Cryptanalytic Extraction of Multi-Head Softmax Attention Models*

Sunan Wang[1,2,3], Hao Lei[1,2,3], Longxiang Wei[1,2,3], Qun Liu[1,2,3], Kai Hu[1,3,4,5], and Meiqin Wang[1,2,3,4]

[1] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China
{sunan.wang, leihao, longxiangwei}@mail.sdu.edu.cn
{qunliu, kai.hu, mqwang}@sdu.edu.cn
[2] State Key Laboratory of Cryptography and Digital Economy Security, Shandong University, Qingdao, 266237, China
[3] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
[4] Quan Cheng Shandong Laboratory, Jinan, China
[5] Suzhou Research Institute, Shandong University, Suzhou, 215123, China

Abstract. Since the seminal work of Carlini et al. at CRYPTO 2020, cryptanalytic model extraction has shown neural-networks parameters can be recovered from black-box queries. Existing attacks are largely built around piecewise-linear phenomena. Softmax attention, as the key component of the transformer architecture, presents a different extraction landscape: its nonlinearity is smooth and sequence-dependent, which renders the existing piecewise-linear-based method inapplicable. Recent work has investigated the learnability of a single-head attention model, while in the multi-head case, the parameters of the multi-head attention layer cannot be uniquely identified from value queries alone.

In this paper, we propose the first attack against multi-head attention models. We formalize the extractable representative of multi-head attention and give a polynomial-time algorithm for extracting the parameters of the canonical representative model. We also test our algorithm end to end under finite precision, and successfully extract the parameters of a softmax attention model with token dimension 8 and 6 heads to accuracy 2^{-51}. Moreover, we overcome the limitation that existing parameter extraction algorithms for one-layer single-head Transformers fail when the ReLU feedforward networks (FFNs) include bias terms. The effectiveness of our approach is demonstrated through model extraction attacks in finite-precision experiments. These results show that softmax normalization itself exposes exploitable algebraic structure, extending cryptanalytic extraction beyond ReLU-centric techniques.

* This version of the work was submitted to ASIACRYPT 2026 on May 22, 2026; related work by Kim et al. (arXiv:2608.03294), first posted on August 4, 2026, shares with our work the use of repeated-token queries and rational interpolation for multi-head softmax-attention recovery, while differing in its reconstruction procedure, extensions, and experimental evaluation.

Keywords: Cryptanalysis · Multi-head attention · Softmax · Model Extraction.

1 Introduction

Neural-network parameters are valuable intellectual property. Modern models are expensive to train, tune, and deploy, and a service provider often protects this investment by exposing only a black-box prediction interface. This has motivated a large body of model-extraction and model-stealing attacks, ranging from functionality stealing and distillation-style imitation to high-fidelity and parameter-recovery attacks [16,13,11]. Following the cryptanalytic viewpoint initiated by Carlini, Jagielski, and Mironov at CRYPTO 2020 [4], we view a trained neural network as a keyed mathematical object: the trained parameters play the role of a secret key, and a model-extraction adversary adaptively submits chosen inputs and observes outputs in order to recover an equivalent key.

Most existing cryptanalytic extraction attacks exploit the piecewise-linear structure of ReLU-based neural networks. The original attack of Carlini et al. [4] showed that ReLU critical points leak algebraic information about hidden weights through local differential behavior. Subsequent work substantially refined this paradigm. Canales-Martínez et al. [2] reduced the sign-recovery bottleneck from exponential time to polynomial time. Later work identified signature recovery, numerical stability, and end-to-end layer reconstruction as central obstacles for deeper networks, and developed techniques to navigate these difficulties [10,12]. Other works extended cryptanalytic extraction to more restrictive oracle models. At ASIACRYPT 2024, Chen et al. [7] initiated functionally equivalent extraction in the hard-label setting, while follow-up work introduced dual-point and dual-space techniques to obtain polynomial-time and polynomial-query attacks under hard-label access [3].

A second line of work studies how the choice of activation function changes the extraction problem. PReLU activations, for example, no longer block the negative half-line and therefore invalidate several ReLU-specific sign-recovery arguments; this led to dedicated attacks for PReLU networks [8]. More recent work further generalized the study to activations such as Leaky ReLU, HardTanh, ELU, and Step, aiming for a more unified view across different feedback settings [14]. Beyond fully connected networks, architectural structure has also become a major focus. Recent attacks exploit convolutional structure for average-pooling CNNs [15] and extend the setting to the more challenging max-pooling case [9]. For RNN models, parameter sharing across unfolded time steps has been used to push extraction to much larger effective depths [18].

At EUROCRYPT 2024, Canales-Martínez et al. [2] explicitly listed Transformer extraction as an open direction. Transformers pose a qualitatively different extraction problem. Their central component is softmax attention [17], whose output is a smooth and globally normalized function of all tokens in the input sequence. Unlike ReLU networks, softmax attention does not expose piecewise-linear activation regions, critical hyperplanes, or activation signatures. Moreover,

in multi-head attention the observable output mixes contributions from several heads, and different parameterizations can induce exactly the same input-output function. These features make existing ReLU-based extraction tools insufficient for attention layers.

More recently, Bhattamishra et al. [1] show that a single-head softmax attention regressor can be exactly recovered from value queries using $O(d^2)$ queries. Their work also proves that literal multi-head parameter recovery is impossible in general, since distinct multi-head parameterizations may define the same function. As a consequence, they left the functionally equivalent extraction of multi-head attention as an open question. Separately, Chen and Li [6] study passive learning of a multi-head attention layer from random examples. But their method requires exponential query and time complexity in the number of heads H. Recent attacks on production language models have also shown that nontrivial Transformer components can leak through API interfaces [5], while that attack focuses more on the embedding projection layer instead of the softmax attention.

In this work, we take this non-identifiability as the starting point. As is standard in cryptanalytic extraction of ReLU networks, where parameters are recovered to a functionally equivalent form, we target the identifiable representative of the black-box function. For multi-head attention, this leads to a canonical representative obtained by merging heads with identical attention matrices and removing zero-contribution heads. To the best of our knowledge, we give the first functionally equivalent extraction attack that recovers a multi-head softmax attention model from value queries, and we evaluate the attack under finite-precision arithmetic through end-to-end experiments.

1.1 Our contributions

Prior work shows that literal multi-head attention parameters are not identifiable from value queries, and leaves open the extraction of one-layer Transformer with single-head attention whose FFN contains bias terms. We address both issues: for multi-head attention, we separate the unavoidable non-identifiability of the original head list from the identifiable object recovered by the attack by canonicalizing the model, i.e., merging heads with identical attention matrices and removing zero-contribution heads, thereby obtaining a canonical representative identifiable up to permutation; we then introduce a duplicate-token rational-function method to recover this canonical multi-head representative. For biased FFNs, we draw on the piecewise-affine structure used in prior DNN extraction attacks: by querying within local affine regions of the recovered FFN, we bypass the failure of the antisymmetric reduction and recover the attention matrix. Concretely, we make the following contributions.

Identifiable extraction target for multi-head attention. We formalize the extraction target for multi-head softmax attention by canonicalizing the head representation: heads with identical attention matrices are merged by summing their value vectors, and zero-contribution heads are removed. This representative

has the same input-output function as the original model and is unique up to a permutation of the remaining heads. It resolves the non-identifiability shown in [1] by targeting the right quantity.

Rational-pole extraction of multi-head softmax attention. We give a polynomial-time algorithm that recovers the multi-head attention models up to functional equivalence. The core tool is the duplicate-token probe: querying a sequence where one token appears s times turns the multi-head oracle response into a rational function of s. In the exact-arithmetic model, for token dimension d and the heads number H, the attack uses $d+2H(1+d+d^2)$ value queries and the maximum sequence length is $2H+1$.

Bias-aware extraction for single-head Transformer blocks. We study single-head softmax attention followed by a biased scalar ReLU FFN. The anti-symmetric reduction used in prior work [1] fails once the FFN has a bias term. We bypass this obstruction in two steps. First, length-one queries reduce the oracle to a pure FFN evaluation, which we extract using FFN extraction primitive. Second, the recovered FFN identifies local affine regions where the oracle output is linear in the attention weight, allowing attention matrix recovery via two-token probes.

Table 1. Comparison of state-of-the-art results.

Model / Task		**Extraction / Learning**						**Approach**
Architecture	Target	Oracle / data	Multi-head	Biased FFN	Query / sample	Time	Finite prec.	Reference
ReLU-DNNs	parameter extraction	chosen queries	–	–	*poly*	*poly*	✓	EC'24 [2]
PReLU-DNNs	parameter extraction	chosen queries	–	–	*poly*	*poly*	✓	AC'25 [8]
LeakyReLU-DNNs	parameter extraction	chosen queries	–	–	*poly*	*poly*	✓	FSE'26 [14]
Max-pooling CNN	parameter extraction	chosen queries	–	–	*poly*	*poly*	✓	ePrint'26 [9]
RNN	parameter extraction	chosen queries	–	–	*poly*	*poly*	✓	ePrint'26 [18]
Transformer	parameter extraction	chosen queries	✗	restricted†	*poly*	*poly*	theory‡	ICML'26 [1]
Transformer	small-error learning	known queries	✓	✗	*exp*	*exp*	theory‡	STOC'25 [6]
Transformer	parameter extraction	chosen queries	✓	✗	*poly*	*poly*	✓	*This work*
Transformer	parameter extraction	chosen queries	✗	✓	*poly*	*poly*	✓	*This work*

† Applies only to the bias-free FFN setting.
‡ No finite-precision end-to-end extraction result is reported.

In Table 1, we compare our results to prior state-of-the-art works. Prior cryptanalytic extraction attacks mainly apply to piecewise-linear neural networks, while existing attention results either recover only single-head attention or study learning in exponential costs. Our work gives a chosen-query cryptanalytic extraction attack for canonical multi-head softmax attention and a bias-aware extraction attack for single-head Transformer, both validated in finite precision. Our code is accessible in `https://anonymous.4open.science/r/End-to-End-Softmax-Attention-Extraction-632C`.

2 Preliminaries

2.1 Basic Definitions and Notions

Definition 1 (Input Sequence) *An input sequence of length N with token dimension d is represented by a matrix*

$$X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_N^\top \end{pmatrix} \in \mathbb{R}^{N\times d}, \qquad x_i \in \mathbb{R}^d. \tag{1}$$

Throughout this paper, the last token x_N is used as the query token.

Definition 2 (Single-head Attention) *Let $Q, K, V \in \mathbb{R}^{r\times d}$ be the query, key, and value projection matrices, and let $w_o \in \mathbb{R}^r$ be the output projection vector. For an input sequence $X \in \mathbb{R}^{N\times d}$, a single-head attention regressor outputs*

$$\mathrm{SHA}(X) = \mathrm{softmax}\left(x_1^\top K^\top Q x_N, \ldots, x_N^\top K^\top Q x_N\right)^\top XV^\top w_o. \tag{2}$$

Since the input-output map only depends on the products $K^\top Q$ and $V^\top w_o$, we use the merged parameters

$$W := K^\top Q \in \mathbb{R}^{d\times d}, \qquad v := V^\top w_o \in \mathbb{R}^d. \tag{3}$$

For an input sequence $X \in \mathbb{R}^{N\times d}$, define the attention score of the i-th token by

$$s_i(X, W) = x_i^\top W x_N, \qquad i = 1, \ldots, N, \tag{4}$$

and define the corresponding attention weights by

$$\alpha(X, W) = \mathrm{softmax}\big(s_1(X, W), \ldots, s_N(X, W)\big) \in \mathbb{R}^N, \tag{5}$$

where $\mathrm{softmax}(z)_i = \exp(z_i)\Big/\sum_j \exp(z_j)$. The single-head attention regressor with merged parameters (W, v) is then denoted by

$$f_{W,v}(X) = \alpha(X, W)^\top Xv = \sum_{i=1}^{N} \alpha_i(X, W)\, x_i^\top v. \tag{6}$$

This merged-parameter representation is used throughout the paper.

Definition 3 (Multi-head Attention) *For $H \geq 1$, an H-head attention regressor is given by merged parameters*

$$(W_h, v_h) \in \mathbb{R}^{d\times d} \times \mathbb{R}^d, \qquad h = 1, \ldots, H. \tag{7}$$

For an input sequence $X \in \mathbb{R}^{N\times d}$, its output is

$$f_{W,v}^{H}(X) = \sum_{h=1}^{H} \alpha(X, W_h)^\top Xv_h. \tag{8}$$

Thus, a multi-head attention model is a sum of H single-head attention regressors, each with its own merged parameters (W_h, v_h).

Definition 4 (Model Parameters) *For any parameterized model considered in this paper, we use θ to denote the parameters that define its input-output map, and denote the corresponding function by f_θ.*
For a single-head attention regressor, the merged parameters are

$$\theta = (W, v).$$

For an H-head attention regressor, the merged parameters are

$$\theta = \{(W_h, v_h)\}_{h=1}^{H}.$$

We use $\theta^\star$ for the unknown target parameters and $\widehat{\theta}$ for the extracted parameters. Unless explicitly stated otherwise, θ refers to merged attention parameters rather than to a particular factorization into query, key, value, and output projection matrices.

2.2 Adversarial Goals and Assumptions

There are two parties in model extraction attack: the oracle $\mathcal{O}$ who returns $f_{\theta^\star}(X)$, and the adversary who adaptively generates input sequences X to the oracle. The goal of the adversary is to extract the parameters defined in Def. 4 and recover the functionally equivalent model.

Assumptions. We make the following assumptions about the oracle and the capabilities of the attacker.

- **Known architecture.** The adversary knows the token dimension d and an upper bound H on the number of heads.
- **Full-domain inputs.** The adversary may query arbitrary real-valued token sequences $X \in \mathbb{R}^{N\times d}$ and may choose the sequence length N adaptively.
- **Raw scalar outputs.** The oracle returns the scalar value $f_{\theta^\star}(X)$ without label-only feedback, rounding, truncation, or additive noise.
- **Exact arithmetic.** The main theorems are analyzed in an exact-arithmetic model. Finite-precision behavior is evaluated experimentally.
- **Generic non-degeneracy.** Some results require generic conditions excluding measure-zero degeneracies, such as indistinguishable heads or zero observable residues. These conditions are stated explicitly when used.

Definition 5 ((ε, δ)-Functional Equivalence) *Let $f, \widehat{f} : \mathcal{X} \to \mathbb{R}$ be two raw-output models, and let S be an input set or distribution over $\mathcal{X}$. Following Carlini et al. [4], we say that $\widehat{f}$ is* (ε, δ)-functionally equivalent *to f on S if*

$$\Pr_{x\sim S}\left[|f(x) - \widehat{f}(x)| \le \varepsilon\right] \ge 1 - \delta. \tag{9}$$

When the equality $\widehat{f}(x) = f(x)$ holds for every x in the full input domain $\mathcal{X}$, we call this exact functional equivalence and write

$$\widehat{f} \equiv f. \tag{10}$$

The theoretical results in this paper target exact functional equivalence, while finite-precision experiments are evaluated using the (ε, δ) metric.

3 From Single-Head Identifiability to Attention Extraction Barriers

Prior work shows that the merged parameters of a single-head attention regressor are identifiable from value queries. More precisely, for

$$f_{W^\star,v^\star}(X) = \alpha(X, W^\star)^\top X v^\star,$$

the pair $(W^\star, v^\star)$ can be recovered with $O(d^2)$ value queries whenever $v^\star \neq 0$ [1]. The key observation is that short chosen sequences isolate the softmax nonlinearity: length-one queries make the softmax output identically 1, and length-two queries reduce softmax to an invertible sigmoid.

In this section, we recall this recovery in our notation because it is the baseline from which our work departs. It also clarifies why the two settings studied in this paper require new techniques: multiple attention heads destroy the direct sigmoid inversion used in the single-head case while biased feed-forward networks prevent the usual linearization trick.

3.1 Prior Single-Head Recovery

We first recall the value-query recovery algorithm for a single-head attention regressor, adapted to our notation.

Lemma 1 (Single-head recovery from value queries [1]). *Assume $v^\star \neq 0$. Given value-query access to $\mathcal{O}(X) = f_{W^\star,v^\star}(X)$, the adversary can recover the parameter $v^\star$ and $W^\star$ with $O(d^2)$ queries.*

Proof. For a one-token query $X = [x^\top]$, the softmax weight is 1, hence $\mathcal{O}([x^\top]) = x^\top v^\star$. Querying $x = e_i$ for $i = 1, \ldots, d$ recovers $v^\star$.

It remains to recover $W^\star$. Fix a column index j and write $w_j := W^\star e_j$. For any probe vector u with $u^\top v^\star \neq 0$, query

$$X_{u,j} = \begin{pmatrix} (u + e_j)^\top \\ e_j^\top \end{pmatrix}.$$

Since the last token is e_j, the score difference between the first and second tokens is

$$(u + e_j)^\top W^\star e_j - e_j^\top W^\star e_j = u^\top w_j.$$

Thus the attention weight on the first token is $\sigma(u^\top w_j)$, where $\sigma(t) = 1/(1+\mathrm{e}^{-t})$. The oracle output satisfies

$$\mathcal{O}(X_{u,j}) = (v^\star)_j + \sigma(u^\top w_j)\, u^\top v^\star.$$

Since $v^\star$ is known, we can compute

$$\alpha_{u,j} = \frac{\mathcal{O}(X_{u,j}) - (v^\star)_j}{u^\top v^\star}, \qquad u^\top w_j = \sigma^{-1}(\alpha_{u,j}) = \log \frac{\alpha_{u,j}}{1 - \alpha_{u,j}}.$$

Therefore each two-token query gives one linear measurement of the column w_j. Taking d linearly independent probes $u_1, \ldots, u_d$ gives an invertible linear system for w_j. Repeating the same procedure for $j = 1, \ldots, d$ recovers all columns of $W^\star$. The total number of queries is $d + d^2$. □

Lemma 1 explains why pure single-head attention is identifiable in the value-query model. The argument relies on isolating a single scalar sigmoid term from a two-token query. The rest of this section explains why this strategy cannot be directly extended to the models considered in this paper.

3.2 Barrier I: Multiple Heads Cannot Be Inverted One by One

The single-head strategy fails for multi-head attention. Consider an H-head target as in Def. 3,

$$f_{\theta^\star}(X) = \sum_{h=1}^{H} \alpha(X, W_h^\star)^\top X v_h^\star.$$

Even if each individual head has the same form as in the single-head case, the oracle only returns the sum of all head contributions. Thus a query response no longer isolates one sigmoid term corresponding to one attention matrix. More fundamentally, Bhattamishra et al. [1] show that, unlike single-head attention, multi-head attention is not identifiable from value queries alone. Consequently, parameter recovery guarantees analogous to the single-head case are impossible without additional structural assumptions.

Identifiability obstruction. If two heads have the same attention matrix W, then for every input sequence X,

$$\alpha(X, W)^\top X v_1 + \alpha(X, W)^\top X v_2 = \alpha(X, W)^\top X (v_1 + v_2).$$

Thus, black-box access cannot distinguish two heads $(W, v_1), (W, v_2)$ from one head $(W, v_1 + v_2)$.

Consequently, literal recovery of the original head list is not the right target. The appropriate goal is to recover a canonical functionally equivalent representative. We formalize this target in Section 4.

Section 4 introduces our main technique for this setting. Instead of trying to invert the heads one by one, we use duplicate-token probes to convert the multi-head softmax response into a rational function. The hidden canonical heads then appear as poles of this function, allowing them to be separated.

3.3 Barrier II: Bias Breaks the Antisymmetric Reduction

Lemma 1 applies to a pure attention regressor. Prior work also composes the single-head attention block with a two-layer ReLU MLP with no bias [1]. After merging the query-key projections and the value projection with the first FFN layer, we write

$$W^\star := K^{\star\top} Q^\star \in \mathbb{R}^{d \times d}, \qquad A^\star := V^{\star\top} W_1^{\star\top} \in \mathbb{R}^{d \times m}.$$

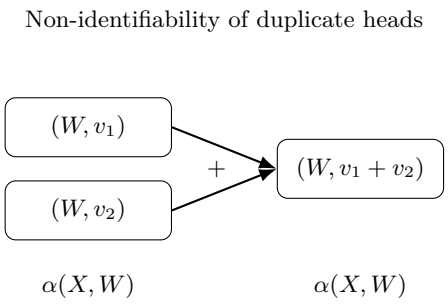

Fig. 1. Non-identifiability of duplicate heads in multi-head attention. Two heads with the same attention matrix are functionally equivalent to a single head with the summed value vector.

In this merged notation, the bias-free block is

$$\mathrm{TF}_0(X) = w_o^{\star\top}\,\mathrm{ReLU}\left(A^{\star\top}X^\top\alpha(X, W^\star)\right).$$

Let $z(X) = \alpha(X, W^\star)^\top XA^\star$. Using $\mathrm{ReLU}(z) - \mathrm{ReLU}(-z) = z$ coordinate-wise, the antisymmetric difference satisfies

$$\mathrm{TF}_0(X) - \mathrm{TF}_0(-X) = \alpha(X, W^\star)^\top X(A^\star w_o^\star).$$

Thus, in the bias-free case, by querying a pair of antipodal inputs $(X, -X)$ and taking the corresponding output difference, the one-layer Transformer oracle can be reduced to a single-head attention regression problem. Consequently, by applying Lemma 1, the attention matrix $W^\star$ can be recovered.

However, this reduction fails once the FFN has a bias. For the biased block

$$\mathrm{TF}_b(X) = w_o^{\star\top}\,\mathrm{ReLU}\left(\alpha(X, W^\star)^\top XA^\star + b^\star\right) + c^\star, \tag{11}$$

the same antisymmetric query gives

$$\mathrm{TF}_b(X) - \mathrm{TF}_b(-X) = w_o^{\star\top}\left[\mathrm{ReLU}(z(X) + b^\star) - \mathrm{ReLU}(-z(X) + b^\star)\right],$$

which is not linear in $z(X)$ in general. The bias shifts the ReLU breakpoints and prevents the FFN nonlinearity from being eliminated.

Section 5 shows how to bypass it: instead of canceling the FFN, we first extract a functionally equivalent biased FFN from length-one queries, and then use its local affine regions to recover the attention matrix $W^\star$.

4 Multi-Head Attention Extraction via Rational Poles

As discussed in Section 3, the single-head recovery relies on the fact that a length-two softmax reduces to one invertible sigmoid. This argument no longer applies directly when the oracle output is a sum of several heads: a single query

returns a mixture of several heads, and the individual heads are not separated by direct inversion. In this section, we first specify the identifiable extraction target and then show how duplicate-token queries turn this mixture into a rational function whose atoms separate the heads.

4.1 Canonical Extraction Target

Exact recovery of the original syntactic head list is not the right target for multi-head attention. If two heads have the same attention matrix W, then for every input sequence X,

$$\alpha(X,W)^\top X v_1 + \alpha(X,W)^\top X v_2 = \alpha(X,W)^\top X(v_1+v_2).$$

Thus black-box access cannot distinguish the two-head representation $(W,v_1),(W,v_2)$ from the one-head representation (W,v_1+v_2). This is analogous to the unavoidable scaling, permutation, and redundancy ambiguities that appear in cryptanalytic extraction of ReLU networks: the meaningful target is the identifiable representative of the input-output function.

Definition 6 (Canonical Multi-Head Representative) *Let $\theta=\{(W_h,v_h)\}_{h=1}^H$ be an H-head attention model. Let $\overline{W}_1,\dots,\overline{W}_M$ be the distinct matrices appearing among $W_1,\dots,W_H$. For each distinct matrix $\overline{W}_\ell$, define the index set*

$$I_\ell=\{h\in\{1,\dots,H\}: W_h=\overline{W}_\ell\},$$

and define the merged value vector

$$\overline{v}_\ell=\sum_{h\in I_\ell} v_h.$$

We delete every group with $\overline{v}_\ell=0$ and relabel the remaining pairs as

$$\widetilde{\theta}=\{(\widetilde{W}_g,\widetilde{v}_g)\}_{g=1}^G.$$

Equivalently, the matrices $\widetilde{W}_1,\dots,\widetilde{W}_G$ are pairwise distinct and $\widetilde{v}_g\neq 0$ for every $g=1,\dots,G$.

We call $\widetilde{\theta}$ the canonical multi-head representative *of θ, and call G the* effective number of heads*. The ordering of the pairs $(\widetilde{W}_g,\widetilde{v}_g)$ is irrelevant.*

The canonical representative induces the function

$$f_{\widetilde{\theta}}(X)=\sum_{g=1}^{G}\alpha(X,\widetilde{W}_g)^\top X\widetilde{v}_g.$$

Lemma 2 (Canonicalization preserves the function). *Let θ be an H-head attention model and let $\widetilde{\theta}$ be its canonical representative as in Def. 6. Then*

$$f_{\widetilde{\theta}}(X)=f_\theta(X)$$

for every input sequence X. Moreover, for the fixed parameter list θ, the canonical representative is uniquely determined up to a permutation of its heads.

Proof. For any group I of heads with the same score matrix W, all heads in the group have the same attention weights. Therefore,

$$\sum_{h\in I} \alpha(X,W)^\top X v_h = \alpha(X,W)^\top X \left(\sum_{h\in I} v_h\right)$$

for every input X. Replacing the group by one head with the summed value vector does not change the function, and a group whose summed value is zero contributes nothing and can be removed. This proves $f_{\widetilde{\theta}} \equiv f_\theta$.

The groups are uniquely determined by the equalities among the matrices W_h, and each merged value vector is uniquely determined by summation. The only remaining ambiguity is the order of the resulting heads. □

The previous lemma shows that canonicalization does not change the input-output function. The next theorem shows that canonicalization removes all non-identifiable redundancy.

Theorem 1 (Canonical functional identifiability). *Let*

$$\theta = \{(W_g, v_g)\}_{g=1}^{G}, \qquad \theta' = \{(W'_k, v'_k)\}_{k=1}^{G'}$$

be two canonical multi-head attention representations. If $f_\theta(X) = f_{\theta'}(X)$ for every $N \geq 1$ and every $X \in \mathbb{R}^{N\times d}$, then $G = G'$, and there exists a permutation π such that

$$W_g = W'_{\pi(g)}, \qquad v_g = v'_{\pi(g)} \qquad \text{for all } g = 1,\dots,G.$$

Proof. See Appendix A, since it uses the duplicate-token probes introduced later for the extraction algorithm. □

Thus the canonical representative is the unique black-box identifiable representative, up to permutation of heads.

For the target model $\theta^\star$, we write its canonical representative as

$$\widetilde{\theta}^\star = \{(\widetilde{W}^\star_g, \widetilde{v}^\star_g)\}_{g=1}^{G}.$$

Since $f_{\widetilde{\theta}^\star} \equiv f_{\theta^\star}$, we work directly with this canonical representative. To simplify notation, we relabel the canonical heads and the oracle as

$$\{(W^\star_g, v^\star_g)\}_{g=1}^{G}, \quad \mathcal{O}(X) = f_{\theta^\star}(X) = \sum_{g=1}^{G} \alpha(X, W^\star_g)^\top X v^\star_g.$$

Our goal is to recover the parameters $\widehat{\theta}$ such that

$$\forall N \geq 1,\ \forall X \in \mathbb{R}^{N\times d}, \quad f_{\widehat{\theta}}(X) \equiv f_\theta(X). \tag{12}$$

4.2 Attack Overview

In Figure 2, we illustrate the workflow of our multi-head attention extraction attack. The attack turns the original value-query oracle into a simpler algebraic object. The first step is to recover the aggregate value vector

$$v_{\Sigma}^{\star} = \sum_{g=1}^{G} v_g^{\star}$$

using length-one queries. This vector is not enough to separate the heads, but it enables us to construct a rational function from the oracle output of duplicate-token probes.

The core primitive is the duplicate-token probe. For chosen vectors $a, z \in \mathbb{R}^d$, a nonzero scalar t, and a chosen duplicate count s, we query a sequence of the form

$$X_s(a, z, t) = \begin{pmatrix} (z + ta)^\top \\ z^\top \\ \vdots \\ z^\top \end{pmatrix}.$$

The repeated copies of z make the softmax denominator depend explicitly on s. After subtracting the known $v_{\Sigma}^{\star}$ and dividing by t, the oracle response becomes a rational function

$$B_{a,z,t}(s) := \frac{\mathcal{O}(X_s(a, z, t)) - z^\top v_{\Sigma}^{\star}}{t} = \sum_{g=1}^{G} \frac{\mu_g(a) r_g}{s + r_g}.$$

Definition 7 (Rational atom) *A* rational atom *in the variable s is a simple rational term of the form*

$$\frac{c}{s + r}, \qquad r > 0,\ c \neq 0.$$

We call r the atom location parameter *and c the* atom coefficient*. Equivalently, the denominator has root $s = -r$, so the term has a simple pole at $s = -r$.*

Each canonical head contributes one rational atom. Recovering these atoms gives

$$a^\top W_g^{\star} z \qquad \text{and} \qquad a^\top v_g^{\star}$$

for all heads, up to an unknown permutation.

The remaining task is to combine such atom sets across several probes. We first recover the value vectors $v_g^{\star}$ using perturbations of the vector a. Once $v_g^{\star}$ is known, the scalar $a^\top v_g^{\star}$ becomes a fingerprint for the g-th head. We then use these fingerprints to label later atom sets and recover each matrix $W_g^{\star}$ by solving linear systems.

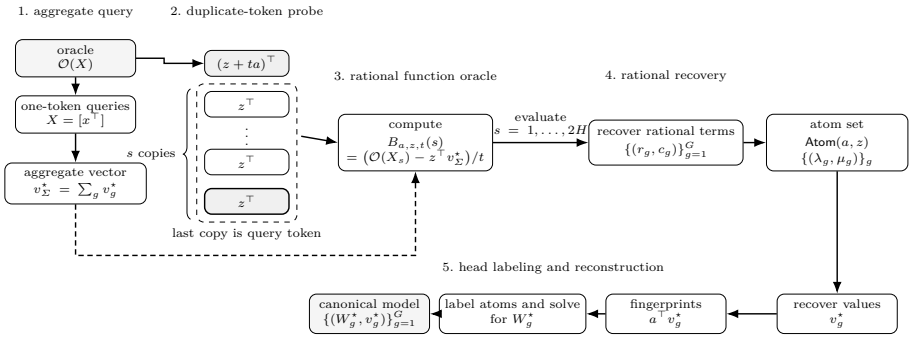

Fig. 2. Workflow of the multi-head attention extraction attack.

4.3 Duplicate-Token Probes

The purpose of the duplicate-token probe is to transform the attention softmax output, which is dominated by input token, into a rational function dominated by a single parameter. More precisely, we construct the duplicate-token probes shown in Figure 2. By repeating the same token z, the repeated tokens contribute identical score and value terms. Thus the sequence length itself becomes a controllable variable in the denominator.

We first recover the aggregate value vector. For a length-one input $X=[x^\top]$, every head assigns weight one to the only token. Hence

$$\mathcal{O}([x^\top]) = x^\top v_\Sigma^\star, \qquad v_\Sigma^\star := \sum_{g=1}^{G} v_g^\star. \tag{13}$$

Querying $x=e_i$ for $i=1,\dots,d$ recovers $v_\Sigma^\star$.

Now fix $a,z\in\mathbb{R}^d$ and $t\neq 0$. For an integer $s\geq 1$, query

$$X_s(a,z,t) = \begin{pmatrix} (z+ta)^\top \\ z^\top \\ \vdots \\ z^\top \end{pmatrix} \in \mathbb{R}^{(s+1)\times d}, \tag{14}$$

where z is repeated s times and the last copy is the query token.

Lemma 3 (Duplicate-token atom oracle). *Fix $a,z\in\mathbb{R}^d$ and $t\neq 0$. For each canonical head g, define*

$$\lambda_g(a,z) = a^\top W_g^\star z, \qquad \mu_g(a) = a^\top v_g^\star, \qquad r_g = e^{t\lambda_g(a,z)}. \tag{15}$$

Then the normalized response

$$B_{a,z,t}(s) := \frac{\mathcal{O}(X_s(a,z,t)) - z^\top v_\Sigma^\star}{t} \tag{16}$$

satisfies

$$B_{a,z,t}(s) = \sum_{g=1}^{G} \frac{\mu_g(a) r_g}{s + r_g}. \tag{17}$$

Thus each canonical head contributes one rational atom with pole $s = -r_g$.

Proof. Consider a fixed head g. The first token has score

$$(z + ta)^\top W_g^\star z = z^\top W_g^\star z + t\, a^\top W_g^\star z,$$

whereas each of the s repeated tokens has score $z^\top W_g^\star z$. After canceling the common factor $e^{z^\top W_g^\star z}$ in the softmax denominator, the attention weight of the first token is

$$\frac{e^{ta^\top W_g^\star z}}{s + e^{ta^\top W_g^\star z}} = \frac{r_g}{s + r_g}.$$

The first token has value

$$(z + ta)^\top v_g^\star = z^\top v_g^\star + t\, a^\top v_g^\star = z^\top v_g^\star + t\mu_g(a),$$

while every repeated token has value $z^\top v_g^\star$. Hence the output of head g is

$$z^\top v_g^\star + t\mu_g(a) \frac{r_g}{s + r_g}.$$

Summing over $g = 1, \ldots, G$ gives

$$\mathcal{O}(X_s(a, z, t)) = z^\top v_\Sigma^\star + t \sum_{g=1}^{G} \frac{\mu_g(a) r_g}{s + r_g}.$$

Subtracting $z^\top v_\Sigma^\star$ and dividing by t proves the claim. □

The importance of Lemma 3 is that it transforms multi-head attention into a one-variable rational function. This gives us black-box access to a rational function oracle $B_{a,z,t}(s)$, from which we can recover its partial fraction atoms, which in turn reveal

$$a^\top W_g^\star z \quad \text{and} \quad a^\top v_g^\star,$$

up to a permutation of the heads.

4.4 Rational Atom Recovery

Definition 8 (Atom Set) *Fix a probe pair $(a, z) \in \mathbb{R}^d \times \mathbb{R}^d$. With $\lambda_g(a, z)$ and $\mu_g(a)$ defined as in Eq. (15), the* atom set $\mathsf{Atom}(a, z)$ *is the unordered set $\{(\lambda_g(a, z), \mu_g(a))\}_{g=1}^{G}$.*

Algorithm 1 RecoverAtomSetKnownG$(B_{a,z,t}, t, G)$

Input: Rational-response oracle $B_{a,z,t}$, scalar $t \neq 0$, effective degree G
Output: Unordered atom set $\mathsf{Atom}(a,z) = \{(\lambda_g, \mu_g)\}_{g=1}^{G}$

for $\ell \leftarrow 1$ to $2G$ **do**
  $s_\ell \leftarrow \ell$
  $B_\ell \leftarrow B_{a,z,t}(s_\ell)$
Solve $\mathsf{LinSys}(G, \{(s_\ell, B_\ell)\}_{\ell=1}^{2G})$ for P and monic Q
Let $\rho_1, \dots, \rho_G$ be the roots of Q
for $g \leftarrow 1$ to G **do**
  $r_g \leftarrow -\rho_g$
  $c_g \leftarrow P(\rho_g)/Q'(\rho_g)$
  $\lambda_g \leftarrow \log(r_g)/t$
  $\mu_g \leftarrow c_g/r_g$
return $\{(\lambda_g, \mu_g)\}_{g=1}^{G}$

Algorithm 2 DetectEffectiveDegree$(B_{a,z,t}, H)$

Input: Rational-response oracle $B_{a,z,t}$ and head upper bound H
Output: Effective degree G

for $\ell \leftarrow 1$ to $2H$ **do**
  $s_\ell \leftarrow \ell$
  $B_\ell \leftarrow B_{a,z,t}(s_\ell)$
$\mathcal{S}_H \leftarrow \{(s_\ell, B_\ell)\}_{\ell=1}^{2H}$
for $m \leftarrow 1$ to H **do**
  Solve $\mathsf{LinSys}(m, \mathcal{S}_H)$ for P_m, Q_m
  if m is feasible and $Q_m(s_\ell) \neq 0$ for all $\ell = 1, \dots, 2H$ **then**
    return m

An atom set provides separated measurements for each individual attention head: bilinear measurements $a^\top W_g^\star z$ and projections $a^\top v_g^\star$. The recovery of the atom set from the rational function oracle $B_{a,z,t}(s)$ is purely algebraic.

For clarity, we first describe the recovery procedure when the effective degree G is known, formalized in Algorithm 1. Then we explain how G is detected from an upper bound H, formalized in Algorithm 2. In the actual attack, both steps use the same $2H$ samples of the rational function.

Proposition 1 (Known-degree atom recovery). *Assume*

$$B_{a,z,t}(s) = \sum_{g=1}^{G} \frac{\mu_g(a) r_g}{s + r_g}$$

has G distinct positive parameters r_g and nonzero coefficients $\mu_g(a) r_g$. Given $2G$ evaluations of $B_{a,z,t}$, Algorithm 1 recovers $\mathsf{Atom}(a,z)$ in exact arithmetic.

Proof. Write $B(s) = P(s)/Q(s)$ in minimal form, where Q is monic.

$$Q(s) = \prod_{g=1}^{G} (s + r_g) = s^G + \sum_{j=0}^{G-1} q_j s^j, \qquad P(s) = \sum_{j=0}^{G-1} p_j s^j.$$

For each sampled point s_ℓ, the equality $B(s_\ell)Q(s_\ell) = P(s_\ell)$ gives

$$B(s_\ell) \left(s_\ell^G + \sum_{j=0}^{G-1} q_j s_\ell^j \right) = \sum_{j=0}^{G-1} p_j s_\ell^j.$$

Let $B_\ell = B(s_\ell)$. Equivalently, this is the linear constraint

$$\sum_{j=0}^{G-1} B_\ell s_\ell^j q_j - \sum_{j=0}^{G-1} s_\ell^j p_j = -B_\ell s_\ell^G. \tag{18}$$

Thus the ℓ-th sample contributes one linear equation in the $2G$ unknown coefficients

$$q_0, \dots, q_{G-1}, p_0, \dots, p_{G-1}.$$

In row form, this equation is

$$\begin{pmatrix} B_\ell & B_\ell s_\ell & \cdots & B_\ell s_\ell^{G-1} & -1 & -s_\ell & \cdots & -s_\ell^{G-1} \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{G-1} \\ p_0 \\ p_1 \\ \vdots \\ p_{G-1} \end{pmatrix} = -B_\ell s_\ell^G.$$

When the exact number of atoms G is known, the true polynomials P and Q satisfy the interpolation equations, so a solution exists.

For uniqueness, suppose two rational functions P_1/Q_1 and P_2/Q_2, both with monic denominators of degree G and numerators of degree at most $G-1$, fit the same $2G$ samples. Then

$$P_1(s_\ell)Q_2(s_\ell) - P_2(s_\ell)Q_1(s_\ell) = 0$$

for all $\ell = 1, \dots, 2G$. The left-hand side is a polynomial of degree at most $2G-1$ with $2G$ distinct roots. Hence it is identically zero, and the two rational functions are equal. Since the denominator is monic and the representation is minimal, the coefficients of P and Q are uniquely determined.

Once Q is recovered, its roots are $\rho_g = -r_g$. The coefficient of the term with denominator $s + r_g$ is the residue

$$c_g = \frac{P(\rho_g)}{Q'(\rho_g)}.$$

Since $c_g = \mu_g(a) r_g$, Algorithm 1 outputs

$$\lambda_g(a, z) = \frac{\log r_g}{t}, \qquad \mu_g(a) = \frac{c_g}{r_g}.$$

This is exactly the unordered atom set $\mathsf{Atom}(a, z)$. □

Proposition 2 (Effective degree detection). *Assume that*

$$B_{a,z,t}(s) = \sum_{g=1}^{G} \frac{\mu_g(a) r_g}{s + r_g},$$

where the r_g's are pairwise distinct positive real numbers and $\mu_g(a) \neq 0$ for every g. Let $H \geq G$. Then Algorithm 2 returns the true effective degree G using $2H$ queries to $B_{a,z,t}$ in exact arithmetic.

Proof. The true degree G is feasible by the same argument as in Proposition 1: the true numerator and denominator satisfy all interpolation constraints.

It remains to show that no degree $m < G$ is feasible. Suppose, for contradiction, that a rational function P_m/Q_m with denominator degree $m < G$ fits all $2H$ sampled points. Let P/Q be the true minimal representation of $B_{a,z,t}$, whose denominator degree is G. Then

$$P_m(s_\ell)Q(s_\ell) - P(s_\ell)Q_m(s_\ell) = 0$$

for all sampled points s_ℓ. The left-hand side is a polynomial of degree at most

$$G + m - 1 < 2G \leq 2H,$$

but it has $2H$ distinct roots. Hence it is identically zero, which means that $B_{a,z,t}$ has a representation of denominator degree $m < G$. This contradicts minimality of G.

The minimality of G follows from the pairwise distinct denominator parameters and the nonzero coefficients: every term contributes a simple pole at $s = -r_g$, and no pole cancels. Therefore the first feasible degree found by Algorithm 2 is exactly G. □

Combining Propositions 1 and 2, one duplicate-token probe (a, z, t) returns the atom set

$$\mathsf{Atom}(a, z) = \left\{ \left(a^\top W_g^\star z,\ a^\top v_g^\star\right) \right\}_{g=1}^{G}$$

using $2H$ value queries to the original oracle $\mathcal{O}$, whenever the probe avoids pole collisions and zero coefficients.

4.5 From Atoms to Parameters

A single atom set gives only projected measurements and is unordered. The remaining task is to combine atom sets from multiple probes and consistently assign them to the same canonical heads. We first recover the value vectors $v_g^\star$, and then use them as fingerprints to label later atom sets and recover the score matrices $W_g^\star$.

Let $e_1, \ldots, e_d$ be the standard basis. Choose an anchor pair (a_0, z_0) and a small nonzero perturbation δ. For $i = 1, \ldots, d$, set $a_i = a_0 + \delta e_i$.

We assume the anchor is good: the poles at (a_0, z_0) are separated, the corresponding coefficients are nonzero, and the atoms from the perturbed probes (a_i, z_0) can be matched to the anchor labels. These are generic conditions on the adversary's probes.

Proposition 3 (Recovering the value vectors). *Assume an anchor (a_0, z_0) and perturbation δ are chosen. From the atom sets*

$$\mathsf{Atom}(a_0, z_0), \qquad \mathsf{Atom}(a_i, z_0), \quad i = 1, \dots, d,$$

the adversary recovers every canonical value vector $v_g^\star$, up to a common permutation of the heads.

Proof. The good-anchor condition allows the atoms at (a_i, z_0) to be matched to the anchor labels. Therefore, for each head g, the adversary knows

$$\mu_g(a_0) = a_0^\top v_g^\star, \qquad \mu_g(a_i) = a_i^\top v_g^\star$$

for every $i = 1, \dots, d$. Since $a_i = a_0 + \delta e_i$,

$$\mu_g(a_i) - \mu_g(a_0) = \delta e_i^\top v_g^\star.$$

Hence

$$(\widehat{v}_g)_i = \frac{\mu_g(a_i) - \mu_g(a_0)}{\delta} = e_i^\top v_g^\star. \tag{19}$$

This recovers all coordinates of $v_g^\star$. □

After recovering $v_g^\star$, we can label future atom sets without relying on small perturbations of the pole. For a chosen direction a, the atom associated with head g has second coordinate

$$\mu_g(a) = a^\top v_g^\star.$$

For generic choices of a, these fingerprints are distinct across heads. Thus the recovered value vectors provide a direct matching rule for later probes.

Proposition 4 (Recovering the score matrices). *Let $b_1, \dots, b_d \in \mathbb{R}^d$ be linearly independent directions and $z_1, \dots, z_d \in \mathbb{R}^d$ be linearly independent base tokens. Assume that for every i, the fingerprints*

$$b_i^\top v_1^\star, \dots, b_i^\top v_G^\star$$

are pairwise distinct. Then, using atom recovery on the probes

$$(b_i, z_j), \qquad i, j = 1, \dots, d,$$

the adversary recovers every $W_g^\star$.

Proof. For each probe (b_i, z_j), rational atom recovery gives the unordered atoms

$$\left\{ \left(b_i^\top W_g^\star z_j,\, b_i^\top v_g^\star \right) \right\}_{g=1}^{G}.$$

Since $v_g^\star$ is already known, the values $b_i^\top v_g^\star$ identify which atom belongs to which head. Hence, for every fixed g, i, and j, the adversary learns

$$b_i^\top W_g^\star z_j.$$

Fix a head g and a base token z_j. Let

$$B = \begin{pmatrix} b_1^\top \\ \vdots \\ b_d^\top \end{pmatrix}.$$

The d measurements for this z_j form the linear system

$$B(W_g^\star z_j) = \begin{pmatrix} b_1^\top W_g^\star z_j \\ \vdots \\ b_d^\top W_g^\star z_j \end{pmatrix}.$$

Since $b_1, \ldots, b_d$ are linearly independent, B is invertible, so the adversary recovers $W_g^\star z_j$. Repeating this for $j = 1, \ldots, d$ gives the columns of $W_g^\star Z$, where

$$Z = \begin{pmatrix} z_1 \cdots z_d \end{pmatrix}.$$

Since $z_1, \ldots, z_d$ are linearly independent, Z is invertible, and

$$W_g^\star = (W_g^\star Z) Z^{-1}.$$

This recovers $W_g^\star$. □

4.6 Discussion

We make two additional remarks about our extraction procedure. The first concerns the query complexity of the attention-extraction stage. The second concerns a mild non-generic case in which two distinct heads have the same value vector.

Query complexity. Recovering $v_\Sigma^\star$ uses d length-one queries. Each atom-set recovery uses $2H$ duplicate-token queries. The attack uses one anchor atom set, d perturbed atom sets for recovering the value vectors, and d^2 atom sets for recovering the score matrices. Hence the total number of value queries is

$$Q = d + 2H(1 + d + d^2).$$

The largest duplicate count used is $s = 2H$, so the maximum queried sequence length is $2H + 1$.

A generic canonical G-head softmax attention model in dimension d has

$$P = G(d^2 + d)$$

identifiable merged parameters: Gd^2 parameters in the attention matrices and Gd parameters in the value vectors. Since each value query returns one scalar constraint, a dimension-count argument implies that any locally identifiable extraction procedure requires $\Omega(G(d^2 + d))$ scalar queries in the generic case.

Our attack uses $d + 2H(1 + d + d^2)$ queries, where $H \geq G$ is a known upper bound on the effective number of heads. In the worst case $H = G$,

$$\frac{Q}{P} = 2 + \frac{2}{d(d+1)} + \frac{1}{G(d+1)}.$$

Thus the query complexity is within a small constant factor of the natural parameter-count lower bound, and the factor approaches 2 as d grows.

Coincident value vectors. Our reconstruction uses the recovered value measurements to label the rational atoms across different probes. In the generic case, a direction a separates the heads through the scalars $a^\top v_g^\star$. If two distinct canonical heads have the same value vector, this fingerprint no longer distinguishes them. This does not prevent the recovery of their attention matrices. We can instead keep an anchor (a_0, z_0) for which the corresponding poles are distinct, and match nearby probes by following the pole locations.

For each matched head g, the recovered pole gives

$$r_g(a, z, t) = \exp\big(t\, \lambda_g(a, z)\big), \qquad \lambda_g(a, z) = a^\top W_g^\star z.$$

Thus $\lambda_g(a, z)$ is available from the pole location. Since λ_g is bilinear in (a, z), the entries of $W_g^\star$ can be recovered by the finite-difference identity

$$(W_g^\star)_{ij} = \frac{\lambda_g(a_0 + \delta e_i, z_0 + \delta e_j) - \lambda_g(a_0 + \delta e_i, z_0) - \lambda_g(a_0, z_0 + \delta e_j) + \lambda_g(a_0, z_0)}{\delta^2},$$

for any nonzero δ. If we apply this method to extract the attention matrix W, this requires $1 + 2d + d^2$ atom-set recoveries and a total of $2H(1 + 2d + d^2)$ queries, and thus the asymptotic query complexity remains unchanged.

5 Bias-Aware Single-Head Transformer Block Extraction

In Section 3.3, we showed that the antisymmetric reduction for bias-free one-layer Transformers fails once the FFN contains a bias. In this section, we show how to bypass this obstruction. Instead of eliminating the FFN, we first extract a functionally equivalent biased FFN from length-one queries, and then use its local affine structure to recover the attention matrix.

5.1 Model and Extraction Goal

We work with the merged parametrization introduced in (11). For notational convenience, define

$$g^\star(x) = w_o^{\star\top} \operatorname{ReLU}\left(A^{\star\top} x + b^\star\right) + c^\star, \qquad x \in \mathbb{R}^d. \tag{20}$$

Then the biased one-layer single-head Transformer can be written as

$$\mathrm{TF}_{\theta^\star}(X) = g^\star\big(X^\top \alpha(X, W^\star)\big). \tag{21}$$

The adversary is given value-query access to an oracle

$$\mathcal{O}(X) = \mathrm{TF}_{\theta^\star}(X) = g^\star\big(X^\top \alpha(X, W^\star)\big).$$

The goal is to output an attention matrix $\widehat{W} \in \mathbb{R}^{d\times d}$ and a scalar one-hidden-layer ReLU model $\widehat{g} : \mathbb{R}^d \to \mathbb{R}$ such that

$$\widehat{g}\big(X^\top \alpha(X, \widehat{W})\big) = g^\star\big(X^\top \alpha(X, W^\star)\big) \tag{22}$$

for all $N \geq 1$ and all $X \in \mathbb{R}^{N\times d}$.

5.2 Stage I: Extracting the Biased FFN

The first stage uses only length-one queries. If $X = [x^\top]$, then the softmax weight is 1 and the attention output equals x. Therefore we have

$$\mathcal{O}([x^\top]) = g^\star(x) \tag{23}$$

reduced to a two-layer ReLU-based FFN with bias. Thus, by restricting the Transformer oracle to length-one inputs, the adversary obtains the black-box access of the biased FFN $g^\star$.

We use the following FFN extraction primitive.

FFN extraction primitive. Given value-query access to a scalar one-hidden-layer biased ReLU FFN

$$g^\star(x) = w_o^{\star\top}\,\mathrm{ReLU}\left(A^{\star\top}x + b^\star\right) + c^\star,$$

the primitive returns a functionally equivalent one-hidden-layer ReLU model $\widehat{g} \equiv g^\star$ with m recovered ReLU neurons. For the j-th recovered neuron, its kink hyperplane is

$$\widehat{H}_j = \{x : \widehat{r}_j^\top x + \widehat{s}_j = 0\}, \qquad j = 1, \ldots, m.$$

This primitive follows the critical-point methodology of prior cryptanalytic ReLU extraction attacks [4,2,12]. In Section 6 we evaluate the performance of the FFN extraction primitive in our end-to-end experiments.

After Stage I, all future evaluations of $g^\star$ can be replaced by local evaluations of $\widehat{g}$. Moreover, the recovered hyperplanes provide certificates for when $\widehat{g}$ is affine on a chosen line segment.

5.3 Local Affine Certificates

As discussed in Section 3.3, the bias prevents us from eliminating the FFN by the antisymmetric reduction. Nevertheless, after Stage I, the FFN is no longer an unknown obstacle: we have recovered a functionally equivalent model $\widehat{g} \equiv g^\star$ together with its effective kink hyperplanes. Since a ReLU FFN is affine inside each region of a fixed activation pattern, we can choose probes whose attention aggregates stay inside one affine region of $\widehat{g}$. On such probes, the FFN acts as

an affine map, and the scalar oracle output can be used to recover the attention weight.

Stage I returns effective kink hyperplanes

$$\widehat{h}_j(x) = \widehat{r}_j^\top x + \widehat{s}_j, \qquad j = 1, \dots, m.$$

For a point q and a direction u, we write

$$[q, q+u] = \{q + \lambda u : \lambda \in [0,1]\}$$

for the line segment between the two endpoints.

Definition 9 (Safe and non-degenerate probe) *A pair $(q,u) \in \mathbb{R}^d \times \mathbb{R}^d$ is called* safe *for $\widehat{g}$ if the endpoints q and $q+u$ lie on the same side of every recovered kink hyperplane, i.e.,*

$$\widehat{h}_j(q)\,\widehat{h}_j(q+u) > 0 \qquad \text{for every } j = 1, \dots, m. \tag{24}$$

For a safe pair, define

$$\Delta(q,u) = \widehat{g}(q+u) - \widehat{g}(q). \tag{25}$$

We call the safe pair non-degenerate *if*

$$\Delta(q,u) \neq 0. \tag{26}$$

The safety condition guarantees that the segment $[q, q+u]$ does not cross any recovered kink hyperplane, so the activation pattern of $\widehat{g}$ is fixed along the entire segment. The non-degeneracy condition excludes directions along which the extracted FFN is locally constant; in such directions, the scalar oracle output would not reveal the attention weight.

For later use, one can generate safe probes by choosing u sufficiently small. For example, define the margin

$$\rho_{\widehat{g}}(q) = \min_{1 \le j \le \widehat{m}} \frac{|\widehat{r}_j^\top q + \widehat{s}_j|}{\|\widehat{r}_j\|_2},$$

with $\rho_{\widehat{g}}(q) = +\infty$ when $\widehat{m} = 0$. If

$$\|u\|_2 < \rho_{\widehat{g}}(q),$$

then (q,u) is safe. This margin condition is only a sufficient certificate; the sign test in Def. 9 is the actual condition used by the attack.

Lemma 4 (Local affine interpolation). *Let (q,u) be safe for $\widehat{g}$. Then, for every $\lambda \in [0,1]$,*

$$\widehat{g}(q+\lambda u) = \widehat{g}(q) + \lambda\big(\widehat{g}(q+u) - \widehat{g}(q)\big). \tag{27}$$

Proof. Since (q,u) is safe, the segment $[q, q+u]$ does not cross any recovered ReLU kink hyperplane. Hence the activation pattern of $\widehat{g}$ is constant on this segment, and $\widehat{g}$ is affine there. The interpolation identity follows by evaluating this affine function at the two endpoints. □

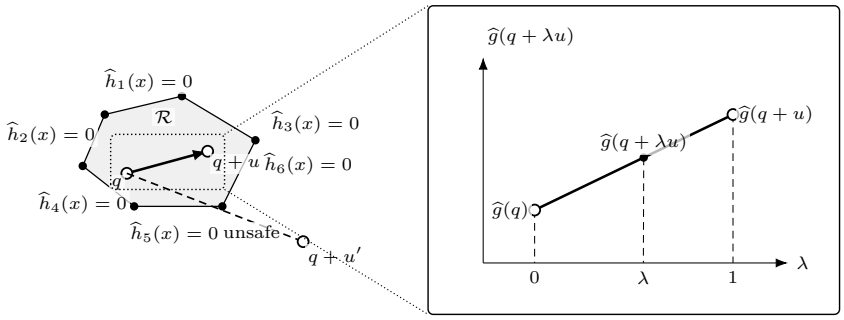

Fig. 3. Local affine certificate for a safe probe. The solid probe (q, u) stays inside one affine region $\mathcal{R}$, so the oracle value along the probe is affine in λ. The right-hand inset is an abstract local view of the oracle value along the safe probe, not a geometric magnification of the line segment. The dashed probe from q to $q + u'$ is unsafe because it crosses a recovered kink boundary.

5.4 Stage II: Recovering the Attention Matrix

We now show how one safe two-token query yields one bilinear measurement of $W^\star$.

Fix a safe pair (q, u) with $\Delta(q, u) \neq 0$, and query

$$X_{q,u} = \begin{pmatrix} (q+u)^\top \\ q^\top \end{pmatrix}. \tag{28}$$

The last token q is the query token. The two attention scores are

$$s_1 = (q+u)^\top W^\star q, \qquad s_2 = q^\top W^\star q.$$

Therefore, the attention weight on the first token is

$$\alpha_1 = \frac{\exp(s_1)}{\exp(s_1) + \exp(s_2)} = \sigma(u^\top W^\star q), \tag{29}$$

where $\sigma(t) = 1/(1 + e^{-t})$ denotes the sigmoid function.
The attention aggregate is

$$X_{q,u}^\top \alpha(X_{q,u}, W^\star) = \alpha_1(q+u) + (1-\alpha_1)q = q + \alpha_1 u.$$

Since $\alpha_1 \in (0, 1)$, this point lies on the segment $[q, q+u]$. By Lemma 4,

$$\widehat{g}(q + \alpha_1 u) = \widehat{g}(q) + \alpha_1\big(\widehat{g}(q+u) - \widehat{g}(q)\big).$$

Using $\widehat{g} \equiv g^\star$, the oracle output satisfies

$$\mathcal{O}(X_{q,u}) = \widehat{g}(q) + \alpha_1\big(\widehat{g}(q+u) - \widehat{g}(q)\big). \tag{30}$$

Hence the attention weight can be recovered as

$$\alpha_1 = \frac{\mathcal{O}(X_{q,u}) - \widehat{g}(q)}{\widehat{g}(q+u) - \widehat{g}(q)}. \tag{31}$$

Inverting the sigmoid gives the bilinear measurement

$$u^\top W^\star q = \log \frac{\mathcal{O}(X_{q,u}) - \widehat{g}(q)}{\widehat{g}(q+u) - \mathcal{O}(X_{q,u})}. \tag{32}$$

Only $\mathcal{O}(X_{q,u})$ requires a black-box query. The values $\widehat{g}(q)$ and $\widehat{g}(q+u)$ are computed locally from the extracted FFN.

5.5 Recovering the Full Matrix

It remains to solve a bilinear recovery problem. From the previous subsection, each safe and non-degenerate two-token query gives one measurement of the form

$$u^\top W^\star q.$$

We first fix an anchor $q \in \mathbb{R}^d$ and use d linearly independent probe $u \in \mathbb{R}^d$ directions to recover the vector $W^\star q$. Repeating this procedure for d linearly independent anchors then recovers the full matrix $W^\star$.

For a fixed anchor q, let $z_q = W^\star q$. Choose d linearly independent safe directions $u_1, \ldots, u_d$ such that $\Delta(q, u_j) \neq 0$ for all j. Applying Eq. (32) to each direction gives

$$\beta_j = u_j^\top W^\star q, \qquad j = 1, \ldots, d.$$

Let

$$U_q = \begin{pmatrix} u_1^\top \\ \vdots \\ u_d^\top \end{pmatrix}, \qquad \beta_q = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_d \end{pmatrix}.$$

Since U_q is invertible,

$$W^\star q = U_q^{-1} \beta_q. \tag{33}$$

To recover $W^\star$, choose d linearly independent anchors $q_1, \ldots, q_d$. For each anchor q_i, recover $y_i = W^\star q_i$ using the procedure above. Define

$$Q = \begin{pmatrix} q_1 \cdots q_d \end{pmatrix}, \qquad Y = \begin{pmatrix} y_1 \cdots y_d \end{pmatrix}.$$

Then $Y = W^\star Q$. Since Q is invertible,

$$W^\star = Y Q^{-1}. \tag{34}$$

The second stage uses one two-token oracle query for each pair (q_i, u_j); thus, after the FFN extraction stage, recovering the full attention matrix requires d^2 additional black-box queries, and the values $\widehat{g}(q_i)$ and $\widehat{g}(q_i + u_j)$ are computed locally from the extracted FFN and do not require additional oracle queries.

Algorithm 3 End-to-End Attack Procedure for BiasAwareSHA

Input: Value-query oracle $\mathcal{O}$, token dimension d, number of anchors $n_{\text{anc}} \geq d$[6]
Output: Extracted model $(\widehat{g}, \widehat{W})$

Define $\mathcal{O}_{\text{FFN}}(x) \leftarrow \mathcal{O}([x^\top])$
$(\widehat{g}, \{\widehat{h}_k\}_{k=1}^{\widehat{m}}) \leftarrow \textsc{ExtractFFN}(\mathcal{O}_{\text{FFN}})$
Choose anchors $q_1, \ldots, q_{n_{\text{anc}}}$ with $\text{rank}(Q) = d$
for $i \leftarrow 1$ to n_{anc} **do**
    Choose linearly independent safe directions $u_{i,1}, \ldots, u_{i,d}$
    for $j \leftarrow 1$ to d **do**
        $\beta_{i,j} \leftarrow \textsc{BilinearMeasure}(\mathcal{O}, \widehat{g}, q_i, u_{i,j})$
    Solve $U_i y_i = \beta_i$
Stack $Y = (y_1, \ldots, y_{n_{\text{anc}}})$
Recover $\widehat{W} \leftarrow YQ^\top (QQ^\top)^{-1}$
return $(\widehat{g}, \widehat{W})$

Combining the FFN extraction stage with the safe-probe measurements gives the end-to-end attack summarized in Algorithm 3, where BilinearMeasure denotes the two-token inversion in Eq. (32) that returns $u^\top W^\star q$ for a safe pair (q, u).

Oracle calls and limitations. The correctness of the recovery follows directly from the previous construction. Stage I gives a functionally equivalent FFN $\widehat{g} \equiv g^\star$. For every safe and non-degenerate pair (q, u), Eq. (32) gives the exact bilinear measurement $u^\top W^\star q$. For each fixed anchor q_i, d linearly independent safe directions recover $W^\star q_i$ by solving an invertible linear system. Stacking the recovered vectors for d linearly independent anchors gives $Y = W^\star Q$, and therefore $\widehat{W} = YQ^{-1} = W^\star$. Hence the extracted model

$$\widehat{\text{TF}}(X) = \widehat{g}\big(X^\top \alpha(X, \widehat{W})\big)$$

satisfies

$$\widehat{\text{TF}}(X) = g^\star\big(X^\top \alpha(X, W^\star)\big) = \text{TF}_{\theta^\star}(X)$$

for every input sequence X.

If the FFN extraction primitive uses Q_{FFN} oracle queries, then the full attack uses

$$Q_{\text{FFN}} + d^2$$

black-box queries in the exact setting: Q_{FFN} length-one queries for Stage I and a two-token query for each pair $(q_i, u_{i,j})$ in Stage II. The evaluations $\widehat{g}(q_i)$ and $\widehat{g}(q_i + u_{i,j})$ are local computations and do not require additional oracle queries. If $n > d$ anchors are used for numerical stability, the second-stage query cost becomes nd.

[6] For the exact query bound, we set $n_{\text{anc}} = d$. In finite precision, we use $n_{\text{anc}} > d$ and solve the final system in the least-squares sense to improve conditioning.

The only additional non-degeneracy required by Stage II is that the chosen safe directions satisfy

$$\widehat{g}(q_i + u_{i,j}) - \widehat{g}(q_i) \neq 0.$$

This excludes directions along which the extracted FFN is locally constant, in which case the scalar output cannot reveal the attention weight. In finite precision, we use oversampling and least-squares reconstruction of the final linear systems; the experimental section evaluates this numerical behavior.

6 End-to-End Practical Experiments

We evaluate both extraction attacks developed in this paper on victim models generated following the training pipeline of Carlini et al. [4], giving the adversary only black-box value-query access throughout.

6.1 Multi-Head Softmax Attention

The victim is implemented with separate Q/K/V projection factors; the attacker never observes these factors and recovers only the identifiable merged canonical parameters. We report the oracle query count, the empirical $(\varepsilon, 0)$-functional error on held-out inputs, and the maximum coordinate-wise parameter error after matching recovered heads to the target canonical heads.

Table 2 reports two finite-precision settings. The $G = H$ rows evaluate the case where the public head count equals the effective canonical head count. The $G < H$ rows evaluate the case where the attacker only knows an upper bound H, while the effective number of canonical heads is smaller.

Table 2. Multi-head attention extraction results.

Setting	Model	Method	Queries	$\max \lvert\theta - \widehat{\theta}\rvert$	$(\varepsilon, 0)$
float64, $G = H$	$d = 4,\ G = H = 4$	no refine	4,572	$2^{-6.75}$	$2^{-6.64}$
	$d = 4,\ G = H = 4$	refine	5,084	$2^{-49.19}$	$2^{-51.00}$
	$d = 8,\ G = H = 6$	no refine	43,284	$2^{-5.95}$	$2^{-4.09}$
	$d = 8,\ G = H = 6$	refine	45,012	$2^{-51.54}$	$2^{-50.00}$
float64, $G < H$	$d = 8,\ G = 2,\ H = 4$	refine	2,776	$2^{-52.00}$	$2^{-48.68}$
	$d = 8,\ G = 3,\ H = 6$	refine	3,892	$2^{-51.83}$	$2^{-48.00}$
	$d = 8,\ G = 4,\ H = 8$	refine	28,360	$2^{-50.68}$	$2^{-47.42}$
	$d = 8,\ G = 5,\ H = 10$	refine	123,544	$2^{-49.19}$	$2^{-47.00}$

Query counts are original value-oracle calls, including float64 oversampling and output-consistency refinement when used. Errors are reported in base-2 logarithmic scale. The parameter error is the maximum entrywise error over the recovered identifiable merged parameters.

The $G = H$ rows show that the float64 implementation recovers nontrivial trained Q/K/V victims to near machine-precision functional error after refine-

ment. The $G < H$ rows show that the public upper bound H need not equal the effective canonical head count G.

Output-consistency refinement. The optional `refine` stage uses fresh black-box outputs to correct finite-precision numerical error. When the algebraic recovery is already close to the correct solution, refinement makes a first-order linear approximation of how the model outputs change with the parameters near the recovered parameters, and then uses least squares to infer a small local parameter correction from the output residuals. In Table 2, this step turns the moderately accurate float64 algebraic reconstructions into near machine-precision functionally equivalent models.[7]

Float64 implementation and query complexity. The exact attack is algebraic, but a direct float64 implementation can be numerically unstable. The main source of error comes from the root recovery step for the denominator polynomial $Q(s)$ in Algorithm 1. When the roots of $Q(s)$ are too close to each other, small numerical errors can make it difficult to match the recovered roots to the correct attention heads. Therefore, the float64 implementation uses additional value-oracle calls for oversampled atom recovery, anchor validation, probe accept/reject tests, and oversampled least-squares solves. These stabilizing queries do not change the asymptotic dependence on H and d: with fixed implementation constants, the float64 query complexity remains $O(Hd^2)$. However, the constants are much larger than in the exact-arithmetic bound, and rejected probes still consume oracle queries. Thus, the query counts in Table 2 should be read as practical float64 query counts rather than as the minimal exact-arithmetic query count.

6.2 Bias-Aware Single-Head Transformer Block

Table 3 reports end-to-end extraction of the biased single-head Transformer block for $d = 16, m = 32$. Stage I uses $Q_{\mathrm{FFN}} = 42{,}542$ length-one queries and recovers the biased FFN. Stage II then uses $Q_{\mathrm{attn}} = n_{\mathrm{anc}} d$ two-token queries to recover the attention matrix. The reported parameter error is the maximum entrywise error over the effective extracted parameters, namely W_{attn} together with the FFN parameters.

Increasing the number of anchors gives an overdetermined final linear system and improves numerical conditioning in the attention-recovery stage, at the cost of a linear increase in Q_{attn}. In these runs, the best functional error is obtained at $n_{\mathrm{anc}} = 32$, while all three settings recover functionally equivalent models with small held-out error.

[7] Concretely, for residuals $r_i(\theta) = O(X_i) - f_\theta(X_i)$, refinement minimizes $\sum_i r_i(\theta)^2$. The implementation uses damped Gauss–Newton. It linearizes $f_{\theta+\Delta}(X_i) \approx f_\theta(X_i) + J_i\Delta$ and solves $(J^\top J + \lambda I)\Delta = J^\top r$, where $r = (r_1, \ldots, r_S)^\top$.

Table 3. Bias-aware single-head Transformer block extraction for $d = 16, m = 32$. Query counts are reported as integers.

Model	Q_{FFN}	Q_{attn}	Q_{total}	$\max \lvert\theta^\star - \widehat{\theta}\rvert$	$(\varepsilon, 0)$
$d = 16, m = 32, n_{\text{anc}} = 16$	42,542	256	42,798	$2^{-27.72}$	$2^{-22.59}$
$d = 16, m = 32, n_{\text{anc}} = 32$	42,542	512	43,054	$2^{-29.71}$	$2^{-25.89}$
$d = 16, m = 32, n_{\text{anc}} = 64$	42,542	1,024	43,566	$2^{-29.46}$	$2^{-25.46}$

7 Conclusions

In this paper, we extend cryptanalytic extraction to transformer architecture. We resolve the inherent non-identifiability of the multi-head attention by targeting a canonical representative obtained by merging indistinguishable heads and deleting zero-contribution heads. We present the first polynomial-time algorithm for extracting the parameters of the canonical multi-head attention models and give the functionally equivalent recovery. We also show that single-head Transformer-style blocks with biased ReLU FFNs remain extractable by combining biased FFN extraction with local affine certificates for two-token attention inversion.

Our finite-precision experiments support the feasibility of the algebraic attack and reveal the main numerical bottlenecks. Multi-head extraction is numerically sensitive to close rational denominator roots and small coefficients, while the biased single-head block is mainly affected by the conditioning of the final linear systems.

Future work. A natural next step is to extract Transformer-style blocks in which multi-head attention is followed by a biased FFN. The duplicate-token construction exposes a rational structure before the downstream nonlinearity, whereas the FFN can hide this structure unless the attention aggregate remains inside one certified affine region of the recovered FFN. Thus an extension would require probes that simultaneously separate the canonical heads and control the affine region of the FFN. Moreover, recovering the full value/output maps of multiple heads may require observing the block under several affine regions whose local FFN gradients span the relevant input space. We leave this combined setting as a future work.

References

1. Bhattamishra, S., Shah, K., Hahn, M., Kanade, V.: Provably learning attention with queries. arXiv preprint arXiv:2601.16873 (2026), accepted to ICML 2026
2. Canales-Martínez, I.A., Chávez-Saab, J., Hambitzer, A., Rodríguez-Henríquez, F., Satpute, N., Shamir, A.: Polynomial time cryptanalytic extraction of neural network models. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 3–33. Springer (2024)

3. Carlini, N., Chávez-Saab, J., Hambitzer, A., Rodríguez-Henríquez, F., Shamir, A.: Polynomial time cryptanalytic extraction of deep neural networks in the hard-label setting. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 364–396. Springer (2025)
4. Carlini, N., Jagielski, M., Mironov, I.: Cryptanalytic extraction of neural network models. In: Annual international cryptology conference. pp. 189–218. Springer (2020)
5. Carlini, N., Paleka, D., Dvijotham, K.D., Steinke, T., Hayase, J., Cooper, A.F., Lee, K., Jagielski, M., Nasr, M., Conmy, A., et al.: Stealing part of a production language model. arXiv preprint arXiv:2403.06634 (2024)
6. Chen, S., Li, Y.: Provably learning a multi-head attention layer. In: Proceedings of the 57th Annual ACM Symposium on Theory of Computing. pp. 1744–1754 (2025)
7. Chen, Y., Dong, X., Guo, J., Shen, Y., Wang, A., Wang, X.: Hard-label cryptanalytic extraction of neural network models. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 207–236. Springer (2024)
8. Chen, Y., Dong, X., Ma, R., Shen, Y., Wang, A., Yu, H., Wang, X.: Delving into cryptanalytic extraction of prelu neural networks. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 576–607. Springer (2025)
9. Chen, Z., Tang, S., Gao, Z., Su, Y., Qin, L., Dong, X.: Algebraic attack on convolutional neural network with max pooling. Cryptology ePrint Archive (2026)
10. Foerster, H., Mullins, R., Shumailov, I., Hayes, J.: Beyond slow signs in high-fidelity model extraction. Advances in Neural Information Processing Systems **37**, 19496–19522 (2024)
11. Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., Papernot, N.: High accuracy and high fidelity extraction of neural networks. In: 29th USENIX security symposium (USENIX Security 20). pp. 1345–1362 (2020)
12. Liu, H., Siproudhis, A., Experton, S., Lorenz, P., Boura, C., Peyrin, T.: Navigating the deep: End-to-end extraction on deep neural networks. Cryptology ePrint Archive (2026)
13. Orekondy, T., Schiele, B., Fritz, M.: Knockoff nets: Stealing functionality of black-box models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4954–4963 (2019)
14. Qi, X., Lei, H., Wei, L., Sun, X., Wang, M.: Cryptanalytic extraction of neural networks with various activation functions. IACR Trans. Symmetric Cryptol. **2026**(1), 468–505 (2026)
15. Sun, X., Lei, H., Wei, L., Qi, X., Hu, K., Wang, M., Wang, W.: Cryptanalytic extraction of convolutional neural networks. Cryptology ePrint Archive (2026)
16. Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T.: Stealing machine learning models via prediction {APIs}. In: 25th USENIX security symposium (USENIX Security 16). pp. 601–618 (2016)
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
18. Wei, L., Lei, H., Qi, X., Sun, X., Gao, L., Hu, K., Wang, W., Wang, M.: Cryptanalytic extraction of recurrent neural network models. Cryptology ePrint Archive (2026)

A Proof of Canonical Identifiability Theorem

The proof of Theorem 1 is as follows:

Proof. Assume that $f_\theta(X) = f_{\theta'}(X)$ for all inputs X. We compare the two representations after aligning their heads by attention matrices.

Let

$$\mathcal{M} = \{W_1, \dots, W_G\} \cup \{W'_1, \dots, W'_{G'}\}$$

be the set of all attention matrices appearing in either representation. For each $M \in \mathcal{M}$, define c_M as the value-vector difference assigned to M by the two representations: if both representations contain a head with attention matrix M, say (M, v_g) and (M, v'_k), then $c_M = v_g - v'_k$; if only the first representation contains M, then $c_M = v_g$; and if only the second representation contains M, then $c_M = -v'_k$. This definition is unambiguous because both representations are canonical, so each M appears at most once in each representation.

With this notation, subtracting the two models gives

$$\sum_{M \in \mathcal{M}} \alpha(X, M)^\top X c_M = 0 \qquad \text{for all } X.$$

We first show that every c_M must be zero.

For a one-token input $X = [x^\top]$, every attention weight is equal to one. Hence the previous identity gives $x^\top \sum_{M \in \mathcal{M}} c_M = 0$ for every x, and therefore $\sum_{M \in \mathcal{M}} c_M = 0$. This is exactly the cancellation of the constant term in the duplicate-token probe below.

Suppose, for contradiction, that some $c_M \neq 0$. Choose a, z so that $a^\top c_M \neq 0$ for every nonzero c_M, and so that the numbers $a^\top M z$ are distinct for distinct $M \in \mathcal{M}$. Such a choice is possible because there are only finitely many unwanted equalities to avoid.

Fix $t \neq 0$. Applying the duplicate-token calculation from Lemma 3 to the zero function $\sum_{M \in \mathcal{M}} \alpha(X, M)^\top X c_M$, we get, for every positive integer s,

$$\sum_{M \in \mathcal{M}} \frac{(a^\top c_M) e^{t a^\top M z}}{s + e^{t a^\top M z}} = 0.$$

Set $r_M = e^{t a^\top M z}$. By our choice of a, z, the numbers r_M are distinct. Thus

$$\sum_{M \in \mathcal{M}} \frac{(a^\top c_M) r_M}{s + r_M} = 0$$

for infinitely many values of s.

We use the elementary fact that the functions $r/(s+r)$ with distinct nonzero r's cannot cancel each other. Indeed, after multiplying by $\prod_M (s + r_M)$, we get a polynomial that vanishes at infinitely many points; evaluating it at $s = -r_M$ shows that the corresponding coefficient $a^\top c_M$ must be zero. Hence $a^\top c_M = 0$

for every M, contradicting our choice of a whenever $c_M \neq 0$. Therefore $c_M = 0$ for every $M \in \mathcal{M}$.

It remains to translate this back to heads. If a matrix W_g appeared only in the first representation, then $c_{W_g} = v_g$, which is nonzero because the representation is canonical. This is impossible since we have shown $c_{W_g} = 0$. Hence W_g must also appear in the second representation. Since the second representation is canonical, there is a unique k such that $W'_k = W_g$. For this k, the identity $c_{W_g} = 0$ gives $v_g = v'_k$.

The same argument in the other direction shows that every head of θ' is matched by exactly one head of θ. Therefore $G = G'$, and after a permutation of heads,

$$W_g = W'_g, \qquad v_g = v'_g \qquad \text{for all } g = 1, \ldots, G.$$

This proves the theorem. □