How do Small Transformer Models Learn Hard Math Tasks?

A mechanistic interpretability study of transformers learning a cryptography task

AUTHORS

Eshika Saxena, Kristin Lauter

AFFILIATION

FAIR at Meta

TL;DR

Transformers trained on a cryptography problem can recover the secret even with ~0% prediction accuracy by storing the information in their positional embeddings; we apply sparsity regularization to turn this into a directly extractable representation.

Background / Motivation

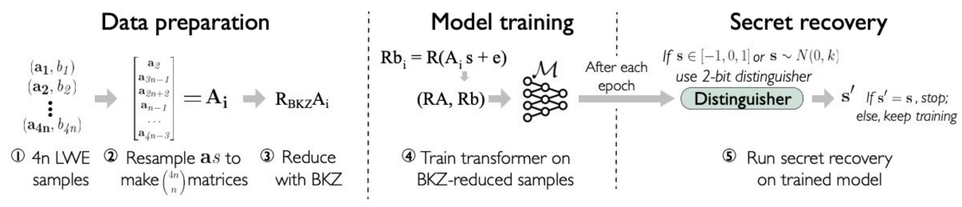
- Learning with Errors (LWE) underpins many post-quantum cryptographic (PQC) schemes, as they rely on the hardness of recovering the secret s.

LWE Encryption

$$(\mathbf{A} \cdot \mathbf{s} + \mathbf{e}) \bmod q = \mathbf{b}$$

$\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is a uniformly random $m \times n$ matrix
$\boldsymbol{s}$ = secret, $\boldsymbol{s} \in \mathbb{Z}_q^n$
$\boldsymbol{e}$ = error, $\boldsymbol{e} \in \mathbb{Z}_q^n$, $\boldsymbol{e} \sim N(\mu, \sigma)$
q = modulus, typically a large prime

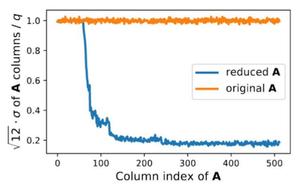
- Prior ML attacks (SALSA[5], PICANTE[3], VERDE[2], FRESCA[8]) train transformers on (a, b) pairs, then use a post-hoc "distinguisher" to recover s by probing the model's sensitivity to input perturbations[5].



- This work ***opens the black box*** on where/how the secret gets embedded in the encoder-only transformers that are able to recover the secret.

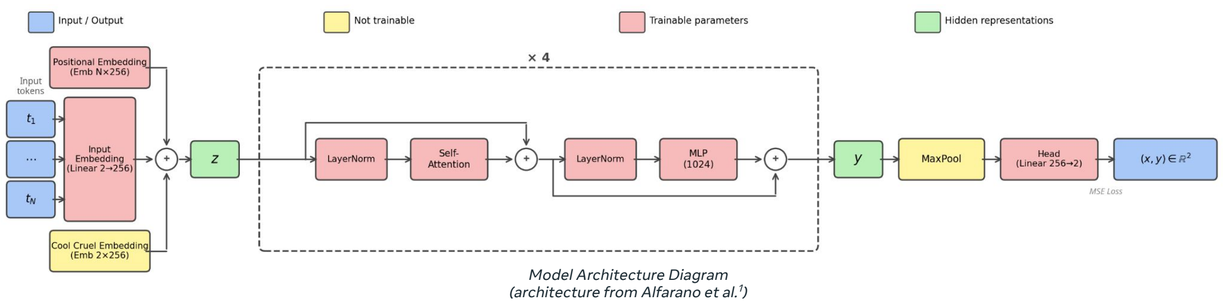
Experimental Setup

- **Problem**: LWE with N = 256, $\log_2 q \approx 20$ (q = 842779), Gaussian error σ = 3.0
- **Secrets**: 30 randomly generated binary secrets, Hamming weights h ∈ {20, 21, 22} (10 each), chosen near but below the learnability threshold.
- **Data**: Open-sourced TAPAS dataset[6]; preprocessed via lattice reduction (BKZ + Flatter), which creates a variance "cliff" → high-variance "cruel" columns vs. low-variance "cool" columns[7].



Reduced LWE data has cruel (unreduced) and cool bits [7]

- **Model**: Encoder-only transformer (FRESCA-style[1, 8]). Angular embeddings $x \mapsto (\cos 2\pi x/q, \sin 2\pi x/q)$; 4 layers, 4 heads, hidden dim 256, GELU, max-pooling then linear head → $\mathbb{R}^2$.
- **Training**: One model per secret; 50 epochs, ~1M samples, single H100/H200 GPU, ~2.5 hrs each.



Model Architecture Diagram (architecture from Alfarano et al.[1])

Methods / Results

The Prediction Paradox

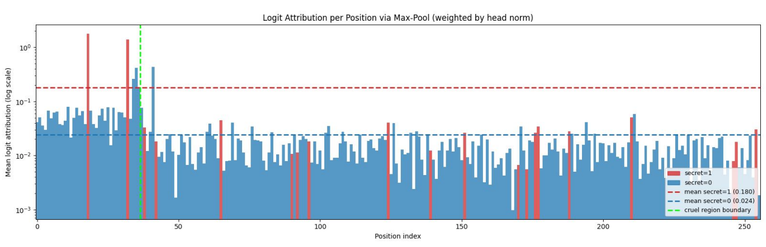
- Standard performance metrics show low accuracy and high loss/error across the models, regardless of secret recovery
- Error distribution is roughly uniform, not zero-centered, so there is no behavioral evidence the model learned LWE arithmetic in a generalized way.

Status	# Models	Eval Loss (↓)	MAE / q (↓)	% Acc (15% margin) (↑)
Recovered	13	$\mathbf{1.04 \pm 0.05}$	$\mathbf{0.22 \pm 0.01}$	$\mathbf{38.2 \pm 3.0}\%$
Not recovered	17	1.08 ± 0.04	0.24 ± 0.01	$32.2 \pm 3.1\%$

Logit Attribution

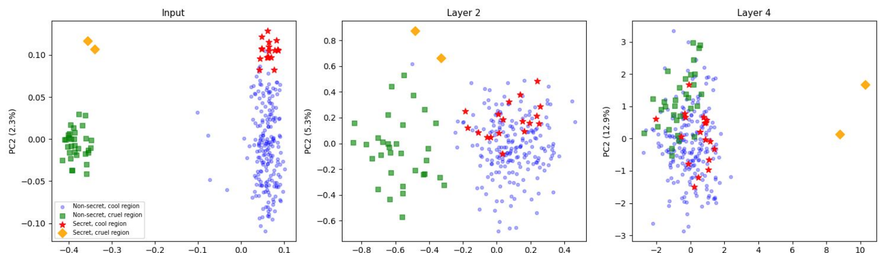
- Secret positions not only "win" max-pooling more often, they dominate the hidden dimensions that matter most for the output.
- Attribution ratio suggests that the model genuinely uses the secret to predict b.

Status	Recovered	Not recovered
# Models	13	17
Attribution ($\mathbf{s}_i = 0$)	0.03 ± 0.01	0.07 ± 0.02
Attribution ($\mathbf{s}_i = 1$)	0.16 ± 0.05	0.12 ± 0.07
Attribution Ratio ($\mathbf{s}_i = 1/\mathbf{s}_i = 0$)	$\mathbf{5.35 \pm 2.39}$	2.32 ± 2.31



Residual Stream Analysis

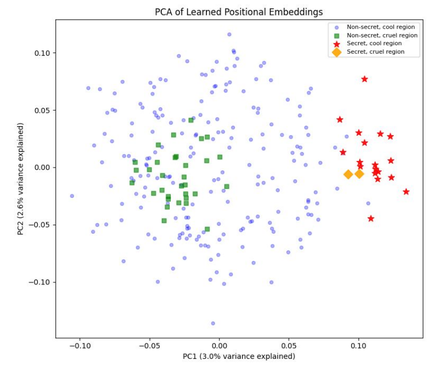
- Surprising finding: the secret/non-secret + cool/cruel clusters emerge immediately after the input layer before any self-attention.
- Later layers actually obfuscate them, although we still see outliers from the secret indices in the "cruel" region.
- Evidence that positional embeddings are the primary storage mechanism for the secret.



Positional Embedding Weight Analysis

- PCA directly on the learned positional-embedding weights shows secret indices cluster distinctly along PC1.
- Recovered models consistently show higher silhouette & separation scores than failed ones.

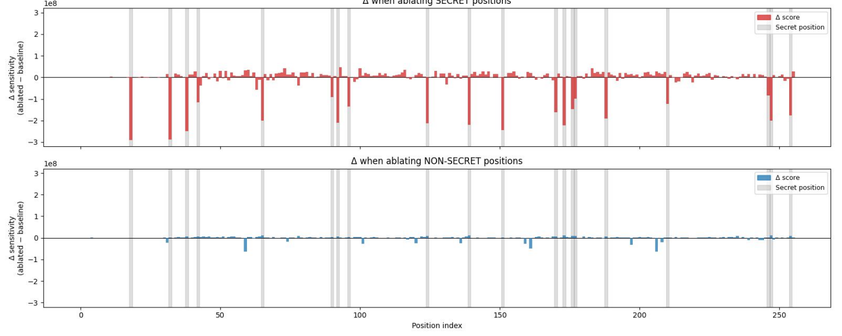
		Recovered	Not recovered
# Models		13	17
2 Clusters	**Silhouette**	$\mathbf{0.0264 \pm 0.0130}$	0.0076 ± 0.0108
	Sep. Ratio	$\mathbf{0.43 \pm 0.06}$	0.28 ± 0.09
4 Clusters	**Silhouette**	$\mathbf{0.0034 \pm 0.0125}$	-0.0015 ± 0.0087
	Sep. Ratio	$\mathbf{0.81 \pm 0.21}$	0.52 ± 0.11



Causal Intervention

- Zeroing out positional embeddings at secret indices collapses recovery
- Ablating the same number of non-secret positions has near-zero effect (19.4 → 18.7).
- The secret seems to be encoded specifically in the positional embeddings of the secret support, not in a distributed representation.

Status	Recovered	Not recovered
# Models	13	17
Baseline	19.4 ± 2.1	6.5 ± 6.9
Ablate Secret	0.8 ± 2.8	0.0 ± 0.0
Ablate Non-secret	18.7 ± 3.6	6.31 ± 7.2



Architectural Intervention: Factored, Sparse Positional Embedding

- Decompose each positional embedding into magnitude α + unit direction v: pos_emb[i] = $\alpha_i \cdot v_i$.
- Add an L1 sparsity penalty on the α's (sum of all-but-top-h magnitudes, β = 0.01) → drives non-secret magnitudes toward zero.
- Result: secret indices isolate within the top-h to top-2h α magnitudes (the very top is taken by high-variance "cruel" positions), and this structure emerges within ~10 epochs → secret becomes recoverable by direct parameter inspection.

Conclusion

Key Findings

- Encoder-only transformers solving LWE bypass algorithmic reasoning and embed the secret directly in their positional embeddings — a form of structured memorization induced by the high-noise, combinatorial task.
- This insight enables an interpretable-by-design architecture: sparsity regularization turns expensive post-hoc secret recovery into transparent parameter inspection.
- *Broader impact*: Understanding how ML attacks on LWE succeed contributes to the long-term security assessment of post-quantum cryptographic standards.

Limitations & Future Work

- Restricted to regimes where SALSA-family attacks already succeed, not yet deployed-crypto parameters. Analysis depends on lattice-reduced data.
- L1 prior assumes sparse secrets; benefit diminishes as secrets densify.
- Future work: behavior on unreduced data, denser secrets, and other LWE settings.

References

1. Alberto Alfarano, Eshika Saxena, Emily Wenger, Francois Charton, and Kristin Lauter. Improving ML attacks on LWE with data repetition and stepwise regression. In Proc. of *ICML, 2026.*
2. Cathy Yuanchen Li, Emily Wenger, Zeyuan Allen-Zhu, Francois Charton, and Kristin Lauter. SALSA VERDE: a machine learning attack on Learning With Errors with sparse small secrets. In Proc. of *NeurIPS*, 2023.
3. Cathy Yuanchen Li, Jana Sotáková, Emily Wenger, Mohamed Malhou, Evrard Garcelon, François Charton, and Kristin Lauter. Salsa Picante: A Machine Learning Attack on LWE with Binary Secrets. In Proc. of *ACM CCS,* 2023.
4. Emily Wenger, Eshika Saxena, Mohamed Malhou, Ellie Thieu, and Kristin Lauter. Benchmarking attacks on learning with errors. In Proc. of *IEEE S&P*, 2025.
5. Emily Wenger, Mingjie Chen, François Charton, and Kristin E Lauter. Salsa: Attacking lattice cryptography with transformers. In Proc. of NeurIPS, 2022
6. Eshika Saxena, Alberto Alfarano, Francois Charton, Emily Wenger, and Kristin Lauter. TAPAS: Datasets for Learning the Learning with Errors Problem. In Proc. of *NeurIPS,* 2025.
7. Niklas Nolte, Mohamed Malhou, Emily Wenger, Samuel Stevens, Cathy Li, Francois Charton, and Kristin Lauter. The cool and the cruel: separating hard parts of lwe secrets. In Proc. of *AFRICACRYPT*, 2024.
8. Samuel Stevens, Emily Wenger, Cathy Yuanchen Li, Niklas Nolte, Eshika Saxena, Francois Charton, and Kristin Lauter. Salsa fresca: Angular embeddings and pre-training for ml attacks on learning with errors. In *TMLR*, 2025.