Models Designed to Forget: Machine Unlearning via Key Deletion

Sonia Laguna[*], **Jorge da Silva Gonçalves, Moritz Vandenhirtz, Alain Ryser, Irene Cannistraci**[†], **Julia E. Vogt**[†]
Department of Computer Science, ETH Zurich, Switzerland

Abstract

Machine unlearning for vision models is rapidly becoming a practical requirement, driven by privacy regulations, data errors, and the need to remove harmful or corrupted training images. Despite this, most existing approximate unlearning methods tackle the problem from a post-hoc perspective. They attempt to erase the influence of targeted samples through parameter updates that typically require access to the full training data. This creates a mismatch with real deployment scenarios where unlearning requests can be anticipated, revealing a fundamental limitation of post-hoc approaches. We motivate *unlearning by design*, a novel paradigm for approximate methods in which models are directly trained to support forgetting as an inherent architectural capability. We instantiate this idea with Machine UNlearning via KEY deletion (MUNKEY), a memory-augmented transformer that decouples instance-specific memorization from model weights. Here, unlearning corresponds to removing the instance-identifying key, enabling zero-shot forgetting without weight updates or access to the original samples or labels. Across natural image benchmarks, fine-grained visual recognition, and medical datasets, MUNKEY outperforms all post-hoc baselines. Our results establish that *unlearning by design* enables fast, deployment-oriented unlearning while preserving predictive performance.

1 Introduction

As machine learning systems are increasingly deployed in high-stakes domains, the ability to *unlearn* (Cao & Yang, 2015; Bourtoule et al., 2021) specific training samples on demand has become essential. This necessity is driven both by the need to remove harmful, biased, or corrupted data and by the legal imperative to comply with privacy regulations, such as the European Union's GDPR "right to be forgotten" (Mantelero, 2013) and the California Consumer Privacy Act (California State Legislature, 2018). Because neural networks are capable of memorizing training samples (Arpit et al., 2017), the central challenge of machine unlearning is to surgically excise the influence of targeted data points. This process must eliminate the data, often verified via membership inference attacks (Shokri et al., 2017), without degrading the model's performance on the remaining "retained" and test data.

Current unlearning approaches generally fall into two categories: exact and approximate. Exact unlearning ensures that all influence of the target data is removed, mostly by retraining the model from scratch, a baseline that is often computationally prohibitive. To address this, approximate unlearning has emerged as a more scalable alternative, seeking to diminish the impact of forget-set data to a negligible level by updating the model's parameters. However, existing approximate methods (Nguyen et al., 2025) are strictly *post-hoc*: they rely on computationally intensive weight updates at the time of unlearning, such as gradient ascent, distillation, or saliency-based corrections, often requiring access to most of the data samples and corresponding labels.

Beyond computational cost and data availability, the reliance on post-hoc interventions highlights a fundamental structural flaw in standard architectures: knowledge entanglement. When a neural

[*]Correspondence to `slaguna@inf.ethz.ch`. [†]Shared senior authorship.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

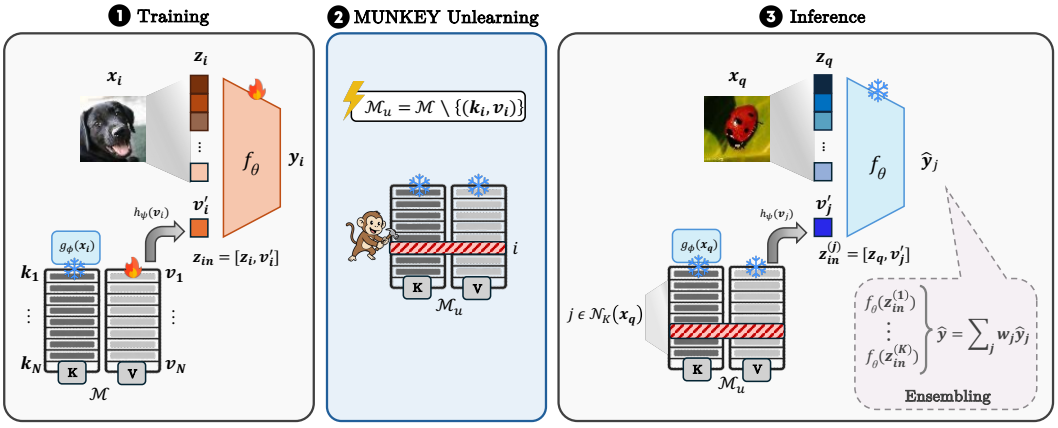


Figure 1: **Overview of MUNKEY. (1) Training:** The learnable exemplar token $\boldsymbol{v}_i$ from the memory bank $\mathcal{M}$ is concatenated to image tokens $\boldsymbol{z}_i$ to predict the label $\boldsymbol{y}_i$. **(2) Unlearning** results in a simple instance-specific key deletion in $\mathcal{M}$. **(3) Inference:** For a query x_q, the model retrieves K-nearest neighbor tokens from the updated memory $\mathcal{M}_u$ and ensembles their predictions for the final output.

network is trained, instance-specific details and generalizable features are fused into a monolithic set of static weights (Arpit et al., 2017), making it challenging and inefficient to remove the influence of a single instance *a posteriori*.

Can the paradigm be shifted towards *unlearning by design*? In many applications, such as clinical studies where patients withdraw consent or consumer platforms where users opt out of data sharing, deletion requests are an anticipated part of the model lifecycle. By accounting for these requests *a priori*, we can design models with unlearning in mind, where forgetting is streamlined and efficient.

Machine UNlearning via KEY deletion (MUNKEY) bridges this gap. We introduce MUNKEY, a vision method that externalizes image-specific information to separate it structurally from static model parameters. By associating each training instance with a learnable exemplar token, MUNKEY transforms unlearning from a complex optimization problem into a simple set-theoretic operation: deleting entries from an external memory. This method offers two primary advantages: *(i)* efficiency, enabling "zero-shot" near-instantaneous unlearning without weight updates, and *(ii)* privacy-centric deployment, requiring only instance identifiers rather than raw samples or labels. Crucially, decoupled memories have been studied to improve predictive performance, perception, or few-shot efficiency (M. Jia et al., 2022; Long et al., 2022; Sandler et al., 2022), while MUNKEY explicitly designs the memory for model adaptation and targeted data deletion. Our work builds on recent results showing that visual memories can scale to billions of samples (Geirhos et al., 2025). By repurposing memory-augmented structures for the "inverse" task of forgetting, we enable a new class of models that are not only high-performing but are natively compliant with the dynamic requirements of data privacy and integrity.

Finally, while a portion of the literature focuses on class-level removal (Z. Zhang et al., 2025; Gandikota et al., 2023), we target the general case of approximate instance-level unlearning as it more directly addresses individual privacy and consent. This work targets the vision domain, building upon the extensive body of literature (Ebrahimpour-Boroojeny et al., 2025; Hsu et al., 2025; Khalil et al., 2025; Fan et al., 2024; S. Lee et al., 2025) that focuses on unlearning for vision. Across benchmarks spanning natural, fine-grained, and medical imagery, MUNKEY outperforms state-of-the-art post-hoc baselines, offering a superior balance of efficiency and robustness against membership inference attacks without compromising utility. The decoupling of representation (backbone) from editable knowledge (memory) provides additional advantages, such as dynamic scalability, allowing new data to be incorporated by expanding the memory bank without backbone retraining (Li et al., 2024); or interpretability, where decisions are traceable to specific retrieved memory entries or "prototypes" (Geirhos et al., 2025).

Contributions. Our contributions are threefold: *(i) Paradigm Shift:* We propose *Unlearning by Design* in approximate methods, shifting from costly post-hoc corrections to intrinsic architectural modularity. By anticipating deletions during training, unlearning can be streamlined. *(ii) Algorithmic Framework:* We introduce MUNKEY, a vision unlearning method that leverages memory-augmented

models for instance-level forgetting by formalizing a way to decouple instance-specific information from model weights, which enables optimization-free unlearning. *(iii) Empirical Validation:* Extensive evaluations across natural, large-scale, and medical vision datasets show that MUNKEY outperforms state-of-the-art post-hoc baselines in efficiency, utility, and detectable membership signal.

2 Related Work

Machine Unlearning. Machine unlearning removes the influence of specific training samples from a trained model (Cao & Yang, 2015). While early work explores exact approaches with strong theoretical guarantees (Bourtoule et al., 2021), the prohibitive computational cost of these methods has shifted recent research to more efficient approximate alternatives (Nguyen et al., 2025).

A prominent direction in approximate unlearning leverages fine-tuning to induce catastrophic forgetting (CF) (McCloskey & Cohen, 1989). This concept has motivated more targeted strategies such as CF-k and Exact Unlearning-k (EU-k) (Goel et al., 2022), which restrict unlearning to the final k layers, as well as gradient ascent approaches such as NegGrad and NegGrad+ (Kurmanji et al., 2023), or supervising on held-out sets (Mirlach et al., 2026). To improve model utility, prior works have proposed selectively adjusting specific layers or parameters: ℓ_1-sparse unlearning (J. Jia et al., 2023) promotes sparsity via ℓ_1 regularization, while SSD (Foster et al., 2024) and SalUN (Fan et al., 2024) update only important parameters identified through Fisher information and gradient-based saliency, respectively. Meanwhile, other approaches focus on data manipulation and model distillation. These include utilizing label perturbation (L. Graves et al., 2021), adversarial examples (Ebrahimpour-Boroojeny et al., 2025; Cha et al., 2024), student-teacher methods such as Bad-Teacher (Chundawat et al., 2023a) and its later refinement SCRUB (Kurmanji et al., 2023), as well as recent contrastive-learning-based approaches via CoUn (Khalil et al., 2025).

Despite their differences, these methods share a limitation: they operate post-hoc, requiring access to the forget set, retain set, labels, or a combination thereof to modify a trained model. Approaches like SISA (Bourtoule et al., 2021) hint at unlearning by design by anticipating deletion through data partitioning, but remain exact methods and require optimization over affected shards upon deletion. Similarly, some approximate methods, including zero-shot (Chundawat et al., 2023b) approaches that remove the need for retain data, and sparsity-based methods that prepare models for later removal (Lin et al., 2023), still rely on post-hoc optimization. We instead advocate for a paradigm shift where approximate unlearning is anticipated and enforced during training. We propose a memory-augmented method designed for instance-level removal: by externalizing instance-specific information into a memory, our approach enables targeted removal without destabilizing the core model parameters.

External Memory. External memory mechanisms have been widely explored to augment neural networks with explicit storage and retrieval capabilities. Architectures such as Neural Turing Machines (A. Graves et al., 2014), Memory Networks (Weston et al., 2015; Sukhbaatar et al., 2015), and Memorizing Transformers (Y. Wu et al., 2022) traditionally rely on decoupling memory storage from the model's primary parameters. In parallel, complementary work focuses on associative memory, where learnable structures are embedded directly into network components to store patterns or knowledge (Sandler et al., 2022; M. Jia et al., 2022). However, while both utilize memory to enhance predictive performance (Li et al., 2024) or facilitate few-shot learning, we repurpose the external memory framework for a distinct objective: efficient data localization and removal. To achieve this, we introduce architectural and training choices that encourage the model to route deletable information through external storage rather than entangle it in the backbone. To the best of our knowledge, this is the first external-memory framework designed specifically for machine unlearning. An extended discussion of related work is presented in Appendix A.

3 Methods

3.1 Problem Formalization: Machine Unlearning

Let $f_\theta : \mathcal{X} \to \mathcal{Y}$ be a neural network parameterized by θ. The model is initially optimized over a training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ via Empirical Risk Minimization with a task-specific loss, in our case cross-entropy for the classification task. Machine unlearning considers a request to remove a subset of data $\mathcal{D}_f \subset \mathcal{D}$, the *forget set*, while preserving performance on $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$, the *retain set*. The gold standard is the *Retrain* model f_r, optimized exclusively on $\mathcal{D}_r$. The objective of

an approximate machine unlearning method is to achieve a behavior such that $f_{\theta_u} \approx f_r$ without the prohibitive cost of full retraining, ensuring both *Forgetting* (zero influence of $\mathcal{D}_f$) and *Utility* (high performance on $\mathcal{D}_r$ and on the test set $\mathcal{D}_t$). In this work, we employ the Vision Transformer (ViT) (Dosovitskiy et al., 2021) as our architectural backbone, followed by a linear classification head to form f_θ. A ViT partitions an input image x_i into a sequence of n flattened patches and projects them onto a latent space $\{\boldsymbol{z}_i^1, \ldots, \boldsymbol{z}_i^n\}$ through an embedding layer. These patches are then processed via attention layers. In this work, we will refer to f_θ as the model after the embedding layer.

3.2 MUNKEY: Machine UNlearning via KEY Deletion

We propose MUNKEY[1] to reformulate unlearning as a modular data-deletion task within a memory bank $\mathcal{M}$. Unlike traditional post-hoc approximate methods, MUNKEY is constructed for *unlearning by design* by conditioning classification on a joint sequence of image patch embeddings and retrieved exemplar tokens, as illustrated in Figure 1. This structural separation encourages the influence of a specific sample to be confined to its corresponding entry in $\mathcal{M}$, turning unlearning from an optimization problem over θ into a deletion operation over $\mathcal{M}$.

External Exemplar Memory. MUNKEY exploits the permutation-invariant modularity of transformers by augmenting the standard patch sequence with a decoupled external exemplar token. By injecting instance-specific knowledge directly into the input sequence rather than the static weights θ, the burden of memorization is shifted to the externalized memory, enabling unlearning via simple deletion of the memory entry. To operationalize this decoupling, an explicit exemplar memory bank $\mathcal{M} = \{(\boldsymbol{k}_i, \boldsymbol{v}_i)\}_{i=1}^N$ is defined for all $\boldsymbol{x}_i \in \mathcal{D}$. For each instance, $\boldsymbol{k}_i = g_\phi(\boldsymbol{x}_i) \in \mathbb{R}^{d_k}$ is a fixed key extracted via a frozen encoder g_ϕ, while $\boldsymbol{v}_i \in \mathbb{R}^{d_e}$ is a learnable exemplar token that is treated as a learnable parameter. g_ϕ can be instantiated with any suitable visual encoder. We evaluate both general-purpose pre-trained encoders, as well as encoders trained on same-distribution held-out data. The latter setting ensures that MUNKEY also operates under a controlled scenario where the encoder has not seen any samples that may later be forgotten, addressing cases where key-encoder leakage or overlap with general-purpose pre-training data is a concern. To ensure compatibility, an adapter layer h_ψ projects the exemplar token into the ViT's latent space, treating it as an auxiliary input token.

Training with Stochastic Pathway Dropout. To train MUNKEY, the image patch embeddings $\boldsymbol{z}_i$ are concatenated with the learnable projected exemplar token $\boldsymbol{v}_i' = h_\psi(\boldsymbol{v}_i)$ to form the input sequence $\boldsymbol{z}_{in}$. Subsequently, $\boldsymbol{z}_{in}$ is passed through f_θ to predict the class, thereby not requiring any adaptation of the standard ViT architecture. A fundamental challenge in such dual-stream processing is *modality dominance* (W. Wang et al., 2020; Hussen Abdelaziz et al., 2020), where the model over-relies on the most predictive input. In our case, depending on the dataset nature, the backbone can over-rely on high-capacity exemplar tokens and fail to learn robust features from the raw image patches, or vice versa.

To enforce a balanced information flow and prevent pathway dominance, a stochastic pathway dropout is applied to both image patch embeddings, $\boldsymbol{z}_i$, and projected exemplar tokens, $\boldsymbol{v}_i'$. A stochastic masking vector $\mathbf{m} = [\gamma_{img}, \gamma_{tok}] \in \{0,1\}^2$ follows a categorical distribution, where $P(\mathbf{m} = [0,1]) = p_i$, $P(\mathbf{m} = [1,0]) = p_t$, $P(\mathbf{m} = [1,1]) = 1 - p_i - p_t$, and $P(\mathbf{m} = [0,0]) = 0$, with p_i and p_t being the probabilities for image and token dropout, respectively. Specifically, the dropout operator $\mathcal{P}$ is formulated for each pathway using the sampled mask component γ as: $\mathcal{P}(\boldsymbol{u}, \emptyset, \gamma) = \gamma \boldsymbol{u} + (1 - \gamma)\emptyset$, for any input token $\boldsymbol{u}$ and a pathway-specific globally learned null token $\emptyset$, i.e. $\emptyset_{img}$ or $\emptyset_{tok}$.

Inference via Nearest Neighbors. At inference time, the specific exemplar token $\boldsymbol{v}_q$ for an unseen query image $\boldsymbol{x}_q$ is unavailable. To resolve this, the query key $\boldsymbol{k}_q = g_\phi(\boldsymbol{x}_q)$ is first extracted from the frozen encoder. Subsequently, the K nearest neighbors $\mathcal{N}_K(\boldsymbol{x}_q)$ are retrieved from the memory bank $\mathcal{M}$ based on the cosine similarity of $\boldsymbol{k}_q$ with the stored keys $\{\boldsymbol{k}_i\}_{i=1}^N$. Note that following the efficiency paradigms of large-scale retrieval-augmented models (Geirhos et al., 2025), this operation can be scaled efficiently through approximate nearest neighbor indexing, i.e. ScaNN (R. Guo et al., 2020). The final prediction is an ensemble of the model's output for each of the neighbors. More concretely, for each retrieved exemplar token $\{\boldsymbol{v}_j\}_{j \in \mathcal{N}_K(\boldsymbol{x}_q)}$, an input sequence $\boldsymbol{z}_{in}^{(j)}$ as $[\boldsymbol{z}_q, \boldsymbol{v}_j']$ is constructed. The final output $\hat{\boldsymbol{y}}$ is computed as a weighted average of the logits: $\hat{\boldsymbol{y}} = \sum_{j \in \mathcal{N}_K(\boldsymbol{x}_q)} w_j$.

[1]The code is publicly available at `https://github.com/sonialagunac/MUNKEY`.

$f_\theta\big(\boldsymbol{z}_{in}^{(j)}\big)$, where the weights w_j are determined by a softmax over key similarities: $w_j = \frac{\exp(\text{sim}(\boldsymbol{k}_q, \boldsymbol{k}_j)/\tau)}{\sum_{l \in \mathcal{N}_K(\boldsymbol{x}_q)} \exp(\text{sim}(\boldsymbol{k}_q, \boldsymbol{k}_l)/\tau)}$. Here, $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ a temperature parameter.

Retrieval-based Regularization at Training. To ensure the model generalizes to retrieved context at inference, where the specific $\boldsymbol{v}_q$ for a test sample is absent, this state is simulated during training. For each training sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, a fixed key $\boldsymbol{k}_i = g_\phi(\boldsymbol{x}_i)$ is extracted using the frozen encoder. These keys populate the memory $\mathcal{M}$ and serve as the basis for neighborhood discovery. To construct the augmented input sequence $\boldsymbol{z}_{in}$, either the instance-specific token $\boldsymbol{v}_i'$, or the neighboring $\boldsymbol{v}_n'$ are selected. $\boldsymbol{v}_n'$ is the result of simulating the inference state by retrieving a randomly varying number K' of neighbor tokens $\{\boldsymbol{v}_j\}_{j \in \mathcal{N}_{K'}(\boldsymbol{x}_i)}$ from $\mathcal{M} \setminus \{(\boldsymbol{k}_i, \boldsymbol{v}_i)\}$ and aggregating them via $\text{Agg}(\cdot)$ into $\boldsymbol{v}_n$, where $\text{Agg}(\cdot)$ stands for a similarity-weighted average operation. Let $r \sim \text{Bernoulli}(p_r)$ be an indicator variable for the retrieval state. Combined with the masking vector, the final input sequence at training time $\boldsymbol{z}_{in}$ is defined as:

$$\boldsymbol{z}_{in}(\boldsymbol{x}_i, \mathbf{m}, r) = \left[\mathcal{P}(\boldsymbol{z}_i, \emptyset_{img}, \gamma_{img}),\ \mathcal{P}\big((1-r)\boldsymbol{v}_i' + r\boldsymbol{v}_n', \emptyset_{tok}, \gamma_{tok}\big)\right]. \tag{1}$$

The model parameters $\{\theta, \psi\}$, and exemplar tokens $\{\boldsymbol{v}_i\}_{i=1}^N$ are optimized jointly by minimizing the expected masked training objective:

$$\min_{\theta, \psi, \{\boldsymbol{v}_i\}_{i=1}^N} \mathbb{E}_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \sim \mathcal{D}}\, \mathbb{E}_{\mathbf{m} \sim p(\mathbf{m}),\, r \sim \text{Bern}(p_r)} \left[\ell\left(f_\theta(\boldsymbol{z}_{in}(\boldsymbol{x}_i, \mathbf{m}, r)), \boldsymbol{y}_i\right)\right], \tag{2}$$

where the gradient $\nabla_{\boldsymbol{v}_j} \ell$ is clipped to 0 for all $j \in \mathcal{N}_{K'}(\boldsymbol{x}_i)$ when the retrieval branch is active. This ensures the backbone learns to aggregate context, preventing neighboring tokens from drifting to accommodate sample-specific gradients. This objective can be viewed as a variational pathway-masking formulation: $\mathbf{m}$ and r induce a distribution over information pathways, and training minimizes the expected risk under it. By requiring stable predictions when either pathway is partially suppressed, stochastic masking discourages brittle reliance on a single source. Consequently, it imposes an implicit information bottleneck on the exemplar pathway, pushing $\boldsymbol{v}_i$ to encode complementary information rather than act as an unconstrained lookup. We note that, as in approximate unlearning, this decoupling should be interpreted as an architectural inductive bias rather than a strict formal guarantee. Following standard practice in approximate methods, we evaluate forgetting empirically using membership inference attacks as a proxy for residual membership signal.

Unlearning. Unlike traditional methods that rely on iterative optimization to erase information, unlearning in MUNKEY is reduced to a simple set-theoretic operation: $\mathcal{M}_u = \mathcal{M} \setminus \{(\boldsymbol{k}_i, \boldsymbol{v}_i) \mid (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}_f\}$. By removing the forget set entries from the memory bank, the model is rendered incapable of retrieving the specific tokens v_i associated with the requested data. As the model is trained to rely only on the externalized memory for instance-specific details, this deletion results in "zero-shot" forgetting. All the steps are summarized in Algorithm 1, Appendix B.1.

Pathway Sensitivity and Model Selection. To prevent *pathway collapse*, where the model either ignores the memory (poor unlearning) or degenerates into a lookup table (poor generalization), we introduce the Pathway Sensitivity Score (P_s). P_s quantifies the balance of information flow by evaluating performance when one pathway is replaced by the learned null tokens $\emptyset$. We define the modality-specific accuracies $\mathcal{A}_{img}$ and $\mathcal{A}_{tok}$ by evaluating the model on the ablated input sequences $\boldsymbol{z}_{in}^{(img)} = [\boldsymbol{z}_i, \emptyset_{tok}]$ and $\boldsymbol{z}_{in}^{(tok)} = [\emptyset_{img}, \boldsymbol{v}_i']$, respectively, and $\mathcal{A}_{both}$ as the standard joint accuracy using the complete sequence $\boldsymbol{z}_{in} = [\boldsymbol{z}_i, \boldsymbol{v}_i']$. The sensitivity score is then formulated as $P_s = \frac{|\mathcal{A}_{img} - \mathcal{A}_{tok}|}{\mathcal{A}_{both} + \epsilon}$, where ϵ is a small constant added for numerical stability.

By utilizing the global null tokens $\emptyset$ learned during training, the sensitivity evaluation remains within the model's learned distribution. We define ξ as a threshold that bounds the disparity between pathway contributions, preventing dominance. P_s enables multi-stage hyperparameter selection for p_i and p_t: (i) a "health check" filters configurations where $P_s \geq \xi$, and (ii) following the "rewind" procedure proposed in Kurmanji et al. (2023), we prioritize models that minimize the validation utility gap, Δ_{val}, i.e., the accuracy difference between validation and disjoint proxy forget sets. We refer to Appendix E for extended details. Therefore, among candidate hyperparameter configurations $\lambda = (p_i, p_t)$, we select

$$\lambda^* = \arg\min_\lambda\ \Delta_{\text{val}}(\lambda) \quad \text{s.t.} \quad P_s(\lambda) < \xi. \tag{3}$$

4 Experimental Setup

Datasets. We evaluate MUNKEY across natural and medical imaging benchmarks. For natural images, we utilize CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and large-scale Tiny ImageNet (Deng et al., 2009). For the medical domain, we incorporate three datasets from the high-resolution MedMNIST v2 (Yang et al., 2023): BloodMNIST (Acevedo et al., 2020), with 17,092 images across 8 classes for fine-grained cellular morphology; PathMNIST (Kather et al., 2019), a large-scale histology dataset with 107,180 images and 9 classes; and DermaMNIST (Tschandl et al., 2018), with 10,015 images across 7 imbalanced skin-disease classes, with multi-source noise, mimicking real-world clinical prevalence where common conditions outnumber rare pathologies. Due to its difficulty, DermaMNIST is featured as a primary benchmark alongside CIFAR-10, while evaluations for the remaining datasets are provided in Appendix C and D.1. Dataset details are included in Appendix B.2.

Baselines. We compare against a diverse suite of eleven baselines whose mechanisms, while originally designed for CNNs, apply directly to Transformers. Recognizing ViTs as the current gold standard for vision, we implement both MUNKEY and all baselines using a ViT-Tiny backbone to ensure a fair comparison and to bridge the reporting gap in transformer-based unlearning. Additional model scales, e.g., ViT-Base, are evaluated in Appendix D.4. We include: *Retrain from Scratch* with $\mathcal{D}_r$, the oracle for exact unlearning on both the standard ViT-Tiny, referred to as *Retrain (T)*, and the MUNKEY-specific model, *Retrain (M)*. Catastrophic Forgetting (*CF*), *CF-k*, and *EU-k* (Goel et al., 2022) fine-tune the model on $\mathcal{D}_r$ while freezing or restoring parts of the network. Similarly, *NegGrad+* (Kurmanji et al., 2023) and *Amnesiac* (Foster et al., 2024), utilize gradient ascent and label noise to induce forgetting. To represent weight-level manipulation, we include ℓ_1*-sparse* as in J. Jia et al. (2023), encouraging parameter sparsity. For distillation-based frameworks, we use *SCRUB* (Kurmanji et al., 2023). Moreover, optimization-free parameter dampening is represented by *SSD* (Foster et al., 2024), which utilizes Fisher Information to identify and damp relevant parameters without iterative training. We also evaluate *AMUN* (Ebrahimpour-Boroojeny et al., 2025), which generates adversarial samples to force forgetting, and CoUn (Khalil et al., 2025), a contrastive-learning-based method. Finally, we include a memory-based reference using *KNN* (Fix, 1985), as a point of comparison for retrieval-based inference. Further implementation details are provided in Appendix B.3.

Metrics. Following prior work (Patel & Qiu, 2025; Fan et al., 2024; Ebrahimpour-Boroojeny et al., 2025), we evaluate our model using four primary metrics and an average gap to assess the trade-off between utility and forgetting: Test Accuracy (TA) and Retain Accuracy (RA) for model utility, and Forget Accuracy (FA) and Membership Inference Attack (MIA) to measure forgetting. For privacy evaluation, we use the SOTA Robust MIA (Zarifzadeh et al., 2024). This attack compares the confidence score of a sample against reference distributions obtained from 64 shadow models. As in prior work (Ebrahimpour-Boroojeny et al., 2025), we compute the confidence values using the Soft-Margin Taylor expansion of the Softmax with temperature $T = 2$ and a ratio threshold γ of 1. We report the AUC for distinguishing training samples from unseen samples. An MIA AUC close to 0.5 on the forget set $\mathcal{D}_f$ indicates that the unlearned model does not retain a detectable membership signal. We aggregate these results into the Average Gap (Avg Gap), which measures the mean absolute difference between each of these four metrics and their respective counterparts in the "gold-standard" oracles. For architectural fairness, baselines are compared against a retrained ViT-Tiny, *Retrain (T)*, while our method is compared against a retrained MUNKEY, *Retrain (M)*.

Implementation Details. We evaluate MUNKEY under two forgetting regimes: a standard 10% removal of random samples benchmark and a more conservative 2%, which we posit represents a more realistic scenario where data deletion requests are frequent but scarce. More extreme 50% removal is shown in Appendix C. We additionally evaluate full-class, class-imbalanced, and single-instance deletion requests on CIFAR-10 in Appendix C. To maintain label distribution and prevent the removal of entire classes, mainly in imbalanced datasets or with a high number of classes, we utilize stratified random sampling of the forget sets for all datasets except CIFAR-10. All experiments are conducted over three independent seeds. The training pipeline and dynamics are standardized across all base models and datasets, and further details are reported in Appendix B.4. We provide ablations of all different subcomponents of MUNKEY in Appendix D to isolate the effect of each one of them on the final performance. In particular, we study the trade-off between performance and dimension d_e of the exemplar token ($\boldsymbol{v}_i \in \mathbb{R}^{d_e}$), the effect of the number of neighbors K used at inference time, or the choice of key encoder. Beyond unlearning quality and runtime, there are

system-level considerations associated with storing and querying the external memory bank. We discuss these practical costs, including memory requirements and retrieval overhead, in Appendix B.5.

5 Results

To validate the proposed *unlearning by design*, we evaluate MUNKEY across visual domains, addressing three questions: *(i) Decoupled Memorization:* Do instance-specific tokens enable zero-shot unlearning? *(ii) Information Encoding:* Does the memory capture granular signals beyond class labels in representations? *(iii) Architectural Interaction:* How does pathway aggregation impact performance?

Table 1: Comparison of machine unlearning performance across random forget rates (10% and 2%) on DermaMNIST and CIFAR-10. We present the accuracy on the test (TA), retain (RA), and forget (FA) sets, as well as the membership inference attack (MIA) and average gap values. Across all tables in the manuscript, when applicable, we **bold** the lowest Avg Gap and underline the second lowest.

	Random Forget (10%)					Random Forget (2%)				
Method	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**
DermaMNIST										
Retrain (T)	$76.46_{\pm 0.22}$	$100.0_{\pm 0.00}$	$74.86_{\pm 1.77}$	$48.88_{\pm 0.69}$	–	$76.49_{\pm 0.26}$	$100.0_{\pm 0.00}$	$75.71_{\pm 2.02}$	$50.72_{\pm 1.60}$	–
Retrain (M)	$81.25_{\pm 0.66}$	$99.70_{\pm 0.08}$	$80.29_{\pm 1.41}$	$50.04_{\pm 1.05}$	–	$81.80_{\pm 0.04}$	$99.75_{\pm 0.05}$	$81.43_{\pm 2.10}$	$49.36_{\pm 2.86}$	–
KNN	$74.60_{\pm 0.24}$	$85.54_{\pm 0.14}$	$74.81_{\pm 1.23}$	$49.94_{\pm 0.90}$	–	$75.01_{\pm 0.07}$	$85.84_{\pm 0.09}$	$74.52_{\pm 0.67}$	$50.32_{\pm 3.30}$	–
CF	$75.28_{\pm 0.98}$	$97.50_{\pm 1.55}$	$94.48_{\pm 2.23}$	$64.41_{\pm 1.96}$	$9.71_{\pm 0.99}$	$76.38_{\pm 0.28}$	$97.50_{\pm 1.16}$	$95.48_{\pm 1.21}$	$61.20_{\pm 1.80}$	$8.22_{\pm 0.90}$
CF-k	$76.94_{\pm 0.42}$	$99.86_{\pm 0.13}$	$99.90_{\pm 0.13}$	$67.10_{\pm 3.73}$	$10.97_{\pm 1.05}$	$76.96_{\pm 0.43}$	$99.86_{\pm 0.13}$	$100.0_{\pm 0.00}$	$66.73_{\pm 1.38}$	$10.23_{\pm 0.74}$
EU-k	$76.89_{\pm 0.58}$	$99.60_{\pm 0.23}$	$99.52_{\pm 0.18}$	$69.27_{\pm 0.23}$	$11.47_{\pm 0.51}$	$76.97_{\pm 0.47}$	$99.63_{\pm 0.22}$	$99.76_{\pm 0.34}$	$68.54_{\pm 1.16}$	$10.68_{\pm 0.73}$
ℓ_1-sparse	$75.81_{\pm 0.45}$	$92.36_{\pm 1.80}$	$86.81_{\pm 1.17}$	$57.69_{\pm 1.13}$	$7.26_{\pm 0.78}$	$75.61_{\pm 0.67}$	$89.46_{\pm 3.01}$	$80.24_{\pm 1.21}$	$54.02_{\pm 0.48}$	$4.81_{\pm 1.06}$
NegGrad+	$76.26_{\pm 0.37}$	$96.31_{\pm 1.03}$	$83.48_{\pm 1.25}$	$55.83_{\pm 1.14}$	$4.87_{\pm 0.69}$	$75.73_{\pm 0.96}$	$98.08_{\pm 0.42}$	$74.05_{\pm 2.99}$	$48.63_{\pm 1.79}$	$1.61_{\pm 1.12}$
Amnesiac	$75.56_{\pm 0.23}$	$98.30_{\pm 0.85}$	$82.00_{\pm 2.38}$	$47.01_{\pm 1.11}$	$2.93_{\pm 0.73}$	$75.89_{\pm 0.29}$	$98.92_{\pm 0.17}$	$69.76_{\pm 1.47}$	$36.59_{\pm 1.82}$	$5.44_{\pm 0.88}$
SCRUB	$73.97_{\pm 1.81}$	$76.02_{\pm 2.48}$	$74.68_{\pm 1.23}$	$49.62_{\pm 0.25}$	$6.85_{\pm 0.96}$	$76.38_{\pm 0.76}$	$99.55_{\pm 0.53}$	$86.94_{\pm 1.70}$	$57.51_{\pm 1.00}$	$4.65_{\pm 0.85}$
SSD	$75.07_{\pm 2.08}$	$92.07_{\pm 10.2}$	$89.73_{\pm 12.9}$	$63.29_{\pm 8.28}$	$9.65_{\pm 4.67}$	$72.58_{\pm 5.56}$	$87.87_{\pm 16.3}$	$78.52_{\pm 30.3}$	$60.98_{\pm 9.33}$	$7.28_{\pm 9.07}$
AMUN	$76.46_{\pm 0.69}$	$98.33_{\pm 0.94}$	$95.05_{\pm 0.64}$	$63.70_{\pm 0.27}$	$9.17_{\pm 0.59}$	$76.46_{\pm 0.69}$	$97.99_{\pm 0.91}$	$98.57_{\pm 1.01}$	$65.95_{\pm 1.01}$	$10.03_{\pm 0.79}$
CoUn	$79.50_{\pm 0.35}$	$99.80_{\pm 0.05}$	$87.24_{\pm 1.33}$	$54.92_{\pm 1.29}$	$5.42_{\pm 0.67}$	$79.50_{\pm 0.61}$	$99.81_{\pm 0.06}$	$89.52_{\pm 1.47}$	$55.11_{\pm 2.04}$	$5.35_{\pm 0.92}$
MUNKEY	$81.06_{\pm 0.44}$	$99.71_{\pm 0.08}$	$84.00_{\pm 1.22}$	$51.19_{\pm 1.19}$	$\mathbf{1.27}_{\pm 0.64}$	$81.23_{\pm 0.53}$	$99.71_{\pm 0.10}$	$85.48_{\pm 2.76}$	$50.96_{\pm 2.97}$	$\mathbf{1.57}_{\pm 1.22}$
CIFAR-10										
Retrain (T)	$78.13_{\pm 0.64}$	$91.62_{\pm 0.55}$	$77.57_{\pm 0.83}$	$49.77_{\pm 0.32}$	–	$78.87_{\pm 1.02}$	$91.83_{\pm 0.61}$	$79.60_{\pm 0.86}$	$50.30_{\pm 0.78}$	–
Retrain (M)	$92.38_{\pm 0.06}$	$99.93_{\pm 0.03}$	$92.05_{\pm 0.23}$	$49.90_{\pm 0.34}$	–	$92.59_{\pm 0.20}$	$99.93_{\pm 0.02}$	$92.83_{\pm 0.54}$	$50.67_{\pm 2.02}$	–
KNN	$91.06_{\pm 0.05}$	$94.56_{\pm 0.06}$	$91.02_{\pm 0.13}$	$50.16_{\pm 0.10}$	–	$91.28_{\pm 0.08}$	$94.58_{\pm 0.02}$	$92.10_{\pm 0.28}$	$51.19_{\pm 0.66}$	–
CF	$77.66_{\pm 0.50}$	$91.74_{\pm 0.48}$	$87.25_{\pm 1.04}$	$55.86_{\pm 0.16}$	$4.09_{\pm 0.44}$	$78.53_{\pm 0.95}$	$91.97_{\pm 0.91}$	$87.10_{\pm 0.59}$	$56.38_{\pm 1.35}$	$3.52_{\pm 0.65}$
CF-k	$79.60_{\pm 0.63}$	$94.30_{\pm 0.55}$	$94.16_{\pm 0.61}$	$58.93_{\pm 0.40}$	$7.48_{\pm 0.41}$	$79.53_{\pm 0.60}$	$94.30_{\pm 0.55}$	$94.53_{\pm 0.29}$	$59.51_{\pm 0.71}$	$6.82_{\pm 0.50}$
EU-k	$79.44_{\pm 0.72}$	$94.22_{\pm 0.55}$	$94.00_{\pm 0.65}$	$58.87_{\pm 0.30}$	$7.36_{\pm 0.42}$	$79.36_{\pm 0.64}$	$94.20_{\pm 0.58}$	$94.63_{\pm 0.12}$	$59.16_{\pm 0.69}$	$6.69_{\pm 0.50}$
ℓ_1-sparse	$52.80_{\pm 0.55}$	$53.47_{\pm 0.75}$	$52.97_{\pm 0.86}$	$50.79_{\pm 0.79}$	$22.28_{\pm 0.48}$	$52.70_{\pm 0.73}$	$53.79_{\pm 0.89}$	$53.70_{\pm 2.00}$	$50.70_{\pm 0.77}$	$22.63_{\pm 0.74}$
NegGrad+	$77.29_{\pm 0.27}$	$91.08_{\pm 0.42}$	$83.13_{\pm 0.89}$	$53.65_{\pm 0.32}$	$2.71_{\pm 0.41}$	$77.03_{\pm 0.48}$	$90.72_{\pm 0.74}$	$72.03_{\pm 1.06}$	$48.62_{\pm 0.75}$	$3.05_{\pm 0.57}$
Amnesiac	$78.31_{\pm 0.35}$	$92.17_{\pm 0.75}$	$85.67_{\pm 1.05}$	$52.51_{\pm 0.21}$	$2.89_{\pm 0.46}$	$77.77_{\pm 0.29}$	$91.11_{\pm 0.68}$	$76.97_{\pm 1.21}$	$44.95_{\pm 0.86}$	$2.45_{\pm 0.59}$
SCRUB	$76.51_{\pm 0.78}$	$85.31_{\pm 0.71}$	$83.72_{\pm 0.94}$	$54.18_{\pm 0.47}$	$4.62_{\pm 0.48}$	$77.74_{\pm 1.19}$	$89.83_{\pm 1.11}$	$85.61_{\pm 0.84}$	$56.03_{\pm 1.31}$	$3.72_{\pm 0.70}$
SSD	$78.54_{\pm 1.08}$	$93.29_{\pm 0.71}$	$92.84_{\pm 0.96}$	$58.91_{\pm 0.43}$	$6.62_{\pm 0.52}$	$79.02_{\pm 1.06}$	$93.63_{\pm 0.62}$	$94.00_{\pm 0.22}$	$58.75_{\pm 0.78}$	$6.20_{\pm 0.55}$
AMUN	$77.71_{\pm 0.46}$	$91.25_{\pm 0.56}$	$86.85_{\pm 3.46}$	$55.90_{\pm 1.94}$	$4.05_{\pm 1.05}$	$77.71_{\pm 0.46}$	$90.89_{\pm 0.94}$	$87.27_{\pm 3.74}$	$56.37_{\pm 2.08}$	$3.96_{\pm 1.18}$
CoUn	$80.68_{\pm 0.44}$	$95.72_{\pm 0.01}$	$91.32_{\pm 0.54}$	$55.75_{\pm 0.34}$	$6.60_{\pm 0.36}$	$81.01_{\pm 0.53}$	$95.84_{\pm 0.01}$	$92.03_{\pm 0.17}$	$55.16_{\pm 0.43}$	$5.86_{\pm 0.45}$
MUNKEY	$92.63_{\pm 0.20}$	$99.94_{\pm 0.01}$	$92.53_{\pm 0.52}$	$51.40_{\pm 0.44}$	$\mathbf{0.56}_{\pm 0.21}$	$92.69_{\pm 0.21}$	$99.94_{\pm 0.01}$	$93.57_{\pm 0.26}$	$50.05_{\pm 2.13}$	$\mathbf{0.38}_{\pm 0.75}$

Decoupled Memorization: Forgetting and Utility Results. Table 1 reports unlearning performance in two datasets across two random forget rates, 10% and 2%. Across all evaluations, MUNKEY consistently achieves the lowest Avg Gap, which quantifies the trade-off between model utility and unlearning faithfulness. Results on four additional datasets, full-class and class-imbalanced forget sets, and larger forget rates, i.e., 50%, are provided in Appendix C, confirming that these performance improvements generalize to diverse tasks. We include CIFAR-10 and DermaMNIST in our primary analysis to present two distinct trends. On CIFAR-10, where visual distributions align closely with the ImageNet-based features of our frozen key encoder, MUNKEY achieves predictive accuracies significantly higher than all post-hoc baselines. These results demonstrate that when the key encoder captures the underlying data manifold effectively, we can leverage these features for superior predictive performance while maintaining robust "black-box" unlearning, as evidenced by the MIA scores. We emphasize that the key encoder does not constitute an unfair advantage as each method's unlearning performance is evaluated against its respective gold standard, i.e., Retrain (T) or Retrain (M). Shifting to DermaMNIST, where medical morphologies differ significantly from the pre-training data of the key encoder, we still outperform baselines, not only in Avg Gap, but also in predictive performance. This indicates that the method also prevails in settings where the key encoder's pre-training does not align with the dataset. Overall, high forget accuracy does not necessarily indicate unsuccessful unlearning, as forgotten samples can still be classified through features shared with retained data, just as unseen test samples. Indeed, Retrain (M), which never

trains on the forget set, also achieves high forget accuracy, while MUNKEY remains close to its oracle in both forget accuracy and membership inference.

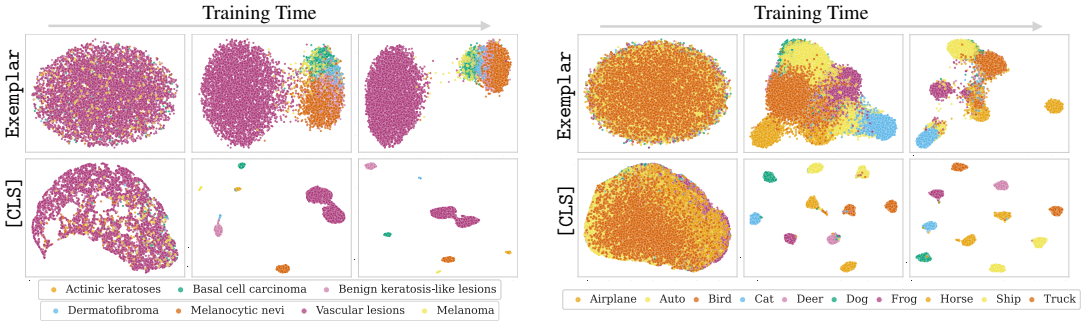
To isolate the contribution of the pre-trained key encoder, Appendix D.2 includes a controlled setting where the encoder is trained on a same-dataset held-out 20% split. This highlights a flexible design choice: when a pre-trained encoder is available, it can be directly leveraged; otherwise, or if a potential overlap between encoder and forgotten samples is a concern, an encoder trained on disjoint data remains sufficient. We further explore the effect of other key encoders, showing that MUNKEY remains effective under alternative self-supervised (Oquab et al., 2024) and multimodal (Radford et al., 2021) pre-trained models.

In both datasets, while pure retrieval classifiers, i.e., KNN, achieve good MIA scores, their predictive performance is lower. This disparity shows that MUNKEY's image pathway boosts generalization beyond simple memory retrieval, a critical requirement for applications where both high utility and reliable unlearning are paramount. Beyond our results, we observe performance variability among baselines. While recent contrastive, distillation, and adversarial-based methods are generally competitive, they are occasionally outperformed by simpler gradient ascent (NegGrad+) or label-noising strategies. These fluctuations suggest that unlearning techniques originally optimized for CNNs may not seamlessly transfer to ViTs, highlighting a need for more specialized research in unlearning within transformer-based architectures. We further compare against a RAC-inspired (Long et al., 2022) retrieval-fusion baseline in Appendix D.7, showing that an editable retrieval component alone does not achieve comparable unlearning performance without deletion-oriented training.

Beyond end-to-end MIA, Appendix D.5 studies the membership signal of the isolated backbone by removing the exemplar pathway at evaluation time. Predictive performance drops, confirming that the memory pathway contributes to utility, while the membership inference signal remains close to random (≈ 0.5), indicating low detectable instance-specific leakage in the backbone alone. Finally, a core advantage of the *unlearning by design* paradigm is the elimination of iterative post-hoc optimization. Table 2 shows runtime efficiency across methods, with the detailed per-method timings included in Table 4. Post-hoc baselines span a wide unlearning runtime range, where even the fastest remains slower than MUNKEY, that requires O(1) computations, hence negligible time ($\approx$ 0s). Forgetting is reduced to simply dropping the forgotten samples' key-value pairs from the memory bank, rendering the process near-instantaneous even at scale. We also report the per-epoch training time for both standard ViTs and MUNKEY. The results indicate that the overhead of maintaining and learning the exemplar memory bank is minimal. Given that this specialized training occurs only once in a compute-rich environment, the marginal increase in training time is a negligible cost for the permanent benefit of zero-shot unlearning throughout the model's deployment lifecycle. In Appendix B.5 we include an overview of system requirements, i.e., additional memory cost.

Table 2: Runtime efficiency, full timings in Table 4. *Unlearn* is post-training deletion time. *Train/Ep.* denotes per-epoch training time for the base models.

Method	Cost	DermaMNIST	CIFAR-10
Standard ViT	Train/Ep.	24.7s	174.6s
MUNKEY	Train/Ep.	25.9s	178.5s
Fastest baseline	Unlearn	23.1s	105.8s
Slowest baseline	Unlearn	313.4s	1940.5s
MUNKEY	Unlearn	$\approx$ **0s**	$\approx$ **0s**



(a) DermaMNIST (b) CIFAR-10

Figure 2: UMAP visualization of learned exemplar tokens in the memory and $[CLS]$ tokens at epochs 0, 50, and 100 of training. Left: DermaMNIST. Right: CIFAR-10.

Information Encoded in the Memory. We investigate whether the learned exemplar tokens capture instance-specific information beyond generic class labels by analyzing the representation space of the memory. Figure 2 contrasts the 2D UMAP (McInnes et al., 2018) projections of the Exemplar Memory Bank $\{\boldsymbol{v}_i\}_{i=1}^{N}$ with those of the $[CLS]$ tokens. The $[CLS]$ tokens are extracted from the model backbone's output for each training point and serve as features for classification, therefore capturing a class-guided baseline representation. We observe two distinct patterns consistent with our quantitative results: In CIFAR-10, where the key encoder's pre-trained features are highly relevant to the class, the memory bank exhibits significant class-based clustering, albeit substantially weaker than the $[CLS]$. Conversely, in DermaMNIST, the memory bank representations are more diffuse, reflecting the distribution shift from the encoder's pre-training data. As shown in our temporal visualizations across three training stages, the $[CLS]$ tokens rapidly form tight class-specific clusters to minimize the objective loss. In contrast, the exemplar tokens evolve more granularly; they do not merely replicate class separation but instead organize into a richer topology mirroring instance-specific features.

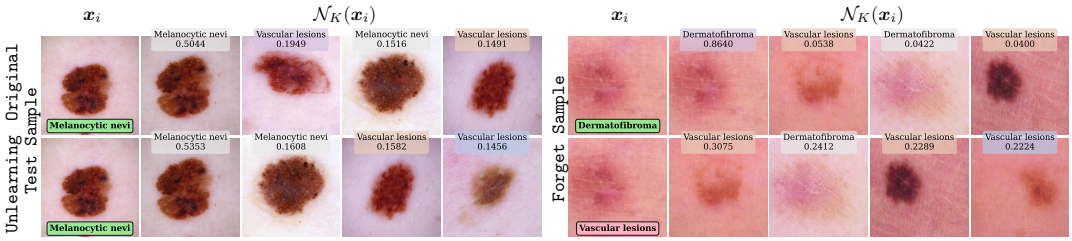


Figure 3: Retrieval visualizations on DermaMNIST on a test (left) and a forget (right) sample. Rows show the original and unlearned behavior, and their retrieved counterparts. Labels: new neighbors, correct matches, wrong-class neighbors, forgotten samples, correct and incorrect predictions.

To further illustrate the mechanics of unlearning within the memory bank, Figure 3 presents retrieval examples from the DermaMNIST test and forget sets, with $K = 4$ neighbors. By identifying a targeted instance and removing its key-value pair from the memory bank, we observe distinct behaviors: for a test sample $\boldsymbol{x}_i$, e.g., left rows, the key encoder continues to retrieve visually similar images $\mathcal{N}_K(\boldsymbol{x}_i)$, even if belonging to different classes, and MUNKEY retrieves all the updated $\boldsymbol{v}_j$ from its memory without changing the final predicted class. In contrast, for a training sample $\boldsymbol{x}_i$ marked for forgetting, e.g., right rows, the retrieved neighbor with the strongest weight prior to unlearning is the instance itself, measured through the softmax of cosine similarities. However, once unlearning is performed, a new instance is retrieved. This shift in neighbors, now without the instance-specific information, effectively alters the content and reweighting of all the retrieved $\boldsymbol{v}_j$ and changes the final class prediction. We further quantify the effect of neighboring key deletions on retained samples in Appendix D.6.

Effect of Aggregation Strategies. As established in Section 3.2, MUNKEY takes advantage of the synergistic benefits of image and memory pathways. While our architecture employs a pathway ensemble during inference, we investigate alternative aggregation strategies to determine how instance-specific information is integrated. These variants include Cross Attention, integrated at varying transformer depths; Deep Prompting (M. Jia et al., 2022), where the exemplar token is re-inserted as a prefix for every transformer block; and Direct Key, which bypasses the learnable memory by embedding the key encoder's output. Furthermore, within the standard input-level aggregation framework, we explore late-fusion techniques of the retrieved neighbors, comparing ensembling, rank-based mixing, and softmax-based weighting of exemplar tokens. We view MUNKEY not merely as a fixed model, but as a realization within the proposed paradigm *unlearning by design*, where different architectural choices yield distinct trade-offs and an analysis of these strategies is provided in Appendix D.1.

Table 3: Ablation of aggregation strategies on DermaMNIST and CIFAR-10. P_s denotes Pathway Sensitivity (lower is more balanced), Avg Gap denotes unlearning faithfulness (lower is better).

	DermaMNIST		CIFAR-10	
Strategy	P_s (↓)	Avg Gap (↓)	P_s (↓)	Avg Gap (↓)
Direct Key	$0.212_{\pm.012}$	$8.23_{\pm.70}$	$0.239_{\pm.007}$	$3.34_{\pm.18}$
Deep Prompt	$0.172_{\pm.006}$	$1.79_{\pm.69}$	$0.188_{\pm.006}$	$0.92_{\pm.52}$
Cross Attn.	$0.208_{\pm.005}$	$\mathbf{1.02}_{\pm.46}$	$0.292_{\pm.005}$	$\mathbf{0.29}_{\pm.13}$
Ensemble	$0.169_{\pm.004}$	$\underline{1.52}_{\pm.56}$	$0.184_{\pm.006}$	$\underline{0.58}_{\pm.22}$
Softmax	$0.168_{\pm.006}$	$2.01_{\pm.89}$	$0.186_{\pm.006}$	$0.71_{\pm.23}$
Rank	$0.166_{\pm.006}$	$2.58_{\pm.78}$	$0.184_{\pm.006}$	$0.67_{\pm.21}$

We report a subset of results on DermaMNIST and CIFAR-10 in Table 3, focused on the Pathway Sensitivity Score, P_s, and Avg Gap to represent unlearning and pathway reliance. We see that both

direct key and cross attention result in elevated P_s, indicating pathway imbalance. In the case of cross attention, there are utility improvements on Avg Gap, but those do not outweigh the added model complexity and worse P_s. Similarly, the limitations of the direct key approach empirically motivate the need for MUNKEY's instance-specific exemplar memory. Conversely, late-fusion methods and deep prompting maintain consistently lower P_s, signaling a more balanced integration of features. Among these, ensembling is the most robust, with the smallest Avg Gap and a stable P_s across both natural and medical domains. Although we observe fluctuations across datasets, the results suggest that pathway aggregation serves as a critical structural lever for unlearning, while the choice of late-fusion strategy is less impactful. Crucially, in scenarios where the computational overhead of K forward passes makes ensembling challenging, aggregating neighbors at the token level and doing a single inference pass, as in softmax and rank, offers a cost-effective alternative with competitive performance.

6 Conclusion

In this work, we advocate for a shift toward the *unlearning by design* paradigm in approximate methods. Current research focuses largely on post-hoc methods to adapt models that were never intended to forget. As data privacy regulations and security concerns intensify, there is a growing need for architectures that are inherently modular by construction. Within this paradigm, we propose MUNKEY, a framework that externalizes instance-specific knowledge into an explicitly editable memory bank. We reformulate the training and interaction of memory-augmented transformers so that the burden of instance-specific memorization is shifted from static weights to a learnable memory, enabling near-instantaneous, zero-shot forgetting. Results across diverse domains, including natural imagery and medical datasets, highlight the robustness of this approach. More broadly, our findings open new doors to a rich design space of modular architectures centered on adaptability for settings where stored information must be selectively adapted, updated, or removed over time. Beyond unlearning, this structural separation enables a form of interpretability, critical to high-stakes domains: by identifying retrieved "prototypes", i.e., influential neighbors for a decision, the model provides explicit evidence for its predictions, supporting transparent, case-based reasoning often absent in standard black-box systems. This design also entails practical trade-offs, including dataset-scaling memory overhead, typically modest relative to model size, and reliance on a key encoder, which trades additional storage for instantaneous and targeted unlearning. As with other approximate unlearning methods, our results provide empirical evidence under black-box membership inference rather than formal guarantees of forgetting. Ultimately, this work motivates exploration into approximate models that treat the ability to forget not as an afterthought, but as a fundamental architectural requirement, through decoupled memories or other instantiations of the *unlearning by design* paradigm.

Acknowledgments and Disclosure of Funding

MV and SL are supported by the Swiss State Secretariat for Education, Research, and Innovation (SERI) under contract number MB22.00047. JSG is supported by the grant #2021-911 of the Strategic Focal Area "Personalized Health and Related Technologies (PHRT)" of the ETH Domain (Swiss Federal Institutes of Technology). AR is supported by the StimuLoop grant #1-007811-002 and the Vontobel Foundation. Computational data analysis was performed at Leonhard Med (`https://sis.id.ethz.ch/services/sensitiveresearchdata/`) secure trusted research environment at ETH Zurich.

References

Acevedo, A., Merino, A., Alférez, S., Molina, Á., Boldú, L., & Rodellar, J. (2020). A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in brief*, *30*, 105474. [Referenced on page 6, 19]

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., ... Lacoste-Julien, S. (2017). A closer look at memorization in deep networks. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning, ICML 2017, sydney, nsw, australia, 6-11 august 2017* (Vol. 70, pp. 233–242). PMLR. Retrieved from `http://proceedings.mlr.press/v70/arpit17a.html` [Referenced on page 1, 2]

Biswas, S. D., Roy, A., & Roy, K. (2025). Cure: Concept unlearning via orthogonal representation editing in diffusion models. In *The thirty-ninth annual conference on neural information processing systems.* [Referenced on page 17]

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., ... Papernot, N. (2021). Machine unlearning. In *IEEE symposium on security and privacy (sp)* (pp. 141–159). [Referenced on page 1, 3, 17]

Cai, Z., Tan, Y., & Asif, M. S. (2025). Targeted unlearning with single layer unlearning gradient. In *Forty-second international conference on machine learning.* [Referenced on page 17]

California State Legislature. (2018). *California consumer privacy act of 2018 (CCPA).* Retrieved from `https://cppa.ca.gov/regulations/pdf/cppa_act.pdf` [Referenced on page 1]

Cao, Y., & Yang, J. (2015). Towards making systems forget with machine unlearning. In *IEEE symposium on security and privacy* (pp. 463–480). [Referenced on page 1, 3, 17]

Cha, S., Cho, S., Hwang, D., Lee, H., Moon, T., & Lee, M. (2024). Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In M. J. Wooldridge, J. G. Dy, & S. Natarajan (Eds.), *Thirty-eighth AAAI conference on artificial intelligence, AAAI 2024, thirty-sixth conference on innovative applications of artificial intelligence, IAAI 2024, fourteenth symposium on educational advances in artificial intelligence, EAAI 2014, february 20-27, 2024, vancouver, canada* (pp. 11186–11194). AAAI Press. Retrieved from `https://doi.org/10.1609/aaai.v38i10.28996` doi: 10.1609/AAAI.V38I10.28996 [Referenced on page 3]

Chen, C., Sun, F., Zhang, M., & Ding, B. (2022). Recommendation unlearning. In F. Laforest et al. (Eds.), *WWW '22: The ACM web conference 2022, virtual event, lyon, france, april 25 - 29, 2022* (pp. 2768–2777). ACM. Retrieved from `https://doi.org/10.1145/3485447.3511997` doi: 10.1145/3485447.3511997 [Referenced on page 17]

Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., & Zhang, Y. (2022). Graph unlearning. In *Proceedings of the 2022 acm sigsac conference on computer and communications security* (pp. 499–513). [Referenced on page 17]

Chundawat, V. S., Tarun, A. K., Mandal, M., & Kankanhalli, M. (2023b). Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, *18*, 2345–2354. [Referenced on page 3]

Chundawat, V. S., Tarun, A. K., Mandal, M., & Kankanhalli, M. S. (2023a). Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In B. Williams, Y. Chen, & J. Neville (Eds.), *Thirty-seventh AAAI conference on artificial intelligence, AAAI 2023, thirty-fifth conference on innovative applications of artificial intelligence, IAAI 2023, thirteenth symposium on educational advances in artificial intelligence, EAAI 2023, washington, dc, usa, february 7-14, 2023* (pp. 7210–7217). AAAI Press. Retrieved from `https://doi.org/10.1609/aaai.v37i6.25879` doi: 10.1609/AAAI.V37I6.25879 [Referenced on page 3]

Codella, N., Rotemberg, V., Tschandl, P., Celebi, M. E., Dusza, S., Gutman, D., ... others (2019). Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *ArXiv preprint*, *abs/1902.03368*. Retrieved from `https://arxiv.org/abs/1902.03368` [Referenced on page 19]

Cywiński, B., & Deja, K. (2025). Saeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders. In *Forty-second international conference on machine learning.* [Referenced on page 17]

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE computer society conference on computer vision and pattern*

recognition (CVPR 2009), 20-25 june 2009, miami, florida, USA (pp. 248–255). IEEE Computer Society. Retrieved from https://doi.org/10.1109/CVPR.2009.5206848 doi: 10.1109/CVPR.2009.5206848 [Referenced on page 6, 18]

Di, Z., Yu, S., Vorobeychik, Y., & Liu, Y. (2025). Adversarial machine unlearning. In *13th international conference on learning representations, ICLR 2025* (pp. 79672–79693). [Referenced on page 17]

Dong, J., Zhu, H., Zhang, Y., Qu, X., Ong, Y.-S., & Koniusz, P. (2025). Machine unlearning via task simplex arithmetic. In *The thirty-ninth annual conference on neural information processing systems.* [Referenced on page 17]

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *9th international conference on learning representations, ICLR 2021, virtual event, austria, may 3-7, 2021*. OpenReview.net. Retrieved from https://openreview.net/forum?id=YicbFdNTTy [Referenced on page 4]

Ebrahimpour-Boroojeny, A., Sundaram, H., & Chandrasekaran, V. (2025). Not all wrong is bad: Using adversarial examples for unlearning. In *Forty-second international conference on machine learning.* [Referenced on page 2, 3, 6, 20]

Fan, C., Liu, J., Zhang, Y., Wong, E., Wei, D., & Liu, S. (2024). Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The twelfth international conference on learning representations, ICLR 2024, vienna, austria, may 7-11, 2024*. OpenReview.net. Retrieved from https://openreview.net/forum?id=gn0mIhQGNM [Referenced on page 2, 3, 6]

Feng, Q., Tu, J., Kang, M., Zhao, H., Zhang, C., & Qian, H. (2025). Fg-oriu: Towards better forgetting via feature-gradient orthogonality for incremental unlearning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1957–1967). [Referenced on page 17]

Fix, E. (1985). *Discriminatory analysis: nonparametric discrimination, consistency properties* (Vol. 1). USAF school of Aviation Medicine. [Referenced on page 6, 20]

Foster, J., Schoepf, S., & Brintrup, A. (2024). Fast machine unlearning without retraining through selective synaptic dampening. In M. J. Wooldridge, J. G. Dy, & S. Natarajan (Eds.), *Thirty-eighth AAAI conference on artificial intelligence, AAAI 2024, thirty-sixth conference on innovative applications of artificial intelligence, IAAI 2024, fourteenth symposium on educational advances in artificial intelligence, EAAI 2014, february 20-27, 2024, vancouver, canada* (pp. 12043–12051). AAAI Press. Retrieved from https://doi.org/10.1609/aaai.v38i11.29092 doi: 10.1609/AAAI.V38I11.29092 [Referenced on page 3, 6, 19, 20]

Gandikota, R., Materzynska, J., Fiotto-Kaufman, J., & Bau, D. (2023). Erasing concepts from diffusion models. In *IEEE/CVF international conference on computer vision, ICCV 2023, paris, france, october 1-6, 2023* (pp. 2426–2436). IEEE. Retrieved from https://doi.org/10.1109/ICCV51070.2023.00230 doi: 10.1109/ICCV51070.2023.00230 [Referenced on page 2, 17]

Gao, H., Pang, T., Du, C., Hu, T., Deng, Z., & Lin, M. (2025). Meta-unlearning on diffusion models: Preventing relearning unlearned concepts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2131–2141). [Referenced on page 17]

Geirhos, R., Jaini, P., Stone, A., Medapati, S., Yi, X., Toderici, G., ... Shlens, J. (2025). Towards flexible perception with visual memory. In *Forty-second international conference on machine learning.* [Referenced on page 2, 4, 21, 22]

Ginart, A., Guan, M. Y., Valiant, G., & Zou, J. (2019). Making AI forget you: Data deletion in machine learning. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, neurips 2019, december 8-14, 2019, vancouver, bc, canada* (pp. 3513–3526). Retrieved from https://proceedings.neurips.cc/paper/2019/hash/cb79f8fa58b91d3af6c9c991f63962d3-Abstract.html [Referenced on page 17]

Goel, S., Prabhu, A., Sanyal, A., Lim, S.-N., Torr, P., & Kumaraguru, P. (2022). Towards adversarial evaluations for inexact machine unlearning. *ArXiv preprint*, *abs/2201.06640*. Retrieved from https://arxiv.org/abs/2201.06640 [Referenced on page 3, 6, 19]

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*. [Referenced on page 3, 17, 22]

Graves, L., Nagisetty, V., & Ganesh, V. (2021). Amnesiac machine learning. In *Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, virtual event, february 2-9, 2021* (pp. 11516–11524). AAAI Press. Retrieved from `https://ojs.aaai.org/index.php/AAAI/article/view/17371` [Referenced on page 3]

Guo, C., Goldstein, T., Hannun, A. Y., & van der Maaten, L. (2020). Certified data removal from machine learning models. In *Proceedings of the 37th international conference on machine learning, ICML 2020, 13-18 july 2020, virtual event* (Vol. 119, pp. 3832–3842). PMLR. Retrieved from `http://proceedings.mlr.press/v119/guo20c.html` [Referenced on page 17]

Guo, P. H., Syed, A., Sheshadri, A., Ewart, A., & Dziugaite, G. K. (2025). Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization. In *Forty-second international conference on machine learning*. [Referenced on page 17]

Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., & Kumar, S. (2020). Accelerating large-scale inference with anisotropic vector quantization. In *Proceedings of the 37th international conference on machine learning, ICML 2020, 13-18 july 2020, virtual event* (Vol. 119, pp. 3887–3896). PMLR. Retrieved from `http://proceedings.mlr.press/v119/guo20h.html` [Referenced on page 4]

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9729–9738). [Referenced on page 21]

Hsu, H., Niroula, P., He, Z., Brugere, I., Lecue, F., & Chen, C.-F. (2025). The unseen threat: Residual knowledge in machine unlearning under perturbed samples. In *The thirty-ninth annual conference on neural information processing systems*. [Referenced on page 2]

Huang, M. H., Foo, L. G., & Liu, J. (2024). Learning to unlearn for robust machine unlearning. In *European conference on computer vision* (pp. 202–219). [Referenced on page 17]

Hussen Abdelaziz, A., Theobald, B.-J., Dixon, P., Knothe, R., Apostoloff, N., & Kajareker, S. (2020). Modality dropout for improved performance-driven talking faces. In *Proceedings of the 2020 international conference on multimodal interaction* (pp. 378–386). [Referenced on page 4]

Jayaram Subramanya, S., Devvrit, F., Simhadri, H. V., Krishnawamy, R., & Kadekodi, R. (2019). Diskann: Fast accurate billion-point nearest neighbor search on a single node. *Advances in neural information processing Systems*, *32*. [Referenced on page 22]

Jegou, H., Douze, M., & Schmid, C. (2010). Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, *33*(1), 117–128. [Referenced on page 22]

Jia, J., Liu, J., Ram, P., Yao, Y., Liu, G., Liu, Y., … Liu, S. (2023). Model sparsity can simplify machine unlearning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems 36: Annual conference on neural information processing systems 2023, neurips 2023, new orleans, la, usa, december 10 - 16, 2023*. Retrieved from `http://papers.nips.cc/paper_files/paper/2023/hash/a204aa68ab4e970e1ceccfb5b5cdc5e4-Abstract-Conference.html` [Referenced on page 3, 6, 19]

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., & Lim, S.-N. (2022). Visual prompt tuning. In *European conference on computer vision* (pp. 709–727). [Referenced on page 2, 3, 9, 17, 26]

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE transactions on big data*, *7*(3), 535–547. [Referenced on page 22]

Kather, J. N., Halama, N., & Marx, A. (2018). 100,000 histological images of human colorectal cancer and healthy tissue. *(No Title)*. [Referenced on page 19]

Kather, J. N., Krisam, J., Charoentong, P., Luedde, T., Herpel, E., Weis, C.-A., … others (2019). Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, *16*(1), e1002730. [Referenced on page 6, 19]

Kawamura, K., Goto, Y., Yanagi, R., Kataoka, H., & Irie, G. (2025). Approximate domain unlearning for vision-language models. In *The thirty-ninth annual conference on neural information processing systems.* [Referenced on page 17]

Khalil, Y. H., Setayesh, M., & Li, H. (2025). Coun: Empowering machine unlearning via contrastive learning. In *The thirty-ninth annual conference on neural information processing systems.* [Referenced on page 2, 3, 6, 20]

Kim, H., Kim, K., Chae, S., & Yoon, S. (2025). Unlearning-aware minimization. In *The thirty-ninth annual conference on neural information processing systems.* [Referenced on page 17]

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. *Department of Computer Science, University of Toronto, ON, Canada*. [Referenced on page 6, 18]

Kurmanji, M., Triantafillou, P., Hayes, J., & Triantafillou, E. (2023). Towards unbounded machine unlearning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems 36: Annual conference on neural information processing systems 2023, neurips 2023, new orleans, la, usa, december 10 - 16, 2023*. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/062d711fb777322e2152435459e6e9d9-Abstract-Conference.html [Referenced on page 3, 5, 6, 19, 20, 34]

Lample, G., Sablayrolles, A., Ranzato, M., Denoyer, L., & Jégou, H. (2019). Large memory layers with product keys. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, neurips 2019, december 8-14, 2019, vancouver, bc, canada* (pp. 8546–8557). Retrieved from https://proceedings.neurips.cc/paper/2019/hash/9d8df73a3cfbf3c5b47bc9b50f214aff-Abstract.html [Referenced on page 17]

Lee, B. W., Foote, A., Infanger, A., Shor, L., Kamath, H., Goldman-Wetzler, J., ... Turner, A. M. (2025). Distillation robustifies unlearning. *Forty-second International Conference on Machine Learning*. [Referenced on page 17]

Lee, S., Chung, S., & Woo, S. S. (2025). Ruago: Effective and practical retain-free unlearning via adversarial attack and ood generator. In *The thirty-ninth annual conference on neural information processing systems.* [Referenced on page 2, 17]

Li, J., Vo, D. M., Sugimoto, A., & Nakayama, H. (2024). Evcap: Retrieval-augmented image captioning with external visual-name memory for open-world comprehension. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13733–13742). [Referenced on page 2, 3]

Lin, S., Zhang, X., Chen, C., Chen, X., & Susilo, W. (2023). ERM-KTP: knowledge-level machine unlearning via knowledge transfer. In *IEEE/CVF conference on computer vision and pattern recognition, CVPR 2023, vancouver, bc, canada, june 17-24, 2023* (pp. 20147–20155). IEEE. Retrieved from https://doi.org/10.1109/CVPR52729.2023.01929 doi: 10.1109/CVPR52729.2023.01929 [Referenced on page 3]

Long, A., Yin, W., Ajanthan, T., Nguyen, V., Purkait, P., Garg, R., ... Van den Hengel, A. (2022). Retrieval augmented classification for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6959–6969). [Referenced on page 2, 8, 17, 33]

Mantelero, A. (2013). The eu proposal for a general data protection regulation and the roots of the right to be forgotten. *Computer Law & Security Review*, *29*(3), 229–235. [Referenced on page 1]

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation* (Vol. 24, pp. 109–165). Elsevier. [Referenced on page 3]

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv preprint*, *abs/1802.03426*. Retrieved from https://arxiv.org/abs/1802.03426 [Referenced on page 9]

Mekala, A. R., Dorna, V., Dubey, S., Lalwani, A., Koleczek, D., Rungta, M., ... Lobo, E. A. (2025). Alternate preference optimization for unlearning factual knowledge in large language models. In *Proceedings of the 31st international conference on computational linguistics* (pp. 3732–3752). [Referenced on page 17]

Mirlach, J., Laguna, S., & Vogt, J. E. (2026). Reference-guided machine unlearning. In *Transactions on machine learning research.* [Referenced on page 3]

Nguyen, T. T., Huynh, T. T., Ren, Z., Nguyen, P. L., Liew, A. W.-C., Yin, H., & Nguyen, Q. V. H. (2025). A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, *16*(5), 1–46. [Referenced on page 1, 3, 17]

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., ... others (2024). Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 1–31. [Referenced on page 8]

Patel, G., & Qiu, Q. (2025). Learning to unlearn while retaining: Combating gradient conflicts in machine unlearning. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 4211–4221). [Referenced on page 6, 17]

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning, ICML 2021, 18-24 july 2021, virtual event* (Vol. 139, pp. 8748–8763). PMLR. Retrieved from http://proceedings.mlr.press/v139/radford21a.html [Referenced on page 8]

Sandler, M., Zhmoginov, A., Vladymyrov, M., & Jackson, A. (2022). Fine-tuning image transformers using learnable memory. In *IEEE/CVF conference on computer vision and pattern recognition, CVPR 2022, new orleans, la, usa, june 18-24, 2022* (pp. 12145–12154). IEEE. Retrieved from https://doi.org/10.1109/CVPR52688.2022.01184 doi: 10.1109/CVPR52688.2022.01184 [Referenced on page 2, 3, 17]

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. P. (2016). Meta-learning with memory-augmented neural networks. In M. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33rd international conference on machine learning, ICML 2016, new york city, ny, usa, june 19-24, 2016* (Vol. 48, pp. 1842–1850). JMLR.org. Retrieved from http://proceedings.mlr.press/v48/santoro16.html [Referenced on page 17, 22]

Shen, S., Zhang, C., Zhao, Y., Bialkowski, A., Chen, W., & Xu, M. (2024). Label-agnostic forgetting: A supervision-free unlearning in deep models. In *The twelfth international conference on learning representations, ICLR 2024, vienna, austria, may 7-11, 2024*. OpenReview.net. Retrieved from https://openreview.net/forum?id=SIZWiya7FE [Referenced on page 17]

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (sp)* (pp. 3–18). [Referenced on page 1]

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems 28: Annual conference on neural information processing systems 2015, december 7-12, 2015, montreal, quebec, canada* (pp. 2440–2448). [Referenced on page 3, 17]

Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, *5*(1), 1–9. [Referenced on page 6, 19]

Wang, S., Ye, J., & Wang, X. (2025). Machine unlearning in 3d generation: A perspective-coherent acceleration framework. In *The thirty-ninth annual conference on neural information processing systems.* [Referenced on page 17]

Wang, W., Tran, D., & Feiszli, M. (2020). What makes training multi-modal classification networks hard? In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, seattle, wa, usa, june 13-19, 2020* (pp. 12692–12702). IEEE. Retrieved from https://doi.org/10.1109/CVPR42600.2020.01271 doi: 10.1109/CVPR42600.2020.01271 [Referenced on page 4]

Wang, Y., Wei, J., Liu, C. Y., Pang, J., Liu, Q., Shah, A., ... Wei, W. (2025). LLM unlearning via loss adjustment with only forget data. In *The thirteenth international conference on learning representations.* Retrieved from https://openreview.net/forum?id=6ESRicalFE [Referenced on page 17]

Weston, J., Chopra, S., & Bordes, A. (2015). Memory networks. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings*. Retrieved from http://arxiv.org/abs/1410.3916 [Referenced on page 3, 17, 22]

Wu, X., Pang, Y., Liu, T., & Wu, S. (2025). Unlearned but not forgotten: Data extraction after exact unlearning in LLM. In *The thirty-ninth annual conference on neural information processing systems*. [Referenced on page 17]

Wu, Y., Rabe, M. N., Hutchins, D., & Szegedy, C. (2022). Memorizing transformers. In *The tenth international conference on learning representations, ICLR 2022, virtual event, april 25-29, 2022*. OpenReview.net. Retrieved from https://openreview.net/forum?id=TrjbxzRcnf- [Referenced on page 3, 17]

Wu, Z., Xiong, Y., Yu, S. X., & Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3733–3742). [Referenced on page 21]

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., … Ni, B. (2023). Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, *10*(1), 41. [Referenced on page 6, 19]

Zarifzadeh, S., Liu, P., & Shokri, R. (2024). Low-cost high-power membership inference attacks. In *Proceedings of the 41st international conference on machine learning* (pp. 58244–58282). [Referenced on page 6]

Zhang, R., Zhang, W., Fang, R., Gao, P., Li, K., Dai, J., … Li, H. (2022). Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision* (pp. 493–510). [Referenced on page 17]

Zhang, Z., Liu, G., Fleming, C., Kompella, R. R., & Xu, C. (2025). Targeted forgetting of image subgroups in CLIP models. In *Proceedings of the computer vision and pattern recognition conference* (pp. 9870–9880). [Referenced on page 2, 17]

Zhou, Y., Zheng, D., Mo, Q., Lu, R., Lin, K.-Y., & Zheng, W.-S. (2025). Decoupled distillation to erase: A general unlearning method for any class-centric tasks. In *Proceedings of the computer vision and pattern recognition conference* (pp. 20350–20359). [Referenced on page 17]

A Extended Related Work

Machine Unlearning. In this section, we describe additional work tangential to MUNKEY but relevant to the broader picture of machine unlearning. Beyond approximate methods discussed in Section 2, exact unlearning has also been studied, typically with formal theoretical guarantees (Ginart et al., 2019; C. Guo et al., 2020; X. Wu et al., 2025). Retraining from scratch is sometimes considered the only truly exact unlearning method, as it fully guarantees that information from the forget set is not retained in the model (Nguyen et al., 2025); however, this approach is often impractical. To mitigate this, prior work proposes data sharding and slicing strategies, where models are trained on disjoint partitions and ensembled to enable exact unlearning at the shard level (Cao & Yang, 2015; Bourtoule et al., 2021; C. Guo et al., 2020). Such approaches have been applied in domains including graph neural networks (M. Chen et al., 2022) and recommendation systems (C. Chen et al., 2022). Nevertheless, exact unlearning methods often incur substantial computational costs due to retraining on large retain sets, or high storage costs from maintaining multiple models trained on disjoint data subsets. Further lines of research address unlearning from alternative optimization perspectives, aiming to better balance forgetting and retention (S. Lee et al., 2025). This includes formulations based on game theory, i.e., where unlearning is cast as a Stackelberg game between the unlearner and an auditor (Di et al., 2025). Other works (Kim et al., 2025) adopt meta-learning perspectives to jointly optimize forgetting and remembering (Huang et al., 2024), or explicitly mitigate gradient conflicts between unlearning and retention objectives through implicit regularization mechanisms (Patel & Qiu, 2025). Incremental unlearning has also been explored in pre-trained vision models via feature subspace decomposition and dynamic subspace adaptation (Feng et al., 2025; Kawamura et al., 2025; S. Wang et al., 2025). Finally, a growing body of work studies unsupervised unlearning using variational inference (Shen et al., 2024); while such methods can achieve performance comparable to supervised approaches, they are generally not superior, and we therefore do not include them in our comparisons.

Unlearning approaches can be distinguished by the level of information they aim to remove. While many of the aforementioned methods focus on instance- or class-level unlearning, i.e. (Zhou et al., 2025), a growing body of work targets concept unlearning, particularly in generative settings. For example, recent work studies concept removal and relearning prevention in diffusion models (Gandikota et al., 2023; Gao et al., 2025; Biswas et al., 2025), and in large foundation models such as CLIP, Stable Diffusion, or VLMs, by identifying and modifying relevant layers with efficient gradient-based localization (Cai et al., 2025; Z. Zhang et al., 2025; Dong et al., 2025). Similarly, a large body of work targets large language models, including approaches based on distillation (B. W. Lee et al., 2025), mechanistic localization (P. H. Guo et al., 2025), or alternative supervision strategies (Mekala et al., 2025; Y. Wang et al., 2025). From another angle, SAeUron (Cywiński & Deja, 2025) adopts sparse autoencoders from mechanistic interpretability to remove specific concepts in diffusion models. While highly relevant for generative modeling, these approaches address a different unlearning setting from our discriminative vision task and are therefore beyond the scope of this work.

Associative and External Memories. To give a broader context to the exemplar memory we propose in MUNKEY, we discuss here associative and external memory mechanisms that have been widely explored to augment neural networks with explicit storage and retrieval capabilities. Associative (internal) memory approaches embed learnable memory structures directly into network components, including product key memory layers (Lample et al., 2019) and fine-tuning methods that introduce learnable memory tokens to provide persistent, task-specific context in Vision Transformer encoders (Sandler et al., 2022). Similarly, Visual Prompt Tuning (M. Jia et al., 2022) introduces learnable tokens that are embedded into the model to adapt its behavior. In contrast, as described in Section 2, external memory architectures decouple memory from the model parameters (A. Graves et al., 2014; Weston et al., 2015; Sukhbaatar et al., 2015; Santoro et al., 2016; Y. Wu et al., 2022). More recent work on retrieval-augmented models, such as Retrieval-Augmented Classification (RAC) (Long et al., 2022), combines a parametric image classifier with an external retrieval memory. However, since training samples influence both the classifier weights and the retrieval memory, deleting an entry does not remove its influence from the learned parameters. In contrast, Tip-Adapter (R. Zhang et al., 2022) keeps CLIP frozen and stores downstream examples only in an external cache, allowing their contribution to be removed through deletion. However, this is a frozen-model adaptation setting rather than unlearning, as these examples never contributed to parameter training. MUNKEY instead explicitly trains the model for subsequent deletion, encouraging instance-specific information to be localized in removable memory rather than shared model parameters.

B Detailed Experimental Setting

B.1 MUNKEY Algorithm

Algorithm 1 summarizes the MUNKEY pipeline, from training with exemplar memory to unlearning by key deletion and inference via nearest-neighbor retrieval, as introduced in Section 3.2.

Algorithm 1 MUNKEY: Machine UNlearning via KEY Deletion

Input: Training set $\mathcal{D}$, frozen key encoder g_ϕ, p_i, p_t, p_r
Initialize: Model θ, adapter ψ, and memory bank $\mathcal{M} = \{(\boldsymbol{k}_i, \boldsymbol{v}_i)\}_{i=1}^{N}$
Phase 1: Training
Precompute $\boldsymbol{k}_i = g_\phi(\boldsymbol{x}_i)$ for all $\boldsymbol{x}_i \in \mathcal{D}$
while not converged **do**
  for each $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}$ **do**
    $\boldsymbol{z}_i \leftarrow$ ViTEmbed$(\boldsymbol{x}_i)$
    Sample $\mathbf{m} = [\gamma_{img}, \gamma_{tok}] \sim$ Cat$(p_i, p_t, 1 - p_i - p_t, 0)$
    Sample $r \sim$ Bern(p_r)
    if $r = 1$ **then**
      Sample $K' \sim \mathcal{U}\{2, 16\}$
      $\boldsymbol{v}_n \leftarrow$ Agg$(\{\boldsymbol{v}_j \mid j \in \mathcal{N}_{K'}(\boldsymbol{x}_i; \mathcal{M} \setminus \{(\boldsymbol{k}_i, \boldsymbol{v}_i)\})\})$
      Exemplar $\leftarrow h_\psi(\boldsymbol{v}_n)$
    else
      Exemplar $\leftarrow h_\psi(\boldsymbol{v}_i)$
    end if
    $\boldsymbol{z}_{in} \leftarrow [\mathcal{P}(\boldsymbol{z}_i, \emptyset_{img}, \gamma_{img}), \mathcal{P}(\text{Exemplar}, \emptyset_{tok}, \gamma_{tok})]$
    $\mathcal{L} \leftarrow \ell(f_\theta(\boldsymbol{z}_{in}), \boldsymbol{y}_i)$
    if $r = 1$ **then**
      Update θ, ψ by minimizing $\mathcal{L}$; set $\nabla_{\boldsymbol{v}_j}\mathcal{L} = 0$ for retrieved neighbors j
    else
      Update $\theta, \psi, \boldsymbol{v}_i$ by minimizing $\mathcal{L}$
    end if
  end for
end while
Phase 2: Zero-Shot Unlearning
Input: Forget set $\mathcal{D}_f$
$\mathcal{M}_u \leftarrow \mathcal{M} \setminus \{(\boldsymbol{k}_i, \boldsymbol{v}_i) \mid (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}_f\}$ {Instance key deletion}
Phase 3: Inference
Input: Query $\boldsymbol{x}_q$, Updated Memory $\mathcal{M}_u$
Retrieve $\mathcal{N}_K(\boldsymbol{x}_q)$ from $\mathcal{M}_u$ based on sim$(g_\phi(\boldsymbol{x}_q), \boldsymbol{k}_j)$
$\boldsymbol{z}_q \leftarrow$ ViTEmbed$(\boldsymbol{x}_q)$
$\hat{\boldsymbol{y}} \leftarrow \sum_{j \in \mathcal{N}_K(\boldsymbol{x}_q)} w_j \cdot f_\theta([\boldsymbol{z}_q, h_\psi(\boldsymbol{v}_j)])$
return $\hat{\boldsymbol{y}}$

B.2 Datasets

To evaluate the performance and robustness of our proposed method, we conduct experiments across a diverse range of benchmarks, spanning both natural and medical imaging domains. All datasets used in this work are publicly available and were obtained from their official sources. We use them exclusively for research and benchmarking purposes, and follow the corresponding dataset terms of use.

Natural Image Benchmarks. We first evaluate our model on the CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009). Both are established computer vision benchmarks consisting of 60,000 32×32 color images, 50,000 in the training set and 10,000 in the validation set. CIFAR-10 contains 10 mutually exclusive classes, whereas CIFAR-100 provides a more complex challenge with 100 fine-grained categories. Additionally, we utilize Tiny ImageNet, a subset[2] of the larger ImageNet database (Deng et al., 2009). This dataset comprises 200 object classes with a total of 100,000 training, 10,000 validation, and 10,000 test unlabeled images. For all three natural image benchmarks, we perform a random split of the original validation data to create a custom validation set (40%) and a test set (60%). All images in these datasets are resized to 224×224 resolution.

[2]Tiny ImageNet Visual Recognition Challenge: `http://cs231n.stanford.edu/tiny-imagenet-200.zip`

Biomedical Image Benchmarks. To assess the method's efficacy in specialized domains, we incorporate high-resolution (224×224) tasks from the MedMNIST v2 collection (Yang et al., 2023). We include three datasets of varying clinical nature: First, BloodMNIST (Acevedo et al., 2020) is a collection of 17,092 microscopic images of normal blood cells across 8 classes with moderate class imbalance. It serves as a benchmark for morphological recognition, requiring the model to distinguish subtle structural features like nuclear shape and cytoplasmic granules against controlled backgrounds. We utilize the predefined splits: 11,959 training, 1,712 validation, and 3,421 test samples. Second, PathMNIST (Kather et al., 2019, 2018) comprises 107,180 histology patches for 9-class colorectal cancer tissue classification. This dataset is chosen to represent a large-scale medical imaging task. We follow the standard training split (89,996 samples); however, as the original test set (7,180 samples) is collected from different clinical centers to test out-of-distribution generalization, which is outside the scope of this unlearning study, we utilize the validation set (10,004 samples) for testing to maintain focus on the core methodology. Lastly, DermaMNIST (Tschandl et al., 2018; Codella et al., 2019) contains 10,015 dermatoscopic images, 7007 for training, 1003 for validation, and 2005 for testing, used for 7-class skin disease classification. It mimics real-world clinical prevalence, where common conditions like *melanocytic nevi* significantly outnumber rarer, critical pathologies such as *melanoma*. Furthermore, its multi-source nature introduces hardware and photographic noise, making it a challenging benchmark for imbalanced, real-world diagnosis. Given its representative difficulty, DermaMNIST is included in our main manuscript alongside CIFAR-10 as primary benchmarks, while the results for the remaining datasets are provided in Appendices C, D.1.

B.3 Baselines Setup

To provide a comprehensive evaluation of MUNKEY, we compare against a diverse suite of baselines spanning traditional optimization, weight sparsification, distillation-based unlearning, parameter dampening, and recent adversarial unlearning. We first note that MUNKEY is Transformer-based; however, most post-hoc unlearning methods are primarily reported on CNN or ResNet backbones, despite being applicable to Transformers. Since ViTs are the current gold standard backbone for vision, we implement *all* baselines using a ViT-Tiny architecture, as in MUNKEY, to ensure a fair comparison and to bridge this reporting gap, bringing results of the state-of-the-art into transformer-based unlearning. For all optimization-based baselines adapted to ViT-Tiny, we replace the commonly used SGD optimizer with AdamW (betas $[0.9, 0.99]$, weight decay 0.05). Additionally, we perform a learning-rate grid search using multiplicative factors in $[0.1, 1.0]$ relative to the original training rate of the base model 1.5×10^{-4}. In this section, we list the baselines considered and their implementation details, and report their training and unlearning runtimes in Table 4. While gradient-based methods like NegGrad+ and SCRUB require significant compute for parameter updates, and methods like SSD require time to calculate the Fisher Information Matrix, MUNKEY's unlearning time is negligible (≈ 0s), hence O(1) computations.

Retrained oracle. Retrain serves as the fundamental oracle baseline. For all baselines, this corresponds to training a ViT-Tiny from scratch on the retain data $\mathcal{D}_r$. For our method, we additionally report an oracle that matches the MUNKEY ViT-Tiny architecture with its hyperparameters and additional components, also trained uniquely on $\mathcal{D}_r$, providing an upper bound aligned with our design.

Traditional optimization-based baselines. These include earlier machine unlearning approaches. Catastrophic Forgetting (CF) fine-tunes the model on $\mathcal{D}_r$. CF-k (Goel et al., 2022) uses the same fine-tuning procedure but freezes the backbone and updates only the classifier weights. Moreover, Exact Unlearning-k (EU-k) (Goel et al., 2022) resets the classifier parameters before fine-tuning on $\mathcal{D}_r$. NegGrad+ (Kurmanji et al., 2023) is an adaptation of the more primitive NegGrad that performs gradient ascent on the forget set $\mathcal{D}_f$ while simultaneously performing gradient descent on $\mathcal{D}_r$; a grid search over the forget/utility trade-off found $0.95/0.05$ to work best as the weight ratio of both loss terms.

Weight sparsification. ℓ_1-sparse (J. Jia et al., 2023) applies an ℓ_1-penalty to the loss, with scaling parameter $\lambda = 0.01$ to the knowledge weights in the backbone to sparsify memorized information, while excluding biases and embeddings for stability.

Noise-based baseline. Amnesiac unlearning, implemented as in Foster et al. (2024), applies random labels as noise supervision on the forget set $\mathcal{D}_f$ during fine-tuning with the rest of the data.

Following recent literature (Ebrahimpour-Boroojeny et al., 2025), and our own empirical findings, we run all the above methods for 10 epochs to achieve the best trade-off between forgetting and utility.

Teacher-Student distillation baseline. SCRUB (Kurmanji et al., 2023) is a common representative of the distillation-based methods, where a student model is trained to selectively inherit knowledge from the original model (the teacher), specifically by minimizing KL-divergence to stay close to the teacher on $\mathcal{D}_r$ while maximizing that divergence to move away from it on $\mathcal{D}_f$. We follow the original codebase and optimization steps, unlearning for 10 epochs, and using their *rewinding* procedure to select the checkpoint with the smallest validation utility gap.

Optimization-free parameter dampening. This category of methods includes SSD (Selective Synaptic Dampening) (Foster et al., 2024). Unlike fine-tuning approaches, and similar to MUNKEY, these methods do not require iterative optimization at unlearning time; instead, they identify and dampen the weights most responsible for the forget set. In particular, SSD relies on computing the Fisher Information Matrix (FIM) to identify relevant parameters. While this allows for near-instant unlearning, the FIM calculation introduces a computational overhead prior to the unlearning phase.

Adversarial-based unlearning. AMUN (Adversarial Machine Unlearning) (Ebrahimpour-Boroojeny et al., 2025) represents a new line of recent work. It generates an adversarial set of images for the forget samples and then fine-tunes the model on $\mathcal{D}_r$ together with these adversarial samples to induce forgetting. We remained faithful to their adversarial set generation pipeline by using an untargeted PGD-50 attack under an ℓ_2-constraint, starting with step size 0.1ε and increasing ε until misclassification.

Contrastive-based Unlearning. CoUn (Contrastive Machine Unlearning) (Khalil et al., 2025) frames unlearning as representation-space adjustment through contrastive learning on the retain data, combining a SimCLR-style InfoNCE loss with supervised cross-entropy. We follow the official protocol and reported hyperparameters of the original work, unlearning for 50 epochs with the data augmentation pipeline used for the contrastive branch, i.e., stochastic image transformations such as random cropping, horizontal flipping, and normalization.

Memory-based baseline. Finally, KNN (Fix, 1985) provides a simple retrieval baseline motivated by the memory component of MUNKEY. We use the same key-encoded embeddings as the search space and predict class probabilities by counting labels among the nearest neighbors. This provides a point of comparison for the simplest form of memory-based inference.

Table 4: Run-Time Efficiency comparison (seconds) on DermaMNIST and CIFAR-10. *Unlearn* is the time to remove data post-training. *Train/Ep.* is the per-epoch training cost of base models.

	DermaMNIST		*CIFAR-10*	
Model	**Unlearn**	**Train/Ep.**	**Unlearn**	**Train/Ep.**
KNN	≈ 0	≈ 0	≈ 0	≈ 0
CF	$114.1_{\pm 1.9}$	$24.7_{\pm 0.2}$	$669.5_{\pm 1.6}$	$174.6_{\pm 1.6}$
CF-k	$58.5_{\pm 2.8}$	$24.7_{\pm 0.2}$	$311.4_{\pm 1.6}$	$174.6_{\pm 1.6}$
EU-k	$58.2_{\pm 0.9}$	$24.7_{\pm 0.2}$	$311.1_{\pm 1.1}$	$174.6_{\pm 1.6}$
ℓ_1-sparse	$114.3_{\pm 0.8}$	$24.7_{\pm 0.2}$	$694.2_{\pm 0.9}$	$174.6_{\pm 1.6}$
NegGrad+	$124.3_{\pm 1.8}$	$24.7_{\pm 0.2}$	$761.7_{\pm 1.1}$	$174.6_{\pm 1.6}$
Amnesiac	$123.8_{\pm 1.3}$	$24.7_{\pm 0.2}$	$760.5_{\pm 1.4}$	$174.6_{\pm 1.6}$
SCRUB	$154.4_{\pm 0.3}$	$24.7_{\pm 0.2}$	$1478.2_{\pm 27.3}$	$174.6_{\pm 1.6}$
SSD	$23.1_{\pm 4.0}$	$24.7_{\pm 0.2}$	$105.8_{\pm 13.6}$	$174.6_{\pm 1.6}$
AMUN	$313.4_{\pm 6.8}$	$24.7_{\pm 0.2}$	$1940.5_{\pm 3.7}$	$174.6_{\pm 1.6}$
CoUn	$191.75_{\pm 5.3}$	$24.7_{\pm 0.2}$	$1255.88_{\pm 11.4}$	$174.6_{\pm 1.6}$
MUNKEY	$\approx \mathbf{0}$	$25.9_{\pm 0.3}$	$\approx \mathbf{0}$	$178.5_{\pm 2.7}$

B.4 Implementation Details

The training pipeline and dynamics are standardized across all base models and datasets. We employ a ViT-Tiny (ViT-T/16) backbone with a patch size of 16 and a hidden dimension of 192 for both MUNKEY and the baseline architectures, and we fit a linear classifier on the final $[CLS]$ token. At training time, natural images are augmented using random resize cropping to 224×224 pixels, while biomedical images are directly resized to 224×224 pixels. All models are trained for 100 epochs using the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.99$, a weight decay of 0.05, and a learning rate

that follows a cosine annealing schedule starting at 1.5×10^{-4} until the end of training. Regarding MUNKEY-specific hyperparameters, the regularizing p_r is 0.2, the memory bank utilizes a token dimension of 128, and we include a linear layer adapter to map it to the 192 ViT-T/16 latent space. We retrieve $K = 4$ nearest neighbors during inference in the experiments in the main text, and show results on the effect of different numbers of K in Appendix D.2.

The temperature parameter τ in the softmax operation for neighbor retrieval is set to 0.07 following standard practice in similarity-based representation learning and retrieval architectures (Z. Wu et al., 2018; He et al., 2020). Moreover, the key encoder used for retrieval is a ViT-B/16 pre-trained on ImageNet (sourced from HuggingFace[3]), and again we provide further ablations regarding the choice of key encoder in Appendix D.2. The dropout probabilities for each dataset, selected based on Appendix E, are $p_i = 0.1, p_t = 0.3$ for CIFAR-10/100; $p_i = 0.1, p_t = 0.1$ for PathMNIST and Tiny ImageNet; and $p_i = 0.3, p_t = 0.1$ for DermaMNIST and BloodMNIST. All experiments were performed on an NVIDIA GeForce RTX 3080 GPU.

B.5 System Costs

A key motivation of the unlearning by design paradigm is that the cost of forgetting is shifted away from iterative optimization and into lightweight memory operations. In this section, we discuss the system characteristics of MUNKEY and the practical implications of this design.

Memory Footprint. The storage requirements of MUNKEY scale linearly with dataset size, as the external memory stores only key-value pairs. Concretely, the memory complexity is $O(N(d_k + d_e))$, where N is the number of stored samples, d_k is the key dimension, and d_e is the exemplar token dimension. We study the effect of the token dimensionality in Appendix D.3 and find that strong performance is maintained even with compact tokens of 64 dimensions. As an example, for CIFAR-10 ($N = 50{,}000$), using a key dimension of 192 and an exemplar token dimension of 64, the total uncompressed float16 memory overhead is 24.41 MiB, consisting of 18.31 MiB for keys and 6.10 MiB for exemplar tokens, with 3.20M additional learnable exemplar parameters. For DermaMNIST ($N = 7{,}007$), the total overhead corresponds to 3.42 MiB and 0.45M learnable exemplar parameters. For larger deployments, this footprint can be further reduced through standard compression and approximate retrieval techniques such as Product Quantization, as shown in (Geirhos et al., 2025). Beyond compression, GPU memory requirements can be further reduced by storing the memory bank on CPU or SSD, transferring only the retrieved exemplar tokens to the accelerator. This reduces the memory footprint associated with retrieval to $O(Kd_e)$ per query, independent of dataset size, at the cost of additional retrieval latency.

Training Cost. Although the number of learnable exemplar parameters grows linearly with dataset size, this does not necessarily translate into increased per-step training cost. To evaluate this, we measure training throughput with memory banks ranging from 1K to 100K entries on Tiny ImageNet with $d_e = 64$, keeping the backbone and batch size fixed. As shown in Table 5, MUNKEY's per-step training time remains stable across the evaluated memory sizes. This is because key embeddings are precomputed before training, and each minibatch accesses only its own exemplar tokens and a fixed number of retrieved neighbors. Nevertheless, the total number of steps per epoch still scales with dataset size, as in regular learning, and exemplar parameters and their optimizer states require $O(Nd_e)$ storage.

Unlearning and Update Costs. The main systems advantage of MUNKEY is that forgetting does not require retraining, gradient updates, or expensive parameter editing. Instead, unlearning is implemented through a lightweight masking operation over the memory bank, where forgotten entries are assigned $-\infty$ in the similarity computation and can no longer be retrieved. This yields a deletion complexity of $O(1)$ per deleted instance, in contrast to optimization-based approaches that scale with dataset size, training epochs, and backpropagation cost. Importantly, multiple deletion requests can be handled simultaneously in a single masking operation, rather than sequentially processing each request. In practice, this makes forgetting operationally simple and substantially faster than post-hoc alternatives.

[3] https://huggingface.co/google/vit-base-patch16-224

Table 5: Training scalability of MUNKEY across different memory bank sizes. We report the number of learnable exemplar parameters and steady-state training step time on an RTX 3080, keeping the backbone and batch size fixed.

Memory Samples (N)	**Exemplar Parameters**	**Step Time (ms)**
1,000	0.06M	$27.32_{\pm 0.03}$
5,000	0.32M	$27.60_{\pm 0.02}$
10,000	0.64M	$29.06_{\pm 0.02}$
25,000	1.60M	$28.22_{\pm 0.02}$
50,000	3.20M	$28.41_{\pm 0.04}$
100,000	6.40M	$27.70_{\pm 0.03}$

Inference Cost. Compared to a standard black-box classifier, inference in MUNKEY introduces an additional retrieval stage. The overall complexity can be written as $O(C_{\text{enc}} + N \cdot d_k + C_{\text{pred}})$ for the non-ensemble variants, where C_{enc} is the cost of computing the query key, $N \cdot d_k$ is the similarity search over the memory bank, and C_{pred} is the cost of the final forward pass over the backbone. Operationally, inference proceeds in three steps: (i) encode the query, (ii) retrieve relevant memory entries through similarity search, and (iii) process the retrieved tokens for the final output. The main source of inference overhead compared to a standard ViT is the additional forward pass through the frozen key encoder, while nearest-neighbor retrieval is relatively inexpensive, requiring only a matrix multiplication against the memory bank. This overhead is common to retrieval-based architectures and could be reduced by using a smaller or faster key encoder. While the retrieval term scales with N, it is highly optimized in practice as a matrix similarity operation and can be reduced to sub-linear time through approximate nearest neighbor indexing, as in (Geirhos et al., 2025).

Scaling Considerations. Although our experiments are conducted at benchmark scale, larger deployments could benefit from established retrieval optimizations, such as product quantization (Jegou et al., 2010), approximate nearest-neighbor indexing with FAISS (Johnson et al., 2019), or SSD-based retrieval with DiskANN (Jayaram Subramanya et al., 2019). Geirhos et al. (2025) demonstrate billion-scale retrieval in related memory-based vision systems. This suggests that the core advantages of MUNKEY persist under scale: memory grows predictably, inference can be accelerated with standard retrieval systems, and unlearning remains a lightweight update rather than a new optimization problem. While these techniques are compatible with MUNKEY's retrieval component, their accuracy–latency–storage trade-offs remain unexplored in our setting, and we leave efficient large-scale retrieval to future work.

B.6 Discussion Beyond Transformers

Although MUNKEY is implemented using transformers, the underlying principle of externalizing instance-specific information into editable memory is not inherently transformer-specific. Coupling parametric networks with externally addressable memory predates transformers (A. Graves et al., 2014; Weston et al., 2015) and has also been explored with conventional neural architectures for visual recognition (Santoro et al., 2016).

For instance, a CNN could extract a feature representation $u = f_{\text{CNN}}(x)$ and aggregate retrieved exemplar tokens into $e(x) = \text{Agg}(\{v_j\}_{j \in \mathcal{N}_K(x)})$. A classification head h_{cls} could then combine these representations through concatenation $h_{\text{cls}}([u; e(x)])$, projected addition $h_{\text{cls}}(u + We(x))$, where W maps the exemplar representation to the CNN feature space, or learned gating. However, the main challenge lies not in integrating external memory, but in encouraging the separation of instance-specific information from shared model parameters. In MUNKEY, attention enables interactions between image patches and exemplar tokens, while pathway dropout regularizes their respective contributions. Standard CNNs do not provide an equivalent token-level interaction mechanism, and dropping the retrieved representation at the fusion point provides coarser regularization. Investigating suitable fusion mechanisms and dropout strategies to achieve comparable separation in other architectures remains an interesting direction for future work. Moreover, extending MUNKEY to generative language models would additionally require defining the granularity of removable memories and how retrieved information conditions generation, which we likewise leave to future work.

C Additional Benchmarks

Extended Datasets. In this section, we provide detailed results for the remaining benchmarks, CIFAR-100, BloodMNIST, PathMNIST, and Tiny ImageNet, using the suite of metrics defined in Section 4, comprising Test Accuracy, Retain Accuracy, Forget Accuracy, MIA, and Average Gap to evaluate performance, utility, and forgetting. We note that MUNKEY consistently outperforms all baselines across these datasets, with the exception of PathMNIST in Table 7, where it remains highly competitive and on par with baselines within a standard deviation, as most methods perform highly on this task. BloodMNIST, seen in Table 6, mirrors the performance trends observed for DermaMNIST in the main text, where MUNKEY outperforms baselines in both accuracy and MIA. This is particularly interesting as another example of non-natural image data that differs significantly from the distribution used for the key encoder.

Table 6: Unlearning performance on a random forget set of 10% of the training data on BloodMNIST.

Model	TA	RA	FA	MIA	Avg Gap ↓
Retrain (T)	$96.88_{\pm 0.16}$	$99.99_{\pm 0.00}$	$96.43_{\pm 0.42}$	$49.60_{\pm 0.28}$	–
Retrain (M)	$97.55_{\pm 0.15}$	$99.49_{\pm 0.13}$	$97.21_{\pm 0.22}$	$49.60_{\pm 0.41}$	–
KNN	$93.05_{\pm 0.05}$	$95.23_{\pm 0.04}$	$92.25_{\pm 0.58}$	$49.80_{\pm 0.61}$	–
CF	$96.65_{\pm 0.21}$	$99.74_{\pm 0.12}$	$99.50_{\pm 0.41}$	$54.10_{\pm 0.12}$	$2.01_{\pm 0.18}$
CF-k	$96.93_{\pm 0.25}$	$99.96_{\pm 0.04}$	$100.00_{\pm 0.00}$	$52.50_{\pm 0.47}$	$1.64_{\pm 0.19}$
EU-k	$96.94_{\pm 0.26}$	$99.94_{\pm 0.07}$	$99.97_{\pm 0.04}$	$52.90_{\pm 0.42}$	$1.74_{\pm 0.18}$
ℓ_1-sparse	$90.92_{\pm 1.65}$	$92.14_{\pm 1.87}$	$91.55_{\pm 1.48}$	$50.99_{\pm 0.62}$	$5.02_{\pm 0.75}$
NegGrad+	$96.48_{\pm 0.25}$	$99.77_{\pm 0.05}$	$97.80_{\pm 0.04}$	$51.45_{\pm 0.56}$	$\underline{0.96}_{\pm 0.20}$
Amnesiac	$96.27_{\pm 0.47}$	$99.63_{\pm 0.42}$	$97.49_{\pm 0.43}$	$40.85_{\pm 1.57}$	$2.70_{\pm 0.46}$
SCRUB	$95.44_{\pm 0.52}$	$97.27_{\pm 0.88}$	$96.52_{\pm 0.76}$	$51.01_{\pm 0.25}$	$1.42_{\pm 0.35}$
SSD	$96.84_{\pm 0.28}$	$99.88_{\pm 0.16}$	$99.90_{\pm 0.10}$	$51.72_{\pm 0.25}$	$1.44_{\pm 0.17}$
AMUN	$96.34_{\pm 0.08}$	$99.61_{\pm 0.19}$	$98.94_{\pm 0.34}$	$51.92_{\pm 0.55}$	$1.44_{\pm 0.22}$
CoUn	$97.90_{\pm 0.23}$	$99.99_{\pm 0.01}$	$98.49_{\pm 0.20}$	$51.30_{\pm 0.74}$	$1.20_{\pm 0.24}$
MUNKEY	$97.72_{\pm 0.08}$	$99.63_{\pm 0.12}$	$99.50_{\pm 0.18}$	$50.66_{\pm 0.50}$	$\mathbf{0.92}_{\pm 0.19}$

Table 7: Unlearning performance on a random forget set of 10% of the training data on PathMNIST.

Model	TA	RA	FA	MIA	Avg Gap ↓
Retrain (T)	$98.64_{\pm 0.01}$	$100.00_{\pm 0.00}$	$98.53_{\pm 0.05}$	$49.95_{\pm 0.15}$	–
Retrain (M)	$98.96_{\pm 0.07}$	$100.00_{\pm 0.00}$	$98.78_{\pm 0.17}$	$50.09_{\pm 0.24}$	–
KNN	$96.69_{\pm 0.08}$	$97.10_{\pm 0.01}$	$96.81_{\pm 0.17}$	$50.27_{\pm 0.17}$	–
CF	$98.30_{\pm 0.08}$	$99.87_{\pm 0.08}$	$99.77_{\pm 0.12}$	$51.31_{\pm 0.27}$	$0.77_{\pm 0.09}$
CF-k	$98.56_{\pm 0.08}$	$99.99_{\pm 0.00}$	$100.00_{\pm 0.01}$	$51.89_{\pm 0.24}$	$0.88_{\pm 0.07}$
EU-k	$98.54_{\pm 0.04}$	$99.99_{\pm 0.00}$	$100.00_{\pm 0.01}$	$51.92_{\pm 0.26}$	$0.89_{\pm 0.08}$
ℓ_1-sparse	$92.12_{\pm 0.70}$	$92.43_{\pm 0.66}$	$91.89_{\pm 0.65}$	$49.88_{\pm 0.24}$	$5.20_{\pm 0.30}$
NegGrad+	$97.92_{\pm 0.25}$	$99.80_{\pm 0.11}$	$98.26_{\pm 0.19}$	$50.74_{\pm 0.12}$	$\mathbf{0.50}_{\pm 0.10}$
Amnesiac	$97.79_{\pm 0.16}$	$99.93_{\pm 0.02}$	$94.14_{\pm 0.23}$	$32.59_{\pm 1.51}$	$5.67_{\pm 0.39}$
SCRUB	$98.08_{\pm 0.15}$	$99.69_{\pm 0.07}$	$99.33_{\pm 0.12}$	$51.19_{\pm 0.16}$	$0.73_{\pm 0.07}$
SSD	$98.31_{\pm 0.20}$	$99.95_{\pm 0.04}$	$99.96_{\pm 0.03}$	$51.49_{\pm 0.36}$	$0.84_{\pm 0.11}$
AMUN	$98.16_{\pm 0.00}$	$99.87_{\pm 0.03}$	$99.71_{\pm 0.02}$	$51.14_{\pm 0.18}$	$0.75_{\pm 0.06}$
CoUn	$98.29_{\pm 0.67}$	$99.90_{\pm 0.03}$	$99.38_{\pm 0.05}$	$51.90_{\pm 0.57}$	$0.81_{\pm 0.22}$
MUNKEY	$98.75_{\pm 0.12}$	$100.00_{\pm 0.00}$	$99.95_{\pm 0.02}$	$50.77_{\pm 0.08}$	$\underline{0.52}_{\pm 0.08}$

Results on CIFAR-100 and Tiny ImageNet, shown in Table 8 and Table 9, respectively, highlight our performance on fine-grained data with a high number of classes, a generally more challenging setting where MUNKEY maintains a superior base performance and utility boost (in task accuracy) thanks to its memory-augmented architecture. While we evaluate final unlearning based on the Average Gap, a fair metric with respect to the baselines since their Retrain (T) oracle also lacks the external memory, it is notable that the memory bank further enhances absolute accuracies. Tiny ImageNet serves as an extreme case example where the data has been seen, in a different resolution, by the pre-trained key encoder; we show that MUNKEY is able to outperform in this setting as well. The relatively low baseline accuracies here are due to the lack of extensive data augmentation, as our primary objective is evaluating unlearning capabilities rather than maximizing task accuracy. Notably, MUNKEY still

performs better than the KNN baseline on Tiny ImageNet, demonstrating that the backbone encoding effectively improves performance and makes MUNKEY behave as "the best of both worlds". To ensure clarity regarding our unlearning claims in this last dataset, we emphasize that it occurs at the output level and distinguish between two levels of membership: pre-training membership, where the key encoder may have seen an image during its initial training, is considered out of scope for our algorithm; however, task-specific membership, which refers to whether an image was used to train the current model for its specific task, is exactly what we target for unlearning. This distinction is critical as we assume a practical deployment setting where a black-box adversary only has access to the model's output space; by utilizing the black-box membership inference attack at the output level as claimed throughout this work, we verify that task-specific influence is successfully neutralized.

Table 8: Unlearning performance on a random forget set of 10% of the training data on CIFAR-100.

Model	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**
Retrain (T)	$47.17_{\pm 0.61}$	$87.37_{\pm 0.41}$	$47.00_{\pm 0.39}$	$49.88_{\pm 0.26}$	–
Retrain (M)	$76.19_{\pm 0.48}$	$99.84_{\pm 0.05}$	$76.05_{\pm 0.17}$	$49.91_{\pm 0.30}$	–
KNN	$71.86_{\pm 0.31}$	$81.67_{\pm 0.07}$	$71.65_{\pm 0.36}$	$49.88_{\pm 0.35}$	–
CF	$46.69_{\pm 1.04}$	$90.68_{\pm 0.67}$	$76.94_{\pm 0.95}$	$68.79_{\pm 0.17}$	$13.16_{\pm 0.45}$
CF-k	$49.03_{\pm 0.48}$	$94.71_{\pm 0.19}$	$94.51_{\pm 0.44}$	$77.88_{\pm 0.45}$	$21.18_{\pm 0.30}$
EU-k	$49.04_{\pm 0.56}$	$92.75_{\pm 0.26}$	$92.61_{\pm 0.41}$	$77.58_{\pm 0.42}$	$20.14_{\pm 0.30}$
ℓ_1-sparse	$33.40_{\pm 0.58}$	$35.49_{\pm 0.77}$	$33.48_{\pm 1.22}$	$50.12_{\pm 0.79}$	$19.85_{\pm 0.49}$
NegGrad+	$46.70_{\pm 0.57}$	$89.80_{\pm 0.13}$	$61.94_{\pm 0.71}$	$60.50_{\pm 0.45}$	$\underline{7.12}_{\pm 0.34}$
Amnesiac	$46.28_{\pm 0.64}$	$90.99_{\pm 0.43}$	$67.50_{\pm 0.81}$	$62.64_{\pm 0.53}$	$9.44_{\pm 0.38}$
SCRUB	$47.47_{\pm 0.64}$	$84.93_{\pm 0.15}$	$76.73_{\pm 0.79}$	$68.70_{\pm 0.45}$	$12.82_{\pm 0.36}$
SSD	$46.60_{\pm 1.69}$	$89.58_{\pm 3.41}$	$87.55_{\pm 3.93}$	$76.23_{\pm 1.52}$	$17.42_{\pm 1.44}$
AMUN	$46.82_{\pm 1.26}$	$90.04_{\pm 1.22}$	$75.09_{\pm 1.06}$	$67.38_{\pm 0.45}$	$12.15_{\pm 0.57}$
CoUn	$50.39_{\pm 0.46}$	$99.98_{\pm 0.00}$	$81.01_{\pm 0.54}$	$65.87_{\pm 0.11}$	$16.46_{\pm 0.28}$
MUNKEY	$76.62_{\pm 0.51}$	$99.85_{\pm 0.03}$	$80.80_{\pm 0.36}$	$53.31_{\pm 0.11}$	$\mathbf{2.15}_{\pm 0.22}$

Table 9: Unlearning performance on a random forget set of 10% of the training data on Tiny ImageNet.

Model	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**
Retrain (T)	$36.74_{\pm 0.76}$	$76.78_{\pm 0.29}$	$36.30_{\pm 0.49}$	$49.87_{\pm 0.23}$	–
Retrain (M)	$70.63_{\pm 0.52}$	$99.94_{\pm 0.00}$	$70.62_{\pm 0.43}$	$50.10_{\pm 0.42}$	–
KNN	$66.56_{\pm 0.31}$	$77.46_{\pm 0.08}$	$66.78_{\pm 0.52}$	$49.99_{\pm 0.24}$	–
CF	$35.46_{\pm 0.37}$	$88.20_{\pm 0.57}$	$68.24_{\pm 0.97}$	$72.71_{\pm 0.25}$	$16.87_{\pm 0.39}$
CF-k	$36.98_{\pm 0.55}$	$92.58_{\pm 0.32}$	$92.26_{\pm 0.54}$	$83.12_{\pm 0.36}$	$26.31_{\pm 0.33}$
EU-k	$36.84_{\pm 0.40}$	$88.64_{\pm 0.33}$	$88.01_{\pm 0.60}$	$81.54_{\pm 0.09}$	$23.84_{\pm 0.32}$
ℓ_1-sparse	$21.19_{\pm 0.63}$	$22.26_{\pm 0.53}$	$21.23_{\pm 0.46}$	$50.37_{\pm 0.15}$	$21.41_{\pm 0.34}$
NegGrad+	$34.73_{\pm 0.25}$	$86.54_{\pm 0.61}$	$49.34_{\pm 0.21}$	$62.31_{\pm 0.05}$	$\underline{9.31}_{\pm 0.30}$
Amnesiac	$34.94_{\pm 0.45}$	$88.36_{\pm 0.34}$	$58.47_{\pm 0.31}$	$68.11_{\pm 0.21}$	$13.45_{\pm 0.30}$
SCRUB	$36.28_{\pm 0.37}$	$86.10_{\pm 1.03}$	$73.64_{\pm 0.69}$	$74.94_{\pm 0.33}$	$18.05_{\pm 0.41}$
SSD	$36.48_{\pm 0.60}$	$91.01_{\pm 0.41}$	$90.59_{\pm 0.49}$	$82.86_{\pm 0.35}$	$25.44_{\pm 0.34}$
AMUN	$35.32_{\pm 1.17}$	$86.86_{\pm 1.22}$	$65.30_{\pm 1.40}$	$70.80_{\pm 0.20}$	$15.36_{\pm 0.60}$
CoUn	$36.86_{\pm 0.31}$	$99.98_{\pm 0.00}$	$69.92_{\pm 0.12}$	$65.62_{\pm 0.25}$	$18.17_{\pm 0.27}$
MUNKEY	$70.86_{\pm 0.42}$	$99.94_{\pm 0.00}$	$71.73_{\pm 0.51}$	$51.19_{\pm 0.30}$	$\mathbf{0.61}_{\pm \mathbf{0.27}}$

Stress-Test Evaluation with Larger Forget Sets. To evaluate whether the proposed unlearning by design paradigm remains effective beyond standard settings, we report results on the primary benchmarks under 50% random forgetting in Table 10, providing a stress test of utility preservation and forgetting quality.

Table 10: Comparison of MUNKEY and studied baselines using ViT-Tiny backbone on a random forget set of 50% and three seeds on DermaMNIST and CIFAR-10.

	DermaMNIST					CIFAR-10				
Model	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**
Retrain (T)	$73.60_{\pm 0.47}$	$99.98_{\pm 0.03}$	$73.39_{\pm 0.46}$	$50.09_{\pm 0.61}$	–	$71.62_{\pm 0.27}$	$87.57_{\pm 0.71}$	$71.16_{\pm 0.44}$	$49.80_{\pm 0.28}$	–
Retrain (M)	$78.84_{\pm 0.51}$	$99.33_{\pm 0.38}$	$77.70_{\pm 0.46}$	$49.86_{\pm 0.36}$	–	$91.32_{\pm 0.13}$	$99.76_{\pm 0.05}$	$91.41_{\pm 0.02}$	$49.73_{\pm 0.18}$	–
KNN	$73.82_{\pm 0.00}$	$83.19_{\pm 0.00}$	$71.77_{\pm 0.00}$	$50.22_{\pm 0.00}$	–	$90.57_{\pm 0.04}$	$94.18_{\pm 0.09}$	$90.85_{\pm 0.06}$	$50.36_{\pm 0.13}$	–
CF	$77.19_{\pm 0.60}$	$100.0_{\pm 0.00}$	$99.75_{\pm 0.08}$	$67.02_{\pm 0.15}$	$11.73_{\pm 0.27}$	$79.82_{\pm 0.31}$	$97.08_{\pm 0.43}$	$93.38_{\pm 0.54}$	$59.54_{\pm 1.33}$	$12.42_{\pm 0.45}$
CF-k	$77.04_{\pm 0.42}$	$99.77_{\pm 0.17}$	$99.83_{\pm 0.12}$	$67.28_{\pm 0.13}$	$11.82_{\pm 0.26}$	$79.44_{\pm 0.66}$	$94.13_{\pm 0.54}$	$94.15_{\pm 0.64}$	$60.36_{\pm 1.29}$	$11.98_{\pm 0.48}$
EU-k	$72.59_{\pm 2.35}$	$88.57_{\pm 3.55}$	$88.67_{\pm 3.46}$	$56.89_{\pm 0.62}$	$8.63_{\pm 1.40}$	$79.08_{\pm 1.09}$	$93.68_{\pm 0.61}$	$93.77_{\pm 0.63}$	$59.67_{\pm 0.68}$	$11.51_{\pm 0.45}$
ℓ_1-sparse	$76.33_{\pm 0.12}$	$95.24_{\pm 0.37}$	$89.69_{\pm 0.52}$	$59.69_{\pm 0.41}$	$8.34_{\pm 0.30}$	$25.16_{\pm 0.52}$	$24.09_{\pm 0.44}$	$23.90_{\pm 0.33}$	$50.47_{\pm 0.57}$	$39.47_{\pm 0.33}$
NegGrad+	$77.09_{\pm 0.50}$	$100.0_{\pm 0.00}$	$96.02_{\pm 0.37}$	$64.75_{\pm 0.45}$	$10.20_{\pm 0.30}$	$79.38_{\pm 0.71}$	$96.71_{\pm 0.43}$	$91.51_{\pm 0.58}$	$58.79_{\pm 1.24}$	$11.56_{\pm 0.46}$
Amnesiac	$76.09_{\pm 0.17}$	$100.0_{\pm 0.00}$	$98.10_{\pm 0.30}$	$60.70_{\pm 0.99}$	$9.46_{\pm 0.35}$	$79.11_{\pm 0.56}$	$97.01_{\pm 0.48}$	$91.74_{\pm 0.61}$	$58.57_{\pm 0.98}$	$11.57_{\pm 0.41}$
SCRUB	$69.79_{\pm 0.70}$	$69.71_{\pm 1.24}$	$69.35_{\pm 0.88}$	$50.61_{\pm 0.39}$	$9.66_{\pm 0.48}$	$76.75_{\pm 1.41}$	$85.24_{\pm 1.96}$	$84.13_{\pm 2.09}$	$55.62_{\pm 0.90}$	$\underline{6.56}_{\pm 0.86}$
SSD	$76.76_{\pm 0.49}$	$99.62_{\pm 0.30}$	$99.70_{\pm 0.23}$	$63.35_{\pm 0.39}$	$10.77_{\pm 0.29}$	$79.03_{\pm 1.04}$	$93.61_{\pm 0.58}$	$93.64_{\pm 0.65}$	$58.95_{\pm 0.10}$	$11.27_{\pm 0.41}$
AMUN	$76.96_{\pm 0.25}$	$100.0_{\pm 0.00}$	$99.49_{\pm 0.28}$	$66.92_{\pm 0.12}$	$11.58_{\pm 0.25}$	$78.18_{\pm 0.42}$	$94.79_{\pm 0.22}$	$89.07_{\pm 0.29}$	$57.11_{\pm 0.70}$	$9.75_{\pm 0.23}$
CoUn	$78.05_{\pm 0.77}$	$99.84_{\pm 0.05}$	$86.11_{\pm 0.38}$	$53.80_{\pm 0.40}$	$\underline{5.26}_{\pm 0.33}$	$79.24_{\pm 0.63}$	$100.0_{\pm 0.00}$	$90.98_{\pm 0.55}$	$57.15_{\pm 0.32}$	$11.81_{\pm 0.32}$
MUNKEY	$80.17_{\pm 0.76}$	$99.78_{\pm 0.07}$	$81.82_{\pm 0.28}$	$51.48_{\pm 0.66}$	$\mathbf{1.88}_{\pm 0.34}$	$92.01_{\pm 0.07}$	$99.94_{\pm 0.01}$	$92.60_{\pm 0.13}$	$50.45_{\pm 0.07}$	$\mathbf{0.70}_{\pm 0.07}$

Effect of Forget-Set Structure. Beyond varying the fraction of forgotten random samples, we investigate the effect of different forget-set structures on CIFAR-10. We consider three settings: (i) full-class removal, where all 5,000 training instances of one class are forgotten, averaged over the 10 classes; (ii) class-imbalanced forgetting, where 80% of a 5,000-sample forget set comes from one class and the remaining 20% from the other nine classes, again repeated over all classes; and (iii) single-instance removal, evaluated over 20 independent leave-one-out experiments. Each run is compared against its corresponding retain-only retraining oracle, with class-matched members and non-members for membership inference to avoid class-composition biases. Across the different forget-set structures, MUNKEY remains close to its corresponding retraining oracle. In particular, full-class removal leads to a substantial reduction in forget accuracy, closely matching retraining, and class-imbalanced forgetting achieves a small Avg Gap of 0.95. For single-instance removal, where Forget Accuracy (FA) denotes the number of correctly classified deleted samples across 20 leave-one-out runs, membership is evaluated using the deleted sample's percentile within a same-class panel of 50 examples, rather than MIA AUC, with 50 corresponding to chance. Overall, these results support the effectiveness of MUNKEY across different deletion structures.

Table 11: Effect of forget-set structure on CIFAR-10. Full-class and class-imbalanced removal use 5,000 forgotten samples, while single-instance removal uses one. Each method is evaluated against its corresponding retraining oracle.

Forget set	**Model**	**TA**	**RA**	**FA**	**Membership**	**Avg Gap**
Full-class	**MUNKEY**	$84.53_{\pm 0.59}$	$99.94_{\pm 0.00}$	$3.70_{\pm 1.89}$	$50.92_{\pm 3.12}$	$\mathbf{1.37}_{\pm 0.99}$
	Retrain (M)	$84.13_{\pm 0.77}$	$99.95_{\pm 0.02}$	$0.00_{\pm 0.00}$	$49.56_{\pm 1.40}$	–
80/20 imbalance	**MUNKEY**	$92.16_{\pm 0.36}$	$99.94_{\pm 0.00}$	$86.57_{\pm 6.19}$	$49.39_{\pm 2.20}$	$\mathbf{0.95}_{\pm 0.68}$
	Retrain (M)	$91.79_{\pm 0.41}$	$99.94_{\pm 0.02}$	$83.48_{\pm 6.92}$	$49.07_{\pm 1.48}$	–
Single-instance	**MUNKEY**	$92.75_{\pm 0.00}$	$99.94_{\pm 0.00}$	19/20	53.9	–
	Retrain (M)	$92.80_{\pm 0.11}$	$99.94_{\pm 0.01}$	18/20	49.7	–

D Robustness Analysis

D.1 Effect of Pathway Aggregation

As discussed in Section 5, the modular nature of the ViT architecture allows for multiple ways to integrate information from the exemplar tokens and the image patches. In this section, we provide a detailed investigation into the architectural paradigms for combining the image sequence $\boldsymbol{z}_i$ with the exemplar token $\boldsymbol{v}_i'$, as well as the late-fusion strategies of neighbors employed at inference time. We primarily investigate four approaches: (1) **Input-level Aggregation**, our default strategy where $\boldsymbol{v}_i'$ is simply prepended to the input sequence as $\boldsymbol{z}_{in} = [\boldsymbol{z}_i, \boldsymbol{v}_i']$; (2) **Deep Prompt** (M. Jia et al., 2022), where the token is re-inserted as a prefix before every transformer block to allow for multi-scale conditioning; (3) **Cross Attention**, which involves inserting a dedicated layer after the self-attention blocks in the beginning, middle, and last layers to derive keys and values from $\boldsymbol{v}_i'$ while using $\boldsymbol{z}_i$ as the query; and (4) **Direct Key**, a baseline where the fixed key from the frozen encoder is directly projected into the input space.

During inference, when multiple neighbors $\mathcal{N}_K(\boldsymbol{x}_i)$ are retrieved from the memory bank, we must determine how to aggregate their information. One approach is **softmax-weighted average**, where we compute a softmax over the cosine similarity scores of the retrieved keys to form a single weighted average token. Alternatively, **rank-based weighting** derives these weights purely from the ranking of the neighbors to provide a more stable distribution independent of absolute similarity values. Finally, **ensemble voting** serves as a more robust but computationally intensive method, requiring K separate inference passes to aggregate the resulting logits.

An evaluation across the six studied datasets in Tables 12 to 17 shows that while trends remain largely similar, the choice of strategy significantly impacts the modularity of the model's information flow measured in P_s, introduced in Section 3. We highlight P_s and Avg Gap as they are the center of focus in this discussion. We observe that cross attention generally achieves a low Avg Gap but suffers from high P_s, indicating an unhealthy over-reliance on specific pathways despite the added architectural complexity. Moreover, the direct key approach often leads not only to high P_s but also to larger Avg Gaps, which empirically motivates the necessity of the learnable exemplar token in our framework. While deep prompting shows promise, it occasionally exhibits high P_s on datasets like Tiny ImageNet and generally yields slightly lower utility than late-fusion methods. Consequently, we adopt the simpler input-level aggregation as our primary method, as it offers the best trade-off between unlearning, simplicity, and balanced feature integration. Finally, regarding late-fusion, all perform generally similarly, with Ensembling providing the smallest Avg Gap and most stable pathway balance. However, if inference time or compute is a constraint, we argue that the Softmax-weighted average performs closely to ensembling and remains superior to standard baselines while being more computationally efficient. For simplicity and ease of exploration and compute requirements, all results shown in this section are evaluated with a simpler membership inference attack that does not require training a large number of shadow models per model variant. This MIA uses a logistic regression classifier on the model's output space, and is trained on the loss, entropy, confidence, and prediction margin to distinguish between train and test samples. As in the main text, an MIA AUROC near 0.5 on $\mathcal{D}_f$ indicates successful unlearning.

Table 12: Results on the effect of pathway aggregation and late-fusion of neighbors on DermaMNIST.

Category	Model Variant	TA	RA	FA	MIA	P_s	Avg Gap ↓
Retrained	Direct Key	$81.83_{\pm 0.68}$	$99.25_{\pm 0.22}$	$80.05_{\pm 1.10}$	$48.30_{\pm 0.98}$	–	–
	Deep Prompt	$79.57_{\pm 0.43}$	$93.85_{\pm 1.30}$	$79.10_{\pm 0.44}$	$50.34_{\pm 0.84}$	–	–
	Cross Attention	$80.62_{\pm 0.13}$	$97.23_{\pm 0.05}$	$80.33_{\pm 1.17}$	$50.31_{\pm 0.65}$	–	–
	Ensembling	$80.80_{\pm 0.54}$	$97.47_{\pm 0.15}$	$80.10_{\pm 0.99}$	$50.53_{\pm 0.89}$	–	–
	Softmax	$79.70_{\pm 0.82}$	$93.37_{\pm 1.75}$	$78.95_{\pm 0.18}$	$50.44_{\pm 0.77}$	–	–
	Rank	$79.32_{\pm 0.58}$	$88.34_{\pm 1.37}$	$78.57_{\pm 0.35}$	$48.90_{\pm 0.93}$	–	–
MUNKEY	Direct Key	$81.70_{\pm 0.74}$	$99.30_{\pm 0.45}$	$99.33_{\pm 0.64}$	$61.76_{\pm 2.00}$	$0.2118_{\pm 0.0120}$	$8.23_{\pm 0.70}$
	Deep Prompt	$79.60_{\pm 0.60}$	$93.32_{\pm 1.87}$	$83.57_{\pm 0.53}$	$52.47_{\pm 0.88}$	$0.1724_{\pm 0.0058}$	$1.79_{\pm 0.69}$
	Cross Attention	$80.47_{\pm 0.28}$	$97.17_{\pm 0.17}$	$83.43_{\pm 0.95}$	$51.08_{\pm 0.75}$	$0.2076_{\pm 0.0047}$	$\mathbf{1.02}_{\pm \mathbf{0.46}}$
	Ensembling	$81.06_{\pm 0.41}$	$97.64_{\pm 0.22}$	$84.10_{\pm 1.58}$	$52.16_{\pm 0.53}$	$0.1690_{\pm 0.0037}$	$\underline{1.52}_{\pm 0.56}$
	Softmax	$79.83_{\pm 1.33}$	$93.14_{\pm 2.44}$	$84.29_{\pm 0.47}$	$52.76_{\pm 0.58}$	$0.1681_{\pm 0.0055}$	$2.01_{\pm 0.89}$
	Rank	$79.65_{\pm 0.86}$	$88.42_{\pm 2.10}$	$84.38_{\pm 0.99}$	$53.00_{\pm 0.56}$	$0.1657_{\pm 0.0062}$	$2.58_{\pm 0.78}$

Table 13: Results on the effect of pathway aggregation and late-fusion of neighbors on CIFAR-10.

Category	Model Variant	TA	RA	FA	MIA	P_s	Avg Gap ↓
Retrained	Direct Key	$92.41_{\pm 0.16}$	$99.56_{\pm 0.07}$	$92.46_{\pm 0.10}$	$50.11_{\pm 0.58}$	–	–
	Deep Prompt	$91.56_{\pm 1.02}$	$99.76_{\pm 0.14}$	$91.32_{\pm 1.05}$	$50.22_{\pm 0.53}$	–	–
	Cross Attention	$92.79_{\pm 0.02}$	$100.0_{\pm 0.00}$	$92.71_{\pm 0.19}$	$50.13_{\pm 0.19}$	–	–
	Ensembling	$92.93_{\pm 0.16}$	$99.93_{\pm 0.02}$	$92.76_{\pm 0.45}$	$49.97_{\pm 0.25}$	–	–
	Softmax	$92.53_{\pm 0.14}$	$99.85_{\pm 0.05}$	$92.29_{\pm 0.46}$	$49.95_{\pm 0.20}$	–	–
	Rank	$92.36_{\pm 0.03}$	$95.69_{\pm 0.07}$	$92.12_{\pm 0.33}$	$49.86_{\pm 0.14}$	–	–
MUNKEY	Direct Key	$92.08_{\pm 0.10}$	$99.47_{\pm 0.12}$	$99.35_{\pm 0.18}$	$56.16_{\pm 0.32}$	$0.2385_{\pm 0.0068}$	$3.34_{\pm 0.18}$
	Deep Prompt	$92.09_{\pm 0.83}$	$99.80_{\pm 0.13}$	$92.99_{\pm 1.04}$	$51.64_{\pm 0.19}$	$0.1880_{\pm 0.0063}$	$0.92_{\pm 0.52}$
	Cross Attention	$92.93_{\pm 0.01}$	$99.99_{\pm 0.00}$	$93.15_{\pm 0.26}$	$50.69_{\pm 0.38}$	$0.2921_{\pm 0.0052}$	$\mathbf{0.29}_{\pm \mathbf{0.13}}$
	Ensembling	$93.12_{\pm 0.08}$	$99.94_{\pm 0.01}$	$93.41_{\pm 0.58}$	$51.45_{\pm 0.39}$	$0.1842_{\pm 0.0060}$	$\underline{0.58}_{\pm 0.22}$
	Softmax	$92.48_{\pm 0.19}$	$99.80_{\pm 0.06}$	$93.46_{\pm 0.64}$	$51.51_{\pm 0.40}$	$0.1857_{\pm 0.0056}$	$0.71_{\pm 0.23}$
	Rank	$92.38_{\pm 0.43}$	$95.76_{\pm 0.23}$	$93.22_{\pm 0.42}$	$51.33_{\pm 0.40}$	$0.1838_{\pm 0.0060}$	$0.67_{\pm 0.21}$

Table 14: Results on the effect of pathway aggregation and late-fusion of neighbors on BloodMNIST.

Category	Model Variant	TA	RA	FA	MIA	P_s	Avg Gap ↓
Retrained	Direct Key	$97.97_{\pm 0.12}$	$99.93_{\pm 0.03}$	$98.38_{\pm 0.48}$	$49.25_{\pm 0.66}$	–	–
	Deep Prompt	$97.52_{\pm 0.27}$	$99.55_{\pm 0.18}$	$97.10_{\pm 0.26}$	$49.29_{\pm 1.04}$	–	–
	Cross Attention	$97.36_{\pm 0.04}$	$98.57_{\pm 0.12}$	$97.27_{\pm 0.42}$	$49.73_{\pm 0.25}$	–	–
	Ensembling	$97.55_{\pm 0.15}$	$99.49_{\pm 0.13}$	$97.21_{\pm 0.22}$	$49.60_{\pm 0.41}$	–	–
	Softmax	$97.62_{\pm 0.21}$	$99.51_{\pm 0.19}$	$97.49_{\pm 0.24}$	$49.63_{\pm 0.72}$	–	–
	Rank	$97.56_{\pm 0.10}$	$99.62_{\pm 0.15}$	$97.27_{\pm 0.10}$	$49.53_{\pm 0.49}$	–	–
MUNKEY	Direct Key	$98.13_{\pm 0.15}$	$99.94_{\pm 0.02}$	$99.97_{\pm 0.04}$	$51.43_{\pm 0.89}$	$0.0273_{\pm 0.0080}$	$0.99_{\pm 0.31}$
	Deep Prompt	$97.50_{\pm 0.42}$	$99.71_{\pm 0.09}$	$99.55_{\pm 0.28}$	$51.21_{\pm 0.70}$	$0.0388_{\pm 0.0110}$	$1.14_{\pm 0.35}$
	Cross Attention	$97.66_{\pm 0.06}$	$98.61_{\pm 0.05}$	$97.94_{\pm 0.28}$	$50.26_{\pm 0.17}$	$0.0432_{\pm 0.0044}$	$\mathbf{0.39}_{\pm 0.15}$
	Ensembling	$97.72_{\pm 0.08}$	$99.63_{\pm 0.12}$	$99.50_{\pm 0.18}$	$50.66_{\pm 0.50}$	$0.0435_{\pm 0.0067}$	$\underline{0.92}_{\pm 0.19}$
	Softmax	$97.60_{\pm 0.13}$	$99.73_{\pm 0.07}$	$99.55_{\pm 0.04}$	$51.11_{\pm 0.51}$	$0.0243_{\pm 0.0109}$	$0.95_{\pm 0.24}$
	Rank	$97.73_{\pm 0.29}$	$99.70_{\pm 0.09}$	$99.47_{\pm 0.16}$	$50.99_{\pm 0.66}$	$0.0441_{\pm 0.0133}$	$0.98_{\pm 0.23}$

Table 15: Results on the effect of pathway aggregation and late-fusion of neighbors on PathMNIST.

Category	Model Variant	TA	RA	FA	MIA	P_s	Avg Gap ↓
Retrained	Direct Key	$98.98_{\pm 0.04}$	$100.0_{\pm 0.00}$	$98.94_{\pm 0.11}$	$49.88_{\pm 0.30}$	–	–
	Deep Prompt	$99.04_{\pm 0.03}$	$100.0_{\pm 0.00}$	$98.90_{\pm 0.11}$	$49.98_{\pm 0.25}$	–	–
	Cross Attention	$98.68_{\pm 0.07}$	$99.97_{\pm 0.01}$	$98.55_{\pm 0.12}$	$49.98_{\pm 0.07}$	–	–
	Ensembling	$98.96_{\pm 0.07}$	$100.00_{\pm 0.00}$	$98.78_{\pm 0.17}$	$50.09_{\pm 0.24}$	–	–
	Softmax	$99.24_{\pm 0.08}$	$100.0_{\pm 0.00}$	$99.23_{\pm 0.05}$	$49.74_{\pm 0.11}$	–	–
	Rank	$99.05_{\pm 0.04}$	$100.0_{\pm 0.00}$	$98.91_{\pm 0.10}$	$49.86_{\pm 0.22}$	–	–
MUNKEY	Direct Key	$98.99_{\pm 0.05}$	$100.0_{\pm 0.00}$	$100.0_{\pm 0.01}$	$51.22_{\pm 0.22}$	$0.0206_{\pm 0.0265}$	$0.60_{\pm 0.10}$
	Deep Prompt	$99.03_{\pm 0.05}$	$100.0_{\pm 0.00}$	$100.0_{\pm 0.01}$	$51.17_{\pm 0.30}$	$0.0636_{\pm 0.0201}$	$0.58_{\pm 0.10}$
	Cross Attention	$98.76_{\pm 0.01}$	$99.97_{\pm 0.01}$	$99.97_{\pm 0.02}$	$50.81_{\pm 0.14}$	$0.0040_{\pm 0.0006}$	$0.58_{\pm 0.05}$
	Ensembling	$98.75_{\pm 0.12}$	$100.00_{\pm 0.00}$	$99.95_{\pm 0.02}$	$50.77_{\pm 0.08}$	$0.0743_{\pm 0.0050}$	$\underline{0.52}_{\pm 0.08}$
	Softmax	$99.30_{\pm 0.03}$	$100.0_{\pm 0.00}$	$100.0_{\pm 0.00}$	$50.74_{\pm 0.24}$	$0.0049_{\pm 0.0010}$	$\mathbf{0.46}_{\pm \mathbf{0.07}}$
	Rank	$99.08_{\pm 0.05}$	$100.0_{\pm 0.00}$	$100.0_{\pm 0.01}$	$51.43_{\pm 0.13}$	$0.0737_{\pm 0.0180}$	$0.67_{\pm 0.07}$

Table 16: Results on the effect of pathway aggregation and late-fusion of neighbors on CIFAR-100.

Category	Model Variant	TA	RA	FA	MIA	P_s	Avg Gap ↓
Retrained	Direct Key	$75.40_{\pm 0.47}$	$99.48_{\pm 0.33}$	$74.77_{\pm 0.53}$	$49.44_{\pm 0.66}$	–	–
	Deep Prompt	$73.31_{\pm 0.78}$	$99.77_{\pm 0.05}$	$73.44_{\pm 0.69}$	$49.91_{\pm 0.27}$	–	–
	Cross Attention	$76.23_{\pm 0.35}$	$99.92_{\pm 0.01}$	$76.17_{\pm 0.30}$	$50.02_{\pm 0.27}$	–	–
	Ensembling	$76.19_{\pm 0.48}$	$99.84_{\pm 0.05}$	$76.05_{\pm 0.17}$	$49.91_{\pm 0.30}$	–	–
	Softmax	$73.28_{\pm 0.55}$	$99.76_{\pm 0.07}$	$73.24_{\pm 0.43}$	$49.88_{\pm 0.03}$	–	–
	Rank	$72.94_{\pm 0.50}$	$87.11_{\pm 0.46}$	$72.58_{\pm 0.40}$	$49.85_{\pm 0.21}$	–	–
MUNKEY	Direct Key	$75.92_{\pm 0.46}$	$99.46_{\pm 0.51}$	$99.48_{\pm 0.50}$	$69.04_{\pm 0.49}$	$0.4162_{\pm 0.0285}$	$11.21_{\pm 0.35}$
	Deep Prompt	$73.49_{\pm 0.69}$	$99.74_{\pm 0.05}$	$81.99_{\pm 0.20}$	$55.04_{\pm 0.52}$	$0.3084_{\pm 0.0008}$	$3.47_{\pm 0.35}$
	Cross Attention	$76.42_{\pm 0.45}$	$99.91_{\pm 0.01}$	$78.86_{\pm 0.26}$	$51.63_{\pm 0.22}$	$0.5019_{\pm 0.0055}$	$\mathbf{1.13}_{\pm \mathbf{0.19}}$
	Ensembling	$76.62_{\pm 0.51}$	$99.85_{\pm 0.03}$	$80.80_{\pm 0.36}$	$53.31_{\pm 0.11}$	$0.2993_{\pm 0.0053}$	$\underline{2.15}_{\pm 0.22}$
	Softmax	$73.32_{\pm 1.2}$	$93.19_{\pm 4.63}$	$79.77_{\pm 1.69}$	$54.20_{\pm 0.43}$	$0.2966_{\pm 0.0001}$	$4.37_{\pm 1.29}$
	Rank	$73.49_{\pm 0.82}$	$87.63_{\pm 0.27}$	$81.74_{\pm 0.37}$	$55.46_{\pm 0.59}$	$0.3058_{\pm 0.0028}$	$3.96_{\pm 0.34}$

Table 17: Results on the effect of pathway aggregation and late-fusion of neighbors on Tiny ImageNet.

Category	Model Variant	TA	RA	FA	MIA	P_s	Avg Gap ↓
Retrained	Direct Key	$68.98_{\pm 0.29}$	$99.98_{\pm 0.00}$	$68.34_{\pm 0.70}$	$49.93_{\pm 0.51}$	–	–
	Deep Prompt	$67.75_{\pm 0.53}$	$99.84_{\pm 0.02}$	$67.68_{\pm 0.59}$	$49.83_{\pm 0.20}$	–	–
	Cross Attention	$70.41_{\pm 0.16}$	$99.94_{\pm 0.00}$	$69.95_{\pm 0.47}$	$49.93_{\pm 0.20}$	–	–
	Ensembling	$70.63_{\pm 0.52}$	$99.94_{\pm 0.00}$	$70.62_{\pm 0.43}$	$50.10_{\pm 0.42}$	–	–
	Softmax	$67.70_{\pm 0.37}$	$99.83_{\pm 0.01}$	$67.68_{\pm 0.60}$	$49.82_{\pm 0.16}$	–	–
	Rank	$67.46_{\pm 0.47}$	$78.75_{\pm 0.37}$	$67.35_{\pm 0.67}$	$49.81_{\pm 0.55}$	–	–
MUNKEY	Direct Key	$69.02_{\pm 0.24}$	$99.98_{\pm 0.00}$	$99.99_{\pm 0.01}$	$77.30_{\pm 0.39}$	$0.7265_{\pm 0.0066}$	$14.77_{\pm 0.26}$
	Deep Prompt	$67.36_{\pm 0.39}$	$99.77_{\pm 0.05}$	$72.19_{\pm 0.37}$	$52.64_{\pm 0.28}$	$0.5917_{\pm 0.0125}$	$1.95_{\pm 0.25}$
	Cross Attention	$70.48_{\pm 0.16}$	$99.94_{\pm 0.01}$	$72.05_{\pm 0.40}$	$50.78_{\pm 0.14}$	$0.7014_{\pm 0.0079}$	$\underline{0.76}_{\pm 0.18}$
	Ensembling	$70.86_{\pm 0.42}$	$99.94_{\pm 0.00}$	$71.73_{\pm 0.51}$	$51.19_{\pm 0.30}$	$0.2377_{\pm 0.1053}$	$\mathbf{0.61}_{\pm \mathbf{0.27}}$
	Softmax	$67.35_{\pm 0.19}$	$99.80_{\pm 0.01}$	$72.44_{\pm 0.45}$	$52.74_{\pm 0.31}$	$0.2276_{\pm 0.1685}$	$2.02_{\pm 0.23}$
	Rank	$66.79_{\pm 0.13}$	$78.31_{\pm 0.24}$	$71.84_{\pm 0.40}$	$53.30_{\pm 0.12}$	$0.5898_{\pm 0.0126}$	$2.27_{\pm 0.29}$

D.2 Retrieval Strategy Ablations

Effect of the Key Encoder. We evaluate the impact of the frozen key encoder g_ϕ by comparing the standard ViT-Base pre-trained on ImageNet, used throughout the manuscript, against two alternative encoders: DINO-v2[4], self-supervised, and trained on curated datasets like Google Landmarks; and CLIP[5], multimodal and trained on broad internet-crawled image-caption pairs. This ablation

[4]https://huggingface.co/facebook/dinov2-base

[5]https://huggingface.co/openai/clip-vit-base-patch32

demonstrates that MUNKEY does not require label-informed keys to effectively decouple and delete instance-specific information. In Table 18 we observe that across both natural and medical datasets, all three pre-training objectives of the key encoder yield robust and comparable results, consistently outperforming post-hoc baselines in both utility and forgetting.

To further isolate the effect of the key encoder and account for cases where the target dataset may have been part of the encoder's pre-training data, we additionally train a ViT-Tiny g_ϕ on only 20% of the original training set and train MUNKEY on the remaining 80%. This ensures that the encoder has not seen the instances later used for MUNKEY training or unlearning. As shown in Table 19, MUNKEY remains effective in this setting, achieving a low Avg Gap. To contextualize the utility drop in this controlled setting, we additionally train a standard ViT-Tiny from scratch as baseline performance on the same 80% split, after removing the corresponding forget set, i.e., Retrain (T). This baseline achieves $75.49 \pm 0.22\%$ test accuracy on CIFAR-10 and $73.80 \pm 1.10\%$ on DermaMNIST, both below our method's performance. Therefore, MUNKEY retains its main unlearning advantages in terms of membership inference and Avg Gap, even when utility is constrained by the weaker scratch-trained encoder. This suggests that strong pre-trained encoders improve utility, while the core unlearning mechanism does not depend on possible pre-training overlap.

Table 18: Performance under varying key encoders g_ϕ for the memory bank. Results on CIFAR-10 and DermaMNIST.

Dataset	Model Variant	TA	RA	FA	MIA	Avg Gap ↓
CIFAR-10	Retrain (CLIP)	$92.61_{\pm 0.04}$	$98.23_{\pm 0.02}$	$93.19_{\pm 0.15}$	$50.82_{\pm 0.04}$	–
	Retrain (DINO-v2)	$95.32_{\pm 0.02}$	$99.96_{\pm 0.01}$	$95.68_{\pm 0.22}$	$50.71_{\pm 0.34}$	–
	Retrain (ViT/B)	$92.93_{\pm 0.16}$	$99.93_{\pm 0.02}$	$92.76_{\pm 0.45}$	$49.97_{\pm 0.25}$	–
	Key Encoder CLIP	$92.69_{\pm 0.08}$	$98.25_{\pm 0.09}$	$93.84_{\pm 0.16}$	$51.85_{\pm 0.06}$	$\underline{0.45}_{\pm 0.07}$
	Key Encoder DINO-v2	$95.34_{\pm 0.08}$	$99.95_{\pm 0.02}$	$96.04_{\pm 0.34}$	$51.98_{\pm 0.12}$	$\mathbf{0.42}_{\pm 0.14}$
	Key Encoder Pre-trained-ViT/B	$93.12_{\pm 0.08}$	$99.94_{\pm 0.01}$	$93.41_{\pm 0.58}$	$51.45_{\pm 0.39}$	$0.58_{\pm 0.22}$
DermaMNIST	Retrain (CLIP)	$81.08_{\pm 0.02}$	$91.59_{\pm 0.76}$	$79.62_{\pm 1.41}$	$49.58_{\pm 0.89}$	–
	Retrain (DINO-v2)	$81.90_{\pm 0.29}$	$95.66_{\pm 0.56}$	$80.10_{\pm 0.64}$	$50.17_{\pm 0.91}$	–
	Retrain (ViT/B)	$80.80_{\pm 0.54}$	$97.47_{\pm 0.15}$	$80.10_{\pm 0.99}$	$50.53_{\pm 0.89}$	–
	Key Encoder CLIP	$81.43_{\pm 0.20}$	$91.79_{\pm 0.43}$	$84.05_{\pm 1.87}$	$52.09_{\pm 1.28}$	$1.87_{\pm 0.74}$
	Key Encoder DINO-v2	$82.16_{\pm 0.08}$	$96.10_{\pm 0.16}$	$84.52_{\pm 1.12}$	$52.17_{\pm 1.16}$	$\underline{1.78}_{\pm 0.52}$
	Key Encoder Pre-trained-ViT/B	$81.06_{\pm 0.41}$	$97.64_{\pm 0.22}$	$84.10_{\pm 1.58}$	$52.16_{\pm 0.53}$	$\mathbf{1.52}_{\pm 0.56}$

Table 19: Performance on encoder g_ϕ trained from scratch. We compare the default pre-trained encoder against a key encoder trained on only 20% of the original training data, with MUNKEY trained on the remaining 80%. Results are shown on CIFAR-10 and DermaMNIST.

Dataset	Model Variant	TA	RA	FA	MIA	Avg Gap ↓
CIFAR-10	Retrain (20% $\mathcal{D}$ ViT/T)	$76.72_{\pm 1.11}$	$99.40_{\pm 0.05}$	$76.10_{\pm 0.73}$	$50.34_{\pm 0.23}$	–
	Retrain (ViT/B)	$92.14_{\pm 0.09}$	$99.96_{\pm 0.01}$	$92.07_{\pm 0.70}$	$50.18_{\pm 0.22}$	–
	Key Encoder 20% $\mathcal{D}$ ViT/T	$76.42_{\pm 0.73}$	$99.34_{\pm 0.07}$	$79.11_{\pm 0.09}$	$51.88_{\pm 0.27}$	$1.23_{\pm 0.39}$
	Key Encoder Pre-trained-ViT/B	$92.28_{\pm 0.16}$	$99.95_{\pm 0.01}$	$92.64_{\pm 0.72}$	$51.39_{\pm 0.56}$	$0.48_{\pm 0.30}$
DermaMNIST	Retrain (20% $\mathcal{D}$) ViT/T	$74.11_{\pm 0.43}$	$92.96_{\pm 0.23}$	$73.87_{\pm 0.22}$	$50.37_{\pm 0.58}$	–
	Retrain (ViT/B)	$81.05_{\pm 0.25}$	$99.81_{\pm 0.04}$	$79.82_{\pm 0.76}$	$49.68_{\pm 0.80}$	–
	Key Encoder 20% $\mathcal{D}$ ViT/T	$74.41_{\pm 0.12}$	$93.48_{\pm 0.41}$	$77.50_{\pm 1.16}$	$52.41_{\pm 0.66}$	$1.62_{\pm 0.40}$
	Key Encoder Pre-trained-ViT/B	$80.62_{\pm 0.18}$	$99.80_{\pm 0.03}$	$83.57_{\pm 1.52}$	$51.86_{\pm 1.33}$	$1.59_{\pm 0.58}$

Effect of the Neighborhood Size As discussed in Appendix D.1, the inference process involves fusing information from the K retrieved neighbors $\mathcal{N}_K(\mathbf{x}_q)$. In this section, we investigate the impact of varying the neighborhood size, K, on model utility and unlearning efficacy. Figure 4 illustrates the relationship between the number of neighbors and two key metrics across the DermaMNIST and CIFAR-10 datasets: test accuracy, which represents the model's ability to perform standard classification, and Avg Gap, which serves as a proxy for unlearning performance. In both datasets, we observe that test accuracy increases slightly with the number of neighbors before reaching a plateau.

This effect is notably more pronounced on DermaMNIST, likely due to the inherent complexity and fine-grained nature of medical image classification. A similar pattern is seen in Avg Gap, growing slightly before plateauing, all while remaining within small magnitudes. We attribute this to an edge case in model behavior where lower overall accuracy artificially reduces the gap. Consequently, we select $K = 4$ for the results of this work, as it provides a robust balance between high classification performance and effective unlearning.

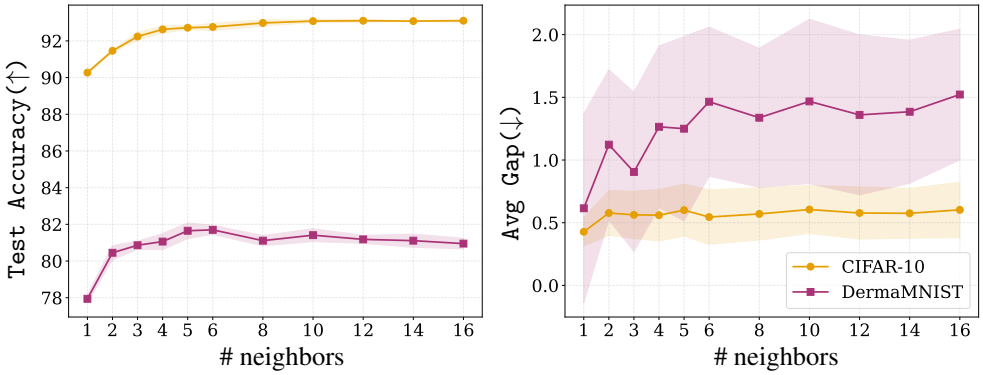


Figure 4: Performance sensitivity to the number of neighbors on DermaMNIST and CIFAR-10. We report Test Accuracy (%) (left) and Avg Gap (right) across different neighborhood sizes.

D.3 Memory Ablations: Varying Exemplar Token Size

We study the effect of memory capacity by varying the dimensionality of the exemplar token, d_e, analyzing how the size of the external memory representation influences unlearning, shown in Figure 5 and Tables 20 and 21.

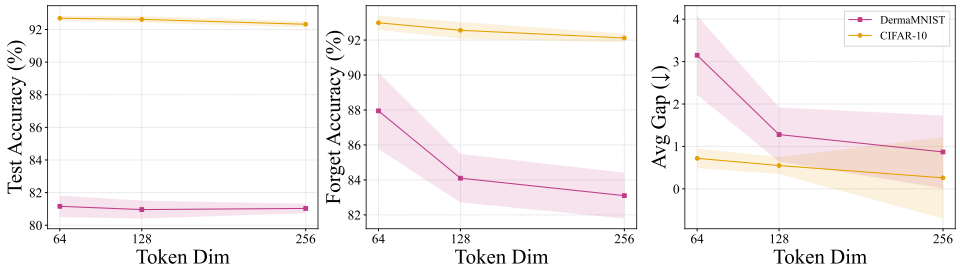


Figure 5: **Performance of MUNKEY under varying exemplar token size.** The impact of the exemplar token dimensionality ($d_e \in \{64, 128, 256\}$) on model utility and unlearning efficacy is illustrated through Test Accuracy, Forget Accuracy, and Avg Gap. Results on DermaMNIST and CIFAR-10 evaluated on three seeds with a forget fraction of 10%.

Table 20: Extended comparison of MUNKEY across exemplar token size with random forget rate 10% on DermaMNIST. We present the accuracy on the test (TA), retain (RA), and forget (FA) sets, as well as membership inference attack (MIA) and average gap values across three seeds.

Model	Token Size	TA	RA	FA	MIA	Avg Gap ↓
Retrain (M)	64	$80.23_{\pm 0.80}$	$99.76_{\pm 0.08}$	$79.95_{\pm 1.81}$	$49.90_{\pm 1.08}$	–
Retrain (M)	128	$81.25_{\pm 0.66}$	$99.70_{\pm 0.08}$	$80.29_{\pm 1.41}$	$50.04_{\pm 1.05}$	–
Retrain (M)	256	$80.98_{\pm 0.55}$	$99.87_{\pm 0.03}$	$80.48_{\pm 1.52}$	$50.08_{\pm 0.97}$	–
MUNKEY	64	$81.16_{\pm 0.66}$	$99.73_{\pm 0.04}$	$87.95_{\pm 2.19}$	$53.52_{\pm 1.91}$	$3.15_{\pm 0.93}$
MUNKEY	128	$81.06_{\pm 0.44}$	$99.71_{\pm 0.08}$	$84.00_{\pm 1.22}$	$51.19_{\pm 1.19}$	$1.27_{\pm 0.64}$
MUNKEY	256	$81.03_{\pm 0.29}$	$99.84_{\pm 0.05}$	$83.10_{\pm 1.31}$	$50.84_{\pm 1.05}$	$0.87_{\pm 0.86}$

Table 21: Extended comparison of MUNKEY across exemplar token size with random forget rate 10% on CIFAR-10. We present the accuracy on the test (TA), retain (RA), and forget (FA) sets, as well as membership inference attack (MIA) and average gap values across three seeds.

Model	Token Size	TA	RA	FA	MIA	Avg Gap ↓
Retrain (M)	64	$92.55_{\pm 0.18}$	$99.82_{\pm 0.05}$	$92.06_{\pm 0.55}$	$49.93_{\pm 0.35}$	–
Retrain (M)	128	$92.38_{\pm 0.06}$	$99.93_{\pm 0.03}$	$92.05_{\pm 0.23}$	$49.90_{\pm 0.34}$	–
Retrain (M)	256	$92.28_{\pm 0.10}$	$99.99_{\pm 0.00}$	$91.86_{\pm 0.21}$	$49.84_{\pm 0.22}$	–
MUNKEY	64	$92.69_{\pm 0.15}$	$99.82_{\pm 0.03}$	$92.99_{\pm 0.40}$	$51.73_{\pm 0.48}$	$0.72_{\pm 0.23}$
MUNKEY	128	$92.63_{\pm 0.20}$	$99.94_{\pm 0.01}$	$92.53_{\pm 0.52}$	$51.40_{\pm 0.44}$	$0.56_{\pm 0.21}$
MUNKEY	256	$92.32_{\pm 0.19}$	$100.00_{\pm 0.00}$	$92.12_{\pm 0.25}$	$50.57_{\pm 0.19}$	$0.26_{\pm 0.12}$

D.4 Architecture Scaling

We examine whether the benefits of MUNKEY persist under larger transformer architectures, assessing how model capacity interacts with utility and unlearning performance.

Scaling Transformer Depth. To isolate the role of representational depth, we vary the number of transformer blocks while keeping the overall framework unchanged, analyzing how the unlearning behavior scales with the architecture backbone parameters in Tables 22 and 23.

Table 22: Comparison of MUNKEY across number of hidden layers with random forget rate 10% on DermaMNIST. We present the accuracy on the test (TA), retain (RA), and forget (FA) sets, membership inference attack (MIA), and average gap values across three seeds.

Model	# Layers	# Parameters	TA	RA	FA	MIA	Avg Gap ↓
Retrain (M)	10	4.6M	$81.15_{\pm 0.86}$	$99.91_{\pm 0.01}$	$80.38_{\pm 1.30}$	$50.49_{\pm 0.74}$	–
Retrain (M)	11	5.1M	$81.33_{\pm 0.36}$	$99.93_{\pm 0.01}$	$80.86_{\pm 1.69}$	$50.55_{\pm 0.89}$	–
Retrain (M)	12 (ViT-T)	5.5M	$81.25_{\pm 0.66}$	$99.70_{\pm 0.08}$	$80.29_{\pm 1.41}$	$50.04_{\pm 1.05}$	–
Retrain (M)	13	6.0M	$81.25_{\pm 0.55}$	$99.92_{\pm 0.02}$	$80.29_{\pm 0.73}$	$50.50_{\pm 0.45}$	–
Retrain (M)	14	6.4M	$81.01_{\pm 0.12}$	$99.89_{\pm 0.03}$	$80.38_{\pm 1.59}$	$50.32_{\pm 1.22}$	–
MUNKEY	10	4.6M	$81.80_{\pm 0.45}$	$99.78_{\pm 0.03}$	$84.86_{\pm 1.53}$	$51.53_{\pm 0.90}$	$1.58_{\pm 0.63}$
MUNKEY	11	5.1M	$81.68_{\pm 0.18}$	$99.76_{\pm 0.06}$	$85.19_{\pm 0.66}$	$51.86_{\pm 0.66}$	$1.54_{\pm 0.54}$
MUNKEY	12 (ViT-T)	5.5M	$81.06_{\pm 0.44}$	$99.71_{\pm 0.08}$	$84.00_{\pm 1.22}$	$51.19_{\pm 1.19}$	$1.27_{\pm 0.64}$
MUNKEY	13	6.0M	$81.88_{\pm 0.19}$	$99.76_{\pm 0.10}$	$84.05_{\pm 1.79}$	$51.48_{\pm 1.30}$	$1.38_{\pm 0.61}$
MUNKEY	14	6.4M	$81.56_{\pm 0.50}$	$99.74_{\pm 0.13}$	$83.57_{\pm 0.88}$	$51.29_{\pm 0.55}$	$1.22_{\pm 0.58}$

Table 23: Comparison of MUNKEY across number of hidden layers with random forget rate 10% on CIFAR-10. We present the accuracy on the test (TA), retain (RA), and forget (FA) sets, membership inference attack (MIA), and average gap values across three seeds.

Model	# Layers	# Parameters	TA	RA	FA	MIA	Avg Gap ↓
Retrain (M)	10	4.6M	$91.95_{\pm 0.08}$	$100.00_{\pm 0.00}$	$91.52_{\pm 0.13}$	$50.08_{\pm 0.31}$	–
Retrain (M)	11	5.1M	$91.94_{\pm 0.06}$	$100.00_{\pm 0.00}$	$91.53_{\pm 0.33}$	$49.89_{\pm 0.22}$	–
Retrain (M)	12 (ViT-T)	5.5M	$92.38_{\pm 0.06}$	$99.93_{\pm 0.03}$	$92.05_{\pm 0.23}$	$49.90_{\pm 0.34}$	–
Retrain (M)	13	6.0M	$91.91_{\pm 0.05}$	$100.00_{\pm 0.00}$	$91.57_{\pm 0.13}$	$50.02_{\pm 0.08}$	–
Retrain (M)	14	6.4M	$91.91_{\pm 0.10}$	$100.00_{\pm 0.00}$	$91.61_{\pm 0.10}$	$50.13_{\pm 0.11}$	–
MUNKEY	10	4.6M	$92.46_{\pm 0.08}$	$99.96_{\pm 0.01}$	$92.51_{\pm 0.34}$	$51.11_{\pm 0.42}$	$0.64_{\pm 0.16}$
MUNKEY	11	5.1M	$92.60_{\pm 0.19}$	$99.95_{\pm 0.01}$	$92.60_{\pm 0.29}$	$51.27_{\pm 0.25}$	$0.79_{\pm 0.15}$
MUNKEY	12 (ViT-T)	5.5M	$92.63_{\pm 0.20}$	$99.94_{\pm 0.01}$	$92.53_{\pm 0.52}$	$51.40_{\pm 0.44}$	$0.56_{\pm 0.21}$
MUNKEY	13	6.0M	$92.54_{\pm 0.08}$	$99.95_{\pm 0.02}$	$92.72_{\pm 0.26}$	$51.38_{\pm 0.30}$	$0.80_{\pm 0.11}$
MUNKEY	14	6.4M	$92.59_{\pm 0.12}$	$99.96_{\pm 0.01}$	$92.67_{\pm 0.25}$	$51.09_{\pm 0.63}$	$0.69_{\pm 0.18}$

Additional Backbone: ViT-Base. Beyond the ViT-Tiny backbone used in the main experiments, we evaluate all methods with ViT-Base and verify that the observed trends transfer to larger-scale models in Table 24.

Table 24: Comparison of MUNKEY and studied baselines using ViT-B/16 backbone (85.8M parameters) on a random forget set of 10% and three seeds on DermaMNIST and CIFAR-10.

	DermaMNIST					CIFAR-10				
Model	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**
Retrain (B)	$76.74_{\pm 0.20}$	$100.0_{\pm 0.00}$	$75.71_{\pm 0.84}$	$49.54_{\pm 0.22}$	–	$75.58_{\pm 0.19}$	$99.64_{\pm 0.02}$	$75.22_{\pm 0.14}$	$50.30_{\pm 0.07}$	–
Retrain (M)	$79.25_{\pm 0.75}$	$99.01_{\pm 0.26}$	$78.05_{\pm 2.12}$	$49.75_{\pm 0.93}$	–	$91.99_{\pm 0.06}$	$99.95_{\pm 0.02}$	$92.16_{\pm 0.50}$	$50.02_{\pm 0.49}$	–
KNN	$74.60_{\pm 0.24}$	$85.54_{\pm 0.14}$	$74.81_{\pm 1.23}$	$49.94_{\pm 0.90}$	–	$91.06_{\pm 0.05}$	$94.56_{\pm 0.06}$	$91.02_{\pm 0.13}$	$50.16_{\pm 0.10}$	–
CF	$75.99_{\pm 0.82}$	$98.11_{\pm 0.73}$	$95.00_{\pm 1.85}$	$64.09_{\pm 1.53}$	$9.12_{\pm 0.70}$	$74.26_{\pm 0.83}$	$95.55_{\pm 0.15}$	$88.49_{\pm 0.58}$	$59.22_{\pm 0.38}$	$6.90_{\pm 0.28}$
CF-k	$77.64_{\pm 0.52}$	$100.0_{\pm 0.00}$	$100.0_{\pm 0.00}$	$64.44_{\pm 0.47}$	$10.02_{\pm 0.28}$	$77.22_{\pm 0.30}$	$99.70_{\pm 0.05}$	$99.63_{\pm 0.13}$	$69.50_{\pm 0.23}$	$11.33_{\pm 0.12}$
EU-k	$77.61_{\pm 0.53}$	$100.0_{\pm 0.00}$	$100.0_{\pm 0.00}$	$62.58_{\pm 0.40}$	$9.55_{\pm 0.28}$	$77.22_{\pm 0.30}$	$99.69_{\pm 0.05}$	$99.63_{\pm 0.11}$	$69.43_{\pm 0.16}$	$11.31_{\pm 0.11}$
ℓ_1-sparse	$73.83_{\pm 1.04}$	$83.35_{\pm 1.32}$	$77.90_{\pm 1.66}$	$53.17_{\pm 1.02}$	$6.35_{\pm 0.68}$	$45.26_{\pm 0.56}$	$45.63_{\pm 0.56}$	$45.85_{\pm 0.78}$	$50.41_{\pm 0.89}$	$28.45_{\pm 0.36}$
NegGrad+	$74.01_{\pm 0.22}$	$96.08_{\pm 1.65}$	$78.71_{\pm 1.72}$	$53.01_{\pm 1.60}$	$3.28_{\pm 0.75}$	$72.53_{\pm 0.24}$	$93.96_{\pm 0.40}$	$75.71_{\pm 0.67}$	$52.79_{\pm 0.18}$	$2.93_{\pm 0.22}$
Amnesiac	$75.56_{\pm 0.23}$	$98.30_{\pm 0.85}$	$82.00_{\pm 2.38}$	$47.01_{\pm 1.11}$	$2.93_{\pm 0.73}$	$73.08_{\pm 0.05}$	$95.11_{\pm 0.15}$	$79.64_{\pm 0.82}$	$48.81_{\pm 0.21}$	$3.24_{\pm 0.22}$
SCRUB	$73.58_{\pm 0.20}$	$80.28_{\pm 1.66}$	$76.52_{\pm 0.54}$	$51.97_{\pm 1.28}$	$6.53_{\pm 0.59}$	$74.48_{\pm 0.00}$	$97.42_{\pm 0.00}$	$94.49_{\pm 0.00}$	$62.69_{\pm 0.00}$	$8.75_{\pm 0.06}$
SSD	$77.42_{\pm 0.41}$	$100.0_{\pm 0.00}$	$100.0_{\pm 0.00}$	$64.37_{\pm 0.05}$	$9.95_{\pm 0.25}$	$77.23_{\pm 0.27}$	$99.70_{\pm 0.06}$	$99.65_{\pm 0.10}$	$69.55_{\pm 0.26}$	$11.34_{\pm 0.11}$
AMUN	$75.08_{\pm 0.72}$	$97.95_{\pm 0.33}$	$92.81_{\pm 1.48}$	$62.08_{\pm 1.33}$	$8.34_{\pm 0.58}$	$73.14_{\pm 0.46}$	$94.63_{\pm 0.37}$	$86.89_{\pm 1.13}$	$58.47_{\pm 0.49}$	$6.82_{\pm 0.34}$
CoUn	$78.39_{\pm 0.76}$	$100.0_{\pm 0.00}$	$82.95_{\pm 1.92}$	$51.81_{\pm 0.96}$	$2.79_{\pm 0.61}$	$76.52_{\pm 0.77}$	$99.74_{\pm 0.01}$	$85.21_{\pm 0.39}$	$51.10_{\pm 0.28}$	$2.96_{\pm 0.24}$
MUNKEY	$80.15_{\pm 0.59}$	$99.47_{\pm 0.10}$	$82.00_{\pm 1.74}$	$52.79_{\pm 0.68}$	$\mathbf{2.09}_{\pm 0.75}$	$92.05_{\pm 0.18}$	$99.92_{\pm 0.04}$	$95.13_{\pm 0.13}$	$55.63_{\pm 0.17}$	$\mathbf{2.17}_{\pm 0.15}$

D.5 Membership Signals on Model Components

Throughout the paper and following the literature, we report membership inference attacks on model outputs under the standard MUNKEY inference setting with retrieved exemplar tokens. In this section, we evaluate whether the isolated image backbone retains a membership signal. To this end, we remove the retrieval branch entirely at evaluation time, so that the model processes only image patches without retrieved exemplar tokens. Removing this pathway eliminates the instance-specific information supplied by the memory. As shown in Table 25, this substantially reduces predictive performance, confirming that the exemplar pathway is actively used by the model. Nevertheless, performance remains above chance, indicating that the image backbone retains meaningful global structure and making this setting a useful stress test for membership signal in the backbone alone. We note that this no-retrieval evaluation also changes the evaluation distribution relative to standard inference, and should therefore be interpreted as a controlled stress test rather than a distribution-preserving deployment setting. We compare MUNKEY against the corresponding Retrain (M) oracle under the same no-retrieval protocol, and include the standard retrieval-based results for reference. The MIA scores remain close to random guessing, indicating that the isolated backbone does not retain a measurable membership signal for forgotten samples.

Table 25: Membership inference on isolated model subcomponents: Standard MUNKEY and removing retrieval from the memory bank (w/o $\mathcal{M}$) across three seeds.

Dataset	**Model**	**TA**	**RA**	**FA**	**MIA**
DermaMNIST	Retrain (M)	$81.25_{\pm 0.66}$	$99.70_{\pm 0.08}$	$80.29_{\pm 1.41}$	$50.04_{\pm 1.05}$
	Retrain (M) w/o $\mathcal{M}$	$33.25_{\pm 6.73}$	$35.41_{\pm 6.96}$	$33.57_{\pm 5.94}$	$50.23_{\pm 1.68}$
	MUNKEY	$81.06_{\pm 0.44}$	$99.71_{\pm 0.08}$	$84.00_{\pm 1.22}$	$51.19_{\pm 1.19}$
	MUNKEY w/o $\mathcal{M}$	$37.68_{\pm 8.40}$	$39.34_{\pm 8.20}$	$38.29_{\pm 8.52}$	$50.81_{\pm 0.51}$
CIFAR-10	Retrain (M)	$92.38_{\pm 0.06}$	$99.93_{\pm 0.03}$	$92.05_{\pm 0.23}$	$49.90_{\pm 0.34}$
	Retrain (M) w/o $\mathcal{M}$	$34.43_{\pm 7.37}$	$35.84_{\pm 7.76}$	$34.74_{\pm 7.74}$	$50.26_{\pm 0.21}$
	MUNKEY	$92.63_{\pm 0.20}$	$99.94_{\pm 0.01}$	$92.53_{\pm 0.52}$	$51.40_{\pm 0.44}$
	MUNKEY w/o $\mathcal{M}$	$39.12_{\pm 9.28}$	$40.89_{\pm 9.52}$	$39.55_{\pm 9.55}$	$51.03_{\pm 0.42}$

D.6 Robustness to Neighbor Deletion

We further investigate whether deleting a key affects the predictions of similar retained samples. While removing a key does not prevent retrieving K neighbors, as the next nearest entries are selected instead, it may affect the predictions of samples that previously relied on the deleted entry. To

quantify this effect, we evaluate CIFAR-10 and DermaMNIST under 10% random forgetting. For each retained sample, we identify forgotten keys within its top-4 neighborhood and compare its accuracy before and after their removal, excluding the sample's own key in both cases to prevent self-retrieval. We group samples by the fraction of their original retrieval weight assigned to the deleted neighbors, as reported in Table 26. MUNKEY's training mechanisms encourage localized information storage and reduce dependence on individual neighboring tokens, potentially limiting interference from key deletion. We see how retained samples with no deleted neighbors remain unaffected, which we include as a reference, while those losing up to half their retrieval weight show accuracy changes of at most 2.83 percentage points. As an additional stress test, deleting each sample's nearest neighbor reduces accuracy by 1.50 and 3.00 percentage points on CIFAR-10 and DermaMNIST, respectively. These results suggest that MUNKEY is robust to neighboring key deletions, limiting unlearning interference with retained samples.

Table 26: Effect of neighbor deletion on retained samples. Accuracy before and after removing forgotten keys from the top-4 neighborhood, grouped by the fraction of retrieval weight removed.

Dataset	Deleted retrieval mass	Acc. before (%)	Acc. after (%)	Δ
DermaMNIST	No deleted neighbor	$83.69_{\pm 0.71}$	$83.69_{\pm 0.71}$	0.00
	0–25%	$83.97_{\pm 1.43}$	$84.58_{\pm 1.41}$	$+0.61$
	25–50%	$85.09_{\pm 1.55}$	$82.26_{\pm 1.66}$	-2.83
CIFAR-10	No deleted neighbor	$93.65_{\pm 0.48}$	$93.65_{\pm 0.48}$	0.00
	0–25%	$92.77_{\pm 0.98}$	$92.62_{\pm 0.98}$	-0.14
	25–50%	$95.10_{\pm 0.94}$	$95.29_{\pm 0.92}$	$+0.19$

D.7 Retrieval-Augmented Study

We further investigate whether an editable retrieval component alone is sufficient for unlearning by comparing different retrieval-based approaches. In addition to the existing baselines, we implement a RAC-inspired retrieval-fusion variant (Long et al., 2022), combining a normally trained ViT classifier with an editable memory of frozen key-label pairs. Classifier logits $\mathbf{s}_{\text{cls}}$ and retrieval probabilities $\mathbf{p}_{\text{ret}}$ are fused as $\mathbf{s}_{\text{cls}} + \alpha \log \mathbf{p}_{\text{ret}}$, where $\alpha > 0$ is learned jointly with the classifier. We use the same frozen key encoder, optimizer, and training schedule as MUNKEY, but without its deletion-oriented training mechanisms. Unlearning consists of removing memory entries without updating the model parameters.

Table 27 compares the RAC-inspired baseline, Direct Key, KNN, and MUNKEY on DermaMNIST under 10% random forgetting. Each trainable method is evaluated against its corresponding retain-only retraining oracle over three seeds. While KNN enables deletion through purely non-parametric retrieval, its predictive utility is lower. Both RAC-inspired retrieval fusion and Direct Key retain substantial membership signals after deletion, whereas MUNKEY remains considerably closer to its retraining oracle. These results highlight the importance of deletion-oriented training for encouraging localized information storage.

Table 27: Comparison of retrieval-augmented approaches on DermaMNIST under 10% random forgetting. Trainable methods are compared against their corresponding retraining oracles across three seeds.

Model	TA	RA	FA	MIA	Avg Gap
RAC-inspired	$76.28_{\pm 0.22}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$61.51_{\pm 0.40}$	$9.29_{\pm 0.33}$
Retrain (RAC)	$76.77_{\pm 0.48}$	$100.00_{\pm 0.00}$	$75.52_{\pm 1.09}$	$49.30_{\pm 1.40}$	–
Direct Key	$81.70_{\pm 0.74}$	$99.30_{\pm 0.45}$	$99.33_{\pm 0.64}$	$61.76_{\pm 2.00}$	$8.23_{\pm 0.70}$
Retrain (Direct Key)	$81.83_{\pm 0.68}$	$99.25_{\pm 0.22}$	$80.05_{\pm 1.10}$	$48.30_{\pm 0.98}$	–
KNN	$74.60_{\pm 0.24}$	$85.54_{\pm 0.14}$	$74.81_{\pm 1.23}$	$49.94_{\pm 0.90}$	–
MUNKEY	$81.06_{\pm 0.41}$	$97.64_{\pm 0.22}$	$84.10_{\pm 1.58}$	$52.16_{\pm 0.53}$	$\mathbf{1.52}_{\pm 0.56}$
Retrain (M)	$80.80_{\pm 0.54}$	$97.47_{\pm 0.15}$	$80.10_{\pm 0.99}$	$50.53_{\pm 0.89}$	–

E Sensitivity Score and Model Selection

As detailed in Section 3, we introduce the Pathway Sensitivity Score (P_s) as a quantitative measure to assess the informational balance between the image and token pathways in MUNKEY. This metric is used in our model selection process, particularly to determine the stochastic dropout probabilities p_i and p_t used in training, which are tuned to account for dataset-specific characteristics. In certain datasets, the model may tend to "shortcut" information by over-relying on either the raw image patches or the exemplar tokens, a behavior that risks pathway collapse and can hinder unlearning and generalization. With this regularization strategy, we aim to push the decoupling of global image features from instance-specific memories. To ensure a robust configuration, we first implement a stability constraint by filtering out models where $P_s \geq \xi$, with $\xi = 0.3$ serving as a global threshold to prevent disproportionate reliance on a single modality. This accounts for the inherent asymmetry between image patches and optimized tokens while ensuring that neither pathway becomes a "dead branch" during training. After this filtering, we apply a selection strategy following the "rewind" procedure in Kurmanji et al. (2023), prioritizing models with the smallest validation utility gap, i.e., the difference between validation accuracy and disjoint proxy forget accuracy. This proxy forget split is sampled only for model selection and is disjoint from the evaluation forget set used in all reported unlearning results. In the event of comparable utility gaps, we select the model with the lower P_s score to favor configurations that maintain a more balanced modality equilibrium. Figure 6 and Figure 7 illustrate these dynamics across various parameter sets in CIFAR-10 and DermaMNIST, with their corresponding utility gaps and P_s in the legend. In CIFAR-10, lower token dropout generally leads to higher pathway divergence, as the model defaults to over-relying on the token pathway. Moreover, the two best parameter configurations have comparable gaps, and we rely on P_s to select the optimal set. Conversely, DermaMNIST displays a wider diversity of dropout patterns, where the best configuration is identified directly through the utility gap. In both benchmarks, the best-performing parameter set is marked with a dashed black line, while the baseline corner case ($p_i = 0, p_t = 0$) is shown in gray. These results highlight that without the introduced $p_i > 0$ and $p_t > 0$, pathway sensitivity is strongly degraded and utility gaps remain sub-optimal, motivating the necessity of our pathway dropout approach.

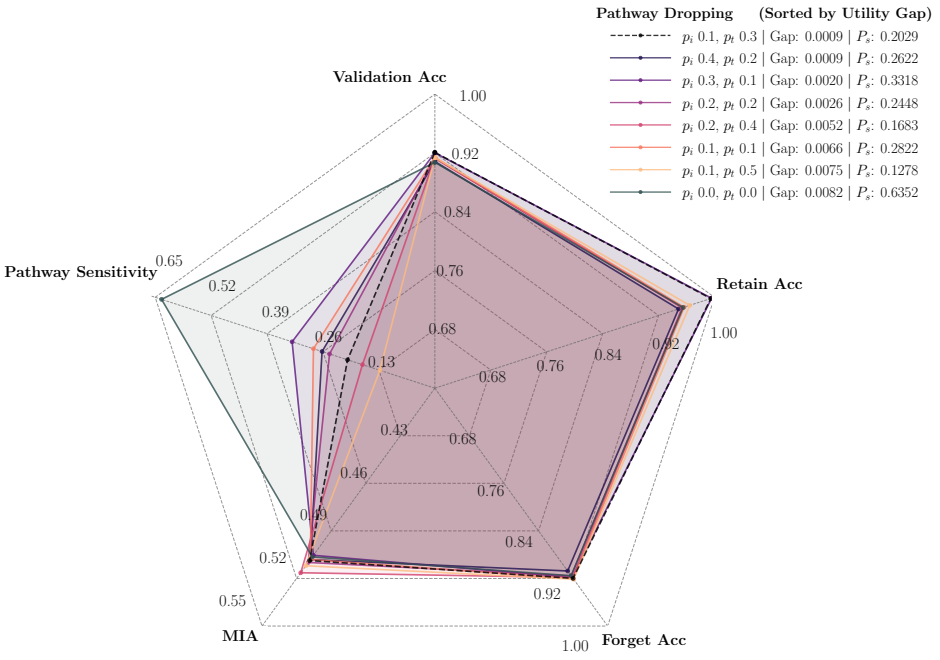


Figure 6: Overview of the performance metrics under varying image and token dropout probabilities in MUNKEY training in CIFAR-10 for visualization of model selection.

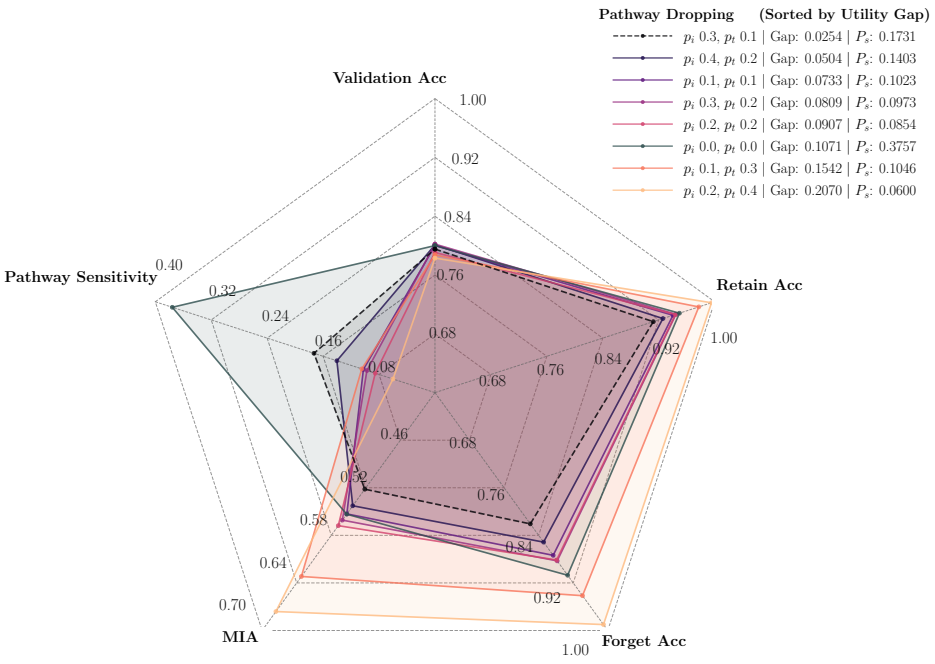


Figure 7: Overview of the resulting performance metrics under varying image and token dropout probabilities in MUNKEY training in DermaMNIST for visualization of model selection.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the proposed unlearning-by-design paradigm, the MUNKEY method, and the empirical findings reported in the experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations throughout the manuscript where they apply.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical theorems or proofs. Instead, it introduces an algorithmic framework supported by mathematical formulations, implementation details, and empirical validation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides implementation details, training procedures, hyperparameters, datasets, baselines, and evaluation protocols in Section 4 and Appendix B sufficient to reproduce the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in the supplemental material?

   Answer: [Yes]

   Justification: We provide a code repository containing the implementation and scripts used for the reported experiments. All datasets are publicly available, and access links and preparation details are provided in Appendix B.2.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify datasets, forget-set construction, model architectures, optimizer settings, hyperparameter selection, baselines, and evaluation metrics in the main text and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All reported experimental results are averaged over three independent random seeds, and tables include standard deviations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report hardware specifications, GPU type, memory, and runtime comparisons for both training and unlearning procedures in the experimental section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, the research complies with the NeurIPS Code of Ethics. The work uses publicly available datasets, and focuses on privacy-enhancing machine unlearning methods.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We motivate the method as a response to the current privacy regulations and the needs for society/users to withdraw consent to use their data. In this setting, machine unlearning comes as a positive societal impact to allow for this data removal. This is clarified in the introduction and conclusion. To the best of our knowledge, there are no further societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This work studies discriminative vision models on standard benchmarks and does not release high-risk generative systems or sensitive new datasets requiring additional safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: External assets used in the paper, including datasets and pretrained models, are cited appropriately. Dataset sources, access information, and licensing details are summarized in Appendix B.2. The proposed codebase was developed by the authors.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the code used for the experiments and proposed method in a repository reported in the main text, which includes its corresponding documentation in a README.md.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not include crowdsourcing experiments or research involving human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work did not require new IRB approval because it uses only publicly available, pre-existing benchmark datasets and does not involve new data collection, intervention, interaction with human participants, or access to identifiable private information. The medical datasets are used through MedMNIST v2, which provides standardized public biomedical image benchmarks for research and educational purposes and is not intended for clinical use. We follow the corresponding dataset licenses and cite the original dataset sources.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing and language editing support. They were not part of the proposed method, experiments, or scientific contributions of the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.