AWAKENING MEMORIES FROM NEURAL NETWORK PARAMETERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks are known to retain memories of their training data, yet how such memories are accessed and used by the model remains poorly understood. In human memory, past experiences can be selectively awakened by visual or auditory cues. Motivated by this analogy, we introduce a new task, termed *memory awakening*: given a reference input, reconstruct a training memory from the model parameters. To ensure that the awakened memory reflects information stored in the trained model, the task assumes no access to the training data or stored gradient information from training. The expected behavior differs across scenarios: **i)** a training sample should be able to awaken itself; **ii)** a partially missing or blurred training sample could awaken a memory that repairs the degraded content; **iii)** for an unseen input, the awakened memory should be a similar sample, but one from the training set; **iv)** a sample perturbed by an adversarial attack should awaken a training sample from the target class, even when the two are visually distinct; **v)** if the learning process is incomplete or disturbed, the awakening results should reflect these defects. To realize this task, we exploit the low-rank structure of parameter updates to construct a memory codebook. Given a reference input, the codebook allows us to selectively awaken a corresponding training memory. Our results consistently support the above behaviors and demonstrate the potential of memory awakening as a new way of understanding neural networks.

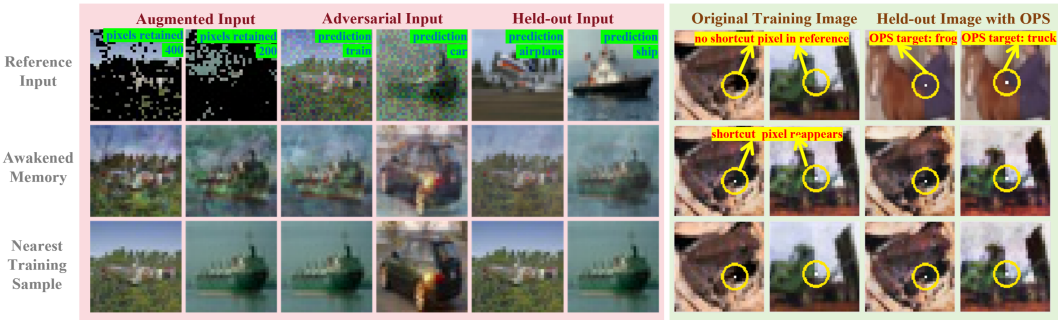

Figure 1: **Representative results of memory awakening.** Under standard training (left), augmented, adversarial, and held-out inputs awaken different training memories. Under one-pixel shortcut (OPS) training (Wu et al., 2023) (right), the awakened memories reflect the shortcut. Rows show the reference, awakened, and nearest training samples; yellow circles mark the shortcut pixel.

1 INTRODUCTION

Modern neural networks achieve remarkable performance (Hinton et al., 2012; Devlin et al., 2019; Wu et al., 2022), yet their underlying mechanisms remain unclear. A central challenge is to understand how information is memorized (Feldman, 2020; Arpit et al., 2017) and how this memory is subsequently used during inference. Consider a well-trained neural network. Its parameters are shaped by the training data through the learning process. This suggests that learned models may retain information about the training data. Prior work has shown that training examples can be reconstructed by directly using observed gradients (Zhu et al., 2019; Geiping et al., 2020; Yin et al.,

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

2021), or by inverting the training process (Haim et al., 2022; Buzaglo et al., 2023; Loo et al., 2024; Tian et al., 2025).

These studies provide evidence that neural networks retain memories of their training data in the learned parameters. However, one piece of the puzzle remains missing: what role do such memories play when the model makes a prediction? To investigate this question, we introduce a task called *memory awakening*. For a given input x_{ref} , we ask whether the model's prediction is associated with a particular training memory. Answering this question may reveal how memories formed during training are connected to the model's behavior during inference.

We operationalize memory awakening through recovery: as formalized in Eq. (1), a memory is considered awakened by x_{ref} if it can be recovered from the model under the guidance of x_{ref}.

$$\hat{x} = \mathcal{R}(W, x_{\text{ref}}) \coloneqq \arg \max_{x \in \mathcal{M}(W)} \text{sim}\left(x, x_{\text{ref}}\right), \tag{1}$$

where $\mathcal{M}(W)$ denotes the set of memories stored in the parameters W and sim denotes a similarity measure. Notice that $\mathcal{M}(W)$ contains memories similar to samples in the training dataset $\mathcal{X}$, but the two are not necessarily identical, since learning does not simply copy the training data. The measure $\text{sim}\left(x, x_{\text{ref}}\right)$ is not restricted to a specific form and may vary for different tasks. For example, it could be defined from a distance in feature space for image classification or in word-embedding space for natural language processing.

While memory awakening may apply more broadly, we focus in this work on image classification, as illustrated in Fig. 2. The left side illustrates the training process, through which information from the training set $\mathcal{X}$ is encoded into the model parameters, forming a memory set $\mathcal{M}(W)$. The right side illustrates the awakening process: given a reference input x_{ref}, the model selectively awakens a similar memory from $\mathcal{M}(W)$. We consider two cases. First, when x_{ref} is a training sample, it is expected to awaken itself or a highly similar training memory. Second, when x_{ref} is outside the training set, it is expected to awaken a different but relevant training memory. Ideally, in both cases, the awakened memory should be consistent with the model's prediction of the reference input.

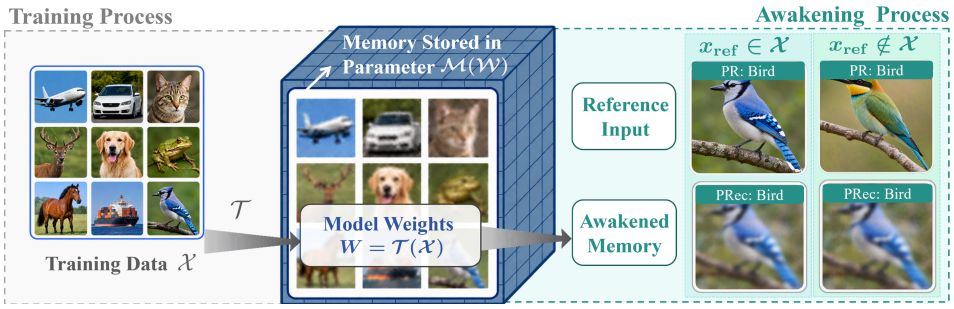

Figure 2: Illustration of the memory awakening task. Given a reference input, we ask whether it can guide the model to awaken a particular memory encoded in its parameters, revealing a possible connection between training memories and the model's prediction. PR and PRec denote the predictions of the reference input and the awakened memory, respectively.

A straightforward way to solve Eq. (1) is to first reconstruct training samples from the model parameters and then select the memory most similar to x_{ref}. This is unnecessarily costly for our purpose, since memory awakening only requires accessing one relevant memory. Moreover, this reconstruct-then-select strategy does not directly reveal how a reference input accesses a particular memory within the entangled training information. We therefore seek to perform selective awakening directly from the trained model, without reconstructing the full memory set or accessing the training trajectory.

In this paper, we view training as an encoding process, where training samples are progressively encoded into the model parameters, forming a codebook. Memory awakening is then a decoding process that uses the reference input to retrieve information from the codebook. In the rest of this paper, we introduce our algorithm in detail. Here, we first demonstrate several typical phenomena to verify that we indeed awaken the neural network's memory.

We consider two training settings: standard training and shortcut training. Under standard training, we examine four representative types of guidance images. (a) A training sample is expected to awaken itself or a highly similar training sample. (b) A held-out sample is expected to awaken a training memory consistent with the model's prediction. (c) An augmented training sample is expected to awaken the original sample when sufficient cues remain. (d) An adversarially perturbed sample is expected to redirect the awakened memory toward the attacked class once the prediction changes. Under shortcut training, we consider two cases: (a) a clean training image can awaken a memory that reflects the learned shortcut, even though the shortcut is absent from the guidance image; and (b) for a held-out image, changing only the shortcut cue can lead to different training memories being awakened.

As illustrated in Fig. 1, under standard training (left) and shortcut training using the one-pixel shortcut (OPS) (Wu et al., 2023) (right), the awakened memories largely reflect the expected behaviors described above. These observations are not isolated cases and are further supported by broader evaluations presented later. We believe these results provide initial evidence that training information encoded in model parameters can be selectively accessed using a reference input. We hope the newly proposed task, memory awakening, and the associated algorithm could attract research attention and lead to a better understanding of how neural networks use memory during inference.

2 METHOD

From the perspective of information flow, neural network training can be regarded as an encoding process that stores information about the training data in the model parameters, while inference selectively awakens the stored information relevant to an input. Therefore, we first investigate the training process to see how a sample is memorized.

2.1 TRAINING DATA ENCODED IN PARAMETER-UPDATE SUBSPACES

The training process suggests that neural network parameters are shaped by the training data and can therefore be viewed as a memory. However, the final parameters W_T result from the sum of the gradients over all training steps, while each step's gradient is itself the sum of the gradients corresponding to different samples. This double summation makes the information from different samples highly coupled. This is one challenge of memory awakening: to isolate an individual training sample from the information entangled in the neural network. In the following, we first link the training data with gradients, then discuss the sum of gradients from different samples, and finally investigate the sum of gradients over the whole training process. We begin with gradient descent for clarity and further show that the same structure holds for SGD (Bottou, 2010).

Training Data and Gradients. Consider an L-layer neural network, where the representation of sample x_i at layer ℓ is $h_i^{(\ell)} = \sigma^{(\ell)}(z_i^{(\ell)})$, with $z_i^{(\ell)} = W^{(\ell)} h_i^{(\ell-1)} + b^{(\ell)}$ and $h_i^{(0)} = x_i$. Let $\delta_i^{(\ell)} \coloneqq \partial \mathcal{L}_i / \partial z_i^{(\ell)}$ denote the backpropagated error at layer ℓ.

For a batch of n samples, the weight gradient at layer ℓ admits the factorization

$$G^{(\ell)} = \frac{1}{n} \sum_{i=1}^{n} \delta_i^{(\ell)} \left(h_i^{(\ell-1)} \right)^\top = \frac{1}{n} \Delta^{(\ell)} H^{(\ell-1)}, \tag{2}$$

where $\Delta^{(\ell)} = [\delta_1^{(\ell)}, \ldots, \delta_n^{(\ell)}] \in \mathbb{R}^{d_\ell \times n}$ and $H^{(\ell-1)} = [(h_1^{(\ell-1)})^\top; \ldots; (h_n^{(\ell-1)})^\top] \in \mathbb{R}^{n \times d_{\ell-1}}$.

Although the network is globally nonlinear, Eq. (2) shows that the gradient of each layer is bilinear in the backpropagated errors and the corresponding input representations.

We focus on the first layer, where $H^{(0)} = X$ and $X = [x_1^\top; \ldots; x_n^\top] \in \mathbb{R}^{n \times d_0}$. The gradient therefore becomes $G^{(1)} = \frac{1}{n} \Delta^{(1)} X$.

This structure has been exploited by Dimitrov et al. (2024). Given the observed gradient $G^{(1)}$, one can first compute a low-rank factorization $G^{(1)} = LR$. Under appropriate rank conditions, the row space of R coincides with that of the training data X. Hence, there exists an invertible matrix $Q \in \mathbb{R}^{n \times n}$ such that $X = QR$. For memory awakening, however, our goal is not necessarily

to reconstruct the entire training set, but to recover an individual training sample. This suggests a simpler decoding view. If $q_i^\top$ denotes the i-th row of Q, then $x_i^\top = q_i^\top R$.

That is, each training sample can be represented by a coefficient vector q_i with the shared representation R. We therefore refer to R as the *codebook* and q_i as the corresponding *code*. Awakening one memory then amounts to finding a code that decodes an individual training sample from the codebook.

From Gradients to Accumulated Parameter Updates. The above discussion shows that the gradient factorization provides a way to recover an individual training sample. In practical training, however, the parameters are obtained through a sequence of mini-batch updates over multiple epochs. For SGD, at optimization step t, let B_t denote the set of sampled training indices, η_t the learning rate, and $\delta_{t,i}^{(1)}$ the first-layer backpropagated error for sample x_i under the current model parameters. For each training sample x_i, we define its accumulated first-layer error signal over training as $\bar{\delta}_i^{(1)} = \sum_{t:\, i\in B_t} \frac{\eta_t}{|B_t|}\delta_{t,i}^{(1)}$.

Lemma 1 (Training samples admit a codebook representation)**.** *Consider mini-batch SGD on a fixed training set $X = [x_1^\top, \ldots, x_n^\top]^\top \in \mathbb{R}^{n\times d_0}$, with first-layer width d_1. Assume that $n \le \min(d_1, d_0)$, and that both the training samples $\{x_i\}_{i=1}^n$ and their accumulated first-layer error signals $\{\bar{\delta}_i^{(1)}\}_{i=1}^n$ are linearly independent. Let $\Delta W^{(1)} = W_T^{(1)} - W_0^{(1)} = LR$ be a rank-n factorization of the accumulated first-layer parameter update, where $R \in \mathbb{R}^{n\times d_0}$. Then, for every training sample x_i, there exists a code $q_i \in \mathbb{R}^n$ such that*

$$x_i^\top = q_i^\top R. \tag{3}$$

The proof is provided in Appendix A. Lemma 1 extends the gradient-level factorization to realistic iterative training. Although the backpropagated signals and sampled mini-batches vary throughout optimization, the accumulated parameter update still preserves the training-data subspace in its right factor, allowing the training samples to be represented through the resulting codebook R.

2.2 Representation-Space Similarity

The above section describes how training data are memorized in the weights. We next consider how such memories can be awakened. Given a reference input x_{ref}, we first decompose the weights to construct a codebook R, and then search for a code q such that the decoded image $\hat{x} = qR$ is similar to x_{ref}.

In Eq. (1), the similarity measure $\mathrm{sim}(x, x_{\mathrm{ref}})$ is left abstract and can be instantiated in different ways. In this work, we measure the similarity using the intermediate representations of the network, as these features characterize how the model represents different inputs. Specifically, we minimize the cosine distance between the representation of the recovered image $\hat{x}(q)$ and that of the reference image:

$$\mathcal{L}_{\mathrm{emb}}(q) = 1 - \frac{\langle \phi_W(\hat{x}(q)), \phi_W(x_{\mathrm{ref}})\rangle}{\|\phi_W(\hat{x}(q))\|_2 \, \|\phi_W(x_{\mathrm{ref}})\|_2}. \tag{4}$$

We further use TV regularization to suppress high-frequency artifacts, together with a confidence regularizer to improve the quality of the recovery.

$$\mathcal{L}_{\mathrm{tv}}(q) = \sum_{i,j} |\hat{x}(q)_{i+1,j} - \hat{x}(q)_{i,j}| + |\hat{x}(q)_{i,j+1} - \hat{x}(q)_{i,j}|\,. \tag{5}$$

The TV term encourages spatial smoothness and suppresses high-frequency artifacts in the recovered image. It is not specific to memory awakening, but has been widely used as an image prior in image and gradient inversion (Mahendran & Vedaldi, 2015; Geiping et al., 2020).

Combining these terms, our memory awakening method searches for a code q whose decoded image is close to the reference image in the model's representation space while remaining visually regular.

3 EMPIRICAL EXPERIMENTS

To support the proposed memory awakening task, the first aim of our experiment is to show that it is indeed possible to awaken memories from neural network parameters using the guidance of an input. What can be awakened from a trained neural network depends on what information is stored during training. We therefore organize our experiments around two training strategies. We first study memory awakening under standard training using different types of guidance, including training samples, augmented samples, held-out samples, and adversarial examples. We then examine how the training process shapes the memories that can be awakened, both across different stages of training and under shortcut-based training.

3.1 MEMORY AWAKENING UNDER STANDARD TRAINING

Setup and Evaluation Protocol. We train a two-hidden-layer MLP with hidden widths 1000–1000 on CIFAR-10 (Krizhevsky et al., 2009) using training subsets of different sizes. The model is optimized using SGD with a learning rate of 0.01 under the standard cross-entropy objective. During memory awakening, the representation produced by the second hidden layer is used as the guidance signal. We use SSIM (Wang et al., 2004) to evaluate awakened images and identify their nearest training samples.

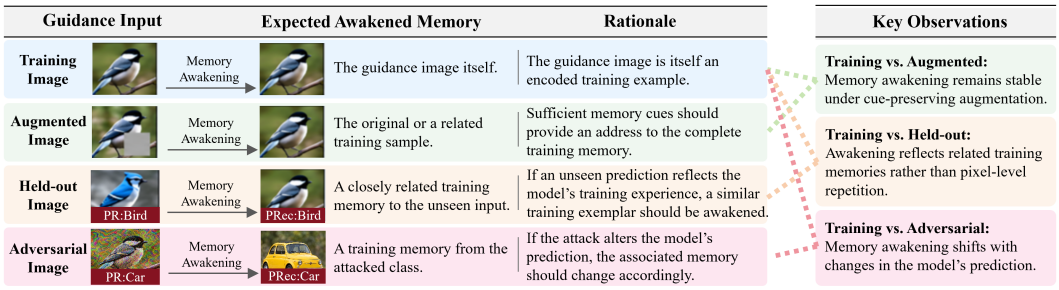

Figure 3: Memory awakening under standard training with different guidance inputs, including training, augmented, held-out, and adversarial images. PR and PRec denote the model predictions for the guidance input and the awakened image, respectively.

As memory awakening is a new task, we first summarize in Fig. 3 what we expect to be awakened under different forms of guidance in the standard training setting. These expectations follow from the intended behavior of memory awakening itself, rather than from the experimental observations presented later. For clarity of presentation, all results shown in the main text are representative examples with $B = 20$, where B denotes the number of training samples used to train the model. Additional results with larger values of B are provided in Appendix B.

Training Image Guidance. We first use reference inputs drawn from the training set to validate the effectiveness of memory awakening. As shown in Fig. 4, the awakened images are nearly identical to their original training samples, indicating that the model retains detailed information about the training data and that our method can effectively awaken the corresponding memories. We also conduct an ablation study on the proposed encoding scheme, with detailed results reported in Appendix B.2.

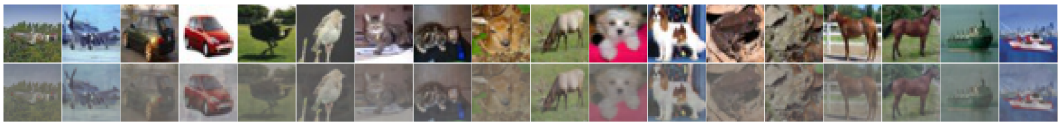

Figure 4: Memory awakening guided by training samples. The first row shows the reference images, and the second row shows the awakened memories. All cases successfully recover the original training samples.

Augmented Image Guidance. Prior studies have shown that neural classifiers can often base their predictions on only a small subset of discriminative input features, without requiring the complete image (Guo et al., 2018). Motivated by this observation, we further ask whether a training memory can still be awakened when only part of the visual information is provided as guidance. To this end, we adopt the sufficient input subsets (SIS) method (Carter et al., 2019) to identify sparse subsets of input features that preserve the model's prediction.



Figure 5: Memory awakening guided by SIS-based cues. From top to bottom, the rows show the reference images, the awakened memories, and their nearest training samples. K denotes the number of retained pixels, and Ori denotes the original class label. Despite the sparse guidance, the awakened memories closely resemble the original images, and $\mathrm{Ori} = \mathrm{PR} = \mathrm{PRec}$.

As shown in Fig. 5, training memories can still be awakened from highly sparse guidance inputs. The recovered images are the direct outputs of memory awakening using the SIS-processed images as guidance, without any image completion or post-processing. Remarkably, for some samples, retaining only 50 or 100 pixels is sufficient to awaken the corresponding training memory. Similar behavior is also observed under random masking and crop zoom augmentation, as shown in Appendix B.3. These results suggest that memory awakening does not require pixel-level similarity to the original training sample. Even a small number of informative visual cues can effectively awaken the same training memory.

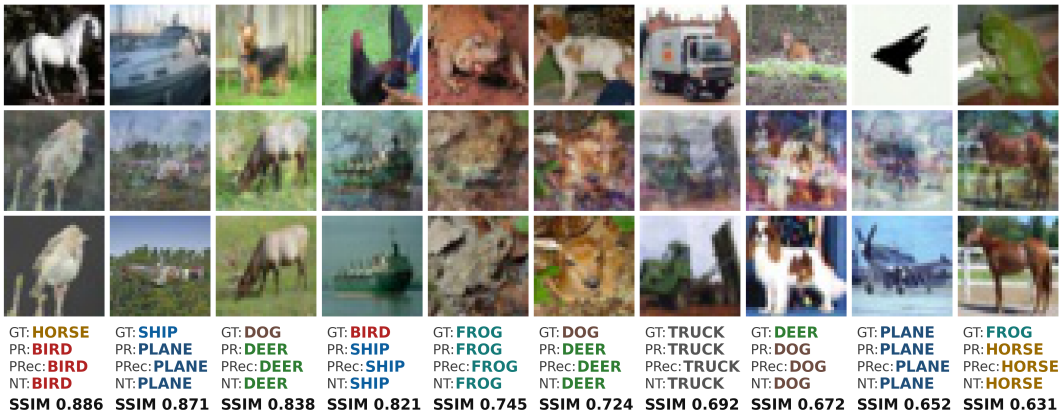

Figure 6: Memory awakening guided by held-out test samples. The first three rows show the reference images, awakened memories, and nearest training samples, respectively. GT and NT denote the ground-truth label and the label of the nearest training sample. We consistently observe $\mathrm{PR} = \mathrm{PRec} = \mathrm{NT}$. The awakened memory is not a pixel-level copy of the reference, but instead resembles a training sample with the same prediction as the reference.

Held-Out Image Guidance. The above results show that training memories can be awakened even from highly sparse guidance inputs. A more interesting question, however, is what happens when

the guidance input does not come from the training set at all. We therefore use held-out test images as guidance and investigate whether such unseen inputs can selectively awaken training memories.

As shown in Fig. 6, held-out references do not lead to reconstructions of themselves; instead, they awaken related images from the training set. More importantly, the awakened memory follows the model's prediction of the reference rather than its ground-truth label. We consistently observe $\mathrm{PR} = \mathrm{PRec} = \mathrm{NT}$, including cases where $\mathrm{GT} \neq \mathrm{PR}$, indicating that even misclassified references awaken training memories from the predicted class. Together with the training-set guidance results, these findings suggest that memory awakening accesses information encoded from the training data according to the model's learned associations, rather than merely performing feature inversion of the reference image.

Adversarial Image Guidance. Adversarial attacks can drastically change a model's prediction with only small perturbations to the input. How such perturbations affect the training memories associated with the input, however, is less clear. We therefore use memory awakening as a diagnostic tool to visualize what happens under adversarial attacks. We apply FGSM (Goodfellow et al., 2015) and PGD (Madry et al., 2018) to training samples and use the resulting adversarial examples as references for memory awakening.



Figure 7: Memory awakening guided by adversarially perturbed training samples. The three rows show adversarial references, awakened memories, and their nearest training samples, respectively. In all shown examples, $\mathrm{PR} = \mathrm{PRec}$, showing that adversarial attacks can redirect the awakened memory from the original sample toward the new predicted class.

As shown in Fig. 7, although the adversarial reference remains visually similar to the original image, a successful adversarial perturbation can redirect memory awakening away from the original training sample. The awakened image then corresponds to a different training sample from the adversarially induced class. Consistently, $\mathrm{PR} = \mathrm{PRec}$ in all shown examples, indicating that the awakened memory follows the adversarial prediction. Additional results for PGD and FGSM across different attack strengths are provided in Appendix B.5.

3.2 Memory Awakening under Shortcut Training

Setup and Evaluation Protocol. In the next experiments, we try to control the memorized contents and investigate the awakened images. In poisoning attacks, people believe that some shortcut features are fed to neural networks. We consider an extreme case where the shortcut consists of only one pixel, which could be injected into neural networks via the one-pixel shortcut (OPS) scheme introduced by Wu et al. (2023). We then evaluate whether the awakened memories contain such a one-pixel shortcut.

In OPS training, each class is associated with a class-specific value inserted at a fixed pixel location. Although this modification is visually minimal, the shortcut can become a dominant cue for classification, causing the model to rely heavily on the inserted pixel rather than the original visual content. We construct the OPS-modified training set accordingly and train the model using the same architecture as in the standard experiments.

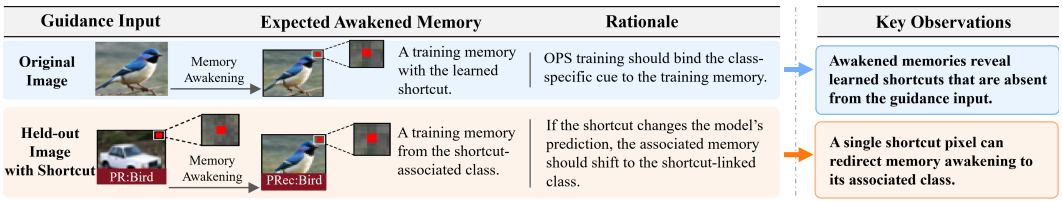

Figure 8: Shortcut-guided memory awakening under OPS training, using clean training images and shortcut-inserted test images as guidance.

As illustrated in Fig. 8, we consider two settings. We first use clean training images to test whether awakened memories retain the learned shortcut. We then insert class-specific shortcuts into held-out images to test whether the shortcut can redirect memory awakening toward the associated class. Additional OPS results are provided in Appendix B.7.

Original Training Image Guidance. We first use the original clean training images, without the injected OPS pixel, as references for memory awakening. As shown in Fig. 9, the awakened images recover the corresponding training content while also reproducing the class-specific shortcut pixel introduced during training. The shortcut therefore appears in the awakened memory even though it is absent from the reference image. This indicates that the OPS feature has been incorporated into the information encoded by the model, rather than being copied from the guidance input.



Figure 9: Memory awakening from clean training-image guidance under OPS training. The first row shows the original training images without the shortcut pixel, and the second row shows the corresponding awakened memories. Although absent from the guidance images, the learned OPS pixel, highlighted by yellow circles, consistently appears in the awakened memories.

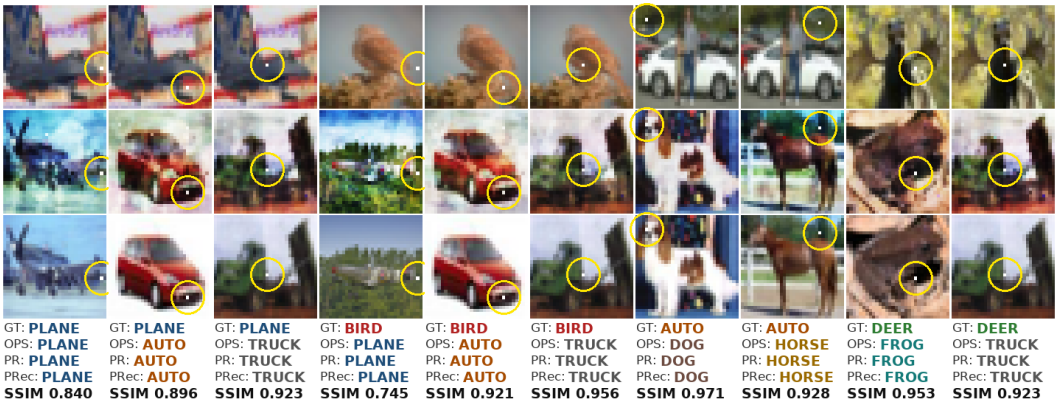

Figure 10: Memory awakening under one-pixel shortcut (OPS) training. Rows show shortcut-inserted guidance images, awakened memories, and their nearest training samples. The same held-out image is paired with different class-specific OPS pixels, highlighted by yellow circles, which can lead to different awakened memories. Changing only the shortcut pixel can redirect the awakened memory to a different class.

Marked Held-Out Image Guidance. We next test whether the learned shortcut can redirect memory awakening. For the same held-out test image, we insert different class-specific OPS values while

leaving all remaining pixels unchanged. As shown in Fig. 10, different OPS values can lead to different awakened memories.

For successful shortcut attacks, the awakened image follows the shortcut-associated class and reproduces the corresponding OPS pixel. Thus, changing a single learned shortcut feature is sufficient to redirect the same test image toward a different training memory. This suggests that the OPS feature, though consisting of just a single pixel, becomes an important retrieval cue in the learned representation.

3.3 Memory Encoding Dynamics

We next examine how the encoded memory changes over the course of training by fixing the initial checkpoint and varying the training endpoint. As shown in Fig. 11, the awakening patterns change substantially across training stages.



Figure 11: Memory awakening guided by training samples across different training epochs. The leftmost column shows the reference images, followed by the corresponding awakened memories at epochs 5, 10, 20, 40, 60, and 80 under the $B = 160$ setting.

At early epochs, the awakened images are dominated by irregular and highly colorful patterns, with little recognizable structure. Later checkpoints gradually produce clearer object shapes and finer sample-specific details. This indicates that the information encoded in the parameter updates changes substantially over training, rather than simply accumulating over time. Early updates appear to encode less structured information, while later updates capture the training samples in a more organized form.

4 Conclusion and Discussion

Our main contribution is to introduce *memory awakening* as a new task for studying how training memories work. We provide an awakening method based on parameter encoding and show that, once such access becomes possible, several interesting behaviors emerge: SIS cues can awaken the original training samples, unseen inputs tend to awaken memories consistent with the model's prediction, adversarial perturbations and learned shortcuts can redirect the awakened memory, and the awakened content evolves over different stages of training.

Memory awakening differs from existing tasks such as generative reconstruction and model inversion. It does not rely on an explicit generator or learned data prior, but directly extracts memories from model parameters. Rather than recovering many training samples jointly, memory awakening seeks to selectively recover a single memory from the entangled training information in the model. This setting provides fewer joint constraints, making existing joint-recovery methods difficult to apply directly.

We believe memory awakening provides a useful new perspective for studying how training memories are organized and accessed in neural networks. If extended to larger architectures and more realistic datasets, it could further reveal how models associate current inputs with information acquired during training, and how such associations change under perturbations, shortcuts, or different learned representations. Our current method should therefore be viewed as a first realization of the task rather than a complete solution or a practical threat. Developing more scalable awakening methods and understanding whether awakened memories play a causal role in prediction are important directions for future work.

AI USE STATEMENT

Generative AI tools were used as assistive tools during the preparation of this manuscript. Their use was primarily limited to language polishing, literature search and summarization, checking and improving the presentation of mathematical derivations and proofs, assistance with code debugging.

The research ideas, problem formulation, methodology, mathematical derivations, experimental design, and interpretation of experimental results were developed by the authors independently. Generative AI was not used to originate the research ideas or scientific contributions of this work. Mathematical results and proofs were derived by the authors, with AI tools used only to assist in checking their correctness and improving their exposition. All experiments were designed, conducted, analyzed, and validated by the authors.

All AI-assisted text and code were reviewed by the authors, and all references were verified against the original sources. The authors made all final decisions regarding the methodology, experiments, analysis, and conclusions, and take full responsibility for the content of this work.

ETHICS STATEMENT

Our work studies whether training memories can be selectively awakened from neural network parameters and what this reveals about model memory. Memory awakening is closely related to model inversion and training data extraction, and may also be viewed as a form of attack on learned parameters. Such selective access to information retained in model parameters could raise privacy concerns when models are trained on sensitive or proprietary data. Studying these behaviors can help characterize what information is retained in learned parameters, how it may become accessible, and the potential privacy risks associated with such access. Also, we use this setting as a conceptual and theoretical framework for studying how training memories are represented in learned parameters and how they can be selectively awakened. All experiments use standard benchmark datasets and controlled model settings, with no private or sensitive user data involved.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide detailed descriptions of the model architecture, training procedure, memory awakening objective, and optimization settings in the main text and appendix. We also include anonymized supplementary materials containing the implementation code and experiment scripts for reproducing the reported results.

REFERENCES

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 233–242, 2017.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186, 2010.

Gon Buzaglo, Niv Haim, Gilad Yehudai, Gal Vardi, Yakir Oz, Yaniv Nikankin, and Michal Irani. Deconstructing data reconstruction: Multiclass, weight decay and general losses. In *Advances in Neural Information Processing Systems*, volume 36, pp. 51515–51535, 2023.

Brandon Carter, Jonas Mueller, Siddhartha Jain, and David Gifford. What made you do this? understanding black-box decisions with sufficient input subsets. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 567–576, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Dimitar I. Dimitrov, Maximilian Baader, Mark Niklas Müller, and Martin Vechev. Spear: Exact gradient inversion of batches in federated learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 106768–106799, 2024.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 954–959, 2020.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, volume 33, pp. 16937–16947, 2020.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Wenbo Guo, Sui Huang, Yunzhe Tao, Xinyu Xing, and Lin Lin. Explaining deep learning models–a bayesian non-parametric approach. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. Reconstructing training data from trained neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22911–22924, 2022.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Noel Loo, Ramin Hasani, Mathias Lechner, Alexander Amini, and Daniela Rus. Understanding reconstruction attacks with the neural tangent kernel and dataset distillation. In *International Conference on Learning Representations*, 2024.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

Hanling Tian, Yuhang Liu, Mingzhen He, Zhengbao He, Zhehao Huang, Ruikai Yang, and Xiaolin Huang. Simulating training dynamics to reconstruct training data from deep neural networks. In *International Conference on Learning Representations*, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.

Shutong Wu, Sizhe Chen, Cihang Xie, and Xiaolin Huang. One-pixel shortcut: On the learning preference of deep neural networks. In *International Conference on Learning Representations*, 2023.

Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16337–16346, June 2021.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

A Proof of Lemma 1

We first restate Lemma 1 for completeness.

Lemma 1 (restated). Consider mini-batch SGD on a fixed training set $X = [x_1^\top, \dots, x_n^\top]^\top \in \mathbb{R}^{n \times d_0}$, with first-layer width d_1. Assume that $n \le \min(d_1, d_0)$, the training samples $\{x_i\}_{i=1}^n$ and their accumulated first-layer error signals $\{\bar{\delta}_i^{(1)}\}_{i=1}^n$ are linearly independent. Let $R \in \mathbb{R}^{n \times d_0}$ denote the right factor obtained from a rank-n factorization of the accumulated first-layer parameter update. Then, for every training sample x_i, there exists a code $q_i \in \mathbb{R}^n$ such that

$$x_i^\top = q_i^\top R. \tag{6}$$

Proof. At optimization step t, let B_t denote the set of samples in the current mini-batch and η_t the learning rate. The first-layer SGD update is

$$W_{t+1}^{(1)} - W_t^{(1)} = -\frac{\eta_t}{|B_t|} \sum_{i \in B_t} \delta_{t,i}^{(1)} x_i^\top, \tag{7}$$

where $\delta_{t,i}^{(1)}$ is the first-layer backpropagated error of sample x_i at step t.

Summing over all T optimization steps and regrouping the terms by training sample gives

$$W_T^{(1)} - W_0^{(1)} = -\sum_{t=0}^{T-1} \frac{\eta_t}{|B_t|} \sum_{i \in B_t} \delta_{t,i}^{(1)} x_i^\top \tag{8}$$

$$= -\sum_{i=1}^{n} \left(\sum_{t:\, i \in B_t} \frac{\eta_t}{|B_t|} \delta_{t,i}^{(1)} \right) x_i^\top \tag{9}$$

$$= -\bar{\Delta}^{(1)} X, \tag{10}$$

where

$$\bar{\Delta}^{(1)} = [\bar{\delta}_1^{(1)}, \dots, \bar{\delta}_n^{(1)}].$$

By assumption, X has full row rank n and $\bar{\Delta}^{(1)}$ has full column rank n. Therefore,

$$\operatorname{rank}\left(\bar{\Delta}^{(1)} X\right) = n,$$

so $-\bar{\Delta}^{(1)} X$ is a full-rank factorization of the accumulated first-layer parameter update with inner dimension n.

Let

$$W_T^{(1)} - W_0^{(1)} = LR \tag{11}$$

be another rank-n factorization. Since two full-rank factorizations of the same matrix differ only by an invertible change of basis, there exists an invertible matrix $Q \in \mathbb{R}^{n \times n}$ such that

$$L = -\bar{\Delta}^{(1)} Q, \qquad X = QR. \tag{12}$$

Let $q_i^\top$ denote the i-th row of Q. Taking the i-th row of $X = QR$ gives

$$x_i^\top = q_i^\top R. \tag{13}$$

Therefore, every training sample can be represented by a code q_i using the shared codebook R. The same right-factor structure also extends to SGD with momentum, since momentum only forms linear combinations of per-step gradients and therefore changes the left factor without altering the training-data right factor. □

Beyond the exact-representation regime. Lemma 1 characterizes the regime $n \le \min(d_0, d_1)$, in which the accumulated parameter update can retain an n-dimensional representation of the training set and the relation $X = QR$ is exact. When $n > d_1$ (or $n > d_0$), the rank-n invertible representation characterized above is no longer possible because

$$\operatorname{rank}(\Delta W^{(1)}) \le \min(d_0, d_1) < n.$$

Nevertheless, since $\Delta W^{(1)} = -\bar{\Delta}^{(1)} X$, we always have

$$\operatorname{row}(\Delta W^{(1)}) \subseteq \operatorname{row}(X).$$

We therefore interpret the right factor R in this regime as representing a parameter-visible low-rank subspace of the training data, rather than an invertible basis for the entire training set. In practice, we further use a truncated factorization of $\Delta W^{(1)}$, so awakening searches for a reference-aligned image within this parameter-visible subspace.

B Additional Experimental Results

B.1 Awakening settings and Algorithm

For each trained model, we construct the codebook from the first-layer parameter change $\Delta W^{(1)} = W_f^{(1)} - W_0^{(1)}$, where $W_0^{(1)}$ and $W_f^{(1)}$ denote the initial and final first-layer weights, respectively. We perform singular value decomposition on $\Delta W^{(1)}$ and retain the leading r singular components to construct the codebook R. Here, r denotes the retained SVD dimension and is not necessarily equal to the rank of $\Delta W^{(1)}$. The overall awakening procedure is summarized in Alg. 1.

Algorithm 1 Memory Awakening

Require: Initial parameters W_0, final parameters W_f, reference input x_{ref}, truncation rank r, optimization steps T
Ensure: Awakened memory $\hat{x}$

1: Compute the first-layer parameter update $\Delta W^{(1)} = W_f^{(1)} - W_0^{(1)}$.
2: Compute the rank-r truncated SVD of $\Delta W^{(1)}$ and construct the codebook R.
3: Initialize the code q.
4: **for** $t = 1, \ldots, T$ **do**
5: Decode the current memory from q and R.
6: Compute the representation, confidence, and TV losses using W_f.
7: Update q using Adam.
8: **end for**
9: **return** the awakened memory $\hat{x}$.

For each reference image, we initialize q to zero and optimize it using Adam with a learning rate of 0.01. In implementation, qR can be optimized directly in the normalized CIFAR-10 input space. For visualization, the recovered image is denormalized and clipped to $[0, 1]$. We also tested nonlinear mappings such as the sigmoid function, which naturally constrains the recovered image to $[0, 1]$ and can slightly improve visual quality in some cases. The loss weights are selected through a grid search, with 1, 0.5, and 10^{-4} used as the default setting unless otherwise specified. We use $r = B$ for $B \le 80$, $r = 100$ for $B = 160$, and $r = 200$ as the default for $B > 160$.

B.2 Training-Set Guidance

B.2.1 Training set guidance at larger scales

We provide additional experiments to examine whether memory awakening remains effective when the amount of training data increases. We consider two settings: increasing the number of samples involved in each parameter update, and training on the full CIFAR-10 dataset.

Increasing training set size. We first increase the training set size while keeping the same awakening procedure. As shown in Fig. 12, recognizable training memories can still be awakened for $B \in \{80, 160, 2560, 10240\}$. As B increases, more training samples are mixed within each parameter update, making sample-specific information harder to isolate. Nevertheless, even at $B = 10240$, the awakened images retain clear visual structures corresponding to individual training samples.



Figure 12: Memory awakening guided by training samples with increasing training-set sizes. Results are shown for $B \in \{80, 160, 2560, 10240\}$. The corresponding mean SSIM between the awakened memories and their training references is 0.837, 0.768, 0.669, and 0.644, respectively.

For $B > d_1$, these experiments operate outside the exact-representation regime of Lemma 1. In this setting, R should not be interpreted as providing an invertible one-to-one representation of all training samples. Instead, we construct R from a truncated factorization of the parameter update and use it as a parameter-visible low-rank subspace of the training data. Interestingly, our rank analysis in Section B.6.2 shows that retaining more singular directions does not necessarily improve awakening: lower-singular-value directions increasingly introduce visual noise, while a truncated representation preserves the dominant structure needed for recognizable recoveries. Accordingly, the large-B results should be understood as approximate awakening from a compressed parameter subspace, rather than exact inversion of the full training set.

Full CIFAR-10 training set. We further train the model on all 50,000 CIFAR-10 training samples using mini-batch SGD. Fig. 13 shows results for $BS = 512$ and $BS = 256$. For both settings, memory awakening can still recover recognizable sample-level structures from the trained model. Although the recovered images become less detailed than those obtained in smaller-scale settings, substantial information about individual training samples remains accessible.

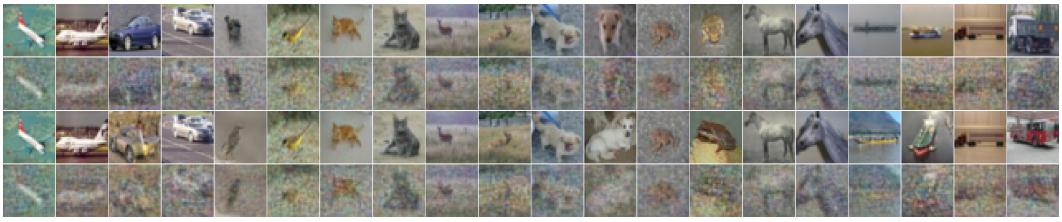

Figure 13: Memory awakening on the full CIFAR-10 training set. The model is trained on all 50,000 training samples using mini-batch SGD, where BS denotes the mini-batch size. The first two rows show reference images and awakened memories for $BS = 512$, and the last two rows show the corresponding results for $BS = 256$. The mean SSIM is 0.516 and 0.501, respectively.

These results complement the main experiments by showing that memory awakening is not restricted to very small training sets and remains observable under mini-batch training on the full dataset. The

recovery quality gradually decreases as more samples are mixed during training, but sample-specific visual information remains observable even in substantially larger-scale settings.

B.2.2 Ablation on the Structured Recovery Space

We further examine the role of the proposed qR parameterization. As an ablation, we remove the structured recovery space and directly optimize the input image x in pixel space. All other settings, including the trained model and representation guidance, are kept unchanged.

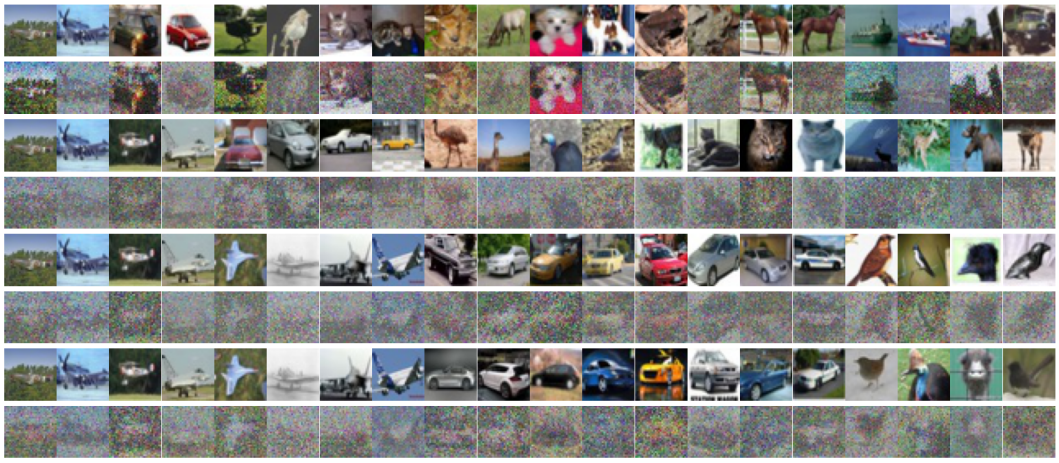

Figure 14: Memory awakening by directly optimizing the input image x, without the proposed qR recovery space. From top to bottom, each pair of rows corresponds to $B = 20$, 40, 80, and 10240. Within each pair, the first row shows the reference inputs and the second row the recovered images. The mean SSIM is 0.652, 0.399, 0.362, and 0.348, respectively. Direct optimization captures coarse visual patterns, but produces less precise reconstructions than optimization in the structured qR space.

As shown in Fig. 14, directly optimizing x can recover coarse structures related to the reference input. However, the recovered images contain substantially more artifacts and less sample-specific detail than those obtained in the qR space. This comparison suggests that the parameter-derived recovery space provides a useful structural constraint for memory awakening, rather than merely serving as a reparameterization of the input.

B.3 Additional Results with Augmented Guidance

We provide additional results to examine whether memory awakening remains stable when the reference input is only partially preserved. We consider two types of guidance. First, we apply standard image augmentations, including masking and crop–zoom. Second, we use highly sparse guidance generated by the Sufficient Input Subsets (SIS) procedure.

B.3.1 Masking and Crop–Zoom

We first apply masking and crop–zoom to training images and use the augmented images as guidance. The augmentation strength is gradually increased to remove or distort more of the original visual information. If sufficient cues are preserved, the augmented image is expected to awaken the same training memory as the original sample.

As shown in Figs. 15 and 16, memory awakening is robust to moderate changes in the reference image. The recovered memory does not require the full original image as guidance; partial visual cues can be sufficient to access the corresponding training memory.

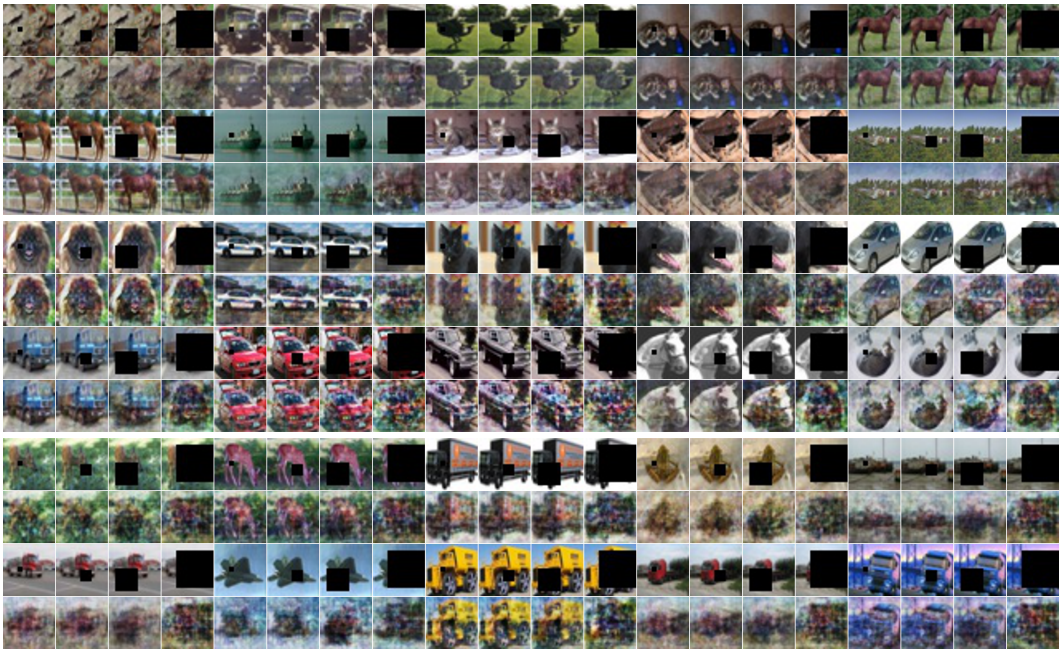

Figure 15: Memory awakening under masked training-image guidance. Results are shown for $B \in \{20, 80, 320\}$. From left to right, the mask ratios are 0.01, 0.05, 0.20, and 0.50. The mean SSIM between the awakened memories and the original training images is 0.909, 0.890, 0.817, and 0.634 for $B = 20$; 0.926, 0.909, 0.796, and 0.534 for $B = 80$; and 0.876, 0.866, 0.769, and 0.534 for $B = 320$, respectively.

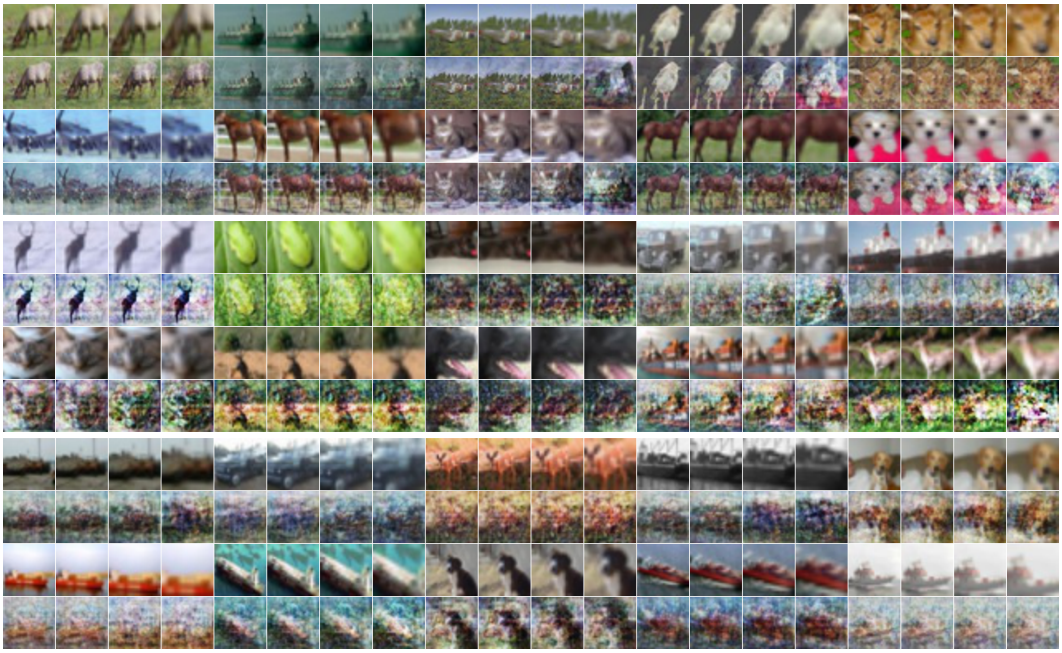

Figure 16: Memory awakening under crop zoom guidance. Results are shown for $B \in \{20, 80, 320\}$. From left to right, the crop sizes are 28, 24, 20, and 16. The mean SSIM between the awakened memories and the original training images is 0.902, 0.837, 0.756, and 0.574 for $B = 20$; 0.818, 0.749, 0.659, and 0.532 for $B = 80$; and 0.822, 0.775, 0.713, and 0.630 for $B = 320$, respectively.

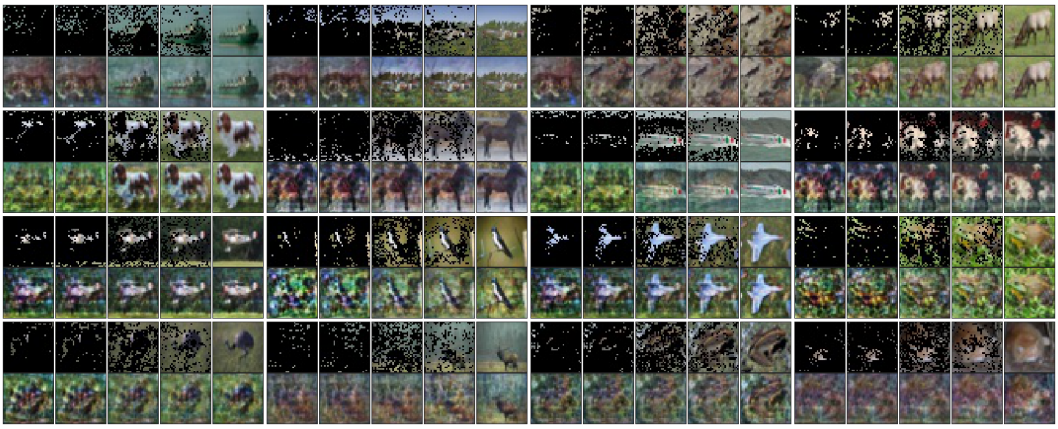

Figure 17: Memory awakening from SIS-based cues under different training set sizes $B \in \{20, 40, 80, 160\}$. Each block corresponds to one training set size and shows four training examples. For each example, the top row presents SIS cues with $K \in \{50, 100, 400, 800, 1024\}$ retained pixels from left to right, and the bottom row shows the corresponding awakened memories. The mean SSIM between the awakened memories and the original training images is 0.168, 0.445, 0.676, 0.759, and 0.935 for $B = 20$; 0.293, 0.347, 0.689, 0.751, and 0.906 for $B = 40$; 0.321, 0.393, 0.534, 0.613, and 0.793 for $B = 80$; and 0.240, 0.262, 0.338, 0.417, and 0.616 for $B = 160$, respectively.

B.3.2 SIS Guidance

We further consider substantially sparser guidance using Sufficient Input Subsets (SIS). For each training image, we retain only a subset of pixels identified by SIS and use the resulting sparse image directly as guidance. We vary the number of retained pixels K to examine how much visual information is needed to awaken the corresponding training memory.

As shown in Fig. 17, the full training image is not required to awaken its corresponding memory. In many cases, recognizable training-image structure can already emerge from highly sparse SIS cues, and becomes clearer as more pixels are retained. These results show that memory awakening does not require a visually complete reference image. A small set of informative cues can be sufficient to access a much more complete training memory encoded in the model parameters.

B.4 Additional Results on Held-out Guidance

We provide additional results using held-out test images as guidance. These experiments examine whether unseen inputs can selectively awaken training memories across different training set sizes.

As shown in Fig. 18, held-out inputs can awaken distinct training memories across all four training set sizes. The awakened images often resemble specific training samples while remaining visually different from the held-out references. These additional results support the observation that memory awakening from unseen inputs accesses training-derived information rather than simply reconstructing the input.

B.5 Additional Results on Adversarial Guidance

We further examine memory awakening under adversarial perturbations of different strengths. For each example, we gradually increase the attack strength and use the resulting adversarial images as guidance. This allows us to study how the awakened memory changes as the perturbation progressively alters the model's prediction. For both attack sweeps, we use $\epsilon \in \{8, 12, 16, 20, 24, 32, 40, 48, 64, 74, 84, 94, 104, 114, 124, 131\}/255$.

As shown in Figs. 19 and 20, both PGD and FGSM can redirect memory awakening as the perturbation strength increases. In many cases, the awakened memory switches under relatively weak

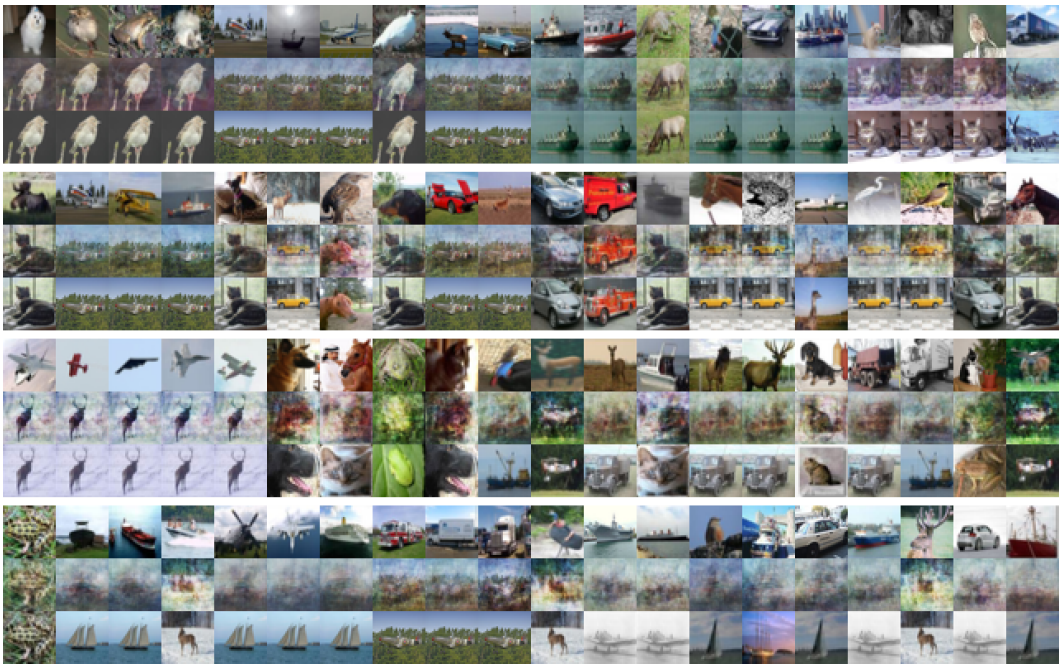

Figure 18: Additional memory awakening results guided by held-out test images. From top to bottom, the four groups correspond to $B = 20$, $B = 40$, $B = 80$, and $B = 160$. Within each group, the rows show the held-out reference images, awakened memories, and the nearest training samples to the awakened memories. The mean SSIM between the awakened memories and their nearest training samples is 0.642, 0.550, 0.507, and 0.483, respectively. For all four settings, the prediction of the awakened memory agrees with that of the reference image for all $10{,}000$ held-out samples ($\mathrm{PR} = \mathrm{PRec}$, 100%).

perturbations, before the guidance image undergoes a substantial visual change. The two attacks exhibit the same overall phenomenon, although the transition points and intermediate awakened memories can differ across attack methods. This suggests that memory awakening is sensitive to adversarially induced changes in model behavior rather than simply following the visual appearance of the perturbed input. From this perspective, adversarial attacks can be viewed as redirecting the training memory accessed through the model.

B.6 Training Dynamics Details

We further investigate the training dynamics of memory awakening from two perspectives: how it evolves across training epochs and how it changes with the rank retained in the low-rank decomposition.

B.6.1 Memory Awakening across Training Epochs

Fig. 21 examines how memory awakening evolves over training by fixing the starting checkpoint at epoch 0 and progressively moving the ending checkpoint to later epochs. With updates from only the early stages of training, the awakened images are dominated by irregular visual patterns and contain little recognizable object structure. As the ending epoch increases, coarse shapes and class-related visual structure begin to emerge, and later checkpoints recover increasingly clear object appearance and finer instance-level details. The transition is gradual: recognizable memories are not immediately present in the early parameter updates, but become progressively more accessible as training proceeds. This suggests that the memory structure exposed by awakening is progressively formed over the course of optimization.



Figure 19: Memory awakening under untargeted PGD attacks of increasing strength. From left to right, the perturbation bound ϵ increases from $8/255$ to $131/255$. PGD uses 10 steps with step size $\alpha = \epsilon/4$ and no random initialization, with only ϵ varying across columns. Each example is shown in three rows: adversarial guidance images, awakened memories, and the nearest training samples to the awakened memories. As the attack becomes stronger, the model prediction changes and the awakened memory often shifts accordingly.



Figure 20: Memory awakening under untargeted FGSM attacks of increasing strength. From left to right, the perturbation bound ϵ increases from $8/255$ to $131/255$. FGSM uses the cross-entropy objective, with only ϵ varying across columns. Each example is shown in three rows: adversarial guidance images, awakened memories, and the nearest training samples to the awakened memories. As the attack becomes stronger, the model prediction changes and the awakened memory often shifts accordingly.

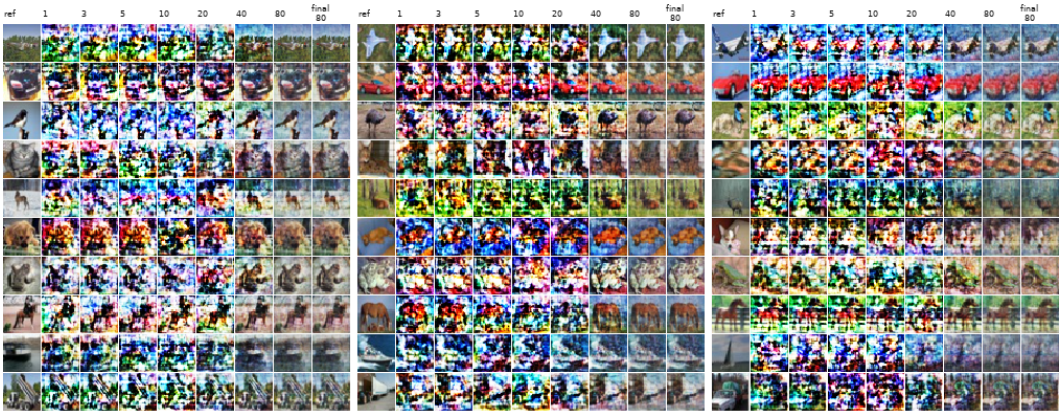

Figure 21: Evolution of memory awakening over training for $B = 160$. In each group, the leftmost column shows the reference image, followed by awakened images obtained using parameter updates ending at epochs $1, 3, 5, 10, 20, 40$, and 80. The starting checkpoint is fixed at epoch 0 for all recoveries. At early epochs, the awakened images are dominated by irregular visual patterns, while recognizable object structure gradually emerges as training proceeds, followed by increasingly clear appearance and finer instance-level details at later epochs.

B.6.2 Memory Awakening across Ranks

We next vary the retained rank while keeping the guidance image and training checkpoint fixed. As shown in Fig. 22, singular directions are included in descending order of singular value.

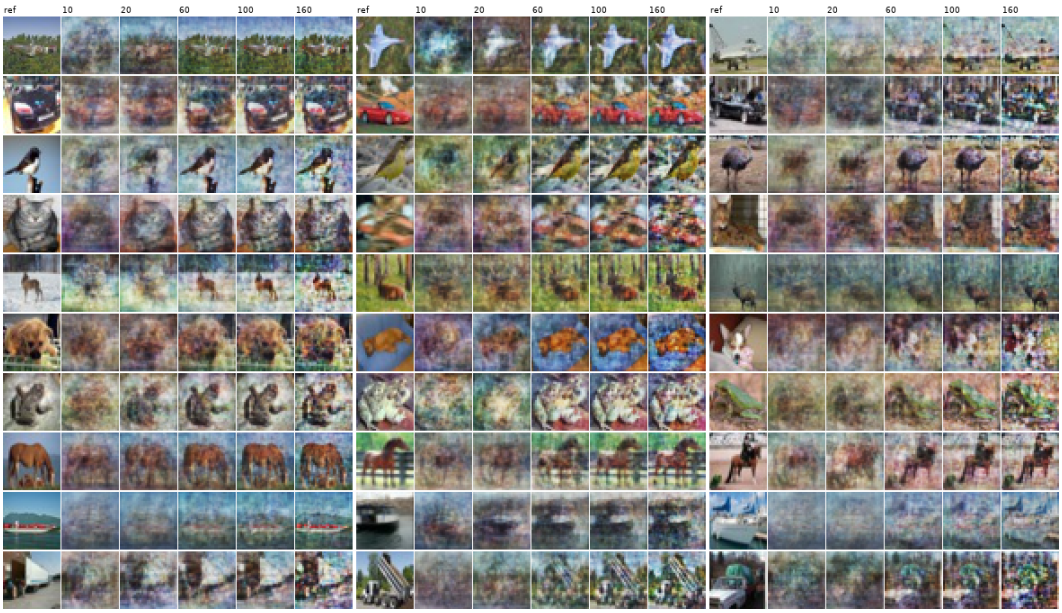

Figure 22: Effect of the retained rank on memory awakening for $B = 160$. In each group, the leftmost column shows the reference image, followed by awakened images with ranks $r = 10, 20, 60, 100$, and 160. The singular directions are retained in descending order of singular value, and $r = 160$ corresponds to the full rank of ΔW. At low ranks, the awakened images are blurry and mainly preserve large color regions and coarse object structure. As the rank increases, finer and more sample-specific details appear. When the full rank is used, however, the awakened images contain much more visible noise.

These results suggest that information is distributed unevenly across the singular directions of the parameter update. The dominant directions may mainly capture visual structure shared across training samples, which is sufficient to recover coarse image content. Additional directions appear to contribute finer, more sample-specific details. At larger ranks, the newly included weak singular directions may contain not only fine image details but also noisy or less stable components. This may explain why the awakened images become noisier near full rank.

B.7 OPS TRAINING

We further study memory awakening under one-pixel shortcut (OPS) training. Following the OPS setting, each class is assigned a class-specific shortcut defined by a fixed pixel location and pixel value. During training, this shortcut is inserted into every training image of the corresponding class. Prior work has shown that such a one-pixel cue can form a strong shortcut and redirect the model's prediction toward the class associated with that cue. Here, we use memory awakening to examine whether a shortcut that redirects the model's prediction also redirects the memory that can be awakened from the model parameters. For OPS training, we use SGD with a learning rate of 0.01.

Clean Training-Image Guidance. We first use clean training images as guidance, without adding the OPS pixel to the reference images. This allows us to examine whether the shortcut learned during training is reflected in the awakened memory even when the shortcut itself is absent from the guidance input. As shown in Fig. 23, the awakened memories recover the corresponding class-specific OPS pixels across different training set sizes.



Figure 23: Memory awakening from clean training-image guidance under OPS training. Results are shown for $B \in \{20, 40, 80\}$. For each setting, the first row shows the clean guidance images, the second row shows the corresponding awakened memories, and the third row shows the OPS-modified training images. Yellow circles indicate the class-specific shortcut-pixel locations. The shortcut pixels are absent from the clean guidance images but appear in the awakened memories at the same locations as in the OPS-modified training images.

As shown in Fig. 23, the shortcut pixel is not merely a training-time artifact but becomes part of the awakened training memory. Even when the guidance image is clean, the recovered memory consistently includes the class-specific OPS cue.

Held-out Guidance with Different OPS Targets. We next consider held-out guidance images and insert different class-specific OPS pixels into the same reference image. This setting examines whether a shortcut cue that changes the model's prediction also changes the memory awakened from the model parameters. As shown in Fig. 24, different OPS pixels applied to the same held-out image lead to substantially different awakened memories. The corresponding nearest training samples also change accordingly, showing that memory awakening shifts with the shortcut-induced change in model behavior.



Figure 24: Memory awakening from held-out guidance under OPS training. Results are shown for $B \in \{20, 40, 80\}$. For each example, the first row shows held-out guidance images with different class-specific OPS pixels inserted, the second row shows the corresponding awakened memories, and the third row shows their nearest training samples. Yellow circles indicate the inserted OPS-pixel locations. For the same held-out image, changing only the OPS pixel leads to different awakened memories and different nearest training samples.