PLOS One

RESEARCH ARTICLE

Paint: Accurate, efficient, and sparse Model Inversion via peripheral sparsification and anchor alignment

Jaehyeon Choi, Minjun Kim, U Kang*

Data Mining Lab, Seoul National University, Seoul, South Korea

* ukang@snu.ac.kr

Check for updates

OPEN ACCESS

Citation: Choi J, Kim M, Kang U (2026) Paint: Accurate, efficient, and sparse Model Inversion via peripheral sparsification and anchor alignment. PLoS One 21(11): e0362597. https://doi.org/10.1371/journal.pone.0362597

Editor: Xizi Chen, Huazhong Agricultural University, CHINA

Received: April 15, 2026

Accepted: October 26, 2026

Published: November 17, 2026

Copyright: © 2026 Choi et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data availability statement: The code is available on GitHub via https://github.com/snudm-starlab/Paint. The following existing data sources were used: ImageNet from the Hugging Face Datasets repository, available via https://huggingface.co/datasets/ILSVRC/imagenet-1k, and CIFAR from the University

Abstract

How can we reconstruct training data from a pre-trained model without access to the original dataset? Model Inversion (MI) attempts to synthesize samples similar to the training dataset. This capability is particularly useful in data-free applications, such as data-free quantization and data-free knowledge transfer, where synthetic data replace inaccessible training samples. For these applications, MI needs to generate high-fidelity samples to preserve semantic information while remaining computationally efficient, enabling scalable generation for reliable downstream performance. However, existing MI methods fail to achieve both high fidelity and efficiency simultaneously. In this paper, we propose PAINT (**P**eripheral sparsification and **A**nchor-aligned **IN**version **T**echnique), an accurate, efficient, and sparse MI framework for generating high-fidelity synthetic data. PAINT integrates low-pass filtering during inversion to suppress high-frequency noise, aligns salient regions toward anchor with peripheral sparsification to preserve object structures during sparse inversion, and exploits soft targets to better capture the predictive uncertainty. Extensive experiments demonstrate that PAINT improves inversion efficiency for ViTs and achieves accuracy gains of up to 5.33%p in data-free scenarios.

Introduction

How can we reconstruct training data from a pre-trained model without relying on its original training data? Model Inversion (MI) generates synthetic data that induce target predictions under a pre-trained model. These synthetic data are useful in data-constrained settings due to privacy, licensing, or policy restrictions, including Data-Free Quantization (DFQ) [1,2] and Data-Free Knowledge Transfer (DFKT) [3–5], where they serve as surrogates for inaccessible training samples. Overall, MI provides a practical solution when direct access to training data is limited.

Existing MI methods [3,6,7] adopt dense MI strategies that reconstruct the entire image. Although these approaches produce plausible samples, they become

of Toronto website, available via https://cave.cs.toronto.edu/kriz/cifar.html.

Funding: This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) [No. RS-2020-II200894, Flexible and Efficient Model Compression Method for Various Applications and Environments], [No. RS-2025-25442338, AI Star Fellowship Support Program (Seoul National University)], [No. RS-2024-00469482 and No. RS-2024-00509257, Global AI Frontier Lab], and [No. RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)]. The Institute of Engineering Research at Seoul National University provided research facilities for this work. The ICT at Seoul National University provided research facilities for this study. U Kang is the corresponding author. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

computationally inefficient for high-resolution inversion with large-scale models. Dense inversion limits scalability under a fixed budget by reconstructing background regions irrelevant to target predictions. As shown in Fig 1(a), saliency is concentrated on only a small subset of image patches, suggesting that much of this computation is redundant. Reducing such redundancy is important because, in data-free settings, downstream performance is closely related to the number of synthetic samples [1]. Fig 1(b) shows that generating more samples consistently improves accuracy. This observation suggests that improving inversion efficiency is critical for scalable synthetic data generation and downstream performance.

To improve efficiency, SMI [8] proposes a sparse MI strategy that selectively reconstructs informative patches instead of the entire image. Consequently, sparse inversion achieves higher downstream performance under a fixed computational budget by generating more synthetic samples than dense inversion. However, this strategy selects patches solely based on importance scores, without considering spatial or structural coherence. As shown in Fig 2(b), SMI often fragments object structures in synthesized images, degrading structural fidelity.

We propose PAINT (**P**eripheral sparsification and **A**nchor-aligned **IN**version **T**echnique), an accurate, efficient, and sparse MI method that addresses the limitations of existing inversion methods. PAINT suppresses high-frequency noise via low-pass filtering during inversion. It also enables structured sparse inversion while preserving object structures by aligning salient regions toward an anchor and sparsifying peripheral areas. As shown in Fig 2(c), this alignment strategy prevents object fragmentation and yields structurally coherent reconstructions compared to SMI. PAINT further captures predictive uncertainty through soft targets, generating more informative samples. Table 1 compares PAINT and other inversion methods by various aspects. Experimental results demonstrate that PAINT improves inversion efficiency for ViTs while maintaining competitive or superior downstream accuracy in DFQ (Tables 3 and 4) and DFKT (Table 5) by up to 5.33%p over existing methods. Paint supports both CNN and ViT architectures, overcoming the architectural constraints of prior sparse MI methods.

Our contributions are summarized as follows:

- **Observation.** We identify three key challenges in existing MI methods: (1) excessive high-frequency noise in synthetic images, (2) structural fragmentation under sparse inversion, and (3) mismatched confidence distributions compared to real images (see Figs 2, 4, and 5).
- **Algorithm.** We propose PAINT, an accurate, efficient, and sparse MI framework that overcomes the aforementioned limitations. PAINT adopts low-pass filtering during inversion to suppress high-frequency noise, aligns salient regions toward an anchor with peripheral sparsification to preserve object structure under efficiency constraints, and exploits soft targets instead of hard labels to better capture predictive uncertainty.
- **Experiments.** We experimentally show that PAINT improves inversion efficiency for ViTs and achieves competitive or superior performance in DFQ and DFKT across various models and datasets, achieving classification accuracy improvement of up to 5.33%p.

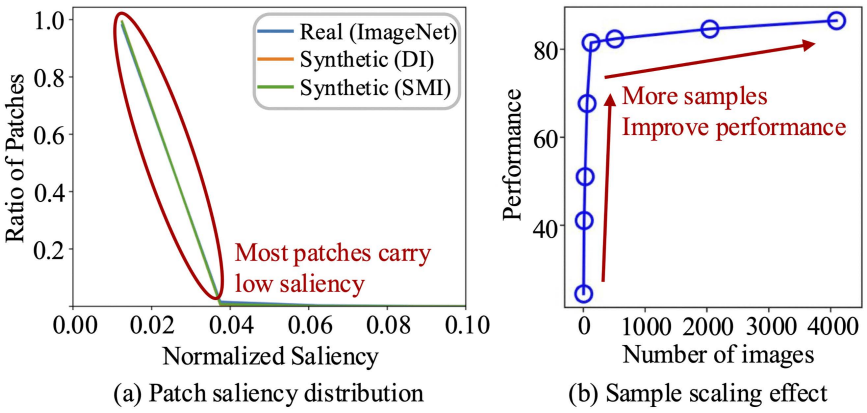

Fig 1. Observations on synthetic samples. (a) distribution of normalized saliency values across image patches and (b) downstream performance with respect to the number of synthetic samples.

https://doi.org/10.1371/journal.pone.0362597.g001

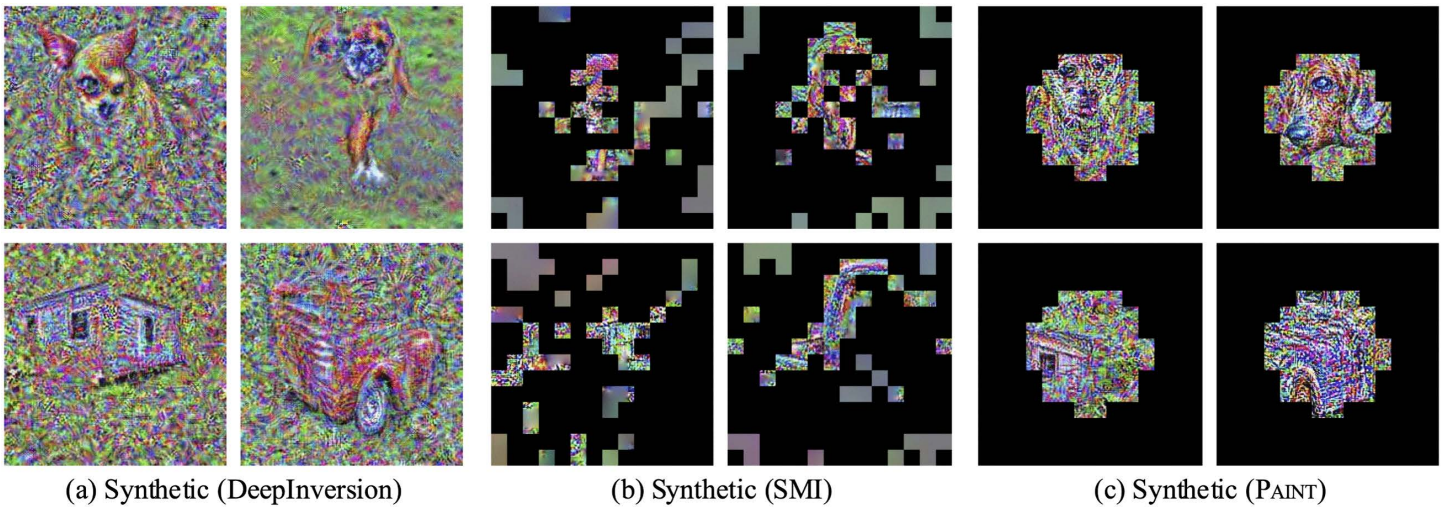

Fig 2. Comparison of synthetic images generated by different inversion methods. (a) DeepInversion reconstructs the entire image, often introducing irrelevant background patterns. (b) SMI sparsely reconstructs salient patches, but often fragments object structures. (c) In contrast, PAINT better preserves object structure while reducing irrelevant background generation.

https://doi.org/10.1371/journal.pone.0362597.g002

Table 1. Qualitative comparison of PAINT and other inversion algorithms. PAINT is the only sparse inversion approach applicable to both CNNs and ViTs.

	DeepInversion	SMI	Paint (proposed)
Inversion Strategy	Dense	**Sparse**	**Sparse**
Structural Integrity	**High**	Low	**High**
Support CNN?	**Yes**	No	**Yes**
Support ViT?	**Yes**	**Yes**	**Yes**
Inversion Efficiency	Low	**High**†	**High**†

†: Computation reduction through sparsification applies only to ViTs.

https://doi.org/10.1371/journal.pone.0362597.t001

In the rest of the paper, we provide the preliminaries and related work, describe our proposed method, evaluate the proposed method and competitors on various datasets, and conclude this work. The source code of PAINT and datasets are available at https://github.com/snudm-starlab/Paint.

Preliminaries and related works

We provide background about Model Inversion (MI) and present data-free applications, including Data-Free Quantization (DFQ) and Data-Free Knowledge Transfer (DFKT). Table 2 presents notations frequently used in this paper.

Model inversion

MI aims to synthesize inputs that resemble the original training data given a pre-trained model. MI originated as a privacy attack aimed at recovering sensitive training data from model outputs [9–12]. More recently, MI has been increasingly adopted in data-inaccessible scenarios to generate surrogate samples for downstream tasks such as DFQ [1,2,13–15] and DFKT [3–5,16,17].

Dense inversion. Most existing MI methods [6,18] follow a dense inversion paradigm that directly optimizes the entire image via iterative gradient-based updates. Specifically, given a pre-trained model $\mathcal{M}_\theta$, MI initializes a synthetic input $\mathbf{X}$ from Gaussian noise and then updates $\mathbf{X}$ by minimizing an inversion loss $\mathcal{L}_{\text{MI}}$ to induce $\mathcal{M}_\theta(\mathbf{X})$ to predict the target label y. $\mathcal{L}_{\text{MI}}$ is typically formulated as Eq (1).

$$\min_{\mathbf{X}} \mathcal{L}_{\text{MI}} = \mathcal{L}_{\text{cls}}(\mathcal{M}_\theta(\mathbf{X}), y) + \lambda_{\text{img}}\mathcal{L}_{\text{img}}(\mathbf{X}), \tag{1}$$

where classification loss $\mathcal{L}_{\text{cls}}$ encourages $\mathbf{X}$ to be classified as the target label y, image regularization $\mathcal{L}_{\text{img}}$ promotes spatial smoothness and constrains pixel magnitudes, and λ_{img} balances the contribution of the regularization term.

Table 2. Frequently used notations.

Symbol	Description
$\mathcal{M}_\theta$	pre-trained model
$\mathbb{D}_{\text{syn}} = \{(\mathbf{X}_i, y_i)\}_{i=1}^{N}$	synthetic dataset
$\mathbf{X}$, y	synthetic image and corresponding target label
$\mathbf{X}^F$	filtered synthetic image
$\mathcal{G}$	Gaussian low-pass filter
R_c	cutoff radius in the frequency domain
ρ_c	cutoff ratio
$\mathcal{F}(\cdot)$, $\mathcal{F}^{-1}(\cdot)$	Fourier and inverse Fourier transforms
$D_f(u, v)$	Euclidean distance between (u,v) and the center in frequency domain
$\mathbf{S}(i, j)$	saliency weight at location (i, j)
(a_x, a_y)	anchor coordinate
$D_s(i,j)$	Euclidean distance between (i,j) and the anchor (a_x, a_y) in spatial domain
$\mathbf{M}_t$	distance-based mask matrix at iteration t
d_t	distance threshold at iteration t
p_t	sparsity ratio at iteration t
y^{soft}	soft target label
τ	hyperparameter to control target confidence
K	the number of classes

https://doi.org/10.1371/journal.pone.0362597.t002

Sparse Inversion. However, optimizing the full image at every iteration incurs substantial computational overhead, limiting scalability. To improve efficiency, sparse inversion approaches reconstruct only salient regions, enabling the generation of high-quality synthetic samples within the same budget. In particular, SMI [8] employs importance-based patch selection for Vision Transformers (ViT), in which less informative tokens are masked during optimization. Nevertheless, this approach is largely restricted to ViT architectures and ignores spatial coherence, frequently fragmenting object structures and degrading structural fidelity. In contrast, our proposed PAINT generalizes sparse inversion to both CNN and ViT architectures while preserving structural coherence via saliency-to-anchor alignment with peripheral sparsification.

Data-free applications

Data-Free Quantization. Given a pre-trained full-precision model and target bit B, Data-Free Quantization (DFQ) aims to obtain a Bbit quantized model without access to real data [1,15,19]. Quantization typically requires calibration data to estimate activation ranges and mitigate performance degradation. In the absence of real images, DFQ leverages synthetic datasets as a surrogate for calibration datasets. Following prior DFQ studies [2,20], we employ uniform affine quantization to map real-valued parameters $\mathbf{W}$ to discrete representations $\mathbf{W}_q$ via scaling and rounding, as follows.

$$\mathbf{W}_q = \left\lfloor \frac{\mathbf{W}}{s} + z + \frac{1}{2} \right\rfloor,$$

where $s = (\beta - \alpha)/(2^B - 1)$ is the scaling factor and $z = -2^{B-1} - \alpha/s$ is the zero-point. We set α and β to the minimum and maximum values of $\mathbf{W}$.

Data-Free Knowledge Transfer. Given a pre-trained teacher model $\mathcal{M}_\theta$ with parameters θ and a student model $\mathcal{M}_{\theta_S}$ with parameters θ_S, Data-Free Knowledge Transfer (DFKT) aims to train $\mathcal{M}_{\theta_S}$ without access to real data [21,22]. Instead, DFKT employs a synthetic dataset $\mathbb{D}_{\text{syn}} = \{(\mathbf{X}_i, y_i)\}_{i=1}^N$ as a surrogate for the training set, and optimizes $\mathcal{M}_{\theta_S}$ to match the teacher's predictive distribution on $\mathbb{D}_{\text{syn}}$. We align the output distributions of the $\mathcal{M}_\theta$ and $\mathcal{M}_{\theta_S}$ as follows.

$$\min_{\theta_S} \frac{1}{|\mathbb{D}_{\text{syn}}|} \sum_{\mathbf{X}_i \in \mathbb{D}_{\text{syn}}} \text{KL}\left(\mathcal{M}_\theta(\mathbf{X}_i) \,\|\, \mathcal{M}_{\theta_S}(\mathbf{X}_i)\right),$$

where $\text{KL}(\cdot \,\|\, \cdot)$ denotes the Kullback–Leibler divergence.

Proposed method

We propose PAINT (**P**eripheral sparsification and **A**nchor-aligned **IN**version **T**echnique), an accurate, efficient, and sparse Model Inversion (MI) framework for synthesizing high-fidelity images while preserving object structures. Existing MI methods suffer from the following three technical challenges.

C1. Noise in synthetic images. Existing inversion methods typically rely on local smoothing to suppress noise in synthetic images. However, they still produce excessively high-frequency components compared to real images. How can we effectively suppress noise in synthetic images during MI?

C2. Object fragmentation in synthetic images. Recent methods improve inversion efficiency through patch-wise reconstruction. However, this strategy, relying solely on importance scores, often fragments object structures in synthetic images. How can we preserve object structures while enabling efficient MI?

C3. Limited coverage of predictive uncertainty. Existing inversion methods generate synthetic images with overly confident predictions. As a result, the confidence distributions of synthetic images differ from those of real images. How can we generate synthetic data that better capture predictive uncertainty?

We propose the following three ideas to deal with the challenges:

I1. Low-pass filtering. We directly mitigate excessive high-frequency noise in synthetic images through low-pass filtering during inversion.

I2. Saliency-to-anchor alignment and peripheral sparsification. We address object fragmentation in synthetic images while enabling sparse inversion through saliency-to-anchor alignment and peripheral sparsification.

I3. Soft targets. We address the limited coverage of predictive uncertainty by using soft targets, which promote a broader confidence distribution.

Given a pre-trained model $\mathcal{M}_\theta$ with parameters θ and initialized synthetic inputs $\mathbf{X}$, PAINT iteratively updates $\mathbf{X}$ to obtain high-fidelity samples. To improve fidelity, PAINT periodically adopts a low-pass filter during inversion to suppress high-frequency noise. PAINT encourages saliency-to-anchor alignment and progressively sparsifies peripheral areas according to a pre-defined scheduler, preserving object structures during sparse inversion. PAINT leverages soft targets instead of hard targets to broaden the predictive uncertainty. Fig 3 illustrates the overall framework of PAINT. Across iterations, PAINT progressively transitions from dense to sparse updates. At the initial iteration (e.g., t_1), PAINT updates the entire image. During scheduled intermediate iterations (e.g., t_i), PAINT applies low-pass filtering and peripheral sparsification. Specifically, a distance-based mask progressively removes peripheral regions, while saliency-to-anchor alignment encourages salient regions to concentrate around the anchor. In later iterations (e.g., t_j), Paint updates only the unmasked region while preserving object structures. Combining these ideas, we reformulate the model inversion loss $\mathcal{L}_{\text{MI}}$ in Eq (1) as $\mathcal{L}_{\text{Paint}}$ in Eq (2).

$$\mathcal{L}_{\text{PAINT}} = \mathcal{L}_{\text{cls}}(\mathcal{M}_\theta(\tilde{\mathbf{X}}), y^{\text{soft}}) + \lambda_{\text{img}}\mathcal{L}_{\text{img}}(\tilde{\mathbf{X}}) + \lambda_{\text{anchor}}\mathcal{L}_{\text{anchor}}(\tilde{\mathbf{X}}), \tag{2}$$

where $\mathcal{L}_{\text{cls}}$, $\mathcal{L}_{\text{img}}$, and $\mathcal{L}_{\text{anchor}}$ denote classification, image regularization, and saliency-to-anchor alignment losses, respectively. λ_{img} and λ_{anchor} are balancing hyperparameters for $\mathcal{L}_{\text{img}}$ and $\mathcal{L}_{\text{anchor}}$, respectively. Note that $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{img}}$ are identical to those in Eq (1). We obtain the filtered sample $\mathbf{X}^F$ by periodically applying a low-pass filter during inversion to

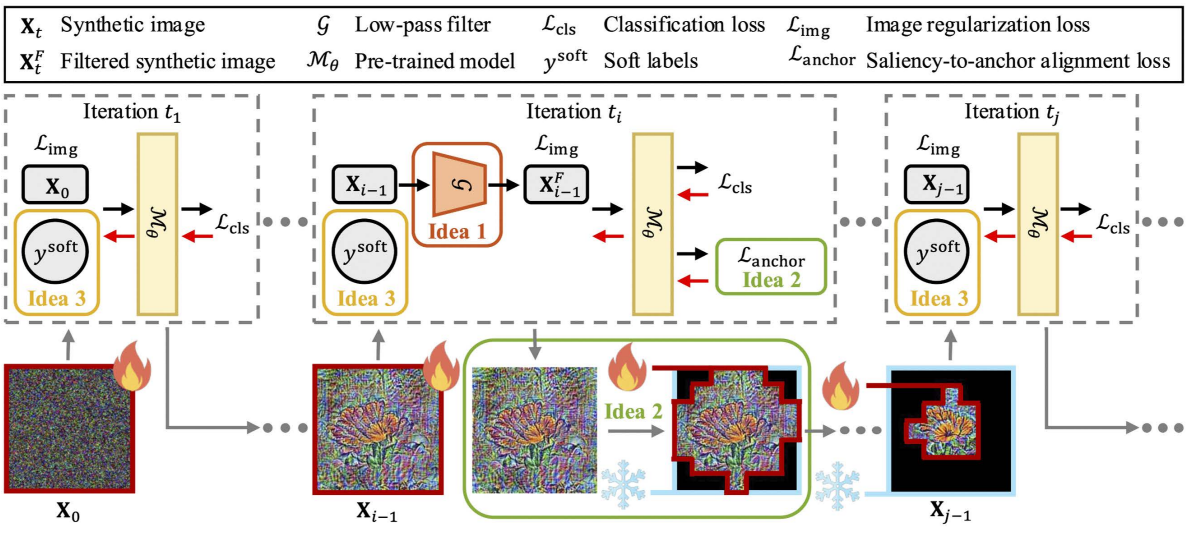

Fig 3. Overall architecture of Paint. Our main ideas are 1) low-pass filter, 2) saliency-to-anchor alignment and peripheral sparsification, and 3) soft targets.

https://doi.org/10.1371/journal.pone.0362597.g003

suppress high-frequency noise in the synthetic image. We optimize $\mathcal{L}_{\text{anchor}}$ to encourage structural coherence. We exploit soft targets y^{soft} instead of hard labels to mitigate overconfident predictions and better capture predictive uncertainty. We employ an interval t_S to reduce the computational overhead of low-pass filtering and saliency-to-anchor alignment. Specifically, $\tilde{\mathbf{X}}$ is set to the filtered input $\mathbf{X}^F$ every t_S iterations and defaults to $\mathbf{X}$ otherwise. At the same interval, $\mathcal{L}_{\text{anchor}}$ is evaluated only before the final scheduled sparsification step and is set to zero otherwise. We describe the inversion procedure of PAINT in Algorithm 1.

Algorithm 1 Overall process of PAINT.

Input: pre-trained model $\mathcal{M}_\theta$, number T of iterations, interval t_S, sparsification schedule $[\mathbf{p}^{\text{iter}}, \mathbf{p}^{\text{ratio}}]$, anchor coordinate (a_x, a_y), and hyperparameters λ_{img}, λ_{anchor}, ρ_c, and τ
Output: Synthetic sample $(\mathbf{X}_T, y)$

Initialize synthetic image $\mathbf{X}_0$ with Gaussian noise.
Randomly sample a label y.
Soften y into y^{soft} with target confidence τ. ▷ *Idea 3: Soft targets.*
Initialize the binary update mask $\mathbf{M} \leftarrow \mathbf{1}$.
for $t = 0$ to $T-1$ **do**
  $\mathcal{L}_{\text{anchor}} \leftarrow 0$.
  if $(t+1) \,\%\, t_S = 0$ **then**
    Obtain $\mathbf{X}_t^F$ by applying a low-pass filter with cutoff ratio ρ_c to $\mathbf{X}_t$. ▷ *Idea 1: Low-pass filtering.*
    $\tilde{\mathbf{X}}_t \leftarrow \mathbf{X}_t^F$.
    if $t+1 < \max(\mathbf{p}^{\text{iter}})$ **then**
      Compute saliency-to-anchor alignment loss $\mathcal{L}_{\text{anchor}}$ from $\tilde{\mathbf{X}}_t$ with anchor (a_x, a_y).
      ▷ *Idea 2-1: Saliency-to-anchor alignment.*
      end if
    else
    $\tilde{\mathbf{X}}_t \leftarrow \mathbf{X}_t$.
  end if
  if $(t+1) \in \mathbf{p}^{\text{iter}}$ **then**
    Update $\mathbf{M}$ by masking out regions far from the anchor (a_x, a_y) according to $\mathbf{p}^{\text{ratio}}$.
    ▷ *Idea 2-2: Peripheral sparsification.*
  end if
  Compute $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{img}}$ from $\tilde{\mathbf{X}}_t$ according to the architecture-specific application of $\mathbf{M}$.
  Total loss $\mathcal{L} \leftarrow \mathcal{L}_{\text{cls}} + \lambda_{\text{img}}\mathcal{L}_{\text{img}} + \lambda_{\text{anchor}}\mathcal{L}_{\text{anchor}}$.
  Update $\mathbf{X}_t$ to $\mathbf{X}_{t+1}$ by minimizing $\mathcal{L}$ only on the active region specified by $\mathbf{M}$.
end for
return $(\mathbf{X}_T, y)$

Problem definition

Given a pre-trained model, MI aims to generate a synthetic dataset similar to the original training one. The similarity between datasets is defined on a per-class basis. Specifically, each synthetic sample is optimized so that the pre-trained model confidently predicts it as the intended target class, ensuring consistency in semantic label space. We provide a formal definition as in Problem 1.

Problem 1 (Model Inversion). *Given a pre-trained model $\mathcal{M}_\theta$ with parameters θ, generate a synthetic dataset $\mathbb{D}_{\text{syn}} = \{(\mathbf{X}_i, y_i)\}_{i=1}^N$ such that each synthesized input $\mathbf{X}_i \in \mathbb{R}^{H\times W\times C}$ induces a prediction consistent with the target label y_i (i.e., $y_i = \mathcal{M}_\theta(\mathbf{X}_i)$).*

Low-pass filtering

Observation. We present two empirical observations that reveal the limitations of existing MI methods and the necessity of explicit noise suppression. First, we examine the frequency discrepancy between real and synthetic images by

sampling one image per class from ImageNet and generating synthetic images using DeepInversion [6], yielding 1,000 images per set. Fig 4(a) shows that synthetic images exhibit a significantly higher High-Frequency Ratio (HFR) than real images. Specifically, for each 224×224 RGB image $\mathbf{X}_i$, HFR is defined as $\mathrm{HFR}(\mathbf{X}_i) = \frac{\sum_{r=r_{\mathrm{hf}}}^{R} \bar{P}_{\mathbf{X}_i}(r)}{\sum_{r=0}^{R} \bar{P}_{\mathbf{X}_i}(r)+\epsilon}$, where $\bar{P}_{\mathbf{X}_i}(r)$ denotes the mean Fourier power at radial frequency r. We use $R = 112$ and $r_{\mathrm{hf}} = 80$, corresponding to a normalized frequency threshold of $r_{\mathrm{hf}}/R \approx 0.714$. The mean HFR values of ImageNet and synthetic images are 0.0007 and 0.0044, respectively. This indicates that synthetic images contain approximately $6.3\times$ higher HFR than ImageNet images. To quantitatively validate this visual observation, we conduct a one-sided Mann-Whitney U test with the alternative hypothesis that synthetic images have higher HFR values than ImageNet images. Since $p = 3.089 \times 10^{-204} < 0.05$, we reject the null hypothesis that synthetic images do not have higher HFR values than ImageNet images do. This result confirms that synthetic images contain significantly stronger high-frequency components than real images. Next, we examine the relationship between high-frequency noise and downstream task performance. Specifically, we measure the HFR of synthetic images generated by DeepInversion and the W4A8 quantized accuracy of a DeiT-Base model, repeating 40 times. Fig 4(b) shows a weak negative Pearson correlation between HFR and quantized model accuracy, with Pearson Correlation Coefficient (PCC) of –0.3691. A two-sided test yields $p = 0.0191$, and the 95% confidence interval for PCC is [–0.6105, –0.0651]. This result suggests that higher HFR tends to be associated with lower quantized accuracy. Overall, these observations indicate that suppressing excessive high-frequency noise in synthetic images is crucial.

Solution. Motivated by the above observations, we explicitly suppress high-frequency noise during inversion. Existing inversion methods typically rely on implicit or local smoothing, which is insufficient for controlling high-frequency components. To address this limitation, we integrate a noise reduction filter into the inversion process. Inspired by SynQ [1], we adopt a Gaussian low-pass filter for its simplicity and explicit frequency-domain control, which allows for effective noise suppression while preserving low-frequency structures critical to object perception. However, in contrast to SynQ which applies the filter as a post-processing step to synthesized images, PAINT applies the filter to intermediate synthetic inputs at every t_S iterations during optimization to reduce computational overhead. Given a synthetic image $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ and cutoff radius R_c, we compute the filtered sample $\mathbf{X}^F$ by applying the filter $\mathcal{G}$ independently to each channel, as shown in Eq (3).

$$\mathbf{X}^F = \mathcal{F}^{-1}\left(\mathcal{G} \odot \mathcal{F}(\mathbf{X})\right), \quad \mathcal{G}_{uv} = \exp\left(-\frac{D_f(u,v)^2}{2R_c^2}\right), \tag{3}$$

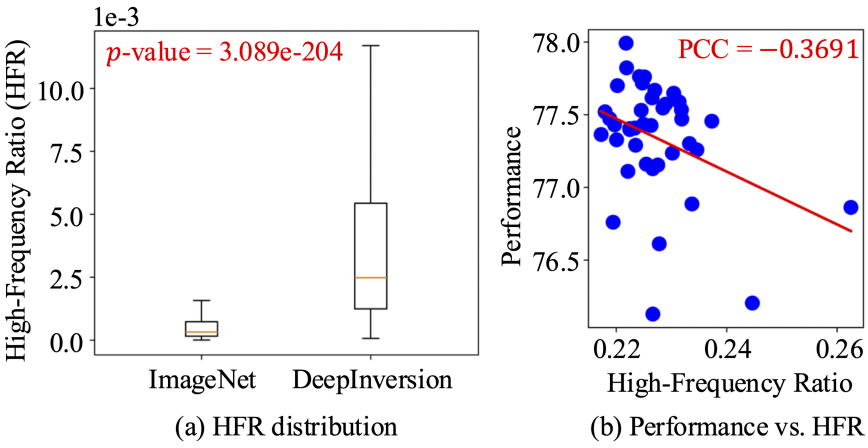

Fig 4. Observation results for high-frequency noise in synthetic images. (a) Synthetic images generated by DeepInversion contain substantially higher high-frequency components than real images from ImageNet do. (b) High-frequency ratio exhibits a weak negative Pearson correlation with quantization performance.

https://doi.org/10.1371/journal.pone.0362597.g004

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote the Fourier and inverse Fourier transforms, respectively, $\odot$ represents element-wise multiplication, and $D_f(u, v)$ denotes the Euclidean distance between the coordinate (u,v) and the center $(\frac{W}{2}, \frac{H}{2})$ in the frequency domain. Based on this frequency distance, we parameterize the cutoff radius R_c by a cutoff ratio ρ_c to make the filtering strength comparable across different image resolutions. Let $D_{\max} = \max_{u,v} D_f(u, v)$ be the maximum distance in the frequency grid; the actual cutoff radius is set to $R_c = \rho_c D_{\max}$. Thus, the same ρ_c preserves the same relative frequency range across different image resolutions.

Saliency-to-anchor alignment and peripheral sparsification

Observation. We present empirical observations that reveal both the feasibility and limitations of sparse MI. First, we analyze the spatial distribution of saliency maps by sampling one image per class from ImageNet and generating synthetic images. Fig 1(a) shows that discriminative regions occupy only a small portion of the image, indicating that reconstructing the entire image space is inherently inefficient. Motivated by this observation, sparse MI methods [8] perform patch-wise early stopping based solely on importance scores. However, Fig 2(b) shows that this strategy often fragments object structures in synthesized images. Although such sparsification improves efficiency, it degrades structural fidelity, which negatively affects downstream performance. These observations suggest that efficient inversion is feasible, but naive importance-based sparsification leads to object fragmentation.

Solution. Our goal is to perform sparse inversion without fragmenting spatially coherent object structures. To this end, we propose a two-stage sparsification mechanism: (1) saliency-to-anchor alignment and (2) peripheral sparsification. First, we explicitly encourage salient regions to concentrate around a predefined anchor coordinate (a_x, a_y) by minimizing saliency-to-anchor alignment loss $\mathcal{L}_{\text{anchor}}$ defined in Eq (4). This loss pulls discriminative regions toward the anchor, preserving structural coherence during inversion.

$$\mathcal{L}_{\text{anchor}} = \sum_{i=1}^{W} \sum_{j=1}^{H} \mathbf{S}(i, j; \mathbf{X}, \mathcal{M}_\theta) \cdot \frac{(a_x - i)^2 + (a_y - j)^2}{W^2 + H^2}, \tag{4}$$

where $\mathbf{S}(i, j; \mathbf{X}, \mathcal{M}_\theta)$ represents the saliency weight at location (i,j). We compute the saliency weight $\mathbf{S}$ from the input gradient of the target-class logit. For each spatial location, we aggregate the absolute input gradients by taking the maximum value across channels, and normalize the resulting saliency map over all spatial locations, i.e., $\sum_{i=1}^{W} \sum_{j=1}^{H} \mathbf{S}(i, j; \mathbf{X}, \mathcal{M}_\theta) = 1$. This normalization makes $\mathcal{L}_{\text{anchor}}$ depend on the spatial distribution of target-class saliency rather than its absolute gradient scale. The saliency map is not detached from the computational graph. Consequently, optimizing $\mathcal{L}_{\text{anchor}}$ with respect to $\mathbf{X}$ differentiates through the input gradient of the target-class logit and involves higher-order derivatives. We adopt $\mathcal{L}_{\text{anchor}}$ periodically rather than at every iteration to reduce the overhead of higher-order differentiation. Specifically, we evaluate $\mathcal{L}_{\text{anchor}}$ every t_S iterations only before the final scheduled sparsification step, where t_S is defined as the smallest interval between consecutive sparsification steps in $\mathbf{p}^{\text{iter}}$, including the interval from iteration 0 to the first sparsification step; e.g., $\mathbf{p}^{\text{iter}} = [100, 150]$ gives $t_S = 50$.

Second, we progressively sparsify peripheral regions by stopping their updates according to their distance from the anchor. Given sparsification schedule $[\mathbf{p}^{\text{iter}}, \mathbf{p}^{\text{ratio}}]$, we define a distance-based mask $\mathbf{M}$ initialized as an all-one mask, and update $\mathbf{M}$ only at the iterations specified by $\mathbf{p}^{\text{iter}}$. At each such iteration, the corresponding sparsification ratio in $\mathbf{p}^{\text{ratio}}$ is used to mask out peripheral regions with respect to the anchor. Formally, the mask is defined as in Eq (5).

$$\mathbf{M}(i, j) = \begin{cases} 1, & \text{if } D_s(i, j) \leq d_t \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $D_s(i, j)$ denotes the Euclidean distance between the coordinate (i,j) and the anchor (a_x, a_y). d_t is a distance threshold at iteration t determined by the corresponding sparsification ratio in $\mathbf{p}^{\text{ratio}}$; a specified fraction of the currently active spatial locations farthest from the anchor is additionally masked out at each update step, progressively shrinking the

active region over the scheduled iterations. Since $\mathbf{p}^{\text{ratio}}$ is applied to the currently active spatial locations at each scheduled step, the final sparsity ratio after K steps is computed as $1-\prod_{k=1}^{K}(1-p_k^{\text{ratio}})$. The mask is realized differently depending on the architecture. For CNNs, it is applied as a persistent full-resolution mask to the synthetic input used for the model forward and image regularization throughout subsequent inversion iterations. For ViTs, a patch-level mask identifies active patch tokens, which are gathered after patch embedding and passed with the retained CLS token to subsequent Transformer blocks. The patch-level mask is expanded over the corresponding pixel regions to form a full-resolution mask for image regularization and persistent zeroing.

By combining saliency-to-anchor alignment and peripheral sparsification, PAINT enables structured sparsification while preserving object structure. $\mathcal{L}_{\text{anchor}}$ assigns larger penalties to salient pixels located farther from the anchor, encouraging discriminative regions to concentrate near the anchor before peripheral regions are progressively masked out by $\mathbf{M}$. This coupling of saliency-to-anchor alignment and peripheral sparsification mitigates object fragmentation, since regions far from the anchor are more likely to be removed during peripheral sparsification.

Soft targets

Observation. We present an observation that synthetic images generated by existing MI methods exhibit overly confident predictions compared to real training data. To verify this, we sample one image per class from ImageNet and generate synthetic images using DeepInversion [6]. Fig 5 compares the confidence score distributions, where the confidence score is defined as the predicted probability of the target class. Real images show diverse confidence levels, while synthetic images concentrate heavily on near-one confidence values. This discrepancy indicates that existing MI methods tend to generate samples that strongly maximize classification confidence. Such overconfident samples reduce diversity in predictive behavior and limit downstream performance.

Solution. To better capture predictive uncertainty, we replace hard target labels with soft targets during inversion. Instead of the conventional hard target y, we employ a softened label distribution defined in Eq (6):

$$y_k^{\text{soft}} = \begin{cases} \tau, & \text{if } k = y, \\ \frac{1-\tau}{K-1}, & \text{otherwise,} \end{cases} \tag{6}$$

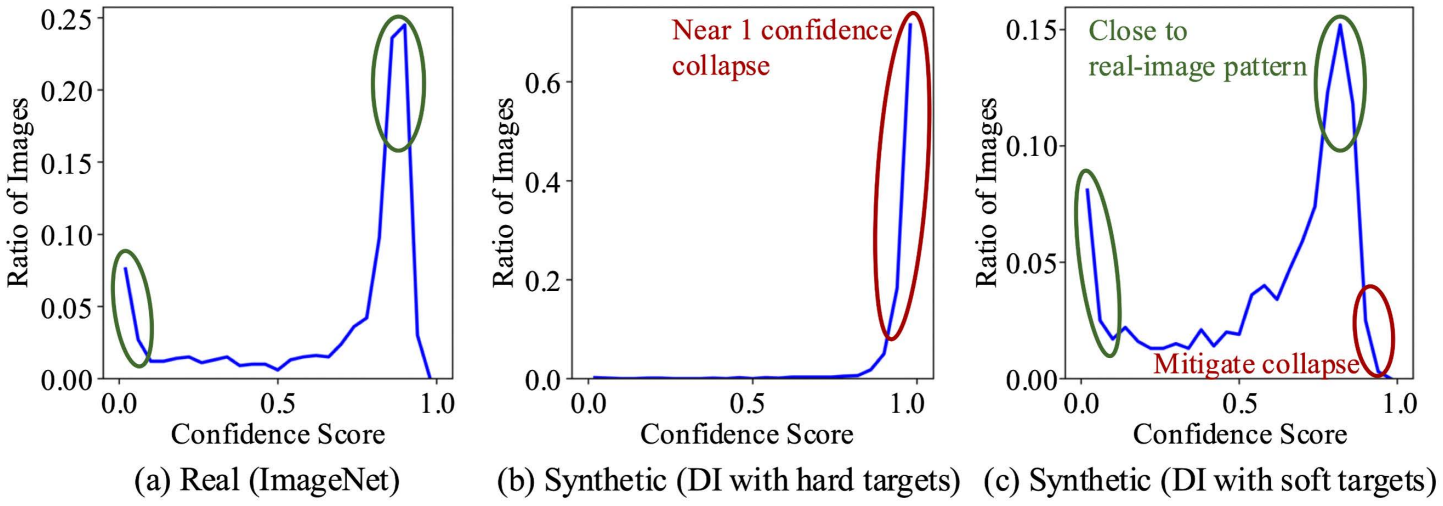

Fig 5. Comparison of confidence score distributions. (a) Real images show a broad confidence distribution, (b) DeepInversion with hard targets is heavily concentrated near 1, and (c) soft targets alleviate this overconfidence.

https://doi.org/10.1371/journal.pone.0362597.g005

where K denotes the number of classes and $\tau \in (0, 1)$ controls the target confidence.

We then optimize the classification loss using this soft target distribution. By preventing the model from producing overly confident outputs, soft targets encourage broader confidence distributions and generate more informative synthetic samples. As shown in Fig 5(c), the resulting confidence distribution better aligns with that of real data, improving downstream effectiveness.

Computational complexity and memory analysis

We distinguish the overhead introduced by the additional operations of PAINT from the architecture-dependent computational effect of peripheral sparsification.

Computational overhead. Low-pass filtering and saliency-to-anchor alignment employ the same interval t_S, but their numbers of executions differ because $\mathcal{L}_{\text{anchor}}$ is evaluated only before the final scheduled sparsification step. Let N_{LPF} and N_{anchor} denote their respective numbers of executions over T inversion iterations. Under the default setting with $T = 4000$, $\mathbf{p}^{\text{iter}} = [50, 100, 200, 300]$, and $t_S = 50$, we have $N_{\text{LPF}} = 80$ and $N_{\text{anchor}} = 5$. Let $\mathbf{X} \in \mathbb{R}^{B\times C\times H\times W}$ denote a batch of synthetic images, with batch size B, channel C, height H, and width W. Let $\mathcal{C}_{\text{HO}}(\mathcal{M}_\theta)$ denote the cost of computing the input gradient while retaining its computational graph and subsequently backpropagating through it when optimizing $\mathcal{L}_{\text{anchor}}$.

Lemma 1. *The average computational overhead introduced by low-pass filtering and saliency-to-anchor alignment in* PAINT *is* $O\left(\frac{N_{\text{LPF}}BCHW\log(HW)+N_{\text{anchor}}\left(\mathcal{C}_{\text{HO}}(\mathcal{M}_\theta)+BCHW\right)}{T}\right)$.

Proof. First, low-pass filtering is implemented via Fast Fourier Transform (FFT) and inverse FFT over the spatial dimensions $H \times W$ of each image. For each channel, FFT-based filtering over the $H \times W$ grid requires $O(HW\log(HW))$ operations. Since this operation is applied to B images and all C channels, the overall computational complexity of low-pass filtering is $O(BCHW\log(HW))$.

Second, the saliency map is computed from the gradient of the target-class logit with respect to the input image. Since the saliency map is not detached, optimizing $\mathcal{L}_{\text{anchor}}$ backpropagates through this input gradient and incurs the higher-order computation $\mathcal{C}_{\text{HO}}(\mathcal{M}_\theta)$. After obtaining the input gradient, PAINT performs channel-wise aggregation and spatial normalization, which requires $O(BCHW)$ operations. Therefore, the computational cost of saliency-to-anchor alignment is given by $O(\mathcal{C}_{\text{HO}}(\mathcal{M}_\theta) + BCHW)$.

Over T inversion iterations, low-pass filtering and saliency-to-anchor alignment are performed N_{LPF} and N_{anchor} times, respectively. Their average additional cost per iteration is therefore $O\left(\frac{N_{\text{LPF}}BCHW\log(HW)+N_{\text{anchor}}\left(\mathcal{C}_{\text{HO}}(\mathcal{M}_\theta)+BCHW\right)}{T}\right)$. □

Runtime and memory overhead of anchor alignment. Higher-order differentiation requires temporarily retaining the input-gradient graph during steps where $\mathcal{L}_{\text{anchor}}$ is applied. When all patch tokens are active, a single $\mathcal{L}_{\text{anchor}}$-active step takes $3.19\times$ and $4.28\times$ as long as the same step without $\mathcal{L}_{\text{anchor}}$ for DeiT-Tiny and DeiT-Base, respectively. It also requires $2.83\times$ and $2.77\times$ as much peak allocated GPU memory, respectively. These values are conservative estimates because they are measured with the maximum number of active tokens. Under our sparsification schedule, this maximum token count is retained only until the first scheduled sparsification at iteration 50, corresponding to 1.25% of the 4,000 inversion iterations. Subsequent steps use shorter token sequences, thereby reducing the sequence-dependent computational and memory costs of higher-order differentiation.

To evaluate the overall runtime overhead of $\mathcal{L}_{\text{anchor}}$, we compare PAINT with the same configuration but with only $\mathcal{L}_{\text{anchor}}$ disabled. Both configurations share the same low-pass filtering and peripheral sparsification settings. Over five seeds and 4,000 inversion iterations, DeiT-Tiny shows a mean runtime overhead of $5.4 \pm 4.7\%$, whereas DeiT-Base shows a mean

runtime overhead of $2.7 \pm 1.6\%$. Overall, the runtime overhead remains small because $\mathcal{L}_{\text{anchor}}$ activates the higher-order differentiation path only five times, corresponding to 0.125% of the total inversion iterations.

Peripheral sparsification for ViTs. For ViTs, peripheral sparsification removes inactive patch tokens from the token sequence while retaining the CLS token. After the k-th sparsification step, the number of active patch tokens is $N_k = \left\lfloor N_0 \prod_{i=1}^{k} \left(1 - \mathbf{p}_i^{\text{ratio}}\right) \right\rfloor$, where N_0 is the initial number of patch tokens. The corresponding sequence length is $L_k = N_k + 1$, including the CLS token. Since the shortened sequence is passed to subsequent Transformer blocks, their sequence-dependent complexity decreases from $O(L_0^2 d + L_0 d^2)$ to $O(L_k^2 d + L_k d^2)$, where d denotes the token embedding dimension. This reduction applies only to self-attention, token-wise projection, and MLP operations in subsequent Transformer blocks; other operations do not benefit from the reduced token sequence length.

Peripheral sparsification for CNNs. For CNNs, peripheral sparsification applies a persistent spatial mask to the synthetic input, maintaining zero-valued inactive regions through subsequent inversion iterations. Applying this mask requires $O(BCHW)$ element-wise operations. However, the masked input retains its original dense tensor dimensions and is processed using standard dense convolution. Thus, for a convolutional layer with spatial dimension $H_l \times W_l$, kernel size $K_l \times K_l$, and input and output channels C_l^{in} and C_l^{out}, respectively, its computational complexity remains $O(H_l W_l K_l^2 C_l^{\text{in}} C_l^{\text{out}})$. Consequently, both forward and backward convolutional computations retain the same asymptotic dense complexity, despite the zero-valued inactive regions. Therefore, peripheral sparsification does not reduce the theoretical convolutional FLOPs of CNNs in the current implementation.

Experiments

We perform experiments to answer the following questions:

Q1. Accuracy in Data-Free Quantization. How accurate is the quantized model with PAINT compared to existing MI methods for CNNs and ViTs?

Q2. Accuracy in Data-Free Knowledge Transfer. How accurate is the student model trained with PAINT compared to existing MI methods for CNNs and ViTs?

Q3. Accuracy-Efficiency Trade-Off. How does the sparsification schedule in PAINT affect the trade-off between downstream accuracy and inversion efficiency?

Q4. Qualitative Analysis. How does the visual quality of images synthesized by PAINT compare to that of existing MI methods?

Q5. Analysis on Anchor Coordinates. How does the choice of anchor coordinate in PAINT affect downstream task performance?

Q6. Ablation Study. How does each component of PAINT contribute to downstream performance?

Q7. Hyperparameter Analysis. How robust are the performance gains by PAINT in hyperparameters λ_{anchor}, ρ_c, and τ?

Experimental setup

We briefly describe the experimental setup.

Metrics. We evaluate PAINT in two axes: inversion efficiency and synthetic data quality. Inversion efficiency is measured by throughput, which is defined as the number of iterations per second during inversion. For fair comparison, all methods are evaluated under the same numbers of inversion iterations and generated synthetic samples. Under this controlled comparison protocol, throughput directly reflects the computational efficiency of each inversion method. Synthetic data quality is assessed by top-1 classification accuracy on two downstream tasks (DFQ and DFKT) across three target datasets: CIFAR-10, CIFAR-100 [23], and ImageNet [24].

Implementation details. We implement PAINT in Python with PyTorch and follow the experimental setup of DeepInversion [6] and SMI [8] for fair comparison. For model inversion, we generate 32 synthetic samples over 4,000 inversion iterations using the Adam optimizer [25] with learning rate of 0.25. To ensure a fair comparison in sparse inversion, both PAINT and SMI adopt the same sparsification schedule from SMI. Specifically, we set $\mathbf{p}^{\text{iter}} = [50, 100, 200, 300]$ and $\mathbf{p}^{\text{ratio}} = [0.3, 0.3, 0.3, 0.3]$, resulting in a final sparsity of approximately 76%. For ViT models with $14 \times 14 = 196$ spatial patches, this schedule retains about $196 \times 0.7^4 \approx 47$ active patches, resulting in a final sparsity of approximately 76%. Accordingly, t_S is set to 50 for low-pass filtering and saliency-to-anchor alignment, as it is the smallest interval between consecutive sparsification steps. Synthesized images are generated as 224×224 for ViTs, and as the dataset's native resolution for CNNs (32×32 for CIFAR-10/100, and 224×224 for ImageNet). We search λ_{anchor}, ρ_c, and τ within the ranges [0.2, 0.4, 0.6, 0.8, 1.0], [0.6, 0.7, 0.8, 0.9], and [0.70, 0.75, 0.80, 0.85, 0.90, 0.95], respectively. All ViT experiments are conducted on a single NVIDIA L40 GPU, whereas all CNN experiments are conducted on a single NVIDIA RTX 3080 GPU.

Accuracy in data-free quantization (Q1)

We evaluate the quality of the synthetic dataset generated by PAINT via Data-Free Quantization (DFQ) on both ViT and CNN models.

DFQ Performance on ViTs. We evaluate the quality of the synthetic dataset generated by PAINT against existing MI methods on ImageNet dataset. Table 3 shows the W4/A8 and W8/A8 DFQ accuracy of two ViT models, DeiT-Tiny and DeiT-Base, where accuracy is reported as mean ± standard deviation over 10 runs with different random seeds. We have three findings from the results. First, PAINT achieves the highest inversion throughput among all competitors. This shows that PAINT improves inversion efficiency despite incorporating low-pass filtering and saliency-to-anchor alignment. Second, PAINT improves accuracy by up to 1.03%p over the sparse MI method SMI [8] at the same sparsity level. This suggests that PAINT improves the fidelity of sparse inversion by producing more structurally coherent and less noisy synthetic images than SMI. Third, PAINT achieves competitive or higher quantization accuracy compared with DeepInversion while requiring substantially fewer active spatial updates. In particular, PAINT improves accuracy by up to 2.24%p over DeepInversion. These results demonstrate that PAINT provides a favorable accuracy-efficiency trade-off for ViT-based DFQ.

DFQ Performance on CNNs. We demonstrate the quality of the synthetic dataset generated by PAINT in enhancing DFQ performance for CNNs. Table 4 shows the W3/A3 and W4/A4 DFQ accuracy of two CNN models, ResNet-18 (R-18) [26] and MobileNetV2 (MV2) [27] pre-trained on ImageNet dataset. The results yield three observations. First, PAINT is applicable to both CNN and ViT architectures, whereas SMI is restricted to ViT models due to its token-level sparsification design. Second, the effectiveness of PAINT is more pronounced for larger models; specifically, PAINT performs more

Table 3. Data-free quantization accuracy [%] of DeiT-Tiny and DeiT-Base on ImageNet. W*B*/A*B* denotes that both weights and activations are quantized *B* bits. Full Precision (FP) indicates W32/A32. Accuracy is reported as mean ± standard deviation over 10 runs with different random seeds. Note that PAINT achieves the highest inversion throughput and higher accuracy than SMI under the same sparsity level.

Model	Method	Model Inversion (Upstream)		Quantization (Downstream)			
		Sparsity	Throughput (its/s)	Bit Width	Top-1	Bit Width	Top-1
DeiT-Tiny	Original	–	–	FP	71.43	FP	71.43
	DeepInversion [6]	0	17.72	W4/A8	64.43 ± 0.08	W8/A8	70.57 ± 0.11
	SMI [8]	76%	32.04 (× 1.81)	W4/A8	64.01 ± 0.09	W8/A8	70.29 ± 0.11
	Paint (proposed)	76%	45.89 (× 2.59)	W4/A8	64.47 ± 0.15	W8/A8	70.47 ± 0.30
DeiT-Base	Original	–	–	FP	81.49	FP	81.49
	DeepInversion [6]	0	3.57	W4/A8	75.11 ± 0.74	W8/A8	77.75 ± 0.57
	SMI [8]	76%	13.48 (× 3.78)	W4/A8	76.32 ± 0.48	W8/A8	78.65 ± 0.35
	Paint (proposed)	76%	18.17 (× 5.09)	W4/A8	77.35 ± 0.17	W8/A8	79.53 ± 0.15

https://doi.org/10.1371/journal.pone.0362597.t003

Table 4. Data-free quantization accuracy [%] of R-18 (ResNet-18) and MV2 (MobileNetV2) on ImageNet dataset. W*B*/A*B* denotes that both weights and activations are quantized *B* bits. Full Precision (FP) indicates W32/A32. Note that PAINT is applicable to CNN models, whereas the token-based sparse inversion method SMI is restricted to ViTs.

Model	Method	Model Inversion	Quantization			
		Sparsity	Bit Width	Top-1	Bit Width	Top-1
R-18	Original	–	FP	71.00	FP	71.00
	DeepInversion [6]	0	W3/A3	65.25 ± 0.18	W4/A4	69.60 ± 0.09
	SMI [8]	N/A	W3/A3	N/A	W4/A4	N/A
	Paint (proposed)	76%	W3/A3	59.26 ± 0.85	W4/A4	68.28 ± 0.12
MV2	Original	–	FP	72.63	FP	72.63
	DeepInversion [6]	0	W3/A3	15.46 ± 1.24	W4/A4	62.11 ± 0.28
	SMI [8]	N/A	W3/A3	N/A	W4/A4	N/A
	Paint (proposed)	76%	W3/A3	8.63 ± 1.58	W4/A4	57.92 ± 1.37

https://doi.org/10.1371/journal.pone.0362597.t004

favorably on ResNet-18 than on MobileNetV2. This indicates that lightweight architectures like MobileNetV2 are inherently more sensitive to synthetic images from model inversion, whereas larger models are more robust to them. Third, Paint yields lower DFQ accuracy than DeepInversion, particularly under W3A3 quantization. These results suggest that aggressive sparsification is more challenging for CNN-based DFQ than ViT-based DFQ.

Accuracy in data-free knowledge transfer (Q2)

We evaluate the quality of the synthetic dataset generated by PAINT via Data-Free Knowledge Transfer (DFKT) on both ViT and CNN models in Table 5.

DFKT Performance on ViTs. We report DFKT accuracy of ViTs trained on a synthetic dataset generated by MI methods. First, student models trained on synthetic images generated by PAINT achieve performance comparable to those trained with DeepInversion (see Table 5), while requiring substantially fewer resources for image synthesis, as reflected in their higher inversion throughput (see Table 3). For instance, PAINT outperforms DeepInversion by up to 5.33%p on DeiT-Base for CIFAR-10. Second, PAINT shows strong DFKT performance on CIFAR-10, achieving the highest accuracy among competitors on both models. On CIFAR-100, PAINT achieves slightly lower accuracy than SMI on DeiT-Tiny, while showing comparable performance on DeiT-Base. These results indicate that PAINT remains effective for DFKT, but its advantage becomes less pronounced on the more fine-grained CIFAR-100 dataset. One possible reason is that CIFAR-100 requires more diverse and fine-grained visual evidence for student training; in this setting, saliency-to-anchor alignment and peripheral sparsification may make the synthetic images more compact around the anchor, reducing peripheral or contextual cues that are useful for distinguishing visually similar classes.

DFKT Performance on CNNs. We investigate the effectiveness of the synthetic dataset generated by PAINT on CNN architectures. We have three findings from the results. First, PAINT achieves performance comparable to DeepInversion

Table 5. Data-free knowledge transfer accuracy [%] of DeiT-Tiny, DeiT-Base, VGG-11, and ResNet-34 on CIFAR-10 (C-10) and CIFAR-100 (C-100) datasets. Note that PAINT achieves competitive DFKT performance.

Method	DeiT-Tiny		DeiT-Base		VGG-11	ResNet-34
	C-10	C-100	C-10	C-100	C-10	C-10
Teacher	90.46	70.96	95.45	80.57	92.38	95.33
DeepInversion [6]	90.13	71.03	90.02	78.17	61.19	49.28
SMI [8]	90.19	70.56	95.10	77.58	N/A	N/A
Paint (proposed)	90.48	69.42	95.35	77.56	60.10	50.84

https://doi.org/10.1371/journal.pone.0362597.t005

despite performing sparse inversion through peripheral sparsification. Second, PAINT is directly applicable to CNNs, generalizing sparsification to saliency-map space (see Table 5, VGG-11/ResNet-34 columns). In contrast, SMI is inherently patch-based and is not directly applicable to CNNs; thus, we omit SMI results for CNN architectures. Third, the benefit of PAINT becomes more pronounced for teacher models with larger capacity. While DeepInversion outperforms PAINT on the relatively shallower VGG-11, PAINT surpasses DeepInversion on the deeper ResNet-34 architecture. This suggests that richer representations from larger models provide more informative signals for knowledge extraction, making them more robust to sparsified inversion.

Accuracy-efficiency trade-off (Q3)

We analyze how the sparsification schedule in PAINT affects the trade-off between downstream accuracy and inversion efficiency. Specifically, we vary the sparsification schedule $[\mathbf{p}^{\text{iter}}, \mathbf{p}^{\text{ratio}}]$ while keeping the number of inversion iterations and generated synthetic samples. Table 6 shows the W4A8 DFQ accuracy of DeiT-Tiny and the corresponding inversion throughput under each sparsification schedule on a single A100 GPU. We present two observations from the results. First, throughput consistently increases as sparsity becomes higher. This monotonic improvement indicates that peripheral sparsification reduces ViT computation by removing inactive patch tokens from subsequent Transformer blocks. Second, downstream accuracy increases up to 59% sparsity and decreases afterward. This indicates that peripheral sparsification improves synthetic data quality by removing noisy or task-irrelevant background regions, while excessive sparsification discards task-relevant object parts and degrades accuracy. Overall, these results show that the sparsification schedule directly controls the accuracy-efficiency trade-off in PAINT.

Qualitative analysis (Q4)

We qualitatively evaluate images synthesized by PAINT against existing MI methods. We visualize images inverted from CLIP ViT-B/16, as (i) CLIP provides semantically aligned, human-interpretable features via large-scale image–text contrastive pretraining, and (ii) larger-capacity models yield clearer perceptual structure in inversions. We fine-tune CLIP ViT-B/16 on CIFAR-100, MiniImageNet, Oxford 102 Flowers, and CUB-200–2011. We then generate inverted images using the fine-tuned model, as shown in Fig 6. We present three observations from the results. First, PAINT preserves salient object evidence while suppressing unnecessary background artifacts and noise. Second, DeepInversion often introduces background clutter, whereas SMI tends to produce fragmented object structures; PAINT mitigates this trade-off by achieving both cleanliness and integrity. Third, PAINT produces high-fidelity images not only on generic datasets (CIFAR-100 and MiniImageNet) but also on fine-grained, category-specific datasets (Oxford 102 Flowers for flowers and CUB-200–2011 for birds), retaining discriminative details. These results indicate that saliency-guided sparsification improves perceptual fidelity by filtering background artifacts while maintaining task-relevant visual evidence across diverse domains.

Table 6. Accuracy-efficiency trade-off under different sparsification schedules. The results show that a moderate sparsification schedule improves inversion throughput while maintaining competitive downstream task accuracy [%] on DeiT-Tiny.

Sparsification Schedule		Sparsity	Throughput (its/s)	Accuracy
$\mathbf{p}^{\text{iter}}$	$\mathbf{p}^{\text{ratio}}$			
[50, 100, 200, 300]	[0.0, 0.0, 0.0, 0.0]	0%	13.78	64.43
	[0.1, 0.1, 0.1, 0.1]	34%	14.27 (× 1.03)	64.53 (+0.10)
	[0.2, 0.2, 0.2, 0.2]	59%	15.22 (× 1.10)	64.61 (+0.18)
	[0.3, 0.3, 0.3, 0.3]	76%	15.95 (× 1.16)	64.48 (+0.05)
	[0.4, 0.4, 0.4, 0.4]	87%	16.36 (× 1.19)	63.76 (−0.67)
	[0.5, 0.5, 0.5, 0.5]	94%	16.42 (× 1.19)	63.11 (−1.32)

https://doi.org/10.1371/journal.pone.0362597.t006

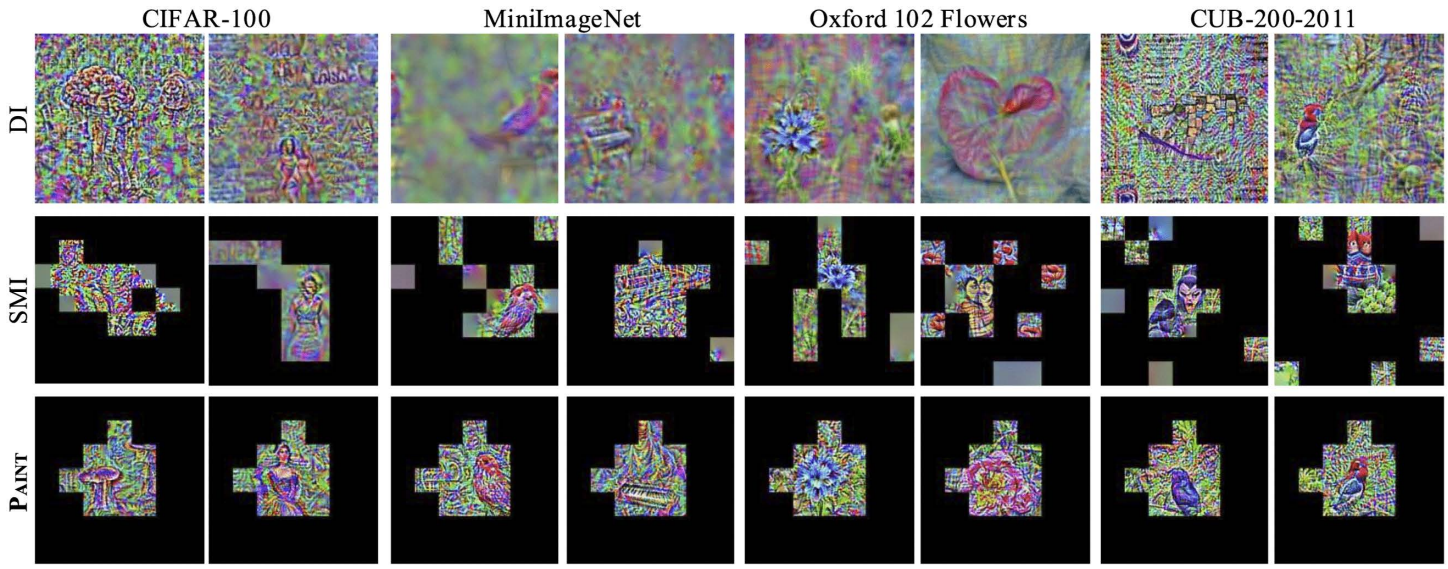

Fig 6. Comparison of synthetic images generated by various methods on multiple datasets. PAINT maintains structural integrity, yielding semantically coherent samples across domains.

https://doi.org/10.1371/journal.pone.0362597.g006

Analysis on anchor coordinates (Q5)

We analyze how the anchor coordinate in PAINT affects downstream performance. Specifically, we divide the image into a 5×5 grid and evaluate 25 anchor coordinates, ranging from the top-left corner (0,0) to the bottom-right corner (4,4). Fig 7 shows the W4A8 DFQ accuracy of DeiT-Base for each coordinate. We observe that the center anchor (2,2) yields the best performance among all configurations. This result suggests that aligning salient regions toward the image center provides a more effective sparsification configuration, likely by better preserving structurally important regions during

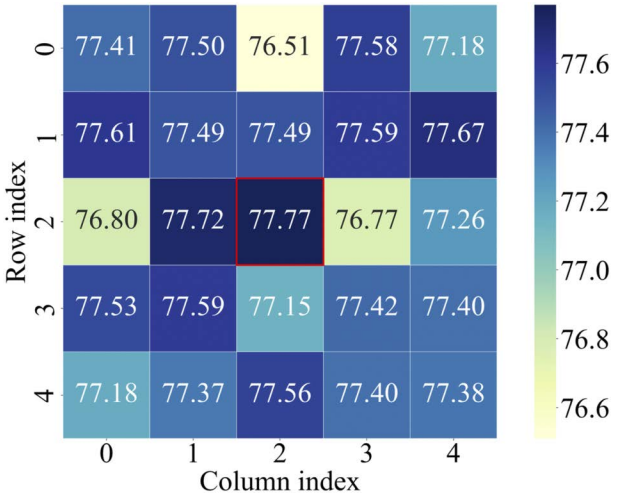

Fig 7. DFQ accuracy across different anchor coordinates on 5×5 grid. The center anchor (2,2) achieves the best performance, indicating that centering saliency-guided sparsification is the most effective configuration.

https://doi.org/10.1371/journal.pone.0362597.g007

inversion. Therefore, we adopt the center coordinate as the default anchor location for saliency-to-anchor alignment in all experiments.

Ablation study (Q6)

We perform an ablation study to evaluate the contribution of each component of PAINT, including Gaussian low-pass filtering (I1), saliency-to-anchor alignment and peripheral sparsification (I2), and soft targets (I3), to the classification accuracy of the quantized model. We summarize the W4/A8 quantization results of DeiT-Tiny on ImageNet in Table 7. Each component consistently improves performance, with I2 yielding the largest gain (2.05%p) over DeepInversion.

Hyperparameter sensitivity (Q7)

We analyze the robustness of PAINT concerning the newly introduced hyperparameters λ_{anchor}, ρ_c, and τ in Fig 8. We report the W4A8 quantization accuracy for the DeiT-Base model trained on the ImageNet dataset. We have three observations from the results. First, Fig 8(a) illustrates the effect of the balancing hyperparameter λ_{anchor} for saliency-to-anchor alignment loss. When λ_{anchor} is too small, the alignment effect becomes insufficient, limiting structural preservation. Conversely, excessively large values overly constrain the optimization process and degrade performance. These results

Table 7. Ablation study on the main ideas of PAINT. All components contribute to the improved performance.

Idea 1	Idea 2	Idea 3	Sparsity	Acc. [%]	Improvement [%p]
✓			–	75.83	+0.43
	✓		76%	77.45	+2.05
		✓	–	75.70	+0.30
✓	✓		76%	77.29	+1.89
✓		✓	–	75.99	+0.59
	✓	✓	76%	77.33	+1.93
✓	✓	✓	76%	78.00	+2.60

https://doi.org/10.1371/journal.pone.0362597.t007

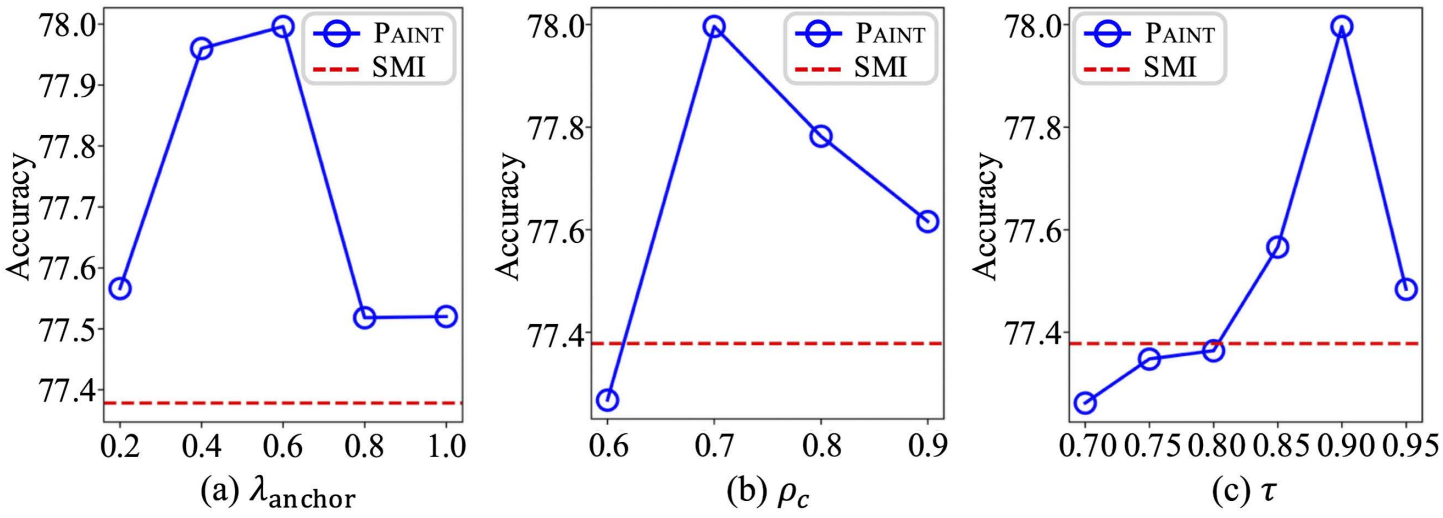

Fig 8. Hyperparameter analysis. We evaluate the effect of three key hyperparameters in PAINT: (a) balancing hyperparameter λ_{anchor}, (b) cutoff ratio ρ_c, and (c) target confidence τ. The red dashed line denotes the performance of SMI.

https://doi.org/10.1371/journal.pone.0362597.g008

indicate that a moderate alignment strength provides the best trade-off between structural guidance and optimization flexibility. Second, Fig 8(b) shows the impact of the cutoff ratio ρ_c. An appropriate choice of ρ_c is crucial for maintaining performance. Low ρ_c leads to excessive filtering that oversmooths the images and removes informative components, resulting in performance degradation. Conversely, a too large ρ_c fails to sufficiently suppress high-frequency noise, diminishing the benefit of low-pass filtering. Third, as shown in Fig 8(c), an appropriate choice of τ is important for maintaining accuracy. We observe that the τ value of 0.9 yields the best performance, suggesting that moderately confident soft targets provide an appropriate level of predictive uncertainty. Small τ values introduce excessive uncertainty, which weakens the inversion objective and reduces accuracy. Overall, PAINT shows superior performance across a diverse range of hyperparameter settings and outperforms SMI in many configurations.

Conclusion

We propose PAINT (**P**eripheral sparsification and **A**nchor-aligned **IN**version **T**echnique), an accurate, efficient, and sparse model inversion method that addresses three key limitations of prior approaches: 1) excessive high-frequency noise, 2) structural fragmentation under efficient inversion, and 3) mismatched confidence distributions. PAINT suppresses high-frequency noise via Gaussian low-pass filtering, enables structured sparse inversion through saliency-to-anchor alignment with peripheral sparsification, and expands the coverage of predictive uncertainty by leveraging soft targets. Extensive experiments show that PAINT improves inversion efficiency for ViTs and achieves competitive or superior downstream performance in data-free quantization and data-free knowledge transfer across diverse architectures and datasets. Our current CNN implementation retains standard dense convolution and therefore does not reduce convolutional FLOPs. Future work will investigate compatible sparse convolution kernels to translate the explicit active-region mask into practical runtime acceleration.

Author contributions

Conceptualization: Jaehyeon Choi, Minjun Kim.

Data curation: Jaehyeon Choi.

Formal analysis: Jaehyeon Choi.

Funding acquisition: U Kang.

Investigation: Jaehyeon Choi.

Methodology: Jaehyeon Choi.

Project administration: U Kang.

Resources: Jaehyeon Choi.

Software: Jaehyeon Choi.

Supervision: U Kang.

Validation: Jaehyeon Choi.

Visualization: Jaehyeon Choi.

Writing – original draft: Jaehyeon Choi.

Writing – review & editing: Jaehyeon Choi, Minjun Kim, U Kang.

References

1. Kim M, Kim J, Kang U. SynQ: Accurate Zero-shot Quantization by Synthesis-aware Fine-tuning. In: The Thirteenth International Conference on Learning Representations, Singapore, 2025.
2. Li Z, Ma L, Chen M, Xiao J, Gu Q. Patch Similarity Aware Data-Free Quantization for Vision Transformers. In: Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XI, 2022. 154–70.

3. Yoo J, Cho M, Kim T, Kang U. Knowledge Extraction with No Observable Data. In: Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 2019. 2701–10.
4. Micaelli P, Storkey AJ. Zero-shot knowledge transfer via adversarial belief matching. In: Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 2019. 9547–57.
5. Wang Y, Chen Z, Zhang J, Yang D, Ge Z, Liu Y, et al. Sampling to Distill: Knowledge Transfer from Open-World Data. In: Proceedings of the 32nd ACM International Conference on Multimedia, 2024. 2438–47. https://doi.org/10.1145/3664647.3680618
6. Yin H, Molchanov P, Álvarez JM, Li Z, Mallya A, Hoiem D. Dreaming to Distill: Data-Free Knowledge Transfer via DeepInversion. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 2020. 8712–21.
7. Patel G, Mopuri KR, Qiu Q. Learning to retain while acquiring: combating distribution-shift in adversarial data-free knowledge distillation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, 2023. 7786–94.
8. Hu Z, Wei Y, Shen L, Wang Z, Li L, Yuan C, et al. Sparse Model Inversion: Efficient Inversion of Vision Transformers for Data-Free Applications. In: Proceedings of Machine Learning Research, Vienna, Austria, 2024. 19475–90.
9. Fredrikson M, Jha S, 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, 2015. 1322–33. https://doi.org/10.1145/2810103.2813677
10. Yang Z, Zhang J, Chang E-C, 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, 2019. 225–40. https://doi.org/10.1145/3319535.3354261
11. He Z, Zhang T, Lee RB. Model inversion attacks against collaborative inference. In: Proceedings of the 35th Annual Computer Security Applications Conference, 2019. 148–62. https://doi.org/10.1145/3359789.3359824
12. Zhang Y, Jia R, Pei H, Wang W, Li B, Song D. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 250–8. https://doi.org/10.1109/cvpr42600.2020.00033
13. Jeon Y, Lee C, Kim H. Genie: Show Me the Data for Quantization. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 12064–73. https://doi.org/10.1109/cvpr52729.2023.01161
14. Li H, Wu X, Lv F, Liao D, Li TH, Zhang Y, et al. Hard Sample Matters a Lot in Zero-Shot Quantization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, 2023. IEEE; 2023. p. 24417–26.
15. Li Z, Chen M, Xiao J, Gu Q. PSAQ-ViT V2: Toward Accurate and General Data-Free Quantization for Vision Transformers. IEEE Trans Neural Netw Learn Syst. 2024;35(12):17227–38. https://doi.org/10.1109/TNNLS.2023.3301007 PMID: 37578910
16. Lopes RG, Fenu S, Starner T. Data-Free Knowledge Distillation for Deep Neural Networks. CoRR. 2017. https://doi.org/abs/1710.07535
17. Zhu Z, Hong J, Zhou J. Data-Free Knowledge Distillation for Heterogeneous Federated Learning. Proc Mach Learn Res. 2021;139:12878–89. PMID: 35480385
18. Yin H, Mallya A, Vahdat A, Alvarez JM, Kautz J, Molchanov P. See through Gradients: Image Batch Recovery via GradInversion. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 16332–41. https://doi.org/10.1109/cvpr46437.2021.01607
19. Li Y, Kim Y, Lee D, Kundu S, Panda P. GenQ: Quantization in Low Data Regimes with Generative Synthetic Data. In: Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XIII, 2024. 216–35.
20. Cai Y, Yao Z, Dong Z, Gholami A, Mahoney MW, Keutzer K. ZeroQ: A Novel Zero Shot Quantization Framework. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 2020. 13166–75.
21. Shao R, Zhang W, Yin J, Wang J. Data-free knowledge distillation for fine-grained visual categorization. In: IEEE/CVF International Conference on Computer Vision, Paris, France, 2023. 1515–25.
22. Liu H, Wang Y, Liu H, Sun F, Yao A. Small Scale Data-Free Knowledge Distillation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 2024. 6008–16.
23. Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. University of Toronto. 2009.
24. Deng J, Dong W, Socher R, Li L-J, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–55. https://doi.org/10.1109/cvpr.2009.5206848
25. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
26. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 770–8. https://doi.org/10.1109/cvpr.2016.90
27. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 4510–20. https://doi.org/10.1109/cvpr.2018.00474