TIGER: INVERTING MULTIMODAL TRANSFORMER GRADIENTS VIA EMBEDDING-SUBSPACE DISTANCE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) allows multiple clients to jointly train a shared model by sending gradient updates to a central server while keeping their raw inputs local. Gradient inversion attacks, however, have shown that these updates can leak enough information to reconstruct client inputs. For transformers, though, existing attacks fall short in some of the following ways: they scale poorly to realistic model sizes, larger inputs, or non-causal attention; they require an enumerable vocabulary of token embeddings, preventing the recovery of continuous non-textual inputs; or they are brittle under numerical noise, e.g., from quantization or Differential Privacy (DP). In this paper, we introduce TIGER, a continuous gradient inversion attack that exploits the natural low-rank structure of attention gradients. Its novel objective aligns the features of dummy inputs across layers with the low-rank gradient subspaces, resulting in a single framework that recovers text from both causal and bidirectional attention models and, for the first time, multimodal inputs. On settings featuring bidirectional attention, TIGER substantially improves reconstruction quality over prior work and can recover images at roughly 720p resolution in the multimodal setting. On causal models, TIGER is significantly more robust than prior attacks, enabling accurate reconstructions even under quantization and DP-style defenses. Overall, TIGER uncovers further privacy vulnerabilities in LLMs, underscoring the need for stronger defenses in deployed FL systems.

1 INTRODUCTION

Federated Learning (FL) has emerged as a practical paradigm for collaboratively training machine-learning models over decentralized data. In FL, clients keep their raw data local and communicate only model updates to a central server that aggregates them to improve a shared model (McMahan et al., 2017). This is particularly appealing for large language models (LLMs), whose training requires large amounts of data, and whose adaptation often requires domain-specific information that may be sensitive or proprietary (Fan et al., 2023; Chen et al., 2024; Yao et al., 2024; Jiang et al., 2025). The privacy requirement is particularly strong in privacy-critical domains such as medicine (Manoel et al., 2023; Zhang & Li, 2025; Li et al., 2026; Sadilek et al., 2021), law (Zhang et al., 2023; Yue et al., 2024), and finance (Gao et al., 2025; Byrd & Polychroniadou, 2020), where centralizing training data may be infeasible due to confidentiality requirements or regulatory constraints.

Gradient inversion attacks Keeping raw data with the client, however, does not by itself guarantee privacy, as gradient inversion attacks have been shown to successfully reconstruct the client data from the shared model updates (Zhu et al., 2019; Zhao et al., 2020; Geiping et al., 2020). These attacks have been highly successful on images (Zhu et al., 2019; Zhao et al., 2020; Geiping et al., 2020; Yin et al., 2021; Jeon et al., 2021; Hatamizadeh et al., 2022; Kariyappa et al., 2023), and have since been extended to text (Deng et al., 2021; Balunovic et al., 2022; Gupta et al., 2022; Li et al., 2023; Petrov et al., 2024). Yet textual reconstruction remains challenging on modern LLMs, with existing attacks relying on expensive gradient-matching optimization (Zhu et al., 2019; Deng et al., 2021; Balunovic et al., 2022), strong language-model priors (Balunovic et al., 2022; Gupta et al., 2022), or discrete filtering that is sensitive to numerical perturbations (Petrov et al., 2024), and none of them extend to the multimodal setting. Their apparent failure in realistic FL deployments, with nontrivial batch sizes and quantized or defended gradients, may thus underestimate the real privacy risk.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

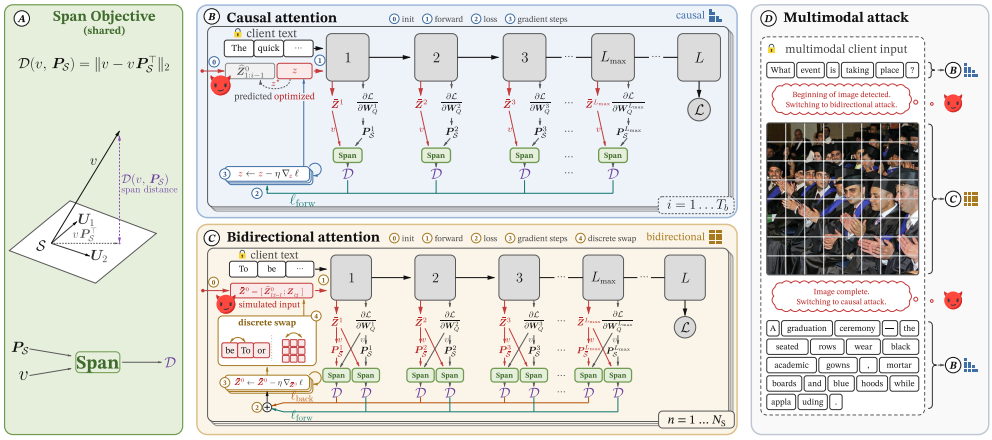

Figure 1: Overview of TIGER. **(a)** The span distance $\mathcal{D}(v,\, \boldsymbol{P}_{\mathcal{S}})$ of a vector v to a subspace $\mathcal{S}$ (SPAN block). **(b)** The causal attack, recovering tokens sequentially ($i = 1 \dots T_b$) with only the forward loss ℓ_{forw}. **(c)** The bidirectional attack, jointly optimizing embeddings on two subspace-alignment objectives, then refining via discrete token swaps. **(d)** The multimodal attack, interleaving **(b)** and **(c)**.

This work: Continuous attack on LLMs in multimodal and defended settings To address these limitations, we develop TIGER (**T**ransformer Data **I**nversion from **G**radients via **E**mbedding-subspace **R**econstruction), a novel gradient inversion attack that reconstructs private transformer training data. TIGER provides a continuous recovery framework, which improves robustness in settings that arise in practical FL, as well as extending the attack surface to multimodal inputs.

TIGER builds on the observation that gradients of transformer linear layers contain exploitable low-rank structure. In particular, the linear layer inputs span the same subspace as the columns of the corresponding weight gradient under mild conditions. TIGER turns this signal into a differentiable objective, illustrated in panel A of Figure 1, directly optimizing continuous embeddings so that their hidden representations align with the subspaces revealed by the observed gradients. This formulation makes the attack robust to numerical imprecision and allows for the recovery of continuous inputs.

For causal transformers (B, Figure 1) TIGER exploits causal attention: as the representation at a position depends only on preceding tokens, it recovers embeddings sequentially from subspace constraints in the first few transformer layers. Under bidirectional attention (C, Figure 1), such sequential recovery is unavailable. TIGER therefore recovers the batch jointly, alternating continuous updates with discrete token swaps, under a bidirectional alignment objective that also requires the observed gradient directions to lie in the optimized embedding span, preventing collapse into repeated tokens or duplicate sequences. Interleaving these two mechanisms allows us to recover multimodal inputs, where the text is causal but, for example, images are not (D, Figure 1). Finally, optimizing against the gradient-induced subspaces rather than matching full model gradients requires only partial forward and backward passes, making reconstruction substantially more efficient.

Evaluation and Results We evaluate TIGER on modern LLMs from the Gemma and Llama families, covering causal, bidirectional, and multimodal architectures used in contemporary fine-tuning applications (Google Research & Google DeepMind, 2025; Sellergren et al., 2026; Huang et al., 2026; Zaitsev, 2026). Compared to prior work, which collapses under DP-style defenses, TIGER continues to recover meaningful text, achieving ROUGE-1 (Lin, 2004) of more than 70 up to $\sigma = 10^{-3}$. It recovers large images with legible PII up to 720p resolution, along with the accompanying text, and scales attacks on encoder models to batches of 32 sequences. Finally, TIGER remains robust under quantization and across datasets of varying difficulty.

Key contributions Our main contributions include:

- TIGER, a robust and scalable continuous gradient inversion attack for LLMs that recovers private data across a range of challenging settings.
- A novel embedding-subspace objective derived from the low-rank gradient structure (§4).
- A broad evaluation on modern LLMs, showing robust recovery of text and multimodal data (§5).
- An open-source implementation of TIGER to facilitate future research

2 Related Work

In this section, we review the most relevant prior work on gradient leakage settings and attacks, text-specific reconstruction methods, and low-rank inversion techniques.

Gradient inversion Federated learning (McMahan et al., 2017) enables multiple clients to train a shared model through a central server without transmitting raw training data. However, this paradigm does not guarantee privacy, as the parameter updates they send can still leak substantial information about their input examples through gradient inversion attacks. Such attacks are commonly distinguished by the adversary's control over the training process. In the *honest-but-curious* setting, the server follows the prescribed protocol but attempts to infer client data from the updates it observes (Zhu et al., 2019; Zhao et al., 2020; Zhu & Blaschko, 2021; Geng et al., 2021; Geiping et al., 2020; Deng et al., 2021; Balunovic et al., 2022; Gupta et al., 2022; Li et al., 2023; Petrov et al., 2024; Dimitrov et al., 2024; Bakarsky et al., 2025), whereas in the *malicious-server* setting, it actively modifies the model shared with the clients to amplify leakage (Fowl et al., 2022; Wen et al., 2022; Fowl et al., 2023; Chu et al., 2023; Sami et al., 2026). The latter enables substantially stronger extraction, but relies on a stronger adversary and may be detectable if clients audit the model or protocol. We therefore focus on the more restrictive and realistic honest-but-curious setting.

Optimization-based gradient inversion A central class of honest-but-curious attacks formulates reconstruction as an optimization problem, iteratively updating dummy inputs, and sometimes labels, so that their induced gradients match the observed updates (Zhu et al., 2019; Geiping et al., 2020; Zhao et al., 2020). This has found success in continuous domains such as images, where priors can further regularize the reconstruction. Textual data is more challenging, as its discreteness forces optimized embeddings back onto valid vocabulary tokens, over a search space growing with the model, vocabulary, sequence length, and batch size. Consequently, they often scale poorly to LLMs and larger inputs and batches (Deng et al., 2021; Balunovic et al., 2022; Petrov et al., 2024).

Recovering text Several works adapt gradient inversion to text by incorporating language-specific structure: TAG (Deng et al., 2021) extends gradient-matching techniques to textual inputs, while later methods combine continuous optimization with language priors, discrete search, or model-specific signals such as activations and pooling layers (Balunovic et al., 2022; Gupta et al., 2022; Li et al., 2023). Most of these additions, however, do not transfer to other modalities, e.g. due to requiring a finite token vocabulary or an LLM-based prior, weakening their use on multimodal inputs.

Exact and low-rank gradient inversion A complementary line of work seeks exact recovery by exploiting algebraic structure in gradients. In the honest-but-curious setting, APRIL (Lu et al., 2022) achieves it for transformers, but only at batch size one. A broader class of methods instead exploits the low-rank structure of linear-layer gradients (Drencheva et al., 2025), which exposes the subspaces spanned by the client inputs, enabling exact batch recovery, albeit under additional assumptions. SPEAR and SPEAR++ require ReLU-induced sparsity, which limits their applicability to standard transformers (Dimitrov et al., 2024; Bakarsky et al., 2025). DAGER (Petrov et al., 2024) exploits the discrete nature of text to recover large batches from LLMs by filtering candidate tokens, and consequently sequences, but these filters are numerically sensitive and hence brittle under quantization or DP-SGD (Abadi et al., 2016) noise. Furthermore, its reliance on a finite vocabulary makes it inapplicable to multimodal inputs with continuous embeddings.

TIGER builds on the same low-rank structure, but turns it into a continuous optimization objective, which retains the low-rank signal while improving robustness and generalizing across input types.

3 Background and Notation

In this section, we define our threat model (§3.1), and then recall the transformer architecture (§3.2), and how the gradient structure is exploited by DAGER (Petrov et al., 2024) and, consequently, TIGER (§3.3). For convenience, we include a notation table in Table 5 in §A.

3.1 Threat Model

We follow the standard assumptions in the honest-but-curious federated learning setting, where the attacker knows the model architecture and current parameter values, and observes the client's gradients for all trainable parameters except the embedding layer. For causal models, we only require knowing the batch size, B, while for bidirectional attention, we follow prior optimization-based gradient-inversion attacks (Balunovic et al., 2022; Deng et al., 2021; Zhu et al., 2019; Geiping et al.,

2020) and assume knowledge of the sequence lengths $\{T_b\}_{b=1}^{B}$. For images, this manifests as the number of image tokens, but not its location within the text. These quantities determine the total number of non-padding tokens T and the attention mask $\boldsymbol{M}$. Unlike most optimization-based attacks, TIGER does not require the true training labels, nor does it optimize over dummy labels during reconstruction. We present a full comparison against the assumptions of prior attacks in §D.

3.2 TRANSFORMERS

We study transformer models with hidden dimension d, vocabulary $\mathcal{V}$ of size V, associated embedding vectors $\boldsymbol{E} = \{e_v : v \in \mathcal{V}\}$, and L transformer blocks. A client batch contains B token sequences, of which the b-th sequence has length T_b. We write $t_{b,i} \in \mathcal{V}$ for the token at position i of the b-th client sequence, and $\hat{t}_{b,i}$ for the respective recovered token. We denote the length of the longest sequence by $T_{\max} = \max_{b\in\{1,\dots,B\}} T_b$ and the total number of non-padding tokens in the batch by $T = \sum_{b=1}^{B} T_b$.

We represent with $\boldsymbol{Z}^l \in \mathbb{R}^{T\times d}$ the input embeddings to the l-th transformer block, where each row corresponds to one token in the flattened batch. The query, key, and value representations of $\boldsymbol{Z}^l$ in block l are $\boldsymbol{Q}^l = \boldsymbol{Z}^l\boldsymbol{W}_Q^l$, $\boldsymbol{K}^l = \boldsymbol{Z}^l\boldsymbol{W}_K^l$, and $\boldsymbol{V}^l = \boldsymbol{Z}^l\boldsymbol{W}_V^l$, where $\boldsymbol{W}_Q^l, \boldsymbol{W}_K^l, \boldsymbol{W}_V^l \in \mathbb{R}^{d\times d}$ are parameters. Given an attention mask $\boldsymbol{M} \in \mathbb{R}^{T\times T}$, the corresponding self-attention is given by:

$$\mathrm{Attn}(\boldsymbol{Q}^l, \boldsymbol{K}^l, \boldsymbol{V}^l; \boldsymbol{M}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}^l(\boldsymbol{K}^l)^\top}{\sqrt{d}} + \boldsymbol{M}\right)\boldsymbol{V}^l,$$

We use the additive convention: entries of $\boldsymbol{M}$ are 0 for allowed attention edges and $-\infty$ for disallowed ones. Namely, tokens from different sequences do not attend to one another, those under bidirectional attention all attend to each other, and those under causal attention only attend to previous positions.

We denote the full transformation computed by block l with $f^l : \mathbb{R}^{T\times d} \times \mathbb{R}^{T\times T} \to \mathbb{R}^{T\times d}$, where $\boldsymbol{Z}^{l+1} = f^l(\boldsymbol{Z}^l; \boldsymbol{M})$ for $l \in \{1, \dots, L\}$. It includes the attention mechanism and the components surrounding it, such as RoPE, normalizations, multi-head projections, residual connections, and MLP layers, all of which are known and can be evaluted by the attacker.

Finally, we distinguish the raw input embedding matrix from the input to the first transformer block. Let $\boldsymbol{Z}^0 \in \mathbb{R}^{T\times d}$ denote the matrix of token embeddings before any model-specific preprocessing. For the client batch, text tokens map to some vocabulary embedding $e_v \in \boldsymbol{E}$, while raw multimodal inputs $\boldsymbol{X}^{\mathrm{mm}}$ are mapped by a known encoder f^{mm} into a contiguous block of embeddings, which are typically placed under bidirectional attention within that block, even in otherwise causal models. For images, f^{mm} is typically such that each token is produced by a fixed-size image patch, so each one corresponds to a specific patch of $\boldsymbol{X}^{\mathrm{mm}}$. We denote by f^0 the known preprocessing applied before the first transformer block, such as adding positional embeddings, giving $\boldsymbol{Z}^1 = f^0(\boldsymbol{Z}^0; \boldsymbol{M})$. In TIGER, we optimize a continuous variable $\bar{\boldsymbol{Z}}^0 \in \mathbb{R}^{T\times d}$ in place of $\boldsymbol{Z}^0$, by propagating it through the same known maps, $\bar{\boldsymbol{Z}}^1 = f^0(\bar{\boldsymbol{Z}}^0; \boldsymbol{M})$ and $\bar{\boldsymbol{Z}}^{l+1} = f^l(\bar{\boldsymbol{Z}}^l; \boldsymbol{M})$, and denote the resulting recovered hidden states by $\bar{\boldsymbol{Z}}^l$. We write $\boldsymbol{Z}_i^l \in \mathbb{R}^d$ for the i-th row of $\boldsymbol{Z}^l$, and $\bar{\boldsymbol{Z}}_{b,i}^l \in \mathbb{R}^d$ for the recovered hidden state at layer l of the i-th token inside the b-th sequence.

3.3 SELF-ATTENTION GRADIENTS

We now recall the low-rank property of the linear layer gradients. As the query projection in block l is $\boldsymbol{Q}^l = \boldsymbol{Z}^l\boldsymbol{W}_Q^l$, the gradient of the loss $\mathcal{L}$ with respect to the query projection weights factors as:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_Q^l} = (\boldsymbol{Z}^l)^\top \frac{\partial \mathcal{L}}{\partial \boldsymbol{Q}^l},$$

as shown by DAGER (Petrov et al., 2024). As $\boldsymbol{Z}^l, \partial\mathcal{L}/\partial\boldsymbol{Q}^l \in \mathbb{R}^{T\times d}$, this implies $\mathrm{rank}(\partial\mathcal{L}/\partial\boldsymbol{W}_Q^l) \le T$, i.e., the query gradient is rank-deficient whenever $T < d$. Under a mild full-rank assumption on the backpropagated query gradients, DAGER shows that it reveals the span of the block inputs $\boldsymbol{Z}^l$:

Theorem 3.1 (Theorem 5.1 of DAGER)**.** *If $T < d$ and the matrix $\partial\mathcal{L}/\partial\boldsymbol{Q}^l$ is of full rank T, then*

$$\mathrm{rowspan}(\boldsymbol{Z}^l) = \mathrm{colspan}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_Q^l}\right). \tag{1}$$

Every row of $\boldsymbol{Z}^l$ therefore lies in the column span of the observed query gradient. We denote this span, an orthonormal basis for it, and the projection matrix it induces by:

$$\mathcal{S}^l = \mathrm{colspan}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_Q^l}\right),\ \boldsymbol{U}^l = \mathrm{basis}(\mathcal{S}^l),\ \boldsymbol{P}^l = \mathrm{proj}\,\mathcal{S}^l = (\boldsymbol{U}^l)^\top\boldsymbol{U}^l.$$

DAGER exploits Theorem 3.1 by testing whether candidate token or sequence embeddings belong to $\mathcal{S}^l$, as incorrect ones are unlikely to lie in it when $T < d$. In practice, it replaces this membership check for a candidate $z \in \mathbb{R}^d$ with a distance-to-subspace computation (Equation 2 in DAGER):

$$\mathcal{D}(z, \boldsymbol{P}^l) = \left\| z - z(\boldsymbol{P}^l)^\top \right\|_2 .$$

Here, $\boldsymbol{U}^l$ comes from a truncated SVD of $\partial\mathcal{L}/\partial\boldsymbol{W}_Q^l$, keeping left singular vectors with singular values above a threshold, and z is accepted when $\mathcal{D}(z, \boldsymbol{P}^l)$ is below a layer-dependent tolerance.

TIGER uses the same distance $\mathcal{D}$ without the discrete test. Rather than enumerating and thresholding candidates, it optimizes the input so its hidden states approach the gradient subspaces.

4 TIGER

We now detail the TIGER attack procedure. We first describe the shared components across attacking causal and bidirectional attention, including the objective and initialization strategy (§4.1). We then give the specific algorithms for causal and full attention attacks in §4.2 and §4.3, respectively.

4.1 Recovery Objective

The common structure of TIGER is the same, regardless of the attention mechanism. For each attacked layer $l \leq L_{\max}$, we use the gradient-induced subspace basis $\boldsymbol{U}^l$ and its projection $\boldsymbol{P}^l$, extracted from $\partial\mathcal{L}/\partial\boldsymbol{W}_Q^l$ by the truncated SVD of §3.3, whose singular-value threshold sets the rank.

In TIGER, we optimize raw input embeddings $\bar{\boldsymbol{Z}}^0$ so that their respective hidden states $\bar{\boldsymbol{Z}}^l$ induced by the known transformer blocks f^l align with the observed gradient-induced subspaces for all $l \leq L_{\max}$. Since the Theorem 3.1 condition only constrains vector directions and not their size, we normalize each recovered hidden state $\bar{\boldsymbol{Z}}_{b,i}^l$ to prevent trivial near-zero solutions that yield a lower distance without recovering a meaningful direction. Our forward span-distance loss is then given by:

$$\ell_{\text{forw}} = \sum_{l=1}^{L_{\max}} \sum_{(b,i)\in\Omega} \mathcal{D}\left(\frac{\bar{\boldsymbol{Z}}_{b,i}^l}{\|\bar{\boldsymbol{Z}}_{b,i}^l\|_2}, \boldsymbol{P}^l \right)^2 , \tag{2}$$

where Ω denotes the token positions that are being optimized in the current instance. For tokens under bidirectional attention, this includes all tokens within the given attention block, while for causal attention sequences, we attack individual tokens, provided our attack has already recovered all preceding ones. We sum over multiple, but not all, layers to provide additional constraints and stabilize our optimization, while keeping the attack computationally efficient.

After the continuous optimization ends, for text, we map each recovered embedding to the nearest token embedding vector via cosine similarity, as described in Algorithm 6 in §B.

Stability of the objective optimum While Theorem 3.1 shows that low-rank gradients reveal the client input subspaces, it says nothing about whether the true input is a global optimum of Equation (2), or whether this optimum is non-degenerate. The true hidden states always attain zero loss, but this minimum is stable only if the probed layers jointly impose enough constraints. The first layer confines each input to the T-dimensional observed span, leaving T unknowns per input, and each further layer adds $d - T$ constraints on them. Counting both yields the following necessary condition, proven in §F:

Corollary 4.1 (Transformers)**.** *Let a transformer of width d be attacked on $T < d$ non-padding tokens through the query gradients of its first L_{max} blocks, with $\operatorname{rank}(\boldsymbol{Z}^l) = \operatorname{rank}(\partial\mathcal{L}/\partial\boldsymbol{Q}^l) = T$ for $l \leq L_{max}$. Then $\boldsymbol{Z}^1$ is a global minimizer of the span objective, and it can be a stable minimizer, around which the objective grows quadratically, only if $(L_{max} - 1)(d - T) \geq T$.*

This condition is equivalent to $T \leq \frac{L_{\max}-1}{L_{\max}} d$, meaning the number of tokens we can stably recover grows with the number of probed layers, with diminishing returns for larger $L_{\max}$. Under stronger architectural assumptions, this condition is also sufficient for the true client batch to be the unique global minimum up to permutation and sample duplication (§F). Recovering $\boldsymbol{Z}^1$ in turn yields $\boldsymbol{Z}^0$ up to the invariances of the preprocessing: positional encodings lose nothing, while RMSNorm discards the input scale. We account for this by mapping embeddings to tokens via cosine similarity for text, and by using scale-aligned image metrics, as described in §5.1.

Initialization The objective in Equation (2) is highly non-convex, so the value it converges to depends strongly on the initialization of $\bar{\boldsymbol{Z}}^0$. Thus, TIGER does N_{init} reconstructions from N_{init} independent initializations, using lightweight priors. For text, we take fixed-length token sequences from WikiText and estimate per-position mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$ of the embeddings of all tokens at that position, and sample from $\bar{\boldsymbol{Z}}^0_{b,i} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ as our prior. For images, we sample pixels i.i.d. from $\mathcal{N}(0.5, \sigma^2)$, using $\sigma = 1.0$ for half the initializations and $\sigma = 0.1$ for the other half, to cover both uniform and diverse images. We use these priors by default, and ablate them in §E.2.

4.2 Causal Attack

Objective For causal architectures, we leverage the causal attention mask to recover tokens sequentially. Under the decoder mask $\boldsymbol{M}_{\text{dec}}$, the hidden states of a prefix are independent of tokens that appear later in the same sequence. Formally, for any prefix length i and any layer l,

$$f^l\left(\bar{\boldsymbol{Z}}^l_{b,1:i}; \boldsymbol{M}_{\text{dec}}\right) = f^l\left(\bar{\boldsymbol{Z}}^l; \boldsymbol{M}_{\text{dec}}\right)_{b,1:i}. \tag{3}$$

We can therefore optimize the loss ℓ_{forw} for individual tokens, assuming a recovered prefix. The loss for optimizing the embedding of token i in sequence b, $\bar{\boldsymbol{Z}}^0_{b,i}$, is given by:

$$\ell^{(b,i)}_{\text{forw}} = \sum_{l=1}^{L_{\max}} \mathcal{D}\left(\frac{\bar{\boldsymbol{Z}}^l_{b,i}}{\|\bar{\boldsymbol{Z}}^l_{b,i}\|_2}, \boldsymbol{P}^l\right)^2, \tag{4}$$

First-token deduplication Recovering the first token of each sequence correctly is crucial, as all subsequent token representations depend on it. However, first-token embeddings are independent across batch elements, so the forward span-distance loss ℓ_{forw} has multiple global minima corresponding to the ground-truth first-token embeddings $\boldsymbol{Z}^0_{:,1}$. To recover as many as possible, we optimize the B client sequences one after another and add a deduplication loss ℓ_{dedup} that encourages the optimized first-token embedding $\widehat{\boldsymbol{Z}}^0_{b,1}$ to differ from the previously recovered ones $\widehat{\boldsymbol{Z}}^0_{1:b-1,1}$. Concretely, for each layer $1 \le l \le L_{\max}$ we construct a projection matrix $\boldsymbol{P}^{l,b}_{\text{prev}}$ onto the part of the gradient-induced subspace $\mathcal{S}^l$ already occupied by the hidden states of the previously recovered first tokens (§B.2), and penalize the component of the current first-token hidden state along these directions:

$$\ell^b_{\text{dedup}} = \sum_{l=1}^{L_{\max}} \left\| \frac{\bar{\boldsymbol{Z}}^l_{b,1}}{\left\|\bar{\boldsymbol{Z}}^l_{b,1}\right\|_2} (\boldsymbol{P}^{l,b}_{\text{prev}})^\top \right\|_2^2. \tag{5}$$

For $b > 1$, the first-token objective is then $\ell^{(b,1)} = \ell^{(b,1)}_{\text{forw}} + \lambda_{\text{dedup}} \ell^b_{\text{dedup}}$, where λ_{dedup} controls the strength of the deduplication loss. All other steps optimize only the forward loss.

Termination condition As sequence lengths are unknown, we stop the attack once the increase in ℓ_{forw} from snapping the optimized embedding to the vocabulary exceeds a threshold τ_{snap}. We calibrate τ_{snap} on public data and evaluate it in §E.4. The full algorithm can be found in §B.

4.3 Bidirectional Attention Attack

Bidirectional attention couples all tokens within a sequence or image block, making joint recovery necessary as the hidden representations become interdependent. Thus, in this setting we optimize the positions Ω while holding the known embeddings at $\mathcal{K}$ fixed. For encoders, Ω contains the entire batch and $\mathcal{K} = \emptyset$, whereas for images, it contains the image block and $\mathcal{K}$ contains the recovered prefix. A joint forward pass computes both sets' hidden states and evaluates ℓ_{forw} only at Ω.

Backward span-distance loss The forward loss encourages each hidden state to lie in the observed subspace, but does not require the recovered states to cover it entirely. Repeated tokens or redundant sequences can therefore achieve a low forward loss without meaningful information recoevered. To discourage this collapse, we additionally introduce a backward span-distance loss that penalizes the distance from each observed basis direction to the recovered subspace:

$$\ell_{\text{back}} = \sum_{l=1}^{L_{\max}} \sum_{i=1}^{T} \mathcal{D}\left(\boldsymbol{U}^l_i, \bar{\boldsymbol{P}}^l\right)^2, \tag{6}$$

where $\bar{\boldsymbol{P}}^l$ is a differentiable approximation of the orthogonal projection onto the span of $\bar{\boldsymbol{Z}}^l_{\Omega \cup \mathcal{K}}$ (§B.5). In the case of images, we are left with unsimulated tokens, such as the text following an



Figure 2: Example TIGER progress on a 256-token image across multiple stages.

image, which leave the recovered subspace incomplete. That said, we show in §G that the loss is well-behaved despite the missing tokens, and the true partial hidden states still minimize ℓ_{back}. We thus optimize the combination of the 2 objectives $\ell = \ell_{\text{forw}} + \lambda_{\text{back}}\,\ell_{\text{back}}$.

Image regularization For images, the patching performed by f^{mm} introduces discontinuities at the patch boundaries. To mitigate this, we add a total-variation prior $\lambda_{\text{TV}}\,\ell_{\text{TV}}$ to ℓ, where $\ell_{\text{TV}} = \ell_{\text{TV,in}} + \ell_{\text{TV},\partial}$. The within-patch term $\ell_{\text{TV,in}}$ encourages local smoothness inside each recovered patch and primarily improves image consistency, while the border term $\ell_{\text{TV},\partial}$ aligns adjacent patches across their shared boundary. The border term uses jointly optimized per-patch scales to handle scale ambiguity, and our ablations show it removes patchiness at almost no token-accuracy cost (§B.6).

Interleaved continuous and discrete optimization Continuous optimization often recovers tokens at incorrect positions, as different orderings can induce similar spans (Balunovic et al., 2022). We therefore alternate continuous optimization with discrete reordering over $N_{\text{S}} = 10$ stages. Each stage retains the best of the N_{init} candidates, then improves its ordering by swapping patches for images or shifting text subsequences, accepting moves that reduce ℓ_{forw} and ℓ_{back}, respectively. The reordered candidate is used to initialize the next stage, with high-loss positions selectively resampled to escape incorrect reconstructions. We preserve one candidate unchanged and resample progressively more positions with the largest contributions to ℓ_{forw} in the others, drawing new embeddings or pixel patches from the original priors. Figure 2 illustrates this process qualitatively, and we can see how reordering recovered the correct structure, and that multiple stages are necessary to get a near-exact reconstruction. We describe the process in detail in §B.3.

5 Experimental Results

We now present the setup and results of our experimental evaluation of TIGER.

5.1 Experimental Setup

First, we summarize the evaluation protocol. Implementation details regarding dataset construction, metric definitions, attack hyperparameters, and hardware details is further provided in §C.

Models and tasks We evaluate TIGER on two causal LLMs, Gemma-3-4B-IT (Gemma Team, 2025), and Llama-3.1-8B (Team, 2024); the multimodal Gemma-4-12B (Team, 2026); and two text encoders, EmbeddingGemma-300M (Schechter Vera et al., 2025) and BERT (Devlin et al., 2018). Causal LLMs use next-token prediction, with the multimodal loss restricted to answer tokens in a QA setting, following visual instruction tuning (Liu et al., 2023). For encoders, we instead use sequence classification with randomly assigned binary labels.

Datasets For text-only tasks, we use SST-2 (Socher et al., 2013) and Rotten Tomatoes (RT) (Pang & Lee, 2005), along with PII-masking-300k (Ai4Privacy, 2024) to assess recovery of fictitious personally identifiable information (PII), using 10 batches per experiment. The multimodal evaluation uses 20 images from VPD-100K (Hu et al., 2026), including 10 documents and 10 scenery images containing visible PII, which we augmented with synthetic QA pairs surrounding each image.

Metrics We report ROUGE-1 and ROUGE-2 F1 (Lin, 2004), denoted R-1 and R-2. We average them after matching recovered to ground-truth sequences by maximizing the overall R-1, and also report PII recovery accuracy where available. For images, we compute PSNR and SSIM after aligning each reconstructed image's scale and shift to the original, accounting for information lost from the LayerNorm. We also measure the image-token accuracy as the fraction of reconstructions with cosine

Table 1: TIGER and the baselines on the encoder models.

			B=1		B=2		B=4		B=8		B=16		B=32	
			R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
EMB. GEMMA	SST-2	TIGER	100.0	100.0	100.0	100.0	99.6	99.3	99.1	98.0	99.0	97.6	60.8	42.2
	Rotten Tomatoes	TIGER	100.0	100.0	100.0	100.0	99.8	99.3	98.3	96.8	40.2	16.7	–	–
BERT	SST-2	LAMP	88.8	56.8	82.4	45.7	69.5	32.5	56.9	19.1	–	–	–	–
		DAGER	**100.0**	**100.0**	99.3	99.0	95.6	93.0	74.1	59.8	–	–	–	–
		TIGER	**100.0**	**100.0**	**100.0**	**100.0**	**99.5**	**95.7**	**95.2**	**88.5**	–	–	–	–
	Rotten Tomatoes	LAMP	64.7	16.5	46.4	7.6	35.1	4.2	27.3	2.0	–	–	–	–
		DAGER	**100.0**	**100.0**	**98.1**	**96.5**	66.8	50.1	37.1	11.4	–	–	–	–
		TIGER	**100.0**	**100.0**	96.0	93.1	**96.5**	**90.7**	**60.5**	**35.2**	–	–	–	–

Table 2: Multimodal reconstruction of TIGER under different image-token budgets N_T.

N_T	Question		Image							Answer	
	R-1	R-2	Tok. acc.	Tok. acc. doc.	Tok. acc. scene	PSNR	SSIM	PII doc.	PII face	R-1	R-2
70	96.5	95.9	95.1	93.0	97.2	20.47	0.730	73.8	90.8	95.7	92.1
140	100.0	100.0	93.3	91.8	94.9	19.84	0.729	86.0	88.9	93.0	88.1
280	100.0	100.0	82.1	68.3	96.0	19.11	0.746	66.2	96.7	90.2	86.6

similarity ≥ 0.90 to the ground truth. To assess visual PII leakage, we use GEMINI-3.1-PRO (Google DeepMind, 2026) to compare annotated private regions in the original and reconstructed images, using character-wise text similarity for documents and no/partial/full recovery scores for faces.

Baseline attacks We compare against LAMP (Balunovic et al., 2022), which combines gradient matching with a language-model prior, and DAGER (Petrov et al., 2024), which filters vocabulary candidates using gradient subspaces. For encoder models, we compare against the papers' published BERT results. However, both remain infeasible for EMBEDDINGGEMMA-300M, as we were unable to obtain non-zero recovery for LAMP, and because DAGER recovers only unordered tokens under RoPE, making sequence enumeration infeasible. For decoder models, we compare on GEMMA-3-4B-IT against DAGER, but discard LAMP, which assumes known labels that reveal the input under next-token prediction. Finally, neither supports multimodal recovery, as LAMP requires answer labels and its prior operates on text, while DAGER requires a discrete vocabulary, inapplicable for continuous tokens.

5.2 MAIN RESULTS

We now present the main reconstruction results for TIGER. Tables 1 and 3 compare TIGER against LAMP and DAGER on BERT and GEMMA-3-4B-IT across batch sizes on SST-2 and Rotten Tomatoes.

Attacking encoder networks On encoder models, TIGER matches or outperforms DAGER across all batch sizes, and does not degrade significantly on larger batches. As shown in Table 1, on GEMMA, where DAGER is not feasible, we maintain near-perfect recovery through B=8 for both datasets. It drops sharply at $B = 32$ for SST-2 and $B = 16$ for Rotten Tomatoes, as the number of tokens hits roughly 400, more than half the hidden dimension of 768, but still recovers non-trivial information.

Attacking multimodal networks As GEMMA-4-12B admits different configurations, supporting a variety of image resolutions, Table 2 reports multimodal reconstruction on GEMMA-4-12B under image-token budgets of 70, 140, and 280, where each image to the largest resolution that fits within the budget before being patched. For 280 tokens, that would correspond to a $768 \times 768px$ image. We can clearly see that TIGER is able to achieve near-perfect reconstructions on average across all evaluated budgets, and can further consistently recover the surrounding text. We observe that document context is slightly more challenging to recover than the main image content, likely due to the images having more tokens with negligible information, but TIGER still captures significant portions of it. At the 280-token budget, reconstructions reach an average PSNR of 19.11 dB and SSIM of 0.746, which suffices to leak private information: faces in scenery images are recovered with a mean PII score of 96.7, and document text with a character-level similarity of 66.2%. We further ran

Table 3: Sequence reconstruction of TIGER and DAGER on causal LLMs.

			B=1		B=2		B=4		B=8	
			R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
GEMMA-3 4B	SST-2	DAGER	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.5**	**93.3**	**93.1**
		TIGER	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	95.0	89.2	86.5
	Rotten Tomatoes	DAGER	**98.1**	**98.0**	**94.6**	**93.3**	**96.1**	**95.4**	–	–
		TIGER	94.4	94.1	89.4	89.0	87.4	86.6	**50.4**	**45.6**
LLAMA-3.1	SST-2	TIGER	79.3	79.2	89.8	87.5	78.1	73.9	–	–

a single-image experiment at a 560 token budget, achieving a 97.7% token accuracy, nearly-perfectly recovering an image above 720p resolution. This example is provided in the supplementary materials.

Attacking causal networks Compared to DAGER, TIGER achieves comparable performance on the undefended setting across most evaluated batch sizes. In Table 3, we show that our attack generalizes across datasets and different model families, achieving strong reconstruction performance consistently. This also scales to recovering actual PII leakage, where we are able to, under $B = 4$, recovering 78.4% of the sensitive information under a comparable 72.7 R-2 score. Where TIGER is unable to fully recover the input, this is mostly due to being unable to correctly construct the first token to start the sequence, which can be alleviated by combining it with DAGER-style approaches.

5.3 TIGER UNDER PERTURBED GRADIENTS

In an unperturbed setting, we have seen that TIGER matches or outperforms prior gradient inversion attacks. However, realistic scenarios often involve more complicated or defended training setups. Here, we investigate the robustness of TIGER under such perturbed gradient conditions.

Attacking noise-based defenses Following (Balunovic et al., 2022), we evaluate reconstruction under additive Gaussian gradient noise of varying levels $\sigma \in [0, 10^{-3}]$. The 10^{-3} noise level is the highest considered, as it corresponds to the upper bound of noise that still preserves training utility, as shown in §E.5. Table 4 compares TIGER and DAGER on GEMMA-3-4B-IT across several batch sizes, where we can see that TIGER recovers information under the highest noise levels, while DAGER performance collapses to trivial recovery.

Table 4: TIGER results in the defended setting across different noise levels σ and batch sizes B.

B	Attack	$\sigma = 0$		$\sigma = 10^{-5}$		$\sigma = 5 \times 10^{-5}$		$\sigma = 10^{-4}$		$\sigma = 10^{-3}$	
		R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
1	DAGER	**100.0**	**100.0**	13.5	0.0	24.1	0.0	27.0	0.0	16.3	0.0
	TIGER	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**72.6**	**64.5**
2	DAGER	**100.0**	**100.0**	12.1	0.0	12.6	0.0	12.6	0.0	0.9	0.0
	TIGER	**100.0**	**100.0**	**96.2**	**95.0**	**50.0**	**50.0**	**49.6**	**49.6**	**19.2**	**10.0**
4	DAGER	**100.0**	**97.5**	5.8	0.0	2.5	0.0	2.5	0.0	2.5	**0.0**
	TIGER	**100.0**	95.0	**100.0**	**95.0**	**41.6**	**35.4**	**21.2**	**12.5**	**6.1**	**0.0**
8	DAGER	**93.3**	**93.1**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.0**
	TIGER	89.2	86.5	**84.4**	**80.7**	**38.3**	**21.9**	**15.0**	**4.2**	**8.4**	**0.0**

Attacking lower quantization Another way gradients can be perturbed is through quantization, where gradients are represented with reduced numerical precision. Under BF16 quantization on the $B = 4$ setting on the PII dataset, while DAGER collapses to a 0.0 R-2 score, TIGER achieves 74.0, matching its 72.7 score under FP32, underscoring its robustness under numerical imprecision.

6 CONCLUSION

We introduced TIGER, a continuous gradient inversion attack that exploits the low-rank structure of transformer gradients. By aligning the hidden representations of optimized embeddings with the gradient-induced subspaces, we design a unified framework that recovers text from both causal and bidirectional attention models and, for the first time, multimodal inputs. On causal models, TIGER remains accurate under quantization and DP noise, where prior attacks collapse, while for encoder LLMs it substantially improves reconstruction quality, runtime, and scalability to larger batches. Finally, TIGER recovers multimodal images at realistic resolutions that reveal sensitive PII. We discuss limitations and future work, including scaling beyond the hidden dimension and broader defense mechanisms, in §I. Overall, transformer gradients retain substantial input information, whether text or images, even under perturbation or quantization, underscoring the need for stronger defenses in deployed FL systems.

ETHICS STATEMENT

Our work focuses on the development and evaluation of gradient inversion attacks in a controlled research setting. As we demonstrate further vulnerabilites in Federated Learning for Large Language Models, our work naturally increases the privacy attack surface. Nevertheless, we believe sharing the capabilities of a potential adversary is important for raising awareness of the risks, and is crucial for developing viable defenses. We intend this research to be used as a tool for improving the security and privacy of machine learning systems, and not for malicious purposes.

As some of our evaluation involves PII (Personally Identifiable Information), we take special care to handle it responsibly. All PII used in our experiments is either synthetically generated, or is sourced from existing privacy-centered PII datasets. In this sense, we retain the same terms and conditions as the original sources, ensuring compliance with their usage policies.

REPRODUCIBILITY STATEMENT

Our work is fully reproducible, and attached the full code for reproducing all our results within the supplementary Python code, attached to our submission. For any LLM-as-a-judge experiment, we include the prompt descriptions in §J, and provide our synthetic dataset as part of the repository. All hyperparameters are listed in the paper, with some remaining design choices in the script files in the repository.

AI USE STATEMENT

Throughout this work, we make use of Large Language Models (LLMs) in various capacities. We use them to check our writing for grammatical mistakes and for clarity, and to generate documentation for replication. We use AI coding agents to assist in implementing the algorithms described in the paper and their evaluations, and to monitor remote resource usage. We also use LLMs in some parts of our mathematical derivations, for example in checking for notation consistency, simplifying mathematical expressions, and for helping us tailor the proofs in the paper towards a wider audience. We use LLMs to generate the PII bounding boxes for our image data, and we implement LLM-as-a-judge for evaluation of this component of our image-based experiments. We have not used AI systems for ideation or for generating the core scientific contributions of this work.

REFERENCES

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (eds.), *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 308–318. ACM, 2016. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145/2976749.2978318.

Ai4Privacy. pii-masking-300k (revision 86db63b), 2024. URL https://huggingface.co/datasets/ai4privacy/pii-masking-300k.

Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 17455–17466, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/91cff01af640a24e7f9f7a5ab407889f-Abstract.html.

Anthropic. System card: Claude opus 5, jul 2026. URL https://www-cdn.anthropic.com/c5fbac3f0b1280a933ebd26d3cb8bb9f5bdeaf48/Claude%20Opus%205%20System%20Card.pdf.

Alexander Bakarsky, Dimitar I. Dimitrov, Maximilian Baader, and Martin T. Vechev. SPEAR++: scaling gradient inversion via sparsely-used dictionary learning. *CoRR*, abs/2510.24200, 2025. doi: 10.48550/ARXIV.2510.24200. URL https://doi.org/10.48550/arXiv.2510.24200.

Mislav Balunovic, Dimitar I. Dimitrov, Nikola Jovanovic, and Martin T. Vechev. LAMP: extracting text from gradients with language model priors. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/32375260090404f907ceae19f3564a7e-Abstract-Conference.html.

Kallista A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (eds.), *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pp. 1175–1191. ACM, 2017. doi: 10.1145/3133956.3133982. URL https://doi.org/10.1145/3133956.3133982.

David Byrd and Antigoni Polychroniadou. Differentially private secure multi-party computation for federated learning in financial applications. In Tucker Balch (ed.), *ICAIF '20: The First ACM International Conference on AI in Finance, New York, NY, USA, October 15-16, 2020*, pp. 16:1–16:9. ACM, 2020. doi: 10.1145/3383455.3422562. URL https://doi.org/10.1145/3383455.3422562.

Chaochao Chen, Xiaohua Feng, Yuyuan Li, Lingjuan Lyu, Jun Zhou, Xiaolin Zheng, and Jianwei Yin. Integration of large language models and federated learning. *Patterns*, 5(12):101098, 2024. doi: 10.1016/J.PATTER.2024.101098. URL https://doi.org/10.1016/j.patter.2024.101098.

Hong-Min Chu, Jonas Geiping, Liam H. Fowl, Micah Goldblum, and Tom Goldstein. Panning for gold in federated learning: Targeted text extraction under arbitrarily large-scale aggregation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=A9WQaxYsfx.

Michel Coste. *An Introduction to O-minimal Geometry*. Dottorato di Ricerca in Matematica. Istituti Editoriali e Poligrafici Internazionali, Pisa, Italy, 2000. ISBN 8881472260.

Jieren Deng, Yijue Wang, Ji Li, Chenghong Wang, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. TAG: gradient attack on transformer-based language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, Findings of ACL, pp. 3600–3610. Association for Computational Linguistics,

2021. doi: 10.18653/V1/2021.FINDINGS-EMNLP.305. URL https://doi.org/10.18653/v1/2021.findings-emnlp.305.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Dimitar I. Dimitrov, Maximilian Baader, Mark Niklas Müller, and Martin T. Vechev. SPEAR: exact gradient inversion of batches in federated learning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/c13cd7feab4beb1a27981e19e2455916-Abstract-Conference.html.

Maria Drencheva, Ivo Petrov, Maximilian Baader, Dimitar Iliev Dimitrov, and Martin T. Vechev. GRAIN: exact graph reconstruction from gradients. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=7bAjVh3CG3.

Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. FATE-LLM: A industrial grade federated learning framework for large language models. *CoRR*, abs/2310.10049, 2023. doi: 10.48550/ARXIV.2310.10049. URL https://doi.org/10.48550/arXiv.2310.10049.

Liam H. Fowl, Jonas Geiping, Wojciech Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=fwzUgo0FM9v.

Liam H. Fowl, Jonas Geiping, Steven Reich, Yuxin Wen, Wojciech Czaja, Micah Goldblum, and Tom Goldstein. Decepticons: Corrupted transformers breach privacy in federated learning for language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=r0BrY4BiEXO.

Yan Gao, Massimo Roberto Scamarcia, Javier Fernández-Marqués, Mohammad Naseri, Chong Shen Ng, Dimitris Stripelis, Zexi Li, Tao Shen, Jiamu Bai, Daoyuan Chen, Zikai Zhang, Rui Hu, Inseo Song, KangYoon Lee, Hong Jia, Ting Dang, Junyan Wang, Zheyuan Liu, Daniel J. Beutel, Lingjuan Lyu, and Nicholas D. Lane. Flowertune: A cross-domain benchmark for federated fine-tuning of large language models. *CoRR*, abs/2506.02961, 2025. doi: 10.48550/ARXIV.2506.02961. URL https://doi.org/10.48550/arXiv.2506.02961.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c4ede56bbd98819ae6112b20ac6bf145-Abstract.html.

Gemma Team. Gemma 3. 2025. URL https://arxiv.org/abs/2503.19786.

Jiahui Geng, Yongli Mou, Feifei Li, Qing Li, Oya Beyan, Stefan Decker, and Chunming Rong. Towards general deep leakage in federated learning. *CoRR*, abs/2110.09074, 2021. URL https://arxiv.org/abs/2110.09074.

Robin C. Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *CoRR*, abs/1712.07557, 2017. URL http://arxiv.org/abs/1712.07557.

Google DeepMind. Gemini 3.1 pro model card, feb 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/.

Google Research and Google DeepMind. Medgemma technical report. *CoRR*, abs/2507.05201, 2025. doi: 10.48550/ARXIV.2507.05201. URL https://doi.org/10.48550/arXiv.2507.05201.

Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen. Recovering private text in federated learning of language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/35b5c175e139bff5f22a5361270fce87-Abstract-Conference.html.

Ali Hatamizadeh, Hongxu Yin, Holger Roth, Wenqi Li, Jan Kautz, Daguang Xu, and Pavlo Molchanov. Gradvit: Gradient inversion of vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10011–10020. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00978. URL https://doi.org/10.1109/CVPR52688.2022.00978.

Xiaobin Hu, Enpu Zuo, Lanping Hu, Kaiwen Yang, Dianshu Liao, Tianyi Zhang, Bo Yin, Yinsi Zhou, Shidong Pan, and Xiaoyu Sun. VPD-100K: towards generalizable and fine-grained visual privacy protection. *CoRR*, abs/2605.10229, 2026. doi: 10.48550/ARXIV.2605.10229. URL https://doi.org/10.48550/arXiv.2605.10229.

Donghao Huang, Tomas Drietomsky, Benjamin Barrett, and Zhaoxia Wang. How small can you go? lora fine-tuning 270m-8b models for merchant information extraction in financial transactions, 2026. URL https://arxiv.org/abs/2606.08051.

Jinwoo Jeon, Jaechang Kim, Kangwook Lee, Sewoong Oh, and Jungseul Ok. Gradient inversion with generative image prior. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 29898–29908, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/fa84632d742f2729dc32ce8cb5d49733-Abstract.html.

Wenhao Jiang, Yuchuan Luo, Guilin Deng, Silong Chen, Xu Yang, Shihong Wu, Xinwen Gao, Lin Liu, and Shaojing Fu. Federated large language models: Feasibility, robustness, security and future directions. *CoRR*, abs/2505.08830, 2025. doi: 10.48550/ARXIV.2505.08830. URL https://doi.org/10.48550/arXiv.2505.08830.

Sanjay Kariyappa, Chuan Guo, Kiwan Maeng, Wenjie Xiong, G. Edward Suh, Moinuddin K. Qureshi, and Hsien-Hsin S. Lee. Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pp. 15884–15899. PMLR, 2023. URL https://proceedings.mlr.press/v202/kariyappa23a.html.

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *CoRR*, abs/1610.05492, 2016. URL http://arxiv.org/abs/1610.05492.

Anran Li, Yuanyuan Chen, Wenjun Long, Yu Yin, Yan Hu, Hyunjae Kim, Weipeng Zhou, Yujia Zhou, Hongyi Peng, Yang Ren, Xuguang Ai, Zhenyue Qin, Ming Hu, Xiaoxiao Li, Han Yu, Yih-Chung Tham, Lucila Ohno-Machado, Hua Xu, and Qingyu Chen. A federated and parameter-efficient framework for large language model training in medicine. *CoRR*, abs/2601.22124, 2026. doi: 10.48550/ARXIV.2601.22124. URL https://doi.org/10.48550/arXiv.2601.22124.

Jianwei Li, Sheng Liu, and Qi Lei. Beyond gradient and priors in privacy attacks: Leveraging pooler layer inputs of language models in federated learning. *CoRR*, abs/2312.05720, 2023. doi: 10.48550/ARXIV.2312.05720. URL https://doi.org/10.48550/arXiv.2312.05720.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SkhQHMW0W.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html.

Jiahao Lu, Xi Sheryl Zhang, Tianli Zhao, Xiangyu He, and Jian Cheng. APRIL: finding the achilles' heel on privacy for vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10041–10050. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00981. URL https://doi.org/10.1109/CVPR52688.2022.00981.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Andre Manoel, Mirian del Carmen Hipolito Garcia, Tal Baumel, Shize Su, Jialei Chen, Robert Sim, Dan Miller, Danny Karmon, and Dimitrios Dimitriadis. Federated multilingual models for medical transcript analysis. In Bobak J. Mortazavi, Tasmie Sarker, Andrew Beam, and Joyce C. Ho (eds.), *Conference on Health, Inference, and Learning, CHIL 2023, Broad Institute of MIT and Harvard (Merkin Building), 415 Main Street, Cambridge, MA, USA*, Proceedings of Machine Learning Research, pp. 147–162. PMLR, 2023. URL https://proceedings.mlr.press/v209/manoel23a.html.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, Proceedings of Machine Learning Research, pp. 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Kevin Knight, Hwee Tou Ng, and Kemal Oflazer (eds.), *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pp. 115–124. The Association for Computer Linguistics, 2005. doi: 10.3115/1219840.1219855. URL https://aclanthology.org/P05-1015/.

Ivo Petrov, Dimitar I. Dimitrov, Maximilian Baader, Mark Niklas Müller, and Martin T. Vechev. DAGER: exact gradient inversion for large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/9ff1577a1f8308df1ccea6b4f64a103f-Abstract-Conference.html.

Adam Sadilek, Luyang Liu, Dung Nguyen, Methun Kamruzzaman, Stylianos Serghiou, Benjamin Rader, Alex Ingerman, Stefan Mellem, Peter Kairouz, Elaine O. Nsoesie, Jamie Macfarlane, Anil Vullikanti, Madhav V. Marathe, Paul Eastham, John S. Brownstein, Blaise Agüera y Arcas, Michael D. Howell, and John Hernandez. Privacy-first health research with federated learning. *npj*

Digit. Medicine, 4, 2021. doi: 10.1038/S41746-021-00489-2. URL https://doi.org/10.1038/s41746-021-00489-2.

Hasin Us Sami, Swapneel Sen, and Basak Guler. Minegrad: Gradient inversion attacks on loRA fine-tuning. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL https://openreview.net/forum?id=dD9XOZUpNc.

Henrique Schechter Vera, Sahil Dua, and EmbeddingGemma Team. Embeddinggemma: Powerful and lightweight text representations. 2025. URL https://arxiv.org/abs/2509.20354.

Andrew Sellergren, Chufan Gao, Fereshteh Mahvar, Timo Kohlberger, Fayaz Jamil, Madeleine Traverse, Alberto Tono, Bashir Sadjad, Lin Yang, Charles Lau, Liron Yatziv, Tiffany L. Chen, Bram Sterling, Kenneth Philbrick, Richa Tiwari, Yun Liu, Madhuram Jajoo, Chandrashekar Sankarapu, Swapnil Vispute, Harshad Purandare, Abhishek Bijay Mishra, Samuel Schmidgall, Tao Tu, Anil Palepu, Chunjong Park, Tim Strother, Rahul Thapa, Yong Cheng, Preeti Singh, Kat Black, Yossi Matias, Katherine Chou, Avinatan Hassidim, Kavi Goel, Joelle K. Barral, Tris Warkentin, Shravya Shetty, Dale R. Webster, Sunny Virmani, David F. Steiner, Can Kirmizibayrak, and Daniel Golden. Medgemma 1.5 technical report. *CoRR*, abs/2604.05081, 2026. doi: 10.48550/ARXIV.2604.05081. URL https://doi.org/10.48550/arXiv.2604.05081.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1631–1642. ACL, 2013. doi: 10.18653/V1/D13-1170. URL https://doi.org/10.18653/v1/d13-1170.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Yuxin Wen, Jonas Geiping, Liam Fowl, Micah Goldblum, and Tom Goldstein. Fishing for user data in large-batch federated learning via gradient magnification. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pp. 23668–23684. PMLR, 2022. URL https://proceedings.mlr.press/v162/wen22a.html.

Alex Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted pfaffian functions and the exponential function. *Journal of the American Mathematical Society*, 9(4):1051–1094, 1996.

Tianyu Xia, Shuheng Shen, Su Yao, Xinyi Fu, Ke Xu, Xiaolong Xu, and Xing Fu. Differentially private learning with per-sample adaptive clipping. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 10444–10452. AAAI Press, 2023. doi: 10.1609/AAAI.V37I9.26242. URL https://doi.org/10.1609/aaai.v37i9.26242.

Yuhang Yao, Jianyi Zhang, Junda Wu, Chengkai Huang, Yu Xia, Tong Yu, Ruiyi Zhang, Sungchul Kim, Ryan A. Rossi, Ang Li, Lina Yao, Julian J. McAuley, Yiran Chen, and Carlee Joe-Wong. Federated large language models: Current progress and future directions. *CoRR*, abs/2409.15723, 2024. doi: 10.48550/ARXIV.2409.15723. URL https://doi.org/10.48550/arXiv.2409.15723.

Ke Ye and Lek-Heng Lim. Schubert varieties and distances between subspaces of different dimensions. *SIAM J. Matrix Anal. Appl.*, 37(3):1176–1197, 2016. doi: 10.1137/15M1054201. URL https://doi.org/10.1137/15M1054201.

Hongxu Yin, Arun Mallya, Arash Vahdat, José M. Álvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *IEEE Conference on*

Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021, pp. 16337–16346. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01607. URL https://openaccess.thecvf.com/content/CVPR2021/html/Yin_See_Through_Gradients_Image_Batch_Recovery_via_GradInversion_CVPR_2021_paper.html.

Linan Yue, Qi Liu, Yichao Du, Weibo Gao, Ye Liu, and Fangzhou Yao. Fedjudge: Federated legal large language model. In Makoto Onizuka, Jae-Gil Lee, Yongxin Tong, Chuan Xiao, Yoshiharu Ishikawa, Sihem Amer-Yahia, H. V. Jagadish, and Kejing Lu (eds.), *Database Systems for Advanced Applications - 29th International Conference, DASFAA 2024, Gifu, Japan, July 2-5, 2024, Proceedings, Part V*, Lecture Notes in Computer Science, pp. 268–285. Springer, 2024. doi: 10.1007/978-981-97-5569-1_17. URL https://doi.org/10.1007/978-981-97-5569-1_17.

Konstantin Zaitsev. An extended annotation scheme for personal-fact classification in dialogue, 2026. Model: https://huggingface.co/adugeen/personal-facts-classifier-embeddinggemma-300m; Dataset: https://huggingface.co/datasets/adugeen/personal-facts-msc.

Lihong Zhang and Yue Li. Federated learning with layer skipping: Efficient training of large language models for healthcare NLP. *CoRR*, abs/2504.10536, 2025. doi: 10.48550/ARXIV.2504.10536. URL https://doi.org/10.48550/arXiv.2504.10536.

Zhuo Zhang, Xiangjing Hu, Jingyuan Zhang, Yating Zhang, Hui Wang, Lizhen Qu, and Zenglin Xu. FEDLEGAL: the first real-world federated learning benchmark for legal NLP. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 3492–3507. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.193. URL https://doi.org/10.18653/v1/2023.acl-long.193.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *CoRR*, abs/2001.02610, 2020. URL http://arxiv.org/abs/2001.02610.

Junyi Zhu and Matthew B. Blaschko. R-GAP: recursive gradient attack on privacy. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=RSU17UoKfJF.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 14747–14756, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html.

A NOTATION

Below, in Table 5, we summarize the notation used throughout the appendix for the reader's convenience.

Symbol	Definition	Symbol	Definition
B	Batch size	η	TIGER's learning rate
$\mathcal{V}$	Vocabulary set	Ω	Optimized token positions
V	Vocabulary size, $V = \lvert\mathcal{V}\rvert$	$\mathcal{K}$	Known token positions whose embeddings are fixed during optimization
$\boldsymbol{E}$	Embedding vectors of the tokens in $\mathcal{V}$	$\mathcal{R}$	Token positions that are not simulated
e_v	The embedding vector associated with token $v \in \mathcal{V}$	$\bar{\boldsymbol{Z}}^0$	Continuous raw input embeddings optimized by TIGER
T_b	Token length for the b-th sequence	$\bar{\boldsymbol{Z}}^l$	Recovered hidden states obtained from $\bar{\boldsymbol{Z}}^0$, $\bar{\boldsymbol{Z}}^1 = f^0(\bar{\boldsymbol{Z}}^0; \boldsymbol{M})$, $\bar{\boldsymbol{Z}}^{l+1} = f^l(\bar{\boldsymbol{Z}}^l; \boldsymbol{M})$
$T_{\max}$	Length of the longest sequence, $T_{\max} = \max_b T_b$	$\bar{\boldsymbol{Z}}^l_{b,i}$	Recovered hidden state at layer l corresponding to i-th token in the b-th sequence
T	Total number of non-padding tokens, $T = \sum_{b=1}^{B} T_b$	$\boldsymbol{\mu}_i$	Mean of the Gaussian prior for tokens at position i
$t_{b,i}$	Token for i-th position of the b-th client sequence	$\boldsymbol{\Sigma}_i$	Covariance of the Gaussian prior for tokens at position i
$\hat{t}_{b,i}$	Recovered token for i-th position of the b-th client sequence	$\mathcal{S}^l$	Gradient subspace at layer l, $\mathcal{S}^l = \mathrm{colspan}\left(\partial\mathcal{L}/\partial \boldsymbol{W}_Q^l\right)$
L	Number of transformer blocks	$\boldsymbol{U}^l$	Orthonormal basis of $\mathcal{S}^l$
d	Transformer hidden dimension	$\boldsymbol{P}^l$	Projection matrix onto $\mathcal{S}^l$
$\boldsymbol{Z}^0$	Raw transformer input embeddings	$\bar{\mathcal{S}}^l$	Recovered activation subspace at layer l, $\bar{\mathcal{S}}^l = \mathrm{rowspan}(\bar{\boldsymbol{Z}}^l_{\Omega\cup\mathcal{K}})$
$\boldsymbol{Z}^l$	Hidden states input to the l-th attention layer	$\bar{\boldsymbol{P}}^l$	Projection matrix onto $\bar{\mathcal{S}}^l$
$\boldsymbol{M}$	Attention mask	$\boldsymbol{P}^{l,b}_{\text{prev}}$	Projection matrix onto the span of the previously recovered first-token hidden states $\widehat{\boldsymbol{Z}}^l_{1:b-1,1}$ within $\mathcal{S}^l$
f^0	Preprocessing done on the raw embeddings before feeding them into the first attention block	$\mathcal{D}(z, \boldsymbol{P})$	Distance from the vector z to its projection using $\boldsymbol{P}$
f^l	l-th transformer block, i.e., $\boldsymbol{Z}^{l+1} = f^l(\boldsymbol{Z}^l; \boldsymbol{M})$	ℓ	TIGER's final loss
$\boldsymbol{W}^l_Q, \boldsymbol{W}^l_K, \boldsymbol{W}^l_V$	Query/Key/Value projection matrices for the l-th attention layer	ℓ_{forw}	Forward span-distance loss used by TIGER
$\boldsymbol{Q}^l, \boldsymbol{K}^l, \boldsymbol{V}^l$	The Query/Key/Value activations in the l-th attention layer	ℓ_{back}	Backward span-distance loss used by TIGER
$\mathcal{L}$	Training loss function used by clients	ℓ_{dedup}	Loss term penalizing repeating tokens during decoder first-token recovery
f^{mm}	Known encoder mapping $\boldsymbol{X}^{\text{mm}}$ to a contiguous block of rows of $\boldsymbol{Z}^0$	ℓ_{TV}	Total-variation prior on recovered images
$\boldsymbol{X}^{\text{mm}}$	Raw multimodal input matrix, e.g., an image, given to f^{mm}	$\ell_{\text{TV,in}}$	Within-patch component of ℓ_{TV}
$\boldsymbol{X}^{\text{mm}}_i$	Patch of $\boldsymbol{X}^{\text{mm}}$ belonging to image token i	$\ell_{\text{TV},\partial}$	Patch-border component of ℓ_{TV}
$\mathcal{E}$	Pairs of spatially adjacent image patches	λ_{dedup}	Weight of the ℓ_{dedup} loss term used during first-token recovery for decoder models
N_{init}	Number of random initializations used in TIGER	λ_{back}	Weight of the ℓ_{back} loss term used under bidirectional attention
N_{steps}	TIGER's number of continuous optimization steps per stage	λ_{inv}	Small diagonal jitter used to stabilize matrix inversions in ℓ_{back}
N_{S}	Number of stages interleaving continuous optimization and discrete reordering under bidirectional attention	λ_{TV}	Weight of the ℓ_{TV} loss term used for images
$L_{\max}$	Number of transformer layers used by TIGER	α_i	Optimized scale of $\boldsymbol{X}^{\text{mm}}_i$ in $\ell_{\text{TV},\partial}$
$\partial\mathcal{L}/\partial\boldsymbol{W}^l_Q$	Gradient of $\mathcal{L}$ w.r.t. $\boldsymbol{W}^l_Q$, observed by the attacker	$\boldsymbol{U}^{l,b}_{\text{prev}}$	Rotated basis of $\mathcal{S}^l$ spanning the previously recovered first-token hidden states, $\boldsymbol{P}^{l,b}_{\text{prev}} = (\boldsymbol{U}^{l,b}_{\text{prev}})^\top \boldsymbol{U}^{l,b}_{\text{prev}}$
$\partial\mathcal{L}/\partial\boldsymbol{Q}^l$	Gradient of $\mathcal{L}$ w.r.t. $\boldsymbol{Q}^l$	τ_{snap}	Threshold on the increase of ℓ_{forw} under vocabulary snapping, used to terminate causal sequence recovery
N_{img}	Number of tokens spanned by an image block	v_{img}	Public vocabulary token that marks the start of an image block
N_{rounds}	Maximum number of greedy reordering rounds per stage	$\ell_{\text{forw},b,i}$	Contribution of position (b,i) to ℓ_{forw}

Table 5: Table of notations used in the technical description of TIGER.

B FURTHER DETAILS ON ATTACK ALGORITHMS

This section provides the full pseudocode of TIGER, together with implementation details omitted from §4.

Conventions All algorithms receive the transformer blocks $\{f^l\}_{l=1}^{L_{\max}-1}$, the preprocessing f^0, the subspace bases $\{\boldsymbol{U}^l\}_{l=1}^{L_{\max}}$ and projections $\{\boldsymbol{P}^l\}_{l=1}^{L_{\max}}$ extracted from the observed gradients, and the vocabulary embeddings $\{e_v\}_{v\in\mathcal{V}}$. We hence omit them from the input lists. We write $\{\bar{\boldsymbol{Z}}^l\}_{l=1}^{L_{\max}} \leftarrow \textsc{Forward}(\bar{\boldsymbol{Z}}^0; \boldsymbol{M})$ for the propagation $\bar{\boldsymbol{Z}}^1 = f^0(\bar{\boldsymbol{Z}}^0; \boldsymbol{M})$, $\bar{\boldsymbol{Z}}^{l+1} = f^l(\bar{\boldsymbol{Z}}^l; \boldsymbol{M})$ of §3.2, which only requires the first $L_{\max} - 1$ blocks. The per-position forward loss

$$\ell_{\text{forw},b,i} = \sum_{l=1}^{L_{\max}} \mathcal{D}\left(\frac{\bar{\boldsymbol{Z}}^l_{b,i}}{\|\bar{\boldsymbol{Z}}^l_{b,i}\|_2}, \boldsymbol{P}^l\right)^2 \tag{7}$$

is the contribution of position (b,i) to Equation (2), so that $\ell_{\text{forw}} = \sum_{(b,i)\in\Omega} \ell_{\text{forw},b,i}$, and $\ell^{(b,i)}_{\text{forw}}$ of Equation (4) is $\ell_{\text{forw},b,i}$ evaluated with the recovered prefix. Where the argument matters, we write $\ell_{\text{forw},b,i}(z)$ for its value when the embedding at position (b,i) is set to z.

B.1 CAUSAL RECOVERY

Algorithm 1, defining a single step of the causal mode of TIGER, recovers a single embedding at position (b, i) given the already recovered prefix $\widehat{\boldsymbol{Z}}^0_{b,1:i-1}$, which is kept as continuous embeddings rather than snapped to tokens. It draws N_{init} candidates from the position-dependent prior $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ (line 3), optimizes each for N_{steps} steps on $\ell_{\text{forw},b,i}$ (lines 4–9), adds the deduplication loss ℓ_{dedup} of Equation (5) whenever deduplication projections are supplied (lines 7–8), and returns the candidate with the lowest final loss.

Algorithm 1 RECOVERTOKEN: Continuous Recovery of One Causal Token

Input: Sequence index b, position i, recovered prefix $\widehat{\boldsymbol{Z}}^0_{b,1:i-1}$,
Attention mask $\boldsymbol{M}$, Gaussian prior $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$,
Deduplication projections $\{\boldsymbol{P}^{l,b}_{\text{prev}}\}_{l=1}^{L_{\max}}$ or $\emptyset$
Output: Recovered embedding $z^\star$ and its loss $\ell^\star$

1: $\ell^\star \leftarrow +\infty$
2: **for** $r = 1$ to N_{init} **do**
3: $\quad z \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
4: $\quad$ **for** $s = 1$ to N_{steps} **do**
5: $\quad\quad \bar{\boldsymbol{Z}}^0_{b,1:i} \leftarrow [\widehat{\boldsymbol{Z}}^0_{b,1:i-1}\,;\, z], \quad \{\bar{\boldsymbol{Z}}^l\}_{l=1}^{L_{\max}} \leftarrow \text{FORWARD}(\bar{\boldsymbol{Z}}^0; \boldsymbol{M})$
6: $\quad\quad \ell \leftarrow \ell_{\text{forw},b,i}$ (Equation (7))
7: $\quad\quad$ **if** $\{\boldsymbol{P}^{l,b}_{\text{prev}}\} \neq \emptyset$ **then**
8: $\quad\quad\quad \ell \leftarrow \ell + \lambda_{\text{dedup}} \sum_{l=1}^{L_{\max}} \left\| \frac{\bar{\boldsymbol{Z}}^l_{b,i}}{\|\bar{\boldsymbol{Z}}^l_{b,i}\|_2} (\boldsymbol{P}^{l,b}_{\text{prev}})^\top \right\|_2^2$
9: $\quad\quad z \leftarrow z - \eta \nabla_z \ell$
10: $\quad$ **if** $\ell < \ell^\star$ **then**
11: $\quad\quad \ell^\star \leftarrow \ell, \quad z^\star \leftarrow z$
12: **return** $(z^\star, \ell^\star)$

This function is then plugged into Algorithm 2, which recovers the batch sequence by sequence and each sequence token by token. For $b > 1$, the first token uses the deduplication projections of DEDUPPROJS (§B.2), which encourage it to differ from the previously recovered first tokens (line 2). As the sequence lengths are unknown, after each recovered embedding $z^\star$ we snap it to its nearest token v and compare the forward loss of the snapped embedding to that of $z^\star$ (lines 7–8). A small increase indicates that a vocabulary token explains the observed subspaces at this position, and we accept $z^\star$ (lines 9–10). An increase above τ_{snap} indicates that no token continues the sequence, and we declare the sequence finished at length $i - 1$ (line 12). We calibrate τ_{snap} on public data, which we set to 0.219 for our experiments, and analyze this criterion in §E.4.

Algorithm 2 DECODERRECOVERY: Sequential Recovery for Causal Models

Input: Batch size B, Causal attention mask $\boldsymbol{M}_{\text{dec}}$,
Per-position Gaussian priors $\{(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\}_{i \geq 1}$,
Termination threshold τ_{snap}
Output: Reconstructed token sequences $\{\hat{t}_{b,i}\}$ and lengths $\{T_b\}_{b=1}^B$

1: **for** $b = 1$ to B **do**
2: $\quad \{\boldsymbol{P}^{l,b}_{\text{prev}}\}_{l=1}^{L_{\max}} \leftarrow \text{DEDUPPROJS}\left(\{\boldsymbol{U}^l\}_{l=1}^{L_{\max}}, \widehat{\boldsymbol{Z}}^0_{1:b-1,1}, \boldsymbol{M}_{\text{dec}}\right)$ **if** $b > 1$ **else** $\emptyset$
3: $\quad i \leftarrow 1$
4: $\quad$ **repeat**
5: $\quad\quad \mathcal{D} \leftarrow \{\boldsymbol{P}^{l,b}_{\text{prev}}\}_{l=1}^{L_{\max}}$ **if** $i = 1$ **else** $\emptyset$
6: $\quad\quad (z^\star, \ell^\star) \leftarrow \text{RECOVERTOKEN}(b, i, \widehat{\boldsymbol{Z}}^0_{b,1:i-1}, \boldsymbol{M}_{\text{dec}}, (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \mathcal{D})$
7: $\quad\quad v \leftarrow \operatorname{argmax}_{v \in \mathcal{V}} \text{CosSim}(z^\star, e_v)$
8: $\quad\quad \delta \leftarrow \ell_{\text{forw},b,i}(e_v) - \ell_{\text{forw},b,i}(z^\star)$
9: $\quad\quad$ **if** $\delta \leq \tau_{\text{snap}}$ **then**
10: $\quad\quad\quad \widehat{\boldsymbol{Z}}^0_{b,i} \leftarrow z^\star, \quad i \leftarrow i + 1$
11: $\quad$ **until** $\delta > \tau_{\text{snap}}$
12: $\quad T_b \leftarrow i - 1$
13: **return** $\text{PROJECTTOVOCAB}(\widehat{\boldsymbol{Z}}^0, \{T_b\}_{b=1}^B), \{T_b\}_{b=1}^B$

B.2 Basis Rotation for First-Token Decoder Deduplication

In decoder-only models, the first token of each sequence is especially important, because every later token representation in that sequence causally depends on it. When attacking batches with $B > 1$, however, the span constraints for the first token can admit repeated solutions, as all correct first-token embeddings are equally valid global minima of the forward loss. To reduce this failure mode, DedupProjs constructs, for each layer l, a projection matrix onto the directions of the gradient-induced subspace $\mathcal{S}^l$ that are already explained by the $b-1$ previously recovered first tokens.

The algorithm first stacks the previously recovered first-token embeddings,

$$\widehat{\boldsymbol{Z}}^0_{\text{prev}} \leftarrow \left[\widehat{\boldsymbol{Z}}^0_{1,1}; \widehat{\boldsymbol{Z}}^0_{2,1}; \cdots ; \widehat{\boldsymbol{Z}}^0_{b-1,1}\right].$$

It then forwards these embeddings through the attacked transformer layers, obtaining $\widehat{\boldsymbol{Z}}^l_{\text{prev}}$ at each layer. Given an orthonormal basis $\boldsymbol{U}^l \in \mathbb{R}^{T\times d}$ of the gradient-induced subspace $\mathcal{S}^l$, the matrix

$$\boldsymbol{C}^l_{:,1:b-1} \leftarrow \boldsymbol{U}^l \left(\widehat{\boldsymbol{Z}}^l_{\text{prev}}\right)^\top$$

contains the coordinates of the previously recovered hidden states in the basis $\boldsymbol{U}^l$. Thus, $\boldsymbol{C}^l$ records how the projections of the previous first-token hidden states decompose along the basis directions of $\mathcal{S}^l$. Each of the first $b-1$ columns corresponds to one previously recovered first token, while each row corresponds to one basis direction of $\mathcal{S}^l$.

The SVD of $\boldsymbol{C}^l$ gives an orthogonal rotation $\boldsymbol{R}^l$ of the coordinates inside the gradient-induced subspace. We use this rotation to form a rotated basis

$$\boldsymbol{U}^l_{\text{prev}} \leftarrow \left(\boldsymbol{R}^l\right)^\top \boldsymbol{U}^l.$$

This full rotated basis still spans the same subspace $\mathcal{S}^l$, but its leading coordinates are aligned with the directions that explain the previously recovered first-token hidden states. For deduplication, we therefore do not use the full rotated basis. Instead, when recovering the first token of sequence b, we keep only the first $b-1$ rotated basis vectors:

$$\boldsymbol{U}^{l,b}_{\text{prev}} \leftarrow \left(\boldsymbol{U}^l_{\text{prev}}\right)_{1:b-1,:}.$$

The previous-token projector is then defined as

$$\boldsymbol{P}^{l,b}_{\text{prev}} \leftarrow \left(\boldsymbol{U}^{l,b}_{\text{prev}}\right)^\top \boldsymbol{U}^{l,b}_{\text{prev}}.$$

Thus, $\boldsymbol{P}^{l,b}_{\text{prev}}$ projects only onto the portion of the gradient-induced subspace $\mathcal{S}^l$ associated with the previously recovered first tokens, rather than onto the entire gradient-induced subspace $\mathcal{S}^l$.

The resulting projector $\boldsymbol{P}^{l,b}_{\text{prev}}$ is used in the decoder objective to penalize overlap between the new first-token hidden state and the directions already used by earlier first tokens:

$$\ell_{\text{dedup}} \leftarrow \sum_{l=1}^{L_{\max}} \left\| \frac{\bar{\boldsymbol{Z}}^l_{b,1}}{\left\|\bar{\boldsymbol{Z}}^l_{b,1}\right\|_2} (\boldsymbol{P}^{l,b}_{\text{prev}})^\top \right\|_2^2.$$

This encourages different batch elements to use distinct first-token directions, while the standard forward span loss still keeps the recovered representation inside the full gradient-induced subspace $\mathcal{S}^l$.

B.3 Bidirectional Recovery

Algorithm 4 implements TIGER for recovering positions Ω under bidirectional attention, with the embeddings at the known positions $\mathcal{K}$ held fixed. It covers both text encoders, where Ω is the whole batch and $\mathcal{K} = \emptyset$, and image blocks inside multimodal sequences, where $\mathcal{K}$ holds the recovered prefix. The flag IsImage is given to the algorithm to distinguish between the two cases for the purposes of correctly instantiating the reordering. The optimized variable θ is the set of raw embeddings $\{z_p\}_{p\in\Omega}$ for text, and the pixel array together with the per-patch scales, $\theta = (\boldsymbol{X}^{\text{mm}}, \{\alpha_i\})$, for images. The map $\text{Embed}(\theta)$ returns the corresponding rows of $\bar{\boldsymbol{Z}}^0$, namely z_p for text and the rows of $f^{\text{mm}}(\boldsymbol{X}^{\text{mm}})$ for images. We define the prior as π, which is a distribution over θ from which both the initial candidates and the resampled positions are drawn, and is instantiated according to the type of input (text or image), as described in §4.

Algorithm 3 DEDUPPROJS: Basis Rotation for Decoder First-Token Deduplication

Input: Transformer blocks $\{f^l\}_{l=0}^{L_{\max}-1}$,
Orthonormal subspace bases $\{\boldsymbol{U}^l\}_{l=1}^{L_{\max}}$,
Previously recovered first-token embeddings $\widehat{\boldsymbol{Z}}^0_{1:b-1,1}$,
Attention mask $\boldsymbol{M}$
Output: Previous-token projections $\{\boldsymbol{P}^{l,b}_{\text{prev}}\}_{l=1}^{L_{\max}}$

$\widehat{\boldsymbol{Z}}^0_{\text{prev}} \leftarrow \left[\widehat{\boldsymbol{Z}}^0_{1,1}; \widehat{\boldsymbol{Z}}^0_{2,1}; \cdots; \widehat{\boldsymbol{Z}}^0_{b-1,1}\right]$
for $l = 1$ to $L_{\max}$ **do**
  $\boldsymbol{C}^l \leftarrow \boldsymbol{0}_{T\times\max(T,b-1)}$
  $\widehat{\boldsymbol{Z}}^l_{\text{prev}} \leftarrow f^{l-1}\left(\widehat{\boldsymbol{Z}}^{l-1}_{\text{prev}}; \boldsymbol{M}\right)$
  $\boldsymbol{C}^l_{:,1:b-1} \leftarrow \boldsymbol{U}^l\left(\widehat{\boldsymbol{Z}}^l_{\text{prev}}\right)^\top$
  $\boldsymbol{R}^l, _, _ \leftarrow \text{SVD}\left(\boldsymbol{C}^l\right)$
  $\boldsymbol{U}^l_{\text{prev}} \leftarrow \left(\boldsymbol{R}^l\right)^\top \boldsymbol{U}^l$
  $\boldsymbol{U}^{l,b}_{\text{prev}} \leftarrow (\boldsymbol{U}^l_{\text{prev}})_{1:b-1,:}$
  $\boldsymbol{P}^{l,b}_{\text{prev}} \leftarrow \left(\boldsymbol{U}^{l,b}_{\text{prev}}\right)^\top \boldsymbol{U}^{l,b}_{\text{prev}}$
return $\{\boldsymbol{P}^{l,b}_{\text{prev}}\}_{l=1}^{L_{\max}}$

Continuous phase In the first stage, the N_{init} candidates are drawn from π (lines 3–4), which instantiates the priors of §4.1: for text, π draws each embedding z_p from $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$; for images, it draws pixels i.i.d. from $\mathcal{N}(0.5, \sigma^2)$, using $\sigma = 1.0$ for the first half of the candidates and $\sigma = 0.1$ for the second half. Each candidate is optimized for N_{steps} steps on $\ell = \ell_{\text{forw}} + \lambda_{\text{back}}\ell_{\text{back}}$, plus $\lambda_{\text{TV}}\ell_{\text{TV}}$ for images (lines 8–11), where ℓ_{forw} sums Equation (7) over Ω and ℓ_{back} uses the recovered-span projection of §B.5. The candidate with the lowest final loss is retained (lines 12–13).

Discrete reordering The retained candidate is then put through a reordering procedure. For text, its embeddings are first snapped to tokens, the tokens are reordered by Algorithm 8, and the embeddings of the reordered tokens are used to generate initializations for the next stage (lines 15–16). For images, the patches of $\boldsymbol{X}^{\text{mm}}$ are reordered by Algorithm 7 (line 18). After the last stage, we return the reordered tokens or pixels (line 19).

Selective restarts Every subsequent stage starts from the reordered candidate $\theta^\star$ rather than from fresh priors. We rank the optimized positions by their per-position forward loss $\ell_{\text{forw},p}$ at $\theta^\star$, and candidate $k \in \{0, \ldots, N_{\text{init}} - 1\}$ resamples the $\min(k, |\Omega|)$ positions with the largest loss from π, i.e., a new embedding z_p for text and new pixels for the corresponding patch for images (lines 5–6). Importantly, candidate $k = 0$ keeps the entire candidate, retaining the current reconstruction while allowing the least consistent positions to be replaced.

Algorithm 4 BIDIRECTIONALRECOVERY: Staged Joint Recovery under Bidirectional Attention

Input: Flag ISIMAGE, Attention mask $\boldsymbol{M}$,
Optimized positions Ω, Known positions $\mathcal{K}$ with embeddings $\widehat{\boldsymbol{Z}}^0_{\mathcal{K}}$,
Prior distribution π over θ,
Coefficients λ_{back}, λ_{TV}, Number of stages N_{S}
Output: Tokens $\{\hat{t}_p\}_{p\in\Omega}$ for text, or pixels $\boldsymbol{X}^{\text{mm}}$ for images

1: **for** $s = 1$ to N_{S} **do**
2: **for** $k = 0$ to $N_{\text{init}} - 1$ **do**
3: **if** $s = 1$ **then**
4: $\theta_k \sim \pi$
5: **else**
6: $\theta_k \leftarrow \theta^\star$ with its $\min(k, |\Omega|)$ positions of largest $\ell_{\text{forw},p}$ resampled from π
7: **for** $t = 1$ to N_{steps} **do**
8: $\bar{\boldsymbol{Z}}^0_{\Omega} \leftarrow \text{EMBED}(\theta_k), \quad \bar{\boldsymbol{Z}}^0_{\mathcal{K}} \leftarrow \widehat{\boldsymbol{Z}}^0_{\mathcal{K}}, \quad \{\bar{\boldsymbol{Z}}^l\}_{l=1}^{L_{\max}} \leftarrow \text{FORWARD}(\bar{\boldsymbol{Z}}^0; \boldsymbol{M})$
9: $\ell \leftarrow \sum_{p\in\Omega} \ell_{\text{forw},p} + \lambda_{\text{back}}\ell_{\text{back}}$ (Equations (6)–(8))
10: $\ell \leftarrow \ell + \lambda_{\text{TV}}\ell_{\text{TV}}$ **if** ISIMAGE (Equation (9))
11: $\theta_k \leftarrow \theta_k - \eta\nabla_{\theta_k}\ell$
12: **if** $k = 0$ or $\ell < \ell^\star$ **then**
13: $\ell^\star \leftarrow \ell, \quad \theta^\star \leftarrow \theta_k$
14: **if** $\neg$ISIMAGE **then**
15: $\{\hat{t}_p\} \leftarrow \text{PROJECTTOVOCAB}(\text{EMBED}(\theta^\star)), \quad \{\hat{t}_p\} \leftarrow \text{TEXTREORDER}(\{\hat{t}_p\}, \boldsymbol{M})$
16: $\theta^\star \leftarrow \{e_{\hat{t}_p}\}_{p\in\Omega}$
17: **else**
18: $\boldsymbol{X}^{\text{mm}} \leftarrow \text{IMAGEREORDER}(\boldsymbol{X}^{\text{mm}}, \Omega, \mathcal{K}, \widehat{\boldsymbol{Z}}^0_{\mathcal{K}}, \boldsymbol{M})$ where $\theta^\star = (\boldsymbol{X}^{\text{mm}}, \{\alpha_i\})$
19: **return** $\boldsymbol{X}^{\text{mm}}$ **if** ISIMAGE **else** $\{\hat{t}_p\}_{p\in\Omega}$

B.4 MULTIMODAL RECOVERY

Algorithm 5 recovers multimodal sequences in a causal model, where the text is under causal attention and each image block of N_{img} tokens is under bidirectional attention. The attacker knows N_{img} but not where the image lies in the sequence (§3.1). The attack proceeds exactly as Algorithm 2, recovering the text before the image token by token, including any chat-template tokens, which are ordinary vocabulary tokens. The location of the image is revealed by the recovery itself: image blocks are delimited by the public image-start token $v_{\text{img}} \in \mathcal{V}$, and once the causal recovery snaps to it (line 11), the next N_{img} positions are known to hold the image. These positions form Ω, the recovered prefix forms $\mathcal{K}$, and the attention mask $\boldsymbol{M}_{\text{mm}}$ is causal over the prefix and bidirectional within the image block, with the block attending to the prefix (line 12). The image is recovered by Algorithm 4 in image mode (line 13), its encoder embeddings are placed into the recovered prefix (line 14), and the causal recovery resumes after the block (line 15), so that the text following the image, including the answer, is recovered with the image held fixed. The termination condition of Algorithm 2 ends each sequence (lines 8–10 and 16–17).

Algorithm 5 MULTIMODALRECOVERY: Interleaved Causal and Bidirectional Recovery

Input: Batch size B, Image encoder f^{mm}, Number of image tokens N_{img}, Image-start token v_{img}, Per-position Gaussian priors $\{(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\}_{i\geq 1}$, Pixel prior π_{img}, Termination threshold τ_{snap}, Coefficients λ_{back}, λ_{TV}, Number of stages N_{S}
Output: Reconstructed token sequences $\{\hat{t}_{b,i}\}$, lengths $\{T_b\}_{b=1}^{B}$, and images $\{\boldsymbol{X}_b^{\text{mm}}\}$

for $b = 1$ to B **do**
  $\{\boldsymbol{P}_{\text{prev}}^{l,b}\}_{l=1}^{L_{\max}} \leftarrow$ DEDUPPROJS $\Big(\{\boldsymbol{U}^l\}_{l=1}^{L_{\max}}, \widehat{\boldsymbol{Z}}^0_{1:b-1,1}, \boldsymbol{M}_{\text{dec}}\Big)$ **if** $b > 1$ **else** $\emptyset$
  $i \leftarrow 1, \quad \boldsymbol{M} \leftarrow \boldsymbol{M}_{\text{dec}}$
  repeat
    $\mathcal{D} \leftarrow \{\boldsymbol{P}_{\text{prev}}^{l,b}\}_{l=1}^{L_{\max}}$ **if** $i = 1$ **else** $\emptyset$
    $(z^\star, \ell^\star) \leftarrow$ RECOVERTOKEN$(b, i, \widehat{\boldsymbol{Z}}^0_{b,1:i-1}, \boldsymbol{M}, (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \mathcal{D})$
    $v \leftarrow \operatorname{argmax}_{v\in\mathcal{V}} \text{CosSim}(z^\star, e_v)$
    $\delta \leftarrow \ell_{\text{forw},b,i}(e_v) - \ell_{\text{forw},b,i}(z^\star)$
    if $\delta \leq \tau_{\text{snap}}$ **then**
      $\widehat{\boldsymbol{Z}}^0_{b,i} \leftarrow z^\star, \quad i \leftarrow i + 1$
    if $\delta \leq \tau_{\text{snap}}$ and $v = v_{\text{img}}$ **then**
      $\mathcal{K} \leftarrow \{(b,1), \ldots, (b, i-1)\}, \quad \Omega \leftarrow \{(b,i), \ldots, (b, i + N_{\text{img}} - 1)\}, \quad \boldsymbol{M} \leftarrow \boldsymbol{M}_{\text{mm}}$
      $\boldsymbol{X}_b^{\text{mm}} \leftarrow$ BIDIRECTIONALRECOVERY(ISIMAGE $=$ true, $\boldsymbol{M}, \Omega, \mathcal{K}, \widehat{\boldsymbol{Z}}^0_{\mathcal{K}}, \pi_{\text{img}}, \lambda_{\text{back}}, \lambda_{\text{TV}}, N_{\text{S}}$)
      $\widehat{\boldsymbol{Z}}^0_{\Omega} \leftarrow f^{\text{mm}}(\boldsymbol{X}_b^{\text{mm}})$
      $i \leftarrow i + N_{\text{img}}$
  until $\delta > \tau_{\text{snap}}$
  $T_b \leftarrow i - 1$
return PROJECTTOVOCAB$(\widehat{\boldsymbol{Z}}^0, \{T_b\}_{b=1}^{B})$ restricted to text positions, $\{T_b\}_{b=1}^{B}$, $\{\boldsymbol{X}_b^{\text{mm}}\}$

B.5 RECOVERED-SPAN PROJECTION

Let $\bar{\boldsymbol{Z}}^l := \bar{\boldsymbol{Z}}^l_{\Omega\cup\mathcal{K}}$ collect the hidden states at both optimized and known positions. We compute the regularized projection used in Equation (6) as

$$\bar{\boldsymbol{P}}^l = (\bar{\boldsymbol{Z}}^l)^\top \left(\bar{\boldsymbol{Z}}^l(\bar{\boldsymbol{Z}}^l)^\top + \lambda_{\text{inv}} I\right)^{-1} \bar{\boldsymbol{Z}}^l. \tag{8}$$

The diagonal jitter $\lambda_{\text{inv}} > 0$ keeps the matrix invertible even when the recovered states become linearly dependent. Computing this expression directly avoids differentiating through an SVD, which significantly slowed down our attack. For positive jitter, it is an approximation to an orthogonal projection; as $\lambda_{\text{inv}} \to 0$, it converges to $(\bar{\boldsymbol{Z}}^l)^\dagger \bar{\boldsymbol{Z}}^l$. The analysis in §G uses this exact projection.

B.6 IMAGE TOTAL-VARIATION PRIOR

Let $\boldsymbol{X}_i^{\text{mm}}$ denote the image patch at token position i, and let $\mathcal{E}$ contain all pairs of spatially adjacent patches. We then create a total-variation loss over the image as follows:

$$\begin{aligned}
\ell_{\text{TV,in}} &= \sum_{i\in\Omega} \text{TV}\big(\boldsymbol{X}_i^{\text{mm}}\big), \\
\ell_{\text{TV},\partial} &= \sum_{(i,j)\in\mathcal{E}} \text{TV}_\partial\big(\alpha_i \boldsymbol{X}_i^{\text{mm}}, \alpha_j \boldsymbol{X}_j^{\text{mm}}\big), \\
\ell_{\text{TV}} &= \ell_{\text{TV,in}} + \ell_{\text{TV},\partial}.
\end{aligned} \tag{9}$$

Here the total ℓ_{TV} is composed of a within-patch $\ell_{\text{TV,in}}$ component, as well as a patch-border $\ell_{\text{TV},\partial}$ component. The within-patch term sums absolute differences between horizontally and vertically adjacent pixels within each patch and primarily improves local image consistency, improving the visual quality of individual tokens. The border term $\ell_{\text{TV},\partial}$, instead, applies the total-variation prior at the borders of patches, aiming to establish a consistent colour and contrast across the image. As the LayerNorm applied directly to the image removes information about the scale and mean of the image, we optimize per-patch scales α_i jointly with $\boldsymbol{X}^{\text{mm}}$ and apply only to $\ell_{\text{TV},\partial}$, allowing neighboring patches to agree up to their individual scales (§4.1).

Quantitative and qualitative ablations of the image prior, together with the corresponding backward-loss ablation, are collected in §E. In short, the results confirm the effectiveness of the within-patch TV to improve reconstruction quality, while the border TV primarily improves the visualisation at negligible cost to the overall token accuracy.

B.7 UTILITY ROUTINES

Here, we instantiate the remaining routines called by the recovery algorithms above. Algorithm 6 maps recovered continuous embeddings to their nearest vocabulary tokens by cosine similarity, as used by §4.1. Algorithms 7 and 8 are the discrete reordering steps of Algorithm 4.

Algorithm 6 PROJECTTOVOCAB: Cosine Projection onto Token Embeddings

Input: Recovered continuous embeddings $\widehat{\boldsymbol{Z}}^0$,
Sequence lengths $\{T_b\}_{b=1}^B$
Output: Reconstructed token sequences $\{\hat{t}_{b,i}\}$
for $b = 1$ to B **do**
  for $i = 1$ to T_b **do**
    $\hat{t}_{b,i} \leftarrow \operatorname{argmax}_{v \in \mathcal{V}} \operatorname{CosSim}\left(\widehat{\boldsymbol{Z}}^0_{b,i}, e_v\right)$
return $\{\hat{t}_{b,i}\}$

Image reordering Algorithm 7 scores a pixel array by the forward loss ℓ_{forw} of its image tokens (line 1), computed from a forward pass with $f^{\text{mm}}(\boldsymbol{X}^{\text{mm}})$ at Ω and the fixed prefix at $\mathcal{K}$. In each round, it evaluates the swap of every pair of patches and applies the swap with the largest decrease of ℓ_{forw}, if one exists. This continues until no swap decreases the loss, or a maximum number of $N_{\text{rounds}} = 30$ rounds is reached.

Algorithm 7 IMAGEREORDER: Greedy Patch Swapping

Input: Pixel array $\boldsymbol{X}^{\text{mm}}$ with patches $\{\boldsymbol{X}_i^{\text{mm}}\}_{i \in \Omega}$,
Positions Ω, $\mathcal{K}$, embeddings $\widehat{\boldsymbol{Z}}^0_{\mathcal{K}}$, Attention mask $\boldsymbol{M}$,
Maximum number of rounds N_{rounds}
Output: Reordered pixel array $\boldsymbol{X}^{\text{mm}}$ (temporaries written $\boldsymbol{X}'$, $\boldsymbol{X}^\star$)
1: **define** $\ell_{\text{forw}}(\boldsymbol{X})$: $\bar{\boldsymbol{Z}}^0_\Omega \leftarrow f^{\text{mm}}(\boldsymbol{X})$, $\bar{\boldsymbol{Z}}^0_{\mathcal{K}} \leftarrow \widehat{\boldsymbol{Z}}^0_{\mathcal{K}}$, $\{\bar{\boldsymbol{Z}}^l\}_{l=1}^{L_{\max}} \leftarrow \text{FORWARD}(\bar{\boldsymbol{Z}}^0; \boldsymbol{M})$, **return** $\sum_{p \in \Omega} \ell_{\text{forw},p}$
2: $r \leftarrow 0$
3: **repeat**
4: $r \leftarrow r + 1$, $\Delta^\star \leftarrow 0$
5: **for all** pairs $i < j$ of patches in Ω **do**
6: $\boldsymbol{X}' \leftarrow \boldsymbol{X}^{\text{mm}}$ with $\boldsymbol{X}_i^{\text{mm}}$ and $\boldsymbol{X}_j^{\text{mm}}$ exchanged
7: $\Delta \leftarrow \ell_{\text{forw}}(\boldsymbol{X}^{\text{mm}}) - \ell_{\text{forw}}(\boldsymbol{X}')$
8: **if** $\Delta > \Delta^\star$ **then**
9: $\Delta^\star \leftarrow \Delta$, $\boldsymbol{X}^\star \leftarrow \boldsymbol{X}'$
10: **if** $\Delta^\star > 0$ **then**
11: $\boldsymbol{X}^{\text{mm}} \leftarrow \boldsymbol{X}^\star$
12: **until** $\Delta^\star = 0$ or $r = N_{\text{rounds}}$
13: **return** $\boldsymbol{X}^{\text{mm}}$

Text reordering Algorithm 8 operates on the discrete token batch $\{\hat{t}_{b,i}\}$ and scores a candidate batch by the backward loss ℓ_{back} (line 1), computed from a forward pass of the token embeddings $e_{\hat{t}_{b,i}}$. We use the backward loss here because a text move can rearrange several positions or redistribute content across sequences, and coverage of the observed subspace measures the effect of such moves better than the per-position forward loss. Two move types are considered:

- MOVESUBSEQ(b, i, k, j) removes the subsequence $u = \hat{t}_{b,i:i+k-1}$ of length k from sequence b and reinserts it so that it starts at index j of the shortened sequence: with w denoting sequence b without u, the new sequence is $[w_{1:j-1}; u; w_{j:T_b-k}]$ for $j \in \{1, \ldots, T_b - k + 1\}$, $j \neq i$. The displaced tokens shift accordingly, and the sequence length is unchanged.
- EXCHANGESUBSEQ(b, b', j), for a shorter sequence b and a longer sequence b' with $T_b < T_{b'}$, inserts the whole of sequence b into sequence b' at index $j \in \{1, \ldots, T_{b'} - T_b + 1\}$, and the displaced tokens become the new sequence b: the new sequences are $[\hat{t}_{b',1:j-1}; \hat{t}_{b,1:T_b}; \hat{t}_{b',j+T_b:T_{b'}}]$ and $\hat{t}_{b',j:j+T_b-1}$, respectively. Both lengths are preserved.

As for images, each round evaluates every move of both types, applies the one with the largest decrease of ℓ_{back}, and stops when no move decreases the loss, or a maximum number of $N_{\text{rounds}} = 30$ rounds is reached.

Algorithm 8 TEXTREORDER: Greedy Subsequence Moves

Input: Token batch $\{\hat{t}_{b,i}\}$ with lengths $\{T_b\}_{b=1}^{B}$, Attention mask $\boldsymbol{M}$, Maximum number of rounds N_{rounds}
Output: Reordered token batch $\{\hat{t}_{b,i}\}$

define $\ell_{\text{back}}(\{\hat{t}_{b,i}\})$: $\bar{\boldsymbol{Z}}^0_{b,i} \leftarrow e_{\hat{t}_{b,i}}$ for all (b,i), $\{\bar{\boldsymbol{Z}}^l\}_{l=1}^{L_{\max}} \leftarrow \text{FORWARD}(\bar{\boldsymbol{Z}}^0; \boldsymbol{M})$, **return** ℓ_{back} of Equation (6)
$r \leftarrow 0$
repeat
  $r \leftarrow r + 1$, $\Delta^\star \leftarrow 0$
  for all moves $m \in \{\text{MOVESUBSEQ}(b,i,k,j)\} \cup \{\text{EXCHANGESUBSEQ}(b,b',j)\}$ **do**
    $\{\hat{t}_{b,i}\}' \leftarrow m(\{\hat{t}_{b,i}\})$
    $\Delta \leftarrow \ell_{\text{back}}(\{\hat{t}_{b,i}\}) - \ell_{\text{back}}(\{\hat{t}_{b,i}\}')$
    if $\Delta > \Delta^\star$ **then**
      $\Delta^\star \leftarrow \Delta$, $\{\hat{t}_{b,i}\}^\star \leftarrow \{\hat{t}_{b,i}\}'$
  if $\Delta^\star > 0$ **then**
    $\{\hat{t}_{b,i}\} \leftarrow \{\hat{t}_{b,i}\}^\star$
until $\Delta^\star = 0$ or $r = N_{\text{rounds}}$
return $\{\hat{t}_{b,i}\}$

C ADDITIONAL EXPERIMENTAL DETAILS

This section expands on the experimental setup of the main evaluation from §5.1, including more on dataset construction, evaluation procedures, attack settings, and baseline adaptations.

C.1 DATASETS

First, we comment on the general structure and preprocessing of the datasets used in our experiments.

Images We select 10 document and 10 scenery images from VPD-100K. This is possible as the dataset contains a classification label for each image, describing the content. We namely sample images within the following categories: "Adult Face Indoor", "Adult Face Outdoor", "Child Face Indoor", "Elder Face Outdoor" for scenery images leaking identifiable faces, as well as "Street Sign", "Store Sign", "Drivers License", "Chinese Car Plate", "Bank Card", "ID Card", "Passport", "Name", "ID Number", "Address" to capture private text from addresses, locations, license plates, and faces. The "Bank Card", "ID Card", "Passport", "Name", "ID Number" categories correspond to document images, while the rest correspond to scenery images. After the images were sampled and manually verified for legibility, we cropped each image to a compact square that retains all visible PII, and used Claude Opus 5 (Anthropic, 2026) to generate bounding boxes around the private information and a question–answer pair for each image. We prompt the model for a single-sentence question and a single-sentence answer about the image content, asking for the questions to be diverse across images and for the answers to be concise. In each sample, the question precedes the image and the answer follows it, so that the image is surrounded by text on both sides.

Textual datasets Each client batch consists of B examples sampled from the respective dataset. As sentences may have discordant lengths, shorter sequences are padded on the right to match the length of the longest one. That said, padding tokens are ignored during gradient computation and evaluation.

In terms of token sequence lengths, SST-2 and RottenTomatoes average 12 and 26 tokens per sequence, while the question and answer within the image average 9.5 and 10 tokens respectively.

C.2 EVALUATION METRICS

Text reconstruction We report token-level ROUGE-1 and ROUGE-2 F1 scores (Lin, 2004) between the reconstructed sequences and the original client batch over the set of token IDs. ROUGE-

1 measures unigram overlap, while ROUGE-2 measures longest-common-subsequence overlap, capturing both token-level recovery and partial sequence-order recovery. Since the attack recovers the client sequences without a canonical ordering within the batch, for batch sizes $B > 1$ we align reconstructions with originals via a one-to-one matching: each reconstructed sequence is assigned to a distinct original sequence, and no two reconstructions may be matched to the same original. Among all such matchings, we select the one that maximizes the sum of ROUGE-1 scores over the batch, breaking ties by the sum of ROUGE-2 scores, and report ROUGE scores averaged over the matched pairs.

For decoder experiments, we exclude the final input token of each sequence from evaluation for all attacks. Under our next-token prediction setup, the final input position has no in-sequence next-token target, and no later loss term can depend on it through the causal attention path. As a result, its embedding is not identifiable from the gradient signal used in our evaluation.

Our PII metric instead relies on specific values provided by the dataset itself, which are classified as PII. As the PII recovery accuracy, we report the fraction of such fields appearing *exactly* within the reconstructed sequence.

Image reconstruction For the PSNR and SSIM metrics on images, we first align the reconstructed images with the originals in terms of scale and shift. We do so because the LayerNorm in the image preprocessing removes scale and shift information, to which PSNR and SSIM are sensitive. This is information that is completely lost to the gradients and cannot be recovered by any gradient inversion attack. To this end, before computing these metrics, we therefore fit a linear transformation for each reconstructed image to align its scale and shift with the original. Image-token accuracy is then computed as the fraction of reconstructed embeddings whose cosine similarity with the corresponding ground-truth embedding exceeds 0.90.

PII recovery from images To understand the amount of PII leakage for images, we extract recovered information using an LLM-as-a-judge to text the legibility of text, and the similarity of faces. We use GEMINI-3.1-PRO (Google DeepMind, 2026) to compare the PII regions identified by bounding boxes in each original image with the corresponding reconstructed regions. For text, we let the judge extract the text, as it interprets it, and then measure character-wise text similarity. This includes both document information and text in license plates, addreses, and other. On the other hand, we use a three-point scale for faces inside scenery images and documents, namely scoring the legibility separately as 0 for not legible, 0.5 for partial recognizability, and 1 for a full face leak. This is ran on both the ground truth and the recovered image on the same resolution using 5-sample majority voting. 0-scoring faces within the ground-truth are discarded, with recovered samples scoring 1 if they match the recoverability of the ground-truth, 0.5 for partial recovery when the originals are fully legible, and 0 otherwise. Faces that are visible but occluded were determined by a separate LLM-as-a-judge call and were removed. The prompts and face-scoring protocol are given in §J.

An author manually went through roughly 100 judgements to validate this protocol. For text, they determined that all the extracted text made logical sense, noting that when illegible, the LLM judge made sensible guesses in all its attempts. On the other hand, for faces, the judge was largely consistent under the 5-sample majority voting paradigm, only noting only a failure mode where slightly different hues caused the judge to slightly underestimate our performance on partially legible faces. This nevertheless happens on no more than 10% of cases, and actively lowers our results, so we consider this component to be sufficiently robust for the purposes of this work.

C.3 HYPERPARAMETERS AND NUMERICAL PRECISION

Unless otherwise stated, TIGER uses $L_{\max} = 15$ layers in the span loss and $N_{\text{init}} = 500$ initializations, each optimized for $N_{\text{steps}} = 3000$ Adam steps with an initial learning rate of $\eta = 3 \times 10^{-2}$. For BERT, which has only 12 layers, we use all of them. The rank of each gradient subspace $\mathcal{S}^l$ is set by the singular-value threshold of the truncated SVD. We use a threshold 100 times smaller than the standard numerical-rank tolerance $\sigma_{\max}\, d\, \epsilon$, where $\sigma_{\max}$ is the largest singular value of $\partial \mathcal{L}/\partial \boldsymbol{W}_Q^l$ and ϵ the machine precision, which is less restrictive than the default and retains weak but informative directions. A `ReduceLROnPlateau` scheduler is used to reduce the learning rate by a factor of 0.1 on plateau, with patience 100 and a minimum learning rate of 10^{-6}.

Under bidirectional attention, we set the backward-loss coefficient to $\lambda_{\text{back}} = 0.3$ for both text encoders and images, use $N_{\text{S}} = 10$ stages of interleaved continuous and discrete optimization with at most $N_{\text{rounds}} = 30$ greedy reordering rounds per stage, and set the total-variation coefficient for images to $\lambda_{\text{TV}} = 0.1$. For decoders, the deduplication coefficient is $\lambda_{\text{dedup}} = 0.1$, 0.05, and 0.0125 for $B = 2$, 4, and 8, respectively. We select these settings on a small set of validation batches and keep them fixed in subsequent experiments. Client gradients are computed in FP32, while attack computations use BF16 where supported, unless otherwise stated.

C.4 Hardware

GPU experiments use a single NVIDIA H200 NVL GPU (141 GB VRAM), 16 CPU cores, and 128 GB of RAM. Most DAGER experiments use 32 CPU cores and 180 GB of RAM. This allocation is sufficient because failed DAGER runs typically terminate early during recovery.

C.5 Baseline Attacks

We compare TIGER against two strong text-specific gradient inversion attacks: LAMP (Balunovic et al., 2022) and DAGER (Petrov et al., 2024). LAMP is an optimization-based attack that alternates between continuous and discrete optimization while incorporating a language-model prior. DAGER is an algebraic attack that exploits low-rank structure in transformer gradients to enable exact or near-exact recovery under certain architectural and batch-size conditions. These methods are strong prior baselines and provide meaningful comparisons across the settings we study.

LAMP We adapt LAMP for the encoder setting. Following its original design, LAMP uses a language-model prior during optimization. We use GEMMA-3-4B-IT as this prior when attacking EMBEDDINGGEMMA-300M, giving LAMP a stronger prior compared to the GPT-2 prior used in the original LAMP evaluation, and without needing to adapt the tokenization. Because LAMP is sensitive to hyperparameters, particularly the perplexity coefficient, learning-rate decay, and gradient-matching loss variant, we tune these parameters before running the main experiments. Specifically, we consider initializations of the form $\bar{Z}_0 = Z_0 + Z_\sigma$, where $Z_\sigma \sim \mathcal{N}(0, \sigma^2)$, representing the ground-truth initialization Z_0 with added Gaussian noise. We increase σ until LAMP's ROUGE-1 score drops to 0. At the largest value of σ for which LAMP still achieves nonzero ROUGE-1, we perform a grid search over the relevant hyperparameters and select the configuration with the highest mean ROUGE-1. We then use this fixed configuration for all subsequent LAMP experiments.

For the decoder setting, LAMP is not directly applicable because it assumes knowledge of the labels. In next-token prediction, these labels are the shifted input tokens and would directly leak the input sequence. We nevertheless attempted a label-free adaptation of LAMP, but found that it failed to achieve nonzero ROUGE-1 even in the $B = 1$ setting. We therefore do not include LAMP in the reported decoder tables.

DAGER For the decoder setting, we adapt DAGER to the models used in our evaluation. We follow the original paper by setting the first-layer filtering threshold to 10^{-5}, but we relax the second-layer threshold to 10^{-1}. We found that smaller second-layer thresholds were too strict in our setting and removed valid candidates. In the undefended decoder setting, this relaxed threshold still allows DAGER to recover efficiently, as shown in §5.

We do not include DAGER on the encoder setting featuring EMBEDDINGGEMMA-300M. In particular, EMBEDDINGGEMMA-300M uses RoPE-style positional embeddings, which differ from the positional-embedding assumptions in the original DAGER setup. Because RoPE is applied after the query and key projections, a direct adaptation of DAGER can recover a bag of tokens for the sequence but not their positions. Recovering ordered batches would then require enumerating possible assignments of the recovered BT tokens into B ordered sequences of length T. This search space is on the order of $\frac{(BT)!}{B!}$. Even for $B = 1$ for RottenTomatoes this is approximately $\frac{27!}{} \approx 10^{28}$ possible enumerated batches, making direct enumeration computationally infeasible.

Table 6: Assumptions of honest-but-curious gradient inversion attacks and of TIGER under causal and bidirectional attention. ✓ marks an assumption the attack makes, × one it does not.

a Causal attention (decoders).

	Known labels	Known seq. lengths	LM prior	Finite vocabulary	Exact gradients	Restricted architecture
TAG (Deng et al., 2021)	×	✓	×	✓	×	×
LAMP (Balunovic et al., 2022)	✓	✓	✓	✓	×	×
FILM (Gupta et al., 2022)	×	×	✓	✓	×	✓
DAGER (Petrov et al., 2024)	×	×	×	✓	✓	×
TIGER (ours)	×	×	×	×	×	×

b Bidirectional attention (encoders).

	Known labels	Known seq. lengths	LM prior	Finite vocabulary	Exact gradients	Restricted architecture
Image gradient matching (Zhu et al., 2019; Zhao et al., 2020; Geiping et al., 2020; Yin et al., 2021; Hatamizadeh et al., 2022)	✓	✓	×	×	×	×
TAG (Deng et al., 2021)	×	✓	×	✓	×	×
LAMP (Balunovic et al., 2022)	✓	✓	✓	✓	×	×
APRIL (Lu et al., 2022)	×	×	×	×	✓	✓
DAGER (Petrov et al., 2024)	×	✓	×	✓	✓	✓
TIGER (ours)	×	✓	×	×	×	×

D Threat Model Comparison

Table 6 compares the assumptions of TIGER against those of prior honest-but-curious gradient inversion attacks, restricted to the assumptions that most affect applicability. As the assumptions differ between attention types, we compare causal (decoder) attacks in Table 6a and bidirectional (encoder) attacks in Table 6b.

Side information All attacks in Table 6 assume an honest-but-curious server that knows the model and observes a single-step gradient. Gradient-matching attacks optimize dummy inputs of a fixed shape and hence need the sequence lengths, and most also need the labels, either assumed known (Geiping et al., 2020; Balunovic et al., 2022) or derived from the last-layer gradient (Zhao et al., 2020; Yin et al., 2021). Under causal attention, the label assumption is what prevents LAMP from being applied to next-token prediction, where the labels reveal the target text, while TIGER never needs said labels. Under causal attention, it recovers the sequence lengths through its termination criterion, and only under bidirectional attention does it assume them, as prior optimization-based attacks do (Zhu et al., 2019; Geiping et al., 2020; Deng et al., 2021; Balunovic et al., 2022; Petrov et al., 2024). For images, this amounts to knowing the number of image tokens, but not their position in the text.

Priors and input domain LAMP (Balunovic et al., 2022) and FILM (Gupta et al., 2022) rely on a pretrained language model to score or generate candidates, which ties them to text and to the availability of a model matching the client's tokenizer. Text attacks further require a finite vocabulary, either to project embeddings back to tokens (Deng et al., 2021; Balunovic et al., 2022) or to enumerate candidates (Gupta et al., 2022; Petrov et al., 2024), which rules out continuous multimodal inputs. Conversely, image attacks operate on continuous inputs but have no mechanism for discrete text. TIGER uses only lightweight priors that require no trained model, namely public-corpus embedding statistics for text and Gaussian pixels for images, and optimizes continuous embeddings throughout, so it recovers discrete text and continuous image tokens within a single batch.



Figure 3: Single-frame stage-1 visualizations across the different ablation settings.

Gradient precision and architecture Algebraic attacks recover the batch exactly, but through hard numerical tests that assume unperturbed, full-precision gradients, such as DAGER's span-membership thresholds, which collapse under gradient noise. Under bidirectional attention, they are additionally derived from the structure of particular layers: learnable positional embeddings for APRIL (Lu et al., 2022), and absolute positional embeddings for DAGER (Petrov et al., 2024), which is why it is not tractable for EMBEDDINGGEMMA-300M (§C.5). TIGER inherits the low-rank condition $T \leq \frac{L_{\max}-1}{L_{\max}} d$ of Corollary 4.1, but replaces the thresholded tests with the continuous objective of Equation (2), so it degrades gracefully under noise (Table 4), and it uses only the query-projection gradients present in every transformer, including RoPE models and multimodal encoders.

E ADDITIONAL EXPERIMENTS AND ABLATION STUDIES

In this section, we present additional experiments to support our design choices and validate the effectiveness of the proposed components.

E.1 EFFECTS OF MULTIMODAL COMPONENTS

We first investigate how the different multimodal additions affect the performance of TIGER. To this end, for the loss-specific components, we ran the first stage's continuous optimization, and compare the resulting image token accuracy when each component is removed. For reordering, which is an operation acting on the optimization chain, we evaluate the full attack with and without the reordering step. The resulting accuracies and the difference with respect to the full method are reported in Table 7.

Effect of continuous optimization components Dropping only $\ell_{\text{TV},\partial}$ leaves token accuracy unchanged within run-to-run noise, consistent with this term primarily cleaning up boundary artifacts. By contrast, removing both TV terms causes a much larger drop, which matches the role of $\ell_{\text{TV,in}}$ in improving the image itself. Removing ℓ_{back} also hurts substantially, showing that the backward span-distance term remains complementary to the image prior, and is valuable to the optimization as a whole.

Table 7: Image token accuracy when discarding different components.

Method	Token acc.	Δ
TIGER (Stage 1)	59.0%	–
No seam TV	59.2%	+0.2%
No backward loss	47.5%	-11.5%
No TV	40.5%	-18.5%
TIGER	95.1%	–
No reordering	88.4%	-6.7%

This is also confirmed visually in Figure 3, which shows exactly the failure modes we would expect from removing the different continuous optimization components. Removing the border TV makes the image patchy, but does not severely degrade the overall structure. In constrast, removing both TV terms leads to a much blurrier and less coherent image, while removing the backward loss results in some hallucinations, particularly some duplicated objects, as some people's hands.

Effect of discrete reordering across stages A key feature of our attack's design is the multi-stage optimization process, where optimization is iteratively refined by discrete operations, such as token reordering or fixing poorly initialized tokens. We see in Figure 4 that more stages consistently

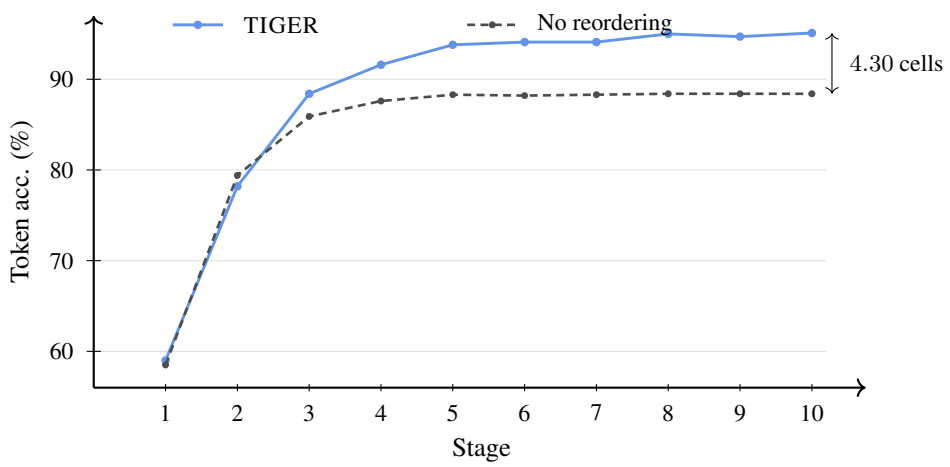

Figure 4: Per-stage mean image-token accuracy, comparing TIGER with the no-reordering variant.

contribute to improved performance, but start to saturate around Stages 8–10, where additional stages mostly refine the existing token arrangements. We further quantify how much the token reordering application contributes to the improved image quality. While the difference is negligible in the first couple of stages, the improved structure during this time allows us TIGER to improve later on, reaching a peak advantage of $4.30/64$ patches by stage 10.

E.2 Impact of Initialization Strategy

Setup The text ablations in this subsection and in §E.3 were run during the development of TIGER, before the final evaluation of §5, on a development dataset derived from WikiText (Merity et al., 2016). For development purposes, we limited the sequences to $T_{\text{max}} = 16$ tokens for GEMMA-3-4B-IT and $T_{\text{max}} = 32$ tokens for EMBEDDINGGEMMA-300M, with batches of $B = 4$. The GEMMA-3-4B-IT gradients were additionally perturbed by Gaussian noise with $\sigma = 5 \times 10^{-5}$ (§5.3), while the EMBEDDINGGEMMA-300M gradients were left unperturbed. Within each ablation, all variants are evaluated on the same 10 batches, so they are directly comparable to each other, but not to the results of §5. Furthermore, EMBEDDINGGEMMA-300M was run on a single optimization iteration to better isolate the effects of our components, hence no reordering is performed in these experiments.

Table 8: Impact of different initialization strategies on TIGER.

Model	Configuration	R-1	R-L
GEMMA-3-4B-IT	Default	54.9	**54.3**
	IMDb Init.	**55.4**	54.0
	Vocabulary Init.	46.0	44.9
	Random Init.	24.3	24.3
EMBEDDINGGEMMA-300M	Default	**72.2**	**54.1**
	IMDb Init.	71.9	51.7
	Vocabulary Init.	67.4	47.2
	Random Init.	68.8	49.9

Result Initialization provides an important prior for the non-convex embedding optimization, especially for discrete modalities, such as text. We compare the default position-dependent WikiText Gaussian prior of §4.1 against alternative initialization strategies in Table 8.

Fitting the same position-dependent Gaussian prior on an IMDb-derived dataset (Maas et al., 2011), instead of WikiText yields nearly identical performance in both settings, suggesting that the specific public corpus used to estimate the prior is not the main factor. In contrast, sampling from a position-agnostic distribution over the vocabulary embedding space reduces reconstruction quality, indicating that position-specific statistics provide useful information. Finally, independently sampling embedding coordinates degrades decoder performance substantially and also underperforms the structured prior for encoders. These results highlight the importance of preserving realistic embedding-space structure during initialization.

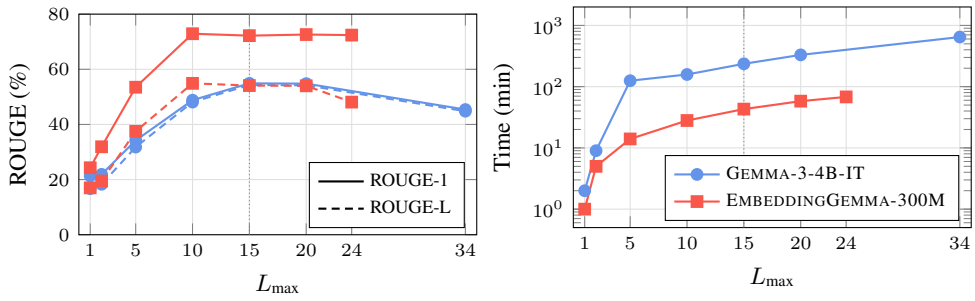

Figure 5: Impact of the number of span-loss layers $L_{\max}$ on TIGER: ROUGE (left) and mean wall-clock time per attack in minutes (right, log scale). The rightmost point of each model uses all of its layers, and the dotted line marks the default $L_{\max} = 15$.

E.3 Number of Span-Loss Layers

The number of span-loss layers $L_{\max}$ controls a tradeoff between reconstruction signal and computational cost. Using more layers adds subspace constraints and stabilizes the optimization, but it also increases the cost of each forward and backward pass. We vary $L_{\max}$ and report ROUGE and the mean wall-clock time per attack in Figure 5.

With a single layer, TIGER fails for both models, in line with Corollary 4.1, under which the true input cannot be a stable minimizer for $L_{\max} = 1$. Reconstruction then improves rapidly with the first few layers and saturates at around 10–15 layers, while the runtime grows roughly linearly in $L_{\max}$. For GEMMA-3-4B-IT, using all 34 layers even lowers ROUGE-1 from 54.9 to 45.4, suggesting that the constraints of the deepest layers make the optimization harder rather than more informative. We therefore use $L_{\max} = 15$ by default (§C.3).

E.4 Sequence Termination

We now ablate our design choice on how we stop sequence recovery in the decoder setting. Since no position that receives a gradient attends to the EOS token under the causal mask, we cannot simply recover the EOS token. We thus use a calibrated metric to determine when our recovery has likely strayed beyond the ground truth.

Score At position i of sequence b, let $z^\star$ be the optimized embedding and $v = \operatorname{argmax}_{v \in \mathcal{V}} \operatorname{CosSim}(z^\star, e_v)$ its nearest token. The snap gap is

$$\delta_{b,i} = \ell_{\text{forw},b,i}(e_v) - \ell_{\text{forw},b,i}(z^\star),$$

as in line 8 of Algorithm 2.

Rule Each sequence is recovered to a public horizon H, and $\delta_{b,i}$ is recorded at every position. The sequence is cut at the first position with $\delta_{b,i} > \tau_{\text{snap}}$. The inferred length is that position, and EOS is appended. A sequence without a crossing keeps length H. For each sequence we report whether the rule fired exactly at the true boundary (*exact*), before it (*early*), after it (*late*), or not at all before the public horizon (*miss*). Table 9 lists these outcomes for the cells of Table 3.

Calibration and validation on public data τ_{snap} is fitted once per model on attacks against public WikiText passages, whose true lengths are known. We refer to the positions before the true end of a sequence as content positions and to the position of the true end as the boundary position. τ_{snap} is the 99th percentile of δ over the content positions of the calibration attacks, so 1% of these positions lie above τ_{snap}. This gives $\tau_{\text{snap}} = 0.2191$ for GEMMA-3-4B-IT and $\tau_{\text{snap}} = 10.1246$ for LLAMA-3.1-8B. Before τ_{snap} is applied to the private datasets, it is checked on public WikiText attacks that were not used in the fit. For GEMMA-3-4B-IT, these are a second, disjoint set of held-out WikiText attacks, scored with the frozen τ_{snap} (Table 10).

Table 9: Inferred-length outcomes under the frozen τ_{snap} (exact / early / late / miss over all sequences of the cell).

Model	Dataset	H	B	Sequences	Exact / early / late / miss
GEMMA-3-4B-IT	SST-2	21	1	10	10 / 0 / 0 / 0
	SST-2	21	2	20	20 / 0 / 0 / 0
	SST-2	21	4	20	20 / 0 / 0 / 0
	SST-2	21	8	40	35 / 3 / 2 / 0
	Rotten Tomatoes	37	1	10	8 / 2 / 0 / 0
	Rotten Tomatoes	37	2	20	17 / 3 / 0 / 0
	Rotten Tomatoes	37	4	40	29 / 11 / 0 / 0
	Rotten Tomatoes	37	8	80	30 / 41 / 9 / 0
LLAMA-3.1-8B	SST-2	23	1	10	7 / 2 / 1 / 0
	SST-2	23	2	20	13 / 4 / 2 / 1

Table 10: Single-position scores on the GEMMA-3-4B-IT public WikiText cells. AUROC contrasts boundary positions with content positions. Held-out outcome: exact lengths / 20 sequences.

Score	AUROC cal.	AUROC held-out	Held-out outcome
Snap gap δ	0.960	0.952	19/20 (0/90)
L_{snap}	0.971	0.972	18/20 (1/90)
Nearest-token cosine distance	0.961	0.952	15/20 (4/90)

Comparison scores To justify the choice of the snap gap δ, we compare it against two alternative single-position scores that could also be used as a termination signal. The first is the snapped-token loss $L_{\text{snap}} = \ell_{\text{forw},b,i}(e_v)$, i.e., the first term of δ alone. It measures how well the nearest vocabulary token explains the observed gradient subspaces, but, unlike δ, it is not compared against the loss $\ell_{\text{forw},b,i}(z^\star)$ reached by the unconstrained optimization at the same position. The second is the nearest-token cosine distance $1 - \text{CosSim}(z^\star, e_v)$, which only measures how far the optimized embedding lies from the vocabulary in embedding space and does not use the gradient at all.

All three scores are evaluated on the same GEMMA-3-4B-IT public WikiText cells used to calibrate and validate τ_{snap}. Every score is thresholded at the same operating point as δ: its cutoff is the 99th percentile of the score over the content positions of the calibration cell, which gives a cutoff of 1.2619 for L_{snap}. The cutoffs are then frozen and applied with the same first-crossing rule to the held-out cell. Table 10 reports the AUROC, which measures how well each score separates boundary positions from content positions over all possible thresholds, and the held-out outcome at the frozen cutoff, which is what determines the recovered lengths in practice.

All three scores separate boundary from content positions well, with AUROCs between 0.95 and 0.97 on both cells, and L_{snap} even attains the highest AUROC. At the frozen operating point, however, the snap gap gives the best held-out outcome: it infers the exact length of 19 of the 20 sequences, the remaining one being late, and it does not fire on any of the 90 content positions. L_{snap} recovers 18 exact lengths with one content false positive, and the cosine distance only 15 exact lengths with four content false positives. The gap between the AUROC and the held-out outcome comes from the first-crossing rule: a single content position above the cutoff truncates the sequence early and discards all tokens after it, so the termination quality is governed by the tail of the score over content positions rather than by its average ranking. Subtracting $\ell_{\text{forw},b,i}(z^\star)$ keeps this tail low, as positions where the gradient subspaces are hard to fit even for the unconstrained embedding raise both terms of δ and do not trigger termination. We therefore use δ, which additionally comes at no extra cost, as it is already computed in line 8 of Algorithm 2 to decide whether to accept a token. On the calibration cell itself, δ yields 18 exact, 1 early, and 1 late length.

E.5 Training Utility under Additive Gradient Noise

To select the noise levels in Table 4, we fine-tune Gemma-3-4B-IT on the English training split of PII-masking-300k (Ai4Privacy, 2024) for three epochs with AdamW (learning rate 10^{-6}, batch size 4). At every step, we add independent noise $\mathcal{N}(0, \sigma^2)$ to each coordinate of the unclipped batch gradient. We report perplexity on the held-out validation split, averaged over three seeds at each run's best epoch. As shown in Figure 6, perplexity stays within 4% of the noise-free value up to $\sigma = 10^{-3}$, but starts increasing at $\sigma = 10^{-3}$ to $\sigma = 10^{-2}$, showing some training degradation. We therefore constrain our evaluation to $\sigma \leq 10^{-3}$.

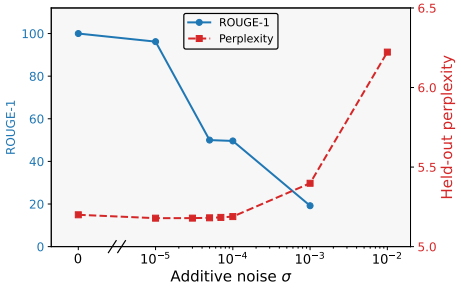

Figure 6: Held-out perplexity of Gemma-3-4B-IT fine-tuned with additive gradient noise σ.

F Feasibility Analysis of Subspace Inversion

In this section, we include supplementary analysis on the feasibility of continuous span distance gradient inversion attacks. In particular, we derive sufficient, and under some architectures necessary, conditions for the uniqueness of global minima under the forward reconstruction objective. We show that, in an exact and noiseless setting, the true hidden states are a zero of the multilayer span-constraint map, locally isolated under a rank condition (Theorem F.1), which applies to transformers (Corollary 4.1), and globally unique up to permutation for a family of networks (Theorem F.2).

F.1 Setup

Network setting Consider a neural network $\mathcal{N}$ with known parameters, and a set of $r \geq 1$ inputs $\boldsymbol{Z}^0 \in \mathbb{R}^{r \times d_0}$. We represent the network $\mathcal{N}$ as L blocks $\Phi^1, \ldots, \Phi^L$, preceded by an input map Φ^0 and followed by an output head Φ^{out},

$$\mathcal{N} = \Phi^{\text{out}} \circ \Phi^L \circ \cdots \circ \Phi^1 \circ \Phi^0,$$

with features, hidden states and output:

$$\boldsymbol{Z}^1 = \Phi^0(\boldsymbol{Z}^0), \qquad \boldsymbol{Z}^{l+1} = \Phi^l(\boldsymbol{Z}^l) \ \ (l = 1, \ldots, L), \qquad \boldsymbol{O} = \Phi^{\text{out}}(\boldsymbol{Z}^{L+1}),$$

where $\boldsymbol{Z}^l \in \mathbb{R}^{r \times d^l_{\text{in}}}$. We consider blocks that begin with a fully connected layer with weights $\boldsymbol{W}^l \in \mathbb{R}^{d^l_{\text{in}} \times d^l_{\text{out}}}$:

$$\boldsymbol{Y}^l = \boldsymbol{Z}^l \boldsymbol{W}^l,$$

where the weights $\boldsymbol{W}^1, \ldots, \boldsymbol{W}^L$ are untied, i.e., not shared with or tied to any other layer, so that the gradient with respect to $\boldsymbol{W}^l$ depends on $\boldsymbol{Z}^l$ alone. The blocks $\Phi^1, \ldots, \Phi^{L_{\max}-1}$ between probed layers are continuously differentiable on open neighborhoods of the true hidden states, and may contain mechanisms that mix the rows. In a transformer, Φ^l is a full transformer block f^l, whose attention mixes the tokens. We assume the batch has fewer hidden states than the layer width and none of them is a linear combination of the others, implying $\operatorname{rank}(\boldsymbol{Z}^l) = r < d^l_{\text{in}}$ for $l = 1, \ldots, L_{\max}$.

The client computes a differentiable training loss $\mathcal{L}(\boldsymbol{O})$ and shares the weight gradients $\boldsymbol{G}^l := \partial \mathcal{L} / \partial \boldsymbol{W}^l$. We write $\boldsymbol{\Delta}^l := \partial \mathcal{L} / \partial \boldsymbol{Y}^l$ for the loss gradient w.r.t the linear layer outputs of block l, which are themselves, not shared. We assume these gradients are full-rank, i.e., $\operatorname{rank}(\boldsymbol{\Delta}^l) = r$ due to the non-linearity of the subsequent layers, which is empirically observed in practice.

Attack setting For the purposes of the analysis, we consider a TIGER-style low-rank attack that probes the initial linear layers of the first $L_{\max} \leq L$ blocks. Let $\mathcal{S}^l := \operatorname{rowspan}(\boldsymbol{Z}^l)$ be the span of the layer-l hidden states. Given the full-rank nature of $\boldsymbol{\Delta}$, Theorem 3.1 implies that $\operatorname{colspan}(\boldsymbol{G}^l) = \mathcal{S}^l$, so the attacker observes each $\mathcal{S}^l$ exactly. Let $\boldsymbol{P}^l$ be the orthogonal projection onto $\mathcal{S}^l$. The attack optimizes a dummy batch $\bar{\boldsymbol{Z}}^1 \in \mathbb{R}^{r \times d^1_{\text{in}}}$, propagated forward as $\bar{\boldsymbol{Z}}^{l+1} = \Phi^l(\bar{\boldsymbol{Z}}^l)$, to minimize

$$\mathcal{L}_{\text{span}}(\bar{\boldsymbol{Z}}^1) := \sum_{l=1}^{L_{\max}} \left\| \bar{\boldsymbol{Z}}^l \left(\boldsymbol{I} - \boldsymbol{P}^l \right) \right\|_F^2, \tag{10}$$

on the open set where the propagation is defined. This defintion matches the forward loss of Equation (2) without its per-row normalization, which does not change its zeros for nonzero rows. We call the attack *identifiable* if

(I1) the true batch $\boldsymbol{Z}^1$ is a global minimizer of $\mathcal{L}_{\text{span}}$, and

(I2) every global minimizer contains only true inputs, that is, each of its rows is a row of $\boldsymbol{Z}^1$.

F.2 A NECESSARY CONDITION FOR EVERY ARCHITECTURE

Theorem F.1. *In the setting above:*

(a) The true batch $\boldsymbol{Z}^1$ is a global minimizer of $\mathcal{L}_{\text{span}}$, with $\mathcal{L}_{\text{span}}(\boldsymbol{Z}^1) = 0$.

(b) The true batch $\boldsymbol{Z}^1$ can be a stable minimizer of $\mathcal{L}_{\text{span}}$, meaning that $\mathcal{L}_{\text{span}}(\bar{\boldsymbol{Z}}^1) \geq c\,\|\bar{\boldsymbol{Z}}^1 - \boldsymbol{Z}^1\|_F^2$ for some $c, \varepsilon > 0$ and all $\bar{\boldsymbol{Z}}^1$ with $\|\bar{\boldsymbol{Z}}^1 - \boldsymbol{Z}^1\|_F < \varepsilon$, only if

$$\sum_{l=2}^{L_{max}} (d_{\text{in}}^l - r) \geq r. \tag{11}$$

Proof. Part (a). At $\bar{\boldsymbol{Z}}^1 = \boldsymbol{Z}^1$, the dummy states equal the true states $\boldsymbol{Z}^l$, whose rows lie in $\mathcal{S}^l$. Hence every term of $\mathcal{L}_{\text{span}}$ vanishes and $\mathcal{L}_{\text{span}}(\boldsymbol{Z}^1) = 0$.

Part (b). For each probed layer, let $\boldsymbol{U}_\perp^l \in \mathbb{R}^{d_{\text{in}}^l \times (d_{\text{in}}^l - r)}$ have orthonormal columns spanning $(\mathcal{S}^l)^\perp$, so that $\boldsymbol{I} - \boldsymbol{P}^l = \boldsymbol{U}_\perp^l (\boldsymbol{U}_\perp^l)^\top$ and $(\boldsymbol{U}_\perp^l)^\top \boldsymbol{U}_\perp^l = \boldsymbol{I}$. For any $\boldsymbol{X}$ with d_{in}^l columns,

$$\big\|\boldsymbol{X}(\boldsymbol{I} - \boldsymbol{P}^l)\big\|_F^2 = \operatorname{tr}\big(\boldsymbol{X}\boldsymbol{U}_\perp^l (\boldsymbol{U}_\perp^l)^\top \boldsymbol{U}_\perp^l (\boldsymbol{U}_\perp^l)^\top \boldsymbol{X}^\top\big) = \operatorname{tr}\big(\boldsymbol{X}\boldsymbol{U}_\perp^l (\boldsymbol{U}_\perp^l)^\top \boldsymbol{X}^\top\big) = \big\|\boldsymbol{X}\boldsymbol{U}_\perp^l\big\|_F^2.$$

The layer-l residual $\bar{\boldsymbol{Z}}^l \boldsymbol{U}_\perp^l$ has r rows for every l, so we can concatenate the residuals of all probed layers side by side into

$$\boldsymbol{R}(\bar{\boldsymbol{Z}}^1) := \Big[\, \bar{\boldsymbol{Z}}^1 \boldsymbol{U}_\perp^1 \;\Big|\; \bar{\boldsymbol{Z}}^2 \boldsymbol{U}_\perp^2 \;\Big|\; \cdots \;\Big|\; \bar{\boldsymbol{Z}}^{L_{\text{max}}} \boldsymbol{U}_\perp^{L_{\text{max}}} \,\Big] \in \mathbb{R}^{r \times m}, \qquad m := \sum_{l=1}^{L_{\text{max}}} \big(d_{\text{in}}^l - r\big), \tag{12}$$

where each $\bar{\boldsymbol{Z}}^l$ is computed from $\bar{\boldsymbol{Z}}^1$ by the forward pass. By the identity above, $\mathcal{L}_{\text{span}}(\bar{\boldsymbol{Z}}^1) = \|\boldsymbol{R}(\bar{\boldsymbol{Z}}^1)\|_F^2$. The loss is therefore nonnegative, and it reaches its minimum value zero exactly when every entry of $\boldsymbol{R}$ vanishes, that is, when the dummy hidden states satisfy the span constraints $\bar{\boldsymbol{Z}}^l \boldsymbol{U}_\perp^l = 0$ at every probed layer simultaneously. By part (a), $\boldsymbol{R}(\boldsymbol{Z}^1) = 0$.

The system $\boldsymbol{R}(\bar{\boldsymbol{Z}}^1) = 0$ has $r\, d_{\text{in}}^1$ unknowns, the entries of $\bar{\boldsymbol{Z}}^1$, and $r\, m$ scalar equations. Layer l contributes $r(d_{\text{in}}^l - r)$ of them: each of the r dummy inputs must be orthogonal to the $d_{\text{in}}^l - r$ directions outside $\mathcal{S}^l$.

Let $\boldsymbol{J}$ denote the Jacobian of $\boldsymbol{R}$ with respect to $\bar{\boldsymbol{Z}}^1$ at $\boldsymbol{Z}^1$, viewed as a linear map that sends a perturbation $\delta\bar{\boldsymbol{Z}}^1 \in \mathbb{R}^{r \times d_{\text{in}}^1}$ to $\boldsymbol{J}\,\delta\bar{\boldsymbol{Z}}^1 \in \mathbb{R}^{r \times m}$. Since the blocks are continuously differentiable and $\boldsymbol{R}(\boldsymbol{Z}^1) = 0$, a first-order Taylor expansion gives

$$\boldsymbol{R}(\boldsymbol{Z}^1 + \delta\bar{\boldsymbol{Z}}^1) = \boldsymbol{J}\,\delta\bar{\boldsymbol{Z}}^1 + o\big(\|\delta\bar{\boldsymbol{Z}}^1\|_F\big).$$

We show that $\boldsymbol{Z}^1$ is stable if and only if $\boldsymbol{J}$ is injective, that is, $\boldsymbol{J}\,\delta\bar{\boldsymbol{Z}}^1 \neq 0$ for every $\delta\bar{\boldsymbol{Z}}^1 \neq 0$.

Suppose $\boldsymbol{J}$ is injective. The unit sphere is compact, so $\sigma := \min_{\|\delta\bar{\boldsymbol{Z}}^1\|_F = 1} \|\boldsymbol{J}\,\delta\bar{\boldsymbol{Z}}^1\|_F$ is positive, and by linearity $\|\boldsymbol{J}\,\delta\bar{\boldsymbol{Z}}^1\|_F \geq \sigma\|\delta\bar{\boldsymbol{Z}}^1\|_F$ for every perturbation. Choose $\varepsilon > 0$ such that the ball of radius ε around $\boldsymbol{Z}^1$ lies in the domain of $\mathcal{L}_{\text{span}}$ and the Taylor remainder is at most $\frac{\sigma}{2}\|\delta\bar{\boldsymbol{Z}}^1\|_F$ on it. For $\|\delta\bar{\boldsymbol{Z}}^1\|_F < \varepsilon$, the triangle inequality then gives

$$\big\|\boldsymbol{R}(\boldsymbol{Z}^1 + \delta\bar{\boldsymbol{Z}}^1)\big\|_F \geq \sigma\|\delta\bar{\boldsymbol{Z}}^1\|_F - \tfrac{\sigma}{2}\|\delta\bar{\boldsymbol{Z}}^1\|_F = \tfrac{\sigma}{2}\|\delta\bar{\boldsymbol{Z}}^1\|_F.$$

Squaring yields $\mathcal{L}_{\text{span}}(\boldsymbol{Z}^1 + \delta\bar{\boldsymbol{Z}}^1) \geq \frac{\sigma^2}{4}\|\delta\bar{\boldsymbol{Z}}^1\|_F^2$, which is stability with $c = \sigma^2/4$.

Conversely, suppose $\boldsymbol{J}\,\delta\bar{\boldsymbol{Z}}^1 = 0$ for some $\delta\bar{\boldsymbol{Z}}^1 \neq 0$. Along the line $\boldsymbol{Z}^1 + t\,\delta\bar{\boldsymbol{Z}}^1$, the expansion gives $\boldsymbol{R}(\boldsymbol{Z}^1 + t\,\delta\bar{\boldsymbol{Z}}^1) = o(t)$, so $\mathcal{L}_{\text{span}}(\boldsymbol{Z}^1 + t\,\delta\bar{\boldsymbol{Z}}^1) = o(t^2)$. This is incompatible with the lower bound $c\,t^2\|\delta\bar{\boldsymbol{Z}}^1\|_F^2$ for small t, so $\boldsymbol{Z}^1$ is not stable.

When the blocks are twice differentiable, the same condition can be read off the Hessian. Differentiating $\mathcal{L}_{\text{span}} = \|\boldsymbol{R}\|_F^2$ twice produces a term with the second derivatives of $\boldsymbol{R}$, which is multiplied by $\boldsymbol{R}(\boldsymbol{Z}^1) = 0$ and so vanishes, and the term $2\boldsymbol{J}^\top\boldsymbol{J}$. The Hessian at $\boldsymbol{Z}^1$ is therefore $2\boldsymbol{J}^\top\boldsymbol{J}$, which is positive definite exactly when $\boldsymbol{J}$ is injective.

A linear map can be injective only if its domain has at most the dimension of its codomain. $\boldsymbol{J}$ maps a space of dimension $r\,d_{\text{in}}^1$ into one of dimension $r\,m$, so stability requires

$$r\sum_{l=1}^{L_{\max}}\bigl(d_{\text{in}}^l - r\bigr) \geq r\,d_{\text{in}}^1.$$

The $l = 1$ term on the left equals $r\,d_{\text{in}}^1 - r^2$. Subtracting it from both sides and dividing by r gives Equation (11). The first probed layer thus accounts for every perturbation that leaves $\mathcal{S}^1$: its block $\delta\bar{\boldsymbol{Z}}^1 \mapsto \delta\bar{\boldsymbol{Z}}^1\boldsymbol{U}_\perp^1$ vanishes exactly on perturbations whose rows stay in $\mathcal{S}^1$. These form a space of dimension r^2, and only the later layers, with their $r\sum_{l=2}^{L_{\max}}(d_{\text{in}}^l - r)$ equations, can detect them. □

In words, the first probed layer confines each dummy input to the r-dimensional span $\mathcal{S}^1$, and each later probed layer imposes $d_{\text{in}}^l - r$ further constraints on it, which together must at least match its r remaining unknowns. With a single probed layer ($L_{\max} = 1$), every dummy batch with rows in $\mathcal{S}^1$ is a global minimizer: one layer's gradient reveals the span of the inputs but nothing within it. Equation (11) is necessary but not sufficient, and it does not address (I2). When it fails, the true batch is not a stable minimizer: along some direction, the loss grows more slowly than quadratically, or not at all. Property (I2) also does not require a global minimizer to contain every true input. Minimizers that repeat some true inputs and omit others are what the backward loss of Equation (6) penalizes.

F.3 A SUFFICIENT CONDITION FOR GENERIC MLP NETWORKS

Theorem F.1 shows that the non-strict inequality is necessary for quadratic local stability. We now show that, for generic multilayer perceptrons (MLPs) satisfying the assumptions below, the corresponding strict inequality is sufficient for identifiability. This is a statement about identifiability, rather than about the convergence or numerical stability of the attack.

We consider an MLP of width d with elementwise activation $\sigma : \mathbb{R} \to \mathbb{R}$ and untied, invertible weights $\boldsymbol{W}^l \in \mathbb{R}^{d\times d}$. The probed layers are its fully connected layers, so that

$$\boldsymbol{Y}^l = \boldsymbol{Z}^l\boldsymbol{W}^l, \qquad \boldsymbol{Z}^{l+1} = \sigma(\boldsymbol{Y}^l), \qquad l = 1, \ldots, L_{\max} - 1.$$

Each row is therefore propagated independently through the same maps. We retain the exact-span assumption from the preceding setup: $\mathcal{S}^l = \text{rowspan}(\boldsymbol{Z}^l)$. An *instance* is a particular choice of the true input batch $\boldsymbol{Z}^1$ and the network weights $\boldsymbol{W}^1, \ldots, \boldsymbol{W}^{L_{\max}-1}$. We write $\theta = (\boldsymbol{Z}^1, \boldsymbol{W}^1, \ldots, \boldsymbol{W}^{L_{\max}-1})$ for one such choice, and call the set of all choices satisfying the invertibility and rank assumptions the *instance space*, denoted by Θ:

$$\Theta := \left\{ (\boldsymbol{Z}^1, \boldsymbol{W}^1, \ldots, \boldsymbol{W}^{L_{\max}-1}) : \begin{array}{l} \boldsymbol{Z}^1 \in \mathbb{R}^{r\times d}, \\ \boldsymbol{W}^l \text{ invertible for } l = 1, \ldots, L_{\max} - 1, \\ \text{rank}(\boldsymbol{Z}^l) = r \text{ for } l = 1, \ldots, L_{\max} \end{array} \right\}.$$

The later hidden states $\boldsymbol{Z}^2, \ldots, \boldsymbol{Z}^{L_{\max}}$ are determined by this choice through the forward recursion above; they are not additional variables in θ. Here the rank conditions concern the true batch, not the optimized batch. The final probed weight $\boldsymbol{W}^{L_{\max}}$ is needed to observe $\mathcal{S}^{L_{\max}}$, but its forward computation does not enter $\mathcal{L}_{\text{span}}$.

By *generic* instances, we mean all instances outside a definable exceptional subset of Θ of strictly smaller dimension. In particular, the exceptional instances have Lebesgue measure zero in this joint space of batches and weights.

Theorem F.2 (Generic identifiability for MLPs)**.** *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be injective, real analytic, and definable in an o-minimal expansion of the real field. Assume that $\sigma(0) \neq 0$ and $\sigma'' \not\equiv 0$. If*

$$(L_{max} - 1)(d - r) > r, \tag{13}$$

then the attack is identifiable for generic instances in Θ: every row of every global minimizer of $\mathcal{L}_{\text{span}}$ is a row of the true batch $\boldsymbol{Z}^1$, possibly with repetitions.

Proof. The true batch has zero loss by Theorem F.1. Since the loss is nonnegative and the MLP acts independently on each row, it remains to show that, generically, a single row can satisfy all observed span constraints only by being a true input.

Step 1: Express a candidate using the true hidden states. Because $\boldsymbol{Z}^l$ has full row rank, every vector in $\mathcal{S}^l$ has a unique representation $u_l \boldsymbol{Z}^l$ with $u_l \in \mathbb{R}^{1\times r}$. Let $\boldsymbol{\varepsilon}_j$ be the j-th standard basis row vector and $E_r := \{\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_r\}$. A candidate at the first layer is a true input exactly when $u_1 \in E_r$.

A candidate $u_l \boldsymbol{Z}^l$ propagates into the next observed span if and only if there is a unique u_{l+1} such that

$$\sigma(u_l \boldsymbol{Y}^l) = u_{l+1}\sigma(\boldsymbol{Y}^l), \qquad l = 1, \dots, L_{\max} - 1. \tag{14}$$

Moreover, if $u_{l+1} = \boldsymbol{\varepsilon}_j$, injectivity of σ gives $u_l \boldsymbol{Y}^l = \boldsymbol{\varepsilon}_j \boldsymbol{Y}^l$. Since $\boldsymbol{Y}^l$ has full row rank, this implies $u_l = \boldsymbol{\varepsilon}_j$. Thus a spurious candidate, with $u_1 \notin E_r$, must have $u_l \notin E_r$ at every layer.

Step 2: A spurious candidate imposes a genuine constraint. For fixed $u, v \in \mathbb{R}^{1\times r}$, define

$$\rho_{u,v}(y) := \sigma(uy) - \sum_{j=1}^{r} v_j \sigma(y_j), \qquad y \in \mathbb{R}^r.$$

We show that this function vanishes identically only when $u = v = \boldsymbol{\varepsilon}_j$ for some j.

Suppose $\rho_{u,v} \equiv 0$. For $i \neq j$, its mixed second derivative is

$$\partial_i \partial_j \rho_{u,v}(y) = u_i u_j \sigma''(uy).$$

If two entries of u were nonzero, then uy would range over all of $\mathbb{R}$, forcing $\sigma'' \equiv 0$. Hence u has at most one nonzero entry. If $u = 0$, differentiation with respect to each y_i forces $v_i = 0$, because σ is nonconstant. This would give $\rho_{u,v} = \sigma(0) \neq 0$, a contradiction.

Therefore $u = a\boldsymbol{\varepsilon}_j$ for some $a \neq 0$. Differentiation with respect to y_i for $i \neq j$ gives $v_i = 0$. Evaluating at $y = 0$ gives $v_j = 1$, and the remaining identity $\sigma(ay_j) = \sigma(y_j)$ gives $a = 1$ by injectivity. Conversely, $u = v = \boldsymbol{\varepsilon}_j$ clearly makes $\rho_{u,v}$ identically zero.

Consequently, whenever $u \notin E_r$, $\rho_{u,v}$ is a nonzero real-analytic function. Its zero set has empty interior by the real-analytic identity theorem. Since this zero set is definable, its dimension is at most $r - 1$ (Coste, 2000, Prop. 3.17(4)). For a matrix $\boldsymbol{Y} \in \mathbb{R}^{r\times d}$, the condition $\sigma(u\boldsymbol{Y}) = v\sigma(\boldsymbol{Y})$ applies separately to each column. Thus the set of such matrices has dimension at most $d(r - 1)$ (Coste, 2000, Prop. 3.17(5)).

Step 3: Count all possible spurious candidates simultaneously. We first use the preactivation matrices as coordinates for an instance. Let $\mathcal{P}$ be the set of tuples

$$(\boldsymbol{Z}^1, \boldsymbol{Y}^1, \dots, \boldsymbol{Y}^{L_{\max}-1})$$

for which $\boldsymbol{Z}^1$, every $\boldsymbol{Y}^l$, and every $\sigma(\boldsymbol{Y}^l)$ have rank r. This is an open definable set of dimension

$$\dim \mathcal{P} = L_{\max} dr.$$

It is nonempty: choose any full-row-rank $\boldsymbol{Z}^1$ and, for each $l = 1, \dots, L_{\max} - 1$, let

$$\boldsymbol{Y}^l = t[\boldsymbol{I}_r \ \boldsymbol{0}]$$

for sufficiently small $t \neq 0$. Then each $\boldsymbol{Y}^l$ has rank r. Moreover, the first r columns of $\sigma(\boldsymbol{Y}^l)$ form

$$(\sigma(t) - \sigma(0))\boldsymbol{I}_r + \sigma(0)\boldsymbol{1}\boldsymbol{1}^\top,$$

which is invertible for sufficiently small nonzero t. Hence $\sigma(\boldsymbol{Y}^l)$ also has rank r. Step 4 below shows that every tuple in $\mathcal{P}$ is realizable by invertible weights.

Let $\mathcal{I}$ contain an instance in $\mathcal{P}$ together with coefficients $u_1, \ldots, u_{L_{\max}}$ satisfying Equation (14) and $u_1 \notin E_r$. This set records every instance together with every possible spurious candidate. For fixed coefficients, the initial batch has dr free entries, and Step 2 bounds the dimension available to each preactivation matrix by $d(r-1)$. Allowing the $L_{\max} r$ coefficients to vary, the definable fiber-dimension theorem (Coste, 2000, Thm. 3.18) gives

$$\begin{aligned}\dim \mathcal{I} &\le dr + L_{\max} r + (L_{\max} - 1) d(r-1) \\ &= \dim \mathcal{P} - \delta, \qquad \delta := (L_{\max} - 1)(d-r) - r.\end{aligned}$$

Projecting $\mathcal{I}$ onto its instance coordinates gives precisely the set $\mathcal{B}_{\mathcal{P}}$ of instances that admit a spurious candidate. A definable projection cannot increase dimension (Coste, 2000, Prop. 3.17(2)), so

$$\dim \mathcal{B}_{\mathcal{P}} \le \dim \mathcal{P} - \delta.$$

Step 4: Transfer the count back to the network weights. For fixed full-row-rank matrices $\boldsymbol{Z}^l$ and $\boldsymbol{Y}^l$, the solutions of

$$\boldsymbol{Z}^l \boldsymbol{W}^l = \boldsymbol{Y}^l$$

form an affine space of dimension $d(d-r)$. An invertible solution exists: extend the rows of each matrix to a basis of $\mathbb{R}^d$ and use the invertible linear map sending the first basis to the second. The invertible solutions form a nonempty open subset of this affine space.

It follows that the definable map $F : \Theta \to \mathcal{P}$ sending a batch and its weights to the corresponding preactivations is surjective, and every fiber has dimension

$$q = (L_{\max} - 1) d(d-r).$$

The exceptional set in the original coordinates is $\mathcal{B}_\Theta = F^{-1}(\mathcal{B}_{\mathcal{P}})$. Applying the fiber-dimension theorem again gives

$$\dim \mathcal{B}_\Theta \le \dim \mathcal{B}_{\mathcal{P}} + q \le \dim \Theta - \delta.$$

Under Equation (13), $\delta > 0$, so this exceptional set has strictly smaller dimension than Θ. Its relative closure also has smaller dimension (Coste, 2000, Thm. 3.22); consequently, identifiability holds on a dense open subset of Θ. Outside the exceptional set, every zero-loss row is a true input, which proves the theorem. □

Examples and interpretation. The logistic sigmoid $\sigma(t) = (1 + e^{-t})^{-1}$ and the exact softplus $\sigma(t) = \log(1 + e^t)$ are injective, real analytic, non-affine, and nonzero at the origin. Both are definable in the real exponential field, which is o-minimal by Wilkie's theorem (Wilkie, 1996). They therefore satisfy the activation assumptions.

The threshold has a simple interpretation. The first observed span leaves r coefficients for a candidate input. Each further layer requires matching d activation coordinates, while introducing r coefficients for the next span: a net count of $d - r$ additional conditions. The proof makes this count rigorous by showing that any spurious candidate requires an exceptional choice of the instance once the total exceeds r. The conclusion excludes incorrect recovered rows, but still permits omitting or repeating true rows. It does not by itself establish quadratic local stability, robustness to noise, or convergence of the optimization algorithm.

F.4 Recovarability for transformers

We now instantiate the two results for the transformer attacks of §4. Theorem F.1 applies directly, because it places no restriction on the blocks.

Corollary 4.1 (Transformers)**.** *Let a transformer of width d be attacked on $T < d$ non-padding tokens through the query gradients of its first L_{max} blocks, with $\operatorname{rank}(\boldsymbol{Z}^l) = \operatorname{rank}(\partial \mathcal{L}/\partial \boldsymbol{Q}^l) = T$ for $l \le L_{max}$. Then $\boldsymbol{Z}^1$ is a global minimizer of the span objective, and it can be a stable minimizer, around which the objective grows quadratically, only if $(L_{max} - 1)(d - T) \ge T$.*

Proof. Here $\boldsymbol{W}^l = \boldsymbol{W}_Q^l$, each Φ^l is the full map from one query-projection input to the next, $\Phi^0 = f^0$, $r = T$, and $d_{\text{in}}^l = d$ for every l. The rest follows from Theorem F.1, where Equation (11) becomes $(L_{\max} - 1)(d - T) \ge T$. □

Recovering the embeddings. The corollary concerns $\boldsymbol{Z}^1$, the input to the first probed layer, which is not the same as the actual input embeddings $\boldsymbol{Z}^0$. Every $\bar{\boldsymbol{Z}}^0$ with $f^0(\bar{\boldsymbol{Z}}^0) = \boldsymbol{Z}^1$ yields the same hidden states and hence the same observed gradients, so the embeddings are identified only up to the invariances of f^0.

Normalization. Normalization in f^0 imposes an additional constraint on each embedding. RMSNorm normalizes each embedding to a fixed standard deviation, while LayerNorm additionally shifts each embedding to have the same mean. This makes the corresponding $\bar{\boldsymbol{Z}}^0$ scale- and/or shift-invariant. Thus, the recovered embeddings can only be identified up to these invariances.

Positional embeddings. Encoders with additive absolute positional embeddings $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_T$ admit spurious global minimizers at the embedding level. Such models add positions only in f^0, and their blocks, which use bidirectional attention, are equivariant to permuting the tokens of a sequence. Permuting the rows of $\boldsymbol{Z}^1$ therefore permutes every later hidden state and leaves every observed span unchanged, so the permuted batch also attains zero loss. Mapped back through f^0, a permutation π corresponds to the dummy embeddings

$$\bar{\boldsymbol{z}}_i^0 = \boldsymbol{z}_{\pi(i)}^0 + \boldsymbol{p}_{\pi(i)} - \boldsymbol{p}_i, \qquad i = 1, \ldots, T,$$

Thus, if the position i is not correctly identified, $\bar{\boldsymbol{Z}}^0$ admits multiple valid global minima. However, for text inputs under a bidirectional-attention model, the embeddings are generally not part of the vocabulary, and may be identifiable from this additional information. Causal decoders avoid this, because the mask breaks the permutation symmetry, and so do models with rotary embeddings, which inject positions inside every attention layer.

Metrics and token matching. These ambiguities inform how we recover and score tokens. As the above limitations also affect any gradient inversion attack, for fair evaluation, we perform the following domain-dependent procedure:

- *Text*: To obtain text, we match the reconstructed embeddings to the vocabulary using cosine similarity.
- *Images*: As images are also affected by the normalization, essentially losing their absolute scale, we match the reconstructed embeddings, and measure our attack's success, using cosine similarity. However, for non-scale-invariant metrics, such as PSNR, SSIM, we additionally fit the best scale and shift per patch before computing the metric.

F.5 Interpretation of the Analysis

This condition provides an upper bound on the number of entries that can be stably recovered given the number of probed blocks. Under the transformer notation, the condition of Corollary 4.1 rearranges into the following bound:

$$(L_{\max} - 1)(d - T) \geq T \quad \Longleftrightarrow \quad (L_{\max} - 1)d \geq L_{\max} T \quad \Longleftrightarrow \quad T \leq \Big(1 - \frac{1}{L_{\max}}\Big) d.$$

This implies that running TIGER on only 2 layers allows up to $d/2$ tokens to be recovered, three up to $2d/3$, etc. Thus, probing more layers increases the number of tokens for which the true batch can be a stable minimizer.

G Backward Span-Distance Loss with Unobserved Tokens

For images, TIGER optimizes the image tokens Ω while holding the recovered prefix $\mathcal{K}$ fixed. The forward pass omits unknown tokens, including the text following the image, but the backward loss measures coverage of the entire observed subspace $\mathcal{S}^l$. We show that, for an exact projection and fixed recovered rank, these omissions raise the minimum loss without preventing the true partial hidden states from attaining it.

Setup In addition to the paper's notation, we label all positions of the client batch that are not simulated by $\mathcal{R}$, so that the client tokens at each attacked layer l split into $\boldsymbol{Z}^l_{\mathcal{K}}$, $\boldsymbol{Z}^l_{\Omega}$, and $\boldsymbol{Z}^l_{\mathcal{R}}$, with $T = |\mathcal{K}| + |\Omega| + |\mathcal{R}|$. For a single sequence, $\mathcal{R}$ is the text suffix after the image, while under a multi-sequence setting, it may include other suffixes, or potentially entire pending sequences.

Consider a single attacked layer l. We assume the conditions of Theorem 3.1 and $\operatorname{rank}(\boldsymbol{Z}^l) = T$, so that $\dim \mathcal{S}^l = T$. Let $\boldsymbol{U}^l \in \mathbb{R}^{T\times d}$ be its orthonormal row basis, with projection $\boldsymbol{P}^l = (\boldsymbol{U}^l)^\top \boldsymbol{U}^l$. Write $\bar{\boldsymbol{Z}} := \bar{\boldsymbol{Z}}^l_{\Omega\cup\mathcal{K}}$ for the simulated hidden states, $k := \operatorname{rank}(\bar{\boldsymbol{Z}}) \le T$ for their rank, and $\bar{\mathcal{S}}^l = \operatorname{rowspan}(\bar{\boldsymbol{Z}})$ for their span. Its orthogonal projection is $\bar{\boldsymbol{P}}^l_0 := \bar{\boldsymbol{Z}}^\dagger \bar{\boldsymbol{Z}}$, the limit of Equation (8) as $\lambda_{\text{inv}} \to 0$. For any orthonormal row basis $\bar{\boldsymbol{V}} \in \mathbb{R}^{k\times d}$ of this span, $\bar{\boldsymbol{P}}^l_0 = \bar{\boldsymbol{V}}^\top \bar{\boldsymbol{V}}$. We analyze the exact-projection loss $\ell^l_{\text{back}} := \sum_{i=1}^T \mathcal{D}(\boldsymbol{U}^l_i, \bar{\boldsymbol{P}}^l_0)^2$; the identities below need not hold for positive jitter.

Partial forward pass Under our attack, $\mathcal{R}$ is always such that no simulated position attends to an unobserved one. This holds for the attention masks we consider: $\boldsymbol{M}$ is block-diagonal across sequences, so a token never attends to another sequence, and within a sequence the prefix is causal and the image block attends only to itself and to the prefix. Under this condition, the generalization of Equation (3) from a causal prefix to the retained set $\mathcal{K}\cup\Omega$ holds at every layer:

$$f^l\left(\bar{\boldsymbol{Z}}^l_{\mathcal{K}\cup\Omega}; \boldsymbol{M}\right) = f^l\left(\bar{\boldsymbol{Z}}^l; \boldsymbol{M}\right)_{\mathcal{K}\cup\Omega}, \tag{15}$$

so the hidden states computed by the partial forward pass over $\mathcal{K}\cup\Omega$ coincide with the corresponding rows of the full client forward pass over all B sequences. Omitting $\mathcal{R}$ therefore does not change the simulated hidden states.

Proposition G.1. *For every attacked layer l,*

$$\ell^l_{back} = (T-k) + \left\|\bar{\boldsymbol{V}}\left(\boldsymbol{I} - \boldsymbol{P}^l\right)\right\|_F^2. \tag{16}$$

Proof. Since $\bar{\boldsymbol{P}}^l_0$ is symmetric and idempotent as a projection matrix, we can write $\mathcal{D}(\boldsymbol{U}^l_i, \bar{\boldsymbol{P}}^l_0)^2 = \|\boldsymbol{U}^l_i(\boldsymbol{I} - \bar{\boldsymbol{P}}^l_0)\|_2^2$, and summing over the T rows of $\boldsymbol{U}^l$ gives

$$\left\|\boldsymbol{U}^l(\boldsymbol{I} - \bar{\boldsymbol{P}}^l_0)\right\|_F^2 = \operatorname{tr}\left(\boldsymbol{U}^l(\boldsymbol{I} - \bar{\boldsymbol{P}}^l_0)(\boldsymbol{U}^l)^\top\right) = T - \operatorname{tr}\left(\boldsymbol{P}^l \bar{\boldsymbol{P}}^l_0\right),$$

using $\boldsymbol{U}^l(\boldsymbol{U}^l)^\top = \boldsymbol{I}_T$, $\boldsymbol{P}^l = (\boldsymbol{U}^l)^\top \boldsymbol{U}^l$, and the cyclicity of the trace. By the same argument applied to $\bar{\boldsymbol{V}}$, which satisfies $\bar{\boldsymbol{V}}\bar{\boldsymbol{V}}^\top = \boldsymbol{I}_k$ and $\bar{\boldsymbol{V}}^\top\bar{\boldsymbol{V}} = \bar{\boldsymbol{P}}^l_0$,

$$\left\|\bar{\boldsymbol{V}}(\boldsymbol{I} - \boldsymbol{P}^l)\right\|_F^2 = k - \operatorname{tr}\left(\bar{\boldsymbol{P}}^l_0 \boldsymbol{P}^l\right).$$

Subtracting the two identities yields Equation (16). □

Corollary G.2. *Over all candidates of rank k, $\ell^l_{back} \ge T - k$, with equality if and only if $\bar{\mathcal{S}}^l \subseteq \mathcal{S}^l$. In particular, if the true partial hidden states $\boldsymbol{Z}^l_{\mathcal{K}\cup\Omega}$ have rank k, they are a global minimizer among rank-k candidates. If the recovered prefix is correct and the simulated hidden states have locally constant rank, the gradient with respect to $\bar{\boldsymbol{Z}}^0_\Omega$ vanishes at the true input.*

Proof. The second term of Equation (16) is nonnegative and vanishes exactly when $\bar{\mathcal{S}}^l \subseteq \mathcal{S}^l$. By Theorem 3.1, the true partial hidden states lie in $\mathcal{S}^l$ and attain the bound. With a correct prefix, Equation (15) identifies them with the states produced by the partial forward pass at the true input. If rank is locally constant, the exact projection is differentiable in a neighborhood of that input, which is a local minimizer, so its gradient vanishes. Under our full-row-rank assumption on $\boldsymbol{Z}^l$, its retained rows are also independent, ensuring locally constant rank at the true input. □

Proposition G.1 and Corollary G.2 thus show that the backward loss with unobserved tokens is a well-defined subspace objective. Up to the constant $T - k$, ℓ^l_{back} is the forward span distance of Equation (2) applied to an orthonormal basis of the recovered subspace instead of to the individual hidden states. It therefore depends only on the two subspaces $\bar{\mathcal{S}}^l$ and $\mathcal{S}^l$, not on the particular rows spanning them. For fixed rank, the true partial hidden states attain the loss floor despite the unobserved tokens. In particular, the objective does not push the image tokens to also represent the unobserved tokens, whether these belong to the same sequence or to other sequences of the batch, since no rank-k candidate can reduce the loss below $T - k$.

Connection to subspace distances The backward loss has a geometric interpretation that explains why a correct partial reconstruction need not have zero loss. Let $\theta_1, \ldots, \theta_k \in [0, \pi/2]$ be the principal angles between $\bar{\mathcal{S}}^l$ and $\mathcal{S}^l$. Since $k \leq T$, Equation (16) becomes

$$\ell^l_{\text{back}} = (T - k) + \sum_{j=1}^{k} \sin^2 \theta_j = d_{\text{ch}}\big(\bar{\mathcal{S}}^l, \mathcal{S}^l\big)^2,$$

where d_{ch} is the extension of the chordal metric to subspaces of different dimensions described by Ye & Lim (2016). This metric treats the $T - k$ missing dimensions as additional principal angles of $\pi/2$, each contributing the largest possible penalty of one. The remaining terms measure the misalignment of the recovered directions, with $\sin^2 \theta_j$ ranging from zero for an aligned principal direction to one for an orthogonal one. For fixed k, minimizing the loss therefore amounts to aligning the recovered subspace with the gradient subspace. At the true partial hidden states, all k principal angles vanish because $\bar{\mathcal{S}}^l \subseteq \mathcal{S}^l$, leaving only the dimension deficit $T - k$, as stated in Corollary G.2.

Implications When $\mathcal{S}^l = \text{rowspan}(\boldsymbol{Z}^l)$ and $\bar{\boldsymbol{P}}^l_{0,\star}$ denotes the projection onto the span of the true partial hidden states, the floor equals $T - k = \text{rank}\big(\boldsymbol{Z}^l_{\mathcal{R}}(\boldsymbol{I} - \bar{\boldsymbol{P}}^l_{0,\star})\big)$, the number of directions that the unobserved tokens add beyond those already spanned by the simulated ones. It is at most $|\mathcal{R}|$, and it does not require the image and text hidden states to be orthogonal. Moving an already recovered sequence from $\mathcal{R}$ into $\mathcal{K}$ lowers the floor by the number of new directions that sequence contributes, while the true image remains a minimizer under the conditions of Corollary G.2. Thus, either sequential or joint image recovery preserves the true input as a minimizer under these assumptions.

Corollary G.2 compares candidates of equal rank. A candidate spanning $k + 1$ directions inside $\mathcal{S}^l$ attains a strictly lower floor, so the backward loss favors broader coverage and discourages repeated or degenerate solutions. If the true partial hidden states have maximal feasible rank, as they do under our full-row-rank assumption, they minimize the loss across all candidate ranks.

H Qualitative Examples

This section shows client batches and the corresponding TIGER reconstructions for two failure types.

H.1 Collapse onto One Sequence

In the example below, the primary reason TIGER fails is due to its inability to recover the first tokens in the separate sequences. The following sequences actually become correct after the fact, but score poorly due to being duplicates.

H.2 Token Repetition

On this example, instead, the reconstructed sequences exhibit significant token repetition. This most often happens when the positional encodings are insufficient to distinguish the embedding, and the token itself is not well attended by its preceding context, and hence the attack finds a solution that repeats tokens excessively.

I Limitations and Future Work

While TIGER demonstrates an effective and robust approach to gradient inversion for transformer LLMs, several limitations suggest natural directions for future work.

First, TIGER builds on the low-rank structure of linear-layer gradients (Petrov et al., 2024; Drencheva et al., 2025; Dimitrov et al., 2024; Bakarsky et al., 2025). This structure is most informative when the number of token representations contributing to a gradient remains smaller than the relevant hidden dimension. As the total number of tokens T approaches or exceeds the hidden dimension d, the corresponding gradient subspace becomes full-rank, reducing or eliminating the discriminative power of the subspace objective.

Second, although our results show that TIGER remains effective under additive DP-style noise, our evaluation does not exhaust the space of possible privacy-preserving defenses. In particular,

Table 11: GEMMA-3-4B-IT, Rotten Tomatoes, B=8, attack 0.

Client sequences
sharp edges and a deep vein of sadness run through its otherwise comic narrative .
this is an extraordinary film , not least because it is japanese and yet feels universal .
so brisk is wang's pacing that none of the excellent cast are given air to breathe .
though it pretends to expose the life of male hustlers , it's exploitive without being insightful .
'[the cockettes] provides a window into a subculture hell-bent on expressing itself in every way imaginable . '
secretary manages a neat trick , bundling the flowers of perversity , comedy and romance into a strangely tempting bouquet of a movie .
can't kick about the assembled talent and the russos show genuine promise as comic filmmakers . still , this thing feels flimsy and ephemeral .
the lightest , most breezy movie steven spielberg has made in more than a decade . and the positive change in tone here seems to have recharged him .
Reconstructed sequences
secretary manages a neat trick , bundling the flowers of perversity , comedy and romance into a strangely tempting bouquet of a movie .
can't kick about
so brisk is wang's pacing that none of the excellent cast are given air to breathe .
though it pretends to expose the life of male hustlers , it
secretary manages a neat trick , bundling the flowers of perversity , comedy and romance into a strangely tempting bouquet of a movie .
secretary manages a neat trick , bundling the flowers of perversity , comedy and romance into a strangely tempting bouquet of a movie .
secretary manages a neat trick , bundling the flowers of perversity , comedy and romance into a strangely tempting bouquet of a movie .
sharp edges and a deep vein of sadness run through its otherwise comic narrative .

Table 12: GEMMA-3-4B-IT, WikiText, B=4, σ=5×10^{-5}, random initialization, attack 1 (Table 8).

Client sequences
After the Spanish conquest of the Aztec Empire , the conquistadors decided to build
Clarkson had also promoted Wrapped in Red in various televised performances , all of which
Markgraf displaced 25 @,@ 796 t (
Two weeks prior to the NHL trade deadline , Columbus announced that unlike earlier in
Reconstructed sequences
Markgraf displaced 25 @ Mark Mark Mark weeks Mark Mark Mark weeks
After Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark
Clarkson Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark
Two Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark Mark

different clipping strategies (Geyer et al., 2017; Andrew et al., 2021; Xia et al., 2023), gradient masking (Konečný et al., 2016), secure aggregation (Bonawitz et al., 2017), compression (Lin et al., 2018), or combinations of these mechanisms may affect the available subspace signal in different ways. A more complete characterization of how such defenses interact with TIGER would be valuable, both for understanding the limits of the reconstruction attack and for designing stronger defenses.

Finally, TIGER focuses on recovering token-level information through continuous embedding optimization and therefore does not always produce an exact ordered reconstruction of the original batch. In many cases, however, most recovered sequences contain substantial portions of the original content, leaving room for hybrid attacks that combine TIGER's continuous subspace recovery with discrete refinement procedures. For example, a language-model prior could be used to reorder recovered tokens or infer missing ones, similar to the refinement used in LAMP (Balunovic et al., 2022). We view such reconstruction pipelines as a promising direction for future work.

J PII EVALUATION PROMPTS

We provide the prompts and response schemas used to evaluate text readability, face recognizability, and face occlusion with GEMINI-3.1-PRO. We use the default temperature and thinking level (medium) for all calls. Each call includes two copies of a single image: the untouched image and a copy with the target region outlined in magenta. The marked copy locates the region, while the judge reads or assesses the untouched copy. Ground-truth and reconstructed images are evaluated in separate calls with identical prompt text, and only the region coordinates and, for text, its class are shared with the reconstruction call.

Shared placeholders Braced names in the templates are filled in for each call. `{W}` and `{H}` give the image dimensions in pixels, `{box_2d}` contains $[y_{\min}, x_{\min}, y_{\max}, x_{\max}]$ on a 0–1000 scale, with the origin at the top-left. The placeholders `{x0}`, `{x1}`, `{y0}`, and `{y1}` give the same box in pixels. For text, this is the enclosing box of the oriented region, while for faces, it is the annotated axis-aligned box.

J.1 TEXT READING

The text-reading prompt is applied once per text region to the ground truth at the reconstruction resolution, a source-resolution reference, and the reconstruction. The source-resolution reference is capped at 2048 pixels. Readings are scored separately after the calls. `{corners}` gives the oriented rectangle's four corners in pixels, in boundary order, and `{what}` describes the annotated class, as listed below.

Text-Reading Prompt

```
You are reading text in an image that may be heavily degraded: blurred,
blocky, with misplaced or discoloured square tiles. Read ONLY the text inside
one marked region.

You are given TWO copies of the same image. The FIRST is untouched. The
SECOND is identical except that the region is outlined by a thin magenta
rectangle drawn just outside it, rotated with the region when the region is
rotated. Use the second copy only to locate the region; read the characters
from the first copy, where nothing covers them.

The region is a rectangle that may be rotated. Its four corners, in pixels of
this {W}x{H} image as (x, y) with (0, 0) the top-left, going round the
rectangle, are:

    {corners}

{tilt}

It lies within box_2d = {box_2d} ([ymin, xmin, ymax, xmax] on a 0-1000 scale,
0,0 the top-left of the image, 1000,1000 the bottom-right; in pixels x {x0}..{x1},
y {y0}..{y1}), but only the rotated rectangle itself is the region.

The region marks one piece of PRIVATE INFORMATION: a name, an ID or card
number, a date, an address, a phone number, an e-mail, a licence plate, a shop
or street name, or similar. Read that value only:
  - Do NOT include printed field labels or captions such as "Name", "DOCUMENT
    No", "DL", "EXP", "Address:", "Tel", "手机", "姓名" -- only the value they
    introduce. If the region holds nothing but a label, read nothing.
  - Read only characters that lie INSIDE the region. A character cut by the
    region's edge counts; a word or line lying outside it does not, even if it
    looks like it belongs to the same field. Do not continue onto neighbouring
    lines.

This region is annotated as holding {what}.

What counts as inside the region -- apply these tests exactly:
  - A LINE of text is inside if at least half of its height lies within the
    region. A line the region's edge only grazes (less than half its height
    inside) is outside: leave it out entirely.
  - Within an included line, a CHARACTER counts if at least half of its width
    lies within the region.
  - The region may touch the edge of the image. Text cut off by the image
    border is read only as far as it is visible; do not complete it.
  - Put each included line on its own line in best_reading, top to bottom.
```

```
Give ONE reading: your best possible interpretation of the text in the
region, and no alternatives. Guessing is allowed and wanted: use what you can
see of letter shapes, lengths and layout. Where a character cannot be seen at
all, write one underscore "_" for it; if you cannot even tell how many
characters are missing, write "[...]". Do not invent text for parts that are
not visible, but do give your best guess wherever there is any visual evidence.

BLANKS -- use exactly this notation for characters you cannot see, and nothing else:
  _      exactly ONE underscore per unseen character. Several unseen
         characters in a row are written together with no spaces ("AB__7"),
         unless a real space is printed there.
  [...]  an unknown NUMBER of unseen characters: an opening square bracket,
         three ASCII full stops, a closing square bracket.
Never use "?", "*", "#", "X", "...", "[...]", "■" or words such as "(illegible)"
for unseen characters.

legibility is one of:
  "none"    - no character in the region can be made out, or there is no text
  "partial" - some characters can be read, some cannot
  "full"    - every character can be read
```

Text-Reading Response Schema

```
{
  "type": "object",
  "properties": {
    "legibility": {"type": "string", "enum": ["none", "partial", "full"]},
    "best_reading": {"type": "string"}
  },
  "required": ["legibility", "best_reading"]
}
```

Orientation placeholder For rectangles rotated by less than $0.5°$ from the image axes, `{tilt}` is replaced by the first template below. Otherwise, the second template is used: N is the angle folded into $(-45°, 45°]$, with positive angles clockwise on screen, and the appropriate direction and long-side orientation are selected.

Orientation: Axis-Aligned Region

```
Its edges are parallel to the image edges: it is not rotated.
```

Orientation: Rotated Region

```
It is rotated N degrees clockwise|counter-clockwise from the image axes, and its long side runs
closer to vertical|horizontal: the text in it is turned with it, so its lines run along the
rectangle's edges, not along the image rows.
```

Text-region classes The following descriptions fill `{what}`. Faces (class 0) use the face-rating prompt. Whole-document classes—chat (1), history (8), ID (10), passport (11), driving licence (12), bank card (13), receipt (14), invoice (15), express order (16), flight pass (17), and train ticket (18)—are excluded from text reading because their individual fields have separate boxes.

ID	Class	Description inserted into the prompt
2	`email`	an e-mail address
3	`password`	a password
4	`account`	an account name or number
5	`address`	a street or postal address
6	`verify_code`	a verification code
7	`identification_number`	an identification number: an ID, passport, licence, card, order or account number, or a date printed as one
9	`name`	a person's name
19	`chinese_car_plate`	a Chinese vehicle registration plate
20	`non-chinese_car_plate`	a vehicle registration plate

ID	Class	Description inserted into the prompt
21	`electric_bike_plate`	an electric-bike registration plate
22	`store_name`	a shop or business name
23	`school_name`	a school name
24	`community_name`	a residential community name
25	`street_name`	a street name or house number
26	`phone`	a phone (or fax) number

J.2 Face Recognizability

Each face is rated five times in the ground-truth image at the reconstruction resolution and five times in the reconstruction. Ratings map to 1 (`recognizable`), 0.5 (`partial`), and 0 (`unrecognizable`); each image's rating is the median of its five votes. Faces rated 0 in the ground truth or more than 40% occluded under the procedure below are excluded. For each remaining face, recovery scores 1 if its rating is at least the ground-truth rating, 0.5 if its rating is 0.5 and the ground-truth rating is 1, and 0 otherwise. Results are grouped by grid size and face size, with small faces defined by a square root of box area below 32 pixels.

Face-Recognizability Prompt

```
You are judging a face in an image that may be heavily degraded: blurred,
blocky, with misplaced or discoloured square tiles. Judge ONLY the face inside
one marked region.

You are given TWO copies of the same image. The FIRST is untouched. The
SECOND is identical except that the region is outlined by a thin magenta
rectangle drawn just outside it. Use the second copy only to locate the
region; judge the face from the first copy, where nothing covers it.

The region is given as box_2d = [ymin, xmin, ymax, xmax] on a 0-1000 scale
(0,0 is the top-left of the image, 1000,1000 the bottom-right):

    box_2d = {box_2d}

which is, in pixels of this {W}x{H} image: x {x0}..{x1}, y {y0}..{y1}.

The region is annotated as holding a human face. Ignore any other face in the
image, even one next to this region.

First, list the facial features you can actually make out in the region, in
this image (visible_features). Count a feature only if you can see it as that
feature -- not because a face would have one there. If you can make out none,
give ["none"].

Then rate how recognizable the person is from this image alone:

  "recognizable"    The eyes, nose and mouth can all be made out clearly, with
                    enough detail that someone who knows this person could
                    identify them from this image.
  "partial"         A face can be made out -- at least the arrangement of the
                    eyes, nose and mouth, or a distinctive trait such as the
                    hairline, glasses or facial hair -- but not enough to
                    identify the person on its own. Given another picture of
                    this person, you could tell whether it is the same person.
  "unrecognizable"  No eyes, nose or mouth can be made out: the region shows a
                    blur or blob of skin tone, a head shape only, scrambled or
                    wrong tiles, or no face at all. Nothing in it would let you
                    tell this person apart from another of similar build.

Judge the face as it appears in this image; do not assume detail that is not
visible.
```

Face-Recognizability Response Schema

```
{
  "type": "object",
  "properties": {
    "visible_features": {
      "type": "array",
      "items": {
```

```
          "type": "string",
          "enum": [
            "face outline",
            "eyes",
            "eyebrows",
            "nose",
            "mouth",
            "hair or hairline",
            "glasses",
            "facial hair",
            "other distinctive mark",
            "none"
          ]
        }
      },
      "rating": {
        "type": "string",
        "enum": [
          "recognizable",
          "partial",
          "unrecognizable"
        ]
      }
    },
    "required": [
      "visible_features",
      "rating"
    ]
  }
```

J.3 Face Occlusion

The occlusion prompt is applied only to ground-truth images at the reconstruction resolution, separately from recognizability assessment. Each face receives five votes. The reported `hidden_percent` values are clamped to $[0, 100]$ and aggregated by their median, using the lower middle value if the vote count is even. A median strictly above 40% excludes the face from scoring. `hidden_parts` provides an explanation and does not affect this rule.

Face-Occlusion Prompt

```
You are judging a face in an image that may be heavily degraded: blurred,
blocky, with misplaced or discoloured square tiles. Judge ONLY the face inside
one marked region.

You are given TWO copies of the same image. The FIRST is untouched. The
SECOND is identical except that the region is outlined by a thin magenta
rectangle drawn just outside it. Use the second copy only to locate the
region; judge the face from the first copy, where nothing covers it.

The region is given as box_2d = [ymin, xmin, ymax, xmax] on a 0-1000 scale
(0,0 is the top-left of the image, 1000,1000 the bottom-right):

    box_2d = {box_2d}

which is, in pixels of this {W}x{H} image: x {x0}..{x1}, y {y0}..{y1}.

The region is annotated as holding a human face. Ignore any other face in the
image, even one next to this region.

How much of this face is hidden? Think of the whole face as it would be if
nothing were in front of it: from the forehead to the chin and from ear to ear.
Estimate the percentage of that area you cannot see because something covers it,
or because the face runs off the edge of the image (hidden_percent, 0 to 100).

Something that hides the face means: an object, a hand, another person, a mask
or a scarf, sunglasses, a hat pulled down over the eyes, a sign or text laid
over it, or the edge of the image. It does NOT include: ordinary clear glasses,
hair that leaves the eyes, nose and mouth visible, a beard, the face being turned
to the side, or blur and low resolution -- those hide nothing, so they count as 0.

Also list the parts of the face that are hidden (hidden_parts); if none are,
give ["none"].
```

Face-Occlusion Response Schema

```
{
  "type": "object",
  "properties": {
    "hidden_percent": {
      "type": "integer"
    },
    "hidden_parts": {
      "type": "array",
      "items": {
        "type": "string",
        "enum": [
          "eyes",
          "eyebrows",
          "nose",
          "mouth",
          "chin or jaw",
          "forehead",
          "cheeks",
          "face outline",
          "none"
        ]
      }
    }
  },
  "required": [
    "hidden_percent",
    "hidden_parts"
  ]
}
```