Surrogate Model Extension (SME): A Fast and Accurate Weight Update Attack on Federated Learning

Junyi Zhu [1] **Ruicong Yao** [2] **Matthew B. Blaschko** [1]

Abstract

In Federated Learning (FL) and many other distributed training frameworks, collaborators can hold their private data locally and only share the network weights trained with the local data after multiple iterations. Gradient inversion is a family of privacy attacks that recovers data from its generated gradients. Seemingly, FL can provide a degree of protection against gradient inversion attacks on weight updates, since the gradient of a single step is concealed by the accumulation of gradients over multiple local iterations. In this work, we propose a principled way to extend gradient inversion attacks to weight updates in FL, thereby better exposing weaknesses in the presumed privacy protection inherent in FL. In particular, we propose a surrogate model method based on the characteristic of two-dimensional gradient flow and low-rank property of local updates. Our method largely boosts the ability of gradient inversion attacks on weight updates containing many iterations and achieves state-of-the-art (SOTA) performance. Additionally, our method runs up to $100\times$ faster than the SOTA baseline in the common FL scenario. Our work re-evaluates and highlights the privacy risk of sharing network weights. Our code is available at https://github.com/JunyiZhu-AI/surrogate_model_extension.

1. Introduction

Privacy concerns arise from many areas. Therefore, data-driven technologies, e.g. machine learning, cannot solely rely on a large data center, but also adapt to distributed data scenarios. Federated Learning (FL) is proposed to this end (McMahan et al., 2017; Kairouz et al., 2021), which is a framework for training a joint model built upon server-client communication. A presumed strength of FL from the perspective of privacy is that the network weights instead of data are shared between the server and clients. However, training data may still be extracted from the transmitted weights due to their statistical dependence.

Gradient inversion presents a way to reconstruct input data from the gradient (Zhu et al., 2019; Geiping et al., 2020). The reconstruction frequently has high fidelity, thus raising alarms about the privacy risk of sharing network weights. However, such methods assume that weights are shared at every iteration, thus the gradient of a single update can be constructed from server-client communications. In practice, communication overhead is a major bottleneck of collaborative training, especially for a wide range of applications of FL on edge devices. Therefore, FL frameworks mostly require clients to train multiple epochs before sharing their updated weights. The gradient of a single step is thus concealed by the accumulation of local iterations. An existing attack on weight updates in FL is the simulation method (Dimitrov et al., 2022), which emulates multiple steps of gradient descent. However, such methods are computationally demanding and may not be possible to apply in practice, providing the illusion of data protection.

In this work, we propose an extension of the gradient inversion method, which attacks weight updates of accumulated local iterations in a principled way. In particular, we introduce a surrogate model, which is a model not necessarily appearing during training. We take the surrogate model as a basis and regard the reversed direction of the weight update as the direction of the gradient of training data computed on this surrogate model. We propose that an effective surrogate model can be found as a linear combination of the weights before and after local training, and identify that this is feasible due to the characteristic of two-dimensional gradient flow and the low-rank property of local steps. Equipped with this Surrogate Model Extension (`SME`), we can then invert the weight update and reconstruct training data using any gradient inversion method. Figure 1 illustrates the threat model and working pipeline of our method `SME`.

Our contributions are summarized as: (i) We propose a

[1]ESAT-PSI, KU Leuven, Belgium [2]Statistics and Data Science, Dept. Mathematics, KU Leuven, Belgium. Correspondence to: Junyi Zhu <Junyi.Zhu@esat.kuleuven.be>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

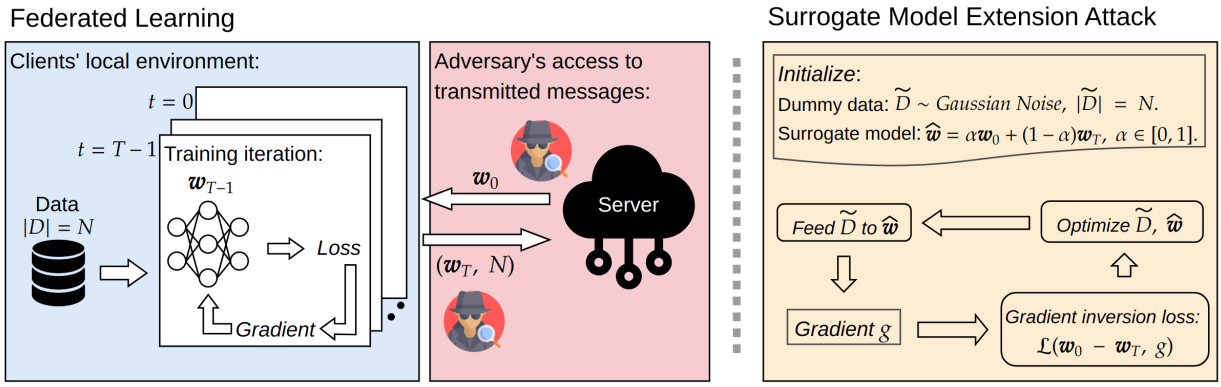


Figure 1. Illustration of the threat model (left) and working pipeline of our surrogate model extension (right). In FL, a client trains the received model $\boldsymbol{w}_0$ for T iterations with local data set D of size N, then sends the weights and the number N back to the server. An adversary observes the messages and launches the SME attack through optimization of dummy data $\tilde{D}$ and surrogate model $\hat{\boldsymbol{w}}$.

method that extends gradient inversion to weight updates computed from multiple iterations over the data. (ii) We also provide an analysis for theoretical understanding of the proposed method. (iii) We demonstrate that our method largely strengthens the ability of gradient inversion in attacking weight updates at a negligible additional cost. (iv) Our method achieves SOTA performance in reconstruction and requires substantially fewer computational resources than the current SOTA approach.

Paper Organization. In Section 2 we discuss the related works. In Section 3 we provide methodological foundations of FL and inversion attacks. In Section 4, we elaborate on our method with theoretical analysis underpinning its generality. In Section 5 we present the experimental results. Finally, we conclude in Section 6.

2. Related Works

The first observation of data leakage from gradients appears to be due to (Phong et al., 2018), where they find that the gradient w.r.t. weights divided by the gradient w.r.t. bias can be used to reconstruct the input of a fully connected layer. Subsequently, Zhu et al. (2019) propose the first general gradient inversion method to reconstruct the input from the gradient over an arbitrary network architecture. A line of follow-up work intends to improve the reconstruction quality by further constraining the reconstruction through some auxiliary information of the input (Geiping et al., 2020; Yin et al., 2021; Lu et al., 2021; Jeon et al., 2021). Additionally, Zhu & Blaschko (2021); Fan et al. (2020) propose analytic methods for gradient inversion. Balunovic et al. (2022) propose a Bayesian framework.

Many works discuss the threat of gradient inversion to FL (Wei et al., 2020; Geng et al., 2021; Lam et al., 2021; Jin et al., 2021). Since in FL and many other distributed training schemes, weight updates are in form of multiple local steps instead of a single step, Geiping et al. (2020); Dimitrov et al. (2022) propose simulation methods which fit the weight updates through mimicking the local training iterations. We provide a broader discussion of other types of privacy attacks and defense strategies in Appendix A.

3. Preliminaries

This section presents the necessary background of this work. In Section 3.1, we review the mechanism of gradient inversion attacks. In Section 3.2 we introduce a general training protocol of the FL framework, which we will use as the attack scenario in this work. In Section 3.3 we clarify the simulation method that is specifically designed for attacking weight updates of many iterations.

3.1. Deep Leakage from Gradients

Data used to train the network can be extracted from the generated gradient. Zhu et al. (2019) propose the first general gradient inversion method called Deep Leakage from Gradient (DLG), which recovers the original data by searching for the data that generates a gradient matching the original one. Denote by $\ell(\cdot)$ the loss function of the network training, and denote by $\boldsymbol{w}$ the weights of the network, $D = \{(x_i, y_i)\}_{i=1}^{N}$ is the data used to generate the gradient, which consists of N pairs of inputs x and labels y. DLG optimizes a randomly initialized dummy data $\tilde{D}$ w.r.t. the Euclidean distance between the original gradient and dummy gradient. Thus, the reconstruction loss used in DLG can be defined as: $\|\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}, D) - \nabla_{\boldsymbol{w}}\ell(\boldsymbol{w}, \tilde{D})\|$.

Based on DLG, another work Inverting Gradient (IG) proposes to replace the Euclidean distance with cosine similarity loss and incorporate the total variation (TV) of the pixel values as a prior for image reconstruction (Geiping

et al., 2020). The reconstruction loss $\mathcal{L}$ of IG can thus be described as:

$$\mathcal{L} = \overbrace{1 - \frac{\langle \nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, D), \nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, \tilde{D}) \rangle}{\|\nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, D)\| \|\nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, \tilde{D})\|}}^{\mathcal{L}_{sim}(\nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, D), \nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, \tilde{D}))} + \lambda \underbrace{\mathrm{TV}(\tilde{D})}_{\mathcal{L}_{prior}}, \quad (1)$$

where λ is a hyperparameter tuning the strength of the prior. Later, we will use $\mathcal{L}_{sim}$ to denote the cosine similarity loss.

IG has been proven to be more stable and efficient than DLG in practice, and its objective is relatively simple and general. In this work, we will use the image reconstruction attack as a proof of concept and rely on the reconstruction loss defined in Equation (1). Our method SME focuses on adapting the core term in Equation (1), i.e. the gradient similarity loss $\mathcal{L}_{sim}$, to the weight update of many iterations.

3.2. Communication in Federated Learning

In this work we consider a classical FL framework: Federated Averaging (FedAvg) (McMahan et al., 2017). The procedure of FedAvg is given in Algorithm 2 in Appendix D. For each communication round in FedAvg, clients train the local model for E epochs before sending back the optimized weights. While for each epoch, the network will be updated for $\lceil N/B \rceil$ steps, where B is the batch size and N is the local data size. Generally, B is fixed by the training protocol and N depends on how much data a client can collect.

Therefore, for one round of communication, the network weights that could be intercepted by an adversary at the communication channel side, or observed by the server, are $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$. The weight update $\boldsymbol{w}_T - \boldsymbol{w}_0$ accumulates $T = E\lceil N/B \rceil$ local steps of gradient descent, thus the gradient of every single step is obfuscated.

Weight updates of multiple steps can still be plugged into the reconstruction loss of gradient inversion (Wei et al., 2020; Geng et al., 2021). Such technologies usually normalize the reversed weight update of T steps, i.e. $(\boldsymbol{w}_0 - \boldsymbol{w}_T)/\eta T$, where η is the learning rate. Then they take the normalized gradient as an approximation of the gradient of the local data set D computed on $\boldsymbol{w}_0$, i.e. $\nabla_{\boldsymbol{w}_0} \ell(\boldsymbol{w}_0, D)$, and apply gradient inversion to reconstruct the local data set D. However, as the number of local steps T increases, the normalized gradient incrementally mismatches the gradient at $\boldsymbol{w}_0$, which induces intrinsic objective error defined as the cosine similarity loss with the optimal reconstruction $\tilde{D} = D$:

$$\delta_{error} := \mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla_{\boldsymbol{w}_0} \ell(\boldsymbol{w}_0, D)), \quad (2)$$

since $1 - \mathcal{L}_{sim}$ is a cosine similarity, it is invariant to the rescaling $1/\eta T$. As a result, when the direction of weight updates deviates from $\nabla_{\boldsymbol{w}_0} \ell(\boldsymbol{w}_0, D)$, this error increases. It has been observed that the reconstruction quality of gradient inversion degrades with the number of local steps T (Wei et al., 2020; Dimitrov et al., 2022).

3.3. Inverting Weight Updates through Simulation

An existing privacy attack targeting weight updates is the simulation method, for which Data Leakage from Federated Averaging (DLFA) achieves SOTA performance in terms of the reconstruction quality (Dimitrov et al., 2022). Simulation methods reconstruct data by fitting internal states of the local training. That is, the adversary mimics the local training steps by optimizing $\boldsymbol{w}_0$ with dummy data $\tilde{D}$ and obtains dummy weights $\{\tilde{\boldsymbol{w}}_t\}_{t=1}^T$. Then the adversary minimizes the difference between the weight updates, i.e. $\tilde{\boldsymbol{w}}_T - \boldsymbol{w}_0$ and $\boldsymbol{w}_T - \boldsymbol{w}_0$. The simulation loss $\mathcal{L}$ can thus be defined as:

$$\begin{aligned} \mathcal{L} &= \|\tilde{\boldsymbol{w}}_T - \boldsymbol{w}_0 - (\boldsymbol{w}_T - \boldsymbol{w}_0)\| \\ &= \|\eta(\nabla \boldsymbol{w}_0 + \nabla \tilde{\boldsymbol{w}}_1 + \cdots + \nabla \tilde{\boldsymbol{w}}_{T-1}) - \boldsymbol{w}_T + \boldsymbol{w}_0\|, \end{aligned}$$

where we define $\nabla \boldsymbol{w} = \nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, \tilde{D})$. The Euclidean distance can also be replaced by a cosine similarity loss.

Simulation methods circumvent the intrinsic objective error of the gradient inversion. However, a major issue of the simulation methods is their computational complexity. Consider a simulation for only two local steps, the reconstruction gradient w.r.t. the dummy data can be derived as:[1]

$$\begin{aligned} \nabla_{\tilde{D}} \mathcal{L} = &\frac{\partial \mathcal{L}}{\partial \nabla \tilde{\boldsymbol{w}}_1} \left(\frac{\partial \nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, \tilde{D})}{\partial \tilde{D}} |_{\boldsymbol{w} = \tilde{\boldsymbol{w}}_1} - \eta \frac{\partial \nabla \tilde{\boldsymbol{w}}_1}{\partial \tilde{\boldsymbol{w}}_1} \frac{\partial \nabla \boldsymbol{w}_0}{\partial \tilde{D}} \right) \\ &+ \frac{\partial \mathcal{L}}{\partial \nabla \boldsymbol{w}_0} \frac{\partial \nabla \boldsymbol{w}_0}{\partial \tilde{D}}. \end{aligned} \quad (3)$$

A full derivation of $\nabla_{\tilde{D}} \mathcal{L}$ for T local steps is deferred to Appendix B. According to Equation (3), we can see that the reconstruction gradient $\nabla_{\tilde{D}} \mathcal{L}$ of simulation methods has a complicated form, which involves the computation of multiple second-order derivatives and their products. When there are many local steps, a long chain of forward and backward propagation will be induced, which leads to very large computational overhead and memory footprint.

4. Inverting Weight Updates through a Surrogate Model

In this section, we elaborate on our surrogate model extension SME. Denote $\widehat{\boldsymbol{w}}$ as the surrogate model, we give the general objective of surrogate model extension attack:

$$\arg\min_{\tilde{D} \in \mathcal{D}} \min_{\widehat{\boldsymbol{w}} \in \mathcal{W}} \mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla_{\widehat{\boldsymbol{w}}} \ell(\widehat{\boldsymbol{w}}, \tilde{D})). \quad (4)$$

[1] In the original work (Dimitrov et al., 2022), the input could be batches of the dummy data. Here we consider the full $\tilde{D}$ as input for clarity. This does not change our discussion.

When we fix $\widehat{\boldsymbol{w}}$ to $\boldsymbol{w}_0$, this objective reverts to the vanilla gradient inversion and contains an intrinsic objective error as shown in Equation (2). It is worth noting that $\widehat{\boldsymbol{w}}$ generally has many more parameters than $\tilde{D}$. If we do not constrain the feasible set $\mathcal{W}$, this objective still cannot give us a good reconstruction.

Our key insight is that the linear combinations of $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$ is a good choice of $\mathcal{W}$, i.e. $\mathcal{W} = \{\alpha\boldsymbol{w}_0 + (1-\alpha)\boldsymbol{w}_T \mid \alpha \in [0,1]\}$. We identify that if the parameter space is two-dimensional (2D), then there exists a $\widehat{\boldsymbol{w}}$ on the connected line between $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$, such that $\nabla_{\widehat{\boldsymbol{w}}}\ell(\widehat{\boldsymbol{w}}, D)$ is parallel to $\boldsymbol{w}_0 - \boldsymbol{w}_T$, denoted as $\nabla_{\widehat{\boldsymbol{w}}}\ell(\widehat{\boldsymbol{w}}, D) \parallel \boldsymbol{w}_0 - \boldsymbol{w}_T$. Thus a minimizer $\widehat{\boldsymbol{w}}$ corresponding to the original data D can be found on the connected line, which leads to $\mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla_{\widehat{\boldsymbol{w}}}\ell(\widehat{\boldsymbol{w}}, D)) = 0$. To keep the notation uncluttered, we will slightly abuse $\nabla\boldsymbol{w}$ and denote $\nabla\boldsymbol{w} := \nabla\ell(\boldsymbol{w}, D)$.

For a general high-dimensional parameter space, we further observe that the local update steps of a network basically transit within a low-dimensional subspace due to the low-rank property of the gradients. Therefore, for a high-dimensional network, it is still possible to find a surrogate model $\widehat{\boldsymbol{w}}$ as a linear combination of $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$, such that the direction of $\nabla\widehat{\boldsymbol{w}}$ and $\boldsymbol{w}_0 - \boldsymbol{w}_T$ are close. The low-rank property of gradients has also been observed and utilized by a wide range of works (Vogels et al., 2019; Gooneratne et al., 2020; Li et al., 2020; Zhou et al., 2021; Li et al., 2022). Additionally, as we can reparamterize the surrogate model with only a scalar α, simultaneously optimizing the reconstructed data $\tilde{D}$ and the surrogate model $\widehat{\boldsymbol{w}}$ becomes tractable. After incorporating the image prior TV, we define the reconstruction loss $\mathcal{L}$ of our SME as:

$$\mathcal{L}(\tilde{D}, \widehat{\boldsymbol{w}}) = \overbrace{1 - \frac{\langle \boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla_{\widehat{\boldsymbol{w}}}\ell(\widehat{\boldsymbol{w}}, \tilde{D})\rangle}{\|\boldsymbol{w}_0 - \boldsymbol{w}_T\|\|\nabla_{\widehat{\boldsymbol{w}}}\ell(\widehat{\boldsymbol{w}}, \tilde{D})\|}}^{\mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla_{\widehat{\boldsymbol{w}}}\ell(\widehat{\boldsymbol{w}}, \tilde{D}))} + \lambda \underbrace{\mathrm{TV}(\tilde{D})}_{\mathcal{L}_{prior}},$$
$$s.t. \quad \widehat{\boldsymbol{w}} \in \{\alpha\boldsymbol{w}_0 + (1-\alpha)\boldsymbol{w}_T \mid \alpha \in [0,1]\}. \tag{5}$$

Next, in Section 4.1 we first prove that for a 2D parameter space, there exists a minimizer of $\widehat{\boldsymbol{w}}$ for the original data on the connected line between the starting point $\boldsymbol{w}_0$ and end point $\boldsymbol{w}_T$ of gradient flow (gradient descent (GD) with infinitesimal step sizes). Then in Section 4.2, we extend our analysis to more general cases. Finally, in Section 4.3 we present the implementation of our surrogate model extension. All proofs are deferred to Appendix C.

4.1. Gradient Flow in a 2-Dimensional Parameter Space

Figure 2 illustrates a gradient flow in a quadratic model and motivates our surrogate model extension. The key observation from this figure is that the gradient $\nabla\boldsymbol{w}_0$ points to the right hand side of $\boldsymbol{w}_T - \boldsymbol{w}_0$, while $\nabla\boldsymbol{w}_T$ points to the other side. Since the gradient is continuous, a surrogate model whose gradient is parallel to $\boldsymbol{w}_T - \boldsymbol{w}_0$ must exist. Although other configurations of the gradient flow exist, we can prove the following proposition holds generally:

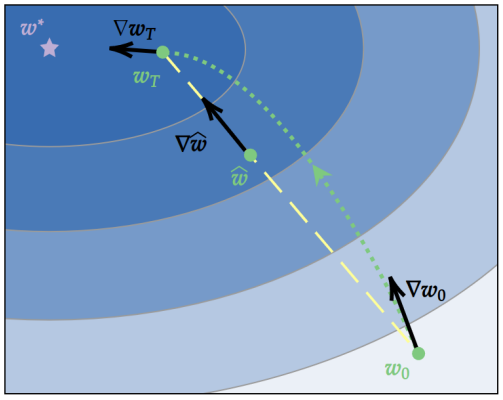


Figure 2. Illustration of gradient flow in a quadratic model. The green dotted line represents the gradient flow, and $\boldsymbol{w}^*$ denotes the global optimum.

Proposition 4.1. *For a 2D gradient flow $\boldsymbol{w}_t$ based on the the loss $\ell(\boldsymbol{w}) \in \mathcal{C}^2$, there exists a surrogate model $\widehat{\boldsymbol{w}}$ on the connected line between $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$ satisfying $\langle\nabla\widehat{\boldsymbol{w}}, \boldsymbol{w}_0 - \boldsymbol{w}_T\rangle \in \{\pm 1\}$, provided that $\langle\nabla\boldsymbol{w}_t, \boldsymbol{w}_0 - \boldsymbol{w}_T\rangle \geq 0$, $\forall t \in [0, T]$. If we further assume that $\ell(\boldsymbol{w})$ is convex, then $\nabla\widehat{\boldsymbol{w}} \parallel \boldsymbol{w}_0 - \boldsymbol{w}_T$.*

4.2. Cosine Similarity between the High-Dimensional Surrogate Gradient and Weight Update

In the 2D gradient flow case, it has been shown that there exists a value of α that achieves a cosine similarity of exactly 1. This may fail in higher dimensional spaces. Consider a gradient flow in $\mathbb{R}^3$ parameterized by $(\cos t - 1, \sin t, t)$, $t \in \mathbb{R}$, such that the origin is the starting point and $(0, 0, 2\pi)$ is the end point. We further assume that the gradient field in $\mathbb{R}^3$ is parameterized by $(\cos t - 1 - s, \sin t - r, t)$, $r, s, t \in \mathbb{R}$ (which is reasonable since the gradient flows can be similar to each other locally). Then we know that the gradient of the weights on the connected line would be $(-\sin t, \cos t, 1)$. This indicates that their cosine similarity with $(0, 0, 2\pi)$ would always be $1/\sqrt{2}$ which deviates from 1.

However, in the above artificial example, the gradient flow curls in three dimensions. In practice, we observe that there exists a 2D subspace that absorbs a majority of the magnitude of the gradients of local steps. Using this low-rank property of the local steps, we show that there exists a surrogate model $\widehat{\boldsymbol{w}}$ on the connected line, s.t. $\mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla\widehat{\boldsymbol{w}}) \approx 0$, where $1 - \mathcal{L}_{sim}$ is known as the cosine similarity. Additionally, we consider (stochastic) gradient descent with step size η. Our assumption and first result are:

Assumption 4.2. We assume that the true loss $\ell(\boldsymbol{w}) :=$

$\mathbb{E}_D[\ell(\boldsymbol{w}, D)]$ is Lipschitz with constant L and $\gamma-$strongly convex, the true gradient $\nabla \boldsymbol{w} := \mathbb{E}_D[\nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}, D)]$ is Lipschitz with constant β, and there exists $0 < G_2 \ll 1$ and a projection matrix P_2 from $\mathbb{R}^p$ to some 2D subspace V_2 s.t.

$$||P_2 \nabla \boldsymbol{w}||^2 \geq (1 - G_2^2)||\nabla \boldsymbol{w}||^2. \tag{6}$$

In the following, the gradient flow path is denoted as $\boldsymbol{w}(t), t \in \mathbb{R}^+$ to be distinguished with the GD path $\{\boldsymbol{w}_t\}_{t=1}^T$ with step size η. In addition, we assume that it satisfies:

$$1 - \mathcal{L}_{sim}(\nabla \boldsymbol{w}(t), \boldsymbol{w}(0) - \boldsymbol{w}(T\eta)) \geq G_2. \tag{7}$$

Theorem 4.3. *Let $\{\boldsymbol{w}_t\}_{t=0}^T$ result from the application of GD in $\mathbb{R}^p$. Under Assumption 4.2 and that $\eta\beta < 1$, $\eta < 1$, we have for $G(\eta, \beta) := \sqrt{\frac{G_2^2 \eta}{1 - \eta\beta/2}}$, there exists a surrogate model $\widehat{\boldsymbol{w}}$ on the connected line between $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$:*

$$\mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla \widehat{\boldsymbol{w}}) \leq \left(G_2 + \sqrt{\frac{TG^2(\eta, \beta)L^2}{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}_T)}} \right)^2 \tag{8}$$

up to a Δ_{trun} (see Equation (27)) due to discretization.

The proof is based on the identity $\langle \nabla \widehat{\boldsymbol{w}}, \boldsymbol{w}_0 - \boldsymbol{w}_T \rangle = \langle P_2 \nabla \widehat{\boldsymbol{w}}, P_2(\boldsymbol{w}_0 - \boldsymbol{w}_T) \rangle + \langle P_2^\perp \nabla \widehat{\boldsymbol{w}}, P_2^\perp(\boldsymbol{w}_0 - \boldsymbol{w}_T) \rangle$ ($P_2^\perp$ is the projection onto $V_2^\perp$) which allows us to estimate the quantity $\mathcal{L}_{sim}$ on two orthogonal subspaces. Here Δ_{trun} stems from the discretization (gradient flow to gradient descent path). The idea in the proof of Theorem 4.5 also adapts to its estimation and we argue that Δ_{trun} is negligible as long as GD approximates the gradient flow well, see Appendix C.2. Thus we omit it in Theorem 4.5 for brevity.

Remark 4.4. Theorem 4.3 shows that small step size η and high concentration of the gradient on V_2 (thus G_2 is small) result in low $\mathcal{L}_{sim}$. It also indicates that there is a 'scaling effect' of $T/(\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}_T))$. This explains why a good surrogate model can be found for long training steps when the drop of loss is adequate, given that G_2 is small.

For SGD, again we first deal with the 2D case. Since a fully random path does not have the property we want, we assume that the magnitude of the gradient noise is moderate so that we can approximate SGD $\{\boldsymbol{w}_t\}_{t=0}^T$ by GD $\{\boldsymbol{w}'_t\}_{t=0}^T$. We first show that with high probability, the angle θ_1 between $\boldsymbol{w}_T - \boldsymbol{w}_0$ and $\boldsymbol{w}'_T - \boldsymbol{w}_0$ is small. Then we can find a surrogate model $\widehat{\boldsymbol{w}}'$ on the connected line between $\boldsymbol{w}_0$ and $\boldsymbol{w}'_T$ by Proposition 4.1. We then try to locate a surrogate model $\widehat{\boldsymbol{w}}$ on the connected line between $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$ such that the angle θ_2 between $\nabla \widehat{\boldsymbol{w}}$ and $\nabla \widehat{\boldsymbol{w}}'$ is small. This is done by upper bounding $||\nabla \widehat{\boldsymbol{w}} - \nabla \widehat{\boldsymbol{w}}'||$ and lower bounding $||\nabla \widehat{\boldsymbol{w}}'||$, where we use strong convexity.

For high-dimensional spaces, we use the aforementioned identity and show that $||P_2^\perp(\boldsymbol{w}_T - \boldsymbol{w}_0)||/||\boldsymbol{w}_T - \boldsymbol{w}_0||$ is small. Thus the weight update is mainly in V_2, and $\mathcal{L}_{sim}$ would be similar to the 2D case. The final result is summarized in the following theorem:

Algorithm 1 Surrogate Model Extension

Input: Victim's weights $\boldsymbol{w}_0, \boldsymbol{w}_T$; Local data size N; Iterations K; Learning rate $\eta_{\tilde{D}}$ for the dummy data and η_α for α; Loss function $\mathcal{L}$.
Initialize $\tilde{D}_0$; $\alpha_0 \leftarrow 0.5$.
for each step $k = 0 \dots K-1$ **do**
  $\widehat{\boldsymbol{w}} = \alpha_k \boldsymbol{w}_0 + (1 - \alpha_k)\boldsymbol{w}_T$
  $\tilde{D}_{k+1} = \tilde{D}_k - \eta_{\tilde{D}} \nabla_{\tilde{D}_k} \mathcal{L}(\tilde{D}_k, \widehat{\boldsymbol{w}})$
  $\alpha_{k+1} = \alpha_k - \eta_\alpha \nabla_{\alpha_k} \mathcal{L}(\tilde{D}_k, \widehat{\boldsymbol{w}})$
end for
Output: Reconstructed data $\tilde{D}_K$.

Theorem 4.5. *Let $\{\boldsymbol{w}_t\}_{t=0}^T$ result from the application of SGD in $\mathbb{R}^p$ and $\{\boldsymbol{w}'_t\}_{t=0}^T$ result from GD where $\boldsymbol{w}_0 = \boldsymbol{w}'_0$. Assume the same assumptions as Theorem 4.3 and that $C_{GD} := \sqrt{\frac{TG^2(\eta,\beta)L^2}{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}_T)}} < 1$. Let $c := \frac{1}{\sqrt{\eta}(1+\eta\beta)^{T-1}}$, E_{max} be the maximum L^2 norm of the gradient noise and define:*

$$r := \min \left\{ \frac{c(\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T))^2}{4L^2 T E_{max}^2}, \frac{c(\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T))}{4LE_{max}} \right\}. \tag{9}$$

Let $C_\eta = \frac{1}{1 - \sqrt{\eta}}$. Then with probability at least $1 - 3T\exp(-r)$ there exists a surrogate model $\widehat{\boldsymbol{w}}$ on the connected line between $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$:

$$\mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla \widehat{\boldsymbol{w}}) \leq 2\eta + \frac{2\eta^2\beta^2 T}{\gamma(1 - G_2^2)} \frac{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)}{\ell(\boldsymbol{w}'_T) - \ell(\boldsymbol{w}^*)} + \frac{C_\eta(1 + G_2)}{1 - C_{GD}} \left(C_{GD} + \frac{LTE_{max}\eta(1 + \eta\beta)^{T-1}}{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)} \right) + G_2^2, \tag{10}$$

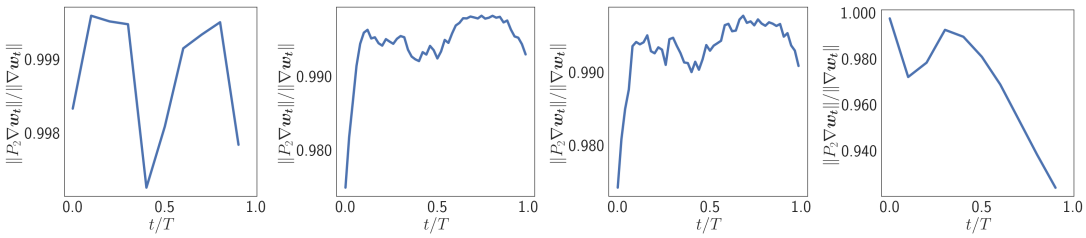
where $\boldsymbol{w}^*$ is the global minimum.

Remark 4.6. The scaling effect in Theorem 4.3 also exists in SGD due to C_{GD}. Note that the process from $\boldsymbol{w}_0$ to $\boldsymbol{w}_T$ is just local training steps (compared to the fully trained model), it's natural to expect that $\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)$ is much less than $\ell(\boldsymbol{w}'_T) - \ell(\boldsymbol{w}^*)$ in the second term. The ratio w.r.t. E_{max} in the third term indicates that a small $\mathcal{L}_{sim}$ may not be achieved when gradient noise is large while loss reduction $\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)$ is small.

4.3. Implementation of the Surrogate Model Extension

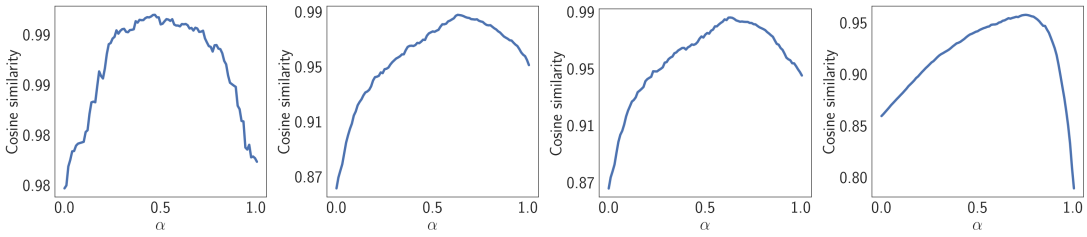
Our method `SME` is easy to implement. Based on the objective of `SME` in Equation (5), in addition to the gradient inversion steps, we only need to update the surrogate model's weights w.r.t. α before the forward propagation and optimize α after the backward propagation. Algorithm 1 presents the optimization steps of our approach.

It is worth emphasizing that the additional computation caused by `SME` due to the optimization of α is negligible.



(a) E=10, B=50, R=1 (b) E=50, B=50, R=1 (c) E=10, B=10, R=1 (d) E=10, B=50, R=20

Figure 3. Gradient norm ratio $\|P_2\nabla \boldsymbol{w}_t\|/\|\nabla \boldsymbol{w}_t\|$ along the local steps $t = 0 \dots T-1$. We consider a client with $N = 50$ data points and in (a) we train a regular number of epochs $E = 10$ with full gradient descent due to $B \geq N$. For comparison, we consider three other settings: (b) a large number of epochs $E = 50$; (c) stochastic mini-batch gradient descent with $N/B = 5$; (d) training on a network that has been optimized for $R = 20$ communication rounds. Note that (b) and (c) have the same local steps $T = 50$, but (c) optimizes with SGD.



(a) E=10, B=50, R=1 (b) E=50, B=50, R=1 (c) E=10, B=10, R=1 (d) E=10, B=50, R=20

Figure 4. Cosine similarity between $\boldsymbol{w}_0 - \boldsymbol{w}_T$ and $\nabla\widehat{\boldsymbol{w}}$ vs. α. The settings are aligned with Figure 3, please refer to the caption above.

Since reconstruction loss $\mathcal{L}$ only depends on α through $\widehat{\boldsymbol{w}}$ due to $\widehat{\boldsymbol{w}} = \alpha\boldsymbol{w}_0 + (1-\alpha)\boldsymbol{w}_T$, we have:

$$\nabla_\alpha \mathcal{L}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla\widehat{\boldsymbol{w}}) = \frac{\partial \mathcal{L}}{\partial \widehat{\boldsymbol{w}}} \cdot (\boldsymbol{w}_T - \boldsymbol{w}_0), \qquad (11)$$

which is simply an inner product of the gradient and weight update. Compared with the simulation methods as discussed in Section 3.3, `SME` is expected to run much faster.

5. Experiments

In this section, we present empirical evidence for our analysis and validate the performance of our method `SME`. We compare `SME` with the vanilla gradient inversion method `IG` (Geiping et al., 2020), which our extension is based on, and the simulation method `DLFA` (Dimitrov et al., 2022), which achieves the current SOTA weight update attack.

Setup: We conduct our experiments on two image classification datasets: FEMNIST (Cohen et al., 2017; Caldas et al., 2018) and CIFAR100 (Krizhevsky, 2012), and investigate the influence of local training for different attack approaches w.r.t. different numbers of epochs E, batch sizes B, and local data size N. In particular, we mainly investigate the privacy risk of small clients with $N \leq 50$. No existing reconstruction attack recovers large data sizes (e.g. a few hundred). On the other hand, small clients, e.g. edge devices, are common in FL and may have lower amounts of data. We also evaluate on larger N to demonstrate the superiority of our method in a broad range of scenarios. Additionally, we consider clients joining the FL group at different communication rounds R, such that the received network $\boldsymbol{w}_0$ is at different training stages. Following Dimitrov et al. (2022) we conduct attacks on two CNNs for the respective datasets. The networks consist of two convolutional layers followed by two fully-connected layers. We also provide results on other architectures in Appendix G. A detailed description of the datasets, federated learning, and hyperparameters for different attack approaches is given in Appendix D.

Threat Model: To implement our method `SME` and the gradient inversion method `IG`, the adversary only needs to know the model weights $\boldsymbol{w}_0$, $\boldsymbol{w}_T$ of a victim, and its local data size N. We note that clients in FL also need to transmit N to the server due to weighted aggregation (Line 8, Algorithm 2). Therefore, any adversary intercepting the messages or an honest-and-curious server has the ingredients to conduct `SME` and `IG`. Whereas, `DLFA` also needs to know the number of epochs E, batch size B, and learning rate η, so that the simulation can be implemented. This may limit the adversary to the server, which has knowledge of the training protocol. In our comparison experiment, we

	(E=10, B=50, R=1)	(E=50, B=50, R=1)	(E=10, B=10, R=1)	(E=10, B=50, R=20)
$\mathbb{E}[\min_t \frac{\|P_2 \nabla \boldsymbol{w}_t\|}{\|\boldsymbol{w}_t\|}]$	.996±.002	.977±.009	.977±.008	.960±.033
$\mathbb{E}[\text{COSIM} \mid \alpha = 0]$	.984±.008	.867±.041	.872±.037	.825±.055
$\mathbb{E}[\max_{\alpha \in [0,1]} \text{COSIM}]$	.997±.002	.980±.007	.979±.007	.973±.016

Table 1. Statistical evaluations corresponding to Figures 3 and 4. We repeat the experiment 100 times to compute the mean and standard deviation. COSIM is a shorthand for the cosine similarity between $\boldsymbol{w}_0 - \boldsymbol{w}_T$ and $\nabla \widehat{\boldsymbol{w}}$.

Dataset	E	N	T	DLFA $\mathcal{L}_{sim} \downarrow$	DLFA PSNR ↑	IG $\mathcal{L}_{sim} \downarrow$	IG PSNR ↑	SME (ours) $\mathcal{L}_{sim} \downarrow$	SME (ours) PSNR ↑	Δ PSNR
FEMNIST	10	10	10	.021 ± .001	24.9 ± 0.2	.044 ± .002	27.8 ± 0.3	.019 ± .001	**30.3 ± 0.3**	+2.5
	20	10	20	.019 ± .001	25.4 ± 0.2	.090 ± .003	24.2 ± 0.3	.019 ± .001	**28.6 ± 0.3**	+3.2
	50	10	50	.016 ± .002	**26.4 ± 0.3**	.202 ± .005	19.5 ± 0.2	.050 ± .004	25.7 ± 0.3	−0.7
	10	50	50	.015 ± .001	21.7 ± 0.1	.091 ± .003	19.1 ± 0.2	.027 ± .001	**22.2 ± 0.2**	+0.5
	20	50	100	.014 ± .001	**21.7 ± 0.2**	.176 ± .011	17.0 ± 0.2	.037 ± .001	21.5 ± 0.2	−0.2
	50	50	250	.015 ± .001	**20.3 ± 0.2**	.322 ± .016	15.0 ± 0.1	.065 ± .002	20.1 ± 0.2	−0.2
CIFAR100	10	10	10	.017 ± .001	26.7 ± 0.2	.050 ± .001	26.0 ± 0.1	.024 ± .001	**28.5 ± 0.1**	+1.8
	20	10	20	.013 ± .000	26.0 ± 0.2	.102 ± .001	22.3 ± 0.1	.040 ± .001	**26.9 ± 0.1**	+0.9
	50	10	50	.011 ± .000	**24.3 ± 0.2**	.225 ± .002	16.4 ± 0.2	.056 ± .001	24.2 ± 0.1	−0.1
	10	50	50	.023 ± .001	20.3 ± 0.1	.094 ± .001	17.9 ± 0.1	.035 ± .000	**23.5 ± 0.1**	+3.2
	20	50	100	.018 ± .000	19.5 ± 0.1	.154 ± .002	14.8 ± 0.1	.047 ± .001	**21.7 ± 0.1**	+2.2
	50	50	250	N/A	N/A	.280 ± .002	12.1 ± 0.0	.056 ± .001	**18.3 ± 0.1**	N/A

Table 2. Average reconstructed image quality measured by PSNR and similarity loss of the reconstruction objective $\mathcal{L}_{sim}$ on FEMNIST and CIFAR100. For clarity, we set batch size $B = 10$ and change local data sizes N and epochs E. Local steps $T = E\lceil N/B \rceil$. The best reconstruction results are bold. The difference of PSNRs between SME and the best baseline is given in the last column. We remark that for PSNR < 18 the reconstruction will be visually corrupted. Also refer to Figure 5 for visualization. Results of DLFA in the last row is not available, as it needs to allocate 102 Gigabytes of GPU memory, which we cannot support.

still grant DLFA this additional required information.

5.1. Low-Rank Property of Local Steps

In Section 4.2 we show that if the magnitude of high-dimensional gradient mostly concentrates on a 2D plane V_2 then we are able to find a proper surrogate model $\widehat{\boldsymbol{w}}$ on the connected line between $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$ achieving high cosine similarity between $\nabla \widehat{\boldsymbol{w}}$ and $\boldsymbol{w}_0 - \boldsymbol{w}_T$. To support Assumption 4.2 empirically, we collect the true gradients of the local training steps $\{\nabla \boldsymbol{w}_t\}_{t=0}^{T-1}$ and compute the two principal components with the largest two singular values. We let P_2 consist of these two principal components. Then we project the gradients $\{\nabla \boldsymbol{w}_t\}_{t=0}^{T-1}$ onto V_2 using P_2 and compute the norm ratio $\{\|P_2 \nabla \boldsymbol{w}_t\| / \|\nabla \boldsymbol{w}_t\|\}_{t=0}^{T-1}$. Furthermore, we measure the cosine similarity between the connected line $\boldsymbol{w}_0 - \boldsymbol{w}_T$ and surrogate gradient $\nabla \widehat{\boldsymbol{w}}$ along the line.

We sample $N = 50$ data points from FEMNIST and investigate different training situations in terms of epochs E, batch size B, and communication rounds R. Figure 3 shows that in different situations, more than 0.9 of the gradient magnitude concentrates on V_2. This implies that the low-rank property generally holds for the local updates of small clients in FL. Figure 4 shows that there exists a $\nabla \widehat{\boldsymbol{w}}$ with much higher cosine similarity than $\nabla \boldsymbol{w}_0$ some where in the middle of the connected line, while the cosine similarity is always positive for any $\nabla \widehat{\boldsymbol{w}}$. Especially, we observe that the cosine similarity along α has a reversed U-shape which indicates that it is adequate to use an optimization method to find the best α. Statistical evaluations of multiple samplings are presented in Table 1. We give more results of other settings in Appendix E.

5.2. Reconstruction Performance

In this part, we compare the quality of the reconstructed data across different approaches. In previous works, it has been proposed that the label information can be independently recovered with high accuracy by inspecting the weight update of the last linear classifier layer (Zhao et al., 2020; Geng et al., 2021; Dimitrov et al., 2022). Thus, we assume the labels have been recovered, and compare the reconstructed images, which is arguably the primary interest of the adversary. To measure the quality of the reconstructed images, we use the *Peak Signal-to-Noise Ratio* (PSNR). As the reconstructed images may have a different ordering from the original images, *linear sum assignment* is used to find the optimal pairing before computing PSNR. We also present the cosine similarity loss $\mathcal{L}_{sim}$ between the original weight update and dummy gradient (or dummy weight update). For

Figure 5. Visualization of the reconstructed images. The results are drawn from the setting ($E = 20$, $N = 50$, $T = 100$) in Table 2. The reconstructed images are paired with the original images through *linear sum assignment*. We randomly sample 16 out of 50 images of one reconstruction.

clarity, we set batch size $B = 10$ and conduct experiments with different epochs $E \in \{10, 20, 50\}$ and local data size $N \in \{10, 50\}$, such that when $N = 10$ the full gradient is used, and when $N = 50$ stochastic mini-batch gradient is computed. We execute attacks at the first communication round of FL when the network is randomly initialized. We also investigate the attacks at other training stages in Appendix F. For each setting, we run 100 experiments, then compute the mean and standard error of PSNR and $\mathcal{L}_{sim}$.

Based on Table 2, we observe that: (i) Compared with the base method `IG`, our method `SME` achieves significantly lower similarity loss $\mathcal{L}_{sim}$. As a result, `SME` improves the reconstruction quality measured with PSNR by a large margin. The benefit of `SME` is consistent across different settings, which proves that `SME` is an efficient and robust extension to the gradient inversion method in attacking weight update of many iterations; (ii) Compared with `DLFA`, our `SME` obtains competitive results and outperforms `DLFA` in several settings. Additionally, we note that `DLFA` usually achieves lower $\mathcal{L}_{sim}$ than `SME`, even with lower reconstruction quality. This may indicate that there are multiple possible transitions from $\boldsymbol{w}_0$ to $\boldsymbol{w}_T$ with different data. Fitting weight updates with simulation does not necessarily reconstruct the original data.

Figure 5 shows the reconstruction samples of the setting ($E = 20$, $N = 50$, $T = 100$). As we can see, `IG` fails in reconstruction when there are 100 local steps, as the main features of the images are corrupted and the reconstruction contains many artifacts. Whereas, our `SME` revives the gradient inversion method and delivers high-quality reconstruction which is competitive to or better than `DLFA`. We provide more visualizations in Appendix H.

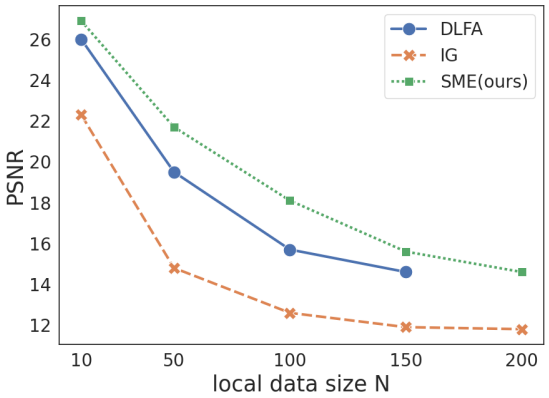


Figure 6. Average reconstructed image quality measured by PSNR vs. local data size N. The setting is (E = 20, B=10) on CIFAR100 from Table 2, we extend N from 50 to 200. Experiments are repeated 100 times then we compute the mean. DLFA requires more than 100 Gigabytes of GPU memory for $N = 200$, which we cannot support.

To evaluate the effectiveness of `SME` on larger local dataset, we conduct experiments with N up to 200. Figure 6 shows

Dataset	E	T	DLFA		IG		SME (ours)		Ratio	
			Mem.	Time	Mem.	Time	Mem.	Time	Mem.	Time
FEMNIST	10	50	2.1	5.9	1.8	.29	1.8	.29	1.2×	20×
	20	100	2.4	11.8	1.8	.29	1.8	.29	1.3×	41×
	50	100	3.9	30.1	1.8	.29	1.8	.29	2.2×	103×
CIFAR100	10	50	11.1	33.2	2.8	1.6	2.8	1.6	7×	21×
	20	100	29.9	67.6	2.8	1.6	2.8	1.6	19×	42×
	50	250	102*	N/A	2.8	1.6	2.8	1.6	64×	N/A

Table 3. Computation resource required by different approaches. We present the GPU memory footprint in Gigabytes and computational time in Hours for 100 times of attack executed on a V100 GPU card. All approaches are implemented in *JAX* (Bradbury et al., 2018). We disable the preallocation behavior and exclude the compilation time. We present the measurements of the experiments in Table 2 with local data size $N = 50$. Note that SME and IG run $2.5\times$ reconstruction iterations than DLFA, but still, run up to $100\times$ faster. The runtime of DLFA in the last row is not available as it requires 102 Gigabytes of GPU memory estimated by *JAX*, which we cannot support.

that: (i) as N increases, the reconstruction of all methods becomes worse as the search space expands. (ii) SME consistently improves over the vanilla gradient inversion method and obtains better results compared to the SOTA baseline DLFA.

5.3. Computational Efficiency

As discussed in Section 3.3, the algorithm of `DLFA` demands a large amount of computational resources as the number of local steps increases, while both `SME` and `IG` do not involve a long chain of simulation. We also show in Section 4.3 that the cost of optimizing the surrogate model in `SME` is negligible. To demonstrate this empirically, we present the runtime and memory footprint of different approaches. The results are shown in Table 3. We see both `SME` and `IG` have a consistent and small runtime across different settings. In particular, our method `SME` achieves the SOTA performance on reconstruction while running up to $100\times$ faster than the SOTA baseline `DLFA`. We remark that the runtime of `SME` is nearly the same as `IG`. More precisely, the increase due to surrogate model optimization is less than 1% of the cost. Additionally, `SME` and `IG` both have moderate and consistent (w.r.t. local steps T) memory footprint. While GPU memory allocated by `DLFA` increases dramatically when the dimension of data grows or the number of local steps increases. This demand of `DLFA` may limit its application to some simple tasks. In contrast, `SME` is applicable on typical hardware and achieves the SOTA performance while running up to $100\times$ faster than `DLFA`, which shows the potential to launch large-scale attacks.

6. Discussion and Conclusion

In this work, we propose a surrogate model extension for gradient inversion. Our method is based on our key insights of the characteristic of 2D gradient flow and the low-rank property of local training steps. We analyze our method theoretically and empirically verify that our method largely strengthens the ability of gradient inversion in attacking weight updates of many training iterations. Compared with the SOTA baseline `DLFA`, our method `SME` achieves the SOTA performance of reconstruction while running up to $100\times$ faster and demanding less GPU memory. These indicate that adversaries can in fact launch effective attacks using low-end devices.

Limitations and Future Work. In this work, we consider that federated optimization is performed with a standard SGD optimizer. Some adaptive optimizer, e.g. Adam (Kingma & Ba, 2014), may lead to different optimization properties and thus interfere with `SME`. This needs further investigation. On the other side, incorporating a more informative prior may empower `SME` to reconstruct larger datasets or input, which has succeeded in vanilla gradient inversion scenarios (Yin et al., 2021; Jeon et al., 2021). Moreover, rather than interpolation of $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$, there may exist other construction of $\mathcal{W}$ that makes the general objective of `SME` effective, i.e. Equation (4). We encourage further study in these directions.

Societal Impact. We believe our effort in studying adversarial attack and presenting it to the open community is beneficial to society. We emphasize that our study does not attempt to claim that federated learning is completely insecure, since it certainly provides a layer of privacy protection by avoiding data collection. Instead, our work intends to evaluate the true risk hiding in the weight sharing in collaborative learning schemes. We hope our research will motivate further research of defense against privacy attacks.

Acknowledgements. This research received funding from the Flemish Government (AI Research Program) and the Research Foundation - Flanders (FWO) through project number G0G2921N. Ruicong Yao would like to thank the guidance and encouragement from Tim Verdonck and Jakob Raymaekers in this research process.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

Atkinson, K. *An introduction to numerical analysis*. John wiley & sons, 1991.

Balunovic, M., Dimitrov, D. I., Staab, R., and Vechev, M. Bayesian framework for gradient leakage. In *International Conference on Learning Representations*, 2022.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pp. 464–473, 2014.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pp. 1175–1191, New York, NY, USA, 2017. Association for Computing Machinery.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. doi: 10.1137/16M1080173. URL https://doi.org/10.1137/16M1080173.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Carlini, N., Liu, C., Erlingsson, U., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pp. 267–284, USA, 2019. USENIX Association.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Oprea, A., and Raffel, C. Extracting training data from large language models, 2020.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership inference attacks from first principles. *arXiv preprint arXiv:2112.03570*, 2021.

Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. Label-only membership inference attacks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1964–1974. PMLR, 18–24 Jul 2021.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Dimitrov, D. I., Balunovic, M., Konstantinov, N., and Vechev, M. Data leakage in federated averaging. *Transactions on Machine Learning Research*, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9:211–407, 2014.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pp. 265–284, 2006.

Fan, L., Ng, K. W., Ju, C., Zhang, T., Liu, C., Chan, C. S., and Yang, Q. Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks. In Yang, Q., Fan, L., and Yu, H. (eds.), *Federated Learning: Privacy and Incentive*, pp. 32–50. Springer, 2020.

Fredrikson, M., Jha, S., and Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pp. 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery.

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. Inverting gradients - how easy is it to break privacy in federated learning? In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16937–16947. Curran Associates, Inc., 2020.

Geng, J., Mou, Y., Li, F., Li, Q., Beyan, O., Decker, S., and Rong, C. Towards general deep leakage in federated learning, 2021.

Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. Cryptonets: Applying neural networks to encrypted data with high throughput

and accuracy. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 201–210, New York, New York, USA, 20–22 Jun 2016. PMLR.

Gooneratne, M., Sim, K. C., Zadrazil, P., Kabel, A., Beaufays, F., and Motta, G. Low-rank gradient approximation for memory-efficient on-device training of deep neural network. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3017–3021, 2020.

Haim, N., Vardi, G., Yehudai, G., michal Irani, and Shamir, O. Reconstructing training data from trained neural networks. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Los Alamitos, CA, USA, 2016. IEEE Computer Society.

He, Z., Zhang, T., and Lee, R. B. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, pp. 148–162, New York, NY, USA, 2019. Association for Computing Machinery.

Huang, Y., Gupta, S., Song, Z., Li, K., and Arora, S. Evaluating gradient inversion attacks and defenses in federated learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Jeon, J., Kim, J., Lee, K., Oh, S., and Ok, J. Gradient inversion with generative image prior. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Jin, X., Chen, P.-Y., Hsu, C.-Y., Yu, C.-M., and Chen, T. Cafe: Catastrophic data leakage in vertical federated learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 994–1006. Curran Associates, Inc., 2021.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and Open Problems in Federated Learning. *arXiv:1912.04977 [cs, stat]*, March 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A. Learning multiple layers of features from tiny images. *University of Toronto*, 2012.

Lam, M., Wei, G.-Y., Brooks, D., Reddi, V. J., and Mitzenmacher, M. Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5959–5968. PMLR, 18–24 Jul 2021.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, J.-W., Kang, H., Lee, Y., Choi, W., Eom, J., Deryabin, M., Lee, E., Lee, J., Yoo, D., Kim, Y.-S., and No, J.-S. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, 10:30039–30054, 2022.

Li, X., Gu, Q., Zhou, Y., Chen, T., and Banerjee, A. Hessian based analysis of sgd for deep nets: Dynamics and generalization. In *Proceedings of the 2020 SIAM International Conference on Data Mining (SDM)*, pp. 190–198, 2020.

Li, X., Liu, D., Hashimoto, T., Inan, H. A., Kulkarni, J., Lee, Y., and Thakurta, A. G. When does differentially private learning not suffer in high dimensions? In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Lu, J., Zhang, X. S., Zhao, T., He, X., and Cheng, J. April: Finding the achilles' heel on privacy for vision transformers, 2021.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017.

Papernot, N., Thakurta, A., Song, S., Chien, S., and Erlingsson, Ú. Tempered sigmoid activations for deep learning with differential privacy. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 9312–9321, 2021.

Phong, L. T., Aono, Y., Hayashi, T., Wang, L., and Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

So, J., Guler, B., and Avestimehr, A. S. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. *CoRR*, abs/2002.04156, 2020.

Tramer, F. and Boneh, D. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

Tropp, J. A. et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Vogels, T., Karimireddy, S. P., and Jaggi, M. *PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization*. 2019.

Wang, D. and Xu, J. Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1182–1189, 2019.

Wei, W., Liu, L., Loper, M. L., Chow, K. H., Gursoy, M. E., Truex, S., and Wu, Y. A framework for evaluating client privacy leakages in federated learning. In *European Symposium on Research in Computer Security*, 2020.

Yang, Z., Zhang, J., Chang, E.-C., and Liang, Z. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pp. 225–240, New York, NY, USA, 2019. Association for Computing Machinery.

Yin, H., Mallya, A., Vahdat, A., Alvarez, J. M., Kautz, J., and Molchanov, P. See through gradients: Image batch recovery via gradinversion. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16332–16341, Los Alamitos, CA, USA, jun 2021.

Zhao, B., Mopuri, K. R., and Bilen, H. iDLG: Improved deep leakage from gradients. arXiv:2001.02610, 2020.

Zhou, Y., Wu, S., and Banerjee, A. Bypassing the ambient dimension: Private SGD with gradient subspace identification. In *International Conference on Learning Representations*, 2021.

Zhu, J. and Blaschko, M. B. R-GAP: Recursive gradient attack on privacy. In *International Conference on Learning Representations*, 2021.

Zhu, L., Liu, Z., and Han, S. Deep leakage from gradients. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

A. Related Works

In addition to gradient inversion methods, another line of work aims to extract training data from a trained network (Fredrikson et al., 2015; Yang et al., 2019; He et al., 2019; Carlini et al., 2019; 2020; Haim et al., 2022). Many other works focus on extracting partial information from the data, e.g. label-only inference and membership inference (Shokri et al., 2017; Zhao et al., 2020; Carlini et al., 2021; Choquette-Choo et al., 2021).

On the other hand, recently many defense strategies and privacy-enhancing methods have emerged. Differential privacy provides a rigorous definition of privacy based on a probabilistic perspective (Dwork et al., 2006; Dwork & Roth, 2014; Bassily et al., 2014). A long line of works adopts it in deep learning to alleviate the privacy leakage from the model or intermediate training products like weights, to name a few (Abadi et al., 2016; Papernot et al., 2021; Tramer & Boneh, 2021). However, differential privacy requires injecting extensive Gaussian noise into the gradients and thus impedes the model performance (Bassily et al., 2014; Wang & Xu, 2019). Moreover, many works exploit encryption technology (Gilad-Bachrach et al., 2016; Bonawitz et al., 2017; So et al., 2020; Lee et al., 2022), such that messages communicated between participants are in cipher text, thus adversaries cannot extract the sensitive information directly. However, these methods in general give rise to a significant additional computational overhead.

B. Reconstruction Gradient of Simulation Methods

Suppose that for each simulation step, $\tilde{D}$ is used to train the network, and consider the following loss function of a simulation method:

$$\begin{aligned}\mathcal{L} &= \|\tilde{\boldsymbol{w}}_T - \boldsymbol{w}_0 - (\boldsymbol{w}_T - \boldsymbol{w}_0)\| &(12)\\ &= \|\eta(\nabla \boldsymbol{w}_0 + \nabla\tilde{\boldsymbol{w}}_1 + \cdots + \nabla\tilde{\boldsymbol{w}}_{T-1}) - \boldsymbol{w}_T + \boldsymbol{w}_0\|, &(13)\end{aligned}$$

the reconstruction gradient $\nabla_{\tilde{D}}\mathcal{L}$ can be derived as:

$$\nabla_{\tilde{D}}\mathcal{L} = \frac{\partial\mathcal{L}}{\partial\nabla\boldsymbol{w}_0}\frac{\partial\nabla\boldsymbol{w}_0}{\partial\tilde{D}} + \frac{\partial\mathcal{L}}{\partial\nabla\tilde{\boldsymbol{w}}_1}\frac{\partial\nabla\tilde{\boldsymbol{w}}_1}{\partial\tilde{D}} + \cdots + \frac{\partial\mathcal{L}}{\partial\nabla\tilde{\boldsymbol{w}}_{T-1}}\frac{\partial\nabla\tilde{\boldsymbol{w}}_{T-1}}{\partial\tilde{D}}. \tag{14}$$

Using the chain rule, we have for $\frac{\partial\nabla\tilde{\boldsymbol{w}}_1}{\partial\tilde{D}}$:

$$\begin{aligned}\frac{\partial\nabla\tilde{\boldsymbol{w}}_1}{\partial\tilde{D}} &= \frac{\partial\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w},\tilde{D})}{\partial\tilde{D}}|_{\boldsymbol{w}=\tilde{\boldsymbol{w}}_1} + \frac{\partial\nabla\tilde{\boldsymbol{w}}_1}{\partial\tilde{\boldsymbol{w}}_1}\frac{\partial\tilde{\boldsymbol{w}}_1}{\partial\nabla\tilde{\boldsymbol{w}}_0}\frac{\partial\nabla\boldsymbol{w}_0}{\partial\tilde{D}} &(15)\\ &= \frac{\partial\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w},\tilde{D})}{\partial\tilde{D}}|_{\boldsymbol{w}=\tilde{\boldsymbol{w}}_1} - \eta\frac{\partial\tilde{\boldsymbol{w}}_1}{\partial\nabla\tilde{\boldsymbol{w}}_1}\frac{\partial\nabla\boldsymbol{w}_0}{\partial\tilde{D}}, &(16)\end{aligned}$$

where Equation (16) uses the fact that $\tilde{\boldsymbol{w}}_1 = \boldsymbol{w}_0 - \eta\nabla\boldsymbol{w}_0$. Then for the reconstruction gradient led by $\nabla\boldsymbol{w}_t$, considering that $\tilde{\boldsymbol{w}}_t = \boldsymbol{w}_0 - \eta(\nabla\boldsymbol{w}_0 + \cdots + \nabla\tilde{\boldsymbol{w}}_{t-1})$, we have:

$$\frac{\partial\nabla\tilde{\boldsymbol{w}}_t}{\partial\tilde{D}} = \frac{\partial\nabla\ell_{\boldsymbol{w}}(\boldsymbol{w},\tilde{D})}{\partial\tilde{D}}|_{\boldsymbol{w}=\tilde{\boldsymbol{w}}_t} - \eta\frac{\partial\nabla\tilde{\boldsymbol{w}}_t}{\partial\tilde{\boldsymbol{w}}_t}(\frac{\partial\nabla\boldsymbol{w}_0}{\partial\tilde{D}} + \frac{\partial\nabla\tilde{\boldsymbol{w}}_1}{\partial\tilde{D}} + \cdots + \frac{\partial\nabla\tilde{\boldsymbol{w}}_{t-1}}{\partial\tilde{D}}) \tag{17}$$

Substituting Equation (17) into Equation (14) for each $t = 1 \ldots T-1$ and rearranging, we have:

$$\begin{aligned}\nabla_{\tilde{D}}\mathcal{L} = &-\eta\frac{\partial\mathcal{L}}{\partial\tilde{\boldsymbol{w}}_{T-1}}(\frac{\partial\nabla\boldsymbol{w}_0}{\partial\tilde{D}} + \frac{\partial\nabla\tilde{\boldsymbol{w}}_1}{\partial\tilde{D}} + \cdots + \frac{\partial\nabla\tilde{\boldsymbol{w}}_{T-3}}{\partial\tilde{D}} + \frac{\partial\nabla\tilde{\boldsymbol{w}}_{T-2}}{\partial\tilde{D}})\\ &-\eta\frac{\partial\mathcal{L}}{\partial\tilde{\boldsymbol{w}}_{T-2}}(\frac{\partial\nabla\boldsymbol{w}_0}{\partial\tilde{D}} + \frac{\partial\nabla\tilde{\boldsymbol{w}}_1}{\partial\tilde{D}} + \cdots + \frac{\partial\nabla\tilde{\boldsymbol{w}}_{T-3}}{\partial\tilde{D}})\\ &\vdots\\ &-\eta\frac{\partial\mathcal{L}}{\partial\tilde{\boldsymbol{w}}_1}\frac{\partial\nabla\boldsymbol{w}_0}{\partial\tilde{D}}\\ &+\frac{\partial\mathcal{L}}{\partial\nabla\tilde{\boldsymbol{w}}_0}\frac{\partial\nabla\ell_{\boldsymbol{w}}(\boldsymbol{w},\tilde{D})}{\partial\tilde{D}}|_{\boldsymbol{w}=\tilde{\boldsymbol{w}}_0} + \frac{\partial\mathcal{L}}{\partial\nabla\tilde{\boldsymbol{w}}_1}\frac{\partial\nabla\ell_{\boldsymbol{w}}(\boldsymbol{w},\tilde{D})}{\partial\tilde{D}}|_{\boldsymbol{w}=\tilde{\boldsymbol{w}}_1} + \cdots + \frac{\partial\mathcal{L}}{\partial\nabla\tilde{\boldsymbol{w}}_{T-1}}\frac{\partial\nabla\ell_{\boldsymbol{w}}(\boldsymbol{w},\tilde{D})}{\partial\tilde{D}}|_{\boldsymbol{w}=\tilde{\boldsymbol{w}}_{T-1}}.\end{aligned} \tag{18}$$

Equation (18) shows that the reconstruction gradient $\nabla_{\tilde{D}}\mathcal{L}$ in the simulation method has a complexity that increases with the number of local steps T.

C. Proofs

In this section we give the proofs of all the theoretical results in Section 4. Recall Assumption 4.2 in Section 4, we say a function $\ell(\boldsymbol{w}) : \mathbb{R}^p \to \mathbb{R}^q$ is L-Lipschitz if for $\forall \boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^p$, it holds that

$$||\ell(\boldsymbol{w}) - \ell(\boldsymbol{v})|| \leq L||\boldsymbol{w} - \boldsymbol{v}||. \tag{19}$$

We say a function $\ell : \mathbb{R}^p \to \mathbb{R}$ is γ-strongly convex if for $\forall \boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^p$, it holds that,

$$\ell(\boldsymbol{w}) \geq \ell(\boldsymbol{v}) + \langle \nabla \boldsymbol{v}, \boldsymbol{w} - \boldsymbol{v} \rangle + \frac{\gamma}{2}||\boldsymbol{w} - \boldsymbol{v}||^2. \tag{20}$$

A well-known property of strongly convex functions is that (cf. (Bottou et al., 2018)) for $\forall \boldsymbol{w} \in \mathbb{R}^p$ and global minimum $\boldsymbol{w}^*$, it holds that

$$2\gamma(\ell(\boldsymbol{w}) - \ell(\boldsymbol{w}^*)) \leq ||\nabla \ell(\boldsymbol{w})||^2. \tag{21}$$

C.1. Proof of Proposition 4.1

The basic idea has been given in Section 4.1. Here, we further illustrate how we prove the case when both $\nabla \boldsymbol{w}_0, \nabla \boldsymbol{w}_T$ point to the right hand side in Figure 1. Since $\boldsymbol{w}_T$ is the end point of the gradient flow, we can argue that there exists a model $\boldsymbol{w}_{t'}$ on the gradient flow near $\boldsymbol{w}_T$ located at the left hand side of $\boldsymbol{w}_T - \boldsymbol{w}_0$. Therefore, the intersection of the gradient flow and the connected line is non-empty. We can further deduce that there exists at least one surrogate model $\widehat{\boldsymbol{w}}'$ on the connected line such that its gradient $\nabla \widehat{\boldsymbol{w}}'$ points to the left hand side of $\boldsymbol{w}_T - \boldsymbol{w}_0$ or is parallel to $\boldsymbol{w}_T - \boldsymbol{w}_0$. Thus in both situations, we can find some $\widehat{\boldsymbol{w}}$ on the connected line between $\widehat{\boldsymbol{w}}'$ and $\boldsymbol{w}_0$ such that its gradient is parallel to $\boldsymbol{w}_T - \boldsymbol{w}_0$. A mathematical proof is given in the following.

Proof of Proposition 4.1. To be aligned with the illustration in Section 4.1, we prove for the gradient ascent case from $\boldsymbol{w}_0$ to $\boldsymbol{w}_T$. Here, the condition would instead be $\langle \nabla \boldsymbol{w}_t, \boldsymbol{w}_T - \boldsymbol{w}_0 \rangle \geq 0, \forall t \in [0, T]$ (we reverse w_0 and w_T) and the second claim would be $\nabla \widehat{\boldsymbol{w}} \parallel \boldsymbol{w}_T - \boldsymbol{w}_0$ assuming convexity. The proof for the gradient descent flow is similar. We do not consider the case that there is a $\boldsymbol{w}^*$ on the connected line such that $\nabla \boldsymbol{w}^* = 0$, which is typically unusual. Let $\boldsymbol{w}^\perp$ be a vector which is orthogonal to $\boldsymbol{w}_T - \boldsymbol{w}_0$. Our first claim immediately follows if

$$\langle \nabla \boldsymbol{w}_0, \boldsymbol{w}^\perp \rangle \cdot \langle \nabla \boldsymbol{w}_T, \boldsymbol{w}^\perp \rangle \leq 0, \tag{22}$$

since $\nabla \widehat{\boldsymbol{w}}$ is continuous on the connected line so that the inner product attains zero for some choice of α.

Otherwise, we assume without loss of generality that both inner products are positive. Mathematically it means that there exists a neighbourhood $\mathcal{O}_{\boldsymbol{w}_T}$ of $\boldsymbol{w}_T$ such that $\langle \nabla \boldsymbol{w}_t, \boldsymbol{w}^\perp \rangle > 0$ for all $\boldsymbol{w}_t \in \mathcal{O}_{\boldsymbol{w}_T}$. Moreover for such $\boldsymbol{w}_t$,

$$\langle \boldsymbol{w}_t - \boldsymbol{w}_0, \boldsymbol{w}^\perp \rangle = \langle \boldsymbol{w}_T - \boldsymbol{w}_0, \boldsymbol{w}^\perp \rangle - \int_t^T \langle \nabla \boldsymbol{w}_s, \boldsymbol{w}^\perp \rangle ds < 0. \tag{23}$$

On the other hand, we can find a neighbour of $\mathcal{O}_{\boldsymbol{w}_0}$ of $\boldsymbol{w}_0$ such that $\langle \nabla \boldsymbol{w}_t, \boldsymbol{w}^\perp \rangle > 0$ and $\langle \boldsymbol{w}_t - \boldsymbol{w}_0, \boldsymbol{w}^\perp \rangle > 0$ for $\forall \boldsymbol{w}_t \in \mathcal{O}_{\boldsymbol{w}_0}$. Combining these two facts, we know by continuity that there exists a non-empty set A of all α s.t. $\alpha \in \mathbb{R}$ and $\widehat{\boldsymbol{w}}_\alpha := \alpha \boldsymbol{w}_0 + (1 - \alpha)\boldsymbol{w}_T$ is on the gradient flow. By the assumption that $\langle \nabla \boldsymbol{w}_t, \boldsymbol{w}_T - \boldsymbol{w}_0 \rangle \geq 0, \forall t \in [0, T]$ (in the gradient ascent flow case), the projection of $\boldsymbol{w}_t$ onto $\boldsymbol{w}_T - \boldsymbol{w}_0$ is always an interpolation of $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$. Thus we have $A \subset (0, 1)$.

We now assume that $\langle \nabla \widehat{\boldsymbol{w}}_\alpha, \boldsymbol{w}^\perp \rangle > 0$ for $\forall \alpha \in A$ and prove by contradiction. Let $\alpha_{\inf}$ be the infimum of A and t_α be such that $\boldsymbol{w}_{t_\alpha} = \widehat{\boldsymbol{w}}_\alpha$. By continuity we have either $\langle \nabla \widehat{\boldsymbol{w}}_{\alpha_{\inf}}, \boldsymbol{w}^\perp \rangle = 0$ or $\alpha_{\inf} \in A$. We continue with the latter. If $\alpha_{\inf} > 0$, then by a similar argument as before we can show that the gradient flow and the connected line intersects with each other for some $0 < t' < t_{\alpha_{\inf}}$, which is a contradiction. If otherwise $\alpha_{\inf} = 0$ we can find an $\alpha' \in A$ s.t. $\boldsymbol{w}_{t_{\alpha'}} \in \mathcal{O}_{\boldsymbol{w}_0}$. Thus by our assumption on $\nabla \widehat{\boldsymbol{w}}_\alpha$, there exists a $0 < t' < t_{\alpha'}$ s.t. $\langle \boldsymbol{w}_{t'} - \boldsymbol{w}_0, \boldsymbol{w}^\perp \rangle < 0$ by a similar argument to Equation (23), which contradicts the property of $\mathcal{O}_{\boldsymbol{w}_0}$.

Therefore in any case, there exists an α s.t. $\langle \nabla \widehat{\boldsymbol{w}}_\alpha, \boldsymbol{w}^\perp \rangle \leq 0$ where $\boldsymbol{w}_\alpha$ is on the connected line and the gradient flow. Then, the problem is reduced to the basic one mentioned at the beginning.

Finally, we show $\nabla\widehat{\boldsymbol{w}}_\alpha \parallel \boldsymbol{w}_T - \boldsymbol{w}_0$ under convexity and gradient ascent flow. Otherwise, we fix an α s.t. $\langle \nabla\widehat{\boldsymbol{w}}_\alpha, \boldsymbol{w}_T - \boldsymbol{w}_0\rangle = -1$. We can thus identify an $\alpha' \in (0, \alpha)$ s.t. $\ell(\widehat{\boldsymbol{w}}_\alpha) < \ell(\widehat{\boldsymbol{w}}_{\alpha'})$ due to

$$\ell(\widehat{\boldsymbol{w}}_\alpha) - \ell(\widehat{\boldsymbol{w}}_{\alpha'}) = \int_0^1 \langle \nabla((1-s)\widehat{\boldsymbol{w}}_{\alpha'} + s\widehat{\boldsymbol{w}}_\alpha), \widehat{\boldsymbol{w}}_\alpha - \widehat{\boldsymbol{w}}_{\alpha'}\rangle ds \tag{24}$$

and the continuity of the gradient. On the other hand, since $\langle \nabla \boldsymbol{w}_0, \boldsymbol{w}_T - \boldsymbol{w}_0\rangle \geq 0$, the loss along the connected line first increases. Therefore, we can identify $0 \leq \alpha_1 \leq \alpha_2 \leq \alpha_3 \leq 1$ such that $\ell(\widehat{\boldsymbol{w}}_{\alpha_2}) > \ell(\widehat{\boldsymbol{w}}_{\alpha_1})$ and $\ell(\widehat{\boldsymbol{w}}_{\alpha_2}) > \ell(\widehat{\boldsymbol{w}}_{\alpha_3})$, while $\widehat{\boldsymbol{w}}_{\alpha_2}$ is an interpolation of $\widehat{\boldsymbol{w}}_{\alpha_1}$ and $\widehat{\boldsymbol{w}}_{\alpha_3}$ This contradicts the convexity. □

By a similar argument to the last paragraph of the proof, we can show a property of gradient flow on convex objects:

Corollary C.1. *Suppose $\boldsymbol{w}_t$, $t \in [0, T]$ is a gradient flow based on a convex loss $\ell(\boldsymbol{w})$, then it holds that $\langle \nabla\widehat{\boldsymbol{w}}_\alpha, \boldsymbol{w}_0 - \boldsymbol{w}_T\rangle \geq 0$, where $\widehat{\boldsymbol{w}}_\alpha := \alpha\boldsymbol{w}_0 + (1-\alpha)\boldsymbol{w}_T$, $\alpha \in [0, 1]$.*

C.2. Proof of Theorem 4.3

We first state a simple lemma which helps the proof of the theorem.

Lemma C.2. *Let $\nabla\boldsymbol{w} \in \mathbb{R}^p$, $p \geq 2$ and P_2 satisfy Equation (6) in Assumption 4.2. Let $\boldsymbol{v} \in \mathbb{R}^d$ be another vector. It holds that*

$$\langle P_2\nabla\boldsymbol{w}, P_2\boldsymbol{v}\rangle = \langle \nabla\boldsymbol{w}, \boldsymbol{v}\rangle - \langle P_2^\perp\nabla\boldsymbol{w}, P_2^\perp\boldsymbol{v}\rangle \geq (\text{COSIM}(\nabla\boldsymbol{w}, \boldsymbol{v}) - G_2)||\nabla\boldsymbol{w}||||\boldsymbol{v}||. \tag{25}$$

Here COSIM *stands for the cosine similarity between the two vectors which is equal to $1 - \mathcal{L}_{sim}$.*

The proof of the lemma is straightforward. Now we prove the theorem. We begin by first identifying a surrogate model $\widehat{\boldsymbol{w}}$ that exhibits a large cosine similarity $\text{COSIM}(P_2\widehat{\boldsymbol{w}}, P_2(\boldsymbol{w}_0 - \boldsymbol{w}_T))$. Subsequently, we establish a lower bound for $\text{COSIM}(\widehat{\boldsymbol{w}}, \boldsymbol{w}_0 - \boldsymbol{w}_T)$ (or equivalently an upper bound for $\mathcal{L}_{sim}(\widehat{\boldsymbol{w}}, \boldsymbol{w}_0 - \boldsymbol{w}_T))$) in high-dimensional space.

Proof of Theorem 4.3. Since Proposition 4.1 only works for 2D continuous gradient flow, we shall discuss how to adapt it to gradient descent method in high-dimensional space and estimate the error term Δ_{trun} related to Equation (8).

We first try to apply Proposition 4.1 to $P_2\boldsymbol{w}(t)$, which is the projected gradient flow from $P_2\boldsymbol{w}(0) = P_2\boldsymbol{w}_0$, so as to identify an intermediate surrogate model $\widehat{\boldsymbol{w}}'$. Let $\boldsymbol{w}(T\eta)$ be the end point of the gradient flow. By Lemma C.2 and the condition that $\text{COSIM}(\nabla\boldsymbol{w}(t), \boldsymbol{w}(0) - \boldsymbol{w}(T\eta)) > G_2$, we have $\langle P_2\nabla\boldsymbol{w}(t), P_2(\boldsymbol{w}(0) - \boldsymbol{w}(T\eta))\rangle > 0$. Thus Proposition 4.1 shows that there exists a surrogate model $\widehat{\boldsymbol{w}}'$ satisfying $\langle P_2\widehat{\boldsymbol{w}}', P_2(\boldsymbol{w}(0) - \boldsymbol{w}(T\eta))\rangle \in \{\pm 1\}$. Moreover, by Assumption 4.2 (convexity and low rank property), Corollary C.1 and Lemma C.2, the surrogate model satisfies $\text{COSIM}(P_2\widehat{\boldsymbol{w}}', P_2(\boldsymbol{w}(0) - \boldsymbol{w}(T\eta))) > -G_2 > -1$. Therefore the only possibility is

$$\langle P_2\widehat{\boldsymbol{w}}', P_2(\boldsymbol{w}(0) - \boldsymbol{w}(T\eta))\rangle = 1, \tag{26}$$

which makes $P_2\widehat{\boldsymbol{w}}'$ an ideal surrogate model for the projected gradient flow.

Next we identify a surrogate model $\widehat{\boldsymbol{w}}$ for the discrete GD path which is related to $\widehat{\boldsymbol{w}}'$. According to the truncation error of the Euler method (Atkinson, 1991), it holds that $||P_2\boldsymbol{w}(T\eta) - P_2\boldsymbol{w}_T|| = O(\eta)$. Thus we expect that the GD curve can well approximate the gradient flow. We can then upper bound the angle θ_1 between $P_2(\boldsymbol{w}_T - \boldsymbol{w}_0)$ and $P_2(\boldsymbol{w}(T\eta) - \boldsymbol{w}_0)$ by $\arcsin\left(\frac{||P_2(\boldsymbol{w}(T\eta) - \boldsymbol{w}_T)||}{||P_2(\boldsymbol{w}_T - \boldsymbol{w}_0)||}\right)$. Finally, we can identify a $\widehat{\boldsymbol{w}}$ as an interpolation of $\boldsymbol{w}_0$ and $\boldsymbol{w}_T$ and estimate $\text{COSIM}(P_2\widehat{\boldsymbol{w}}, P_2(\boldsymbol{w}_0 - \boldsymbol{w}_T))$ using $\widehat{\boldsymbol{w}}'$ and our assumptions (smoothness, strong convexity). This defines the error

$$\Delta_{trun} := 1 - \text{COSIM}(P_2\widehat{\boldsymbol{w}}, P_2(\boldsymbol{w}_0 - \boldsymbol{w}_T)) \geq 0, \tag{27}$$

which is a function of G_2, η, β, L and the loss. This quantity is expected to be close to 0 if the GD curve approximates the gradient flow well. Since this final step is very similar to the proof of Theorem 4.5 and no extra parameters would be required, we do not provide the details for brevity.

Now we estimate $\mathcal{L}_{sim}(\widehat{\boldsymbol{w}}, \boldsymbol{w}_0 - \boldsymbol{w}_T)$. By the identity $\langle \nabla\widehat{\boldsymbol{w}}, \boldsymbol{w}_T - \boldsymbol{w}_0\rangle = \langle P_2\nabla\widehat{\boldsymbol{w}}, P_2(\boldsymbol{w}_T - \boldsymbol{w}_0)\rangle + \langle P_2^\perp\nabla\widehat{\boldsymbol{w}}, P_2^\perp(\boldsymbol{w}_T -$

$\boldsymbol{w}_0)\rangle$ and assumption 4.2, it holds that

$$1-\mathcal{L}_{sim}(\widehat{\boldsymbol{w}},\boldsymbol{w}_0-\boldsymbol{w}_T)=\frac{\langle\nabla\widehat{\boldsymbol{w}},\boldsymbol{w}_0-\boldsymbol{w}_T\rangle}{||\nabla\widehat{\boldsymbol{w}}||||\boldsymbol{w}_T-\boldsymbol{w}_0||} \tag{28}$$

$$\geq\frac{|\langle P_2\nabla\widehat{\boldsymbol{w}},P_2(\boldsymbol{w}_T-\boldsymbol{w}_0)\rangle|}{||\nabla\widehat{\boldsymbol{w}}||||\boldsymbol{w}_T-\boldsymbol{w}_0||}-\frac{|\langle P_2^{\perp}\nabla\widehat{\boldsymbol{w}},P_2^{\perp}(\boldsymbol{w}_T-\boldsymbol{w}_0)\rangle|}{||\nabla\widehat{\boldsymbol{w}}||||\boldsymbol{w}_t-\boldsymbol{w}_0||} \tag{29}$$

$$\geq(1-\Delta_{trun})\frac{(1-G_2^2)||\nabla\widehat{\boldsymbol{w}}||||P_2(\boldsymbol{w}_T-\boldsymbol{w}_0)||}{||\nabla\widehat{\boldsymbol{w}}||||\boldsymbol{w}_T-\boldsymbol{w}_0||}-\frac{||P_2^{\perp}\nabla\widehat{\boldsymbol{w}}||}{||\nabla\widehat{\boldsymbol{w}}||}\frac{||P_2^{\perp}(\boldsymbol{w}_T-\boldsymbol{w}_0)||}{||\boldsymbol{w}_T-\boldsymbol{w}_0||} \tag{30}$$

$$=(1-\Delta_{trun})(1-G_2^2)\frac{||P_2(\boldsymbol{w}_T-\boldsymbol{w}_0)||}{||\boldsymbol{w}_T-\boldsymbol{w}_0||}-G_2\frac{||P_2^{\perp}(\boldsymbol{w}_T-\boldsymbol{w}_0)||}{||\boldsymbol{w}_T-\boldsymbol{w}_0||} \tag{31}$$

$$\geq(1-G_2^2)\frac{||P_2(\boldsymbol{w}_T-\boldsymbol{w}_0)||}{||\boldsymbol{w}_T-\boldsymbol{w}_0||}-G_2\frac{||P_2^{\perp}(\boldsymbol{w}_T-\boldsymbol{w}_0)||}{||\boldsymbol{w}_T-\boldsymbol{w}_0||}-\Delta_{trun} \tag{32}$$

Next we control the magnitude of $||P_2(\boldsymbol{w}_T-\boldsymbol{w}_0)||$ and $||P_2^{\perp}(\boldsymbol{w}_T-\boldsymbol{w}_0)||$. By Equation (4.3) of Bottou et al. (2018), the Lipchitzness of the gradient (also known as β-smoothness of the loss) implies that for $\forall\boldsymbol{w},\boldsymbol{v}\in\mathbb{R}^p$, it holds that

$$\ell(\boldsymbol{w})\leq\ell(\boldsymbol{v})+\langle\nabla\boldsymbol{v},\boldsymbol{w}-\boldsymbol{v}\rangle+\frac{\beta}{2}||\boldsymbol{w}-\boldsymbol{v}||^2. \tag{33}$$

If we let $\boldsymbol{w}=\boldsymbol{w}_{t+1}$ and $\boldsymbol{v}=\boldsymbol{w}_t$ and use the fact that $||\boldsymbol{w}_{t+1}-\boldsymbol{w}_t||^2=\eta^2||\nabla\boldsymbol{w}_t||^2$, we have

$$0\leq\eta\Big(1-\frac{\beta\eta}{2}\Big)||\nabla\boldsymbol{w}_t||^2\leq\ell(\boldsymbol{w}_t)-\ell(\boldsymbol{w}_{t+1}). \tag{34}$$

Then, by definition we have for $\forall\boldsymbol{w}_t,\boldsymbol{w}_{t+1}$, $0\leq t\leq T-1$,

$$||P_2^{\perp}(\boldsymbol{w}_{t+1}-\boldsymbol{w}_t)||^2=\eta^2||P_2^{\perp}\nabla\boldsymbol{w}_t||^2\leq\eta^2G_2^2||\nabla\boldsymbol{w}_t||^2\leq G^2(\eta,\beta)(\ell(\boldsymbol{w}_t)-\ell(\boldsymbol{w}_{t+1})), \tag{35}$$

so that by triangular and Cauchy-Schwartz inequality we have

$$\frac{||P_2^{\perp}(\boldsymbol{w}_T-\boldsymbol{w}_0)||^2}{||\boldsymbol{w}_T-\boldsymbol{w}_0||^2}\leq\frac{(\sum_{t=0}^{T-1}||P_2^{\perp}(\boldsymbol{w}_{t+1}-\boldsymbol{w}_t)||)^2}{||\boldsymbol{w}_T-\boldsymbol{w}_0||^2} \tag{36}$$

$$\leq\frac{T\sum_{t=0}^{T-1}||P_2^{\perp}(\boldsymbol{w}_{t+1}-\boldsymbol{w}_t)||^2}{||\boldsymbol{w}_T-\boldsymbol{w}_0||^2} \tag{37}$$

$$\leq\frac{TG^2(\eta,\beta)(\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T))}{||\boldsymbol{w}_T-\boldsymbol{w}_0||^2} \tag{38}$$

$$=\frac{TG^2(\eta,\beta)(\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T))^2}{||\boldsymbol{w}_T-\boldsymbol{w}_0||^2(\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T))} \tag{39}$$

$$\leq\frac{TG^2(\eta,\beta)L^2}{\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T)}. \tag{40}$$

Therefore we have

$$\frac{||P_2(\boldsymbol{w}_T-\boldsymbol{w}_0)||}{||\boldsymbol{w}_T-\boldsymbol{w}_0||}=\sqrt{1-\frac{||P_2^{\perp}(\boldsymbol{w}_T-\boldsymbol{w}_0)||^2}{||\boldsymbol{w}_T-\boldsymbol{w}_0||^2}}\geq1-\frac{TG^2(\eta,\beta)L^2}{\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T)}, \tag{41}$$

regardless of the magnitude of Equation (40). Summing up everything we finally conclude that

$$1-\mathcal{L}_{sim}(\boldsymbol{w}_0-\boldsymbol{w}_T,\widehat{\boldsymbol{w}})\geq(1-G_2^2)\left(1-\frac{TG^2(\eta,\beta)L^2}{\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T)}\right)-G_2\sqrt{\frac{TG^2(\eta,\beta)L^2}{\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T)}}-\Delta_{trun} \tag{42}$$

$$\geq1-G_2^2-2G_2\sqrt{\frac{TG^2(\eta,\beta)L^2}{\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T)}}-\frac{TG^2(\eta,\beta)L^2}{\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T)}-\Delta_{trun} \tag{43}$$

$$\geq1-\left(G_2+\sqrt{\frac{TG^2(\eta,\beta)L^2}{\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}_T)}}\right)^2-\Delta_{trun}. \tag{44}$$

□

C.3. Proof of Theorem 4.5

Now we try to prove a general result for SGD. The proof consists of 2 steps. In the first step we extend the Proposition 4.1 to the SGD case. Although we may not be able to find a surrogate model that is parallel to the model update due to stochasticity and discretization, we show that the cosine similarity could be large. Then for high-dimensional space, we control $||P_2^{\perp}(\boldsymbol{w}_T - \boldsymbol{w}_0)||^2/||\boldsymbol{w}_T - \boldsymbol{w}_0||^2$ so that the model update on V_2 is dominating. As a result, the overall cosine similarity is large.

In the following, we let the stochastic gradient at step t be denoted as $g_t := \nabla \boldsymbol{w}_t + \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t$ is the gradient noise which has mean zero. For notational simplicity, we write $\ell(\boldsymbol{w}) = \mathbb{E}_D[\ell(\boldsymbol{w}, D)]$. All the assumptions in Theorem 4.5 are applied here.

Step 1:

We first control the distance between the weights achieved by GD and SGD (both start from $\boldsymbol{w}_0$) via the gradient noises.

Lemma C.3. *Consider SGD $\{\boldsymbol{w}_t\}$ and GD $\{\boldsymbol{w}'_t\}$ on some β-smooth loss ℓ. If both paths have step size η, T steps and start from $\boldsymbol{w}_0$, we have*

$$||\boldsymbol{w}_T - \boldsymbol{w}'_T|| \leq \eta \left|\left|\sum_{t=0}^{T-1} \boldsymbol{\epsilon}_t\right|\right| + \sum_{s=0}^{T-2} \eta^2\beta(1+\eta\beta)^{T-2-s} \left|\left|\sum_{t=0}^{s} \boldsymbol{\epsilon}_t\right|\right|. \tag{45}$$

Proof. By definition we have $\boldsymbol{w}_1 - \boldsymbol{w}'_1 = -\eta(\boldsymbol{g}_0 - \nabla \boldsymbol{w}_0) = -\eta\boldsymbol{\epsilon}_0$ and more generally,

$$\begin{aligned}
\boldsymbol{w}_{t+1} - \boldsymbol{w}'_{t+1} &= \boldsymbol{w}_t - \boldsymbol{w}'_t - \eta(\boldsymbol{g}_t - \nabla \boldsymbol{w}'_t) && (46)\\
&= \boldsymbol{w}_t - \boldsymbol{w}'_t - \eta(\boldsymbol{g}_t - \nabla \boldsymbol{w}_t) - \eta(\nabla \boldsymbol{w}_t - \nabla \boldsymbol{w}'_t) && (47)\\
&= \boldsymbol{w}_t - \boldsymbol{w}'_t - \eta\boldsymbol{\epsilon}_t - \eta(\nabla \boldsymbol{w}_t - \nabla \boldsymbol{w}'_t) && (48)
\end{aligned}$$

Thus, by summing the equations for $t = 0$ and $t = 1$ and using the smoothness condition, we have

$$\begin{aligned}
\boldsymbol{w}_2 - \boldsymbol{w}'_2 &= -\sum_{t=0}^{1} \eta\boldsymbol{\epsilon}_t - \eta(\nabla \boldsymbol{w}_1 - \nabla \boldsymbol{w}'_1) && (49)\\
\Rightarrow ||\boldsymbol{w}_2 - \boldsymbol{w}'_2|| &\leq \eta \left|\left|\sum_{t=0}^{1} \boldsymbol{\epsilon}_t\right|\right| + \eta\beta||\boldsymbol{w}_1 - \boldsymbol{w}'_1|| && (50)\\
||\boldsymbol{w}_2 - \boldsymbol{w}'_2|| &\leq \eta \left|\left|\sum_{t=0}^{1} \boldsymbol{\epsilon}_t\right|\right| + \eta^2\beta||\boldsymbol{\epsilon}_0|| && (51)
\end{aligned}$$

Therefore by induction, we finally conclude that

$$\begin{aligned}
||\boldsymbol{w}_T - \boldsymbol{w}'_T|| &\leq \eta \left|\left|\sum_{t=0}^{T-1} \boldsymbol{\epsilon}_t\right|\right| + \sum_{t=0}^{T-1} \eta\beta||\boldsymbol{w}_t - \boldsymbol{w}'_t|| && (52)\\
&\leq \eta \left|\left|\sum_{t=0}^{T-1} \boldsymbol{\epsilon}_t\right|\right| + \eta^2\beta \left|\left|\sum_{t=0}^{T-2} \boldsymbol{\epsilon}_t\right|\right| + \cdots + \eta^2\beta(1+\eta\beta)^{T-2}\,||\boldsymbol{\epsilon}_0|| && (53)\\
&\leq \eta \left|\left|\sum_{t=0}^{T-1} \boldsymbol{\epsilon}_t\right|\right| + \sum_{s=0}^{T-2} \eta^2\beta(1+\eta\beta)^{T-2-s} \left|\left|\sum_{t=0}^{s} \boldsymbol{\epsilon}_t\right|\right|. && (54)
\end{aligned}$$

□

Since the angle θ_1 between $\boldsymbol{w}_T - \boldsymbol{w}_0$ and $\boldsymbol{w}'_T - \boldsymbol{w}_0$ satisfies

$$\sin(\theta_1) \leq \frac{||\boldsymbol{w}_T - \boldsymbol{w}'_T||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||}, \tag{55}$$

the key is to control $||\sum_{t=0}^{s}\boldsymbol{\epsilon}_t||/||\boldsymbol{w}_T-\boldsymbol{w}_0||$. Thanks to the matrix Bernstein inequality below, we can infer with how much probability, $||\sum_{t=0}^{s}\boldsymbol{\epsilon}_t||/||\boldsymbol{w}_T-\boldsymbol{w}_0||$ is small. With a slight abuse of notation, we also write $||A||$ to denote the matrix norm of a $d_1\times d_2$ matrix A, i.e. for any $\boldsymbol{x}\in\mathbb{R}^{d_2}$,

$$||A||:=\max_{||\boldsymbol{x}||=1}\frac{||A\boldsymbol{x}||}{||\boldsymbol{x}||}. \tag{56}$$

Theorem C.4 (Theorem 1.6.2 of Tropp et al. (2015))**.** *Let $\boldsymbol{S}_1,\dots,\boldsymbol{S}_n$ be independent, centered random matrices with common dimension $d_1\times d_2$, and assume that $\mathbb{E}[\boldsymbol{S}_i]=0$ and $||\boldsymbol{S}_i||\le A$. Let $\boldsymbol{Z}=\sum_{k=1}^{n}\boldsymbol{S}_i$ and define*

$$v(\boldsymbol{Z})=\max\left\{\left|\left|\sum_{i=1}^{n}\mathbb{E}[\boldsymbol{S}_i\boldsymbol{S}_i^\top]\right|\right|,\left|\left|\sum_{i=1}^{n}\mathbb{E}[\boldsymbol{S}_i^\top\boldsymbol{S}_i]\right|\right|\right\}. \tag{57}$$

Then

$$P[||\boldsymbol{Z}||\ge t]\le(d_1+d_2)\exp\left(\frac{-t^2/2}{v(\boldsymbol{Z})+At/3}\right)\quad \textit{for all } t\ge 0. \tag{58}$$

The result is the following.

Lemma C.5. *Consider a stochastic gradient path $\{\boldsymbol{w}_t\}$ and gradient descent path $\{\boldsymbol{w}'_t\}$ with T steps on an L-Lipschitz loss ℓ on $\mathbb{R}^2$. For any $c>0$ (e.g. the choice of c in Theorem 4.5), we define*

$$r:=\min\left\{\frac{c^2(\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}'_T))^2}{4TL^2E_{max}^2},\frac{c(\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}'_T))}{4LE_{max}}\right\}, \tag{59}$$

Then, with probability at least $1-3\exp(-r)$, we have

$$\left|\left|\sum_{t=0}^{T-1}\boldsymbol{\epsilon}_t\right|\right|\le\frac{c}{L}(\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}'_T))\le c||\boldsymbol{w}'_T-\boldsymbol{w}_0||. \tag{60}$$

Proof. Since the gradient flow path is deterministic, we can always compare the accumulative noise with $\{\boldsymbol{w}'_t\}$. It's clear that $\boldsymbol{\epsilon}_t$ satisfies the conditions in Theorem C.4. Moreover, we have in the notation of their theorem that

$$v\left(\sum_{t=0}^{T-1}\boldsymbol{\epsilon}_t\right)\le T\max_{t=0,\dots,T-1}\{||\boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t^\top||,||\boldsymbol{\epsilon}_t^\top\boldsymbol{\epsilon}_t||\}\le TE_{max}^2. \tag{61}$$

Therefore, by the matrix Bernstein inequality we have that

$$\begin{aligned}P\left(\left|\left|\sum_{t=0}^{T-1}\boldsymbol{\epsilon}_t\right|\right|\ge c||\boldsymbol{w}'_T-\boldsymbol{w}_0||\right)&\le P\left(\left|\left|\sum_{t=0}^{T-1}\boldsymbol{\epsilon}_t\right|\right|\ge\frac{c}{L}(\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}'_T))\right) &(62)\\ &\le 3\exp\left(-\frac{c^2(\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}'_T))^2}{2L^2(TE_{max}^2+cE_{max}(\ell(\boldsymbol{w}_0)-\ell(\boldsymbol{w}'_T))/3L)}\right) &(63)\\ &\le 3\exp(-r). &(64)\end{aligned}$$

□

Remark C.6. In high dimensional case, the E_{max} here turns to be $\max||P_2\boldsymbol{\epsilon}||$. Thus, if $||P_2\boldsymbol{\epsilon}||\ll||\boldsymbol{\epsilon}||$, e.g. isotropic noise, the probability is close to 1. On the other hand, if $\boldsymbol{\epsilon}_t$ are mainly concentrated on some k-dimensional subspace of $\mathbb{R}^p$, i.e. $||P_k\boldsymbol{\epsilon}||\approx||\boldsymbol{\epsilon}||$ for some projection matrix P_k, we can still prove Equation (60) (up to some small constant) for $\boldsymbol{\epsilon}_t\in\mathbb{R}^p$ with probability at least $1-(k+1)\exp(-r)$.

Now, combining all the previous lemmas we have

Corollary C.7. *Under the conditions of Lemma C.3 and Lemma C.5, if $c < 1/\eta(1+\eta\beta)^{T-1}$, we have with probability at least $1 - 3T\exp(-r)$ that*

$$\theta_1 \leq \textit{arcsin}\left(c\eta(1+\eta\beta)^{T-1}\right), \tag{65}$$

moreover, we have for the SGD and GD that

$$\left(1 - c\eta(1+\eta\beta)^{T-1}\right)||\boldsymbol{w}_T' - \boldsymbol{w}_0|| \leq ||\boldsymbol{w}_T - \boldsymbol{w}_0|| \leq \left(1 + c\eta(1+\eta\beta)^{T-1}\right)||\boldsymbol{w}_T' - \boldsymbol{w}_0||. \tag{66}$$

Proof. By a uniform bound and Lemma C.5 we have that

$$P\left(\forall 0 \leq s \leq T-1, ||\sum_{t=0}^{s}\boldsymbol{\epsilon}_t|| \geq c||\boldsymbol{w}_T' - \boldsymbol{w}_0||\right) \leq \sum_{s=0}^{T-1} P\left(||\sum_{t=0}^{s}\boldsymbol{\epsilon}_t|| \geq c||\boldsymbol{w}_T' - \boldsymbol{w}_0||\right) \tag{67}$$

$$\leq 1 - 3T\exp(-r) \tag{68}$$

By Lemma C.3 we have that with probability at least $1 - 3T\exp(-r)$,

$$||\boldsymbol{w}_T - \boldsymbol{w}_T'|| \leq \eta\left|\left|\sum_{t=0}^{T-1}\boldsymbol{\epsilon}_t\right|\right| + \sum_{s=0}^{T-2}\eta^2\beta(1+\eta\beta)^{T-2-s}\left|\left|\sum_{t=0}^{s}\boldsymbol{\epsilon}_t\right|\right| \tag{69}$$

$$\leq \eta c||\boldsymbol{w}_T' - \boldsymbol{w}_0|| + \sum_{s=0}^{T-2}\eta^2\beta(1+\eta\beta)^{T-2-s}c||\boldsymbol{w}_T' - \boldsymbol{w}_0|| \tag{70}$$

$$\leq c||\boldsymbol{w}_T' - \boldsymbol{w}_0||\left(\eta + \eta^2\beta\frac{(1+\eta\beta)^{T-1} - 1}{\eta\beta}\right) \tag{71}$$

$$= c\eta(1+\eta\beta)^{T-1}||\boldsymbol{w}_T' - \boldsymbol{w}_0||. \tag{72}$$

The second claim follows from the triangle inequality. □

Intuitively, Corollary C.7 shows that if the gradient noise has a small magnitude, there is a large probability that θ_1 is small.

Now, by Proposition 4.1 and the proof of Theorem 4.3, we can find a surrogate model $\widehat{\boldsymbol{w}}'$ on the connected line between $\boldsymbol{w}_T'$ and $\boldsymbol{w}_0$ such that $\text{COSIM}(\nabla\widehat{\boldsymbol{w}}'\boldsymbol{w}_0 - \boldsymbol{w}_T') = 1 - \Delta_{trun} \approx 1$. As discussed in the main text, we omit Δ_{trun} in this theorem. It's clear that we can find a surrogate model $\widehat{\boldsymbol{w}}$ on the connected line between $\boldsymbol{w}_T$ and $\boldsymbol{w}_0$ such that $||\widehat{\boldsymbol{w}} - \widehat{\boldsymbol{w}}'|| \leq ||\boldsymbol{w}_T - \boldsymbol{w}_T'||$ and therefore by the smoothness of the true loss function we have,

$$||\nabla\widehat{\boldsymbol{w}} - \nabla\widehat{\boldsymbol{w}}'|| \leq \beta||\widehat{\boldsymbol{w}} - \widehat{\boldsymbol{w}}'|| \leq \beta||\boldsymbol{w}_T - \boldsymbol{w}_T'|| \leq c\eta\beta(1+\eta\beta)^{T-1}||\boldsymbol{w}_T' - \boldsymbol{w}_0|| \tag{73}$$

with high probability. Let θ_2 be the angle between $\nabla\widehat{\boldsymbol{w}}$ and $\nabla\widehat{\boldsymbol{w}}'$, we have that with probability at least $1 - 3T\exp(-r)$ that

$$\sin^2(\theta_2) \leq \frac{||\nabla\widehat{\boldsymbol{w}} - \nabla\widehat{\boldsymbol{w}}'||^2}{||\nabla\widehat{\boldsymbol{w}}'||^2} \leq \left(c\eta\beta(1+\eta\beta)^{T-1}\right)^2\frac{||\boldsymbol{w}_T' - \boldsymbol{w}_0||^2}{||\nabla\widehat{\boldsymbol{w}}'||^2} \tag{74}$$

$$\overset{(i)}{\leq} \left(c\eta\beta(1+\eta\beta)^{T-1}\right)^2\frac{\left(\sum_{t=0}^{T-1}||\boldsymbol{w}_{t+1}' - \boldsymbol{w}_t'||\right)^2}{||\nabla\widehat{\boldsymbol{w}}'||^2} \tag{75}$$

$$\overset{(ii)}{\leq} \left(c\eta\beta(1+\eta\beta)^{T-1}\right)^2\frac{\eta^2T\sum_{t=0}^{T-1}||\nabla\boldsymbol{w}_t'||^2}{||\nabla\widehat{\boldsymbol{w}}'||^2} \tag{76}$$

$$\leq \left(c\eta^2\beta(1+\eta\beta)^{T-1}\sqrt{T}\right)^2\frac{\sum_{t=0}^{T-1}||\nabla\boldsymbol{w}_t'||^2}{||\nabla\widehat{\boldsymbol{w}}'||^2} \tag{77}$$

$$\overset{(iii)}{\leq} \left(c\eta^2\beta(1+\eta\beta)^{T-1}\sqrt{T}\right)^2\frac{2(\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}_T'))}{\eta||\nabla\widehat{\boldsymbol{w}}'||^2} \tag{78}$$

$$\leq \left(\frac{c\eta^{3/2}\beta(1+\eta\beta)^{T-1}\sqrt{T}}{\sqrt{\gamma}}\right)^2\frac{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}_T')}{\ell(\boldsymbol{w}_T') - \ell(\boldsymbol{w}^*)} \tag{79}$$

where (i) is the triangular inequality, (ii) is the Cauchy-Schwartz inequality, (iii) follows from Theorem 4.8 of Bottou et al. (2018) and the last inequality follows from the definition of γ-strongly convex and Corollary C.1, where $\boldsymbol{w}^*$ is the global minimum.

Let θ be the angle between $\nabla\widehat{\boldsymbol{w}}$ and $\boldsymbol{w}_0 - \boldsymbol{w}_T$. Without loss of generality, we assume both $\theta_1, \theta_2 > 0$, as otherwise θ is bounded by θ_1 or θ_2 and the result is trivial. Then we have with probability at least $1 - 3T\exp(-r)$

$$\sin^2(\theta) = \sin^2(\theta_1 + \theta_2) \leq (|\sin(\theta_1)| + |\sin(\theta_2)|)^2 \tag{80}$$

$$\leq 2\sin^2(\theta_1) + 2\sin^2(\theta_2) \tag{81}$$

$$\leq \left(\frac{c\eta^{3/2}\beta(1+\eta\beta)^{T-1}\sqrt{2T}}{\sqrt{\gamma}}\right)^2 \frac{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)}{\ell(\boldsymbol{w}'_T) - \ell(\boldsymbol{w}^*)} + \left(\sqrt{2}c\eta(1+\eta\beta)^{T-1}\right)^2, \tag{82}$$

So that

$$\mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla\widehat{\boldsymbol{w}}) = 1 - \cos(\theta) \leq 1 - \cos^2(\theta)$$

$$\leq \sin^2(\theta) \leq \left(\frac{c\eta^{3/2}\beta(1+\eta\beta)^{T-1}\sqrt{2T}}{\sqrt{\gamma}}\right)^2 \frac{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)}{\ell(\boldsymbol{w}'_T) - \ell(\boldsymbol{w}^*)} + \left(\sqrt{2}c\eta(1+\eta\beta)^{T-1}\right)^2. \tag{83}$$

The above equation finishes the estimates in the 2D case.

Step 2: Finally, we prove Theorem 4.5.

Proof of Theorem 4.5. We shall apply the preceding results to the projected SGD on V_2. The only difference is that we have another factor $\frac{1}{1-G_2^2}$ in Equation (79) since the denominator in Equation (78) is replaced by $||P_2\nabla\widehat{\boldsymbol{w}}'||$. Thus we have with probability at least $1 - 3T\exp(-r)$ that $||\sum_{t=0}^{T-1} P_2\boldsymbol{\epsilon}_t|| \leq c||P_2(\boldsymbol{w}'_T - \boldsymbol{w}_0)||$ (or even higher probability since it could happen that $||P_2\boldsymbol{\epsilon}_t||$ is much less than $||\boldsymbol{\epsilon}_t||$, see Remark C.6), so that on this event,

$$\frac{||P_2^\perp(\boldsymbol{w}_T - \boldsymbol{w}_0)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} \leq \frac{||P_2^\perp(\boldsymbol{w}'_T - \boldsymbol{w}_0)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} + \frac{||P_2^\perp(\boldsymbol{w}_T - \boldsymbol{w}'_T)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} \tag{84}$$

$$\leq \frac{||P_2^\perp(\boldsymbol{w}'_T - \boldsymbol{w}_0)||}{||P_2(\boldsymbol{w}_T - \boldsymbol{w}_0)||} + \frac{||P_2^\perp(\boldsymbol{w}_T - \boldsymbol{w}'_T)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} \tag{85}$$

$$\leq C_\eta \frac{||P_2^\perp(\boldsymbol{w}'_T - \boldsymbol{w}_0)||}{||P_2(\boldsymbol{w}'_T - \boldsymbol{w}_0)||} + \frac{||\boldsymbol{w}_T - \boldsymbol{w}'_T||}{||P_2(\boldsymbol{w}_T - \boldsymbol{w}_0)||} \tag{86}$$

$$\leq C_\eta \frac{C_{GD}}{1 - C_{GD}} + C_\eta \frac{||\boldsymbol{w}_T - \boldsymbol{w}'_T||}{||P_2(\boldsymbol{w}'_T - \boldsymbol{w}_0)||} \tag{87}$$

$$\leq C_\eta \frac{C_{GD}}{1 - C_{GD}} + \frac{C_\eta}{1 - C_{GD}} \frac{||\boldsymbol{w}_T - \boldsymbol{w}'_T||}{||\boldsymbol{w}'_T - \boldsymbol{w}_0||} \tag{88}$$

$$\leq \frac{C_\eta}{1 - C_{GD}} \left(C_{GD} + \frac{L||\boldsymbol{w}_T - \boldsymbol{w}'_T||}{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)}\right) \tag{89}$$

$$\leq \frac{C_\eta}{1 - C_{GD}} \left(C_{GD} + \frac{LTE_{max}\eta(1+\eta\beta)^{T-1}}{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)}\right). \tag{90}$$

Here, $C_\eta = \frac{1}{1-\sqrt{\eta}}$ is the constant follows from Equation (66) and the choice of c in the theorem, which is close to 1. The 4th and 5th inequalities follow from Equation (40) and Equation (41). The last inequality follows from Lemma C.3 and union bound.

In total, let $\widehat{w}$ be the surrogate model such that the angle θ between $P_2\nabla\widehat{\boldsymbol{w}}$ and $P_2(\boldsymbol{w}_0 - \boldsymbol{w}_T)$ satisfies Equation (83), we have with probability at least $1 - 3T\exp(-r)$ that,

$$\mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \widehat{\boldsymbol{w}}) = 1 - \frac{\langle\nabla\widehat{\boldsymbol{w}}, \boldsymbol{w}_T - \boldsymbol{w}_0\rangle}{||\nabla\widehat{\boldsymbol{w}}||||\boldsymbol{w}_T - \boldsymbol{w}_0||} \tag{91}$$

$$\leq 1 - \left(\frac{|\langle P_2\nabla\widehat{\boldsymbol{w}}, P_2(\boldsymbol{w}_0 - \boldsymbol{w}_T)\rangle|}{||\nabla\widehat{\boldsymbol{w}}||||\boldsymbol{w}_T - \boldsymbol{w}_0||} - \frac{|\langle P_2^\perp\nabla\widehat{\boldsymbol{w}}, P_2^\perp(\boldsymbol{w}_T - \boldsymbol{w}_0)\rangle|}{||\nabla\widehat{\boldsymbol{w}}||||\boldsymbol{w}_T - \boldsymbol{w}_0||}\right) \tag{92}$$

$$\leq 1 - \left(\cos(\theta) \frac{||P_2 \nabla \widehat{\boldsymbol{w}}||||P_2(\boldsymbol{w}_T - \boldsymbol{w}_0)||}{||\nabla \widehat{\boldsymbol{w}}||||\boldsymbol{w}_T - \boldsymbol{w}_0||} - \frac{||P_2^{\perp} \nabla \widehat{\boldsymbol{w}}||}{||\nabla \widehat{\boldsymbol{w}}||} \frac{||P_2^{\perp}(\boldsymbol{w}_T - \boldsymbol{w}_0)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} \right) \tag{93}$$

$$\leq 1 - \left[(1 - G_2^2) \cos(\theta) \frac{||P_2(\boldsymbol{w}_T - \boldsymbol{w}_0)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} - G_2 \frac{||P_2^{\perp}(\boldsymbol{w}_T - \boldsymbol{w}_0)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} \right] \tag{94}$$

$$\leq 1 - \left[(1 - G_2^2) \cos(\theta) \left(1 - \frac{||P_2^{\perp}(\boldsymbol{w}_T - \boldsymbol{w}_0)||}{||\boldsymbol{w}_t - \boldsymbol{w}_0||} \right) - G_2 \frac{||P_2^{\perp}(\boldsymbol{w}_T - \boldsymbol{w}_0)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} \right] \tag{95}$$

$$\leq 1 - (1 - G_2^2) \cos^2(\theta) + [(1 - G_2^2) \cos(\theta) + G_2] \frac{||P_2^{\perp}(\boldsymbol{w}_T - \boldsymbol{w}_0)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} \tag{96}$$

$$\leq 1 - (1 - G_2^2) \cos^2(\theta) + (1 + G_2) \frac{||P_2^{\perp}(\boldsymbol{w}_T - \boldsymbol{w}_0)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} \tag{97}$$

$$\leq G_2^2 + \sin^2(\theta) + (1 + G_2) \frac{||P_2^{\perp}(\boldsymbol{w}_t - \boldsymbol{w}_0)||}{||\boldsymbol{w}_T - \boldsymbol{w}_0||} \tag{98}$$

$$\leq G_2^2 + \frac{1}{1 - G_2^2} \left(\frac{c\eta^{3/2}\beta(1 + \eta\beta)^{T-1}\sqrt{2T}}{\sqrt{\gamma}} \right)^2 \frac{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}_T')}{\ell(\boldsymbol{w}_T') - \ell(\boldsymbol{w}^*)} + \left(\sqrt{2} c\eta(1 + \eta\beta)^{T-1} \right)^2 \tag{99}$$

$$+ \frac{C_\eta(1 + G_2)}{1 - C_{GD}} \left(C_{GD} + \frac{LTE_{max}\eta(1 + \eta\beta)^{T-1}}{\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}_T')} \right). \tag{100}$$

By letting $c = 1/\sqrt{\eta}(1 + \eta\beta)^{T-1}$ we prove the desired result. □

D. Experiment Setup

Dataset. FEMNIST is proposed in the FL bechmark *LEAF* (Caldas et al., 2018). This dataset is adapted from EMNIST (Cohen et al., 2017). which consists of 28×28 grayscale images with 62 classes of hand-written digits and characters. CIFAR100 (Krizhevsky, 2012) consists of 32×32 RGB images with 100 classes of natural images.

Federated Learning. In this work, we consider a typical FL framework: Federated Averaging (`FedAvg`) (McMahan et al., 2017) as the attack scenario. The algorithm of `FedAvg` is given in Algorithm 2. To consider the different situations of local training in FL, we need to change the number of training epochs E, batch size B, local data size N, and learning rate η. To improve the clarity of comparison, we follow the previous work (Dimitrov et al., 2022) and fix the learning rate $\eta = 0.004$ for the two CNNs. This learning rate is also suggested by the benchmark *LEAF* for a good training result in FL using the same network. Then in Table 2, we conduct experiments with different batch sizes B, epochs E, and local data size N to investigate the influence of local training for different attack approaches.

In Figure 3, Figure 4, Figure 9, Figure 10, and Table 6, we also consider clients joining the FL group at different communication rounds, such that the network sent to the clients is at different training stages. To obtain the trained FL network, we generate the FL clients using the data splitting tool provided by *LEAF*. The dataset is FEMNIST. Local data is non-identically distributed across clients and the FL group has approximately 40.000 data points in total. During FL, every client trains the network locally with batch size $B = 50$, epochs $E = 10$, and learning rate $\eta = 0.004$. To simulate the straggler issue in FL, every client has the possibility of 0.1 sending the trained weights back to the server. We save the trained network at the end of communication rounds 5, 10, and 20 to conduct attack experiments. These networks can achieve test accuracy: $\sim 19\%$, $\sim 28\%$, and $\sim 40\%$.

Algorithm 2 Federated Averaging (McMahan et al., 2017)

Notations: The J clients are indexed by j; B is the batch size, E is the number of local epochs, and η is the learning rate.

Sever executes:
Initialize $\boldsymbol{w}^0$.
for each round $r = 0 \dots R$ **do**
  for each client $j = 1 \dots J$ **do**
    $\boldsymbol{w}_j^{r+1} \leftarrow \text{ClientUpdate}(\boldsymbol{w}^r)$
  end for
  $\boldsymbol{w}^{r+1} = n_j \boldsymbol{w}_j^{r+1} / \sum_{j=1}^{J} n_j$
end for
ClientUpdate($\boldsymbol{w}_0$)**:**
Set $t = 0$.
for each epoch $e = 0 \dots E$ **do**
  $\mathcal{B} \leftarrow$ Split local data D into batches of size B.
  for batch $b \in \mathcal{B}$ **do**
    $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla \ell(\boldsymbol{w}_t; b)$
  end for
end for

Hyperparameters for attack approaches. `DLFA` (Dimitrov et al., 2022) has several variants, we use the one that is reported to always give the best reconstruction results, i.e. the full input-reconstruction method including the order-invariant prior $\mathcal{L}_{prior}$. We adopt the officially released code and recommended hypeparameters that is available at `https://github.com/eth-sri/fedavg_leakage`.

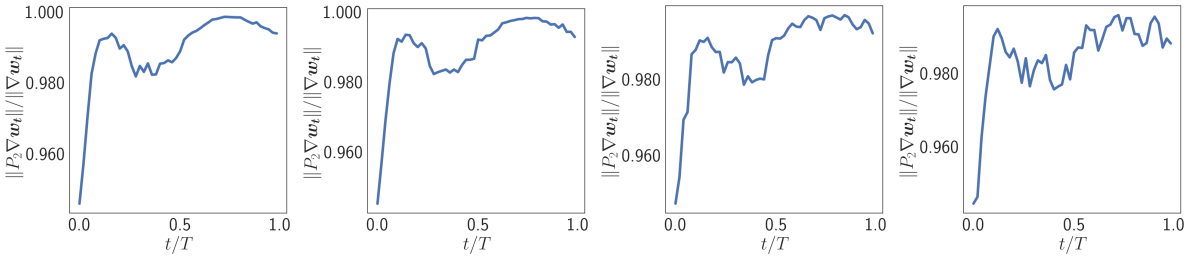
For our `SME` and the base method `IG`, we use the same hyperparameters and adapt them from the work of `IG` (Geiping et al., 2020). We note that due to the similar mechanisms, these two methods achieve the best performance with the same hyperparameters. In particular, we set learning rate $\eta_{\tilde{D}} = 1$ for the optimization of reconstructed data. and learning rate $\eta_\alpha = 0.001$ for the optimization of α. We set $\lambda = 0.01$ for the total variation prior, and use the Adam optimizer (Kingma & Ba, 2014). Additionally, we bound the reconstructed pixel values within the valid range of image pixel values, this has also

been implemented in previous works including DLFA. We let the reconstruction run for 1000 iterations.

E. Additional Empirical Evidence

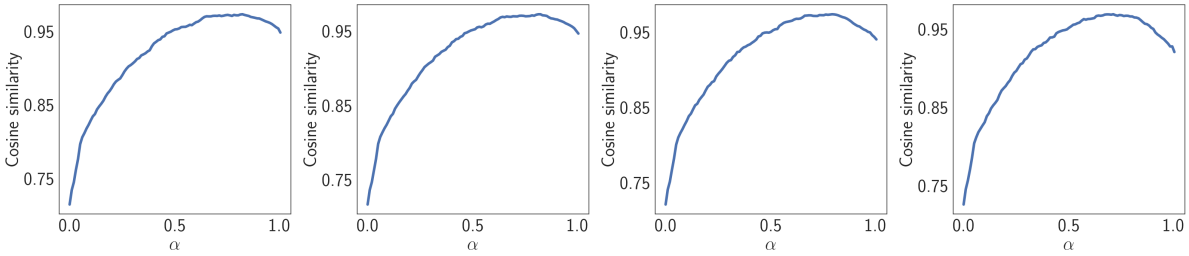
According to Theorem 4.3 and Theorem 4.5, in order to find a surrogate model $\widehat{\boldsymbol{w}}$ on the connected line achieving small $\mathcal{L}_{sim}(\boldsymbol{w}_0 - \boldsymbol{w}_T, \nabla\widehat{\boldsymbol{w}})$, there are two necessary conditions. First, G_2 needs to be small such that the low-rank property holds. Second, the magnitude of gradient noise $\{\boldsymbol{\epsilon}_t\}_{t=0}^{T-1}$ needs to be moderate compared with the loss reduction $\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)$. In Figure 3 and Figure 4 we have shown for many situations we can find such a proper surrogate model. In this section, we further investigate this problem in more challenging situations.

First, we investigate the influence of gradient noise $\boldsymbol{\epsilon}_t$. Since gradient noise is induced by stochastic mini-batch gradient, in Figure 7 and Figure 8 we consider the same local steps T but different batch size B. As we can see, when $B = 5$ the fluctuation in the gradient norm ratio is more obvious. However, overall the plots of the gradient norm ratio and the cosine similarity are similar. This indicates that the stochastic gradient flow oscillates around the true gradient flow. And the impact of gradient noise $\boldsymbol{\epsilon}_t$ to SME is marginal. Statistical evaluations of multiple samplings are presented in Table 4.



(a) E=50, B=50, T=50 (b) E=25, B=25, T=50 (c) E=10, B=10, T=50 (d) E=5, B=5, T=50

Figure 7. Gradient norm ratio $\|P_2\nabla \boldsymbol{w}_t\|/\|\nabla \boldsymbol{w}_t\|$ along the local steps $t = 0 \dots T-1$. We sample a client with $N = 50$ data points and change the batch size B and epochs E, such that there are always $T = 50$ steps.



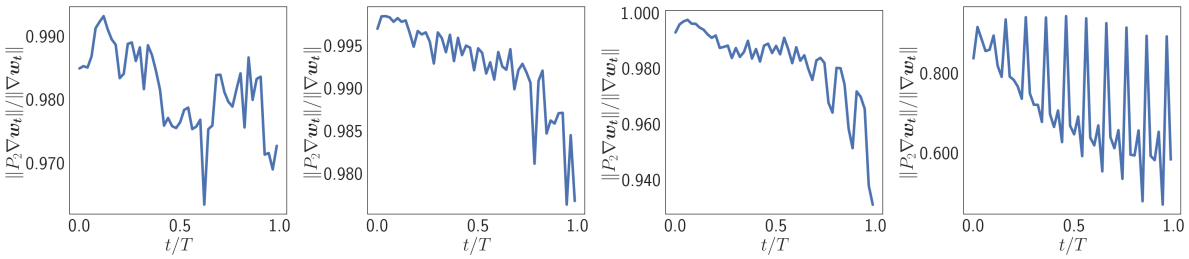
(a) E=50, B=50, T=50 (b) E=25, B=25, T=50 (c) E=10, B=10, T=50 (d) E=5, B=5, T=50

Figure 8. Cosine similarity between $\boldsymbol{w}_0 - \boldsymbol{w}_T$ and $\nabla\widehat{\boldsymbol{w}}$ vs. α. The settings are aligned with Figure 7, please refer to the caption above.

	(E=50, B=50, T=50)	(E=25, B=25, T=50)	(E=10, B=10, T=50)	(E=5, B=5, T=50)
$\mathbb{E}[\min_t \frac{\|P_2\nabla \boldsymbol{w}_t\|}{\|\boldsymbol{w}_t\|}]$	.977±.009	.978±.007	.977±.008	.973±.009
$\mathbb{E}[\text{COSIM}\|\alpha = 0]$	.867±.041	.869±.049	.872±.037	.876±.041
$\mathbb{E}[\max_{\alpha\in[0,1]} \text{COSIM}]$	.980±.007	.978±.026	.979±.007	.977±.008

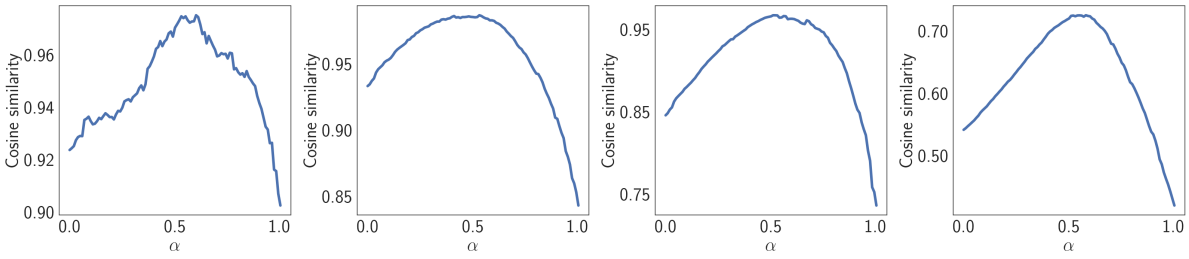
Table 4. Statistical evaluations corresponding to Figures 7 and 8. We repeat the experiment 100 times to compute the mean and standard deviation. COSIM is a shorthand for the cosine similarity between $\boldsymbol{w}_0 - \boldsymbol{w}_T$ and $\nabla\widehat{\boldsymbol{w}}$.

Next, we investigate the influence of the network's training state. During FL, the low-rank property of the gradients may change. Additionally, the loss reduction $\ell(\boldsymbol{w}_0) - \ell(\boldsymbol{w}'_T)$ is expected to be smaller as $\ell(\boldsymbol{w}_0)$ decreases. To study this, we train a network with FL as described in Appendix D, and conduct attacks in a challenging setting with 50 steps of SGD. Figure 9 shows that as the communication round increases, the gradient norm ratio decreases, i.e. G_2 becomes larger. Figure 10 shows that in the middle of the connected line, the surrogate gradient still has a higher cosine similarity. However, the highest cosine similarity that can be reached becomes smaller. Statistical evaluations of multiple samplings are presented in Table 5. We also provide the reconstruction performance on trained networks in Appendix F.



(a) E=10, B=10, R=1 (b) E=10, B=10, R=5 (c) E=10, B=10, R=10 (d) E=10, B=10, R=20

Figure 9. Gradient norm ratio $\|P_2\nabla \boldsymbol{w}_t\|/\|\nabla \boldsymbol{w}_t\|$ along the local steps $t = 0 \dots T-1$. We sample a client with $N = 50$ data points, and set $E = 10$, $B = 10$, such that for each epoch there are five SGD steps and in total $T = 50$ local steps. We change the number of communication rounds R, such that the network $\boldsymbol{w}_0$ received by the clients is at different training stage.



(a) E=10, B=10, R=1 (b) E=10, B=10, R=5 (c) E=10, B=10, R=10 (d) E=10, B=10, R=20

Figure 10. Cosine similarity between $\boldsymbol{w}_0 - \boldsymbol{w}_T$ and $\nabla\widehat{\boldsymbol{w}}$ vs. α. The settings are aligned with Figure 9, please refer to the caption above.

	(E=10, B=10, R=1)	(E=10, B=10, R=5)	(E=10, B=10, R=10)	(E=10, B=10, R=20)
$\mathbb{E}[\min_t \frac{\|P_2\nabla \boldsymbol{w}_t\|}{\|\boldsymbol{w}_t\|}]$	.977±.008	.937±.036	.813±.087	.406±.176
$\mathbb{E}[\text{COSIM}\|\alpha = 0]$	.872±.004	.871±.044	.764±.061	.542±.074
$\mathbb{E}[\max_{\alpha\in[0,1]} \text{COSIM}]$	.979±.007	.962±.019	.902±.052	.602±.010

Table 5. Statistical evaluations corresponding to Figures 9 and 10. We repeat the experiment 100 times to compute the mean and standard deviation. COSIM is a shorthand for the cosine similarity between $\boldsymbol{w}_0 - \boldsymbol{w}_T$ and $\nabla\widehat{\boldsymbol{w}}$.

F. Reconstruction Performance on Trained Networks

During FL, the low-rank property of gradients may be changed, also the loss reduction of local training could be narrowed. Both of these could have a negative impact on the performance of SME according to our Theorem 4.3 and Theorem 4.5. The empirical evidence provided in Appendix E reflects this phenomenon. As the number of communication rounds grows, the gradients are less concentrated in a 2D subspace (see Figure 9), and the highest cosine similarity achieved by the surrogate model also becomes smaller (see Figure 10).

In this section, we present the results of reconstruction performance w.r.t. the influence of the training state of $\boldsymbol{w}_0$. We give the details of FL in Appendix D. The attack results are shown in Table 6. As we can see, for larger communication round R, the difference of PSNR between SME and IG becomes smaller, but we note that SME still consistently outperforms IG. DLFA performs better than SME when R is large. However, we emphasize that DLFA generally has a significantly higher demand for computational resources as discussed in Section 5.3, it may not be able to execute DLFA under some circumstances.

We also observe that the reconstruction performance of all attack approaches becomes worse as the number of communication rounds R grows. Seemingly, clients joining FL after multiple rounds of communication would have better privacy. However, this also leads to less contribution to the collaboration, which is an issue that needs to be considered in FL. We leave the study of the trade-off between privacy and collaboration contribution for future work. Additionally, the results indicate that using a pre-trained network in FL may be preferable not only in terms of network performance but also privacy. We also provide a visualization of the reconstruction in Figure 13.

				DLFA		IG		SME (ours)		
R	E	N	T	$\mathcal{L}_{sim}\downarrow$	PSNR ↑	$\mathcal{L}_{sim}\downarrow$	PSNR ↑	$\mathcal{L}_{sim}\downarrow$	PSNR ↑	**Δ** PSNR
1	10	10	10	$.021 \pm .001$	24.9 ± 0.2	$.044 \pm .002$	27.8 ± 0.3	$.019 \pm .001$	$\mathbf{30.3 \pm 0.3}$	$+2.5$
	20	10	20	$.019 \pm .001$	25.4 ± 0.2	$.090 \pm .003$	24.2 ± 0.3	$.019 \pm .001$	$\mathbf{28.6 \pm 0.3}$	$+3.2$
	10	50	50	$.015 \pm .001$	21.7 ± 0.1	$.091 \pm .003$	19.1 ± 0.2	$.027 \pm .001$	$\mathbf{22.2 \pm 0.2}$	$+0.5$
	20	50	100	$.014 \pm .001$	$\mathbf{21.7 \pm 0.2}$	$.176 \pm .011$	17.0 ± 0.2	$.037 \pm .001$	21.5 ± 0.2	-0.2
5	10	10	10	$.003 \pm .000$	22.1 ± 0.2	$.026 \pm .001$	22.7 ± 0.3	$.009 \pm .001$	$\mathbf{24.8 \pm 0.3}$	$+2.7$
	20	10	20	$.003 \pm .000$	23.0 ± 0.1	$.064 \pm .003$	20.9 ± 0.2	$.018 \pm .001$	$\mathbf{24.7 \pm 0.3}$	$+1.7$
	10	50	50	$.005 \pm .000$	20.8 ± 0.2	$.062 \pm .004$	18.7 ± 0.2	$.020 \pm .002$	$\mathbf{21.4 \pm 0.3}$	$+0.6$
	20	50	100	$.005 \pm .000$	$\mathbf{20.9 \pm 0.2}$	$.131 \pm .007$	17.2 ± 0.2	$.042 \pm .003$	20.3 ± 0.3	-0.6
10	10	10	10	$.003 \pm .000$	22.5 ± 0.1	$.034 \pm .001$	21.4 ± 0.3	$.011 \pm .001$	$\mathbf{23.8 \pm 0.3}$	$+1.3$
	20	10	20	$.004 \pm .000$	23.1 ± 0.2	$.079 \pm .003$	19.5 ± 0.2	$.019 \pm .001$	$\mathbf{23.2 \pm 0.4}$	$+0.1$
	10	50	50	$.006 \pm .000$	$\mathbf{20.7 \pm 0.2}$	$.107 \pm .016$	17.7 ± 0.2	$.027 \pm .001$	19.8 ± 0.3	-0.9
	20	50	100	$.008 \pm .000$	$\mathbf{20.5 \pm 0.2}$	$.298 \pm .035$	16.7 ± 0.2	$.060 \pm .004$	18.4 ± 0.3	-2.1
20	10	10	10	$.008 \pm .001$	$\mathbf{21.3 \pm 0.2}$	$.053 \pm .002$	19.0 ± 0.2	$.019 \pm .001$	20.7 ± 0.3	-0.6
	20	10	20	$.013 \pm .001$	$\mathbf{20.8 \pm 0.2}$	$.109 \pm .013$	17.9 ± 0.2	$.044 \pm .003$	19.6 ± 0.3	-1.2
	10	50	50	$.020 \pm .001$	$\mathbf{19.4 \pm 0.2}$	$.208 \pm .003$	16.2 ± 0.2	$.152 \pm .030$	17.3 ± 0.2	-2.1
	20	50	100	$.020 \pm .001$	$\mathbf{19.0 \pm 0.2}$	$.374 \pm .043$	15.4 ± 0.2	$.306 \pm .043$	16.2 ± 0.2	-2.8

Table 6. Average reconstructed image quality measured by PSNR and similarity loss of the reconstruction objective $\mathcal{L}_{sim}$ on FEMNIST. We set the batch size $B = 10$ and consider different communication round R, local data sizes N, and epochs E. Local steps $T = E \times (N/B)$. The best reconstruction results are bold. The difference of PSNRs between SME and the best baseline is given in the last column. We restate part of the results in Table 2 for $R = 1$. Also refer to Figure 13 for visualization.

G. Reconstruction Performance on other Network Architectures

We provide the reconstruction performance results on other network architectures. In particular, we consider: (1) A MLP with two hidden layers of shape $(1000, 1000)$ equipped with ReLU activation function. (2) A LeNet (Lecun et al., 1998) equipped with Tanh activation function. (3) ResNet (He et al., 2016), a convolutional neural network with skip connections. In particular, we adopt a relatively small ResNet8 with three building blocks in favor of `DLFA` and follow Geiping et al. (2020) to freeze the running statistics of the batch normalization layers during training. Updating the running statistics might disable reconstruction as reported by Huang et al. (2021). (4) ViT (Dosovitskiy et al., 2021), a non-convolutional network with multi-head attention building blocks and layer normalization. We follow Lu et al. (2021) and adopt the architecture of type A. We set the learning rate in FL to 0.4 for LeNet and 0.004 for the other architectures. All attack approaches adopt the hyperparameters described in Appendix D. The results are given in Table 7.

As we can see, in general `SME` achieves the best performance in terms of reconstruction. `DLFA`, on the other hand, fails in the attacks of LeNet and ResNet8. We also notice that `DLFA` works when the learning rate of LeNet is lower. However, when we increase the learning rate to 0.4, the reconstruction of `DLFA` converges in the first few iterations and then diverges dramatically (we record the best PSNR during reconstruction). This may indicate that `DLFA` is prone to failure or sensitive to configuration, while the gradient inversion methods are more robust on different architectures.

				DLFA		IG		SME (ours)		
Network	E	N	T	$\mathcal{L}_{sim}\downarrow$	PSNR ↑	$\mathcal{L}_{sim}\downarrow$	PSNR ↑	$\mathcal{L}_{sim}\downarrow$	PSNR ↑	**Δ** PSNR
MLP	10	10	10	$.018 \pm .000$	$\mathbf{36.0 \pm 0.3}$	$.043 \pm .006$	31.7 ± 0.3	$.009 \pm .000$	35.9 ± 0.4	-0.1
	20	10	20	$.025 \pm .000$	31.3 ± 0.2	$.041 \pm .002$	29.5 ± 0.3	$.013 \pm .000$	$\mathbf{34.1 \pm 0.4}$	$+2.8$
	50	10	50	$.051 \pm .001$	28.0 ± 0.2	$.097 \pm .002$	25.1 ± 0.3	$.023 \pm .001$	$\mathbf{30.1 \pm 0.3}$	$+2.1$
	10	50	50	$.030 \pm .001$	27.0 ± 0.2	$.056 \pm .002$	25.1 ± 0.3	$.009 \pm .000$	$\mathbf{31.4 \pm 0.3}$	$+4.4$
	20	50	100	$.039 \pm .001$	24.6 ± 0.2	$.090 \pm .002$	23.3 ± 0.3	$.017 \pm .001$	$\mathbf{28.5 \pm 0.3}$	$+3.9$
	50	50	250	$.058 \pm .002$	22.0 ± 0.2	$.173 \pm .003$	20.6 ± 0.2	$.054 \pm .001$	$\mathbf{22.8 \pm 0.3}$	$+0.8$
LeNet	10	10	10	$.315 \pm .022$	16.3 ± 0.1	$.000 \pm .000$	29.1 ± 0.3	$.000 \pm .000$	$\mathbf{29.5 \pm 0.3}$	$+0.4$
	20	10	20	$.555 \pm .016$	16.0 ± 0.1	$.002 \pm .000$	27.8 ± 0.3	$.000 \pm .000$	$\mathbf{29.4 \pm 0.3}$	$+1.6$
	50	10	50	$.690 \pm .040$	16.1 ± 0.1	$.015 \pm .002$	23.0 ± 0.3	$.000 \pm .000$	$\mathbf{29.0 \pm 0.3}$	$+6.0$
	10	50	50	$.663 \pm .012$	15.9 ± 0.1	$.003 \pm .000$	23.0 ± 0.4	$.000 \pm .000$	$\mathbf{24.6 \pm 0.3}$	$+1.6$
	20	50	100	$.553 \pm .023$	16.1 ± 0.1	$.017 \pm .002$	19.8 ± 0.4	$.000 \pm .000$	$\mathbf{24.2 \pm 0.3}$	$+4.4$
	50	50	250	$.690 \pm .040$	16.4 ± 0.2	$.159 \pm .010$	15.0 ± 0.2	$.019 \pm .002$	$\mathbf{19.4 \pm 0.5}$	$+3.0$
ResNet8	10	10	10	$.148 \pm .009$	16.5 ± 0.1	$.004 \pm .000$	32.7 ± 0.5	$.001 \pm .000$	$\mathbf{36.8 \pm 0.5}$	$+4.1$
	20	10	20	$.163 \pm .011$	16.3 ± 0.1	$.011 \pm .000$	26.7 ± 0.4	$.001 \pm .000$	$\mathbf{36.6 \pm 0.5}$	$+9.9$
	50	10	50	$.125 \pm .010$	16.2 ± 0.2	$.059 \pm .003$	18.4 ± 0.3	$.001 \pm .000$	$\mathbf{33.2 \pm 0.5}$	$+14.8$
	10	50	50	$.114 \pm .001$	15.9 ± 0.1	$.015 \pm .001$	21.2 ± 0.5	$.002 \pm .000$	$\mathbf{28.2 \pm 0.5}$	$+7.0$
	20	50	100	$.116 \pm .005$	16.1 ± 0.2	$.061 \pm .004$	17.1 ± 0.4	$.002 \pm .000$	$\mathbf{27.3 \pm 0.6}$	$+10.2$
	50	50	250	$.206 \pm .007$	16.2 ± 0.2	$.163 \pm .006$	14.0 ± 0.1	$.028 \pm .003$	$\mathbf{20.8 \pm 0.6}$	$+4.6$
ViT	10	10	10	$.008 \pm .001$	25.6 ± 0.5	$.037 \pm .001$	25.1 ± 0.2	$.012 \pm .000$	$\mathbf{30.3 \pm 0.2}$	$+4.7$
	20	10	20	$.007 \pm .000$	21.1 ± 0.2	$.079 \pm .001$	21.1 ± 0.1	$.028 \pm .000$	$\mathbf{26.4 \pm 0.2}$	$+5.3$
	50	10	50	$.013 \pm .004$	18.9 ± 0.1	$.138 \pm .001$	17.7 ± 0.1	$.053 \pm .001$	$\mathbf{21.3 \pm 0.1}$	$+1.4$
	10	50	50	$.017 \pm .009$	20.6 ± 0.2	$.067 \pm .002$	19.1 ± 0.2	$.022 \pm .001$	$\mathbf{23.3 \pm 0.3}$	$+2.7$
	20	50	100	$.015 \pm .004$	18.3 ± 0.1	$.114 \pm .002$	16.8 ± 0.1	$.044 \pm .001$	$\mathbf{20.0 \pm 0.1}$	$+1.7$
	50	50	250	N/A	N/A	$.157 \pm .001$	14.9 ± 0.1	$.070 \pm .002$	$\mathbf{17.2 \pm 0.1}$	N/A

Table 7. Average reconstructed image quality measured by PSNR and similarity loss of the reconstruction objective $\mathcal{L}_{sim}$ on FEMNIST. We set the batch size $B = 10$ and change the local data sizes N and epochs E. Local steps $T = E \times (N/B)$. The experiments are conducted at the first communication round. The best reconstruction results are bold. The difference of PSNRs between SME and the best baseline is given in the last column. Also refer to Figure 14 for visualization. Results of DLFA in the last row is not available, as *JAX* raises unknown compilation error.

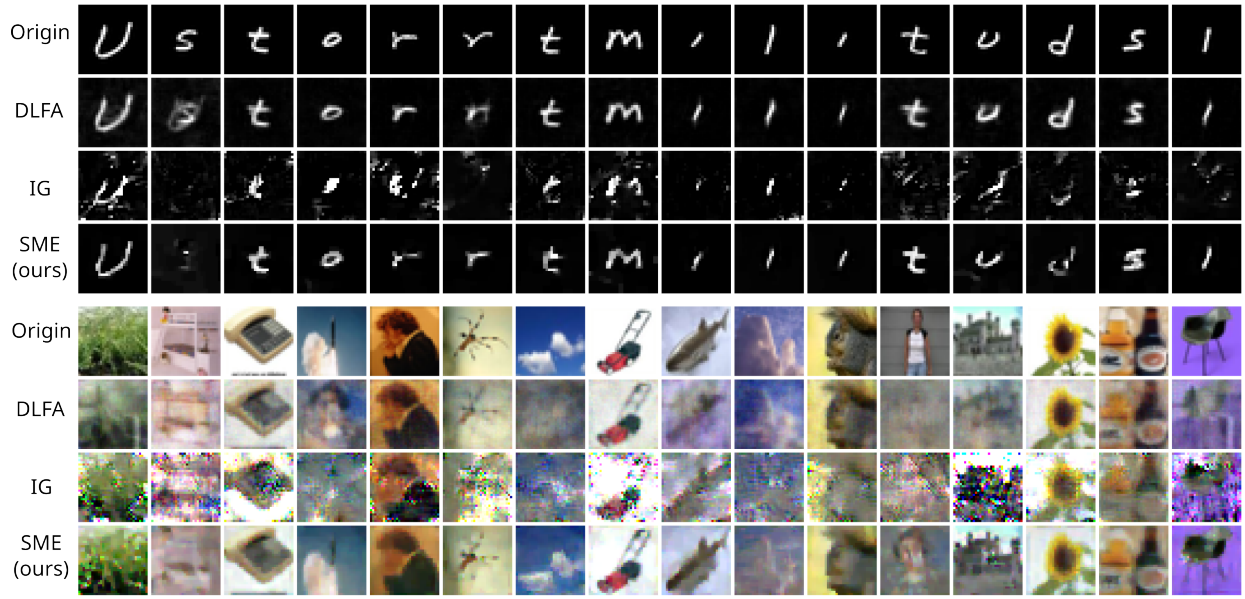
H. Reconstruction Visualization

(a) E=20, B=10, T=20

(b) E=50, B=10, T=50

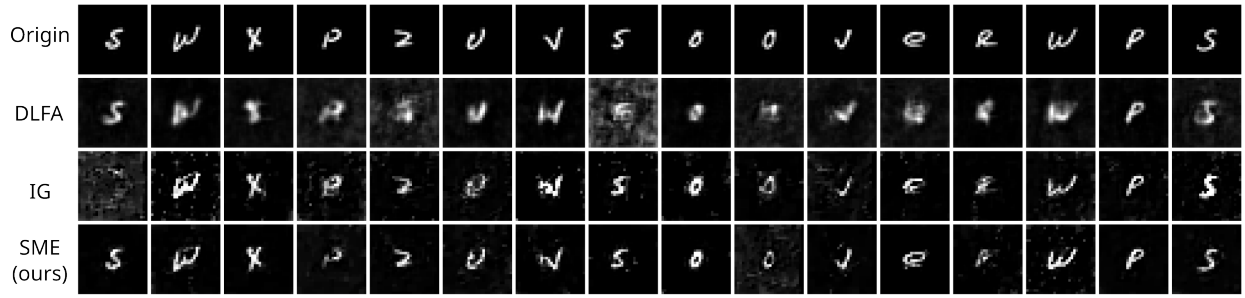
Figure 11. Visualization of the reconstructed images. The results belong to one reconstruction of the settings ($E \in \{20, 50\}$, $N = 10$) in Table 2. The reconstructed images are paired with the original images through *linear sum assignment*.

(a) E=10, B=10, T=50

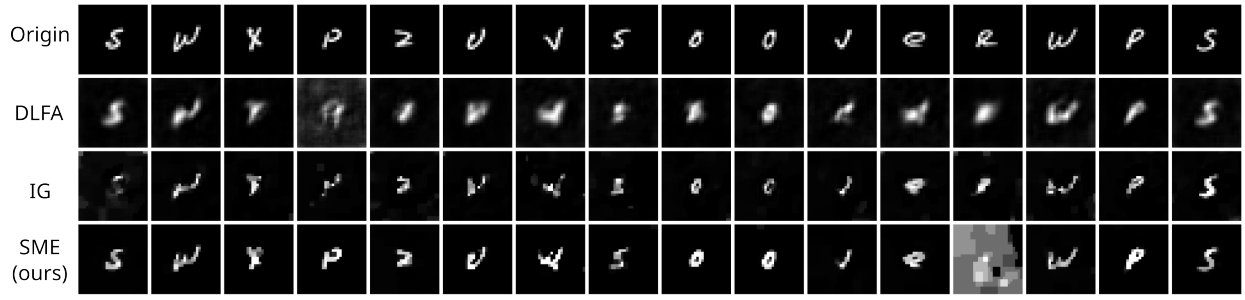


(b) E=20, B=10, T=100

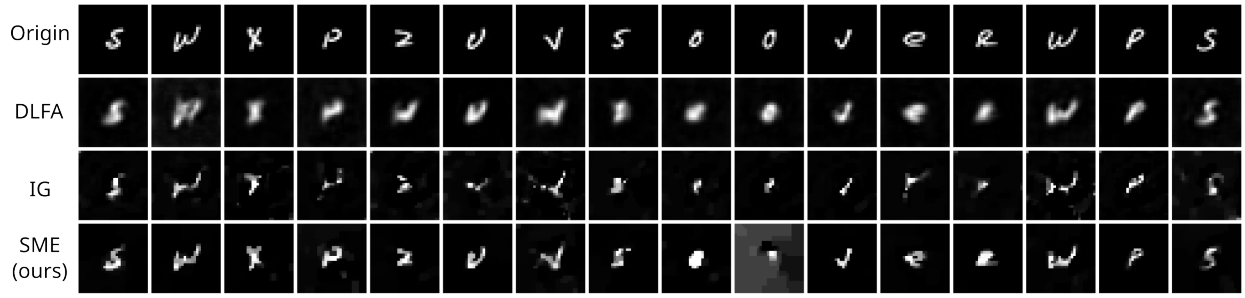
Figure 12. Visualization of the reconstructed images. The results are drawn from the settings ($E \in \{10, 20\}$, $N = 50$) in Table 2. The reconstructed images are paired with the original images through *linear sum assignment*. We randomly sample 16 out of 50 images of one reconstruction.



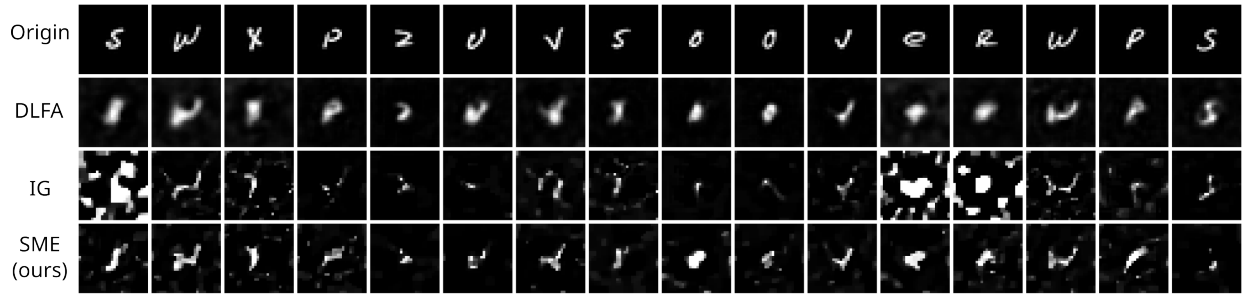
(a) E=10, N=50, R=1



(b) E=10, N=50, R=5

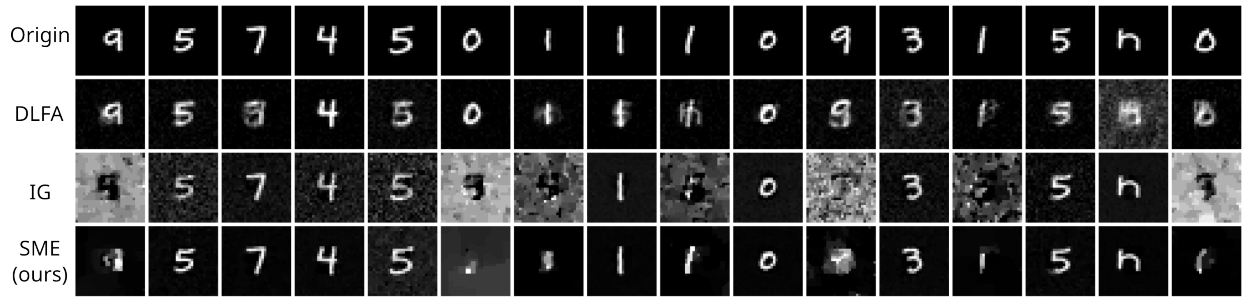


(c) E=10, N=50, R=10

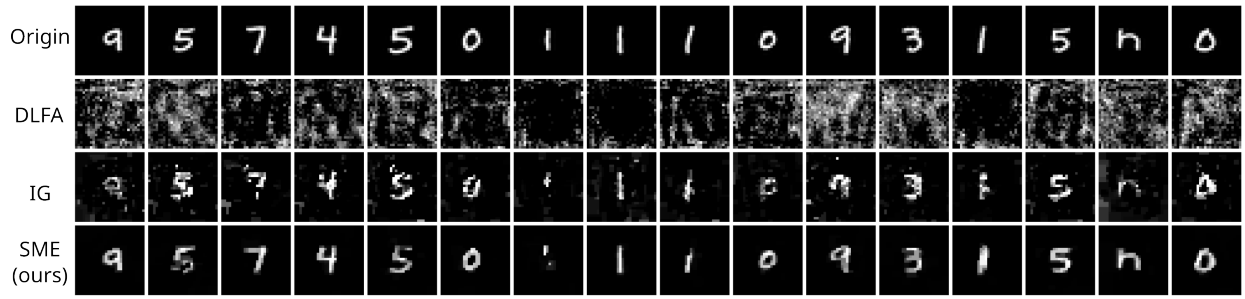


(d) E=10, N=50, R=20

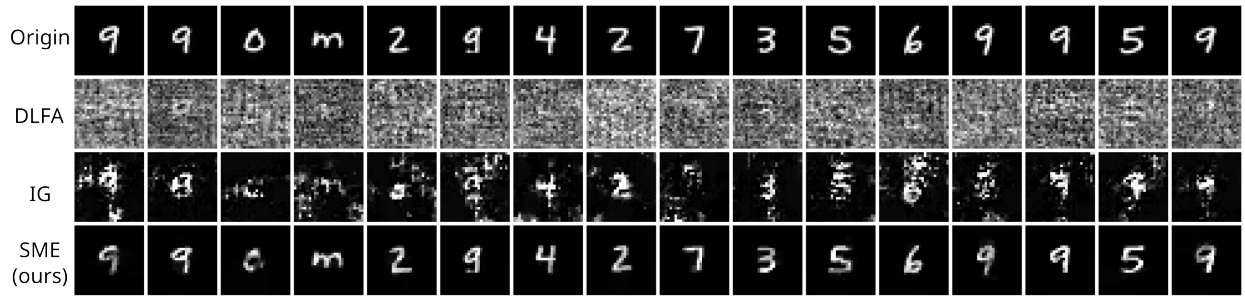
Figure 13. Visualization of the reconstructed images. The results are drawn from the settings ($E = 10$, $N = 50$, $R = \{1, 5, 10, 20\}$) in Table 6. The reconstructed images are paired with the original images through *linear sum assignment*. We randomly sample 16 out of 50 images of one reconstruction.



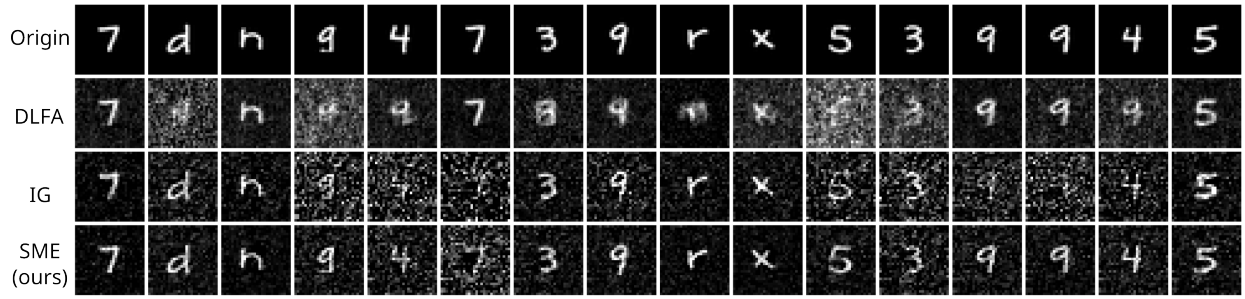
(a) MLP, E=20, N=50, $T = 100$



(b) LeNet, E=20, N=50, $T = 100$



(c) ResNet8, E=20, N=50, $T = 100$



(d) ViT, E=20, N=50, $T = 100$

Figure 14. Visualization of the reconstructed images. The results are drawn from the setting ($E = 20$, $N = 50$, $T = 100$) on the network architectures MLP, LeNet, ResNet8 and ViT in Table 7. The reconstructed images are paired with the original images through *linear sum assignment*. We randomly sample 16 out of 50 images of one reconstruction.