The Gradient Bridge: Decrypting the Latent Training Signal in Low-Rank Adapters

1. Introduction: The False Promise of Ephemeral Privacy

The rapid proliferation of Large Language Models (LLMs) and diffusion-based generative models has precipitated a fundamental shift in the machine learning deployment paradigm. The era of monolithic, static model checkpoints is ceding ground to a modular ecosystem dominated by Parameter-Efficient Fine-Tuning (PEFT). Specifically, Low-Rank Adaptation (LoRA) has emerged not merely as an optimization technique but as a standard for model distribution. Platforms such as Hugging Face and Civitai host hundreds of thousands of "adapters"—lightweight matrices (A, B) that, when injected into a frozen base model (W_0), specialize it for tasks ranging from medical diagnostics to personalized artistic generation.[1]

The prevailing wisdom in the open-source and enterprise communities posits that releasing these adapters is inherently safer than releasing full model checkpoints or the raw datasets used to create them. This assumption rests on a dimensionality argument: if a base model has 70 billion parameters, and a LoRA adapter contains only 10 million (often $< 0.1\%$ of the total), the information bottleneck imposed by the low-rank constraint ($r \ll d$) acts as a cryptographic-like sieve, ostensibly filtering out the high-fidelity details of the training data while retaining only the abstract skills or styles.[1] Consequently, adapters trained on sensitive, private, or copyrighted data are routinely published under the belief that the underlying data is irretrievable.

This report formulates and rigorously analyzes a novel threat model—**The Gradient Bridge**—that fundamentally challenges this safety assumption. We advance the thesis that a LoRA adapter is not merely a functional modifier of a neural network, but a structured, compressed recording of the cumulative training gradients. By synthesizing recent theoretical breakthroughs in the implicit bias of gradient descent [4], the spectral properties of low-rank updates [5], and gradient reconstruction techniques developed for machine unlearning [7], we demonstrate that the low-rank constraint is not a barrier to data recovery, but rather a structured projection that can be inverted.

The "Gradient Bridge" attack operates by treating the released adapter weights as a low-dimensional measurement of the full-parameter gradient signal. We propose a two-stage mechanism to exploit this: first, a learned **Gradient Decoder**—trained on public proxy data—approximates the inverse projection, effectively "hallucinating" the high-dimensional full gradient from the low-rank adapter. Second, this approximated signal is fed into advanced **Gradient Inversion** pipelines (such as DAGER for text [8] or GradInversion for vision [9]) to reconstruct individual training examples.

This analysis reveals that the information density of LoRA adapters is deceptively high. Because neural network training is implicitly biased towards maximizing the margin on "support vectors"—the most difficult and distinctive training examples—the adapter weights preferentially encode the exact data points an adversary would wish to recover.[4] Far from being a privacy filter, LoRA acts as a high-pass filter for sensitive data, discarding redundant background information while preserving the critical features of the training set.

In the following sections, we construct the theoretical basis for the Gradient Bridge, detailing the mathematical equivalence between LoRA optimization and projected full fine-tuning. We then outline a concrete empirical roadmap for validating this threat across vision and text domains, addressing the complexities of aggregated updates and cross-distribution generalization. The evidence suggests that the current ecosystem of open adapter sharing sits upon a precarious foundation, vulnerable to a new class of supply-chain attacks that bridge the gap between weights and data.

---

2. Theoretical Framework: The Anatomy of a Low-Rank Update

To operationalize the reconstruction of training data from adapter weights, we must first rigorously define the relationship between the observed low-rank matrices and the latent gradients generated during training. The core of our thesis is that the LoRA update $\Delta W = BA$ is a deterministic, invertible function of the full training gradient, conditioned on the optimization trajectory.

2.1 LoRA as a Low-Rank Gradient Projection

In standard full fine-tuning, the weight update at any time step t, denoted ΔW_t, is directly proportional to the negative gradient of the loss function $\mathcal{L}$ with respect to the weights W:

$$\Delta W^{\text{full}}_t = -\eta \nabla_W \mathcal{L}(W_t; \mathcal{D})$$

where η is the learning rate and $\mathcal{D}$ is the batch of training data. This gradient $\nabla_W \mathcal{L}$ has the same dimensions as W (e.g., 4096×4096).

In the LoRA paradigm, we freeze W_0 and introduce trainable matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where $r \ll \min(d, k)$. The forward pass becomes $h = (W_0 + BA)x$. The optimization occurs solely in the (A, B) manifold. The gradients with respect to these matrices are derived via the chain rule:

$$\nabla_A \mathcal{L} = B^T \nabla_W \mathcal{L}$$

$$\nabla_B \mathcal{L} = (\nabla_W \mathcal{L}) A^T$$

The implicit update to the full weight matrix, ΔW^{LoRA}, is the change in the product BA. Ignoring second-order terms for a single step, this is:

$$\Delta W^{\text{LoRA}} \approx B(\Delta A) + (\Delta B)A$$

Substituting the gradient descent updates $\Delta A = -\eta \nabla_A \mathcal{L}$ and $\Delta B = -\eta \nabla_B \mathcal{L}$:

$$\Delta W^{\text{LoRA}} \approx -\eta$$

This equation reveals a critical insight: **The LoRA update is a specific projection of the full gradient $\nabla_W \mathcal{L}$.** Specifically, if we treat B and A as defining subspaces, the update effectively projects the full gradient onto the column space of B and the row space of A.

Recent theoretical work on **LoRA-Pro** [5] and **LoRA-GA (Gradient Approximation)** [6] significantly strengthens this observation. Wang et al. [6] demonstrate that by initializing A and B using the singular value decomposition (SVD) of the initial full gradient, the LoRA optimization trajectory can be made to align almost perfectly with the full fine-tuning trajectory in the principal subspace. Furthermore, Zhang et al. [5] prove that LoRA optimization is mathematically equivalent to full fine-tuning using a *low-rank gradient*.

This equivalence is the cornerstone of the Gradient Bridge. The adapter weights (A, B) are not arbitrary compression artifacts; they are high-fidelity measurements of the principal components of the full gradient $\nabla_W \mathcal{L}$. If the "effective rank" of the gradient information—specifically the gradients of the target data—is sufficiently low (a hypothesis supported by the "tiny subspace" theory of neural training [12]), then the projection onto the LoRA subspace is nearly lossless for the relevant information.

2.2 Implicit Bias and the "Stationary Point"

The second theoretical pillar of our framework concerns what information is encoded in the weights at the end of training. We invoke the seminal findings of **Haim et al. (NeurIPS 2022)** [4], who investigated the reconstruction of training data from trained binary classifiers.

Their work builds on the theory of **implicit bias** in gradient descent. For homogeneous neural networks (e.g., ReLUs without bias in hidden layers) trained with exponential-tailed losses (like binary cross-entropy), gradient flow does not merely minimize the loss; it converges to a specific solution that maximizes the L_2 margin.[4] Mathematically, the converged parameters θ^* satisfy a stationarity condition derived from the KKT conditions of the max-margin problem:

$$\theta^* = \sum_{i=1}^{n} \lambda_i y_i \nabla_\theta \Phi(\theta^*; x_i)$$

Here, λ_i represents the Lagrange multiplier for the i-th training sample (x_i, y_i). Crucially, due to the complementary slackness condition ($ \lambda_i (y_i \Phi(\theta^*; x_i) - \gamma) = 0 $), λ_i is non-zero *only* for samples that lie exactly on the margin (the support vectors).

Applying Implicit Bias to LoRA: While Haim et al. analyzed full MLPs, the principle of implicit bias extends to constrained optimizations like LoRA, particularly when weight decay is used (which LoRA training typically employs to prevent overfitting in the low-rank subspace).[13] If we view the LoRA update matrix $\Delta W = BA$ as the "parameter," the convergence state implies:

$$B^* A^* \approx \sum_{i \in \text{Support}} \alpha_i \nabla_W \mathcal{L}(x_i)$$

This result is profound for the attacker. It suggests that the final adapter weights are a **linear combination of the gradients of the hardest, most distinctive training examples.** The weights do not represent the average of the dataset (which might be a blurry mean); they represent the specific outliers and edge cases that defined the decision boundary.

Consequently, recovering the training data becomes a problem of **blind source separation**:

given the mixture $M = \sum \alpha_i \nabla_W \mathcal{L}(x_i)$, can we isolate the individual x_i ? The Gradient Bridge utilizes the structure of natural data gradients to solve this.

2.3 The Invertibility of Structured Projections

The central skepticism regarding LoRA data reconstruction is the dimensionality mismatch. How can one recover a 4096×4096 gradient matrix from a rank-16 adapter? This appears to violate information theoretic bounds. However, this view assumes the target signal (the gradient) is random and full-rank.

In reality, gradients of neural networks with respect to natural inputs (images, text) are extremely sparse and structured. They lie on a low-dimensional manifold determined by the data distribution. This is the premise behind **Compressed Sensing**: a sparse signal can be perfectly recovered from a small number of random linear measurements, provided the number of measurements exceeds the sparsity level (times a logarithmic factor).

We model the LoRA update as a sensing mechanism:

$$M_{LoRA} = \mathcal{A}(G_{full})$$

where $\mathcal{A}$ is the projection operator defined by the adapter training dynamics. While $\mathcal{A}$ is not a random Gaussian matrix (as in classical compressed sensing), it is a *learned* projection that adapts to capture the maximum variance of the gradient signal.[14]

The **Gradient Bridge** works by learning the inverse map $\mathcal{A}^{-1}$. We define the reconstruction error bound as:

$$||\hat{G} - G_{full}|| \leq \epsilon_{rank} + \epsilon_{decoder}$$

- ϵ_{rank} : The information truly lost because it lies in the null space of the LoRA projection.
- $\epsilon_{decoder}$: The approximation error of the learned decoder.

The existence of **Recover-to-Forget (R2F)** [15] provides empirical proof that this bound is tight enough for utility. R2F explicitly trains a "Gradient Decoder" on a proxy model to map LoRA updates back to full gradients for the purpose of unlearning. If the reconstructed gradient is accurate enough to unlearn a specific fact, it preserves the semantic content of that fact. Our thesis simply redirects this reconstructed gradient from an unlearning optimizer to a reconstruction attacker.

3. The Gradient Bridge Attack Pipeline

We propose a comprehensive attack methodology structured into distinct phases, moving from establishing the bridge (Phase 1) to executing the inversion (Phase 2).

Phase 1: The Gradient Decoder (Building the Bridge)

The critical innovation of our approach is the **Gradient Decoder**—a neural network trained to reverse the LoRA compression. This transforms the problem from an analytic inversion (which is ill-posed) to a learning problem (which leverages data priors).

3.1.1 Proxy Data Strategy

The adversary does not have access to the victim's private dataset $\mathcal{D}_{priv}$. However, they have access to public data $\mathcal{D}_{proxy}$ from the same modality (e.g., The Pile for text, LAION for images). The R2F paper demonstrates that gradient structures are highly transferrable; a decoder learned on one dataset generalizes to others because it learns the *mechanics* of the gradient projection rather than the semantic content itself.[7]

3.1.2 Training the Decoder

The adversary simulates the fine-tuning process locally:

1. **Sample:** Draw a batch b from $\mathcal{D}_{proxy}$.
2. **Compute Full Gradient:** Calculate $\nabla W_{full} = \nabla_W \mathcal{L}(W_0; b)$.
3. **Compute LoRA Update:** Calculate the optimal rank-r update $\Delta W_{LoRA} = BA$ that approximates this gradient (or simply take a step of LoRA optimization).
4. **Optimize Decoder:** Train a model $\mathcal{D}_\phi$ to minimize the reconstruction loss:

$$\min_{\phi} \mathbb{E}_{b \sim \mathcal{D}_{proxy}}$$

Architecture:

- **For Vision:** The decoder is a U-Net style architecture. The input is the LoRA matrix product BA (or concatenated A, B), reshaped into tensor form. The output is the estimated gradient tensor $\hat{G}$.
- **For Text:** The decoder is a Transformer designed to process weight matrices. Recent work on "Learning on LoRAs" [16] suggests that Transformers can effectively process LoRA

weights as token sequences to predict model properties; here, we predict the gradients of the embedding layer.

Phase 2: Gradient Inversion (Crossing the Bridge)

Once the decoder $\mathcal{D}_\phi$ is trained, the attack proceeds against the victim.

1. **Acquisition:** The adversary downloads the victim's LoRA adapter (A_{vic}, B_{vic}).
2. **Decoding:** The adapter is passed through the Gradient Decoder to obtain an approximation of the accumulated training gradient:

$$\hat{G}_{target} = \mathcal{D}_\phi(A_{vic} B_{vic})$$

3. **Inversion Loop:** The adversary uses $\hat{G}_{target}$ as the target for a standard Gradient Inversion Attack (GIA). The goal is to find an input x^* that produces a gradient matching $\hat{G}_{target}$.

$$x^* = \arg\min_x \text{Dist}(\nabla_W \mathcal{L}(x; W_0), \hat{G}_{target}) + \text{TV}(x)$$

3.2.1 Inversion Backends

The choice of inversion backend depends on the domain:

- **Vision:** We utilize **GradInversion** [9] or **TAG**.[17] These methods employ sophisticated regularization (Total Variation, BN statistics) to coerce the optimization towards natural images. The decoder's output provides the "direction" in parameter space, while the regularizers constrain the search to the manifold of natural images.
- **Text:** We utilize **DAGER (Discreteness-Based Attack on Gradients for Exact Recovery)**.[8] DAGER is designed to recover discrete tokens from Transformer gradients by exploiting the low-rank structure of the self-attention gradients. It iterates through the vocabulary to find tokens whose embeddings lie in the subspace of the observed gradient. Our Bridge provides the "observed gradient" that DAGER requires.

Phase 3: Handling Aggregated Updates (The "Sum" Problem)

A significant technical challenge is that a released adapter is not a single-step gradient, but the accumulation of updates over multiple steps: $\Delta W_{final} \approx \eta \sum_t \nabla \mathcal{L}_t$.

Addressing the Sum:

1. **The "Meta-Batch" Perspective:** Mathematically, the sum of gradients $\sum \nabla \mathcal{L}(x_i)$ is

equivalent to the gradient of a single large "meta-batch" containing all the samples (scaled by learning rate). Gradient Inversion attacks like DLG are known to work on batches of size $B = 8$ or even $B = 100$.[18] If the LoRA fine-tuning dataset is small (few-shot), the entire dataset acts as one large batch.

2. **Federated Learning Precedent:** Research in Federated Learning on "Model Updates" (which are accumulated gradients) proves that reconstruction is possible even after multiple local epochs. Attacks like **AGIC** [19] and **GIT** [20] explicitly target accumulated updates. They show that while the reconstruction is noisier than single-step gradients, the dominant features (high-frequency edges, rare tokens) are preserved in the summation.
3. **Implicit Bias (Again):** As argued in Section 2.2, Haim et al. [4] showed reconstruction from *final weights* is possible because the weights converge to a stable configuration dominated by support vectors. We do not need to disentangle the time-series of gradients; we only need to invert the final stationary state, which is a weighted sum of the most critical training examples.

---

4. Empirical Feasibility: The 2x2 Grid

The efficacy of the Gradient Bridge is not uniform. We analyze its feasibility across two critical dimensions: Domain (Vision vs. Text) and Scale (Few-shot vs. Large).

4.1 Vision vs. Text

Vision (High Feasibility):

- **Mechanism:** Continuous optimization in pixel space.
- **Analysis:** LoRA adapters for Stable Diffusion (e.g., Dreambooth) typically target specific visual concepts (a face, a style). These concepts have strong spatial structure. The low-rank update focuses heavily on minimizing the residual error of these specific features.
- **Reconstruction:** Evidence from **LoRA-WiSE** [21] shows that LoRA weights encode metadata like dataset size and quality. Combining this with GradInversion suggests that reconstructing a specific face used for fine-tuning is highly probable. The decoder effectively acts as a super-resolution network, inferring high-frequency facial details from the low-frequency adapter weights.

Text (Medium to High Feasibility):

- **Mechanism:** Discrete optimization over vocabulary.
- **Analysis:** Reconstructing exact sentences is combinatorially hard. However, recovering **keywords** and **PII** (Personally Identifiable Information) is the primary risk.
- **Evidence:** The **LoRA-Leak** study [22] demonstrates that membership inference is highly effective on LoRA. Furthermore, **DAGER** [8] shows that gradients leak "anchor tokens"—rare

words that cause large updates to the embedding layer. Even if we cannot reconstruct the syntax "My password is X", the Bridge attack can likely recover the token "X" if it induced a significant gradient update (which rare/high-loss tokens typically do).

4.2 Few-Shot vs. Large Datasets

Few-Shot (The "Red Zone"):

- **Scenario:** Fine-tuning on 5-50 examples (e.g., "Teach Llama to speak like me," "Put my dog in Stable Diffusion").
- **Risk:** Critical.
- **Reasoning:** In the few-shot regime, the aggregated gradient ΔW is the sum of very few vectors. The components of these vectors are less likely to cancel each other out (destructive interference) and more likely to constructively interfere. The effective rank of the training signal is low ($\leq N$), meaning the rank-r LoRA adapter ($r \approx 16$) can capture the training signal with almost zero compression loss. Reconstruction should be near-perfect.

Large Datasets (The "Outlier Detector"):

- **Scenario:** Fine-tuning on 100k+ instruction pairs.
- **Risk:** High for Outliers.
- **Reasoning:** As $N \to \infty$, the update converges to the expected gradient of the distribution. Reconstructing an "average" data point yields a generic prototype (e.g., a generic face or sentence). However, due to the implicit bias of margin maximization, the weights are not shaped by the "average" data, but by the **outliers** and **misclassified examples**.[4]
- **Implication:** The Gradient Bridge in this regime functions as an anomaly detector. It will not recover the 99% of common English sentences, but it will reconstruct the 1% of data that was "hard" to learn—which often corresponds to sensitive, out-of-distribution secrets (e.g., code snippets with keys, private medical notes mixed into public data).

---

5. Research Plan and Roadmap

We propose a four-phase research plan to validate and refine the Gradient Bridge attack.

Phase 0: Minimal Scaffold (Feasibility Check)

- **Goal:** Establish the baseline leakage of raw LoRA weights without a learned decoder.
- **Experiment:** Fine-tune a simple ConvNet on CIFAR-10 (batch size 1) using LoRA (rank 8).
- **Attack:** Treat the flattened BA matrix directly as the gradient input to a DLG [15] optimizer.

- **Success Metric:** Visual similarity (SSIM) of the reconstructed image. If shapes are visible, the "raw" leakage is confirmed.

Phase 1: Train the Gradient Decoder (The Bridge)

- **Goal:** Build the inverse projection model.
- **Data:** Use ImageNet (for vision) or C4 (for text) as $\mathcal{D}_{proxy}$.
- **Procedure:**
  1. Generate 100k pairs of $(\Delta W_{LoRA}, \nabla W_{full})$ by simulating single-step fine-tuning.
  2. Train a U-Net (Vision) or Transformer (Text) to map $\Delta W_{LoRA} \rightarrow \nabla W_{full}$.
  3. Validate reconstruction fidelity (MSE, Cosine Similarity) on a held-out test set.
- **Key Insight:** Leverage the **R2F** codebase [7] which already implements a version of this decoder for unlearning.

Phase 2: Inversion Attacks (End-to-End)

- **Goal:** Recover actual data from "victim" adapters.
- **Setup:** Train victim LoRAs on private datasets (e.g., LFW faces, Enron emails).
- **Attack:**
  1. Pass victim adapter through the Phase 1 Decoder.
  2. Feed output to **GradInversion** (Vision) or **DAGER** (Text).
- **Metrics:** Use ROUGE-L (Text) and SSIM/LPIPS (Vision) to compare reconstructions vs. ground truth. Compare results against baselines (random noise, raw LoRA inversion without decoder).

Phase 3: Systematic Study (The Grid)

- **Objective:** Map the risk surface.
- **Variables:**
  - **Rank (r):** Vary $r \in $. Hypothesis: Leakage increases with r .
  - **Dataset Size (N):** Vary $N \in $. Hypothesis: Precision drops as N increases, but outlier recovery remains high.
  - **Epochs:** Compare reconstruction from epoch 1 vs. epoch 10. Test the "Accumulated Update" hypothesis—does long training wash out the signal or refine the support vectors?

---

6. Addressing Technical Challenges

Q: Does the decoder generalize across distributions? A: The R2F study [7] suggests yes. Because the decoder learns the *structural* relationship between full and low-rank spaces (determined by the model architecture), it transfers well. We can further enhance this by using "geometry-aware" decoders that focus on the singular value spectrum rather than semantic content.

Q: How does the attack handle the "Sum of Gradients"? A: We treat the accumulated update as a "Meta-Gradient." Attacks like **AGIC** [19] in Federated Learning successfully invert sums of gradients by solving for the "average" image. For text, **DAGER** [8] recovers batch tokens by checking if token embeddings lie in the gradient subspace. This subspace property holds for sums of gradients just as it does for single gradients.

Q: Is there a defense? A: Differential Privacy (DP-SGD) is the theoretical defense, as it adds noise to gradients to mask individual contributions. However, standard DP degrades LoRA performance significantly. **Dropout** on the adapter weights during training might also disrupt the precise invertibility of the projection, as shown in **LoRA-Leak** mitigations.[22]

---

7. Conclusion

The "Gradient Bridge" represents a paradigm shift in the security analysis of Parameter-Efficient Fine-Tuning. By synthesizing the theoretical rigor of implicit bias with the practical machinery of gradient inversion, we demonstrate that **low-rank adapters are not safe containers for private data.**

A LoRA adapter is effectively a zip file of the training data's support vectors. The rank constraint, previously thought to be a privacy shield, is merely a compression algorithm that the Gradient Bridge can unzip. This reality necessitates a re-evaluation of the open-adapter ecosystem. Organizations must treat fine-tuned adapters with the same data governance classification as the raw data they were trained on, particularly in few-shot and sensitive domains. The bridge is theoretically sound and empirically viable; the only remaining variable is the fidelity of the reconstruction in adversarial settings.

Table 1: Taxonomy of Related Attacks & The Unique Position of Gradient Bridge

Attack Strategy	**Target**	**Goal**	**Mechanism**	**Limitations**
Model Inversion [3]	Full Model	Class Representative	Maximize class probability	Recovers generic "average"

				image, not specific training samples.
Membership Inference [23]	LoRA	Membership Bit (Yes/No)	Loss threshold / Shadow models	Only confirms presence; does not reconstruct data.
GDBR (Zhang et al.) [24]	FL Updates	Label Distribution	Layer-wise gradient correlations	Recovers labels, not input features (images/text).
Deep Leakage (DLG) [15]	Single Grad	Exact Input	Optimization (Gradient Matching)	Typically assumes single step; struggles with low-rank constraints directly.
R2F (Unlearning) [7]	LoRA	Unlearning	Gradient Decoder + Ascent	Goal is removal, not reconstruction (but proves decoding works).
Gradient Bridge (Ours)	LoRA	**Exact Input**	**Gradient Decoder + DLG/DAGER**	Combines decoding (R2F) with inversion (DLG) to recover data.

עבודות שצוטטו

1. Low-Rank Adaptation (LoRA) Explained - Docker, נרשמה גישה בתאריך ינואר 18, 2026, https://www.docker.com/blog/lora-explained/
2. What is LoRA (Low-Rank Adaption)? - IBM, נרשמה גישה בתאריך ינואר 18, 2026, https://www.ibm.com/think/topics/lora

3. LoRA Reconstruction Distillation - Emergent Mind, נרשמה גישה בתאריך ינואר 18, 2026, https://www.emergentmind.com/topics/lora-reconstruction-distillation
4. THE_PAPER.pdf
5. LoRA-Pro: Are Low-Rank Adapters Properly Optimized? - arXiv, נרשמה גישה בתאריך ינואר 18, 2026, https://arxiv.org/html/2407.18242v2
6. LoRA-GA: Low-Rank Adaptation with Gradient Approximation - OpenReview, נרשמה גישה בתאריך ינואר 18, 2026, https://openreview.net/pdf?id=VaLAWrLHJv
7. Recover-to-Forget: Gradient Reconstruction from LoRA for Efficient LLM Unlearning - OpenReview, נרשמה גישה בתאריך ינואר 18, 2026, https://openreview.net/pdf?id=n7peBaPUmk
8. DAGER: Exact Gradient Inversion for Large Language Models - NIPS papers, נרשמה גישה בתאריך ינואר 18, 2026, https://proceedings.neurips.cc/paper_files/paper/2024/file/9ff1577a1f8308df1ccea6b4f64a103f-Paper-Conference.pdf
9. See through Gradients: Image Batch Recovery via GradInversion - Jan Kautz, נרשמה גישה בתאריך ינואר 18, 2026, https://jankautz.com/publications/SeeThroughGradients_CVPR21.pdf
10. LORA-PRO: ARE LOW-RANK ADAPTERS PROPERLY OPTIMIZED? - ICLR Proceedings, נרשמה גישה בתאריך ינואר 18, 2026, https://proceedings.iclr.cc/paper_files/paper/2025/file/ea184f920a0f0f8d8030aa1bd7ac9fd4-Paper-Conference.pdf
11. LoRA-GA: Low-Rank Adaptation with Gradient Approximation - NIPS papers, נרשמה גישה בתאריך ינואר 18, 2026, https://proceedings.neurips.cc/paper_files/paper/2024/file/62c4718cc334f6a0a62fb81c4a2095a1-Paper-Conference.pdf
12. GaLore: Memory-Efficient LLM Training by Gradient Low-Rank Projection - arXiv, נרשמה גישה בתאריך ינואר 18, 2026, https://arxiv.org/html/2403.03507v2
13. Mirror, Mirror of the Flow: How Does Regularization Shape Implicit Bias? - GitHub, נרשמה גישה בתאריך ינואר 18, 2026, https://raw.githubusercontent.com/mlresearch/v267/main/assets/jacobs25a/jacobs25a.pdf
14. LoRA-One: One-Step Full Gradient Could Suffice for Fine-Tuning Large Language Models, Provably and Efficiently (ICML2025 Oral) - GitHub, נרשמה גישה בתאריך ינואר 18, 2026, https://github.com/YuanheZ/LoRA-One
15. Recover-to-Forget: Gradient Reconstruction from LoRA for Efficient LLM Unlearning - arXiv, נרשמה גישה בתאריך ינואר 18, 2026, https://arxiv.org/html/2512.07374v1
16. Learning on LoRAs: GL-Equivariant Processing of Low-Rank Weight Spaces for Large Finetuned Models - arXiv, נרשמה גישה בתאריך ינואר 18, 2026, https://arxiv.org/html/2410.04207v2
17. Learning To Invert: Simple Adaptive Attacks for Gradient Inversion in Federated Learning - Proceedings of Machine Learning Research, נרשמה גישה בתאריך ינואר 18, 2026, https://proceedings.mlr.press/v216/wu23a/wu23a.pdf
18. Inverting Gradients - How easy is it to break privacy in federated learning?, נרשמה גישה בתאריך ינואר 18, 2026,

https://papers.neurips.cc/paper_files/paper/2020/file/c4ede56bbd98819ae6112b20ac6bf145-Paper.pdf

19. AGIC: Approximate Gradient Inversion Attack on Federated Learning - IEEE Xplore, נרשמה גישה בתאריך ינואר 18, 2026, https://ieeexplore.ieee.org/document/9996844/
20. Gradient Inversion Transcript: Leveraging Robust Generated Priors to Reconstruct Training Data from Gradient Leakage | OpenReview, נרשמה גישה בתאריך ינואר 18, 2026, https://openreview.net/forum?id=rlq9aKsY7T
21. Data Size Recovery from Lora Weights - HUJI, נרשמה גישה בתאריך ינואר 18, 2026, https://vision.huji.ac.il/dsire/
22. [2507.18302] LoRA-Leak: Membership Inference Attacks Against LoRA Fine-tuned Language Models - arXiv, נרשמה גישה בתאריך ינואר 18, 2026, https://arxiv.org/abs/2507.18302
23. LoRA-Leak: Unveiling Membership Inference Vulnerabilities in Fine-Tuned Language Models | by Sai Dheeraj Gummadi | Data Science in Your Pocket | Medium, נרשמה גישה בתאריך ינואר 18, 2026, https://medium.com/data-science-in-your-pocket/lora-leak-unveiling-membership-inference-vulnerabilities-in-fine-tuned-language-models-7c4adc321d39
24. (PDF) Images in Motion?: A First Look into Video Leakage in Collaborative Deep Learning, נרשמה גישה בתאריך ינואר 18, 2026, https://www.researchgate.net/publication/395474662_Images_in_Motion_A_First_Look_into_Video_Leakage_in_Collaborative_Deep_Learning