Moderating the Generalization of Score-based Generative Model

Supplementary Material

1. Compatible with Moderated DDPM

Denoising Diffusion Probabilistic models (DDPMs) [17,28] are a type of generative model that generate samples from a distribution via an iterative Markov denoising method. Initially, a sample $\mathbf{x}_T$ is drawn from a Gaussian distribution and subsequently denoised over T time steps, ultimately resulting in a clean sample $\mathbf{x}_0$. During the training phase, the model learns to predict the noise $\epsilon_\theta(\mathbf{x}_t, t)$ that needs to be removed from the sample $\mathbf{x}_t$ using the following reweighted variational bound:

$$L_g = \mathrm{E}_{\mathbf{x}_0,\epsilon}\left[\rho \cdot \left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^g + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2\right], \quad (1)$$

where $\mathbf{x}_0^g \in \mathcal{D}_g$, $\rho_g = \frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}$ and $\beta_1,\ldots,\beta_T$ is a variance schedule used for adds Gaussian noise to the data in the forward process, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$ and

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0,I). \quad (2)$$

While our method, MSGM, is primarily designed for score-based generative models, it is also compatible with DDPM models. According to the relationship between score and $\epsilon_\theta(\mathbf{x}_t)$:

$$\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t), \quad (3)$$

we derive the following unlearning method by applying our method within the DDPM framework.

Orthogonal-MSGM for DDPM

$$L_f = \mathrm{E}_{\mathbf{x}_0,\epsilon}\left[\rho_{ort} \cdot \left\| \epsilon \cdot \epsilon_\theta^{\mathbf{u}}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^f + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2\right], \quad (4)$$

where $\mathbf{x}_0^f \in \mathcal{D}_f$, $\rho_{ort} = \frac{\beta_t^2}{2\sigma_t^2\alpha_t}$.

Obtuse-MSGM for DDPM

$$L_f = \mathrm{E}_{\mathbf{x}_0,\epsilon}\left[\rho_{obt} \cdot \left(\epsilon \cdot \epsilon_\theta^{\mathbf{u}}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^f + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right)\right], \quad (5)$$

where $\mathbf{x}_0^f \in \mathcal{D}_f$, $\rho_{obt} = -\frac{\beta_t^2}{2\sigma_t^2\alpha_t\sqrt{(1-\bar{\alpha}_t)}}$.

Similarly, the final loss of unlearning DDPM modeling can be solved by Eq.(9).

2. Impact of Forgetting Data Proportion

To further evaluate the robustness of our approach, we conducted additional experiments to examine the impact of reducing the proportion of the forgetting dataset $\mathcal{D}_f$ on the method's effectiveness. Specifically, we performed experiments on the MNIST dataset using the VE SDE model, with digits '3' and '7' designated as the

Table 1. Results of different ratios of D_f on the MNIST Dataset. ratio=1 means 100% removing forgetting data from the training dataset, ratio= 0 means Standard traning.

ratio	Class	UR(%)(↓)	NLL Test		ratio	Class	UR(%)(↓)	NLL Test	
1.0	3	0.4	$\mathcal{D}_g$	3.92	0.8	3	2.3	$\mathcal{D}_g$	3.93
	7	0.8				7	3.5		
	3 and 7	1.2	$\mathcal{D}_f$	13.23		3 and 7	5.8	$\mathcal{D}_f$	6.35
0.6	3	3.6	$\mathcal{D}_g$	3.93	0.4	3	6.8	$\mathcal{D}_g$	4.09
	7	6.5				7	9.2		
	3 and 7	10.1	$\mathcal{D}_f$	4.17		3 and 7	16.0	$\mathcal{D}_f$	4.17
0.2	3	6.5	$\mathcal{D}_g$	4.34	0	3	11.0	$\mathcal{D}_g$	2.82
	7	11.3				7	15.8		
	3 and 7	17.8	$\mathcal{D}_f$	4.38		3 and 7	26.8	$\mathcal{D}_f$	2.78

NSFG data to be forgotten. In these experiments, we use 80%, 60%, 40% and 20% of $\mathcal{D}_f$ for training, while the remaining 20%, 40%, 80% and 60% are included in the remaining dataset $\mathcal{D}_g$. The results in Tab. 1 demonstrate that reducing the proportion of the NSFG data $\mathcal{D}_f$ weakens the unlearning effectiveness, as indicated by higher UR and lower NLL values on $\mathcal{D}_f$. However, the model's performance on the dataset $\mathcal{D}_g$ remains stable, highlighting its ability to recover generalization. Compared to the Standard approach (ratio = 0), the MSGM mechanism consistently achieves better unlearning, even with reduced $\mathcal{D}_f$.

3. Optimization Choices

In Sec.4.5, we discuss the selection of optimization strategies. Considering the different optimization approaches for $\mathcal{D}_f$ and $\mathcal{D}_g$, we conduct experiments on the MNIST dataset, and the loss curves for the two strategies are shown in the Fig. 1 and Fig. 2 .

The top row of Fig. 1 illustrates the results when Orthogonal-MSGM uses a Simultaneously Updating strategy. It is evident that this approach fails to effectively minimize the loss for both $\mathcal{D}_f$ and $\mathcal{D}_g$. Specifically, although the losses of $\mathcal{D}_f$ and $\mathcal{D}_g$ exhibit stability, the final stable values remain significantly high. This indicates that the model fails to learn meaningful content under this strategy. Moreover, in the case of Obtuse-MSGM (as shown in the second row of the Fig. 1), the loss of $\mathcal{D}_f$ (i.e., L_f) reaches very small negative values due to being easier to optimize. This, in turn, hinders the convergence of L_g because the overall optimization prioritizes minimizing L_f, thereby neglecting the optimization of L_g.

To address this issue, we adopt a strategy where L_f is updated every four steps instead of being synchronized with L_g. This asynchronous approach significantly improves optimization, as shown in Fig. 2. The loss for $\mathcal{D}_f$ stabilizes, and L_g decreases consistently, indicating that the model effectively learns useful content. This strategy balances the optimization process and avoids the pitfalls of synchronous updates, leading to better overall performance.
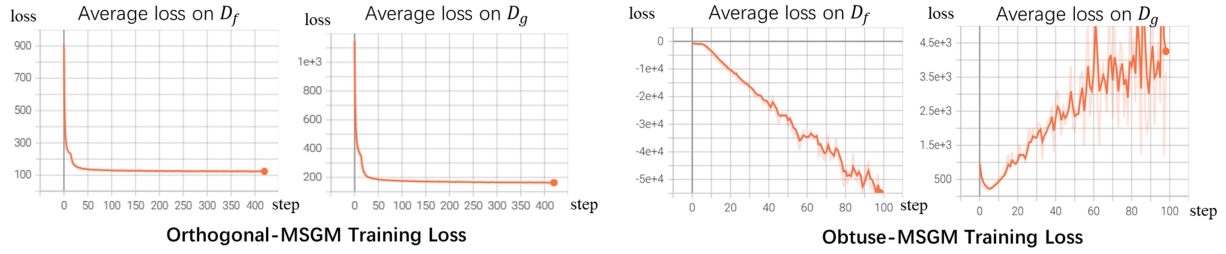
Figure 1. Loss curves for Simultaneously Updating strategy showing instability in both L_f and L_g, resulting in poor optimization and degraded model performance.
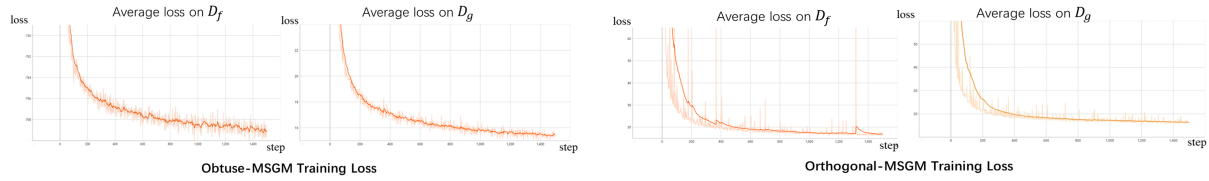

Figure 2. Loss curves for Alternative Updating strategy (updating L_f every four steps), demonstrating stable optimization and effective learning for both $\mathcal{D}_f$ and $\mathcal{D}_g$.

Figure 3. Additional visualization results on three datasets using MSGM for unconditional generation
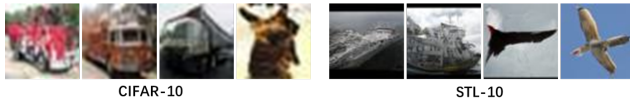

Figure 4. Failure cases of unlearning in unconditional generation. On CIFAR-10, $\mathcal{D}_f$ includes 'dog' and 'automobile', where high visual similarity (e.g., 'dog' vs. 'cat' or 'automobile' vs. 'car') leads to incomplete unlearning. On STL-10, $\mathcal{D}_f$ includes 'airplane', causing anomalies in 'bird' and 'ship' generation.

4. Visual Results and Failure Analysis

Our approach demonstrates effectiveness, though we observe limited failure cases, as shown in Fig. 4. These rare instances highlight challenges in disentangling overlapping features and provide insights for future refinement. Additional visual results in Fig. 5 and Fig. 6 further validate the robustness of our method in generating data while unlearning undesirable content. Experiments across diverse tasks confirm the flexibility and effectiveness of our approach, showcasing its broad applicability.
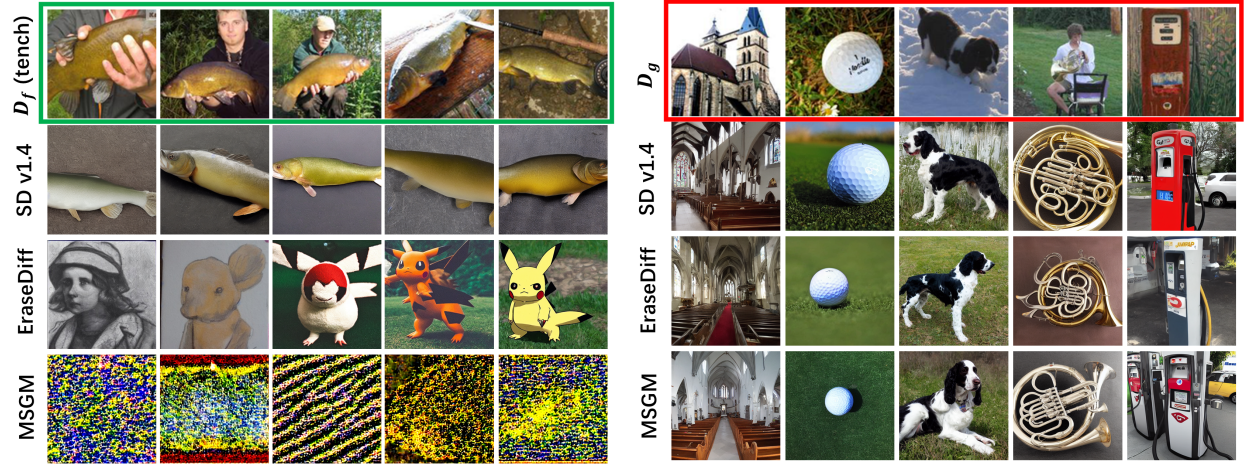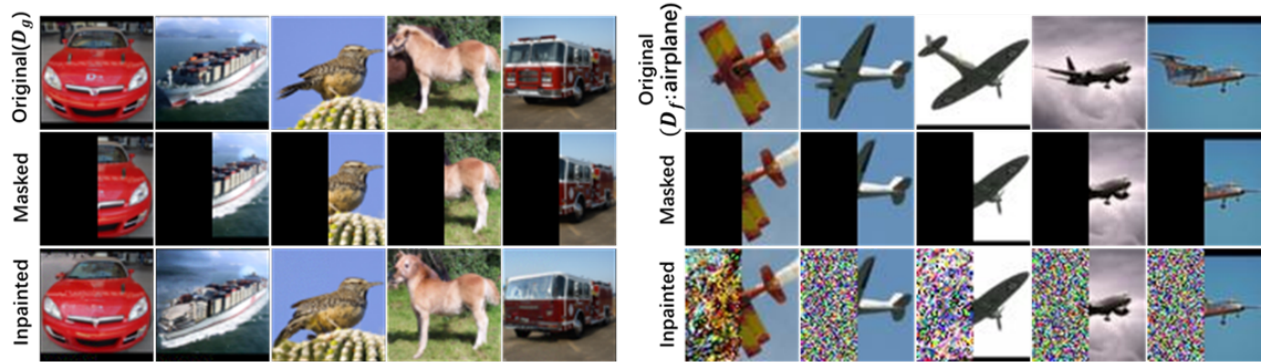
Figure 5. Additional visualization results comparing different unlearning methods on the Imagenette dataset for test-to-image generation task.

(a) Inpainting on STL-10

(b) Reconstruction on CIFAR-10

Figure 6. Additional results of inpainting and reconstruction, highlighting the model's success in unlearning target categories while maintaining image quality.