}$. Therefore,

$$\Pr \left[\exists (\mathbf{x}, \mathbf{y}) \in \mathcal{P} : |\langle \Phi \mathbf{x}, \Phi \mathbf{y} \rangle - \langle \mathbf{x}, \mathbf{y} \rangle| > \varepsilon \right] \leq \sum_{z \in \mathcal{S}_\pm} \Pr[A_z] \leq 2|\mathcal{S}_\pm| \exp \left(-\frac{\varepsilon^2}{8} m \right),$$

upon union bounding over all $(\mathbf{x}, \mathbf{y}) \in \mathcal{S}_\pm$ which proves the claim. □

B.10.9 Proof of Theorem B.5.2

Proof. From our definition of $\rho_{\min}$ (recall that $\mathbf{a}_{i,j} = \tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j$ and $\mathbf{b}_i = \tilde{\mathbf{u}}_i$)

$$\rho_{\min} = \min_{i \neq j} \frac{\langle \mathbf{a}_{ij}, \mathbf{b}_i \rangle}{\|\mathbf{a}_{ij}\| \|\mathbf{b}_i\|} = \min_{i \neq j} \frac{\langle \tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j, \tilde{\mathbf{v}}_i \rangle}{\|\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j\|} = \min_{i \neq j} \sqrt{\frac{1 - \langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j \rangle}{2}}.$$

Note that $\|\tilde{\mathbf{u}}_i\| = 1$.

Let $\mu := \max_{i < j} \langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j \rangle$; since the map $\mathbf{x} \mapsto \sqrt{(1 - \mathbf{x})/2}$ is decreasing on $(-1, 1)$,

$$\rho_{\min} \geq \sqrt{\frac{1 - \mu}{2}}. \quad (27)$$

To control μ , fix $\mathbf{a} \in \mathbb{S}^{d-1}$ and let $\mathbf{X} \sim \text{Unif}(\mathbb{S}^{d-1})$. The function $f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{a} \rangle$ is 1-Lipschitz on $\mathbb{S}^{d-1}$ (geodesic metric) and $\mathbb{E}[f] = 0$ by symmetry. By Theorem 3 of (Aubrun et al., 2024), for all $t > 0$,

$$\Pr\{\langle \mathbf{X}, \mathbf{a} \rangle \geq t\} \leq e^{-dt^2/2}. \quad (28)$$

Conditioning on $\tilde{\mathbf{v}}_j$ and applying Equation (28) with $\mathbf{X} = \tilde{\mathbf{v}}_i$, $\mathbf{a} = \tilde{\mathbf{v}}_j$ yields, for each unordered pair $\{i, j\}$, $\Pr\{\langle \tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j \rangle \geq t\} \leq e^{-dt^2/2}$. Union-bounding over the $\binom{|\mathbf{K}|}{2}$ pairs gives

$$\Pr\{\mu \geq t\} \leq \binom{|\mathbf{K}|}{2} e^{-dt^2/2}.$$

Hence with probability at least $1 - \delta$,

$$\mu \leq \sqrt{\frac{2}{d} \ln \frac{\binom{|\mathbf{K}|}{2}}{\delta}}. \quad (29)$$

Combining 27–29 yields the stated bound. $\square$

B.10.10 Proof of Theorem B.5.5

Proof. Let $\mathbf{Z}_{ik} := \sqrt{d} \xi_{ik}$. Then $\|\mathbf{Z}_{ik}\|_{\psi_2} \leq K$ and $\mathbb{E}[\mathbf{Z}_{ik}^2] = 1$. Note that we also have¹⁹ $\|\mathbf{Z}^2\|_{\psi_1} \leq \|\mathbf{Z}\|_{\psi_2}^2 \leq K^2$. From the definition of the sub-exponential norm²⁰ we have that $\|1\|_{\psi_1} = 1/\ln 2$, so

$$\|\mathbf{Z}_{ik}^2 - 1\|_{\psi_1} \leq \|\mathbf{Z}_{ik}^2\|_{\psi_1} + \|1\|_{\psi_1} \leq K^2 + \frac{1}{\ln 2}$$

Since $\|\tilde{\mathbf{v}}_i\|^2 - 1 = \frac{1}{d} \sum_{k=1}^d (\mathbf{Z}_{ik}^2 - 1)$, we apply the Bernstein bound for sub-exponentials²¹ to find, for all $\eta > 0$,

$$\Pr(\|\tilde{\mathbf{v}}_i\|^2 - 1 \geq \eta) \leq 2 \exp\left(-c_B d \min\left\{\frac{\eta^2}{(K^2 + \frac{1}{\ln 2})^2}, \frac{\eta}{K^2 + \frac{1}{\ln 2}}\right\}\right).$$

Union bound over $i \in [|\mathbf{K}|]$ ²². Using $|\sqrt{1+u} - 1| \leq |u|$ ($u > -1$), with probability $\geq 1 - \delta/2$,

$$|\|\tilde{\mathbf{v}}_i\| - 1| \leq \varepsilon_{|\mathbf{K}|} \quad \text{for all } i, \quad \varepsilon_{|\mathbf{K}|} := (K^2 + \frac{1}{\ln 2}) \max\left(\sqrt{\frac{1}{c_B d} \ln \frac{4|\mathbf{K}|}{\delta}}, \frac{1}{c_B d} \ln \frac{4|\mathbf{K}|}{\delta}\right)$$

We now find a bound for $\langle \tilde{\mathbf{v}}_j, \mathbf{u}_i \rangle$. Condition on $\mathbf{u}_i$. Then for $j \neq i$,

$$\langle \tilde{\mathbf{v}}_j, \mathbf{u}_i \rangle = \sum_{k=1}^d \mathbf{u}_{ik} \xi_{jk}$$

¹⁹This is well known. For instance, it follows directly from Lemma 2.8.6 of (Vershynin, 2018)

²⁰Here we use the usual definition $\|\mathbf{X}\|_{\psi_1} := \inf\{t > 0 : \mathbb{E}[\exp(|\mathbf{X}|/t)] \leq 2\}$

²¹See Theorem 1.2.7 of (Chafaï et al., 2012). This text uses the slightly different Orlicz norm $\|\mathbf{X}\|_{\psi_1}^{(e)} = \inf\{c > 0 : \psi(|\mathbf{X}|/c) \leq \psi(1)\}$ where ψ is some Orlicz function. Recall that our definition has been $\|\mathbf{X}\|_{\psi_1} = \inf\{c > 0 : \exp(|\mathbf{X}|/c) \leq 2\}$. Fortunately, if we set $\psi_1(\mathbf{x}) = \exp(|\mathbf{x}|^\alpha) - 1$ it follows that $\{c > 0 : \exp(|\mathbf{X}|/c) \leq 2\} \subseteq \{c > 0 : \exp(|\mathbf{X}|/c) \leq e\}$, and after taking infimums we have $\|\mathbf{X}\|_{\psi_1}^{(e)} \leq \|\mathbf{X}\|_{\psi_1}$. So we may use the bound as if it were our familiar norm.

²²To find $\varepsilon_{|\mathbf{K}|}$, take the right hand side of the above equation and set it less than or equal to $\delta/2$. Solving for η yields $\varepsilon_{|\mathbf{K}|}$.

is a sum of independent centered subgaussians with $\|\mathbf{u}_{ik}\xi_{jk}\|_{\psi_2} \leq |\mathbf{u}_{ik}|K/\sqrt{d}$. By Theorem 1.1 of (Leskelä & Zhukov, 2025), the corresponding variance proxies are $\sigma_k^2 = (\sqrt{\ln 2} K |\mathbf{u}_{ik}|/\sqrt{d})^2$. The Hoeffding bound for sub-gaussians²³ gives for any $t \geq 0$,

$$\Pr\left(|\langle \tilde{\mathbf{v}}_j, \mathbf{u}_i \rangle| \geq t \mid \mathbf{u}_i\right) \leq 2 \exp\left(-\frac{t^2}{2 \sum_k \sigma_k^2}\right) = 2 \exp\left(-\frac{t^2}{2(\ln 2) K^2/d}\right),$$

since $\sum_k \mathbf{u}_{ik}^2 = 1$. Removing the conditioning and union-bounding over ordered pairs (i, j) shows that, with probability $\geq 1 - \delta/2$,

$$|\langle \tilde{\mathbf{v}}_j, \mathbf{u}_i \rangle| \leq t_{|\mathbf{K}|} \quad \text{for all } i \neq j, \quad t_{|\mathbf{K}|} := K \sqrt{\frac{2 \ln 2}{d} \ln \frac{4|\mathbf{K}|(|\mathbf{K}| - 1)}{\delta}}.$$

On the intersection of the two events (probability $\geq 1 - \delta$), for every $i \neq j$,

$$\langle \tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j, \mathbf{u}_i \rangle = \|\tilde{\mathbf{v}}_i\| - \langle \tilde{\mathbf{v}}_j, \mathbf{u}_i \rangle \geq 1 - \varepsilon_{|\mathbf{K}|} - t_{|\mathbf{K}|}, \quad \|\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j\| \leq \|\tilde{\mathbf{v}}_i\| + \|\tilde{\mathbf{v}}_j\| \leq 2(1 + \varepsilon_{|\mathbf{K}|}).$$

Therefore $(\rho_{\min})_{ij} \geq \frac{1 - \varepsilon_{|\mathbf{K}|} - t_{|\mathbf{K}|}}{2(1 + \varepsilon_{|\mathbf{K}|})}$, and taking the minimum over $i \neq j$ yields the claim. $\square$

Theorem B.10.4 (Noisy decoding via JL, Rademacher case). *Let $\mathbf{D} \in \{-1, +1\}^{m \times d}$ have i.i.d. Rademacher entries ($\Pr(\mathbf{D}_{kl} = 1) = \Pr(\mathbf{D}_{kl} = -1) = \frac{1}{2}$) and set $\mathbf{M} := \frac{1}{m} \mathbf{D}^\top$. For each $i \in [N]$, let $\mathbf{v}_i, \mathbf{u}_i \in \mathbb{R}^d$ and define*

$$\rho_{\min} := \min_{i \neq j} \frac{\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle}{\|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|} > 0.$$

Let the noisy codes be

$$\mathbf{H}[i] := (\mathbf{D}\mathbf{u}_i) \odot (1 + \nu_i), \quad \nu_i \in [-\varepsilon, \varepsilon]^m, \quad \varepsilon \in [0, 1),$$

and define scores $s_{ij} := \langle \mathbf{v}_j, \mathbf{MH}[i] \rangle$. Then there is a universal constant $C > 0$ such that if

$$m \geq \frac{C}{\rho_{\min}^2} \ln \frac{4N(N-1)}{\delta},$$

then with probability at least $1 - \delta$ over $\mathbf{D}$, we have, simultaneously for all $i \neq j$,

$$s_{ii} - s_{ij} \geq \left(\frac{\rho_{\min}}{2} - 4\varepsilon\right) \|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|.$$

Proof. Set $\Phi := \frac{1}{\sqrt{m}} \mathbf{D}$ and $\mathbf{E} := \Phi^\top \Phi - \mathbf{I}$. For $i \neq j$, write

$$\mathbf{a}_{ij} := \mathbf{v}_i - \mathbf{v}_j, \quad \mathbf{b}_i := \mathbf{u}_i.$$

Let $\mathbf{g}_i := \mathbf{D}\mathbf{u}_i$ and $\Delta_i := \mathbf{g}_i \odot \nu_i$, so $\mathbf{H}[i] = \mathbf{g}_i + \Delta_i$. Then

$$\mathbf{MH}[i] = \frac{1}{m} \mathbf{D}^\top (\mathbf{g}_i + \Delta_i) = \Phi^\top \Phi \mathbf{b}_i + \frac{1}{m} \mathbf{D}^\top \Delta_i = (\mathbf{I} + \mathbf{E}) \mathbf{b}_i + \frac{1}{m} \mathbf{D}^\top \Delta_i,$$

and the score gap is

$$s_{ii} - s_{ij} = \langle \mathbf{a}_{ij}, \mathbf{MH}[i] \rangle = \mathbf{a}_{ij}^\top \mathbf{b}_i + \mathbf{a}_{ij}^\top \mathbf{E} \mathbf{b}_i + \frac{1}{m} (\mathbf{D} \mathbf{a}_{ij})^\top \Delta_i. \quad (30)$$

Margin term. By the definition of $\rho_{\min}$,

$$\mathbf{a}_{ij}^\top \mathbf{b}_i = \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle \geq \rho_{\min} \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| \quad \forall i \neq j. \quad (31)$$

²³See Proposition 2.5 of (Wainwright, 2019)

JL event (inner products and norms). Define

$$\hat{\mathbf{a}}_{ij} := \frac{\mathbf{a}_{ij}}{\|\mathbf{a}_{ij}\|}, \quad \hat{\mathbf{b}}_i := \frac{\mathbf{b}_i}{\|\mathbf{b}_i\|},$$

and consider the finite set of unit-vector pairs

$$\mathcal{P} := \{(\hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_i) : i \in [N], j \neq i\} \cup \{(\hat{\mathbf{x}}, \hat{\mathbf{x}}) : \mathbf{x} \in X\},$$

where $X := \{\mathbf{a}_{ij} : i \neq j\} \cup \{\mathbf{b}_i : i \in [N]\}$. Since the rows of Φ are isotropic subgaussian (Rademacher), the Johnson–Lindenstrauss lemma implies:

for $\eta := \rho_{\min}/2$, if

$$m \geq \frac{C}{\rho_{\min}^2} \ln \frac{4N(N-1)}{\delta},$$

then with probability at least $1 - \delta$,

$$|\langle \Phi \mathbf{x}, \Phi \mathbf{y} \rangle - \langle \mathbf{x}, \mathbf{y} \rangle| \leq \eta \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{P}.$$

Following from Theorem B.1.2.

On this event, we get:

(i) For $(\mathbf{x}, \mathbf{y}) = (\hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_i)$,

$$|\hat{\mathbf{a}}_{ij}^\top \mathbf{E} \hat{\mathbf{b}}_i| = |\langle \Phi \hat{\mathbf{a}}_{ij}, \Phi \hat{\mathbf{b}}_i \rangle - \langle \hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_i \rangle| \leq \frac{\rho_{\min}}{2},$$

so

$$|\mathbf{a}_{ij}^\top \mathbf{E} \mathbf{b}_i| \leq \frac{\rho_{\min}}{2} \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| \quad \forall i \neq j. \quad (32)$$

(ii) For $(\mathbf{x}, \mathbf{y}) = (\hat{\mathbf{x}}, \hat{\mathbf{x}})$,

$$|\|\Phi \hat{\mathbf{x}}\|^2 - 1| = |\langle \Phi \hat{\mathbf{x}}, \Phi \hat{\mathbf{x}} \rangle - 1| \leq \frac{\rho_{\min}}{2} \leq 1,$$

so $\|\Phi \hat{\mathbf{x}}\| \leq \sqrt{2} \leq 2$ and hence

$$\|\mathbf{D} \mathbf{x}\| = \sqrt{m} \|\Phi \mathbf{x} / \|\mathbf{x}\|\| \cdot \|\mathbf{x}\| \leq 2\sqrt{m} \|\mathbf{x}\| \quad \forall \mathbf{x} \in X. \quad (33)$$

Noise term. Since $|\nu_{i,k}| \leq \varepsilon$, we have

$$|\Delta_{i,k}| = |\mathbf{g}_{i,k} \nu_{i,k}| \leq \varepsilon |\mathbf{g}_{i,k}|, \quad \Rightarrow \quad \|\Delta_i\| \leq \varepsilon \|\mathbf{g}_i\| = \varepsilon \|\mathbf{D} \mathbf{b}_i\|.$$

By Cauchy–Schwarz and equation 38,

$$|(\mathbf{D} \mathbf{a}_{ij})^\top \Delta_i| \leq \|\mathbf{D} \mathbf{a}_{ij}\| \|\Delta_i\| \leq \varepsilon \|\mathbf{D} \mathbf{a}_{ij}\| \|\mathbf{D} \mathbf{b}_i\| \leq \varepsilon (2\sqrt{m} \|\mathbf{a}_{ij}\|)(2\sqrt{m} \|\mathbf{b}_i\|),$$

so

$$\left| \frac{1}{m} (\mathbf{D} \mathbf{a}_{ij})^\top \Delta_i \right| \leq 4\varepsilon \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| \quad \forall i \neq j. \quad (34)$$

Conclusion. Conditioning on the JL event, combining equation 36, equation 37, and equation 39 in equation 35 gives, for all $i \neq j$,

$$\begin{aligned} s_{ii} - s_{ij} &\geq \rho_{\min} \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| - \frac{\rho_{\min}}{2} \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| - 4\varepsilon \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| \\ &= \left(\frac{\rho_{\min}}{2} - 4\varepsilon \right) \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\|. \end{aligned}$$

Since $\mathbf{a}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ and $\mathbf{b}_i = \mathbf{u}_i$, this is exactly

$$s_{ii} - s_{ij} \geq \left(\frac{\rho_{\min}}{2} - 4\varepsilon \right) \|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|,$$

as claimed. □

Theorem B.10.5 (Polynomial precision for encoder parameters). *Let F be the number of facts, and assume the noisy decoding theorem above holds for some choice of m (so that, for any codes whose noise is at most a fixed constant multiple of $\rho_{\min}$, decoding is still correct).*

Assume the following polynomial bounds:

(i) (Margin) $\rho_{\min} \geq 1/\text{poly}(F)$.

(ii) (Lipschitz in parameters) For each key k_i and all encoder parameter vectors θ, θ' ,

$$\|\text{enc}_{\theta}(k_i) - \text{enc}_{\theta'}(k_i)\| \leq L(F) \|\theta - \theta'\| \quad \text{with } L(F) \leq \text{poly}(F).$$

(iii) (Parameter count) The number of encoder parameters satisfies $P \leq \text{poly}(F)$.

(iv) (Magnitude) There is an encoder θ_* such that $H_{\star}[i] := \text{enc}_{\theta_{\star}}(k_i) = \mathbf{D}\mathbf{u}_i$ and $\|\theta_{\star}\|_{\infty} \leq \text{poly}(F)$.

Then there exists a constant $c > 0$ such that if we quantize each coordinate of θ_* to the grid $F^{-c}\mathbb{Z}$, obtaining $\tilde{\theta}$, the corresponding codes $\tilde{H}[i] := \text{enc}_{\tilde{\theta}}(k_i)$ still satisfy the conditions of the noisy decoding theorem and hence decode all F facts correctly. In particular, each encoder parameter requires only $O(\log F)$ bits of precision.

Proof. Step 1: Allowed code noise. From the noisy decoding theorem, there is a constant $c_0 > 0$ such that, if the code for fact i is perturbed by at most $c_0\rho_{\min}$ in an appropriate sense (as in the theorem's proof), then the score margin remains positive:

$$s_{ii} - s_{ij} \geq \Omega(\rho_{\min}) \|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|.$$

Thus the encoder codes are robust to perturbations of size $\Theta(\rho_{\min})$. Using (i), we have

$$\rho_{\min} \geq \frac{1}{\text{poly}(F)},$$

so the allowed code noise is at least $1/\text{poly}(F)$.

Step 2: From parameter perturbation to code perturbation. Let θ_* be the ideal encoder parameters and $\tilde{\theta}$ any other parameter vector. For each key k_i , define the code perturbation

$$\Delta_i := \text{enc}_{\tilde{\theta}}(k_i) - \text{enc}_{\theta_{\star}}(k_i).$$

By the Lipschitz assumption (ii),

$$\|\Delta_i\| = \|\text{enc}_{\tilde{\theta}}(k_i) - \text{enc}_{\theta_{\star}}(k_i)\| \leq L(F) \|\tilde{\theta} - \theta_{\star}\| \quad \forall i.$$

To keep the codes within the robustness radius from Step 1, it suffices to impose

$$\|\Delta_i\| \leq c_0\rho_{\min} \quad \forall i.$$

A sufficient condition is therefore

$$\|\tilde{\theta} - \theta_{\star}\| \leq \delta(F) := \frac{c_0\rho_{\min}}{L(F)}.$$

Using (i) and (ii), we obtain

$$\delta(F) \geq \frac{c_0}{\text{poly}(F) \text{poly}(F)} = \frac{1}{\text{poly}(F)}.$$

So there is a ball of radius at least $1/\text{poly}(F)$ around θ_* in parameter space such that any $\tilde{\theta}$ in this ball produces codes that the noisy decoding theorem can tolerate.

Step 3: Quantization and choice of grid size. Now quantize each coordinate of θ_* to a grid of step size $\Delta > 0$, obtaining $\tilde{\theta}$. Each coordinate changes by at most $\Delta/2$, so

$$\|\tilde{\theta} - \theta_{\star}\|_2 \leq \sqrt{P} \frac{\Delta}{2}.$$

To guarantee $\|\tilde{\theta} - \theta_\star\| \leq \delta(F)$, it is enough to choose Δ so that

$$\sqrt{P} \frac{\Delta}{2} \leq \delta(F) \iff \Delta \leq \frac{2\delta(F)}{\sqrt{P}}.$$

Using $\delta(F) \geq 1/\text{poly}(F)$ and $P \leq \text{poly}(F)$ from (iii), we get

$$\frac{2\delta(F)}{\sqrt{P}} \geq \frac{1}{\text{poly}(F)}.$$

Thus the admissible step size Δ can be as large as $1/\text{poly}(F)$. In particular, we may pick

$$\Delta := F^{-c}$$

for some constant $c > 0$ large enough so that $\Delta \leq 2\delta(F)/\sqrt{P}$. This ensures $\|\tilde{\theta} - \theta_\star\| \leq \delta(F)$ and, by Step 2, that the induced code perturbations are within the noise budget of the noisy decoding theorem. Hence decoding remains correct.

Step 4: Bit complexity. By (iv), each parameter lies in an interval of length at most $\text{range} \leq 2/\text{poly}(F)$. With grid spacing $\Delta = F^{-c} = 1/\text{poly}(F)$, the number of representable levels per parameter is at most

$$\frac{\text{range}}{\Delta} \leq \frac{\text{poly}(F)}{1/\text{poly}(F)} = \text{poly}(F).$$

Therefore the number of bits per parameter is

$$\log_2 \left(\frac{\text{range}}{\Delta} \right) = O(\log \text{poly}(F)) = O(\log F).$$

This proves that encoder parameters require only $O(\log F)$ bits of precision. $\square$

Note that the last part (assumption 4) is true because σ is analytic, which implies that it is continuously differentiable.

Theorem B.10.6 (Encoder is Lipschitz in the parameters). *Fix a number of facts F and keys $\{k_i\}_{i=1}^F \subset \mathbb{R}^d$. Consider the scalar-output gated encoder*

$$\mathbf{enc}_\theta(\mathbf{x}) = \mathbf{1}_h^\top [\sigma(\mathbf{G}\mathbf{x}) \odot (\mathbf{A}\mathbf{x})] = \sum_{r=1}^h \sigma(\langle \mathbf{g}_r, \mathbf{x} \rangle) \langle \mathbf{a}_r, \mathbf{x} \rangle,$$

where $\mathbf{A}, \mathbf{G} \in \mathbb{R}^{h \times d}$ have rows $\mathbf{a}_r^\top, \mathbf{g}_r^\top$, and $\theta \in \mathbb{R}^P$ is the vector of all entries of $\mathbf{A}, \mathbf{G}$.

Assume:

- (i) $\|k_i\|_2 \leq R_{\mathbf{x}}(F)$ for all i , with $R_{\mathbf{x}}(F) \leq \text{poly}(F)$.
- (ii) $\|\theta\|_2 \leq R_\theta(F)$, with $R_\theta(F) \leq \text{poly}(F)$.
- (iii) The width and input dimension satisfy $h, d \leq \text{poly}(F)$, so that $P = 2hd \leq \text{poly}(F)$.
- (iv) The activation $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is continuously differentiable and on the interval $[-B(F), B(F)]$ with $B(F) := R_\theta(F)R_{\mathbf{x}}(F)$ we have

$$|\sigma(t)| \leq C_\sigma, \quad |\sigma'(t)| \leq C'_\sigma \quad \forall t \in [-B(F), B(F)],$$

for some constants C_σ, C'_σ independent of F .

Then for each key k_i there exists a constant $L(F) \leq \text{poly}(F)$ such that for all parameter vectors θ, θ' with $\|\theta\|_2, \|\theta'\|_2 \leq R_\theta(F)$,

$$|\mathbf{enc}_\theta(k_i) - \mathbf{enc}_{\theta'}(k_i)| \leq L(F) \|\theta - \theta'\|_2.$$

In particular, $\mathbf{enc}_\theta(k_i)$ is Lipschitz in θ with Lipschitz constant at most polynomial in F .

Proof. Fix i and write $\mathbf{x} := k_i$. For fixed $\mathbf{x}$, view $\mathbf{enc}_\theta(\mathbf{x})$ as a function $\mathbb{R}^P \rightarrow \mathbb{R}$ of the parameter vector θ . Its partial derivatives are, for each $r \in [h]$ and $\ell \in [d]$,

$$\frac{\partial \mathbf{enc}_\theta(\mathbf{x})}{\partial \mathbf{A}_{r\ell}} = \sigma(\langle \mathbf{g}_r, \mathbf{x} \rangle) \mathbf{x}_\ell, \quad \frac{\partial \mathbf{enc}_\theta(\mathbf{x})}{\partial \mathbf{G}_{r\ell}} = \sigma'(\langle \mathbf{g}_r, \mathbf{x} \rangle) \langle \mathbf{a}_r, \mathbf{x} \rangle \mathbf{x}_\ell.$$

On the parameter ball $\|\theta\|_2 \leq R_\theta(F)$ and with $\|\mathbf{x}\| \leq R_{\mathbf{x}}(F)$ we have $|\langle \mathbf{g}_r, \mathbf{x} \rangle| \leq \|\mathbf{g}_r\| \|\mathbf{x}\| \leq R_\theta(F) R_{\mathbf{x}}(F) = B(F)$, so by assumption $|\sigma(\langle \mathbf{g}_r, \mathbf{x} \rangle)| \leq C_\sigma$ and $|\sigma'(\langle \mathbf{g}_r, \mathbf{x} \rangle)| \leq C'_\sigma$. Moreover $|\mathbf{x}_\ell| \leq R_{\mathbf{x}}(F)$ and

$$|\langle \mathbf{a}_r, \mathbf{x} \rangle| \leq \|\mathbf{a}_r\| \|\mathbf{x}\| \leq R_\theta(F) R_{\mathbf{x}}(F).$$

Hence

$$\left| \frac{\partial \mathbf{enc}_\theta(\mathbf{x})}{\partial \mathbf{A}_{r\ell}} \right| \leq C_\sigma R_{\mathbf{x}}(F), \quad \left| \frac{\partial \mathbf{enc}_\theta(\mathbf{x})}{\partial \mathbf{G}_{r\ell}} \right| \leq C'_\sigma R_\theta(F) R_{\mathbf{x}}(F)^2.$$

The gradient $\nabla_\theta \mathbf{enc}_\theta(\mathbf{x}) \in \mathbb{R}^P$ collects all these partial derivatives, so its Euclidean norm satisfies

$$\|\nabla_\theta \mathbf{enc}_\theta(\mathbf{x})\|_2^2 \leq P \cdot (\max\{C_\sigma R_{\mathbf{x}}(F), C'_\sigma R_\theta(F) R_{\mathbf{x}}(F)^2\})^2 \leq C \text{poly}(F)^2$$

for some constant $C > 0$, using $P \leq \text{poly}(F)$ and $R_{\mathbf{x}}(F), R_\theta(F) \leq \text{poly}(F)$. Thus there exists $L(F) \leq C^{1/2} \text{poly}(F)$ such that

$$\|\nabla_\theta \mathbf{enc}_\theta(\mathbf{x})\|_2 \leq L(F) \quad \text{for all } \|\theta\|_2 \leq R_\theta(F).$$

For any θ, θ' with $\|\theta\|_2, \|\theta'\|_2 \leq R_\theta(F)$, the mean value inequality in $\mathbb{R}^P$ yields

$$|\mathbf{enc}_\theta(\mathbf{x}) - \mathbf{enc}_{\theta'}(\mathbf{x})| \leq \sup_{\tilde{\theta} \text{ on the segment } [\theta, \theta']} \|\nabla_{\tilde{\theta}} \mathbf{enc}_{\tilde{\theta}}(\mathbf{x})\|_2 \cdot \|\theta - \theta'\|_2 \leq L(F) \|\theta - \theta'\|_2.$$

Since $L(F) \leq \text{poly}(F)$ by construction, this proves the claim. $\square$

Lemma B.10.7 (Encoder weight norm bound). *Fix an output coordinate j and consider the linear system*

$$\mathbf{M} \mathbf{a} = \mathbf{o},$$

where $\mathbf{M} \in \mathbb{R}^{F \times dh}$ and $\mathbf{a} = \text{vec}(\mathbf{A}) \in \mathbb{R}^{dh}$. Assume:

(i) The i -th row of $\mathbf{M}$ is

$$\mathbf{r}_i^\top = (\sigma(\mathbf{g}_1^\top \mathbf{k}_i) \mathbf{k}_i^\top, \dots, \sigma(\mathbf{g}_h^\top \mathbf{k}_i) \mathbf{k}_i^\top),$$

where $\{\mathbf{k}_i\}_{i=1}^F$ and $\{\mathbf{g}_\ell\}_{\ell=1}^h$ are independent subgaussian random vectors in $\mathbb{R}^d$, and σ is analytic and non-constant.

(ii) The covariance $\Sigma_{\text{row}} := \mathbb{E}[\mathbf{r}_i \mathbf{r}_i^\top]$ satisfies $\lambda_{\min}(\Sigma_{\text{row}}) \geq \lambda_0 > 0$ and $\lambda_{\max}(\Sigma_{\text{row}}) \leq \Lambda_0 < \infty$, with λ_0, Λ_0 independent of F .

(iii) The targets $\mathbf{o} \in \mathbb{R}^F$ obey $|o_i| \leq B(F)$ for all i , where $B(F) \leq \text{poly}(F)$.

(iv) $F \geq C_0 dh$ for a sufficiently large absolute constant C_0 .

Let $\mathbf{a}_\star$ be the minimum- ℓ_2 -norm solution of $\mathbf{M} \mathbf{a} = \mathbf{o}$ (i.e. $\mathbf{a}_\star = \mathbf{M}^\dagger \mathbf{o}$). Then

$$\|\mathbf{a}_\star\|_2 \leq \text{poly}(F).$$

Proof. Let $\tilde{\mathbf{r}}_i := \Sigma_{\text{row}}^{-1/2} \mathbf{r}_i$ and let $\tilde{\mathbf{M}} \in \mathbb{R}^{F \times dh}$ have rows $\tilde{\mathbf{r}}_i^\top$. By construction, the rows of $\tilde{\mathbf{M}}$ are independent, isotropic, subgaussian random vectors in $\mathbb{R}^{dh}$, and $\|\tilde{\mathbf{r}}_i\|_{\psi_2}$ is bounded uniformly in F .

Apply Theorem B.1.3 to $\tilde{\mathbf{M}}$ with $N = F$ and $n = dh$. There exist constants $c, C > 0$ depending only on the subgaussian norm such that, with probability at least $1 - 2\exp(-ct^2)$,

$$\sqrt{F} - C\sqrt{dh} - t \leq s_{\min}(\tilde{\mathbf{M}}) \leq s_{\max}(\tilde{\mathbf{M}}) \leq \sqrt{F} + C\sqrt{dh} + t \quad \forall t \geq 0.$$

Choose $t = \sqrt{F}/4$ and use the assumption $F \geq C_0 dh$ with C_0 large enough to obtain

$$s_{\min}(\tilde{\mathbf{M}}) \geq c_1 \sqrt{F}$$

for some constant $c_1 > 0$, with probability at least $1 - \exp(-c_2 F)$.

Since $\mathbf{M} = \tilde{\mathbf{M}} \Sigma_{\text{row}}^{1/2}$, we have

$$s_{\min}(\mathbf{M}) \geq \sqrt{\lambda_{\min}(\Sigma_{\text{row}})} s_{\min}(\tilde{\mathbf{M}}) \geq \sqrt{\lambda_0} c_1 \sqrt{F} = c_3 \sqrt{F}.$$

Furthermore,

$$\|\mathbf{o}\|_2^2 = \sum_{i=1}^F \mathbf{o}_i^2 \leq F B(F)^2, \quad \Rightarrow \quad \|\mathbf{o}\|_2 \leq \sqrt{F} B(F) \leq \text{poly}(F).$$

Let $\mathbf{a}_\star$ be the minimum-norm solution $\mathbf{M}\mathbf{a} = \mathbf{o}$, so $\mathbf{a}_\star = \mathbf{M}^\dagger \mathbf{o}$ and $\|\mathbf{M}^\dagger\|_{\text{op}} = 1/s_{\min}(\mathbf{M})$. Then

$$\|\mathbf{a}_\star\|_2 = \|\mathbf{M}^\dagger \mathbf{o}\|_2 \leq \|\mathbf{M}^\dagger\|_{\text{op}} \|\mathbf{o}\|_2 = \frac{\|\mathbf{o}\|_2}{s_{\min}(\mathbf{M})} \leq \frac{\sqrt{F} B(F)}{c_3 \sqrt{F}} = \frac{B(F)}{c_3} \leq \text{poly}(F).$$

This holds for each output coordinate j , and stacking the corresponding vectors $\mathbf{a}_\star^{(j)}$ over $m = \text{poly}(F)$ coordinates preserves a $\text{poly}(F)$ bound on the encoder parameter norm. $\square$

Lemma B.10.8 (Row covariance is well-conditioned under rotationally invariant model). *Fix $d, h \in \mathbb{N}$ and let*

$$\mathbf{k} \in \mathbb{R}^d \quad \text{and} \quad \mathbf{g}_1, \dots, \mathbf{g}_h \in \mathbb{R}^d$$

be random vectors such that:

- (i) $\mathbf{k}$ has a rotationally invariant distribution with $\mathbb{E}[\mathbf{k}] = 0$ and $\mathbb{E}[\mathbf{k}\mathbf{k}^\top] = \frac{1}{d}\mathbf{I}_d$;
- (ii) $\mathbf{g}_1, \dots, \mathbf{g}_h$ are i.i.d. $\mathcal{N}(0, \mathbf{I}_d/d)$, independent of $\mathbf{k}$;
- (iii) $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is a non-constant measurable function with $\mathbb{E}[\sigma(\mathbf{g}_1^\top \mathbf{k})^2] < \infty$.

Define the random row vector $\mathbf{r}^\top \in \mathbb{R}^{dh}$ by

$$\mathbf{r}^\top := (\sigma(\mathbf{g}_1^\top \mathbf{k})\mathbf{k}^\top, \dots, \sigma(\mathbf{g}_h^\top \mathbf{k})\mathbf{k}^\top),$$

and let

$$\Sigma_{\text{row}} := \mathbb{E}[\mathbf{r}\mathbf{r}^\top] \in \mathbb{R}^{dh \times dh}.$$

Then there exists a constant $c > 0$, depending only on the distributions of $\mathbf{k}$, $\mathbf{g}_\ell$, and σ (but independent of F), such that

$$\lambda_{\min}(\Sigma_{\text{row}}) = c.$$

In particular,

$$\lambda_{\min}(\Sigma_{\text{row}}) \geq F^{-C}$$

for some fixed exponent C and all F (i.e., the lower bound is $\text{poly}(F)$).

Proof. For any orthogonal $\mathbf{U} \in O(d)$, define a block-rotation $\mathbf{T}_{\mathbf{U}} : \mathbb{R}^{dh} \rightarrow \mathbb{R}^{dh}$ by

$$\mathbf{T}_{\mathbf{U}}(\mathbf{x}_1, \dots, \mathbf{x}_h) := (\mathbf{U}\mathbf{x}_1, \dots, \mathbf{U}\mathbf{x}_h), \quad \mathbf{x}_\ell \in \mathbb{R}^d.$$

By rotational invariance of $\mathbf{k}$ and Gaussianity of $\mathbf{g}_\ell$, we have

$$(\mathbf{k}, \mathbf{g}_1, \dots, \mathbf{g}_h) \sim (\mathbf{U}\mathbf{k}, \mathbf{U}\mathbf{g}_1, \dots, \mathbf{U}\mathbf{g}_h),$$

and a direct calculation shows

$$\mathbf{r}(\mathbf{U}\mathbf{k}, \mathbf{U}\mathbf{g}_1, \dots, \mathbf{U}\mathbf{g}_h) = \mathbf{T}_{\mathbf{U}} \mathbf{r}(\mathbf{k}, \mathbf{g}_1, \dots, \mathbf{g}_h).$$

Hence $\mathbf{r} \sim \mathbf{T}_{\mathbf{U}} \mathbf{r}$ for all $\mathbf{U} \in O(d)$. Taking expectations,

$$\mathbf{T}_{\mathbf{U}} \Sigma_{\text{row}} \mathbf{T}_{\mathbf{U}}^\top = \mathbb{E}[\mathbf{T}_{\mathbf{U}} \mathbf{r} \mathbf{r}^\top \mathbf{T}_{\mathbf{U}}^\top] = \mathbb{E}[\mathbf{r} \mathbf{r}^\top] = \Sigma_{\text{row}}, \quad \forall \mathbf{U} \in O(d).$$

Thus Σ_{row} commutes with every block-rotation $\mathbf{T}_{\mathbf{U}}$. By Schur's lemma / symmetry, the only matrices with this property are scalar multiples of the identity, so

$$\Sigma_{\text{row}} = c \mathbf{I}_{dh}$$

for some $c \geq 0$. Since σ is non-constant and $\mathbf{k}, \mathbf{g}_\ell$ are non-degenerate, we have $\text{Var}(\langle \mathbf{r}, \mathbf{u} \rangle) = \mathbf{u}^\top \Sigma_{\text{row}} \mathbf{u} > 0$ for some unit $\mathbf{u}$, forcing $c > 0$. Therefore

$$\lambda_{\min}(\Sigma_{\text{row}}) = c > 0,$$

which is a positive constant independent of F , and hence trivially satisfies $\lambda_{\min}(\Sigma_{\text{row}}) \geq F^{-C}$ for some fixed C . $\square$

B.10.11 Proof of Theorem B.8.1

Proof. The full construction can be described as $g(\mathbf{x}) = \mathbf{D}\mathbf{E}(\sigma(\mathbf{G}\mathbf{x}) \odot (\mathbf{A}\mathbf{x}))$, where $\mathbf{D} \in \mathbb{R}^{d \times m}$, $\mathbf{A}, \mathbf{G} \in \mathbb{R}^{h \times d}$, $\mathbf{E} \in \mathbb{R}^{m \times h}$ and $\mathbf{x} \in \mathbb{R}^d$. A few of these we can bound easily.

1. $\mathbf{E}$ is a matrix which contains just 1s, and thus contributes mh bits.
2. We will show in Theorem B.8.2 that $\mathbf{D}$ is a matrix which can be stored with values in $\{-1, 1\}$, which means that it can be stored using dm bits.
3. The matrices $\mathbf{G}$ and $\mathbf{A}$ are not as easy to determine how many bits they take to store since these matrices can take on continuous values. We need to prove two things. First, we need to show that the parameters of $\mathbf{G}$ and $\mathbf{A}$ are bounded. Since $\mathbf{G}$ has rows that are normal, the magnitude of the parameters of $\mathbf{G}$ are bounded with high probability by Section B.8.2. It remains to be shown that the parameters of $\mathbf{A}$ are bounded by $O(\text{poly} F)$. If this is true, then the integer part of the parameter can be represented by $O(\log \text{poly } F) = O(\log F)$ bits. This is proved in Theorem B.8.5.
4. Second, we will prove that the parameters of these two matrices can be stored with finite precision. That is, if we truncate the decimal expansion of the parameter values of each of the matrices after a certain number of places, the construction still works when each parameter only has $O(\log F)$ bits of precision. This is proved in Theorem B.8.7.

Combining all of these steps completes the proof. $\square$

B.10.12 Proof of Theorem B.8.2

Proof. Set $\Phi := \frac{1}{\sqrt{m}}\mathbf{D}$ and $\mathbf{E} := \Phi^\top \Phi - \mathbf{I}$. For $i \neq j$, write

$$\mathbf{a}_{ij} := \mathbf{v}_i - \mathbf{v}_j, \quad \mathbf{b}_i := \mathbf{u}_i.$$

Let $\mathbf{G}[i] := \mathbf{D}\mathbf{u}_i$ and $\Delta_i := \mathbf{G}[i] \odot \nu_i$, so $\mathbf{H}[i] = \mathbf{G}[i] + \Delta_i$. Then

$$\mathbf{M}\mathbf{H}[i] = \frac{1}{m}\mathbf{D}^\top (\mathbf{G}[i] + \Delta_i) = \Phi^\top \Phi \mathbf{b}_i + \frac{1}{m}\mathbf{D}^\top \Delta_i = (\mathbf{I} + \mathbf{E})\mathbf{b}_i + \frac{1}{m}\mathbf{D}^\top \Delta_i,$$

and the score gap is

$$s_{ii} - s_{ij} = \langle \mathbf{a}_{ij}, \mathbf{M}\mathbf{H}[i] \rangle = \mathbf{a}_{ij}^\top \mathbf{b}_i + \mathbf{a}_{ij}^\top \mathbf{E}\mathbf{b}_i + \frac{1}{m}(\mathbf{D}\mathbf{a}_{ij})^\top \Delta_i. \quad (35)$$

Margin term. By the definition of ρ ,

$$\mathbf{a}_{ij}^\top \mathbf{b}_i = \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{u}_i \rangle \geq \rho \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| \quad \forall i \neq j. \quad (36)$$

JL event (inner products and norms). Define

$$\hat{\mathbf{a}}_{ij} := \frac{\mathbf{a}_{ij}}{\|\mathbf{a}_{ij}\|}, \quad \hat{\mathbf{b}}_i := \frac{\mathbf{b}_i}{\|\mathbf{b}_i\|},$$

and consider the finite set of unit-vector pairs

$$\mathcal{P} := \{(\hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_i) : i \in [N], j \neq i\} \cup \{(\hat{\mathbf{x}}, \hat{\mathbf{x}}) : \mathbf{x} \in X\},$$

where

$$X := \{\mathbf{a}_{ij} : i \neq j\} \cup \{\mathbf{b}_i : i \in [N]\}.$$

Since the rows of Φ are isotropic subgaussian (Rademacher), the Johnson–Lindenstrauss lemma implies: for $\eta := \rho/2$, if

$$m \geq \frac{C}{\rho^2} \ln \frac{4N(N-1)}{\delta},$$

then with probability at least $1 - \delta$,

$$|\langle \Phi \mathbf{x}, \Phi \mathbf{y} \rangle - \langle \mathbf{x}, \mathbf{y} \rangle| \leq \eta \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{P}.$$

Following from Theorem B.1.2.

On this event, we get:

(i) For $(\mathbf{x}, \mathbf{y}) = (\hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_i)$,

$$|\hat{\mathbf{a}}_{ij}^\top \mathbf{E} \hat{\mathbf{b}}_i| = |\langle \Phi \hat{\mathbf{a}}_{ij}, \Phi \hat{\mathbf{b}}_i \rangle - \langle \hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_i \rangle| \leq \frac{\rho}{2},$$

so

$$|\mathbf{a}_{ij}^\top \mathbf{E} \mathbf{b}_i| \leq \frac{\rho}{2} \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| \quad \forall i \neq j. \quad (37)$$

(ii) For $(\mathbf{x}, \mathbf{y}) = (\hat{\mathbf{x}}, \hat{\mathbf{x}})$,

$$|\|\Phi \hat{\mathbf{x}}\|^2 - 1| = |\langle \Phi \hat{\mathbf{x}}, \Phi \hat{\mathbf{x}} \rangle - 1| \leq \frac{\rho}{2} \leq 1,$$

so $\|\Phi \hat{\mathbf{x}}\| \leq \sqrt{2} \leq 2$ and hence

$$\|\mathbf{D} \mathbf{x}\| = \sqrt{m} \|\Phi \mathbf{x} / \|\mathbf{x}\|\| \leq 2\sqrt{m} \|\mathbf{x}\| \quad \forall \mathbf{x} \in X. \quad (38)$$

Noise term. Since $|\nu_{i,k}| \leq \varepsilon$, we have

$$|\Delta_{i,k}| = |\mathbf{G}[i][k] \nu_{i,k}| \leq \varepsilon |\mathbf{G}[i][k]|, \quad \Rightarrow \quad \|\Delta_i\| \leq \varepsilon \|\mathbf{G}[i]\| = \varepsilon \|\mathbf{D} \mathbf{b}_i\|.$$

By Cauchy–Schwarz and equation 38,

$$|(\mathbf{D} \mathbf{a}_{ij})^\top \Delta_i| \leq \|\mathbf{D} \mathbf{a}_{ij}\| \|\Delta_i\| \leq \varepsilon \|\mathbf{D} \mathbf{a}_{ij}\| \|\mathbf{D} \mathbf{b}_i\| \leq \varepsilon (2\sqrt{m} \|\mathbf{a}_{ij}\|)(2\sqrt{m} \|\mathbf{b}_i\|),$$

so

$$\left| \frac{1}{m} (\mathbf{D} \mathbf{a}_{ij})^\top \Delta_i \right| \leq 4\varepsilon \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| \quad \forall i \neq j. \quad (39)$$

Conclusion. On the JL event, combining equation 36, equation 37, and equation 39 in equation 35 gives, for all $i \neq j$,

$$\begin{aligned} s_{ii} - s_{ij} &\geq \rho \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| - \frac{\rho}{2} \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| - 4\varepsilon \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\| \\ &= \left(\frac{\rho}{2} - 4\varepsilon \right) \|\mathbf{a}_{ij}\| \|\mathbf{b}_i\|. \end{aligned}$$

Since $\mathbf{a}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ and $\mathbf{b}_i = \mathbf{u}_i$, this is exactly

$$s_{ii} - s_{ij} \geq \left(\frac{\rho}{2} - 4\varepsilon \right) \|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|,$$

as claimed. $\square$

B.10.13 Proof of Theorem B.8.3

Proof. When the keys are Gaussian, $\mathbf{k}_i \sim \mathcal{N}(0, \mathbf{I}_d)$, we have $\|\mathbf{k}_i\|_2^2 \sim \chi_d^2$ and standard concentration implies

$$\Pr(\|\mathbf{k}_i\|_2 \geq \sqrt{d} + t) \leq \exp(-ct^2) \quad \forall t \geq 0$$

for some absolute constant $c > 0$. (See Theorem 3.1.1) By a union bound,

$$\Pr\left(\max_{1 \leq i \leq F} \|\mathbf{k}_i\|_2 \geq \sqrt{d} + t\right) \leq F \exp(-ct^2).$$

Taking $t = \sqrt{C \log F}$ with C large enough, we obtain

$$\max_{1 \leq i \leq F} \|\mathbf{k}_i\|_2 \leq \sqrt{d} + \sqrt{C \log F}$$

with probability at least $1 - F^{-\Omega(1)}$. Thus, defining $R_{\mathbf{x}}(F) := \sqrt{d} + \sqrt{C \log F}$ and assuming $d \leq \text{poly}(F)$, we have $R_{\mathbf{x}}(F) \leq \text{poly}(F)$, so the deterministic assumption $\|\mathbf{k}_i\|_2 \leq R_{\mathbf{x}}(F)$ for all i holds with high probability. $\square$

B.10.14 Proof of Theorem B.8.4

Proof. For any orthogonal $\mathbf{U} \in O(d) := \{\mathbf{V} \in \mathbb{R}^{d \times d} : \mathbf{V}^\top \mathbf{V} = \mathbf{I}_d\}$, define a block-rotation $T_{\mathbf{U}} : \mathbb{R}^{dh} \rightarrow \mathbb{R}^{dh}$ by

$$T_{\mathbf{U}}(\mathbf{x}_1, \dots, \mathbf{x}_h) := (\mathbf{U}\mathbf{x}_1, \dots, \mathbf{U}\mathbf{x}_h), \quad \mathbf{x}_\ell \in \mathbb{R}^d.$$

By rotational invariance of $\mathbf{k}$ and $\mathbf{G}[\ell]$, we have

$$(\mathbf{k}, \mathbf{G}[1], \dots, \mathbf{G}[h]) \sim (\mathbf{U}\mathbf{k}, \mathbf{U}\mathbf{G}[1], \dots, \mathbf{U}\mathbf{G}[h]),$$

and a direct calculation²⁴ shows

$$\mathbf{r}(\mathbf{U}\mathbf{k}, \mathbf{U}\mathbf{G}[1], \dots, \mathbf{U}\mathbf{G}[h]) = T_{\mathbf{U}} \mathbf{r}(\mathbf{k}, \mathbf{G}[1], \dots, \mathbf{G}[h]).$$

Taking expectations,

$$T_{\mathbf{U}} \Sigma_{\text{row}} T_{\mathbf{U}}^\top = \mathbb{E}[T_{\mathbf{U}} \mathbf{r} \mathbf{r}^\top T_{\mathbf{U}}^\top] = \mathbb{E}[\mathbf{r} \mathbf{r}^\top] = \Sigma_{\text{row}}, \quad \forall \mathbf{U} \in O(d).$$

Thus Σ_{row} commutes with every block-rotation $T_{\mathbf{U}}$.

Looking at the (i, j) block of this identity $\mathbf{A}_{ij} \in \mathbb{R}^{d \times d}$ yields

$$\mathbf{U} \mathbf{A}_{ij} \mathbf{U}^\top = \mathbf{A}_{ij}, \quad \forall \mathbf{U} \in O(d). \quad (1)$$

Step 1: form of $\mathbf{A}_{ij}$. Let $M \in \mathbb{R}^{d \times d}$ be symmetric and satisfy $\mathbf{U} M \mathbf{U}^\top = M$ for all $\mathbf{U} \in O(d)$. Then, it is a well known result that $M = \lambda \mathbf{I}_d$ ²⁵.

Applying this to each symmetric $\mathbf{A}_{ij}$ in (1) gives

$$\mathbf{A}_{ij} = \lambda_{ij} \mathbf{I}_d \quad \text{for some } \lambda_{ij} \in \mathbb{R}. \quad (2)$$

Step 2: diagonal blocks. Since the $\mathbf{g}_\ell$ are i.i.d., each $\mathbf{r}_i$ has the same distribution, so $\mathbf{A}_{11} = \dots = \mathbf{A}_{hh} = c \mathbf{I}_d$ for some $c \geq 0$. Moreover,

$$c \mathbf{I}_d = \mathbf{A}_{11} = \mathbb{E}[\mathbf{r}_1 \mathbf{r}_1^\top] = \mathbb{E}[\sigma(\mathbf{g}_1^\top \mathbf{k})^2 \mathbf{k} \mathbf{k}^\top],$$

and by non-degeneracy of $(\mathbf{k}, \mathbf{g}_1)$ and non-constancy of σ we have $\mathbb{E}[\sigma(\mathbf{g}_1^\top \mathbf{k})^2 \|\mathbf{k}\|_2^2] > 0$, so $c > 0$.

Step 3: off-diagonal blocks vanish. For $i \neq j$,

$$\mathbf{A}_{ij} = \mathbb{E}[\sigma(\mathbf{g}_i^\top \mathbf{k}) \sigma(\mathbf{g}_j^\top \mathbf{k}) \mathbf{k} \mathbf{k}^\top].$$

Conditioning on $\mathbf{k}$ and using $\mathbb{E}(f(Z)Y \mid Z) = f(Z)\mathbb{E}(Y \mid Z)$, we obtain

$$\mathbf{A}_{ij} = \mathbb{E}[\mathbf{k} \mathbf{k}^\top \mathbb{E}[\sigma(\mathbf{g}_i^\top \mathbf{k}) \sigma(\mathbf{g}_j^\top \mathbf{k}) \mid \mathbf{k}]].$$

Given $\mathbf{k}$, the vectors $\mathbf{g}_i, \mathbf{g}_j$ are independent and identically distributed, hence

$$\mathbb{E}[\sigma(\mathbf{g}_i^\top \mathbf{k}) \sigma(\mathbf{g}_j^\top \mathbf{k}) \mid \mathbf{k}] = \mathbb{E}[\sigma(\mathbf{g}_1^\top \mathbf{k}) \mid \mathbf{k}]^2.$$

Let $\lambda(\mathbf{k}) := \mathbb{E}[\sigma(\mathbf{g}_1^\top \mathbf{k}) \mid \mathbf{k}]$. Assumption (iv) gives $\lambda(\mathbf{k}) = 0$ a.s., so $\lambda(\mathbf{k})^2 = 0$ a.s. and therefore

$$\mathbf{A}_{ij} = \mathbb{E}[\mathbf{k} \mathbf{k}^\top \lambda(\mathbf{k})^2] = 0, \quad i \neq j. \quad (3)$$

Combining (2), (3), and the identification of the diagonal blocks,

$$\Sigma_{\text{row}} = \text{diag}(c \mathbf{I}_d, \dots, c \mathbf{I}_d) = c \mathbf{I}_{dh},$$

so all eigenvalues of Σ_{row} equal $c > 0$. □

²⁴

$$\mathbf{r}(\mathbf{U}\mathbf{k}, \mathbf{U}\mathbf{G}[1], \dots, \mathbf{U}\mathbf{G}[h]) = (\sigma((\mathbf{U}\mathbf{G}[1])^\top \mathbf{U}\mathbf{k})(\mathbf{U}\mathbf{k})^\top, \dots, \sigma((\mathbf{U}\mathbf{G}[h])^\top \mathbf{U}\mathbf{k})(\mathbf{U}\mathbf{k})^\top).$$

Since $\mathbf{U}$ is orthogonal, $(\mathbf{U}\mathbf{G}[\ell])^\top \mathbf{U}\mathbf{k} = \mathbf{G}[\ell]^\top \mathbf{U}^\top \mathbf{U}\mathbf{k} = \mathbf{G}[\ell]^\top \mathbf{k}$, so this becomes

$$\mathbf{r}(\mathbf{U}\mathbf{k}, \mathbf{U}\mathbf{G}[1], \dots, \mathbf{U}\mathbf{G}[h]) = (\sigma(\mathbf{G}[1]^\top \mathbf{k})(\mathbf{U}\mathbf{k})^\top, \dots, \sigma(\mathbf{G}[h]^\top \mathbf{k})(\mathbf{U}\mathbf{k})^\top).$$

On the other hand, applying $T_{\mathbf{U}}$ to $\mathbf{r}(\mathbf{k}, \mathbf{G}[1], \dots, \mathbf{G}[h]) = (\sigma(\mathbf{G}[1]^\top \mathbf{k})\mathbf{k}^\top, \dots, \sigma(\mathbf{G}[h]^\top \mathbf{k})\mathbf{k}^\top)$ clearly gives the same result, so the two expressions coincide.

²⁵Theorem A.4 in (Kotelenez et al., 2008)

B.10.15 Proof of Theorem B.8.5

Proof. Let $\tilde{\mathbf{r}}_i := \Sigma_{\text{row}}^{-1/2} \mathbf{r}_i$ and let $\tilde{\mathbf{M}} \in \mathbb{R}^{F \times dh}$ have rows $\tilde{\mathbf{r}}_i^\top$. By construction, the rows of $\tilde{\mathbf{M}}$ are independent, isotropic, subgaussian random vectors in $\mathbb{R}^{dh}$, and $\|\tilde{\mathbf{r}}_i\|_{\psi_2}$ is bounded uniformly²⁶ in F .

Apply Theorem B.1.3 to $\tilde{\mathbf{M}}$ with $N = F$ and $n = dh$. There exist constants $c, C > 0$ depending only on the subgaussian norm such that, with probability at least $1 - 2\exp(-ct^2)$,

$$\sqrt{F} - C\sqrt{dh} - t \leq s_{\min}(\tilde{\mathbf{M}}) \leq s_{\max}(\tilde{\mathbf{M}}) \leq \sqrt{F} + C\sqrt{dh} + t \quad \forall t \geq 0.$$

Choose $t = \sqrt{F}/4$ and use the assumption $F \geq C_0 dh$ with C_0 large enough to obtain

$$s_{\min}(\tilde{\mathbf{M}}) \geq c_1 \sqrt{F}$$

for some constant $c_1 > 0$, with probability at least $1 - \exp(-c_2 F)$.

Since $\mathbf{M} = \tilde{\mathbf{M}} \Sigma_{\text{row}}^{1/2}$, we have

$$s_{\min}(\mathbf{M}) \geq \sqrt{\lambda_{\min}(\Sigma_{\text{row}})} s_{\min}(\tilde{\mathbf{M}}) \geq \sqrt{\lambda_0} c_1 \sqrt{F} = c_3 \sqrt{F}.$$

Furthermore,

$$\|\mathbf{o}\|_2^2 = \sum_{i=1}^F \mathbf{o}_i^2 \leq F B(F)^2, \quad \Rightarrow \quad \|\mathbf{o}\|_2 \leq \sqrt{F} B(F) \leq \text{poly}(F).$$

Let $\mathbf{a}_\star$ be the minimum-norm solution $\mathbf{M}\mathbf{a} = \mathbf{o}$, so $\mathbf{a}_\star = \mathbf{M}^\dagger \mathbf{o}$ and $\|\mathbf{M}^\dagger\|_{\text{op}} = 1/s_{\min}(\mathbf{M})$. Then

$$\|\mathbf{a}_\star\|_2 = \|\mathbf{M}^\dagger \mathbf{o}\|_2 \leq \|\mathbf{M}^\dagger\|_{\text{op}} \|\mathbf{o}\|_2 = \frac{\|\mathbf{o}\|_2}{s_{\min}(\mathbf{M})} \leq \frac{\sqrt{F} B(F)}{c_3 \sqrt{F}} = \frac{B(F)}{c_3} \leq \text{poly}(F).$$

This holds for each output coordinate j , and stacking the corresponding vectors $\mathbf{a}_\star^{(j)}$ over $m = \text{poly}(F)$ coordinates preserves a $\text{poly}(F)$ bound on the encoder parameter norm. $\square$

B.10.16 Proof of Theorem B.8.6

Proof. Fix i and write $\mathbf{x} := \mathbf{k}_i$. For fixed $\mathbf{x}$, view $\text{enc}_\theta(\mathbf{x})$ as a function $\mathbb{R}^P \rightarrow \mathbb{R}$ of the parameter vector θ . Its partial derivatives are, for each $r \in [h]$ and $\ell \in [d]$,

$$\frac{\partial \text{enc}_\theta(\mathbf{x})}{\partial \mathbf{A}_{r\ell}} = \sigma(\langle \mathbf{G}[r], \mathbf{x} \rangle) \mathbf{x}_\ell, \quad \frac{\partial \text{enc}_\theta(\mathbf{x})}{\partial \mathbf{G}_{r\ell}} = \sigma'(\langle \mathbf{G}[r], \mathbf{x} \rangle) \langle \mathbf{a}_r, \mathbf{x} \rangle \mathbf{x}_\ell.$$

On the parameter ball $\|\theta\|_2 \leq R_\theta(F)$ and with $\|\mathbf{x}\| \leq R_{\mathbf{x}}(F)$ we have $|\langle \mathbf{G}[r], \mathbf{x} \rangle| \leq \|\mathbf{G}[r]\| \|\mathbf{x}\| \leq R_\theta(F) R_{\mathbf{x}}(F) = B(F)$, so by assumption $|\sigma(\langle \mathbf{G}[r], \mathbf{x} \rangle)| \leq C_\sigma$ and $|\sigma'(\langle \mathbf{G}[r], \mathbf{x} \rangle)| \leq C'_\sigma$. Moreover $|\mathbf{x}_\ell| \leq R_{\mathbf{x}}(F)$ and

$$|\langle \mathbf{a}_r, \mathbf{x} \rangle| \leq \|\mathbf{a}_r\| \|\mathbf{x}\| \leq R_\theta(F) R_{\mathbf{x}}(F).$$

Hence

$$\left| \frac{\partial \text{enc}_\theta(\mathbf{x})}{\partial \mathbf{A}_{r\ell}} \right| \leq C_\sigma R_{\mathbf{x}}(F), \quad \left| \frac{\partial \text{enc}_\theta(\mathbf{x})}{\partial \mathbf{G}_{r\ell}} \right| \leq C'_\sigma R_\theta(F) R_{\mathbf{x}}(F)^2.$$

The gradient $\nabla_\theta \text{enc}_\theta(\mathbf{x}) \in \mathbb{R}^P$ collects all these partial derivatives, so its Euclidean norm satisfies

$$\|\nabla_\theta \text{enc}_\theta(\mathbf{x})\|_2^2 \leq P \cdot (\max\{C_\sigma R_{\mathbf{x}}(F), C'_\sigma R_\theta(F) R_{\mathbf{x}}(F)^2\})^2 \leq C \text{poly}(F)^2$$

for some constant $C > 0$, using $P \leq \text{poly}(F)$ and $R_{\mathbf{x}}(F), R_\theta(F) \leq \text{poly}(F)$. Thus there exists $L(F) \leq C^{1/2} \text{poly}(F)$ such that

$$\|\nabla_\theta \text{enc}_\theta(\mathbf{x})\|_2 \leq L(F) \quad \text{for all } \|\theta\|_2 \leq R_\theta(F).$$

For any θ, θ' with $\|\theta\|_2, \|\theta'\|_2 \leq R_\theta(F)$, the mean value inequality in $\mathbb{R}^P$ yields

$$|\text{enc}_\theta(\mathbf{x}) - \text{enc}_{\theta'}(\mathbf{x})| \leq \sup_{\tilde{\theta} \in [\theta, \theta']} \|\nabla_{\tilde{\theta}} \text{enc}_{\tilde{\theta}}(\mathbf{x})\|_2 \cdot \|\theta - \theta'\|_2 \leq L(F) \|\theta - \theta'\|_2.$$

Since $L(F) \leq \text{poly}(F)$ by construction, this proves the claim. TODO: Cite ??? to show that assumption 4 holds. $\square$

²⁶Subgaussianity is preserved under linear maps: for any $\mathbf{u} \in \mathbb{R}^{dh}$, $\langle \mathbf{u}, \tilde{\mathbf{r}}_i \rangle = \langle \Sigma_{\text{row}}^{-1/2\top} \mathbf{u}, \mathbf{r}_i \rangle$ is subgaussian with $\|\langle \mathbf{u}, \tilde{\mathbf{r}}_i \rangle\|_{\psi_2} \leq K \|\Sigma_{\text{row}}^{-1/2}\|_{\text{op}} \|\mathbf{u}\|_2$, where K bounds $\|\mathbf{r}_i\|_{\psi_2}$. By assumption 2, $\|\Sigma_{\text{row}}^{-1/2}\|_{\text{op}} = 1/\sqrt{\lambda_{\min}(\Sigma_{\text{row}})} \leq 1/\sqrt{\lambda_0}$, so $\|\tilde{\mathbf{r}}_i\|_{\psi_2} \lesssim K/\sqrt{\lambda_0}$, a constant independent of F .

B.10.17 Proof of Theorem B.8.7

Proof. Step 1: Allowed code noise. From Theorem B.8.2, there is a constant $c_0 > 0$ such that, if the code for fact i is perturbed by at most $c_0\rho$ in an appropriate sense (as in the theorem's proof), then the score margin remains positive:

$$s_{ii} - s_{ij} \geq \Omega(\rho) \|\mathbf{v}_i - \mathbf{v}_j\| \|\mathbf{u}_i\|.$$

Thus the encoder codes are robust to perturbations of size $\Theta(\rho)$. Using (i), we have

$$\rho \geq \frac{1}{\text{poly}(F)},$$

so the allowed code noise is at least $1/\text{poly}(F)$.

Step 2: From parameter perturbation to code perturbation. Let $\theta_\star$ be the ideal encoder parameters and $\tilde{\theta}$ any other parameter vector. For each key $\mathbf{k}_i$, define the code perturbation

$$\Delta_i := \text{enc}_{\tilde{\theta}}(\mathbf{k}_i) - \text{enc}_{\theta_\star}(\mathbf{k}_i).$$

By the Lipschitz assumption (ii),

$$\|\Delta_i\| = \|\text{enc}_{\tilde{\theta}}(\mathbf{k}_i) - \text{enc}_{\theta_\star}(\mathbf{k}_i)\| \leq L(F) \|\tilde{\theta} - \theta_\star\| \quad \forall i.$$

To keep the codes within the robustness radius from Step 1, it suffices to impose

$$\|\Delta_i\| \leq c_0\rho \quad \forall i.$$

A sufficient condition is therefore

$$\|\tilde{\theta} - \theta_\star\| \leq \delta(F) := \frac{c_0\rho}{L(F)}.$$

Using (i) and (ii), we obtain

$$\delta(F) \geq \frac{c_0}{\text{poly}(F) \text{poly}(F)} = \frac{1}{\text{poly}(F)}.$$

So there is a ball of radius at least $1/\text{poly}(F)$ around $\theta_\star$ in parameter space such that any $\tilde{\theta}$ in this ball produces codes that Theorem B.8.2 can tolerate.

Step 3: Quantization and choice of grid size. Now quantize each coordinate of $\theta_\star$ to a grid of step size $\Delta > 0$, obtaining $\tilde{\theta}$. Each coordinate changes by at most $\Delta/2$, so

$$\|\tilde{\theta} - \theta_\star\|_2 \leq \sqrt{P} \frac{\Delta}{2}.$$

To guarantee $\|\tilde{\theta} - \theta_\star\| \leq \delta(F)$, it is enough to choose Δ so that

$$\sqrt{P} \frac{\Delta}{2} \leq \delta(F) \iff \Delta \leq \frac{2\delta(F)}{\sqrt{P}}.$$

Using $\delta(F) \geq 1/\text{poly}(F)$ and $P \leq \text{poly}(F)$ from (iii), we get

$$\frac{2\delta(F)}{\sqrt{P}} \geq \frac{1}{\text{poly}(F)}.$$

Thus the admissible step size Δ can be as large as $1/\text{poly}(F)$. In particular, we may pick

$$\Delta := F^{-c}$$

for some constant $c > 0$ large enough so that $\Delta \leq 2\delta(F)/\sqrt{P}$. This ensures $\|\tilde{\theta} - \theta_\star\| \leq \delta(F)$ and, by Step 2, that the induced code perturbations are within the noise budget of Theorem B.8.2. Hence decoding remains correct.

Step 4: Bit complexity. By Item (iv), each parameter lies in an interval of length at most $\text{range} \leq 2/\text{poly}(F)$. With grid spacing $\Delta = F^{-c} = 1/\text{poly}(F)$, the number of distinct values per parameter is at most

$$\frac{\text{range}}{\Delta} \leq \frac{\text{poly}(F)}{1/\text{poly}(F)} = \text{poly}(F).$$

Therefore the number of bits per parameter is

$$\log_2 \left(\frac{\text{range}}{\Delta} \right) = O(\log \text{poly}(F)) = O(\log F).$$

This proves that encoder parameters require only $O(\log F)$ bits of precision.

□

Algorithm 7 Gated Encoder Gadget Construction (GATEDENCODERGADGET)

Require: $\mathbf{o} \in \mathbb{R}^{|\mathbf{K}|}$, generic $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$

Require: Hidden size h with $dh \geq |\mathbf{K}|$, analytic σ , bias flag BIAS

- 1: Sample generic $\mathbf{G} \in \mathbb{R}^{h \times d}$ (e.g., i.i.d. Gaussian)
 - 2: **if** BIAS:
 - 3: Sample arbitrary $\mathbf{b}_G \in \mathbb{R}^h$ (e.g., all zeros)
 - 4: **else:**
 - 5: $\mathbf{b}_G := \mathbf{0}_h \in \mathbb{R}^h$ (e.g., all zeros)
 - 6: $\Sigma := \sigma(\mathbf{G}\mathbf{K}^\top + \mathbf{b}_G) \in \mathbb{R}^{h \times |\mathbf{K}|}$
 - 7: **if** BIAS:
 - 8: $\tilde{d} := d + 1$
 - 9: $\tilde{\mathbf{K}} := [\mathbf{K}, \mathbf{1}_{|\mathbf{K}|}] \in \mathbb{R}^{|\mathbf{K}| \times \tilde{d}}$
 - 10: **else:**
 - 11: $\tilde{d} := d$
 - 12: $\tilde{\mathbf{K}} := \mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times \tilde{d}}$
 - 13: $\mathbf{M} := [\text{diag}(\Sigma_1)\tilde{\mathbf{K}}, \dots, \text{diag}(\Sigma_h)\tilde{\mathbf{K}}] \in \mathbb{R}^{|\mathbf{K}| \times (dh)}$
 - 14: **if** BIAS:
 - 15: $\tilde{D} := dh + 1$
 - 16: $\tilde{\mathbf{M}} := [\mathbf{M}, \mathbf{1}_{|\mathbf{K}|}] \in \mathbb{R}^{|\mathbf{K}| \times \tilde{D}}$
 - 17: **else:**
 - 18: $\tilde{D} := dh$
 - 19: $\tilde{\mathbf{M}} := \mathbf{M} \in \mathbb{R}^{|\mathbf{K}| \times \tilde{D}}$
 - 20: Solve for $\mathbf{v} \in \mathbb{R}^{dh}$ in $\tilde{\mathbf{M}}\mathbf{v} = \mathbf{o}$
 - 21: $\mathbf{A} := \begin{bmatrix} \mathbf{v}[1 : \tilde{d} - 1] \\ \mathbf{v}[\tilde{d} + 1 : 2\tilde{d} - 1] \\ \vdots \\ \mathbf{v}[(h-1)\tilde{d} + 1 : h\tilde{d} - 1] \end{bmatrix} \in \mathbb{R}^{h \times d}$
 - 22: **if** BIAS:
 - 23: $\mathbf{b}_A := \begin{bmatrix} \mathbf{v}[\tilde{d}] \\ \mathbf{v}[2\tilde{d}] \\ \vdots \\ \mathbf{v}[h\tilde{d}] \end{bmatrix} \in \mathbb{R}^h$
 - 24: $b_E := \mathbf{v}[D] \in \mathbb{R}$
 - 25: **else:**
 - 26: $\mathbf{b}_A := \mathbf{0}_h \in \mathbb{R}^h$
 - 27: $b_E := 0 \in \mathbb{R}$
 - 28: $\text{enc}(\mathbf{x}) := \mathbf{1}_h (\sigma(\mathbf{G}\mathbf{x} + \mathbf{b}_G) \odot (\mathbf{A}\mathbf{x} + \mathbf{b}_A)) + b_E$
 - 29: **return enc**
-

Algorithm 8 Encoder Gadget Construction (ENCODERGADGET)

Require: $\mathbf{o} \in \mathbb{R}^{|\mathbf{K}|}$, generic $\mathbf{K} \in \mathbb{R}^{|\mathbf{K}| \times d}$

Require: Hidden size h with $dh \geq |\mathbf{K}|$, analytic σ , bias flag BIAS, tolerance δ

1: $\mathbf{enc}(\mathbf{x}) := \mathbf{1}_{1 \times h/2} \left(\frac{d\sigma}{dx}(\mathbf{G}\mathbf{x} + \mathbf{b}_G) \odot (\mathbf{A}\mathbf{x} + \mathbf{b}_A) \right) + b_E \leftarrow \text{GATEDENCODERGADGET}(\mathbf{K}, \mathbf{o}, h/2, \frac{d\sigma}{dx}, \text{BIAS})$

2: **for** $i = 1$ **to** $|\mathbf{K}|$ **do**

3: $S_i := \left\{ \epsilon \left| \left| [\epsilon^{-1}/2, -\epsilon^{-1}/2] \sigma \left(\begin{bmatrix} \mathbf{G} + \text{diag}(\epsilon)\mathbf{A} \\ \mathbf{G} - \text{diag}(\epsilon)\mathbf{A} \end{bmatrix} \mathbf{k}_i + \begin{bmatrix} \mathbf{b}_G + \epsilon \odot \mathbf{b}_A \\ \mathbf{b}_G - \epsilon \odot \mathbf{b}_A \end{bmatrix} \right) - \mathbf{enc}(\mathbf{k}_i) \right| \leq \delta \right\}$

4: **end for**

5: Pick any $\epsilon \in \bigcap_{i=1}^{|\mathbf{K}|} S_i$

6: $\mathbf{A} := \begin{bmatrix} \mathbf{G} + \text{diag}(\epsilon)\mathbf{A} \\ \mathbf{G} - \text{diag}(\epsilon)\mathbf{A} \end{bmatrix} \in \mathbb{R}^{h \times d}$

7: $\mathbf{b}_A := \begin{bmatrix} \mathbf{b}_G + \epsilon \odot \mathbf{b}_A \\ \mathbf{b}_G - \epsilon \odot \mathbf{b}_A \end{bmatrix} \in \mathbb{R}^h$

8: $\mathbf{E} := [\epsilon^{-1}/2, -\epsilon^{-1}/2] \in \mathbb{R}^{1 \times h}$

9: $\mathbf{enc}(\mathbf{x}) := \mathbf{E}\sigma(\mathbf{A}\mathbf{x} + \mathbf{b}_A) + b_E$

10: **return** $\mathbf{enc}$
