ReTarget: Representation Transformation via Adversarial Regularization for Geometric Misalignment

Jia-You Chen[1] and Shang-Tse Chen[1]

National Taiwan University, Taipei, ROC
{r14922a05,stchen}@csie.ntu.edu.tw

Abstract. Split inference (SI) offloads computation from edge devices and is often considered privacy-friendly since raw inputs remain local. However, recent data reconstruction attacks (DRA) show that intermediate features can be inverted to recover sensitive content. Existing defenses suppress feature information through pruning, noise injection, or decorrelation, yet strong reconstructions persist. In this paper, we argue that reconstruction vulnerability is not solely determined by mutual information between inputs and representations, but also by the local organization of representations that facilitates reliable inversion in practice. Therefore, we propose **ReTarget**, a lightweight transformation that preserves task-discriminative representations while perturbing inversion-consistent features. ReTarget combines adversarial reconstruction, task preservation, and semantic regularization training to substantially degrade inversion performance in practice without sacrificing utility. Experiments on CLIP-ViTB/16, CLIP-RN50 and DINOv2 across multiple split points and datasets show substantial degradation in reconstruction quality. Under both state-of-the-art and adaptive attacks, particularly at shallow split points where prior defenses fail, our method remains effective, where achieve up to 35% improvement while maintaining competitive task accuracy.

Keywords: split inference · reconstruction attacks · privacy-preserving learning

1 Introduction

The rapid growth in the size and computational cost of deep neural networks has made it increasingly difficult to deploy state-of-the-art models on resource-constrained edge devices such as mobile phones and IoT sensors [13, 17, 26].

To address this challenge, split inference (SI) [12, 31] was proposed as a paradigm in which a deep model is partitioned between a client and a server. The client processes the raw input locally and transmits only an intermediate activation to the server for the remaining computation, enabling efficient offloading without requiring full on-device execution.

In addition to its computational benefits, SI is widely considered privacy-friendly because raw images never leave the client device, and the server only observes intermediate feature representations [32].

However, recent studies have shown that SI is far from privacy-neutral [11, 16,18,29,35]. Due to the structure learned by neural networks, intermediate representations may preserve information that enables adversaries to perform data reconstruction attacks (DRA) aimed at recovering the client's original input.

A common line of defense such as feature pruning, noise injection, and statistical decorrelation [9,30,33] share a common assumption: privacy leakage is proportional to the amount of information retained. Consequently, they attempt to suppress or decorrelate features to minimize dependence on the input. However, prior studies show that DRA quality often remains surprisingly high [16, 29, 35].

In this work, we explore the hypothesis that DRA succeeds not merely because representations retain input-dependent information such as mutual-information, but because the resulting features preserve predictable local structure. Nearby feature vectors typically correspond to similar inputs, enabling reconstruction models to learn stable inverse mappings.

Prior observations provide motivation for this perspective. DRAG [16] shows that simply shuffling patch tokens in CLIP-ViTB/16 [8, 28], an operation that preserves most information [15], can drastically reduce DRA quality. Recent work [27] on model inversion also shows that the compatibility between an inversion procedure and model representations can affect inversion performance.

These findings suggest that DRA performance depends not only on how much information is retained, but also on the local organization of representations in feature space. Thus, we raise a fundamental question for privacy in SI:

Is suppressing retained information sufficient, or
should we consider how representations are organized in feature space?

Based on this insight, we propose **ReTarget**, **Re**presentation **T**ransformation via **A**dversarial **R**egularization for **Ge**ometric Misalignmen**t**. Fig. 1 provides an overview of our pipeline. ReTarget learns a feature transformation that preserves task relevant discriminative utility while inducing a substantial shift in representation space. Importantly, rather than suppressing information, our objective is to guide representations toward class-level prototypes, preserving task-level semantics while reducing instance-level reconstruction quality. As a result, the transformed representations remain highly effective for downstream tasks, while reconstruction quality is substantially degraded across the attacks we evaluate.

Our contributions are summarized as follows:

- We introduce ReTarget, a split inference defense that shifts the focus from information suppression to representation transformation.
- We design a training mechanism that guides representations toward class-level prototypes, preserving task-relevant discriminative information while reducing input-specific reconstruction cues.
- We demonstrate that ReTarget significantly reduces reconstruction fidelity under strong and adaptive attacks, while maintaining task accuracy.

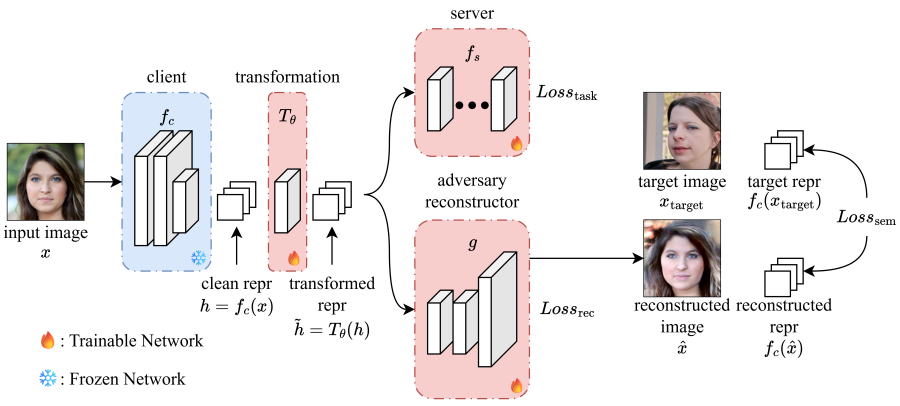

Fig. 1: Overview of ReTarget. The client sends transformed representation $\tilde{h} = T_\theta(h)$ to the server. During training, an adversarial reconstruction decoder is introduced to train T_θ via gradient reversal to perturb inversion, together with a semantic regularization term to prevent collapse. At inference time, this adversarial decoder is removed and only the client, T_θ and the server are used.

2 Related Work

2.1 Split Inference and Privacy Risk

Split inference (SI) [12, 31] has emerged as a practical paradigm that partitions a model between client and server to balance computation and communication. Beyond mobile vision applications [26], similar collaborative inference strategies have also been explored in large-scale model serving scenarios such as generative AI [24]. However, recent studies reveal that intermediate representations transmitted to the server may still leak sensitive visual information [11]. This has motivated a line of work on privacy risk analysis [6] and defense mechanisms [11, 30, 33] in split learning settings.

2.2 Reconstruction Attacks in Split Inference

Diffusion-based inversion. Recent attacks have substantially strengthened the threat model by combining powerful generative priors with representation conditioned guidance. In particular, DRAG [16] performs white-box feature inversion using guided diffusion [7], demonstrating high-fidelity reconstructions across multiple split points.

GAN and optimization-based inversion. Beyond diffusion, GAN-based [10] methods such as GLASS [18] train a generator or decoder to synthesize images that are consistent with the observed activations, often producing semantically plausible outputs. Optimization and statistical-based approaches, such as rMLE and LM-style formulations [11, 30], recover inputs by directly optimizing an image to match the observed representation under a given encoder.

Data-efficient / black-box feature inversion. Several works also explore black-box inference settings [4,29,37]. For example, FIA-FLOW [29] performs feature-space alignment using a VAE and requires access only to f_c, without knowledge of the specific model architecture.

In conclusion, although these attacks vary in priors, computational cost, and threat assumptions, they share a common requirement: the transmitted features must lie in a representation region where a stable inverse mapping can be learned or effectively optimized.

2.3 Defenses via Information Suppression and Obfuscation

Pruning and channel obfuscation. **DISCO** [30] mitigates leakage in split inference by selectively masking intermediate feature channels that encode sensitive information. Concretely, DISCO introduces a filter-generating network that predicts a binary pruning mask over convolutional channels conditioned on the current input, enabling per-sample channel obfuscation. The privatization module is decoupled from the main feature extractor: the client computes features $\hat{z} = f_1(x)$, then applies a learned gating function $z = g(\hat{z}; \phi, R)$ that masks a fraction of channels according to a pruning ratio R, and only the pruned representation z is transmitted to the untrusted server. Training follows a privacy–utility formulation where the task objective is preserved, while a proxy adversary is optimized to infer sensitive attributes. The pruning module is optimized to increase the adversary's loss subject to minimal utility degradation.

Decorrelation and information reduction. **NoPeek** [33] takes a complementary approach: instead of explicitly masking channels, it reduces invertible information by penalizing statistical dependence between raw inputs and shared activations. Specifically, NoPeek adds a regularization term with a distance correlation regularizer:

$$\mathcal{L} = \mathcal{L}_{\text{util}} + \lambda \mathcal{L}_{\text{priv}}, \qquad \mathcal{L}_{\text{priv}} = \text{dCor}(x, h), \tag{1}$$

where λ controls the privacy–utility trade-off. Distance correlation is also used as a leakage metric: stronger dependence between x and h indicates higher risk of reconstruction from transmitted activations. Reducing $\text{dCor}(x, h)$ improves robustness to reconstruction-style attacks while largely preserving task accuracy. In contrast to obfuscation methods, NoPeek aims to make transmitted representations less directly predictive of the raw input distribution overall, especially when the set of sensitive factors is unknown or incomplete at training time.

Contrast with suppression-based defenses. Unlike DISCO and NoPeek, ReTarget uses class-level prototypes as semantic targets. This encourages reconstructed outputs of the decoder to be class-consistent rather than instance-specific.

3 Main Approach

In this section, we present ReTarget, our representation transformation approach for defending against reconstruction attacks. We begin with the problem setup and then describe the proposed method in detail.

3.1 Problem Setup

Following previous work [16, 21, 30], we consider a standard split inference (SI) setting in which a client model f_c processes an input image $x \in \mathcal{X}$ into an intermediate representation

$$h = f_c(x), \tag{2}$$

which is transmitted to a server model f_s for downstream prediction

$$y = f_s(h). \tag{3}$$

Threat Model. We consider a strong white-box adversary consistent with modern data reconstruction attacks (DRA) in SI. The adversary:

- has full knowledge of the client model f_c,
- can observe a large number of transmitted representations h during inference,
- has access to auxiliary images drawn from the same data distribution,
- can train a decoder g to map representations back to the image space.

Under this threat model, the adversary aims to learn an inverse mapping

$$\hat{x} = g(h), \tag{4}$$

such that $\hat{x}$ is perceptually or semantically aligned with the true input x. Importantly, successful DRA does not require pixel-perfect recovery of the input. It only requires producing a perceptually or semantically plausible reconstruction.

This threat model covers a wide range of existing attacks, including diffusion-based inversion [16], GAN-based reconstruction [18], and data-efficient black-box feature inversion [29], as they all seek to recover an image from its representation.

Objective. Under the above threat model, the adversary aims to construct an estimate $\hat{x}$ from the observed representation h that is perceptually or semantically meaningful with respect to the original input x.

To defend against reconstruction, we modify the transmitted representation. In the deployed setting, the client sends a transformed feature $\tilde{h}$ instead of h, and thus $\tilde{h}$ becomes the attacker's observation. Our objective is to construct such a transformed representation $\tilde{h}$ such that a server trained on $\tilde{h}$ achieves performance comparable to that obtained from h, while preventing any adversary from obtaining a meaningful estimate $\hat{x}$ from $\tilde{h}$.

We consider both non-adaptive and adaptive attack settings. In the non-adaptive setting, reconstruction methods designed for clean features are directly applied to $\tilde{h}$. In the adaptive setting, the attacker incorporates the defended representation $\tilde{h}$ into the inversion pipeline. In Sec. 3.2, we realize this transformation via a parametric mapping T_θ that produces $\tilde{h} = T_\theta(h)$.

3.2 ReTarget

We view strong DRAs as being facilitated when the attacked representations lie in high-density, in-distribution regions of the feature space induced by the client encoder [1, 2, 20, 22].

Let $\mathcal{R}_h \subset \mathbb{R}^d$ denote this high-density, in-distribution subset of representation space. Since inversion models are typically trained on samples from $\mathcal{R}_h$, their inverse mappings tend to generalize reliably within this regime.

We use this geometric view as a **motivating interpretation rather than a complete theory** of reconstruction leakage. In particular, we do not claim that privacy is determined solely by whether a feature lies inside or outside $\mathcal{R}_h$. Instead, our hypothesis is that reconstruction attacks are helped by the local organization of natural-image representations where nearby or similarly structured features often correspond to visually or semantically related inputs.

Motivated by this perspective, ReTarget learns a transformation

$$T_\theta : \mathbb{R}^d \to \mathbb{R}^d$$

that aims to preserve task-relevant decision information while reducing the compatibility of the transformed features with common inversion procedures. Rather than attempting to remove all information or enforcing a large global departure from $\mathcal{R}_h$, we focus on inducing a task-preserving representation shift that modifies local feature relationships exploited by inversion models.

As a result, a server trained on $T_\theta(h)$ can achieve accuracy comparable to training on h. At the same time, across the attacks we evaluate, both learned decoders and training-free inversion methods exhibit substantially degraded reconstruction quality when applied to $T_\theta(h)$.

Instantiation of T_θ. We implement T_θ as a lightweight transformation on h that can be trained jointly with the downstream server. Specifically, we instantiate it as a residual MLP: $T_\theta(h) = h + \mathrm{MLP}(\mathrm{LN}(h))$. The MLP consists of two linear layers, each mapping $\mathbb{R}^d$ to $\mathbb{R}^d$, with a GELU activation between them. This design enables the server to adapt to the transformed representation while keeping the client encoder fixed.

Desired properties of T_θ. The transformation should satisfy:

1. **Task preservation**
   A server trained on $T_\theta(h)$ should achieve performance comparable to that obtained from h, ensuring that task-relevant decision structure is preserved.
2. **Inversion destabilization**
   The transformed representation should modify local neighborhoods in a way that prevents reconstruction models from learning stable inverse mappings. Specifically, for any decoder g trained to approximate an inverse mapping, the expected reconstruction error around $T_\theta(h)$ should remain high:

$$\mathbb{E}_x\big[\|g(T_\theta(h)) - x\|\big] \text{ is large.}$$

Training objectives of T_θ. To achieve the desired properties of T_θ, we introduce three objectives.

Adversarial reconstruction loss.

$$\hat{x} = g(T_\theta(h)), \qquad \mathcal{L}_{\text{rec}} = \|\hat{x} - x\|_2^2,$$
$$\nabla_{\phi_g}\mathcal{L}_{\text{rec}} \text{ (standard for } g), \qquad \nabla_\theta(-\mathcal{L}_{\text{rec}}) \text{ (reversed for } T_\theta).$$

We train g to *minimize* the reconstruction error, while training T_θ adversarially to *maximize* it via gradient reversal. This explicitly pushes representations toward regions where the decoder fails. We instantiate g as a lightweight transformer-based patch decoder similar to the MAE reconstruction head.

However, a purely adversarial objective against reconstruction can drive the transformed representation toward unstructured regions, causing representation collapse and degrading downstream utility. Thus, to preserve semantic structure without enabling instance-level reconstruction, we introduce a semantic consistency regularizer defined in the feature space of the fixed client encoder.

Semantic consistency loss.

$$\mathcal{L}_{\text{sem}} = 1 - \cos\big(f_c(\hat{x}), f_c(\hat{x}_{\text{target}})\big),$$

where $\hat{x}$ is the decoded image obtained from the transformed representation.

The target image x_{target} is not the original input x, but a pre-selected class-consistent prototype. For each class y, we compute the feature centroid:

$$\mu_y = \frac{1}{N_y}\sum_{i:y_i=y} f_c(x_i),$$

here, N_y denotes the number of training samples in class y, and x_{target} is the training instance which has the highest cosine similarity to μ_y. Thus, x_{target} corresponds to a point near the class prototype in the representation space.

Importantly, gradients from $\mathcal{L}_{\text{sem}}$ are applied to the transformation module while keeping the decoder fixed. The decoded image therefore acts only as a semantic proxy. By aligning $\hat{x}$ with a class-level prototype rather than the original instance, the regularizer preserves discriminative structure for classification while deliberately reducing instance-level identifiability.

Task preservation loss.

$$\mathcal{L}_{\text{task}} = \text{CE}(f_s(T_\theta(h)), y).$$

To ensure that the transformed representation remains effective for downstream prediction, we include a standard classification objective. This term preserves label-level discriminative structure in representation space, counterbalancing the inversion-disrupting transformation and ensuring that the task-relevant information is retained.

The final objective is

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \rho\mathcal{L}_{\text{task}} + \lambda\mathcal{L}_{\text{sem}}, \tag{5}$$

In all experiments, we set $\rho = 0.95$ and $\lambda = 1.0$ unless otherwise specified. Algorithm 1 summarizes the training procedure.

Algorithm 1 Training procedure of ReTarget

Require: Frozen client encoder f_c, transformation module T_θ, server model f_s, reconstruction decoder g_ϕ, training batch (x, y), class-level target x_{target}

1: Feature transformation and task prediction.
$h = f_c(x),\ h_{target} = f_c(x_{target})$.
$\tilde{h} = T_\theta(h),\ \tilde{h}_{target} = T_\theta(h_{target})$.
$\hat{y} = f_s(\tilde{h})$.
2: Adversarial reconstruction.
$\hat{x} = g_\phi(\tilde{h}),\ \hat{x}_{target} = g_\phi(\tilde{h}_{target})$.
3: Optimization losses.
$\mathcal{L}_{\text{rec}} = \|\hat{x} - x\|_2^2$.
$\mathcal{L}_{\text{task}} = \text{CE}(\hat{y}, y)$.
$\mathcal{L}_{\text{sem}} = 1 - \cos\big(f_c(\hat{x}), f_c(\hat{x}_{\text{target}})\big)$.
4: Update the decoder g_ϕ by minimizing $\mathcal{L}_{\text{rec}}$, with f_c, T_θ, and f_s fixed.
5: Update T_θ and f_s by minimizing $\rho\mathcal{L}_{\text{task}} + \lambda\mathcal{L}_{\text{sem}} - \mathcal{L}_{\text{rec}}$, with f_c and g_ϕ fixed. The negative reconstruction term is implemented using gradient reversal.
6: At inference time, discard g_ϕ and transmit only $\tilde{h} = T_\theta(f_c(x))$ to the server.

4 Experiments

4.1 Datasets and Models

Defense training dataset. Following common practice in split inference defenses, we train the transformation on ImageNet-1K [5], which provides diverse semantic categories and enables learning task-preserving transformations at scale.

Attack evaluation datasets. To evaluate reconstruction attacks under realistic and diverse image distributions, we follow the evaluation protocol of DRAG [16] and conduct attacks on three datasets: ImageNet-1K [5], MS COCO [19], and FFHQ [14]. For fair comparison with DRAG [16], we use the same set of randomly sampled images released in their evaluation, consisting of 10 images per dataset.

Backbone model and split settings. We adopt CLIP-ViTB/16 [28] as the backbone encoder, which is widely used in recent reconstruction attack studies. Following prior work [16], we evaluate split inference at four different transformer block depths: layers 3, 6, 9, and 12. Note that we also conduct experiments on CLIP-RN50 and DINOv2. Details are provided in the supplementary materials.

4.2 Baselines and Metrics

Reconstruction attacks. We evaluate against state-of-the-art reconstruction attacks representing different paradigms:

- **DRAG** [16]: a diffusion-based white-box reconstruction attack.
- **DRAG++** [16]: an enhanced variant of DRAG that uses a learned inverse network to obtain a coarse initial estimate.
- **GLASS** [18]: a GAN-based feature inversion attack.
- **rMLE, LM** [11, 30]: statistical and optimization-based inversion methods.

Defense baselines. We compare ReTarget with representative defenses designed to mitigate reconstruction attacks by suppressing or obfuscating information in intermediate representations:

- **DISCO** [30]: channel obfuscation via dynamic feature pruning.
- **NoPeek** [33]: information leakage reduction through feature decorrelation.

For DISCO, we set the pruning hyper-parameters as $\rho = 0.75$ and the pruning ratio $R = 0.50$ following the original implementation. For NoPeek, the distance-correlation loss weight was set to $\lambda = 10.0$, and the distance metric used was cosine similarity.

Evaluation metrics. Reconstruction quality is measured using standard perceptual similarity metrics:

- MS-SSIM [34]
- LPIPS [36]
- Feature similarity based on DINO ViT-S/16 [3] embeddings

These metrics jointly measure structural similarity, perceptual distance, and semantic alignment between reconstructed images and the original inputs.

Table 1: Image similarity metrics grouped by split point. Lower MS-SSIM together with higher LPIPS indicate stronger perturbation and better privacy. **Bold** values indicate the best result among methods for the same layer and metric.

		Dataset: MSCOCO [19]			**Dataset: FFHQ [14]**		
Method	Split Point	MS-SSIM ↓	LPIPS ↑	DINO ↓	MS-SSIM ↓	LPIPS ↑	DINO ↓
DISCO	Layer 3	0.7818	0.1025	0.9314	0.8942	0.0522	0.9645
NoPeek		0.3760	0.4690	0.4263	0.6187	0.3095	0.5999
Ours		**0.0209**	**0.6337**	**0.1695**	**0.0238**	**0.6170**	**0.1599**
DISCO	Layer 6	0.5683	0.2195	0.9003	0.8986	0.0546	0.9611
NoPeek		0.3704	0.3923	0.6361	0.6430	0.2025	0.8177
Ours		**0.0588**	**0.7665**	**0.1869**	**0.2018**	**0.7032**	**0.3507**
DISCO	Layer 9	0.6963	0.1587	0.9537	0.9125	0.0555	0.9553
NoPeek		0.4244	0.3993	0.7907	0.6479	0.2341	0.8135
Ours		**0.2556**	**0.6218**	**0.5873**	**0.5747**	**0.2935**	**0.8027**
DISCO	Layer 12	0.5115	0.2960	0.8777	0.6372	0.2288	0.8126
NoPeek		0.0694	0.7645	0.2558	0.2940	0.5060	0.5054
Ours		**0.0134**	**0.8389**	**0.1846**	**0.0244**	**0.7285**	**0.2263**

4.3 Reconstruction Attack Evaluation

We evaluate privacy leakage by measuring how well a strong adversary can reconstruct the original input image from the transmitted intermediate representation. Unless otherwise specified, we evaluate privacy leakage using a strong white-box reconstruction attack, DRAG [16], which represents the state-of-the-art diffusion-based inversion method in split inference.

Following the threat model in Sec. 3.1 and the setup in Sec. 4.1–4.2, the attacker observes the client representation $h = f_c(x)$ and applies a reconstruction procedure to obtain $\hat{x}$ from h. We report results across multiple split points (layers 3/6/9/12 of CLIP-ViTB/16 [28]) to cover shallow-to-deep representations.

Quantitative results. Table 1 summarizes reconstruction quality grouped by split point using the metrics defined in Sec. 4.2. Lower MS-SSIM [34] together with higher LPIPS [36] indicate stronger privacy protection, while DINO [3] feature similarity measures semantic alignment between $\hat{x}$ and x.

Across all split points, our method consistently degrades reconstruction fidelity compared to DISCO [30] and NoPeek [33], achieving the lowest similarity and the highest distance. We also observe strong degradation at shallow layers 3 and 6, where reconstructions are typically easiest due to rich low-level cues [23].

Qualitative comparison. Fig. 2 shows reconstructions produced by DRAG on MS COCO [19] and GLASS [18] on FFHQ [14], while Fig. 3 illustrates results of rMLE [11] and LM [30] on ImageNet-1K [5]. Across most settings, DISCO and NoPeek tend to preserve recognizable semantic content from the original inputs.

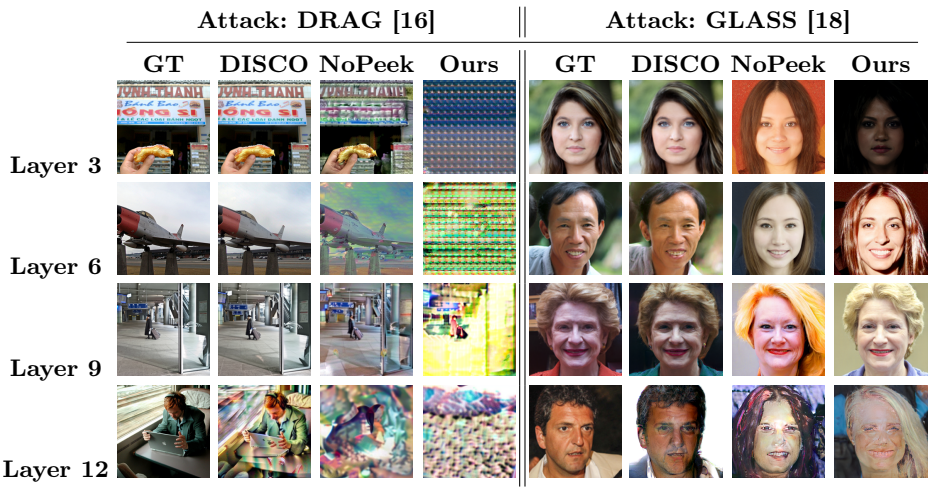

Fig. 2: Qualitative comparison of DRAG and GLASS attacks across different layers. Columns 1–4 show results of DRAG on MS COCO, while 5–8 show results of GLASS on FFHQ. Our method produces reconstructions that deviate from the original inputs.

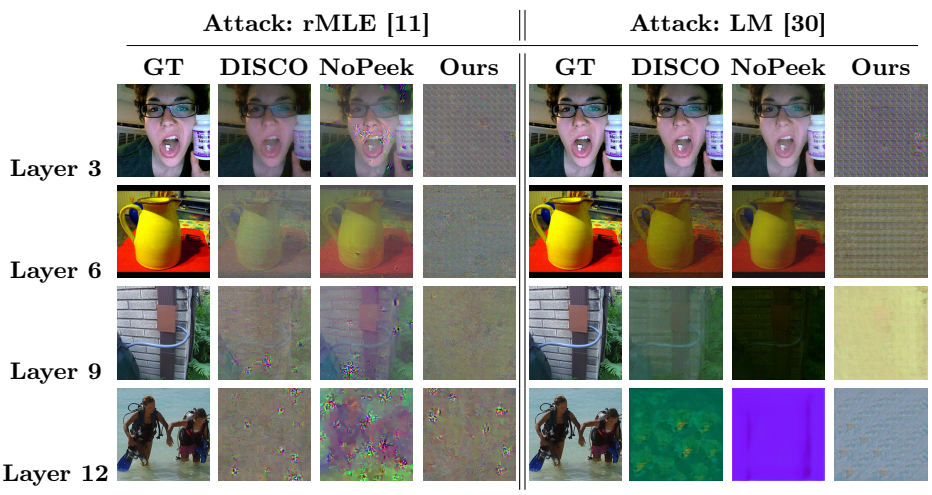

Fig. 3: Qualitative comparison of rMLE and LM attacks across different layers. Both are on ImageNet-1K, columns 1–4 show results of rMLE, while 5–8 show results of LM. Our method produces reconstructions that deviate from the original inputs.

In contrast, our method consistently produces reconstructions that deviate substantially from the ground-truth images. We observe that, under the GLASS attack, NoPeek occasionally provides a stronger defense. One possible explanation is that the stronger GAN prior [10] may generate visually plausible, proxy-like outputs without faithfully recovering the subject's identity.

4.4 Evidence of Feature Space Shift with Preserved Utility

To verify that ReTarget induces representation shifts in feature space rather than merely suppressing information, we analyse representation changes under a frozen server and classifier head. By keeping the decision boundary fixed, any change in prediction reflects displacement of representations in feature space, rather than adaptation of the downstream model.

Specifically, for each input image, we compute the clean representation h and the transformed representation $T_\theta(h)$, and feed both through the same unchanged server. We then quantify semantic shift using multiple metrics.

Table 2 reports four displacement measures under the frozen classifier. Flip@1 measures how often the Top-1 prediction changes, while Overlap@5 captures the consistency of Top-5 predictions. Logit L2 and cosine shift quantify the magnitude of representation displacement reflected in the induced logit changes under the frozen classifier. Classification accuracy is also reported to verify utility. We compare against Patch Token Dropping from DRAG [16] and suppression based defenses including DISCO [30] and NoPeek [33].

At shallow layers 3 and 6, our method induces substantially larger semantic displacement than all baselines while maintaining competitive accuracy.

Moreover, at deeper layers 9 and 12, the magnitude of feature-space displacement decreases, yet reconstruction quality remains strongly degraded. This behavior aligns with prior information-theoretic observations [23] that deeper representations preserve less input-specific detail. In such cases, even small shifts in feature space may suffice to destabilize reconstruction.

Importantly, semantic shift alone does not guarantee privacy protection. It must be interpreted jointly with reconstruction results. In particular, DISCO and NoPeek exhibit nontrivial semantic displacement across layers, yet achieve weaker reconstruction degradation. This suggests that feature-space shifts, rather than arbitrary perturbations, are important for destabilizing inversion.

To further disentangle feature-space effects from information suppression, we measure changes in task and instance level mutual information, estimated via an InfoNCE [25] lower bound. Table 3 reports $\Delta I(Y;H)$ and $\Delta I(X;H)$.

Our method consistently reduces instance-level dependence $\Delta I(X;H)$ at shallow layers while preserving or even slightly increasing task-level dependence $\Delta I(Y;H)$. Notably, DISCO achieves larger reductions in $\Delta I(X;H)$ at certain layers, yet provides weaker reconstruction protection. This discrepancy suggests that this mutual-information proxy alone is not sufficient to predict reconstruction vulnerability across methods and layers.

Overall, the results show that info suppression metrics and the measured feature shift statistics alone do not fully predict reconstruction vulnerability.

Table 2: Semantic shift metrics under frozen classifier across split layers. The metrics quantify how much the feature representation changes. **Bold** values indicate the best result among methods for the same layer and metric.

Method	Layer	flip@1 ↑	overlap@5 ↓	logit L2 ↑	cos shift ↑	Acc ↑
Patch Token Dropping [16]	3	0.1954	0.7335	24.64	0.0693	0.7103
	6	0.1521	0.7871	20.14	0.0504	0.7375
	9	0.1166	0.8249	17.54	0.0374	0.7575
	12	0.0000	1.0000	0.000	0.0000	**0.7894**
DISCO [30]	3	0.1888	0.7384	25.45	0.0715	0.7207
	6	0.1719	0.7505	25.38	0.0717	0.7333
	9	**0.1909**	**0.7149**	33.61	0.1211	0.7261
	12	**0.1572**	**0.7352**	**42.28**	**0.1895**	0.7437
NoPeek [33]	3	0.1692	0.7484	27.27	0.0721	0.7459
	6	0.1762	0.7366	28.83	0.0788	0.7406
	9	0.1734	0.7289	33.31	0.1031	0.7467
	12	0.1394	0.7783	26.40	0.0713	0.7598
Ours	3	**0.2465**	**0.6516**	**43.39**	**0.1529**	**0.7461**
	6	**0.2460**	**0.6474**	**46.96**	**0.1784**	**0.7574**
	9	0.1634	0.7315	**39.51**	**0.1248**	**0.7874**
	12	0.0579	0.8943	21.09	0.0294	0.7862

Table 3: Change in task-level and instance-level mutual information across split layers. $\Delta I(Y;H)$ denotes task-level MI change, and $\Delta I(X;H)$ denotes instance-level MI change (InfoNCE lower bound). **Bold** values indicate the best result among methods for the same layer and metric.

Method	Layer	$\Delta I(Y;H) \uparrow$	$\Delta I(X;H) \downarrow$	Layer	$\Delta I(Y;H) \uparrow$	$\Delta I(X;H) \downarrow$
Patch Token Dropping	3	-0.346	-0.392	9	-0.153	-0.195
	6	-0.238	-0.316	12	**0.000**	0.000
DISCO	3	-0.293	-0.395	9	-0.270	**-0.771**
	6	-0.232	-0.416	12	-0.227	**-0.409**
NoPeek	3	-0.196	-0.394	9	-0.186	-0.583
	6	-0.215	-0.443	12	-0.120	-0.369
Ours	3	**0.065**	**-0.763**	9	**0.080**	-0.352
	6	**0.079**	**-0.763**	12	-0.004	0.019

Under the evaluated attacks, ReTarget nevertheless provides a favorable privacy–utility trade-off.

4.5 Local Neighborhood Disruption and Reconstruction

We next examine whether ReTarget changes local neighborhoods in the full transmitted representation. We use a class-balanced set of 5,000 ImageNet validation images. For each image, we flatten and ℓ_2-normalize all transmitted tokens, compute its k nearest neighbors in the clean and transformed spaces separately, and measure their overlap. We report Disrupt@10 $= 1 -$ Overlap@10, where a larger value indicates that fewer of an image's clean nearest neighbors remain nearest neighbors after transformation. Table 4 shows that ReTarget substantially changes local neighborhoods at every split point.

Relative to $\lambda = 0$, the prototype-guided variant has higher disruption at all layers while maintaining comparable accuracy. These results are consistent with the interpretation that the transformation changes representation relationships, but do not identify which changes are responsible for reconstruction degradation.

We also conduct an exploratory sample-level analysis on the available reconstruction evaluations. For each attacked image, we compute Disrupt@10 against a fixed 1,000-image MSCOCO reference pool and correlate it with LPIPS, MS-SSIM, and DINO similarity using Spearman's ρ. The correlations are mixed across layers and metrics, indicating that local-neighborhood disruption alone is not a sufficient sample-level predictor of reconstruction difficulty. This result motivates treating the geometric account as an interpretation of the observed behavior rather than a complete theory of reconstruction leakage.

4.6 Robustness to Adaptive Attacks

A natural concern is whether our defense remains effective when the adversary *adapts* to the deployed transformation.

Table 4: Full-token local-neighborhood disruption on a class-balanced 5,000-image ImageNet validation set. $\lambda = 0$ denotes the variant without the prototype-guided.

Method	Layer	Overlap ↓	Disrupt ↑	Layer	Overlap ↓	Disrupt ↑
$\lambda = 0$	3	0.303	0.697	9	0.754	0.246
	6	0.243	0.757	12	0.499	0.501
Ours	3	**0.144**	**0.856**	9	**0.710**	**0.290**
($\lambda = 1$)	6	**0.207**	**0.793**	12	**0.329**	**0.671**

In particular, we consider an adaptive attacker that train a decoder on the *defended* representations $\tilde{h} = T_\theta(h)$, rather than assuming a decoder trained only on clean features. This threat model matches the strong white-box setting in Sec. 3.1, and is consistent with modern reconstruction attacks in split inference where the attacker can collect many transmitted representations and learn an inversion model.

Adaptive attack setup. We instantiate an adaptive variant of DRAG [16] denoted DRAG++ by training a feature-to-image reconstructor on $\tilde{h}$ and using it to initialize the diffusion optimization. We keep the remaining components and hyperparameters identical to Sec. 4.2 for a fair comparison, and evaluate across the same split layers and datasets as in Sec. 4.1.

Results. As shown in Fig. 4 and table 5, allowing the attacker to train a decoder on defended features improves DRA quality compared to non-adaptive attacks. However, our method still yields substantially degraded reconstructions across layers, with outputs deviating significantly from the ground-truth images.

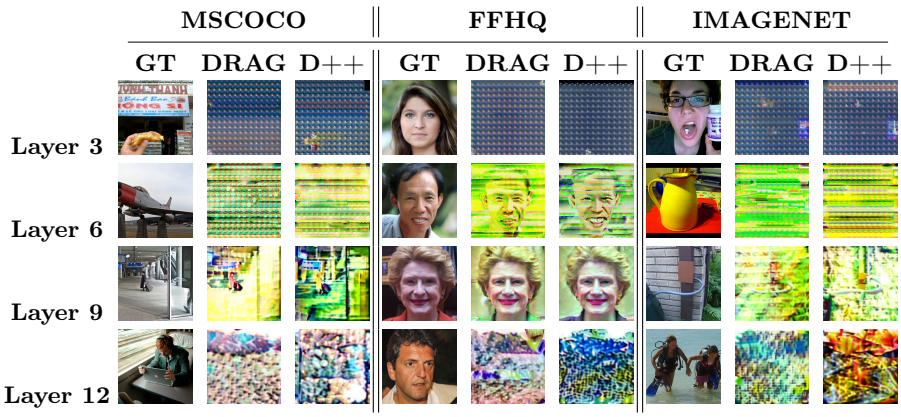

Fig. 4: Qualitative evaluation of defense performance under adaptive attacks across different layers. D++ denotes DRAG++, an adaptive variant of DRAG in which the decoder is retrained on the defended representations.

Table 5: Adaptive-attack reconstruction metrics on MSCOCO [19] grouped by split point. **S.L.** denotes the split point layer. Lower MS-SSIM and DINO, together with higher LPIPS, indicate better privacy. **Bold** values indicate the best result.

Dataset: MSCOCO [19]								
Attack	S.L.	MS-SSIM ↓	LPIPS ↑	DINO ↓	S.L.	MS-SSIM ↓	LPIPS ↑	DINO ↓
DRAG [16]	3	**0.0209**	0.6337	**0.1695**	9	0.2556	0.6218	0.5873
DRAG++		0.0481	**0.6435**	0.1727		**0.1982**	**0.6508**	**0.5460**
DRAG	6	0.0588	**0.7665**	**0.1869**	12	**0.0134**	**0.8389**	**0.1846**
DRAG++		**0.0375**	0.7483	0.1888		0.0188	0.8558	0.1969

This demonstrates that our defense is not merely exploiting a fixed decoder mismatch, but remains robust against adaptive training.

5 Conclusion

In this paper, we proposed ReTarget, motivated by the observation that reconstruction in split inference depends not only on how much information is retained, but also on how representations are structured in feature space. Experiments demonstrate a favorable privacy–utility trade-off, particularly at shallow split layers, where we achieve up to 35% improvement in LPIPS (0.4690 to 0.6337) while maintaining competitive accuracy (0.7461). Our mutual-information analysis further reveals a clear decoupling between instance-level and task-level dependence at shallow layers ($\Delta I(X;H)$ decreases from -0.395 to -0.763, while $\Delta I(Y;H)$ increases from -0.196 to 0.065). Importantly, we observe that baselines achieve even larger reductions in $\Delta I(X;H)$ at deeper layers, yet fail to obtain comparable reconstruction degradation. This suggests that information suppression alone cannot fully explain inversion robustness. Taken together, these findings indicate that systematically shifting representations in feature space to destabilize inversion is an effective strategy for mitigating reconstruction attacks in split inference.

Acknowledgements

This work was supported in part by the National Science and Technology Council (NSTC) under Grants NSTC 114-2634-F-002-004, NSTC 114-2634-F-002-003-MBK, and NSTC 115-2628-E-002-029. We also thank the National Center for High-performance Computing (NCHC) for providing computational and storage resources.

References

1. Asperti, A., Tonelli, V.: Comparing the latent space of generative models. Neural Computing and Applications **35**, 3155–3172 (2023)

2. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**, 1798–1828 (2013)
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
4. Chen, D., Li, S., Zhang, Y., Li, C., Kundu, S., Beerel, P.A.: Dia: Diffusion based inverse network attack on collaborative inference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 124–130 (2024)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 248–255 (2009)
6. Deng, R., Lu, Z., Duan, Q., Hu, S.: Quantifying privacy leakage in split inference via fisher-approximated shannon information analysis. arXiv preprint arXiv:2504.10016 (2025)
7. Dhariwal, P., Nichol, A.Q.: Diffusion models beat GANs on image synthesis. In: Advances in Neural Information Processing Systems. vol. 34, pp. 8780–8794 (2021)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
9. Duan, L., Sun, J., Jia, J., Chen, Y., Gorlatova, M.: Reimagining mutual information for enhanced defense against data leakage in collaborative inference. In: Advances in Neural Information Processing Systems. vol. 37, pp. 44479–44500 (2024)
10. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. vol. 27, pp. 2672–2680 (2014)
11. He, Z., Zhang, T., Lee, R.B.: Model inversion attacks against collaborative inference. In: Proceedings of the 35th annual computer security applications conference. pp. 148–162 (2019)
12. Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J., Tang, L.: Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. ACM SIGARCH Computer Architecture News **45**, 615–629 (2017)
13. Karjee, J., Naik, P., Anand, K., Bhargav, V.N.: Split computing: Dnn inference partition with load balancing in iot-edge platform for beyond 5g. Measurement: Sensors **23**, 100409 (2022)
14. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
15. Kutscher, D., Chan, D.M., Bai, Y., Darrell, T., Gupta, R.: REOrdering patches improves vision models. In: Advances in Neural Information Processing Systems (2025)
16. Lei, W.K., Chen, J.C., Chen, S.T.: DRAG: Data reconstruction attack using guided diffusion. In: Forty-second International Conference on Machine Learning (2025)
17. Li, E., Zeng, L., Zhou, Z., Chen, X.: Edge ai: On-demand accelerating deep neural network inference via edge computing. IEEE transactions on wireless communications **19**, 447–457 (2019)
18. Li, Z., Yang, M., Liu, Y., Wang, J., Hu, H., Yi, W., Xu, X.: Gan you see me? enhanced data reconstruction attacks against split inference. In: Advances in neural information processing systems. vol. 36, pp. 54554–54566 (2023)

19. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755 (2014)
20. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5188–5196 (2015)
21. Na, H., Choi, D.: Trapmi: A data protection method to resist model inversion attacks in split learning. IEEE Access (2025)
22. Nash, C., Kushman, N., Williams, C.K.: Inverting supervised representations with autoregressive neural density models. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 1620–1629 (2019)
23. Nash, C., Kushman, N., Williams, C.K.: Inverting supervised representations with autoregressive neural density models. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 1620–1629 (2019)
24. Ohta, S., Nishio, T.: Lambda-split: A privacy-preserving split computing framework for cloud-powered generative ai. arXiv preprint arXiv:2310.14651 (2023)
25. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
26. Palanisamy, K., Khimani, V., Moti, M.H., Chatzopoulos, D.: Spliteasy: A practical approach for training ml models on mobile devices. In: Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications. pp. 37–43 (2021)
27. Peng, X., Han, B., Yu, F., Liu, T., Liu, F., Zhou, M.: Generative model inversion through the lens of the manifold hypothesis. In: Advances in Neural Information Processing Systems (2025)
28. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763 (2021)
29. Ren, Z., He, L., Liang, J., Fu, X., Bi, H., Li, F.: What your features reveal: Data-efficient black-box feature inversion attack for split dnns. arXiv preprint arXiv:2511.15316 (2025)
30. Singh, A., Chopra, A., Garza, E., Zhang, E., Vepakomma, P., Sharma, V., Raskar, R.: Disco: Dynamic and invariant sensitive channel obfuscation for deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12125–12135 (2021)
31. Teerapittayanon, S., McDanel, B., Kung, H.T.: Distributed deep neural networks over the cloud, the edge and end devices. In: 2017 IEEE 37th international conference on distributed computing systems. pp. 328–339 (2017)
32. Vepakomma, P., Gupta, O., Swedish, T., Raskar, R.: Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564 (2018)
33. Vepakomma, P., Singh, A., Gupta, O., Raskar, R.: Nopeek: Information leakage reduction to share activations in distributed deep learning. In: 2020 International Conference on Data Mining Workshops. pp. 933–942 (2020)
34. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The thrity-seventh asilomar conference on signals, systems & computers, 2003. vol. 2, pp. 1398–1402 (2003)
35. Xu, X., Yang, M., Yi, W., Li, Z., Wang, J., Hu, H., Zhuang, Y., Liu, Y.: A stealthy wrongdoer: Feature-oriented reconstruction attack against split learning. In: Pro-

ceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12130–12139 (2024)
36. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 586–595 (2018)
37. Zhang, S.Q., Li, Z., Guo, C., Mahloujifar, S., Dangwal, D., Suh, G.E., Salvo, B.D., Liu, C.: Unlocking visual secrets: Inverting features with diffusion priors for image reconstruction. Transactions on Machine Learning Research (2025)