RETHINKING MACHINE UNLEARNING: MODELS DESIGNED TO FORGET 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 is rapidly becoming a practical requirement, driven by privacy regulations, data errors, and the need to remove harmful or corrupted training samples. Despite this, most existing methods tackle the problem purely from a post-hoc perspective. They attempt to erase the influence of targeted training 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 propose *unlearning by design*, a novel paradigm in which models are directly trained to support forgetting as an inherent 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 direct zero-shot forgetting without weight updates or access to the original samples or labels. Across natural image benchmarks, fine-grained 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 has become a legal and ethical imperative, driven by regulations such as the GDPR (Mantelero, 2013). Because neural networks can memorize specific data (Arpit et al., 2017), the central challenge is to surgically excise the influence of targeted points without degrading the model's performance on retained data. Current approaches are generally split between exact unlearning, including retraining, which is computationally prohibitive, and approximate methods (Nguyen et al., 2025) that rely on iterative weight updates. These existing methods act post-hoc, after the base model has been trained. They are hampered by *knowledge entanglement*, where instance-specific details are fused into a monolithic set of static weights (Arpit et al., 2017), making *a posteriori* deletion inefficient. To address this, we propose a paradigm shift toward *unlearning by design*. We introduce Machine UNlearning via KEY deletion (MUNKEY), a method that decouples instance-specific information from static parameters by associating training instances with learnable exemplar tokens. By externalizing this data, MUNKEY transforms unlearning from a complex optimization problem into a simple, near-instantaneous set-theoretic operation: deleting entries from an external memory. This approach offers "zero-shot" efficiency with no parameter updates and enhances privacy by requiring only instance identifiers at the time of deletion. While external memories have been used for predictive gain (Jia et al., 2022; Long et al., 2022), we leverage this architecture specifically for the "inverse" task of forgetting. Our approach is further motivated by evidence that such visual memories can be effectively leveraged at scale, even with billion-scale data (Geirhos et al., 2025).

Our contributions are threefold: (i) *Paradigm Shift:* We propose shifting from post-hoc corrections to *unlearning by design*, anticipating deletions during the model lifecycle. (ii) *Framework:* We introduce MUNKEY, which externalizes data into an exemplar memory to enable zero-shot unlearning via

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

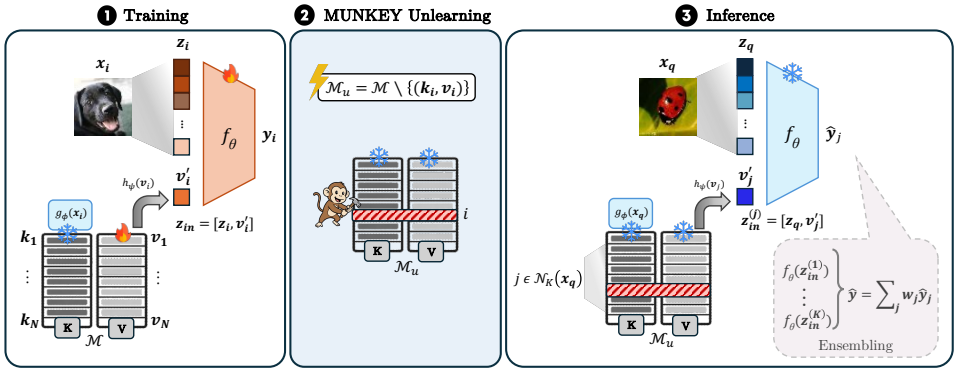

Figure 1: **Overview of MUNKEY. (1) Training:** The learnable exemplar token $\boldsymbol{v}_i$ from the memory bank $\mathcal{M}$ is concatenated to image tokens $\boldsymbol{z}_i$ to predict the label $\boldsymbol{y}_i$. **(2) Unlearning** results in an instance-specific key deletion in $\mathcal{M}$. **(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 to produce the output.

key deletion, bypassing the need for raw data access during unlearning. (iii) *Validation:* Extensive evaluations across natural and medical imagery demonstrate that MUNKEY outperforms state-of-the-art baselines, offering a superior balance of computational efficiency, model utility and forgetting.

2 Methods

2.1 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 $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ with a task-specific loss. 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 on $\mathcal{D}_r$. The objective of approximate machine unlearning is to achieve a behavior such that $f_{\theta_u} \approx f_r$ without the prohibitive cost of full retraining. 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 into a sequence of n flattened patches and projects them onto a latent space $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ through an embedding layer. In this work, we will refer to f_θ as the model after the embedding layer.

2.2 MUNKEY: Machine Unlearning via Key Deletion

We propose MUNKEY[1] to reformulate unlearning as a modular data-deletion task within an external memory bank $\mathcal{M}$. By conditioning classification on a joint representation of raw image patches and retrieved exemplar tokens, we ensure that a specific sample's influence is confined to its corresponding entry in $\mathcal{M}$, as illustrated in Figure 1.

External Exemplar Memory. MUNKEY shifts the burden of memorization from static weights θ to an external memory bank $\mathcal{M} = \{(\boldsymbol{k}_i, \boldsymbol{v}_i)\}_{i=1}^N$ defined for all $\boldsymbol{x}_i \in \mathcal{D}$. Here, $\boldsymbol{k}_i = g_\phi(\boldsymbol{x}_i)$ is a fixed key from a frozen encoder, and $\boldsymbol{v}_i$ is a learnable exemplar token projected into the ViT's latent space via an adapter h_ψ. This enables "zero-shot" unlearning via simple deletion of the memory entry.

Training with Stochastic Pathway Dropout. To train MUNKEY, 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 $\boldsymbol{z}_{in}$. A fundamental challenge in such dual-stream processing is *modality dominance* (Wang et al., 2020; Hussen Abdelaziz et al., 2020), where high-capacity exemplar tokens might cause the model to ignore robust features from raw patches, or vice versa. To enforce a balanced information flow, we apply a stochastic pathway dropout to both pathways. Let $\mathbf{m} = [\gamma_{img}, \gamma_{tok}] \in \{0, 1\}^2$ be a masking vector following a categorical distribution where either pathway or both can be active, but never none. We define the dropout operator $\mathcal{P}$ as $\mathcal{P}(\boldsymbol{u}, \emptyset, \gamma) = \gamma \boldsymbol{u} + (1 - \gamma)\emptyset$, where $\emptyset$ represents a globally learned null token. This architecture ensures that the transformer f_θ learns to predict the class even

[1] https://anonymous.4open.science/r/MUL_token-DF88/README.md

when one modality is missing, preventing the model from over-relying on a single stream. To prevent *pathway collapse*, where the model ignores the memory or fails to generalize, we introduce the Pathway Sensitivity Score (P_s) and leverage it for hyperparameter selection to ensure the model maintains a high utility gap between the two streams, as detailed in Appendix G.

Inference and Retrieval Regularization. At inference, the specific token $\boldsymbol{v}_q$ for a query image $\boldsymbol{x}_q$ is unavailable. Instead, we retrieve the K nearest neighbors $\mathcal{N}_K(\boldsymbol{x}_q)$ from $\mathcal{M}$ based on the cosine similarity of the query key $\boldsymbol{k}_q = g_\phi(\boldsymbol{x}_q)$. The final prediction is a similarity-weighted ensemble: $\hat{\boldsymbol{y}} = \sum_{j \in \mathcal{N}_K} w_j \cdot f_\theta([\boldsymbol{z}_q, \boldsymbol{v}'_j])$, where $w_j = \text{softmax}(\text{sim}(\boldsymbol{k}_q, \boldsymbol{k}_j)/\tau)$.

To ensure the backbone f_θ generalizes to this retrieved context, we simulate this state during training. For a sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, an indicator $r \sim \text{Bernoulli}(p_r)$ selects between the instance token $\boldsymbol{v}'_i$ and an aggregated neighbor token $\boldsymbol{v}'_n$ derived from $\mathcal{N}_K(\boldsymbol{x}_i)$. The augmented input sequence is defined as:

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

During optimization, the gradient $\nabla_{\boldsymbol{v}_j} \ell$ is clipped to zero for all $j \in \mathcal{N}_K$ when the retrieval branch is active ($r = 1$). This prevents neighbor tokens from drifting to accommodate the gradients of a different sample, preserving the integrity of the memory bank.

Unlearning. Unlike traditional methods that rely on iterative optimization to erase information, unlearning in MUNKEY is reduced to a simple set-theoretic operation: $\mathcal{M}_u = \mathcal{M} \setminus \{(\boldsymbol{k}_i, \boldsymbol{v}_i) \mid i \in \mathcal{D}_f\}$. 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 backbone parameters θ and adapter ψ are 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 in Appendix C.

3 Results

To validate the proposed *unlearning by design* paradigm, we address three research questions: *(i) Decoupled Memorization:* Can instance-specific tokens enable near-instantaneous, zero-shot unlearning? *(ii) Information Encoding:* Does the memory capture granular signals beyond class labels in the representation space? *(iii) Architectural Interaction:* How do pathway aggregation strategies impact performance? The experimental setup and details are detailed in Section B.

Decoupled Memorization: Forgetting and Utility Results. Table 1 shows that MUNKEY consistently achieves the lowest Avg Gap across 10% and 2% forget rates. On CIFAR-10, where visual features align with our frozen key encoder, MUNKEY significantly outperforms post-hoc baselines in predictive accuracy while maintaining robust unlearning (MIA scores). This demonstrates that leveraging the data manifold via decoupled forgetting still ensures both performance and privacy. Notably, the key encoder provides no unfair advantage, as all methods are compared against their respective Retrain gold standards. Even on DermaMNIST, where medical images differ from key-encoder pre-training data, MUNKEY still leads in both Avg Gap and accuracy. While the KNN baseline shows competitive MIA scores, its lower predictive power suggests MUNKEY better captures generalization. Further tests in Appendix F.2 confirm robustness using alternative encoders like DINO-v2 or CLIP. The results on four other datasets are provided in Appendix E. A core advantage of the *unlearning by design* paradigm is the elimination of post-hoc optimization. As shown in Table 2 in Appendix D.1, while gradient-based or Fisher-dependent baselines require significant compute, MUNKEY's unlearning time is negligible (≈ 0s). Forgetting is achieved simply by dropping key-value pairs from the memory bank. Though MUNKEY introduces a minimal training overhead compared to standard ViTs, this cost is incurred only once. This small initial investment provides the permanent benefit of instant, zero-shot unlearning throughout the model's entire deployment lifecycle.

Information Encoded in the Memory. We used 2D UMAP (McInnes et al., 2018) projections to determine if exemplar tokens capture instance-specific data beyond class labels. As shown in Figure 2, while $[CLS]$ tokens form tight, class-specific clusters for classification, the Exemplar Memory Bank $\{\boldsymbol{v}_i\}_{i=1}^N$ maintains a more granular, diffuse topology. On CIFAR-10, the memory bank shows moderate clustering due to high feature relevance, whereas on DermaMNIST, representations are more scattered due to distribution shifts. Crucially, temporal visualizations reveal that while $[CLS]$

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 present 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)	$76.46_{\pm0.22}$	$100.0_{\pm0.00}$	$74.86_{\pm1.77}$	$48.88_{\pm0.69}$	–	$76.49_{\pm0.26}$	$100.0_{\pm0.00}$	$75.71_{\pm2.02}$	$49.98_{\pm1.60}$	–
Retrain (M)	$81.25_{\pm0.66}$	$99.70_{\pm0.08}$	$80.29_{\pm1.41}$	$50.04_{\pm1.05}$	–	$81.80_{\pm0.04}$	$99.75_{\pm0.05}$	$81.43_{\pm2.10}$	$48.79_{\pm1.07}$	–
KNN	$74.60_{\pm0.24}$	$85.54_{\pm0.14}$	$74.81_{\pm1.23}$	$49.94_{\pm0.90}$	–	$75.01_{\pm0.07}$	$85.84_{\pm0.09}$	$74.52_{\pm0.67}$	$50.32_{\pm3.30}$	–
CF	$75.28_{\pm0.98}$	$97.50_{\pm1.55}$	$94.48_{\pm2.23}$	$64.41_{\pm1.96}$	$9.71_{\pm0.99}$	$76.38_{\pm0.28}$	$97.50_{\pm1.16}$	$95.48_{\pm1.21}$	$62.20_{\pm1.80}$	$8.65_{\pm0.90}$
CF-k	$76.94_{\pm0.42}$	$99.86_{\pm0.13}$	$99.90_{\pm0.13}$	$67.10_{\pm3.73}$	$10.97_{\pm1.05}$	$76.96_{\pm0.43}$	$99.86_{\pm0.13}$	$100.0_{\pm0.00}$	$65.73_{\pm1.38}$	$10.16_{\pm0.74}$
EU-k	$76.89_{\pm0.58}$	$99.60_{\pm0.23}$	$99.52_{\pm0.18}$	$69.27_{\pm0.23}$	$11.47_{\pm0.51}$	$76.97_{\pm0.47}$	$99.63_{\pm0.22}$	$99.76_{\pm0.34}$	$68.54_{\pm1.16}$	$10.87_{\pm0.73}$
ℓ_1-sparse	$75.81_{\pm0.45}$	$92.36_{\pm1.80}$	$86.81_{\pm1.17}$	$57.69_{\pm1.13}$	$7.26_{\pm0.78}$	$75.61_{\pm0.67}$	$89.46_{\pm3.01}$	$80.24_{\pm1.21}$	$54.62_{\pm0.48}$	$5.15_{\pm1.06}$
NegGrad+	$76.26_{\pm0.37}$	$96.31_{\pm1.03}$	$83.48_{\pm1.25}$	$55.83_{\pm1.14}$	$\underline{4.87}_{\pm0.69}$	$75.73_{\pm0.96}$	$98.08_{\pm0.42}$	$74.05_{\pm2.99}$	$47.63_{\pm1.79}$	$\underline{1.67}_{\pm1.12}$
SCRUB	$73.97_{\pm1.81}$	$76.02_{\pm2.48}$	$74.68_{\pm1.23}$	$49.62_{\pm0.25}$	$6.85_{\pm0.96}$	$76.38_{\pm0.76}$	$99.55_{\pm0.53}$	$86.94_{\pm1.70}$	$57.41_{\pm1.00}$	$4.81_{\pm0.85}$
SSD	$75.07_{\pm2.08}$	$92.07_{\pm10.2}$	$89.73_{\pm12.9}$	$63.29_{\pm8.28}$	$9.65_{\pm4.67}$	$72.58_{\pm5.56}$	$87.87_{\pm16.3}$	$78.52_{\pm30.3}$	$60.98_{\pm9.33}$	$7.46_{\pm9.07}$
AMUN	$76.46_{\pm0.69}$	$98.33_{\pm0.94}$	$95.05_{\pm0.64}$	$63.70_{\pm0.27}$	$9.17_{\pm0.59}$	$76.46_{\pm0.69}$	$97.99_{\pm0.91}$	$98.57_{\pm1.01}$	$64.95_{\pm1.01}$	$9.97_{\pm0.79}$
MUNKEY	$81.06_{\pm0.44}$	$99.71_{\pm0.08}$	$84.00_{\pm1.22}$	$51.19_{\pm1.19}$	$\mathbf{1.27}_{\pm0.64}$	$81.23_{\pm0.53}$	$99.71_{\pm0.10}$	$85.48_{\pm2.76}$	$50.34_{\pm0.85}$	$\mathbf{1.55}_{\pm0.94}$
CIFAR-10										
Retrain (T)	$78.13_{\pm0.64}$	$91.62_{\pm0.55}$	$77.57_{\pm0.83}$	$49.77_{\pm0.32}$	–	$78.87_{\pm1.02}$	$91.83_{\pm0.61}$	$79.60_{\pm0.86}$	$50.90_{\pm0.78}$	–
Retrain (M)	$92.38_{\pm0.06}$	$99.93_{\pm0.03}$	$92.05_{\pm0.23}$	$49.90_{\pm0.34}$	–	$92.59_{\pm0.20}$	$99.93_{\pm0.02}$	$92.83_{\pm0.54}$	$50.79_{\pm1.73}$	–
KNN	$91.06_{\pm0.05}$	$94.56_{\pm0.06}$	$91.02_{\pm0.13}$	$50.16_{\pm0.10}$	–	$91.28_{\pm0.08}$	$94.58_{\pm0.02}$	$92.10_{\pm0.28}$	$50.35_{\pm0.39}$	–
CF	$77.66_{\pm0.50}$	$91.74_{\pm0.48}$	$87.25_{\pm1.04}$	$55.86_{\pm0.16}$	$4.09_{\pm0.44}$	$78.53_{\pm0.95}$	$91.97_{\pm0.91}$	$87.10_{\pm0.59}$	$55.38_{\pm1.35}$	$3.12_{\pm0.65}$
CF-k	$79.60_{\pm0.63}$	$94.30_{\pm0.55}$	$94.16_{\pm0.61}$	$58.93_{\pm0.40}$	$7.48_{\pm0.41}$	$79.53_{\pm0.60}$	$94.30_{\pm0.55}$	$94.53_{\pm0.29}$	$59.41_{\pm0.71}$	$6.64_{\pm0.50}$
EU-k	$79.44_{\pm0.72}$	$94.22_{\pm0.55}$	$94.00_{\pm0.65}$	$58.87_{\pm0.30}$	$7.36_{\pm0.42}$	$79.36_{\pm0.64}$	$94.20_{\pm0.58}$	$94.63_{\pm0.12}$	$59.16_{\pm0.69}$	$6.54_{\pm0.50}$
ℓ_1-sparse	$52.80_{\pm0.55}$	$53.47_{\pm0.75}$	$52.97_{\pm0.86}$	$50.79_{\pm0.79}$	$22.28_{\pm0.48}$	$52.70_{\pm0.73}$	$53.79_{\pm0.89}$	$53.70_{\pm2.00}$	$50.50_{\pm0.77}$	$22.63_{\pm0.74}$
NegGrad+	$77.29_{\pm0.27}$	$91.08_{\pm0.42}$	$83.13_{\pm0.89}$	$53.65_{\pm0.32}$	$\underline{2.71}_{\pm0.41}$	$77.03_{\pm0.48}$	$90.72_{\pm0.74}$	$72.03_{\pm1.06}$	$47.62_{\pm0.75}$	$3.45_{\pm0.57}$
Amnesiac	$78.31_{\pm0.35}$	$92.17_{\pm0.75}$	$85.67_{\pm1.05}$	$52.51_{\pm0.21}$	$2.89_{\pm0.46}$	$77.77_{\pm0.29}$	$91.11_{\pm0.68}$	$76.97_{\pm1.21}$	$45.95_{\pm0.86}$	$\underline{2.35}_{\pm0.59}$
SCRUB	$76.51_{\pm0.78}$	$85.31_{\pm0.71}$	$83.72_{\pm0.94}$	$54.18_{\pm0.47}$	$4.62_{\pm0.48}$	$77.74_{\pm1.19}$	$89.83_{\pm1.11}$	$85.61_{\pm0.84}$	$55.03_{\pm1.31}$	$3.32_{\pm0.70}$
SSD	$78.54_{\pm1.08}$	$93.29_{\pm0.71}$	$92.84_{\pm0.96}$	$58.91_{\pm0.43}$	$6.62_{\pm0.52}$	$79.02_{\pm1.06}$	$93.63_{\pm0.62}$	$94.00_{\pm0.22}$	$58.75_{\pm0.78}$	$6.05_{\pm0.55}$
AMUN	$77.71_{\pm0.46}$	$91.25_{\pm0.56}$	$86.85_{\pm3.46}$	$55.90_{\pm1.94}$	$4.05_{\pm1.05}$	$77.71_{\pm0.46}$	$90.89_{\pm0.94}$	$87.27_{\pm3.74}$	$56.17_{\pm2.08}$	$3.76_{\pm1.18}$
MUNKEY	$92.63_{\pm0.20}$	$99.94_{\pm0.01}$	$92.53_{\pm0.52}$	$51.40_{\pm0.44}$	$\mathbf{0.56}_{\pm0.21}$	$92.69_{\pm0.21}$	$99.94_{\pm0.01}$	$93.57_{\pm0.26}$	$51.83_{\pm1.62}$	$\mathbf{0.47}_{\pm0.62}$

tokens quickly polarize by class, exemplar tokens evolve into a richer topology that mirrors specific instance features rather than simple class separation. Moreover, we include detailed qualitative analysis of the k-NN retrieval mechanics and a comprehensive ablation of architectural aggregation strategies in Appendix D.2 and F.1, respectively.

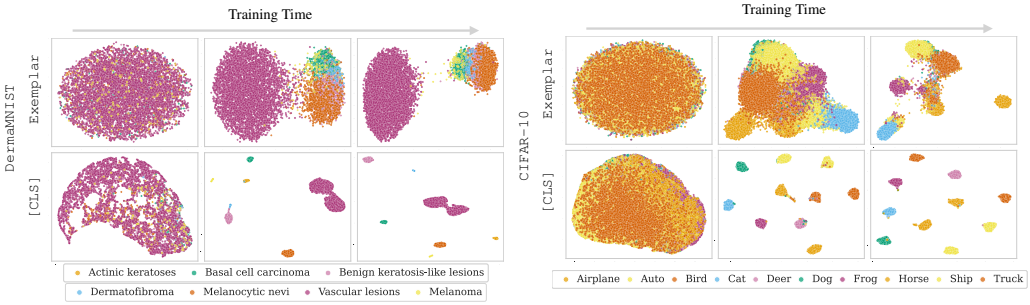

Figure 2: UMAP visualization of exemplar and $[CLS]$ tokens shown at epochs 0, 50, and 100.

4 Conclusion

In this work, we advocate for a shift toward the *unlearning by design* paradigm. Current research focuses largely on post-hoc methods to adapt models that were never intended to forget. We argue that, as data privacy regulations and security concerns intensify, there is a critical need for research into architectures that are inherently modular by construction. Within this paradigm, we introduce MUNKEY, a framework that externalizes instance-specific knowledge into a decoupled memory bank. By shifting the burden of memorization from static weights to a learnable memory, near-instantaneous, zero-shot forgetting is achieved. The results across diverse domains, including natural imagery and medical datasets, highlight the robustness of this approach, outperforming established baselines. Our analysis also provides insights into the pathway interaction between image

features and exemplar tokens, and is complemented by visualizations for better understanding of the learned memory. Ultimately, we believe this work motivates exploration into models that treat the ability to forget not as an afterthought, but as a fundamental architectural requirement, through the use of decoupled memories or other instantiations of the *unlearning by design* paradigm.

ACKNOWLEDGMENTS

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.

REFERENCES

Andrea Acevedo, Anna Merino, Santiago Alférez, Ángel Molina, Laura Boldú, and José Rodellar. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in brief*, 30:105474, 2020. 11

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, 2017. URL http://proceedings.mlr.press/v70/arpit17a.html. 1

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pp. 141–159. IEEE, 2021. 1, 10

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015. 1, 10

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam 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, 2024. doi: 10.1609/AAAI.V38I10.28996. URL https://doi.org/10.1609/aaai.v38i10.28996. 10

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan S. Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In Brian Williams, Yiling Chen, and Jennifer 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, 2023. doi: 10.1609/AAAI.V37I6.25879. URL https://doi.org/10.1609/aaai.v37i6.25879. 10

Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *ArXiv preprint*, abs/1902.03368, 2019. URL https://arxiv.org/abs/1902.03368. 11

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 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, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848. 11

Zonglin Di, Sixie Yu, Yevgeniy Vorobeychik, and Yang Liu. Adversarial machine unlearning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=swWF948IiC. 12

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 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, 2021. URL https://openreview.net/forum?id=YicbFdNTTy. 2

Ali Ebrahimpour-Boroojeny, Hari Sundaram, and Varun Chandrasekaran. Not all wrong is bad: Using adversarial examples for unlearning. In *Forty-second International Conference on Machine Learning*, 2025. 10

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. 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, 2024. URL https://openreview.net/forum?id=gn0mIhQGNM. 10, 12

Evelyn Fix. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine, 1985. 12

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam 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, 2024. doi: 10.1609/AAAI.V38I11.29092. URL https://doi.org/10.1609/aaai.v38i11.29092. 10, 12

Robert Geirhos, Priyank Jaini, Austin Stone, Sourabh Medapati, Xi Yi, George Toderici, Abhijit Ogale, and Jonathon Shlens. Towards flexible perception with visual memory. In *Forty-second International Conference on Machine Learning*, 2025. 1

Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. *ArXiv preprint*, abs/2201.06640, 2022. URL https://arxiv.org/abs/2201.06640. 10, 12

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 10

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. 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, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17371. 10

Ahmed Hussen Abdelaziz, Barry-John Theobald, Paul Dixon, Reinhard Knothe, Nicholas Apostoloff, and Sachin Kajareker. Modality dropout for improved performance-driven talking faces. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, pp. 378–386, 2020. 2

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey 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*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a204aa68ab4e970e1ceccfb5b5cdc5e4-Abstract-Conference.html. 10, 12

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022. 1, 10, 18

Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue. *(No Title)*, 2018. 11

Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nektarios A Valous, Dyke Ferber, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, 16(1):e1002730, 2019. 11

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 11

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey 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*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/062d711fb777322e2152435459e6e9d9-Abstract-Conference.html. 10, 12, 22

Shen Lin, Xiaoyu Zhang, Chenyang Chen, Xiaofeng Chen, and Willy Susilo. 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, 2023. doi: 10.1109/CVPR52729.2023.01929. URL https://doi.org/10.1109/CVPR52729.2023.01929. 10

Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton Van den Hengel. Retrieval augmented classification for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6959–6969, 2022. 1

Alessandro Mantelero. 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, 2013. 1

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989. 10

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv preprint*, abs/1802.03426, 2018. URL https://arxiv.org/abs/1802.03426. 3

Jonas Mirlach, Sonia Laguna, and Julia E. Vogt. Reference-guided machine unlearning. *International Conference on Learning Representations (ICLR) 2026 Workshop on Test-Time Updates*, 2026. 10

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025. 1, 10

Gaurav Patel and Qiang Qiu. 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, 2025. 12

Mark Sandler, Andrey Zhmoginov, Max Vladymyrov, and Andrew Jackson. 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, 2022. doi: 10.1109/CVPR52688.2022.01184. URL https://doi.org/10.1109/CVPR52688.2022.01184. 10

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman 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, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html. 10

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018. 11

Weiyao Wang, Du Tran, and Matt Feiszli. 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, 2020. doi: 10.1109/CVPR42600.2020.01271. URL https://doi.org/10.1109/CVPR42600.2020.01271. 2

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1410.3916. 10

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=TrjbxzRcnf-. 10

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023. 11

Yu Zhou, Dian Zheng, Qijie Mo, Renjie Lu, Kun-Yu Lin, and Wei-Shi Zheng. 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, 2025. 10

A Related Work

Machine Unlearning. The goal of machine unlearning is to remove 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, recent works have proposed selectively adjusting specific layers or parameters: ℓ_1-sparse unlearning (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 (Graves et al., 2021), adversarial examples (Ebrahimpour-Boroojeny et al., 2025; Cha et al., 2024), and student-teacher methods such as Bad-Teacher (Chundawat et al., 2023) and its later refinement SCRUB (Kurmanji et al., 2023), as well as recent class-level distillation methods (Zhou et al., 2025).

Despite their differences, these methods share a common limitation: they operate in a post-hoc setting, requiring access to the forget set, the retain set, the labels, or a combination thereof to modify an already trained model. We instead advocate for a paradigm shift where unlearning is anticipated and explicitly enforced during training. Unlike prior work that relies on inducing sparsity as a precursor to facilitate unlearning (Lin et al., 2023), we propose a memory-augmented method designed specifically for instance-level removal. By externalizing instance-specific information into a dedicated 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 (Graves et al., 2014), Memory Networks (Weston et al., 2015; Sukhbaatar et al., 2015), and Memorizing Transformers (Wu et al., 2022) traditionally rely on decoupling memory storage from the model's primary parameters. In parallel, a complementary line of research focuses on associative memory, where learnable structures are embedded directly into network components to store patterns or knowledge (Sandler et al., 2022; Jia et al., 2022). However, while both traditionally utilize memory to enhance predictive performance or facilitate few-shot learning, we repurpose the external memory framework for a distinct objective: efficient data removal. By leveraging the inherent decoupling of external storage, we enable instant unlearning that avoids the complexities of modifying integrated weights. To the best of our knowledge, this work represents the first application of external memory mechanisms specifically designed for machine unlearning.

B Detailed Experimental Setting

B.1 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.

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.

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 balanced classes. 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 E, F.1.

B.2 Baselines and Implementation

To provide a comprehensive evaluation of MUNKEY, we compare against a suite of nine baselines spanning traditional optimization, weight sparsification, distillation-based unlearning, parameter dumping, 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.

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 $\mathcal{D}_r$, referred to as Retrain (T). For our method, we additionally report an oracle that matches the MUNKEY ViT-Tiny architecture with

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

its hyperparameters and additional components, also trained uniquely on $\mathcal{D}_r$, providing an upper bound aligned with our design and referred to as Retrain (M).

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 finetuning 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 (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 finetuning with the rest of the data.

Following recent literature Di 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) (Di 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.

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.

B.3 Metrics

Following prior work (Patel & Qiu, 2025; Fan et al., 2024; Di et al., 2025), we evaluate our model using four primary metrics and a final 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 evaluate forgetting. Our 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. An MIA AUROC near 0.5 on $\mathcal{D}_f$ indicates successful unlearning. 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)*.

B.4 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. All experiments are conducted over three independent seeds. The training pipeline and dynamics are standardized across all base models and datasets. 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. 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, images are augmented using random resize cropping 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 F.3. Regarding the aggregation of neighbors in the experiments, when applicable, we set the temperature parameter τ to 0.07 in the softmax operations. Moreover, the key encoder used for retrieval is a ViT-B/16 pretrained on ImageNet (sourced from HuggingFace[3]), and again we provide further ablations regarding the choice of key encoder in Appendix F.2. The dropout probabilities for each dataset, selected based on Appendix G, 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.

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

C MUNKEY: ALGORITHM

In Algorithm 1 we describe all the steps for training, unlearning, and doing inference on MUNKEY, as described in Section 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
while not converged **do**
  for each $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}$ **do**
    $\boldsymbol{z}_i \leftarrow \text{ViTEmbed}(\boldsymbol{x}_i)$, $\boldsymbol{k}_i \leftarrow g_\phi(\boldsymbol{x}_i)$
    Sample $\mathbf{m} = [\gamma_{img}, \gamma_{tok}] \sim \text{Cat}(p_i, p_t, 1 - p_i - p_t, 0)$
    Sample $r \sim \text{Bern}(p_r)$
    if $r = 1$ **then**
      Sample $K' \sim \mathcal{U}\{2, 16\}$
      $\boldsymbol{v}_n \leftarrow \text{Agg}(\{\boldsymbol{v}_j \mid j \in \mathcal{N}_{K'}(\boldsymbol{x}_i)\})$
      Exemplar $\leftarrow h_\psi(\boldsymbol{v}_n)$
    else
      Exemplar $\leftarrow h_\psi(\boldsymbol{v}_i)$
    end if
    $\boldsymbol{z}_{in} \leftarrow [\mathcal{P}(\boldsymbol{z}_i, \emptyset_{img}, \gamma_{img}), \mathcal{P}(\text{Exemplar}, \emptyset_{tok}, \gamma_{tok})]$
    $\mathcal{L} \leftarrow \ell(f_\theta(\boldsymbol{z}_{in}), \boldsymbol{y}_i)$
    Update $\theta, \psi, \boldsymbol{v}_i$ by minimizing $\mathcal{L}$
  end for
end while
Phase 2: Zero-Shot Unlearning
Input: Forget set $\mathcal{D}_f$
$\mathcal{M}_u \leftarrow \mathcal{M} \setminus \{(\boldsymbol{k}_i, \boldsymbol{v}_i) \mid \boldsymbol{x}_i \in \mathcal{D}_f\}$ {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 $\text{sim}(g_\phi(\boldsymbol{x}_q), \boldsymbol{k}_j)$
$\hat{\boldsymbol{y}} \leftarrow \sum_{j \in \mathcal{N}_K(\boldsymbol{x}_q)} w_j \cdot f_\theta([\boldsymbol{z}_q, h_\psi(\boldsymbol{v}_j)])$
return $\hat{\boldsymbol{y}}$

D Computational Efficiency and Qualitative Analysis

D.1 Time Complexity

In this subsection we include the run-time efficiency of the studied models against MUNKEY in Table 2. A core advantage of the *unlearning by design* paradigm is the elimination of post-hoc optimization. Table 2 provides a comprehensive runtime efficiency report across all methods. 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 ($\approx$ 0s). Forgetting is reduced to simply dropping the samples' key-value pairs from the memory bank. We also report the per-epoch training time for standard ViT and MUNKEY, indicating that the overhead of 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.

Table 2: Run-Time Efficiency comparison (seconds) on DermaMNIST and CIFAR-10.

	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	$154.4_{\pm 0.3}$	$24.7_{\pm 0.2}$	$1478.2_{\pm 273}$	$174.6_{\pm 1.6}$
SSD	$23.1_{\pm 4.0}$	$24.7_{\pm 0.2}$	$105.8_{\pm 13.6}$	$174.6_{\pm 1.6}$
AMUN	$313.4_{\pm 6.8}$	$24.7_{\pm 0.2}$	$1940.5_{\pm 3.7}$	$174.6_{\pm 1.6}$
MUNKEY	$\approx$ **0**	$25.9_{\pm 0.3}$	$\approx$ **0**	$178.5_{\pm 2.7}$

D.2 Further Visualizations in MUNKEY

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. top rows in Figure 3, the key encoder continues to retrieve visually similar images $\mathcal{N}_K(\boldsymbol{x}_i)$, even if belonging to different classes, and MUNKEY retrieves the updated $\{\boldsymbol{v}_j \mid j \in \mathcal{N}_K(\boldsymbol{x}_i)\}$ from its memory without changing the final predicted class. In contrast, for a training sample $\boldsymbol{x}_i$ marked for forgetting, e.g. bottom rows in Figure 3, 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 the retrieved $\{\boldsymbol{v}_j \mid j \in \mathcal{N}_K(\boldsymbol{x}_i)\}$ and alters the final class prediction.

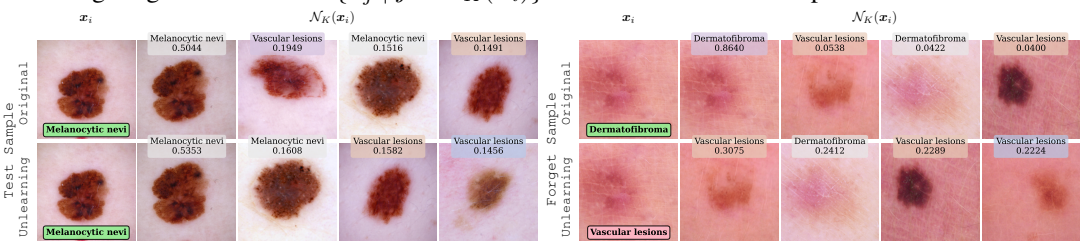

Figure 3: KNN-based retrieval on DermaMNIST on a test (left) and forget (right) samples. Rows are the original and unlearning behavior per sample. Colors indicate retrieval outcomes: correct match, wrong-class, forgotten, and new neighbor; prediction boxes: correct and incorrect.

E Results on Additional Datasets

In this section, we provide detailed results for the remaining benchmarks, CIFAR-100, BloodMNIST, and Tiny ImageNet, using the suite of metrics defined in **??**, 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 4, 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 3, 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 3: Comparison of machine unlearning performance on a random forget set of 10% of the training data on Bloodmnist.

Model	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**
Retrain (V)	$96.88_{\pm 0.16}$	$99.99_{\pm 0.00}$	$96.43_{\pm 0.42}$	$49.60_{\pm 0.28}$	–
Retrain (M)	$97.43_{\pm 0.19}$	$99.83_{\pm 0.06}$	$97.15_{\pm 0.18}$	$49.43_{\pm 0.55}$	–
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	$96.34_{\pm 0.08}$	$99.61_{\pm 0.19}$	$98.94_{\pm 0.34}$	$51.92_{\pm 0.55}$	$1.44_{\pm 0.22}$
MUNKEY	$97.51_{\pm 0.13}$	$99.85_{\pm 0.07}$	$99.27_{\pm 0.20}$	$50.90_{\pm 0.66}$	$\mathbf{0.92}_{\pm 0.23}$

Table 4: Comparison of machine unlearning performance on a random forget set of 10% of the training data on PathMNIST.

Model	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**
Retrain (V)	$98.96_{\pm 0.07}$	$100.00_{\pm 0.00}$	$98.78_{\pm 0.17}$	$50.09_{\pm 0.24}$	–
Retrain (M)	$97.55_{\pm 0.15}$	$99.49_{\pm 0.13}$	$97.21_{\pm 0.22}$	$49.60_{\pm 0.41}$	–
KNN	$92.84_{\pm 4.49}$	$98.40_{\pm 0.01}$	$97.31_{\pm 0.12}$	$54.13_{\pm 4.02}$	–
CF	$96.65_{\pm 0.21}$	$99.74_{\pm 0.12}$	$99.50_{\pm 0.41}$	$54.10_{\pm 0.12}$	$0.77_{\pm 0.09}$
CF-k	$96.93_{\pm 0.25}$	$99.96_{\pm 0.04}$	$100.00_{\pm 0.00}$	$52.50_{\pm 0.47}$	$0.88_{\pm 0.07}$
EU-k	$96.94_{\pm 0.26}$	$99.94_{\pm 0.07}$	$99.97_{\pm 0.04}$	$52.90_{\pm 0.42}$	$0.89_{\pm 0.08}$
ℓ_1-sparse	$90.92_{\pm 1.65}$	$92.14_{\pm 1.87}$	$91.55_{\pm 1.48}$	$50.99_{\pm 0.62}$	$5.20_{\pm 0.30}$
NegGrad+	$96.48_{\pm 0.25}$	$99.77_{\pm 0.05}$	$97.80_{\pm 0.04}$	$51.45_{\pm 0.56}$	$\mathbf{0.50}_{\pm 0.10}$
Amnesiac	$96.27_{\pm 0.47}$	$99.63_{\pm 0.42}$	$97.49_{\pm 0.43}$	$40.85_{\pm 1.57}$	$5.67_{\pm 0.39}$
SCRUB	$95.44_{\pm 0.52}$	$97.27_{\pm 0.88}$	$96.52_{\pm 0.76}$	$51.01_{\pm 0.25}$	$2.91_{\pm 0.06}$
SSD	$96.84_{\pm 0.28}$	$99.88_{\pm 0.16}$	$99.90_{\pm 0.10}$	$51.72_{\pm 0.25}$	$0.84_{\pm 0.11}$
AMUN	$96.34_{\pm 0.08}$	$99.61_{\pm 0.19}$	$98.94_{\pm 0.34}$	$51.92_{\pm 0.55}$	$0.75_{\pm 0.06}$
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 5 and Table 6, 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 (V) 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 5: Comparison of machine unlearning performance on a random forget set of 10% of the training data on CIFAR-100.

Model	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**
Retrain (V)	$47.17_{\pm 0.61}$	$87.37_{\pm 0.41}$	$47.00_{\pm 0.39}$	$49.88_{\pm 0.26}$	–
Retrain (M)	$74.76_{\pm 0.13}$	$99.90_{\pm 0.03}$	$74.73_{\pm 0.22}$	$49.98_{\pm 0.29}$	–
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	$46.60_{\pm 1.69}$	$89.58_{\pm 34.10}$	$87.55_{\pm 39.30}$	$76.23_{\pm 1.52}$	$17.42_{\pm 1.44}$
AMUN	$46.82_{\pm 1.26}$	$90.04_{\pm 1.22}$	$75.09_{\pm 1.06}$	$67.38_{\pm 0.45}$	$12.15_{\pm 0.57}$
MUNKEY	$74.75_{\pm 0.46}$	$99.90_{\pm 0.03}$	$78.07_{\pm 0.19}$	$53.51_{\pm 0.23}$	$\mathbf{1.72}_{\pm 0.17}$

Table 6: Comparison of machine unlearning performance on a random forget set of 10% of the training data on Tiny ImageNet.

Model	**TA**	**RA**	**FA**	**MIA**	**Avg Gap ↓**
Retrain (V)	$36.74_{\pm 0.76}$	$76.78_{\pm 0.29}$	$36.30_{\pm 0.49}$	$49.87_{\pm 0.23}$	–
Retrain (M)	$68.51_{\pm 0.45}$	$99.55_{\pm 0.15}$	$68.95_{\pm 0.40}$	$50.96_{\pm 0.37}$	–
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+	$34.73_{\pm 0.25}$	$86.54_{\pm 0.61}$	$49.34_{\pm 0.21}$	$62.31_{\pm 0.05}$	$19.31_{\pm 0.30}$
Amnesiac	$34.94_{\pm 0.45}$	$88.36_{\pm 0.34}$	$58.47_{\pm 0.31}$	$68.11_{\pm 0.21}$	$\underline{13.45}_{\pm 0.30}$
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	$35.32_{\pm 1.17}$	$86.86_{\pm 1.22}$	$65.30_{\pm 1.40}$	$70.80_{\pm 0.20}$	$15.36_{\pm 0.60}$
MUNKEY	$68.58_{\pm 0.19}$	$99.20_{\pm 0.24}$	$69.04_{\pm 0.38}$	$51.06_{\pm 0.31}$	$\mathbf{0.15}_{\pm 0.23}$

F Evaluation of Model Variants

F.1 Effect of Pathway Aggregation

As discussed in Section 3, 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** (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 7 to 12 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 2. 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 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, they perform generally similarly, with Ensembling providing the smallest Avg Gap and most stable pathway balance. However, if inference time 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.

Table 7: 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 8: 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	$92.36_{\pm 0.03}$	$95.69_{\pm 0.07}$	$92.12_{\pm 0.33}$	$49.86_{\pm 0.14}$	–	–
Key Encoder	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 9: 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	$97.56_{\pm 0.10}$	$99.62_{\pm 0.15}$	$97.27_{\pm 0.10}$	$49.53_{\pm 0.49}$	–	–
Key Encoder	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 10: 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	$99.05_{\pm 0.04}$	$100.0_{\pm 0.00}$	$98.91_{\pm 0.10}$	$49.86_{\pm 0.22}$	–	–
Key Encoder	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 11: 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	$72.94_{\pm 0.50}$	$87.11_{\pm 0.46}$	$72.58_{\pm 0.40}$	$49.85_{\pm 0.21}$	–	–
Key Encoder	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 12: 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	$67.46_{\pm 0.47}$	$78.75_{\pm 0.37}$	$67.35_{\pm 0.67}$	$49.81_{\pm 0.55}$	–	–
Key Encoder	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	$66.79_{\pm 0.13}$	$78.31_{\pm 0.24}$	$71.84_{\pm 0.40}$	$53.30_{\pm 0.12}$	$0.5898_{\pm 0.0126}$	$2.27_{\pm 0.29}$

F.2 Effect of the Key Encoder

We evaluate the impact of the frozen key encoder g_ϕ by comparing the standard ViT-Tiny pretrained 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 demonstrates that MUNKEY does not require label-informed keys to effectively decouple and delete instance-specific information. In Table 13 we see how 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.

F.3 Effect of Neighborhood Size

As discussed in Appendix F.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

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

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

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.

Table 13: Comparison of the performance of MUNKEY 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	Retrained (CLIP)	$92.61_{\pm 0.04}$	$98.23_{\pm 0.02}$	$93.19_{\pm 0.15}$	$50.82_{\pm 0.04}$	–
	Retrained (DINO-v2)	$95.32_{\pm 0.02}$	$99.96_{\pm 0.01}$	$95.68_{\pm 0.22}$	$50.71_{\pm 0.34}$	–
	Retrained (ViT)	$92.93_{\pm 0.16}$	$99.93_{\pm 0.02}$	$92.76_{\pm 0.45}$	$49.97_{\pm 0.25}$	–
	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 Pretrained-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	Retrained (CLIP)	$81.08_{\pm 0.02}$	$91.59_{\pm 0.76}$	$79.62_{\pm 1.41}$	$49.58_{\pm 0.89}$	–
	Retrained (DINO-v2)	$81.90_{\pm 0.29}$	$95.66_{\pm 0.56}$	$80.10_{\pm 0.64}$	$50.17_{\pm 0.91}$	–
	Retrained (ViT)	$80.80_{\pm 0.54}$	$97.47_{\pm 0.15}$	$80.10_{\pm 0.99}$	$50.53_{\pm 0.89}$	–
	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 Pretrained-ViT/B	$81.06_{\pm 0.41}$	$97.64_{\pm 0.22}$	$84.10_{\pm 1.58}$	$52.16_{\pm 0.53}$	$\mathbf{1.52}_{\pm 0.56}$

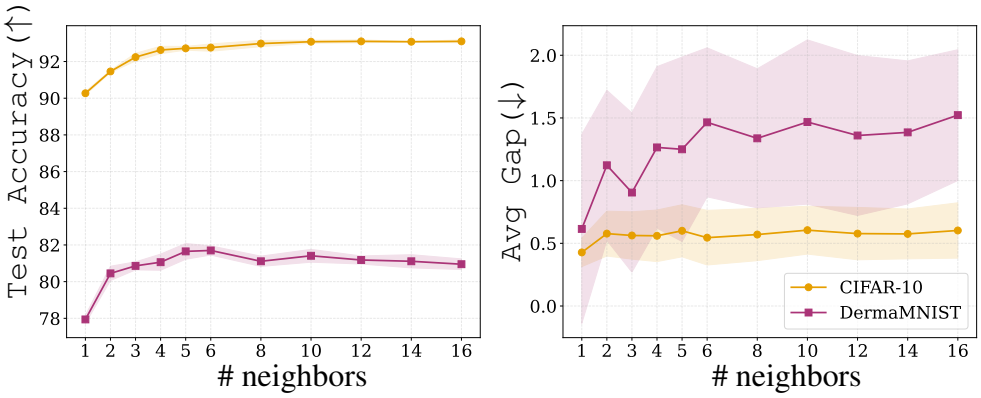

Figure 4: Performance sensitivity to the number of neighbors on DermaMNIST and CIFAR-10. We report Test Accuracy (%) (left) and Avg Gap (right) across different neighborhood sizes.

G Sensitivity Score and Model Selection

To prevent *pathway collapse*, where the model either ignores the memory (resulting in poor unlearning) or becomes a simple lookup table (resulting in poor generalization), we introduce the Pathway Sensitivity Score (P_s). P_s quantifies the balance of information flow by evaluating the model's performance when one pathway is replaced by the learned null tokens $\emptyset$. Specifically, 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:

$$P_s = \frac{|\mathcal{A}_{img} - \mathcal{A}_{tok}|}{\mathcal{A}_{both} + \epsilon}, \tag{2}$$

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 sensitivity threshold that bounds the disparity between pathway contributions, preventing modality redundancy. P_s enables a 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, i.e. the accuracy difference between validation and forget sets. Among candidates with a comparable gap, the one minimizing P_s is selected for an informational equilibrium that is both effective at forgetting and architecturally balanced.

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. More in depth, 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 forget accuracy. 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 5 and Figure 6 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.

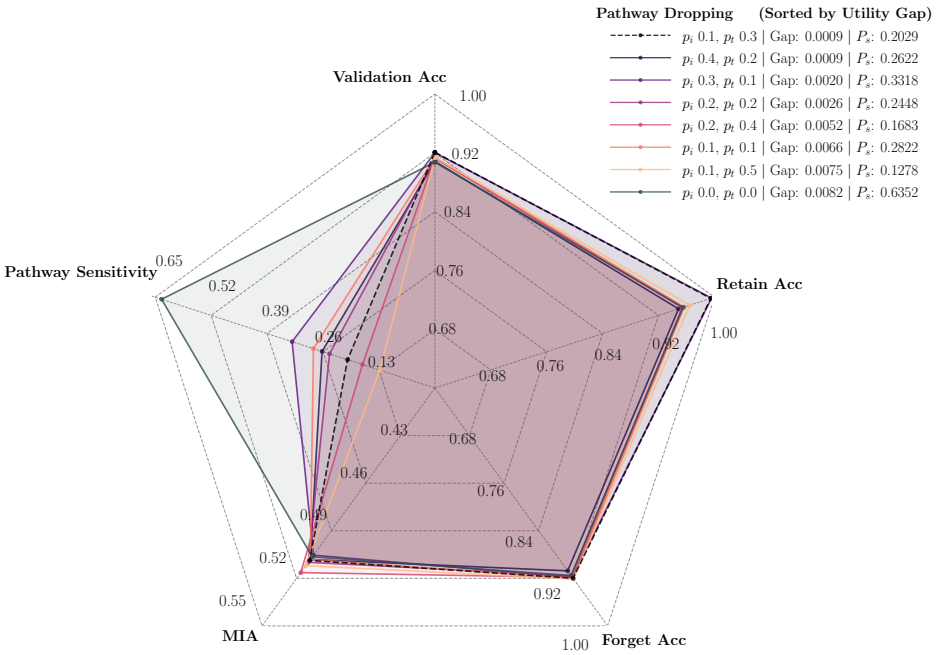

Figure 5: Overview of the performance metrics under varying image and token dropout probabilities in MUNKEY training in CIFAR-10 for visualization of model selection.

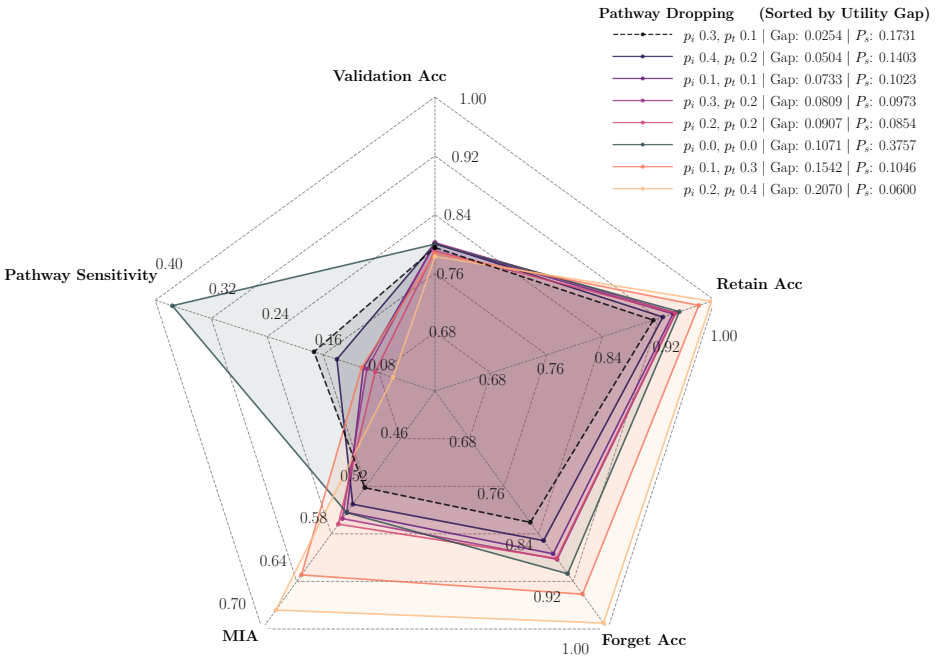

Figure 6: Overview of the resulting performance metrics under varying image and token dropout probabilities in MUNKEY training in DermaMNIST for visualization of model selection.