Appendix

Table of Contents

A UnlearnCanvas Dataset Details

In this section, we provide a detailed description of the dataset following 'datasheet for datasets' [72].

A.1 Motivation

This dataset is collected to enable a precise, comprehensive, and automated quantitative evaluation framework for MU (machine unlearning) methods in DMs (diffusion models). The current evaluation plans used in the existing literature have exposed several weaknesses, which may lead to incomplete, inaccurate or even biased results, see a more detailed discussion in Sec. 2. To the best of our knowledge, there are no datasets specifically designed to meet the assessing requirements of DM unlearning. Therefore, UnlearnCanvas is designed, collected, and made to fill in this gap.

A.2 Composition: Styles and Object Classes

There are 60 predetermined artistic styles provided by Fotor [11]. The images in the same artistic style all share high stylistic consistency, which enables a high-precision style classifier to be trained on them. In Fig. A10, we list some examples of the images in each style to illustrate these styles. There are 20 distinct object classes in UnlearnCanvas. In Fig. A11, we list some examples of the images in each object to illustrate these classes.

A.3 Collection and Labeling Process

The construction of UnlearnCanvas involves two main steps: seed image collection and subsequent image stylization; see Fig. 2 for an illustration. For seed image collection, a set of high-resolution real-world photos are collected from Pexels [69], providing open-sourced photographs. There are 20 seed images collected for each of the 20 object classes; see Fig. A11. After collecting the seed images, we stylize each and every seed image into all 60 predetermined artistic styles; see Fig. A10 with Fotor [11]. After the stylization of all the images, the dataset is structured in a hierarchical manner, and each image is labeled with both its style and object classes. In order to support text-to-image training, each image is annotated with the prompt 'An image of *object* in *style* style.'.

A.4 Uses

UnlearnCanvas can be used to evaluate MU methods in different unlearning scenarios. Please see Sec. 4 for more details. In addition, we stress that UnlearnCanvas can be used for more real-world tasks than unlearning, and we provide an example of how UnlearnCanvas can be used to systematically evaluate another generative task, style transfer, in Appx. F. We provide very detailed instructions with codes in the GitHub code repository.

A.5 Distribution

UnlearnCanvas is an open-sourced dataset and is based on the existing open-sourced data [69]. We make access to the dataset public under the MIT license. We remark that no personally identifiable information or offensive content is included in this dataset. The dataset can be accessed either through Google Drive and HuggingFace. More resources on the dataset, such as the introduction video and the benchmark, can be found in the official *project webpage*.

A.6 Maintenance

The dataset will be maintained by the lead author Yihua Zhang. If needed, the email for contacting is zhan1908@msu.edu. The dataset may be updated if needed (with the inclusion of more seed images, more artistic styles, and more objects). The updates will be ad-hoc and will not be periodical. Each time the dataset is updated, the updates will be reflected in the same GitHub code repository.

A.7 Author Statements

The collector and the lead author of this dataset, Yihua Zhang, bears full responsibility for any violation of rights that may arise from the collection of the data included in this research.

B Reproducibility Statement and Detailed Experiment Settings

In this section, we provide detailed instructions on the reproduction of our results in Sec. 4, including the settings of training, the implementation details of the tested machine unlearning methods, and the evaluation details in each unlearning scenario.

B.1 Finetuning Style and Object Classifier with UNLEARNCANVAS

Style and object classifiers need to be trained as part of the testbed proposed in our evaluation pipeline (Fig. 4). Here, we adopted a ViT-L/16 model [68] pretrained on ImageNet and finetune it on UNLEARNCANVAS. UNLEARNCANVAS are split into the train set and test set with a ratio of $9:1$. After hyper-parameter tuning, the classifiers are trained with Adam optimizer at a learning rate of 0.01 for 10 epochs.

B.2 Finetuning Stable Diffusion with UNLEARNCANVAS

The other part of the testbed is a diffusion model capable of generating high quality images in all the styles associated with all the objects encompassed in UNLEARNCANVAS in order to guarantee a trustworthy and unbiased evaluation.

Training settings. Practically, we finetune the pretrained Stable Diffusion (SD) v1.5 on UNLEARNCANVAS for 20k steps with a learning rate of $1e-6$. Unless otherwise stated, we strictly follow the training configurations used in Stable Diffusion [1]. For each image in UNLEARNCANVAS, we annotate the data with text prompt `An image of {`*object*`} in {`*style*`}`, where the *object* and *style* are the corresponding object and style label. For seed images, the *style* label we use is 'photo' style. We use the training scripts provided by Diffuser official tutorial (`https://huggingface.co/docs/diffusers/v0.13.0/en/training/text2image`) and the pretraining model card is `runwayml/stable-diffusion-v1-5`. During training, the checkpoints will be saved every 1000 steps.

Evaluation. To evaluate the quality of the saved checkpoints and selcect the best one for unlearning study, the checkpoints are first used to generate an image set with the same prompt as training (`An image of {`*object*`} in {`*style*`}`) by traversing all the possible style and object labels. Each prompt are used to generate 5 images with different random seed. Each image are sampled with 100 steps with a guidance coeeficient of 9. The image set for each checkpoint are fed into the style and object classifier trained in Appx. B.1. The model with the highest average performance on all the styles and objects are selected as the testbed for MU study. The classification performance are also used as a reference for later IRA/CRA comparison, which are disclosed in the first row of Fig. 5 (left).

Computing resource. In this work, we employ $40\times$ NVIDIA RTX A6000 GPUs to conduct all the model training, unlearning, image generation, and evaluations. When we finetuned the StableDiffusion on UNLEARNCANVAS, $8\times$ GPUs were used for parallel computing. Other experiments were all carried out in a single-GPU environment. Around 60,000 GPU hours in total were spent to complete all the experiments.

B.3 Implementation of DM Unlearning Methods Studied in This Work

In this work, we inspected a series of stateful MU methods for DMs. For each method, we use their publicly released source codes as code bases, which are listed below:

- ESD [23]: https://github.com/rohitgandikota/erasing
- CA [25]: https://github.com/nupurkmr9/concept-ablation
- UCE [24]: https://github.com/rohitgandikota/unified-concept-editing
- FMN [28]: https://github.com/SHI-Labs/Forget-Me-Not
- SalUn: [27]: https://github.com/OPTML-Group/Unlearn-Saliency
- SEOT: [30]: https://github.com/sen-mao/SuppressEOT
- SPM: [26]: https://github.com/Con6924/SPM
- EDiff: [31]: https://github.com/JingWu321/EraseDiff
- SHS: [32]: https://github.com/JingWu321/Scissorhands

In particular, we adopt the following training settings to adapt the methods to our dataset:

- ESD: Based on the suggestions from the authors, ESD is used to only finetune the cross attention-related model weights (ESD-x). Other settings strict follow the ones used in the paper.
- CA: In order to ablate concepts using CA, we first use ChatGPT to generate a list of simple prompts for each concept (including the styles and the objects), namely anchor prompts. Each anchor prompt is a simple one-sentence description of the unlearning target.
- UCE: This method requires a guided concept (prompt) for each unlearning concept. For style unlearning, we use the prompt `An image in {style*}` as the guided concept, where *style** represents the next style in UnlearnCanvas in alphabetical order. Similarly, `An image of object*` is used for object unlearning, where *object** is the next object in UnlearnCanvas in alphabetical order.
- FMN: This method requires the images associated with the unlearning target. For simplicity and best performance, we randomly select 20 images associated with the unlearning concept. For the first stage of FMN, we run text inversion for 500 steps with a learning rate of $1e-4$, and for the second step, we used the inversed text to unlearn the cross attention layers of the model for 100 steps. The hyper-parameters of learning rate, maximum steps, and tunable parameters (cross-attention or non-cross-attention) are carefully tuned with grid search.
- SalUn: This method involves two steps, the mask finding (weight saliency analysis) and the model unlearning. For mask finding, we tuned the mask ratio, while for unlearning, we tune the hyper-parameter learning rate and unlearning intensity. All the parameters are tuned with grid search. For both steps, the mask or model is trained with 10 epochs.
- SEOT: To generate unlearned images, we use the prompt `An {object*} image in {style*}`. We then suppress either `object*` or `style*` individually. Other settings strict follow the ones used in the paper.
- SPM: Following the hyperparameters provided by the authors, we trained and obtained Pre-tuned SPMs for all `object*` and `style*`. During image generation, we combine the pre-tuned SPMs with the DM. By calculating the association between words in the prompt and the target word, we determine whether to allow the specified word to preserve, and generate the corresponding image.
- EDiff: Based on the authors' suggestions, EDiff is used to finetune only the cross-attention-related model weights (EraseDiff-x). During the unlearning process, we adjusted the hyperparameters, specifically the learning rate and the number of unlearning epochs. The model is trained with 5 epochs.
- SHS: SHS consists of two stages: trimming and repairing. During the trimming stage, certain weights are re-initialized. The repairing stage then restores the model's utility. Throughout the unlearning process, we finetuned the hyperparameters, focusing on the learning rate and the number of unlearning epochs. Ultimately, we selected 2 epochs as the optimal number.

B.4 Evaluation Details of the Adversarial Prompt Generation for Unlearning Robustness

In Sec. 4, we evaluated the robustness of different MU methods against adversarial prompts. Here, we use the state-of-the-art method, UnlearnDiffAtk [59] to generate adversarial prompts. We set the prepended prompt perturbations by $N = 5$ tokens for both style and object unlearning. Following the original attack setting in UnlearnDiffAtk [59], to optimize the adversarial perturbations, we sample 50 diffusion time steps and perform PGD running for 40 iterations with a learning rate of 0.01 at each step. Prior to projection onto the discrete text space, we utilize the AdamW optimizer.

B.5 Experiment Details of the Style-Object Combination Unlearning

Unlearning targets. In Sec. 4, we evaluate the capability of different MU methods on performing unlearning at a finer scale, and we use the style-object combinations as unlearning targets for evaluation. Ideally, the UNLEARNCANVAS dataset can generate 1200 (60×20) style-object combinations. In this work, we randomly select 50 of these combinations for evaluation and we list these combinations below. For each method, the same hyper-parameters are used for each MU method as the ones for style and object unlearning in Tab. 2. The unlearning targets include:

- 'An image of Architectures in Abstractionism style.'
- 'An image of Bears in Artist Sketch style.'
- 'An image of Birds in Blossom Season style.'
- 'An image of Butterfly in Bricks style.'
- 'An image of Cats in Byzantine style.'
- 'An image of Dogs in Cartoon style.'
- 'An image of Fishes in Cold Warm style.'
- 'An image of Flame in Color Fantasy style.'
- 'An image of Flowers in Comic Etch style.'
- 'An image of Frogs in Crayon style.'
- 'An image of Horses in Cubism style.'
- 'An image of Human in Dadaism style.'
- 'An image of Jellyfish in Dapple style.'
- 'An image of Rabbits in Defoliation style.'
- 'An image of Sandwiches in Early Autumn style.'
- 'An image of Sea in Expressionism style.'
- 'An image of Statues in Fauvism style.'
- 'An image of Towers in French style.'
- 'An image of Trees in Glowing Sunset style.'
- 'An image of Waterfalls in Gorgeous Love style.'
- 'An image of Architectures in Greenfield style.'
- 'An image of Bears in Impressionism style.'
- 'An image of Birds in Ink Art style.'
- 'An image of Butterfly in Joy style.'
- 'An image of Cats in Liquid Dreams style.'
- 'An image of Dogs in Magic Cube style.'
- 'An image of Fishes in Meta Physics style.'
- 'An image of Flame in Meteor Shower style.'

- 'An image of Flowers in Monet style.'
- 'An image of Frogs in Mosaic style.'
- 'An image of Horses in Neon Lines style.'
- 'An image of Human in On Fire style.'
- 'An image of Jellyfish in Pastel style.'
- 'An image of Rabbits in Pencil Drawing style.'
- 'An image of Sandwiches in Picasso style.'
- 'An image of Sea in Pop Art style.'
- 'An image of Statues in Red Blue Ink style.'
- 'An image of Towers in Rust style.'
- 'An image of Waterfalls in Sketch style.'
- 'An image of Architectures in Sponge Dabbed style.'
- 'An image of Bears in Structuralism style.'
- 'An image of Birds in Superstring style.'
- 'An image of Butterfly in Surrealism style.'
- 'An image of Cats in Ukiyoe style.'
- 'An image of Dogs in Van Gogh style.'
- 'An image of Fishes in Vibrant Flow style.'
- 'An image of Flame in Warm Love style.'
- 'An image of Flowers in Warm Smear style.'
- 'An image of Frogs in Watercolor style.'
- 'An image of Horses in Winter style.'

Evaluation. The evaluation of the style-object combination unlearning concerns four quantitative metrics, one for unlearning effectiveness and three for retainability. Before the evaluation, an answer set will be generated exactly following the same procedure introduced in Sec. 3 and Fig. 4 after unlearning each target. First, the UA (unlearning accuracy) will be evaluated for each answer set, which stands for the ratio of images generated by the target prompt that are neither classified into the target object nor the target style class. A high UA denotes a better ability to successfully unlearn the target combination. Second, the retainability of generation associated with those prompts close to the unlearning target prompt will be evaluated. These prompts can be divided into two groups, the ones sharing the same style but not the object class and the ones sharing the object but not the style class. The classification accuracy of the former corresponds to the retainability of the style, *i.e.,* style consistency (SC), while the latter one denotes the object consistency (OC). These two quantitative metrics evaluate how well the unlearning method precisely define the unlearning scope and retain the generation ability of those close but innocent concept. Thirdly, the retainability of the rest unrelated prompts (UP) are evaluated, which is the last quantitative evaluation metric. The results reported in Tab. 3 are averaged over all the unlearning cases shown above.

B.6 Experiment Details of Sequential Unlearning

In Sec. 4, we also evaluated the MU methods with the task of sequential unlearning (SU), where the efficacy of MU methods in handling multiple sequential unlearning requests $\{\mathcal{T}_i\}$ are evaluated. This requires models not only to unlearn new targets effectively but also to maintain the unlearning of previous targets, while retaining all other knowledge. In the experiments, 6 styles are randomly selected as the unlearning targets and excluded from the RA evaluation. The UA of all the already unlearned target will be assessed each time a new request is accomplished. The selected 6 styles include:

- Abstractionism
- Byzantine
- Cartoon
- Cold Warm
- Ukiyoe
- Van Gogh

After each unlearning request, the unlearning effectiveness and retainability are evaluated. Specifically, the unlearning accuracy of all the unlearning targets in the previous requests are evaluated to evaluate how the unlearning effect lasts when new unlearning requests arrive. In the meantime, the retainability of all the other concepts that are not selected as unlearning targets are evaluated, and to ease the presentation, the retain accuracy of all the concepts (styles and objects) are averaged and reported.

B.7 Metrics Summary in All Unlearning Settings

Besides the UNLEARNCANVAS dataset, the various quantitative evaluation metrics proposed in this work are part of the major contributions to a comprehensive and precise evaluation for DM unlearning methods. As there are various unlearning scenarios studied in this work, we provide a summary of these metrics in Tab. A1, including their abbreviations, descriptions, and related tables or figures.

Table A1: A summary of the quantitative metrics used in this work, including their abbreviations, meanings and where they are used.

Metrics	**Description**	**Usages (Table & Figure)**
	Style/Object Unlearning	
UA	Unlearning accuracy	Fig. 1, Tab. 1
IRA	In-domain unlearning accuracy	Fig. 1, Tab. 1
CRA	Cross-domain unlearning accuracy	Fig. 1, Tab. 1
	Unlearning Robustness against Adversarial Prompts	
Rob.	Unlearning robustness, unlearning accuracy in the presence of adversarial prompts	Fig. 1
	Style-Object Combination Finer-Scale Unlearning	
FU/UA	Unlearning accuracy in finer-scale unlearning	Fig. 1, Tab. 3
SC	Retainability evaluation of style consistency	Tab. 3
OC	Retainability evaluation of object consistency	Tab. 3
UP	Retainability evaluation of unrelated prompts	Tab. 3
FR	Retainability evaluation in finer-scale unlearning, averaged by SC, OC, and UP	Fig. 1
	Sequential or Continual Unlearning	
SU or CU	Unlearning accuracy in the context of sequential unlearning	Fig. 1, Tab. A5
SR or CR	Retainability in the context of sequential unlearning	Fig. 1, Tab. A5

C A Detailed Comparison between UNLEARNCANVAS and WIKIART

Figure A1: Image examples with the same style label from WIKIART [66] and UNLEARNCANVAS. Images of the same artistic style in UNLEARNCANVAS exhibit high stylistic consistency compared to WIKIART.

UNLEARNCANVAS vs. WIKIART. To the best of our knowledge, WIKIART [73] is the most relevant baseline dataset to ours. In **Tab.** A2, we provide a direct comparison of the key attributes of these two datasets. UNLEARNCANVAS differs from WIKIART in the following aspects.

First, UNLEARNCANVAS includes a greater number of high-resolution images (15M) compared to WIKIART (2M), a factor that may enhance the training of state-of-the-art DMs.

Table A2: Comparison with WIKIART, the most relevant dataset containing stylized images to ours. UNLEARNCANVAS stands out notably from WIKIART due to its characteristics of being supervised, balanced, and maintaining high stylistic cohesiveness.

Dataset	**Resolution (Pixels/Image)**	**Style-wise Supervised**	**High Stylistic Consistency**	**Class-wise Balanced**
WIKIART [73]	$\sim 2M$	✗	✗	Style-wise ✗ Object-wise ✗
UNLEARNCANVAS	$\sim 15M$	✓	✓	Style-wise ✓ Object-wise ✓

Second, UNLEARNCANVAS surpasses WIKIART in terms of both intra-style coherence and inter-style distinctiveness, as illustrated in **Fig.** A1, where images labeled with 'Van Gogh Style' from both datasets are compared. In UNLEARNCANVAS, the images exhibit high stylistic consistency, while WIKIART lacks the necessary clarity for precise assessment. This will hamper the MU evaluation as discussed in the challenges **(C2)** and **(C3)**. This benefits can also be reflected by the training performance using UNLEARNCANVAS and WIKIART. The results are reported in Tab. A3 and Tab. A4, respectively. As we can see, the classifier is much more easily trained on UNLEARNCANVAS, justifying the higher discernible features within each style in UNLEARNCANVAS.

Table A3: Art style reproduction quality using SD v1.5 and SD v2.0 finetuned on WIKIART. Images are generated with the prompt "A painting in *artist* style", where *artist* refers to those included in WIKIART. The test accuracy on DM-generated images and original WIKIART test images is reported using the style classifier finetuned from the pretrained ViT-L/16 on WIKIART.

Image Source	Images by SD v1.5	Images by SD v2.0	WIKIART Test Set
Accuracy	41.2%	56.7%	85.4%

Third, the images in UNLEARNCANVAS are style-wise supervised. For each seed image, a stylized counterpart can be find in each style class. This is beneficial for tasks other than unlearning for text-to-image task, such as image editing, image stylization, and style transfer, which can provide a ground truth image for precise and robust evaluation. This will be detailed in Appx. F.

Table A4: Style classification results of a ViT-Large [68] as a style classifier trained on UnlearnCanvas. After convergence, the classifier is tested on the test set and the image set generated by SD v1.5 finetuned on UnlearnCanvas.

	UnlearnCanvas Train Set	UnlearnCanvas Test Set	Images by SD v1.5 tuned on UnlearnCanvas
Accuracy	100.0%	99.9%	98.8%

D Additional Experiment Results for Unlearning Evaluation

D.1 Visualization of the Style and Object Unlearning Performance

To make a more direct comparison among different MU methods reported in Tab. 1, the results are visualized in the radar chart Fig. A2. This figure illustrates that no method dominates across all assessment dimensions. This underscores the complexity of unlearning in generative models and the need for further improvement.

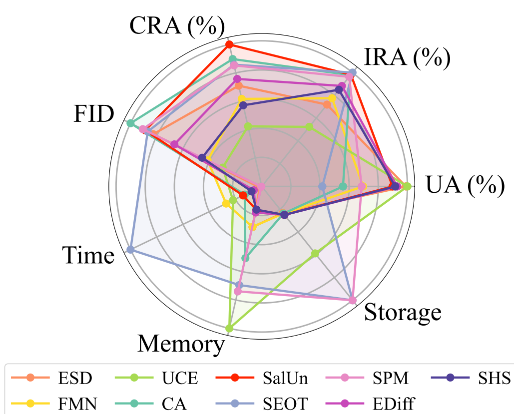


Figure A2: Performance visualization for various unlearning methods as summarized in Table 2. For UA, IRA, and CRA, the results are averaged over the style and object unlearning scenarios. Other metrics undertake the inverse operation as a smaller values represent better performance. Results are normalized to $0\% \sim 100\%$ per metric.

D.2 A Fine-Grained Comparison per Unlearning Target: ESD *vs.* SalUn

Following the analysis of ESD in Fig. 5, we next turn our focus to a comparative analysis with SalUn, a method that demonstrated a better balance between unlearning and retaining according to Tab. 2. A similar accuracy heatmap for SalUn is presented in Fig. A3. Compared to ESD, SalUn exhibits more consistent performance across various unlearning scenarios, as indicated by the more uniform color distribution in the heatmap. This also suggests enhanced retainability. However, it is noticeable that SalUn does not reach the same level of UA (Unlearning Accuracy) as ESD, as evidenced by the darker diagonal values in Fig. A3. This observation reinforces the existence of a trade-off between unlearning effectiveness and retainability in the visual generative MU task, a phenomenon paralleled in other tasks such as classification.

D.3 Unlearning Heatmaps of More Unlearning Methods

In Fig. A4$\sim$Fig. A8, we provide more unlearning heatmap visualizations in the same format as Fig. 5 and Fig. A3 in order to provide a more detailed unlearning performance dissection for all the DM unlearning methods studied in this work.

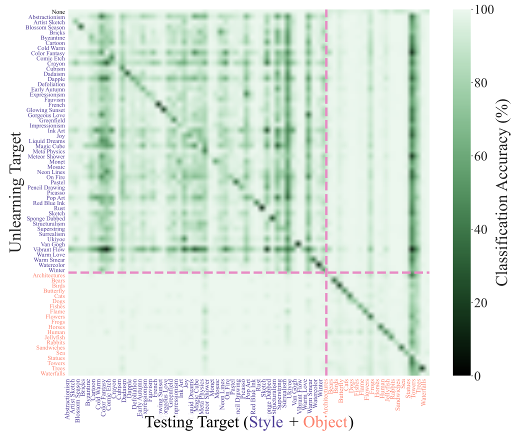


Figure A3: Heatmap visualization of SalUn. The plot setting is identical to Figure 5.

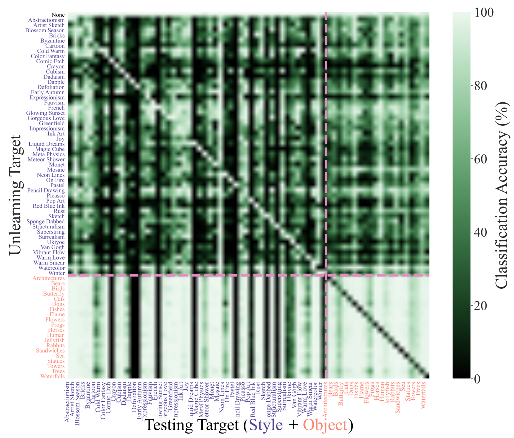


Figure A4: Heatmap visualization of FMN. The plot setting is identical to Figure 5.

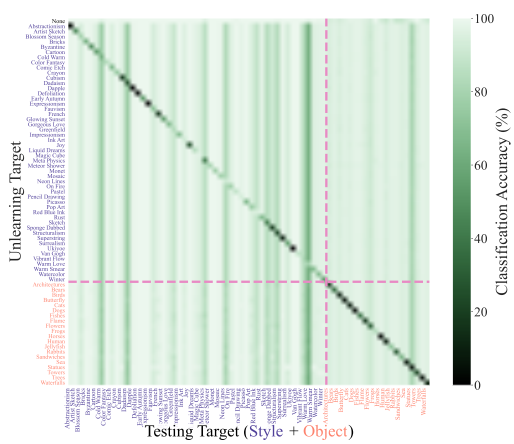


Figure A5: Heatmap visualization of SEOT. The plot setting is identical to Figure 5.

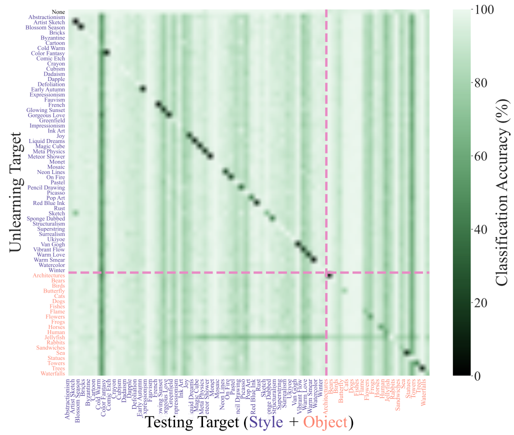


Figure A6: Heatmap visualization of SPM. The plot setting is identical to Figure 5.

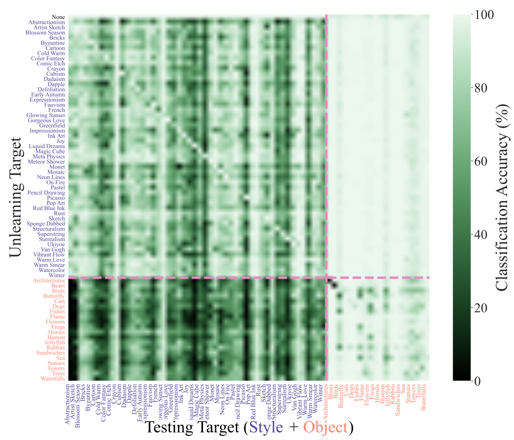


Figure A7: Heatmap visualization of Ediff. The plot setting is identical to Figure 5.

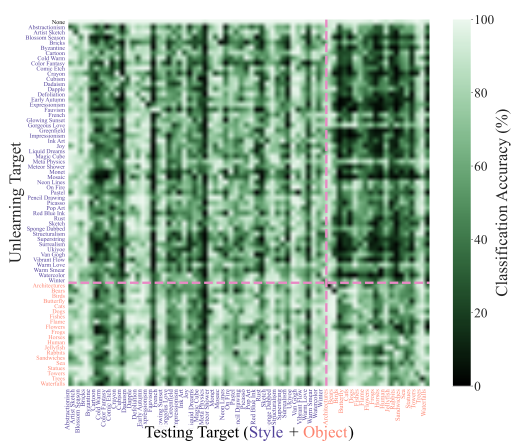
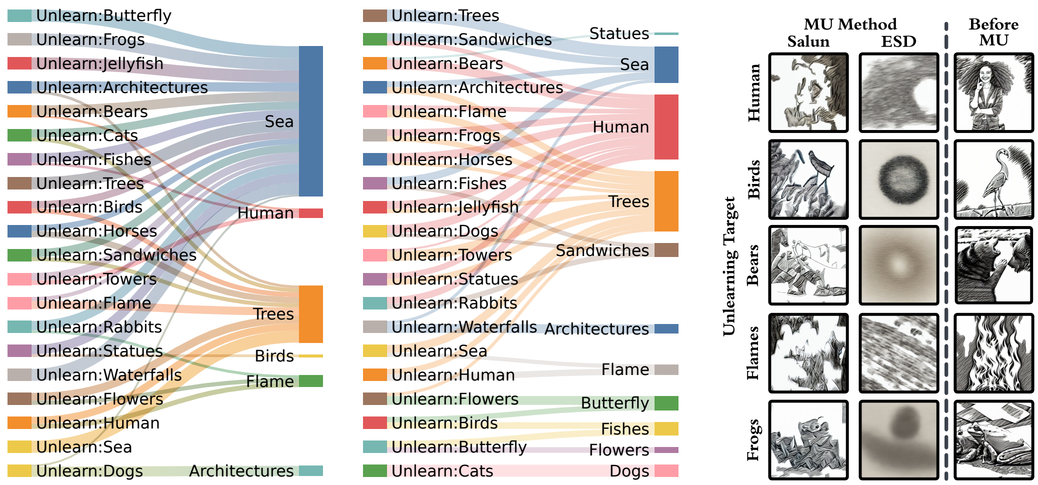


Figure A8: Heatmap visualization of SHS. The plot setting is identical to Figure 5.



(a) ESD (b) SalUn (c) Illustration

Figure A9: Visualization of the unlearning directions of (a) ESD and (b) SalUn. This figure illustrates the conceptual shift of the generated images of an unlearned model conditioned on the unlearning target. Images generated by the post-unlearning models are classified and used to understand this shift. Edges leading from the object in the left column to the right signify that images generated conditioned on unlearning targets are instead classified as the shifted concepts after unlearning. This reveals the primary unlearning direction for each unlearning method. The most dominant unlearning direction for an object is visualized. Figure (c) provides visualizations of generated images using the prompt template 'A painting of {*object*} in Sketch style.' with *object* being each unlearning target.

D.4 Understanding Unlearning Method's Behavior via Unlearning Directions

As noted earlier, different unlearning methods display distinct unlearning behaviors. To gain insights into the underlying reasons for these differences, **Fig. A9** (a) and (b) visualize the 'unlearning directions' for ESD and SalUn, respectively. These unlearning directions are determined by connecting the unlearning target with the predicted label of the generated image from the unlearned DM conditioned on the unlearning target. As shown in Fig. A9 (a), ESD demonstrates a focused shift in image generation after object unlearning, with a predominant transition towards generating images labeled by 'Sea' and 'Trees'. This behavior arises from ESD's optimization process, designed to steer the generation of the DM away from a predefined concept. Consequently, images generated by the ESD-induced unlearned model consistently lack clearly identifiable objects, resembling waves and trees, which leads to their classification into the 'Sea' and 'Trees' classes; see Fig. A9 (c) for examples of generated images. In contrast, SalUn exhibits a more diverse range of unlearning directions, shifting images to 11 different objects. This diversity results from SalUn's requirement to replace the unlearning target with a random concept. As shown in Fig. A9 (c), images generated by SalUn post-object unlearning still maintain some object contours (different from the original unlearning target) and better retain style information compared to ESD.

D.5 MU Methods Evaluation in Sequential Unlearning

In this experiment, we evaluated the MU methods with the task of sequential unlearning (SU), where the efficacy of MU methods in handling multiple sequential unlearning requests are evaluated. More detailed experiment settings are shown in Appx. B.6. Here, we consider unlearning 6 styles sequentially and the results are presented in **Tab. A5**. We remark that the method SEOT does not support sequential unlearning in its original implementation and thus is not included in Tab. A5.

Our findings reveal significant insights. (1) Degraded retainability: Sequential unlearning requests generally degrade retainability across all methods, with RA values frequently dropping below the average levels previously seen in Tab. 2. Here RA is given by the average of IRA and CRA. (2) Unlearning rebound effect: Knowledge previously unlearned can be inadvertently reactivated by new unlearning requests. This is evidenced by decreasing UA values for earlier objectives as more

Table A5: Performance comparison of different DM unlearning methods in the sequential unlearning setting. Each column represents a new unlearning request, denoted by $\mathcal{T}_i$, where $\mathcal{T}_1$ is the oldest. Each row represents the UA for a specific unlearning objective or the retaining accuracy (RA), given by the average of IRA and CRA. Results indicating *unlearning rebound* effect are highlighted in orange, and those signifying *catastrophic retaining failure* are marked in red.

Metrics		$\mathcal{T}_1$	$\mathcal{T}_1 \sim \mathcal{T}_2$	$\mathcal{T}_1 \sim \mathcal{T}_3$	$\mathcal{T}_1 \sim \mathcal{T}_4$	$\mathcal{T}_1 \sim \mathcal{T}_5$	$\mathcal{T}_1 \sim \mathcal{T}_6$
Method: ESD							
UA	$\mathcal{T}_1$	100%	99%	95%	87%	81%	75%
	$\mathcal{T}_2$	-	100%	100%	96%	87%	79%
	$\mathcal{T}_3$	-	-	100%	98%	99%	98%
	$\mathcal{T}_4$	-	-	-	100%	100%	99%
	$\mathcal{T}_5$	-	-	-	-	100%	99%
	$\mathcal{T}_6$	-	-	-	-	-	100%
RA		77.46%	52.94%	35.99%	24.86%	18.69%	12.95%
Method: FMN							
UA	$\mathcal{T}_1$	88%	99%	99%	98%	99%	99%
	$\mathcal{T}_2$	-	95%	99%	99%	98%	99%
	$\mathcal{T}_3$	-	-	97%	98%	99%	99%
	$\mathcal{T}_4$	-	-	-	99%	99%	99%
	$\mathcal{T}_5$	-	-	-	-	99%	99%
	$\mathcal{T}_6$	-	-	-	-	-	100%
RA		82.39%	14.56%	13.34%	10.42%	9.83%	8.76%
Method: UCE							
UA	$\mathcal{T}_1$	93%	95%	98%	96%	97%	98%
	$\mathcal{T}_2$	-	97%	98%	98%	98%	95%
	$\mathcal{T}_3$	-	-	95%	97%	98%	99%
	$\mathcal{T}_4$	-	-	-	98%	98%	98%
	$\mathcal{T}_5$	-	-	-	-	97%	99%
	$\mathcal{T}_6$	-	-	-	-	-	99%
RA		81.42%	29.38%	18.72%	15.34%	13.32%	11.31%
Method: CA							
UA	$\mathcal{T}_1$	58%	55%	59%	45%	44%	40%
	$\mathcal{T}_2$	-	76%	58%	51%	47%	44%
	$\mathcal{T}_3$	-	-	45%	41%	40%	37%
	$\mathcal{T}_4$	-	-	-	71%	70%	60%
	$\mathcal{T}_5$	-	-	-	-	69%	51%
	$\mathcal{T}_6$	-	-	-	-	-	57%
RA		97.24%	93.39%	84.46%	79.32%	71.40%	60.53%
Method: SalUn							
UA	$\mathcal{T}_1$	84%	79%	78%	65%	67%	64%
	$\mathcal{T}_2$	-	81.42%	75%	72%	69%	61%
	$\mathcal{T}_3$	-	-	90%	85%	84%	87%
	$\mathcal{T}_4$	-	-	-	84%	86%	81%
	$\mathcal{T}_5$	-	-	-	-	79%	81%
	$\mathcal{T}_6$	-	-	-	-	-	89%
RA		85.43%	80.32%	71.42%	65.41%	63.24%	60.19%
Method: SPM							
UA	$\mathcal{T}_1$	55%	59%	50%	49%	47%	48%
	$\mathcal{T}_2$	-	62%	59%	58%	60%	63%
	$\mathcal{T}_3$	-	-	42%	39%	40%	41%
	$\mathcal{T}_4$	-	-	-	57%	59%	60%
	$\mathcal{T}_5$	-	-	-	-	51%	51%
	$\mathcal{T}_6$	-	-	-	-	-	43%
RA		72.39%	70.42%	67.89%	60.45%	55.32%	51.12%
Method: EDiff							
UA	$\mathcal{T}_1$	97%	93%	91%	93%	85%	90%
	$\mathcal{T}_2$	-	92%	89%	93%	91%	87%
	$\mathcal{T}_3$	-	-	96%	93%	90%	84%
	$\mathcal{T}_4$	-	-	-	91%	92%	90.22%
	$\mathcal{T}_5$	-	-	-	-	99%	97%
	$\mathcal{T}_6$	-	-	-	-	-	94%
RA		92.34%	89.37%	14.35%	12.31%	12.82%	7.42%
Method: SHS							
UA	$\mathcal{T}_1$	81%	73%	74%	93%	94%	97%
	$\mathcal{T}_2$	-	69%	61%	89%	94%	97%
	$\mathcal{T}_3$	-	-	75%	92%	96%	90%
	$\mathcal{T}_4$	-	-	-	91%	95%	97%
	$\mathcal{T}_5$	-	-	-	-	92%	96%
	$\mathcal{T}_6$	-	-	-	-	-	94%
RA		88.41%	84.32%	73.98%	69.19%	10.76%	10.11%

unlearning tasks are introduced, a trend highlighted in orange. This suggests that residual knowledge remains within the model and can be reactivated, aligning with findings from Fig. 6. This indicates the unlearned models by some MU methods do not essentially lose the generation ability of the unlearning target. (3) Catastrophic retaining failure: RA significantly drops at a certain request, exemplified by a sudden decrease in RA of UCE from 81.42% to 29.38% after the second request, $\mathcal{T}_2$. This indicates that the seemingly acceptable side effects generated by some unlearning methods will drastically modify the knowledge representations when accumulated. This experiment illuminates the complex dynamics of knowledge removal and retention within DMs and highlights the potential pitfalls of existing unlearning methods when faced with sequential unlearning tasks. The observation of the 'unlearning rebound effect' and 'catastrophic retaining failure' particularly emphasizes the need for a more nuanced understanding of how knowledge is managed within DMs.

E Visualizations

In this section, we aim to provide plenty of visualizations, illustrations, and qualitative results of the dataset and the quantitative results shown in Sec. 4 and Appx. D. These visualizations are intended to deepen the understanding of the effects of different MU methods and clearly illustrate the challenges identified in previous sections. We hope these visual aids will enable readers to more effectively grasp the nuances of MU methods and their implications for DMs (Diffusion Models).

E.1 Illustrations of the Styles and Objects in UnlearnCanvas

We first provide an illustration of the styles and object classes included in UnlearnCanvas. Specifically, we show the styles in Fig. A10 and object classes in Fig. A11 with the style and object names disclosed in the captions.

Figure A10: An illustration of the images in each style in UnlearnCanvas used in, which are stylized from the same seed image from the 'Dogs' object class. The seed image is presented in Fig. A11 (6). Images are cropped and down-scaled for illustration purpose. The name of the styles are: (1) Abstractionism; (2) Artist Sketch; (3) Blossom Season; (4) Blue Blooming; (5) Bricks; (6) Byzantine; (7) Cartoon; (8) Cold Warm; (9) Color Fantasy; (10) Comic Etch; (11) Crayon; (12) Crypto Punks; (13) Cubism; (14) Dadaism; (15) Dapple; (16) Defoliation; (17) Dreamweave; (18) Early Autumn; (19) Expressionism; (20) Fauvism; (21) Foliage Patchwork; (22) French; (23) Glowing Sunset; (24) Gorgeous Love; (25) Greenfield; (26) Impasto; (27) Impressionism; (28) Ink Art; (29) Joy; (30) Liquid Dreams; (31) Palette Knife; (32) Magic Cube; (33) Meta Physics; (34) Meteor Shower; (35) Monet; (36) Mosaic; (37) Neon Lines; (38) On Fire; (39) Pastel; (40) Pencil Drawing; (41) Picasso; (42) Pointillism; (43) Pop Art; (44) Rainwash; (45) Realistic Watercolor; (46) Red Blue Ink; (47) Rust; (48) Sketch; (49) Sponge Dabbed; (50) Structuralism; (51) Superstring; (52) Surrealism; (53) Techno; (54) Ukiyoe; (55) Van Gogh; (56) Vibrant Flow; (57) Warm Love; (58) Warm Smear; (59) Watercolor; (60) Winter.

Figure A11: An illustration of the seed images in each object class in UNLEARNCANVAS. Images are cropped and down-scaled for illustration purpose. The name of the object classes are: (1) Architecture; (2) Bear; (3) Bird; (4) Butterfly; (5) Cat; (6) Dog; (7) Fish; (8) Flame; (9) Flowers; (10) Frog; (11) Horse; (12) Human; (13) Jellyfish; (14) Rabbits; (15) Sandwich; (16) Sea; (17) Statue; (18) Tower; (19) Tree; (20) Waterfalls.

E.2 Visualization of Style Unlearning

In Fig. A12, we provide abundant generation examples of all the 9 methods benchmarked in this work in a case study of unlearning the ‘Cartoon’ style. Both the successful and failure cases are demonstrated in the context of unlearning effectiveness, in-domain retainability, and cross-domain retainability.

E.3 Visualization of the Unlearning Performance in the Presence of Adversarial Prompts

In Fig. A13, we provide visualizations for the effect of adversarial prompts. As revealed in Fig. 6, all the DM unlearning methods experience a significant drop in unlearning effectiveness when attacked by the adversarial prompt, enabled by UnlearnDiffAtk [59]. We provide the image generation in four unlearning cases (two for style unlearning and two for object unlearning), and show the images of the unlearning target successfully generated in the presence of adversarial prompts.

Figure A12: visualization of the unlearning performance of different methods on the task of style unlearning. Three text prompt templates are used to evaluate the unlearning effectiveness, in-domain retainability, and cross-domain retainability of each method. Images with green frame denote desirable results, while the ones with red frame denote unlearning or retaining failures.

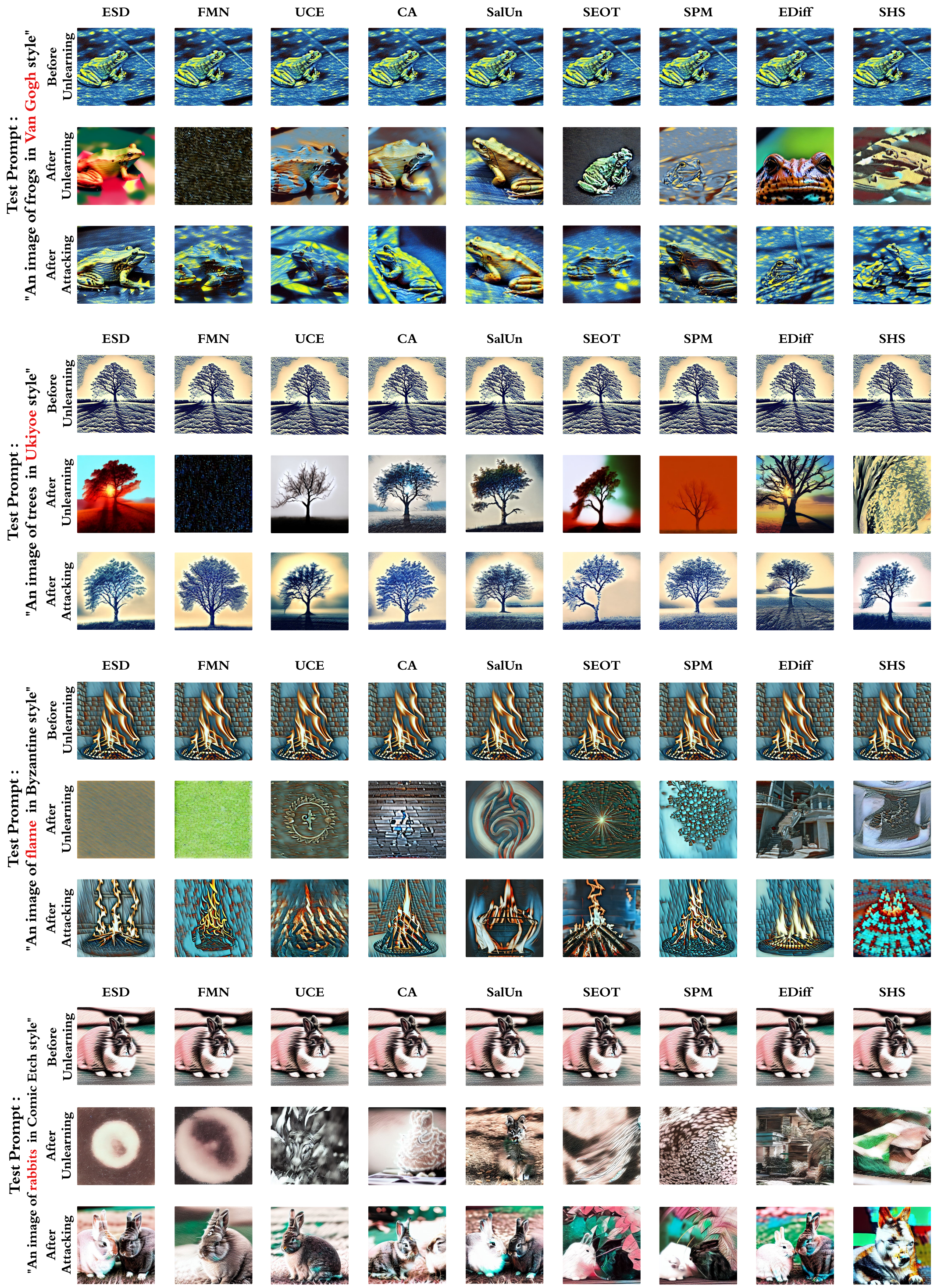


Figure A13: Visualization of the images generated by the unlearned DMs using different unlearning methods in the absence or presence of adversarial prompts.

F Broader Use Cases of UNLEARNCANVAS

Although UNLEARNCANVAS is originally designed for benchmarking MU methods, we would like to demonstrate its broader use cases of benchmarking more generative modeling tasks, thanks to its good properties discussed in Sec. 3. In this section, we start with a case study on the task of style transfer, which is a much more well-studied topic than MU, but surprisingly also faces great challenges in building up a comprehensive and precise evaluation framework. In the next, we will first dissect the key challenges of the current style transfer evaluation framework, and then demonstrate how UNLEARNCANVAS efficiently resolves these challenges and further proposes a comprehensive and automated benchmark. Through extensive experiments, we draw demonstrate new insights from these results and illuminate the challenges of the future research directions. In the end, we will discuss the possibility of using UNLEARNCANVAS to benchmark more generative modeling tasks.

F.1 Benchmarking Style Transfer using UNLEARNCANVAS

Style transfer. Style transfer is a long-standing topic and focuses on transferring the artistic style from one **s**tyle image (also known as the *reference* image) $\mathbf{x}_s$ to a target **c**ontent image $\mathbf{x}_c$. The most recent methods typically employ a neural network, denoted by $\boldsymbol{\theta}_s$, to extract style features and perform stylization in a single inference step, expressed as $\hat{\mathbf{x}}_o = f_{\boldsymbol{\theta}_s}(\mathbf{x}_s, \mathbf{x}_c)$. Current state-of-the-art (SOTA) style transfer techniques exhibit remarkable generalization capabilities, successfully transferring styles not encountered during training and not requiring any further back-propagations. Figure 1 illustrates the pipeline of this task. Most existing literature [74–79] utilize WIKIART [66] to provide different styles for training the stylization network $\boldsymbol{\theta}_s$. During the evaluation, the validation set of WIKIART will serve as the style (reference) images, together with the content images from the COCO dataset [80], to form a test bed for style transfer and style learning methods. However, such a evaluation scheme has some inherent limitations, which will be detailed below.

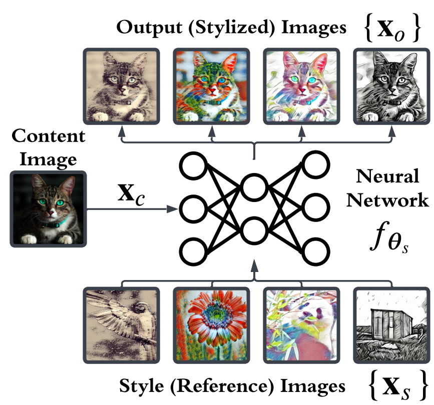
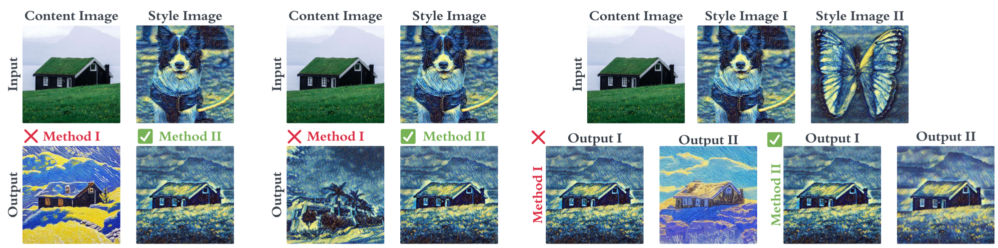


Figure A14: An illustration of the task of style transfer.

Issues and challenges in the evaluating methods for style transfer Although the task of style transfer has been widely studied, its evaluations are still based on very limited quantitative or even sorely qualitative assessments, potentially leading to incomplete and inaccurate assessments [81]. Upon examining the evaluation pipelines of over 10 state-of-the-art (SOTA) style transfer methods, three significant challenges are identified within the current widely accepted evaluation frameworks. • **Challenge I (C1): The lack of the ground truth images for style similarity evaluation.** Unlike other vision tasks like classification [82], detection [83], and segmentation [84], one of the key shortcomings of the current style transfer evaluation lies in the lack of the ground truth images $\mathbf{x}_g$ for the given reference style image $\mathbf{x}_s$ and the content image $\mathbf{x}_c$. Consequently, existing evaluation metrics, such as the style loss ℓ_{style} [85, 86], has to be calculated with the reference style image $\mathbf{x}_s$ as the ground truth $\mathbf{x}_g$, namely $\ell_{\text{style}}(\mathbf{x}_s, \hat{\mathbf{x}}_o)$, rather than directly using the ground truth $\ell_{\text{style}}(\mathbf{x}_o, \mathbf{x}_g)$. Obviously, such a indirect evaluation may lead to inaccurate results due to the different contents held in $\mathbf{x}_s$ and $\mathbf{x}_o$. Existing work has demonstrated that such evaluation metrics can lead to very different conclusion from that of the user study [79]. Therefore, the creation of a *supervised* dataset with ground truth stylized images for every content image under each style is a timely remedy. • **Challenge II (C2): The lack of algorithm stability evaluation against varied reference images $\mathbf{x}_s$.** The evaluation of algorithm stability in style transfer has often been neglected. This assessment requires consistent performance of a method on a content image $\mathbf{x}_c$ across different style reference images $\mathbf{x}_s$ representing the same target artistic style. Ideally, an algorithm should maintain uniform quality across various references within a style, avoiding significant performance variations. Current challenges in such evaluations stem from the lack of reference sets that exhibit *high stylistic consistency* within each style. Figure A1 showcases examples from the widely-used WIKIART dataset. Despite sharing the same artistic label, images within a row show considerable divergence in visual appearance and style. Consequently, using these images as style references can lead to stable algorithms producing stylistically varied outputs, leading to misleading assessment results. Therefore, creating a dataset with high stylistic uniformity within each style category and clear differentiation between styles is crucial for accurate measurement of algorithm stability.



(a) Style Authenticity (b) Content Integrity (c) Algorithm Stability

Figure A15: An illustration of comprehensive performance metrics for style transfer tasks. (a) Style authenticity measures the style similarity between the generated image and the style image. (b) Content integrity measures the content preservation between the generated image and the content image. (c) Algorithm stability reflects the sensitivity of the algorithm to different style images. For all the metrics in the illustration, "Method II" is always better than "Method I".

Building up a comprehensive evaluation pipeline with UNLEARNCANVAS for style transfer. To revolutionize the evaluation framework with UNLEARNCANVAS, it is crucial to first understand the components that form a comprehensive evaluation pipeline, ensuring a thorough and impartial assessment of performance. In the realm of style transfer, three pivotal metrics stand paramount, as depicted in **Figure,A15**:

❶ Style Authenticity: Assesses the extent to which the style of the generated image aligns with that of the provided reference style image(s). ❷ Content Integrity: Measures how well the content features are preserved post style transfer. ❸ Algorithm Stability: Evaluates the algorithm's robustness against variations in content subjects, target styles, or style reference image selections, while keeping other parameters constant.

In addressing the challenges (**C1**-**C2**), UNLEARNCANVAS proves to be inherently advantageous. **First**, the style-specific supervision embedded in UNLEARNCANVAS enables the provision of ground truth for quantitative evaluations of stylized images. **Second**, the extensive image collection within the same style category in UNLEARNCANVAS allows for the assessment of algorithm stability by applying style transfer methods to varied reference images. To encompass the aforementioned aspects of style transfer performance, we propose the following quantitative evaluation metrics:

① Style Loss: Utilizes a feature map-based style loss [87] to quantify the stylistic *dissimilarity* between image pairs, effectively representing the inverse of style authenticity. ② Content Loss: Employs a VGG-based, feature-map content loss [87, 88] to measure the visual dissimilarity between the reference and generated images, essentially mirroring content integrity. ③ Averaged Standard Deviation (STD): Computes the average STD of style and content loss *w.r.t.* the same reference image, reflecting algorithm stability.

Furthermore, similar to the MU task, we also consider efficiency metrics for each method, including:

④ Average Time Consumption: Measures the time required for performing style transfer. ⑤ Peak GPU Memory Consumption: Records the maximum GPU memory usage during the style transfer process. ⑥ Model Storage Memory Consumption: Assesses the memory requirement for storing the style transfer model.

With the introduction of evaluation metrics (①-⑥), we establish a comprehensive evaluation pipeline for style transfer. The process is as follows:

For each style transfer method, style transfer is executed within each object class. Specifically, for every style, images indexed from 1 to 18 serve as style reference images, while seed images corresponding to indices 19 and 20 are used as content images. Style and content loss are computed for each pair of style reference and content images, with the stylized images derived from the used seed content images serving as the ground truth for style loss. This results in a total of $60 \times 20 \times 18 \times 2$ experimental trials for each method. For a specific style and content image pair, 18 experiments are performed to calculate the Standard Deviation (STD) values for both style and content loss. These STD values are then averaged over all content images (amounting to $60 \times 20 \times 2$ cases). The findings from these comprehensive evaluations are presented in Table A6.

Following this evaluation pipeline, we scrutinized 9 prominent style transfer methods, including SANET [78], MCC [89], MAST [90], ARTFLOW with its two variants (AF-ADAIN and AF-WCT) [79], IE-CONTRAST [91], CAST [88], STYTR2 [87], and BLIP [92].

Table A6: Performance overview of different style transfer methods evaluated with UNLEARNCANVAS dataset. The performance are assessed from the perspectives of stylistic authenticity (style loss), content integrity (content loss), algorithm stability (standard deviations from different dimensions), and efficiency. For all the metrics, *smaller* values are always preferred for better performance. The best performance per each metric is highlighted in **bold**. The standard deviations are first calculated with respect to different styles, object classes, or tested content images and then averaged in order to depict the algorithm stability from different perspectives.

Method	Style Loss				Content Loss				Efficiency		
	Mean	STD (Averaged over)			Mean	STD (Averaged over)			Time (s/image)	Memory (GB)	Storage (GB)
		Style	Object	Content Image		Style	Object	Content Image			
SANET	23.48	**2.73**	**2.87**	1.87	0.85	0.12	0.17	0.09	0.29	**2.3**	0.11
MCC	**17.92**	4.59	4.82	2.14	0.96	0.14	0.21	0.07	0.38	5.4	0.10
MAST	24.10	2.87	3.16	1.74	1.42	0.33	0.34	0.18	2.86	4.8	0.16
AF-ADAIN	20.78	2.96	3.13	1.65	1.09	0.11	0.19	0.05	0.53	6.3	0.08
AF-WCT	20.22	2.94	3.19	1.75	1.02	0.12	0.18	0.05	0.53	6.3	**0.08**
IE-CONTRAST	21.27	3.01	3.32	2.05	1.08	0.29	0.31	0.15	**0.05**	3.8	0.11
CAST	24.01	2.78	2.90	**1.35**	1.38	0.32	0.40	0.16	0.32	6.7	0.19
STYTR2	19.75	3.04	3.30	1.91	**0.62**	**0.10**	**0.12**	**0.04**	0.58	3.9	0.21
BLIP	25.43	2.90	3.06	2.03	1.61	0.30	0.34	0.16	8.87	7.2	7.23

Experiment results analysis. Tab. A6 provides a systematic evaluation of the performance of various methods tested. From the analysis, we can derive several crucial insights:

First, it is evident that no single method excels across all evaluation metrics. Notably, MCC demonstrates superior performance in maintaining stylistic authenticity, as indicated by a low style loss. Conversely, STYTR2 stands out in preserving content integrity, reflected by its minimal content loss.

Second, the assessment of standard deviation is indispensable for a comprehensive evaluation. The method with the optimal performance does not necessarily exhibit the greatest stability. This is particularly apparent in the style loss evaluation, where MCC, despite achieving the best result in terms of style loss, exhibits the least stability, denoted by the highest standard deviation.

F.2 Other Possible Applications of UNLEARNCANVAS

In the preceding section, we demonstrated the application of UNLEARNCANVAS in refining evaluation metrics and frameworks for style transfer. Beyond this, we recognize the potential of UNLEARNCANVAS in diverse domains. Here, we delve into two illustrative examples:

Bias mitigation. Bias mitigation in DMs, which are now gaining popularity, can also benefit from UNLEARNCANVAS. Its hierarchical and balanced architecture enables the deliberate introduction of artificial biases by selectively omitting data from specific groups. For instance, by predominantly excluding images from styles other than the 'Van Gogh Style' within the 'Dogs' class, DMs finetuned on this dataset will inherently exhibit a tendency to generate images of dogs in the Van Gogh style, particularly when the style is not explicitly specified in the prompt. This approach not only allows for the manipulation and quantification of biases but also paves the way for UnlearnCanvas to become a standardized benchmark for bias mitigation, similar to the role of MU for DMs.

Vision in-context learning (V-ICL). V-ICL [93–96] is another domain where UnlearnCanvas can be effectively applied. The field of V-ICL is in urgent need of robust, comprehensive methods for the fair assessment of existing models. In this context, the image pairs from UnlearnCanvas are ideally suited for evaluating various tasks such as style transfer, image inpainting, and image segmentation, offering a rich resource for nuanced and quantitative analyses.

G Impact Statement

This work helps improve the assessment and further promotes the advancement of MU (machine unlearning) methods for DMs (diffusion models), which are known to be effective in relieving or mitigating the various negative societal influences brought by the prevalent usage of DMs, which include but are not limited to the following aspects.

• **Avoiding Copyright Issues.** There is an urgent need for the generative model providers to scrub the influence of certain data on an already-trained model. In January 2023, a notable lawsuit targeted two leading AI art generators, Stable Diffusion [1] and Midjourney [2], for alleged copyright infringement. Concurrently, incidents with the recently released Midjourney V6 [2] also highlighted a visual plagiarism issue on famous film scenes. These instances illuminate the broad copyright challenges inherent in the way of training data collection method of those foundation generative models' training datasets. MU methods can be used as an effective method to remove the influence of the private data and avoid unnecessary retraining.

• **Mitigating biases and stereotypes.** Generative AI systems are known to have tendencies towards bias, stereotypes, and reductionism, when it comes to gender, race and national identities [17]. For example, a recent study on the images generated with Midjourney revealed, that images associated with higher-paying job titles featured people with lighter skin tones, and that results for most professional roles were male-dominated [16]. MU is known to be effective in eliminating biases rooted in the training data. Moreover, UnlearnCanvas offers a flexible framework to benchmark MU techniques against bias removal, allowing for the creation and quantitative control of biases across different object classes for comprehensive bias removal studies.