Appendix

The appendix is organized as follows:

- Implementation and experimental details are presented in Sec. A.
- The complete proof of Claim 1 is provided in Sec. B.
- Additional qualitative results for GMM generation are included in Sec. C.
- Additional results on random unlearning for classification are reported in Sec. D.

A. Implementation and Experiment Details

Details for task performance experiments. For SPM on classification, ResNet18 is trained using SGD with a learning rate of 0.1 and a batch size of 256. We train from scratch for 100 epochs on CIFAR-10 and fine-tune the ImageNet-pretrained model for 160 epochs on ImageNet. For each mini-batch, we pair each query with 1,024 randomly sampled instances from the training set, which serve as the inputted set during training. All experiments are repeated with five random seeds.

As for SPM on generation, we train a DDPM for 2,000 epochs with a learning rate of 0.0005 and a batch size of 256. During training, each batch is paired with an inputted set of 256 samples drawn from the training set, serving as the non-parametric memory for SPM. During evaluation, we generate 5,000 samples per class using DDIM [50] with 50 sampling steps. Runtime is measured based on the time required to generate 1,000 images. All the experiments are conducted on Nvidia L40S.

Baselines for task performance. For classification, we adopt ResNet18 and ResNet18-KNN as the parametric and non-parametric baselines, respectively. We train ResNet18 by following the implementation details provided by Fan et al. [16]. ResNet18-KNN performs K-nearest neighbor classification on the embeddings extracted from ResNet18, with the optimal value of K determined to be 50 for achieving the highest accuracy. In both ResNet18-KNN and SPM-R, the samples from the inputted set with the minimum L2 distance to the query are retrieved, with a retrieval size set of 256. This retrieval process is implemented using the FAISS (Facebook AI Similarity Search) library [14], which enables efficient similarity search through product quantization (PQ) codes, allowing for comparison of quantized representations and significantly reducing retrieval latency.

For generation, we adopt DDPM and GMM as the parametric and non-parametric baselines, respectively. We train DDPM following the implementation provided by Fan et al. [16]. For GMM, we model the target distribution $p_c(x)$ for each class using 1,000 Gaussian components, with the mean μ_i and diagonal covariance Σ_i of each component estimated using the Expectation-Maximization (EM) algorithm. The resulting model can be expressed as:

$$p_c(x) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{N}(x|\mu_i, \Sigma_i), \tag{20}$$

where $\mathcal{N}(x|\mu_i, \Sigma_i)$ represents the Gaussian distribution with mean μ_i and covariance Σ_i for the i-th component, and n is the total number of Gaussian components.

Details for unlearning performance experiments. For SPM in classification, we construct the inputted set using the remaining classes, *i.e.*, $\mathcal{T} \setminus \mathcal{U}$. Then, we apply the clustering strategy to aggregate the embeddings of samples from each class into an average.

For SPM on generation, if the class is unlearned, the inputted set is composed of images from another remaining class. For example, if the model is supposed to unlearn cat, we can use the images of planes to replace images of cats as the inputted set. Otherwise, the inputted set is sampled from the target class.

Baselines for unlearning performance. We utilize the code released from Fan et al. [16] and Wu and Harandi [60] for classification, and from Huang et al. [28] for generation, adhering to their instructions to obtain the baseline results.

Comparison with EraseDiff We utilize the code provided by EraseDiff [61] and present our results in Tab. 7. However, we were unable to fully replicate the results reported in their original paper. Furthermore, it is worth noting that their study presents results for only a single class in the class-wise unlearning experiment. This limited evaluation makes it challenging to assess the general effectiveness of the approach across a broader range of classes.

More qualitative results Fig. 4 shows more qualitative results for SPM unlearning, where five CIFAR-10 classes are removed simultaneously. We visualize generated samples for both the unlearned and remaining classes. These results demonstrate that the proposed SPM supports precise, class-level forgetting without harming the generation quality of the remaining categories.

Backbone	Method	Automobile		Cat		Dog		Horse		Truck		t_{setup} (s) ↓
		ΔUA ↓	$\Delta\text{FID}_\text{R}$ ↓	ΔUA ↓	$\Delta\text{FID}_\text{R}$ ↓	ΔUA ↓	$\Delta\text{FID}_\text{R}$ ↓	ΔUA ↓	$\Delta\text{FID}_\text{R}$ ↓	ΔUA ↓	$\Delta\text{FID}_\text{R}$ ↓	
DDPM	Retrain	0.00 (100.00)	0.00 (11.21)	0.00 (99.98)	0.00 (10.84)	0.00 (100.00)	0.00 (10.88)	0.00 (99.98)	0.00 (10.00)	0.00 (99.90)	0.00 (10.15)	145601.4
	EraseDiff [61]	28.06 (71.94)	2.66 (8.55)	72.42 (27.56)	1.96 (8.88)	51.40 (48.60)	2.16 (8.72)	42.24 (57.76)	1.20 (8.80)	38.10 (61.80)	1.80 (8.35)	1429.6
SPM	Retrain	0.00 (99.76)	0.00 (7.40)	0.00 (99.26)	0.00 (7.15)	0.00 (99.78)	0.00 (6.52)	0.00 (99.84)	0.00 (6.89)	0.00 (99.64)	0.00 (7.28)	90935.9
	Ours	**0.10** (99.86)	**0.38** (7.78)	**0.04** (99.22)	**0.50** (7.65)	**0.04** (99.82)	**0.08** (6.60)	**0.08** (99.76)	**0.29** (7.18)	**0.06** (99.58)	**0.40** (7.68)	<1

Table 7. **CIFAR-10 class-wise forgetting on generation.** Comparison with EraseDiff.

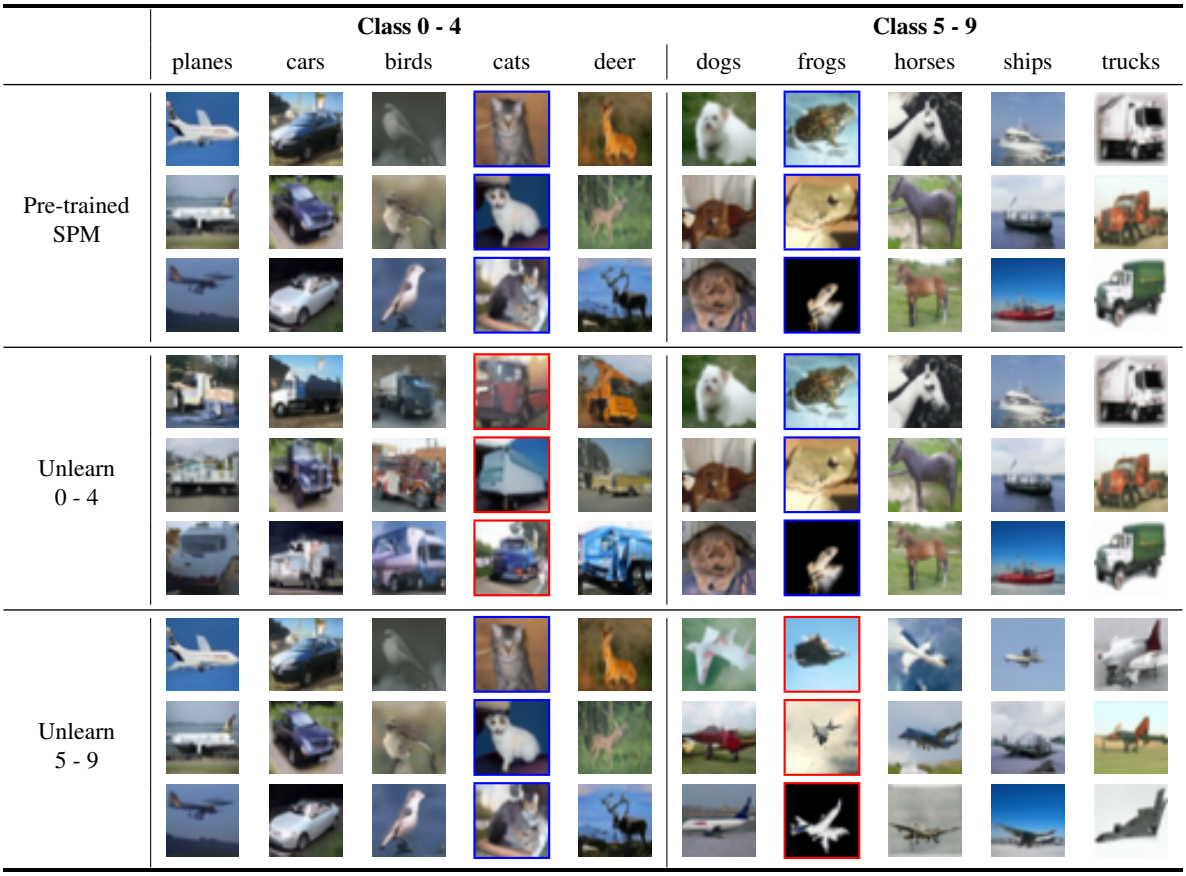

Figure 4. **Qualitative comparison of unlearned SPM**. When a class is unlearned (*e.g.*, cats or frogs framed in red), the model avoids generating that concept and produces samples resembling remaining classes (*e.g.*, a truck replacing a cat). Additionally, the generations from the remaining classes are left unchanged (framed in blue).

B. Proof of Claim 1

Claim 1. *Given that the oracle margin* $\gamma(\boldsymbol{x}) = F'(\boldsymbol{x})[y_1] - F'(\boldsymbol{x})[y_2] \geq \gamma_{\min} > 0$ *for all* $\boldsymbol{x} \in \mathcal{V}$*, where* y_1, y_2 *denote the indices of the largest and second predicted probabilities from the oracle model. We have*

$$\Delta Acc(\mathcal{V}) \leq PG_{\text{H}}(\tilde{F}, F') \;\leq\; \frac{\sqrt{2}}{\gamma_{\min}} \sqrt{PG_{\text{S}}(\tilde{F}, F')}. \tag{19}$$

Proof. **Proof of the first inequality in Eq. (19).** For every $\boldsymbol{x} \in \mathcal{V}$, let $\boldsymbol{y}^\star(\boldsymbol{x})$ be the ground-truth label. Then

$$\left|\mathbf{1}[\tilde{\boldsymbol{y}}(\boldsymbol{x}) = \boldsymbol{y}^\star(\boldsymbol{x})] - \mathbf{1}[\boldsymbol{y}'(\boldsymbol{x}) = \boldsymbol{y}^\star(\boldsymbol{x})]\right| \;\leq\; \mathbf{1}[\tilde{\boldsymbol{y}}(\boldsymbol{x}) \neq \boldsymbol{y}'(\boldsymbol{x})]. \tag{21}$$

Averaging over $\boldsymbol{x} \in \mathcal{V}$ yields

$$\Delta\,\text{Acc}(\mathcal{V}) \;\leq\; \text{PG}_{\text{H}}(\tilde{F}, F'). \tag{22}$$

Proof of the second inequality in Eq. (19). Fix $\boldsymbol{x} \in \mathcal{V}$ and recall the oracle margin $\gamma(\boldsymbol{x}) = F'(\boldsymbol{x})[y_1] - F'(\boldsymbol{x})[y_2] \geq \gamma_{\min} > 0$, where y_1, y_2 are the indices of the largest and second largest coordinates of $F'(\boldsymbol{x})$. If $\tilde{\boldsymbol{y}}(\boldsymbol{x}) = \boldsymbol{y}'(\boldsymbol{x}) = y_1$, then $\mathbf{1}[\tilde{\boldsymbol{y}}(\boldsymbol{x}) \neq \boldsymbol{y}'(\boldsymbol{x})] = 0$. Otherwise, set $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{y}}(\boldsymbol{x}) \neq y_1$. Since $F'(\boldsymbol{x})[\tilde{\boldsymbol{y}}] \leq F'(\boldsymbol{x})[y_2]$, we have

$$\gamma(\boldsymbol{x}) \;\leq\; F'(\boldsymbol{x})[y_1] - F'(\boldsymbol{x})[\tilde{\boldsymbol{y}}]. \tag{23}$$

Adding and subtracting $\tilde{F}(\boldsymbol{x})[y_1]$ and $\tilde{F}(\boldsymbol{x})[\tilde{\boldsymbol{y}}]$, and using $\tilde{F}(\boldsymbol{x})[y_1] \leq \tilde{F}(\boldsymbol{x})[\tilde{\boldsymbol{y}}]$, we obtain

$$\begin{aligned} \gamma(\boldsymbol{x}) \;&\leq\; \big|F'(\boldsymbol{x})[y_1] - \tilde{F}(\boldsymbol{x})[y_1]\big| + \big|F'(\boldsymbol{x})[\tilde{\boldsymbol{y}}] - \tilde{F}(\boldsymbol{x})[\tilde{\boldsymbol{y}}]\big| && (24)\\ &\leq\; \big\|\tilde{F}(\boldsymbol{x}) - F'(\boldsymbol{x})\big\|_1. && (25) \end{aligned}$$

Hence,

$$\mathbf{1}[\tilde{\boldsymbol{y}}(\boldsymbol{x}) \neq \boldsymbol{y}'(\boldsymbol{x})] \;\leq\; \frac{\big\|\tilde{F}(\boldsymbol{x}) - F'(\boldsymbol{x})\big\|_1}{\gamma_{\min}}. \tag{26}$$

By Pinsker's inequality,

$$\big\|\tilde{F}(\boldsymbol{x}) - F'(\boldsymbol{x})\big\|_1 \;\leq\; \sqrt{2\, D_{\mathrm{KL}}\big(\tilde{F}(\boldsymbol{x}), F'(\boldsymbol{x})\big)}. \tag{27}$$

Therefore,

$$\mathbf{1}[\tilde{\boldsymbol{y}}(\boldsymbol{x}) \neq \boldsymbol{y}'(\boldsymbol{x})] \;\leq\; \frac{\sqrt{2}}{\gamma_{\min}} \sqrt{D_{\mathrm{KL}}\big(\tilde{F}(\boldsymbol{x}), F'(\boldsymbol{x})\big)}. \tag{28}$$

Averaging over $\boldsymbol{x} \in \mathcal{V}$ and applying Jensen's inequality to $t \mapsto \sqrt{t}$ yields

$$\mathrm{PG}_{\mathtt{H}}(\tilde{F}, F') \;\leq\; \frac{\sqrt{2}}{\gamma_{\min}} \sqrt{\frac{1}{|\mathcal{V}|} \sum_{\boldsymbol{x} \in \mathcal{V}} D_{\mathrm{KL}}\big(\tilde{F}(\boldsymbol{x}), F'(\boldsymbol{x})\big)} \;=\; \frac{\sqrt{2}}{\gamma_{\min}} \sqrt{\mathrm{PG}_{\mathtt{S}}(\tilde{F}, F')}. \tag{29}$$

Combining the two parts gives

$$\Delta\,\mathrm{Acc}(\mathcal{V}) \;\leq\; \mathrm{PG}_{\mathtt{H}}(\tilde{F}, F') \;\leq\; \frac{\sqrt{2}}{\gamma_{\min}} \sqrt{\mathrm{PG}_{\mathtt{S}}(\tilde{F}, F')}. \tag{30}$$

□

C. Qualitative Results from GMM

Fig. 5 presents the qualitative results of GMM on CIFAR-10. Since GMM generates images by superimposing multiple Gaussian components, the generated images exhibit significant noise and blurriness. This highlights the limitations of traditional semi/non-parametric models in handling complex generative tasks, underscoring the necessity of integrating parametric models to effectively address such challenges.

D. Random Unlearning Results on Classification

In Tab. 8, we present the results of random unlearning on CIFAR-10 classification. Our proposed method achieves low $\mathrm{PG}_{\mathtt{H}}$ and $\mathrm{PG}_{\mathtt{S}}$ in both the 10% and 50% random unlearning scenarios. Moreover, the gaps between our unlearned model and the retrained model in terms of ΔUA, ΔRA, and ΔTA are minimal, demonstrating the effectiveness of our approach. It is worth noting that evaluating this task solely based on UA is not meaningful, since the unlearned data in random unlearning are essentially unseen samples to the model. Consequently, the model is expected to generalize to such data. Therefore, $\mathrm{PG}_{\mathtt{H}}$ and $\mathrm{PG}_{\mathtt{S}}$ serve as more appropriate evaluation metrics in this context, as they measure how closely the unlearned model approximates the logits of the retrained model.

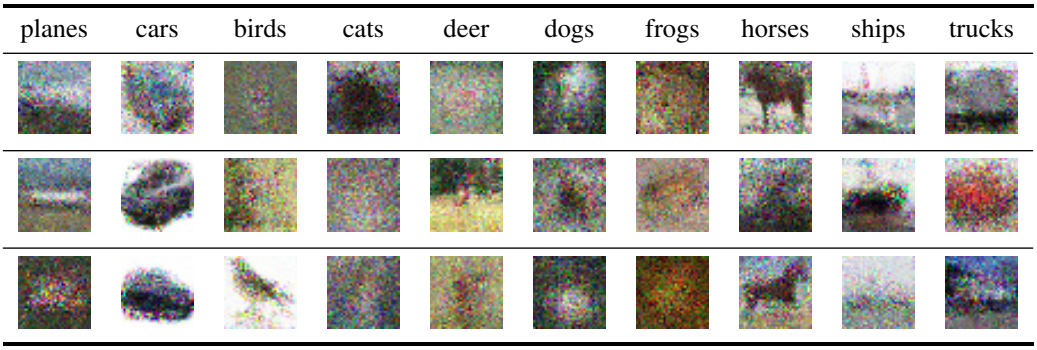

Figure 5. **Generated images of GMM on CIFAR-10.** The low-quality outputs demonstrate the limitations of traditional non-parametric models in handling complex generative tasks.

Backbone	Method	Random Unlearning (10%)					Random Unlearning (50%)				
		$PG_H \downarrow$	$PG_S \downarrow$	ΔUA $\downarrow$	ΔRA $\downarrow$	ΔTA $\downarrow$	$PG_H \downarrow$	$PG_S \downarrow$	ΔUA $\downarrow$	ΔRA $\downarrow$	ΔTA $\downarrow$
ResNet18	Retrain	0.00	0.00	0.00 (5.35)	0.00 (100.00)	0.00 (94.25)	0.00	0.00	0.00 (8.48)	0.00 (100.00)	0.00 (91.08)
	GA [52]	**5.23**	0.17	4.80 (0.55)	4.80 (99.52)	0.48 (94.57)	8.05	0.37	7.94 (0.54)	0.49 (99.51)	3.45 (94.53)
	FT [56]	5.89	0.19	4.46 (0.89)	0.16 (99.84)	0.32 (93.93)	8.50	0.36	7.52 (0.96)	0.18 (99.82)	2.67 (93.75)
	IU [31]	26.60	1.69	17.15 (22.50)	22.32 (77.68)	21.09 (73.16)	83.63	8.29	74.83 (83.32)	83.29 (16.71)	74.53 (16.55)
	BE [10]	5.30	0.17	4.83 (0.52)	0.48 (99.52)	0.29 (94.54)	34.70	1.22	17.33 (25.81)	25.93 (74.07)	23.37 (67.71)
	BS [10]	5.31	0.17	4.82 (0.53)	0.49 (99.51)	0.28 (94.53)	26.55	0.85	8.82 (17.30)	17.36 (82.64)	15.39 (75.69)
	ℓ_1-sparse [29]	6.03	0.19	4.29 (1.06)	0.24 (99.76)	0.50 (93.75)	8.53	0.35	7.53 (0.95)	**0.14** (99.86)	2.67 (93.75)
	SalUn [16]	5.40	0.15	3.89 (1.46)	0.14 (99.86)	**0.08** (94.17)	7.94	**0.33**	7.34 (1.14)	0.52 (99.48)	2.53 (93.61)
	MUNBa [60]	5.39	**0.14**	4.39 (0.96)	**0.03** (99.97)	0.19 (94.44)	12.94	1.50	**2.62** (5.86)	3.26 (96.74)	**2.09** (88.99)
ResNet18-KNN	Retrain	0.00	0.00	0.00 (0.02)	0.00 (99.98)	0.00 (94.45)	0.00	0.00	0.00 (0.02)	0.00 (99.98)	0.00 (94.45)
ResNet18-SPM	Retrain	0.00	0.00	0.00 (0.18)	0.00 (99.43)	0.00 (93.89)	0.00	0.00	0.00 (4.89)	0.00 (95.96)	0.00 (91.08)
	Ours	5.54	0.19	**0.16** (0.02)	0.55 (99.98)	0.56 (94.45)	**7.83**	0.36	4.87 (0.02)	4.02 (99.98)	3.37 (94.45)

Table 8. **CIFAR-10 random unlearning on classification.** Results are averaged over 10 scenarios. We observe that our method can achieve close performance to the retrained model. The UA, RA, and TA values are reported in parentheses with the corresponding gaps.