Forget Me, Not My Friends! Object Unlearning Based on Scene Graphs

Chenhan Zhang*
University of Technology Sydney
School of Computer Science
Sydney, NSW, Australia
chzhang@ieee.org

Benjamin Zhao
Macquarie University
School of Computing
Sydney, NSW, Australia
ben_zi.zhao@mq.edu.au

Hassan Asghar
Macquarie University
School of Computing
Sydney, NSW, Australia
hassan.asghar@mq.edu.au

Weiqi Wang
University of Technology Sydney
School of Computer Science
Sydney, NSW, Australia
weiqi.wang@ieee.org

An Liu
Soochow University
School of Computer Science and Technology
Suzhou, Jiangsu, China
anliu@suda.edu.cn

Dali Kaafar
Macquarie University
School of Computing
Sydney, NSW, Australia
dali.kaafar@mq.edu.au

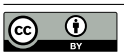
Abstract

Machine unlearning offers a practical technical means for fulfilling users' requests to remove personally identifiable information (PII) under "right to be forgotten" regulations such as GDPR and COPPA. Traditionally, unlearning is performed with the removal of entire data samples (sample unlearning) or whole features across the dataset (feature unlearning). However, when the removal request targets only certain parts of the PII, such as specific objects within a sample, these traditional unlearning approaches fall short of meeting such finer-grained unlearning requirements. To address this gap, we propose a scene graph-based object unlearning framework. This framework utilizes scene graphs, rich in semantic representation, transparently translate unlearning requests into actionable steps. The result, is the preservation of the overall semantic integrity of the generated image, bar the unlearned object. Furthermore, we develop three distinct approaches for object unlearning, grounded in the mainstream unlearning techniques of fine-tuning and model redaction. For validation, we evaluate the unlearned object's fidelity in outputs under the tasks of image reconstruction and image synthesis. Our proposed framework demonstrates improved object unlearning outcomes, with the preservation of unrequested samples in contrast to sample and feature learning methods. This work addresses critical privacy issues by increasing the granularity of targeted machine unlearning through forgetting specific object-level details without sacrificing the utility of the whole data sample or dataset feature.

CCS Concepts

• **Security and privacy → Privacy protections**.

*Work done at Macquarie University.



This work is licensed under a Creative Commons Attribution 4.0 International License.
WSDM '26, Boise, ID, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2292-9/2026/02
https://doi.org/10.1145/3773966.3777964

Keywords

Machine Unlearning, Scene Graph, Data Privacy

ACM Reference Format:
Chenhan Zhang, Benjamin Zhao, Hassan Asghar, Weiqi Wang, An Liu, and Dali Kaafar. 2026. Forget Me, Not My Friends! Object Unlearning Based on Scene Graphs. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining (WSDM '26), February 22–26, 2026, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3773966.3777964

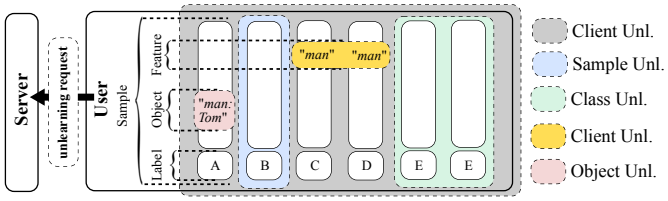
1 Introduction

As machine learning models become increasingly integral to a range of personalized applications, from facial recognition to bespoke content generation, the protection of user privacy and the ability to comply with data removal requests have become paramount. The rise of Machine Learning as a Service (MLaaS) platforms has only intensified this need, as such platforms operate with large-scale, diverse datasets containing personal information. With the implementation of stringent data privacy regulations, e.g., COPPA [9] and GDPR [12], and recently e-Privacy [31] and CCPA [2], users have the legal right to request the deletion of their personal data from these models. One way to entertain such requests is through *machine unlearning*, a process where specific learned information is removed from a model without necessitating complete model retraining from scratch [3, 13, 28, 35, 45]. Generally, these approaches either focus on removing entire samples or specific features across the training space. We argue that in many applications, these methods of machine unlearning are rather coarse-grained, and end up removing more information than necessary, thus adversely impacting the utility of the unlearned model.

We illustrate our point through an example. Consider MLaaS for image generation or reconstruction. Privacy conscious users may request the removal of their personal data from these models. More specifically, the user wants the removal of his/her *face* from any set of images used to train the model. Under existing unlearning methods, the service provider has two efficient approaches available to handle such a request. In the first instance, called *sample unlearning* [1, 3], the service provider can remove all samples containing the user's face from the model. While this is good for images only

Table 1: Five different types of machine unlearning. An illustration is given in the right part.

Unlearning Type	Unl. Request q_{unl}	Unl. Granularity
Client Unlearning [14, 23]	$\mathcal{D}_{\text{user}_i}$	■■■■
Class Unlearning [4, 5]	$\forall I \in \Delta\mathcal{Y}$	■■■
Sample Unlearning [1, 13]	$\forall I \in \Delta\mathcal{D}$	■■
Feature Unlearning [20, 40]	$\forall o \in \Delta C$	■■
Object Unlearning	$o \in \Delta\mathcal{O}$	■



containing the user's face, many images might be more complex containing other rich information such as cars or mountains in the background. These objects which may have no bearing on the user's privacy, yet still valuable to the model, would be removed as collateral damage.

Alternatively, in the second approach, called *feature unlearning* [11, 27], the service provider can opt to erase all facial features from the model. However, this approach risks unintentionally removing facial data of other users who made no requests for erasure of their information, thereby diminishing the model's ability to accurately generate representations of faces of other users.

We can observe that both methods are *coarse-grained* may unlearn much more than what is requested by the user, adversely affecting the model's overall performance. In this work, we develop machine unlearning techniques that work in a more granular level, in the sense that they unlearn parts of a sample while retaining utility both in terms of retaining information about other parts of the sample, and the impact on other samples with similar content. This application is most natural for images, which may contain multiple objects, only a subset of whom are requested to be removed [16]. Another example is text-based data where only certain parts of the document are to be redacted [40].

We call this selective unlearning approach, *object unlearning*, where object specifies individual entities within a sample, e.g., a physical object in an image or an entity in text. This unlearning approach is akin to *targeted therapy* in medicine, where specific malignant cells are removed without damaging healthy tissue [17]. A significant challenge of object unlearning is that objects do not exist in isolation; Instead there exists interwoven relationships between objects and their surrounding context that collectively contribute to the semantic coherence of the data sample [19]. Unlearning a specific object while preserving the rest of the sample's content requires caution to ensure the removal of one element does not unintentionally create inconsistencies or degrade the model's understanding in the remaining structure or within other samples. As captured by the need to preserve the model's performance and generalization capabilities. We contextualize the granularity of object unlearning in Table 1.

To navigate these relational challenges, our approach for object unlearning leverages *scene graphs*. Scene graphs provide a structured representation of an image by capturing objects, their attributes, and the relationships between them [18]. This representation not only offers a high-level semantic understanding of visual content but also facilitates more nuanced and contextualized interpretations of scenes. Many studies have focused on generating scene graphs from images, and vice versa [22]. By leveraging the structure and semantics inherent in scene graphs, we can more precisely target objects for more effective, fine-grained unlearning techniques.

Our contributions are summarized as follows:

- This paper is the first to investigate the unlearning request of specific objects by MLaaS users. We identify the gap of fine-grained machine unlearning, one which allows the removal of specifically requested learned information while minimizing the impact on the model's overall utility. We formally propose the concept of object unlearning, which unlearns specific objects from an image.
- To resolve the major challenge in object unlearning of disentangling interwoven objects, the assurance of one element's removal not unintentionally degrading the model in the remaining sample, we propose a scene graph-based object unlearning framework. Scene graphs provide a direct and transparent means to translate unlearning requests into execution.
- We comprehensively evaluate unlearning techniques developed in isolation for either sample or feature unlearning by adapting said techniques for all unlearning granularities. These techniques include influence functions, negative guidance, and masking techniques of patching and noise addition. The source code and artifacts of our proposed scene graph-based unlearning is released at https://anonymous.4open.science/r/soul-24C8/.

2 Positioning within Prior Work

This work is situated at the intersection of machine unlearning and scene graph-based image generation, with a particular focus on fine-grained object removal. Prior work in machine unlearning has primarily focused on coarse-grained approaches, such as removing entire samples or features, and has largely targeted discriminative tasks on structured data (e.g., graph or tabular domains). Although recent studies have extended unlearning to generative settings, their methods often rely on erasing global features or entire object classes, failing to support the selective forgetting of individual object instances.

In contrast, our study addresses this gap by introducing object unlearning, a finer-grained approach that targets specific object instances without affecting the utility of remaining data. Compared to graph unlearning methods [7, 8, 29, 37, 41, 42], which focus on removing substructures from graph-based models, we reinterpret unlearning through the lens of visual scene graphs to facilitate selective object removal from generative image models. Unlike prior feature unlearning work in generative models [11, 15, 20, 27, 40, 44, 47], which typically removes global attributes or requires carefully crafted datasets, our method leverages structured scene representations to automate and contextualize object selection for unlearning. Furthermore, existing image manipulation

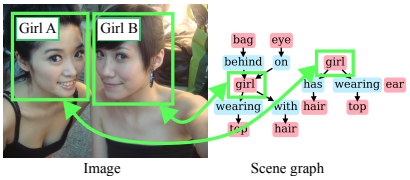


Figure 1: The unique identity of objects in the VSG.

techniques [25, 33, 38, 39, 43] are designed for pre-release obfuscation and cannot be applied post hoc once models have been trained. Our method, by contrast, is designed for post-training unlearning, offering a practical response to regulatory mandates such as GDPR's right to erasure. Finally, while scene graph-based image generation has been widely studied [10, 26, 36, 48], it has not been explored in the context of unlearning. We fill this gap by developing a scene graph-based framework that enables structured and interpretable object removal.

3 Preliminaries

Image Generation from Scene Graphs. We consider a generative learning task where the model is trained to generate images from structured input. More formally, image generation is modeled as a mapping function $f_\theta : \mathbf{x} \to I$, where $\mathbf{x}$ denotes the input (e.g., a scene graph, text prompt, or latent code), I is the generated image, and θ represents the parameters of the generative model.

Particularly, our study specifically leverage the technique of image generation from scene graphs. A scene graph is a structured representation that encodes the contents of a scene by capturing objects, their attributes, and the relationships among them. In this work, we utilize visual scene graphs (VSGs) [6] as an intermediate representation for object unlearning in images. Each VSG G is defined as a tuple $G = (O, \mathcal{E})$, where $O = \{o_1, \ldots, o_m\}$ is a set of objects (nodes), $\mathcal{R}$ a set of relationship types, and $\mathcal{E} \subseteq O \times \mathcal{R} \times O$ is a set of edges of the form $(o_i, \mathbf{r}_{ij}, o_j)$ where $o_i, o_j \in O$. Each object o_i can be expressed as $o_i = (\mathbf{c}_i, \mathbf{a}_i)$, where $\mathbf{c}_i \in C$ is the category of the object, typically determined by $\mathbf{a}_i \subseteq \mathcal{A}$ which represents the attributes of the object.

From the perspective of object, each object is *unique* in identity; that is, given two objects o_i and o_j, even if $\mathbf{c}_i = \mathbf{c}_j$ and $\mathbf{a}_i = \mathbf{a}_j$, it still holds that $o_i \neq o_j$ (consider a pair of twins). We use the notation $o \in I$ to say that the object o is contained in the image I. The notation $\text{cat}(o)$ means the category of object o. Define the sets $O = \{o : o \in I \text{ for some } I \in \mathcal{D}\}$ and $C = \{c : c = \text{cat}(o) \text{ for some } o \in O\}$, as the set of objects and categories in the training dataset, respectively. Figure 1 illustrates a VSG, observe how the graph describes the hierarchical relationship between different objects in the image, not only is the "bag" object present in the image, the graph capture's it's position behind the "girl" on the right. As each graph node has it's own features, the two "girl" nodes are similar, but distinctly unique.

Machine Unlearning. We consider a MLaaS provider with complete control over their model, including its training data and white-box information. This trained model is accessible via an API for end users. When compelled to remove a specific sensitive items from the model, and associated training data, the requester specifies the data/objects to be removed. For example, data contributors to a social media platform are automatically opted in for machine learning training of a friend photo tagging system, however a privacy conscious user can request the platform to unlearn their specific data from trained models, allowing the affected user to opt out.

Machine unlearning refers to the process of removing the influence of specific data points from a trained machine learning model. Let $\mathcal{D}$ be the full training dataset. We use the notation f_{θ^*} to denote the *original model* trained on $\mathcal{D}$. Let $\Delta\mathcal{D}$ be the data to be removed from $\mathcal{D}$, and $\mathcal{D}\backslash\Delta\mathcal{D}$ be the remaining data after removing $\Delta\mathcal{D}$. $\Delta\mathcal{D}$ is usually reflected in the unlearning request $\mathbf{q}_{\text{unl}}$ of the user, which will be sent to the service provider for an unlearning execution $f_{\theta^-} = \mathcal{U}(f_{\theta^*}, \Delta\mathcal{D})$ where $\mathcal{U}$ is an unlearning algorithm and f_{θ^-} denotes the unlearned model. Based on the nature of unlearning requests, there are four types of prevailing machine unlearning techniques: (1) sample unlearning [1, 13], (2) feature unlearning [20, 40], (3) class unlearning [4, 5], and (4) client unlearning [14, 23]. The unlearning techniques differences are summarized in Table 1, together with our proposed object unlearning technique.

It is important to note that these five types of unlearning techniques have varying scopes depending on the scenario. For instance, class unlearning is more likely to occur in discriminative tasks rather than generative tasks. Additionally, client unlearning is more relevant in distributed systems, such as federated learning. In this work, we focus primarily on generative tasks; therefore, our investigation centers on sample unlearning and feature unlearning. Consequently, we compare our proposed object unlearning method with these approaches.

4 Object Unlearning

We now formulate object unlearning and detail an overall framework for scene graph-based object unlearning.

Definitions. Recall the real-world setting of image-based machine learning tasks, specific information from images may have their removal requested, without requirements for the whole image to be removed. In this scenario, the limitations of sample and feature unlearning approaches are clear, much more is removed than what is required. Motivating our proposal for *object unlearning*.

Definition 4.1 (Object Unlearning). Object unlearning is defined as the unlearning technique where a specific object is removed from a generative model. Given the set of requested objects $\mathbf{q}_{\text{unl}}$ for unlearning, we unlearn:

$$\Delta O = \{o \in O : o \in \mathbf{q}_{\text{unl}}\}.$$

For instance, given a specific requested unlearning object, e.g. a boy named "Tom", only this specific object is selectively "unlearned" from the generative model. After unlearning, the model should not generate images containing the visual object identified as "Tom".

Object unlearning allows for the selective removal of a distinct and identifiable object (such as a specific individual or item) from a generative model. This level of granularity ensures that the model can unlearn highly specific visual or conceptual entities while retaining other related features or objects from the training sample. In other words, the model does not remove other objects in the same image.

Unlearning Verification Metrics. To assess whether object unlearning is successful in its task, we construct several metrics. For

generative models, successful unlearning is expected to achieve two quantitative objectives in the unlearning verification phase: *Effectiveness* and *Preservation* [24]. Under the context of our target for object unlearning, as discussed above, we formulate these two objectives as follows:

- **Unlearning Effectiveness:** The unlearned model (f_{θ^-}) should not generate the removed object (ΔO) in its generation. That is, for any input $\mathbf{x} = (I, G)$, which includes an image I and its scene graph G, we require that:
$$\text{For all } o \in \Delta O, o \notin f_{\theta^-}(\mathbf{x}) \tag{1}$$
- **Model Utility Preservation:** The unlearned model (f_{θ^-}) should maintain its performance on the retained objects in its generation. That is, for any input $\mathbf{x} = (I, G)$, we require that
$$\text{For all } o \in \mathbf{x} \text{ such that } o \in O \backslash \Delta O, o \in f_{\theta^-}(\mathbf{x}). \tag{2}$$

The objectives above are generalized to capture that the model achieves both Effectiveness and Preservation in the tasks of (1) *image reconstruction* and (2) *image synthesis*. These two image generation tasks are important for verification in different application settings. For image reconstruction, the focus is to provide an exact evaluation to ensure that the information related to the requested object has truly been removed, given that the input still includes strong associated visual features of the original object samples. In contrast, the image synthesis task is more relevant to realistic scenarios in generative AI applications. Here, the goal is to prevent objects containing personally identifiable information (PII), such as faces, from appearing in generations created by other users. While other scenarios may be constructed, we discuss them as future work in our discussion.

4.1 Scene Graph-Based Object Unlearning Framework

Object unlearning presents two significant technical challenges. The first is accurately identifying a distinct and recognizable object from the unlearning request, especially when similar or semantically close objects are present in the image. Specifically, the question of how we can reliably pinpoint the unique object in question. The second challenge involves the disentanglement of interconnected objects, ensuring that the removal of one element does not unintentionally diminish the model's understanding or introduce inconsistencies in the remaining structure. To address this, we propose a scene graph-based object unlearning framework.

Framework Overview: The proposed scene graph-based object unlearning framework contains integrations within both 1) *learning* and 2) *unlearning* phases, presented in Figure 2.

Learning Phase: During the learning phase, the MLaaS provider and the data provider (user) collaboratively train a scene graph-to-image (SG2I) generator [18]. As stated before in our assumptions, the service provider will generate corresponding scene graphs when new data samples are contributed. We assume that said scene graphs exist. For example, platforms such as Google Cloud can automate processes like object detection, scene understanding, and tagging without user intervention[1]. With both images and scene graphs available, the server will train the SG2I generators to learn how to map scene graphs to images. As SG2I stores the visual feature information of objects, it will later become the focus for object unlearning. After training, we obtain a trained model f_{θ^*} (the original model on which unlearning is to be applied), formulated for a given learning task, i.e., image reconstruction, we have $\theta^* = \arg_\theta \min \mathcal{L}_\mathcal{D}$ where $\mathcal{L}_\mathcal{D}$ represents the loss function defined on the reconstruction of each image $f_\theta(G_i, I_i)$ and the ground truth I_i.

Unlearning Phase: When an unlearning request $\mathbf{q}_{\text{unl}} = o \in \Delta O$ is made to the MLaaS provider to unlearn a specific object, the MLaaS provider will execute unlearning algorithm $\mathcal{U}$ to produce an unlearned model by $f_{\theta^-} = \mathcal{U}(f_{\theta^*}, \Delta O)$. The proposed scene graph-based framework precisely identifies the object of interest within complex visual data. Specifically, when constructing a scene graph, each object is assigned a unique *bounding box* specifying its positional information within the image. This is beneficial by allowing the retrieval of precise regions of interest (ROI) of the object during unlearning. Once these objects are clearly defined, we can effectively apply an unlearning algorithm to remove them.

4.2 Object Unlearning Approaches

Here, we redeploy three efficient approximate unlearning methods to serve as alternatives to the computationally intensive approach of retraining the model sans the unlearning object. Methods 1 and 2 employ fine-tuning techniques, while Method 3 leverages influence functions for model redaction, also known as model editing. In subsequent experiments, we evaluate the effectiveness and efficiency of these approaches in achieving targeted object unlearning.

Methodology 1: Negative Guidance-Based Fine-Tuning. We first propose a *negative guidance-based fine-tuning* method. In each training iteration, for the specific target object, we first locate its bounding box in the scene graph to extract the corresponding ROI of the target object. Then, a reconstruction loss is computed between the corresponding ROI areas of the generated image and the target image. We negate this loss and incorporate it into the total loss as a negative guidance term to facilitate unlearning. The negative guidance loss is defined as $\mathcal{L}_{\text{ng}} = -\lambda \cdot \sum_{I_i \in \mathcal{D}} l\left(I'_{i,o}, I_{i,o}\right)$. Here, $I'_{i,o}$ and $I_{i,o}$ denote the ROI of the generated and original images, respectively; λ is a weighting factor that balances the influence of negative guidance with the generative objective; and l represents the reconstruction loss, which calculates the pixel-wise difference.

This loss function leads the generator to gradually remove the feature representation of the object. Finally, we add the negative guidance loss to the total generator loss, which guides the generator and updates the parameters to unlearn the target object by $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{gen}} + \mathcal{L}_{\text{ng}}$, where $\mathcal{L}_{\text{gen}}$ represents the generator's original loss function. This process will weaken the generator's memory of the target object gradually, thereby realizing object unlearning.

Methodology 2: Mask-Based Fine-Tuning. A *mask-based fine-tuning* process involves two main steps: (1) mask the ROI associated with the requested object o, and (2) fine-tune the model with this masked input.

Let M_o be a mask that covers the region associated with the object o of the scene graph in the image I. Using the bounding box information provided by the scene graph, this mask can be easily localized and constructed. Further, we can obtain a modified input $\tilde{I} = I \circ M_o$, where $\circ$ here denotes element-wise masking, ensuring that only the region associated with o is influenced while

[1] https://cloud.google.com/blog/products/ai-machine-learning/label-your-photos-automagically-with-vision-api/

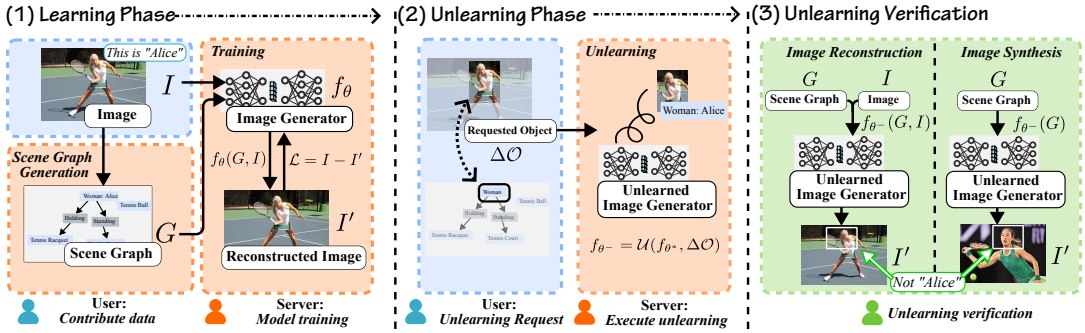


Figure 2: Scene graph-based object unlearning framework. Scene graphs can help both servers and users manage unlearning requests effectively by providing a structured way to understand the relationships between objects in an image. Scene graphs make it easier to identify and remove the requested object, such as a girl in an image. Moreover, this ensures that servers interpret and handle requests accurately, avoiding vague or incomplete unlearning actions. In this way, scene graphs act as a bridge, translating user intentions into actionable and transparent operations for the server.

preserving the remainder of the image. Particularly, we introduce two types of masks M_o for ROI as follows (we use x and y to denote the pixel coordinates below): (1) *Patch Masking*: Set pixel values in M_o to zero: $\tilde{I}_{x,y} = 0 \quad \forall\, x'_{\text{left}} \le x \le x'_{\text{right}},\ y'_{\text{top}} \le y \le y'_{\text{bottom}}$. (2) *Noise Masking*: Inject Gaussian or random noise $\mathcal{N}(0, \sigma^2)$ to the ROI covered by: M_o, $\tilde{I}_{x,y} = I_{x,y} + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2)$.

To unlearn the object o, the model f_θ is fine-tuned with the modified $\tilde{I}$. This process updates the model parameters from θ to θ^- by minimizing a loss function L that measures the model's output consistency with the original unmasked regions in I: $\theta^- = \arg\min_\theta \sum_{I_i \in \mathcal{D}} l\left(f_\theta\left(G_i, \tilde{I}_i\right), \tilde{I}_i\right)$, where $\tilde{I}_i$ represents each instance of a masked input. This design offers a straightforward solution to guide the model in retaining unmasked features. After fine-tuning, the adjusted model f_{θ^-} should avoid generating the object o in future outputs while maintaining other objects in the image.

Methodology 3: Influence Function-based Partial Model Redaction. *Influence functions* permit the approximation of the unlearning process, thereby achieving efficient unlearning. Specifically, our scene graph-based object unlearning can be reformulated as a graph unlearning problem [32]. As each object within the image directly corresponds to a node within the scene graph, we first formulate the object unlearning task as a node-level graph unlearning problem.

To solve this problem, we draw upon the off-the-shelf work of [41], which explores the use of influence functions for node-level graph unlearning. In [41], the authors provide a proven closed-form expression for the model parameter change, $\Delta\theta = \theta^- - \theta^*$, which is applicable to our scenario. The expression is given by $\Delta\theta \approx H_{\theta^*}^{-1} \nabla_{\theta^*} \mathcal{L}_{\Delta O}$, where H_{θ^*} is the Hessian matrix of the learning loss $\mathcal{L}_{\mathcal{D}}$ concerning θ^*. To properly account for the unlearning of specific objects $o \in \Delta O$, we leverage knowledge that a scene graph object corresponds to a node with its own attributes (e.g. label, identity, or location). Consequently, object unlearning can be framed as node feature unlearning within the broader graph unlearning landscape. Finally, the unlearned model can be estimated by model redaction: $\theta^- = \theta^* + \lambda \Delta\theta$, where λ is a scalar multiplier that adjusts the magnitude of the parameter change. We give details of this method in our open repository.

5 Experiments

5.1 Experimental Settings

Datasets. Visual Genome [21] is a large-scale dataset designed to advance research in image understanding, particularly in tasks like object detection, scene recognition, and relationship modeling. It contains 108,077 images annotated with approximately 21.3 million object instances, 10.8 million attributes, and 1.5 million relationships. Additionally, it provides 5.4 million region descriptions and 1.7 million question-answer pairs. This dataset is instrumental in tasks such as scene graph generation, visual question answering, and image captioning, making it a critical benchmark [22].

Pre-processing. To process the dataset, we develop a pipeline for both our training and unlearning processes.

For object processing, we construct vocabularies encompassing objects, attributes, and relationships. We standardize the naming conventions for objects and relationships using alias mappings to ensure consistency. Once the vocabularies are established, we filter the object annotations to retain only those that met specific size criterion and are included in the constructed vocabulary. Additionally, objects and attributes that appear frequently (above a predetermined threshold) were also incorporated into the vocabulary.

For image processing, consistency was ensured by removing images with dimensions below a specified minimum size, particularly those with extremely small objects. With realistic privacy-preserving scenarios in mind, we specifically select all samples containing salient personally identifiable information (PII). This was achieved by identifying and including all samples labeled with any of the following nine human-related object labels: ["man", "woman", "boy", "girl", "child", "person", "kid", "people", "face"].

We then encoded the objects, attributes, and relationships into a scene graph-based representation for each image. To ensure uniformity across all images, the data were padded to maintain a consistent structure, with each image containing a specified number of objects ($|O| = 10$) and relationships.

Model. In our experiment, we employ SIMSG [10] as the SG2I generator. SIMSG provides a general framework that has been widely adopted. This framework integrates VGG [34] as the visual feature extractor, a graph convolution-based heterogeneous GNN as GRL,

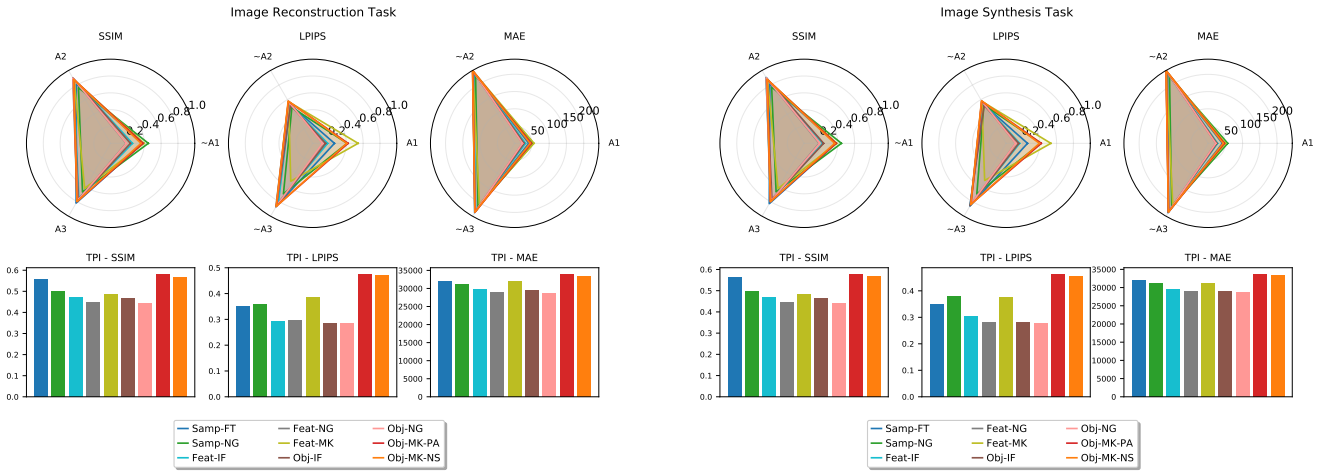


Figure 3: Comparison of overall unlearning performance across different methods on both the image reconstruction (left) and generation (right) tasks. Basically, A1 measures how well the unlearned object is removed, A2 evaluates preservation of unrelated objects in the same sample, and A3 assesses preservation of same-category objects in other samples; higher TPIs indicate better all-round performance.

and SPADE [30] as the image decoder. Due to computing resource limitations, the SG2I generator processes images at a resolution of 64×64 pixels for both input and output.

For training the SG2I model, we first pretrain the entire model on the whole training dataset. Following this, we fine-tune the model on samples containing the selected human-related object labels for 2000 epochs to ensure the original model possesses sufficient generation capability.

Metrics. In the verification of the unlearning framework, we use the metrics of MAE, SSIM, and LPIPS [46] to measure the quality of unlearning, metrics common among related works [27, 47].

Generally speaking, smaller MAE and LPIPS, or higher values of SSIM indicate better recovery of the generated images when compared against the ground truth. Further, as objects have different sizes between samples, we apply normalization to the metrics as needed. For SSIM and LPIPS, we resize each object to the same dimension for calculating the scores.

These four basic metrics address the objectives of the object unlearning verification discussed in Section 4, however, as there are multiple occurring objects including those not subject to the unlearning request, we can further develop three dimensions of metrics for object unlearning:

- *A1: Removal of the unlearned objects.* We will compare the difference between the unlearned object generated by the original model and the unlearned model, to evaluate *unlearning effectiveness* as defined in Section 4. Greater differences of this metric, indicate better unlearning performance of the requested object.
- *A2: Preservation on the retained Objects.* By comparing the differences between "the retained objects of the sample" as generated by the original model and the unlearned model, we can evaluate *model utility preservation*, as defined in Section 4. Smaller differences of this metric, mean better unlearning focus on the requested objects.
- *A3: Preservation on the objects with the same category of the unlearned objects in other samples.* We will also compare the differences between "the objects with the same category of the unlearning objects in other samples" as generated by the original model and the unlearned model, an alternative perspective to evaluate *model utility preservation*. The smaller this metric, the better the unlearning focus on the specific sample.

We will present the metric in an abbreviated form in the evaluation section. For example, "A1_SSIM" represents the SSIM between the unlearned objects generated by the original model and the unlearned model. These metrics are summarized in Table **??**. It is important to note that these three dimensions should not be viewed individually, but rather in unison, to assess the tradeoffs of the unlearning process. Furthermore, to provide clarity for the reader, we have modify the distance metrics to follow a "larger is better" mantra. Specifically, we compute the complement values for A1_SSIM, A2_LPIPS, A3_LPIPS, A2_MAE, and A3_MAE for presentation within the plot. For SSIM and LPIPS, the complement transformation is 1 - value. For MAE, the complement transformation is 255 - value. The complement values are highlighted with a "∼" prefix. For A3 metrics, since they involve multiple other samples, we calculate and report the average value among samples.

Moreover, to provide a single unified measure of object unlearning performance, we introduce a *Triangular Performance Index* (TPI). TPI is a novel metric designed to quantify the overall effectiveness of a method across A1, A2, and A3. The motivation behind TPI is to offer a geometric interpretation of unlearning performance, where higher values across these metrics correspond to better outcomes. By mapping the three metric scores to the vertices of a triangle on a radar plot, the area enclosed by the triangle serves as an intuitive indicator of overall performance. A larger triangle area reflects stronger performance across all dimensions. To compute TPI, the metric values are first converted into polar coordinates, then plotted on a radar chart, and finally the area is calculated using vector cross products.

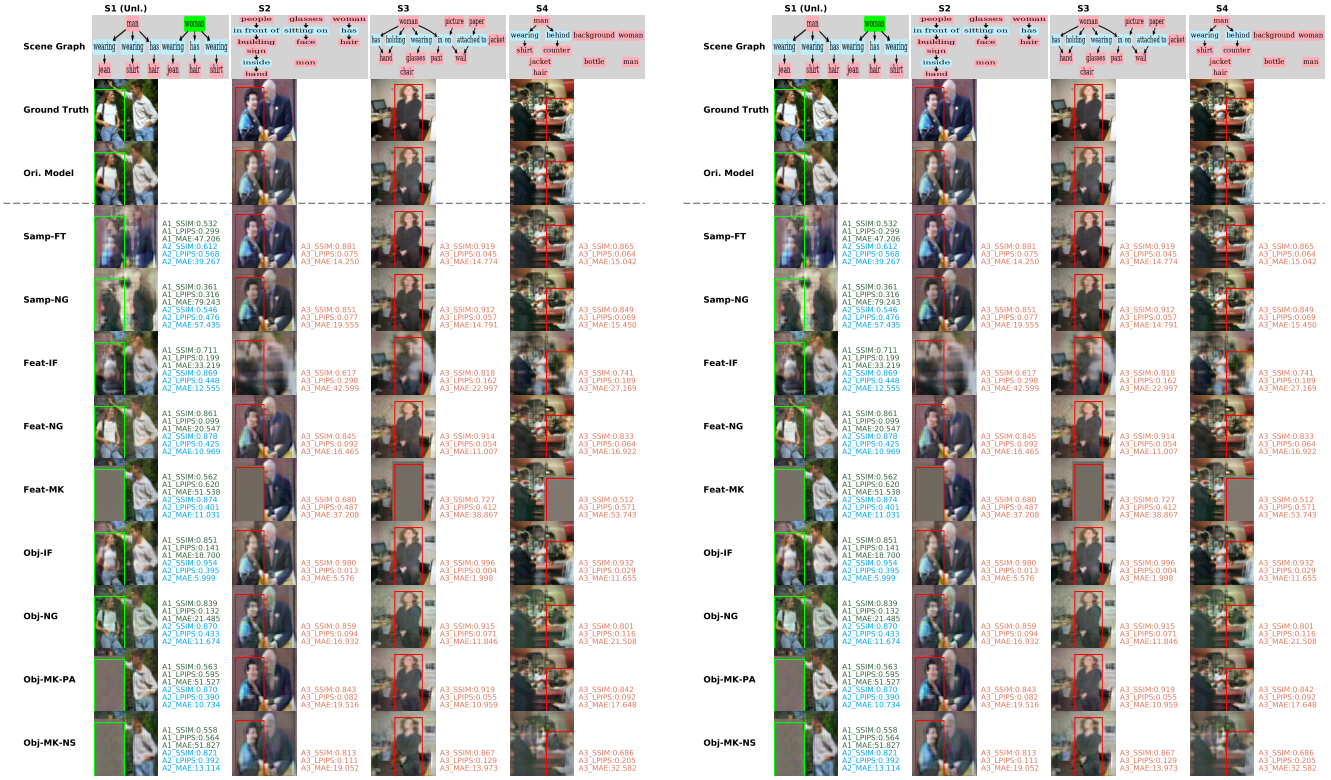


Figure 4: Results of generated samples (left: image reconstruction; right: image generation). In the images, the 'green boxes' localize the unlearned object, while the 'red boxes' indicate objects of the same category as the unlearned object but in different samples. In the scene graph visualization, the 'green node' represents the unlearned object. GT represents the ground truth.

Image Generation Training Settings. As described earlier, the image generator must be sufficiently capable to generate the original image before unlearning is applied. As such, we fine-tune the image generation technique on a smaller subset of the whole training set of Visual Genome dataset, that is the focus of unlearning within this experimentation set.

Unlearning Baselines. As discussed in Section 4, we shall evaluate the effectiveness of object unlearning by comparing the proposed framework against existing unlearning methods for sample unlearning. For this purpose, we introduce five baselines plus our devised four object unlearning-dedicated methods, they are:

- **Sample-FT**: Fine-tune the model by excluding the sample containing the requested object from the training dataset.
- **Sample-NG**: Fine-tune the model by applying negative guidance on the sample containing the requested object to reduce its influence.
- **Feat-IF**: Employ the influence function to remove features associated with the requested object.
- **Feat-NG**: Fine-tune the model with negative guidance applied to specific features associated with the requested object.
- **Feat-MK**: Fine-tune the model with a mask applied to features related to the requested object to obscure them.
- **Obj-IF**: Use the influence function to directly remove the requested object from the model's representation.
- **Obj-NG**: Fine-tune the model by applying negative guidance directly on the requested object to minimize its influence.
- **Obj-MK-PA**: Fine-tune the model with a patch mask applied to the feature area associated with the requested object, obscuring it within the model's internal representation.
- **Obj-MK-NS**: Fine-tune the model with a noise mask applied to the feature area containing the requested object to disrupt its learned features.

5.2 Unlearning Performance Evaluation

In evaluating the effectiveness of object unlearning, we employ image reconstruction as one key technique. Image reconstruction is selected for the inherent analogue it presents for our task of measuring object unlearning. Specifically, the input information used for reconstruction still exists within the object embeddings, making it a best case scenario for the model to recreate the original image. As such any impact of the unlearned object is due to the model no longer understanding this specific object. as such the poor restoration of the object, would indicates that the unlearning process has been highly effective.

This method provides a stringent test of the model's ability to selectively forget specific objects while retaining the integrity of other visual elements. We present both the metric results and sample outputs in Figures 3 and 4, respectively.

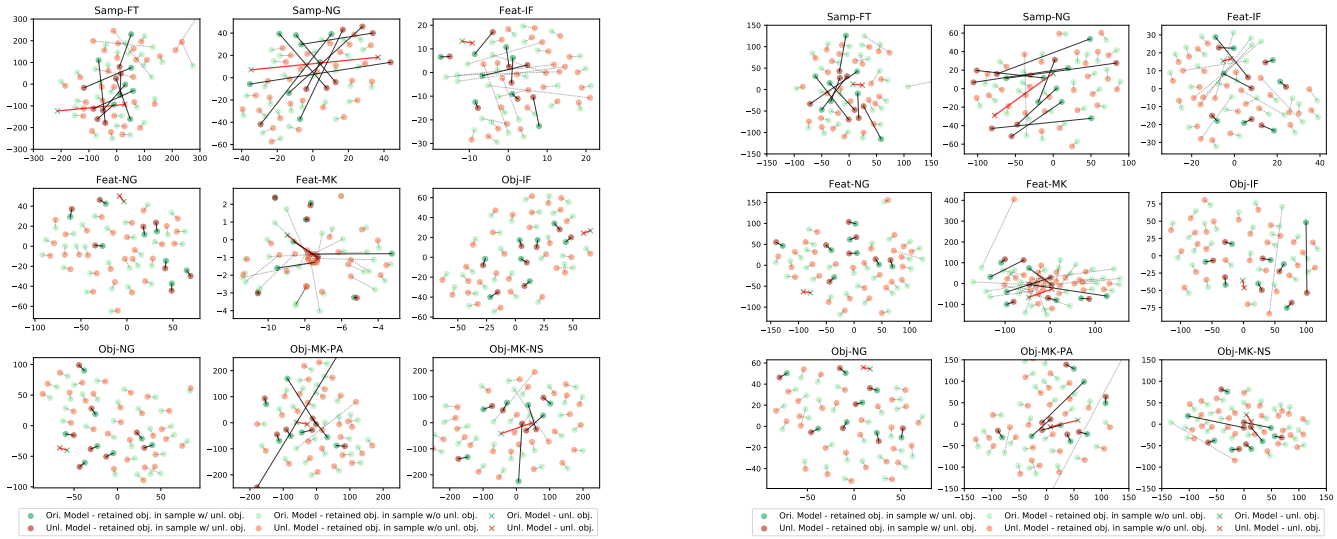


Figure 5: The latent features of the object, as developed by the original model and the unlearned models through various unlearning methods in both the image reconstruction (left) and generation (right) task, are projected into a two-dimensional space using t-SNE.

Observation 1: Methods based on negative guidance and model redaction demonstrate poor performance. These approaches either fail to effectively forget the requested object or significantly compromise model utility. In particular, the Obj-IF method, which we developed based on the existing influence function-based model redaction, demonstrated suboptimal performance. To further investigate the potential contributing factors, we conducted a detailed ablation study. This suggests that in more complex generation tasks, negative guidance must be designed in a more innovative manner to effectively facilitate the forgetting process.

Observation 2: All sample unlearning methods successfully eliminate the requested object; however, they also erase the remaining information from the original sample, resulting in a loss of model utility. This outcome aligns with our hypothesis that sample unlearning is limited to the sample level and struggles to achieve selective unlearning.

Observation 3: Among all the methods, the masking-based approach proves to be the most effective, i.e., Feat-MK, Obj-MK-PA, and Obj-MK-NS. As shown in the experimental results, the requested object is successfully removed while preserving other information in the image. This performance is notably superior to sample unlearning. Furthermore, unlike the feature unlearning method, objects with similar features in other images are retained. These results highlight the advantages of our approach, which enables selective unlearning with greater precision and efficiency.

5.3 Unlearning Analysis in Latent Space

We conducted a detailed comparison of the latent features of objects generated by the original model and the unlearned model. The results are visualized in Figure 5 for the image reconstruction and task. The analysis aligns well with the observed unlearning effects as discussed above. Generally, we have three major observations.

Observation 1: The methods Obj-MK-PA and Obj-MK-NS, which exhibited the best performance in unlearning, showed significantly larger latent space feature distances for the requested objects. In contrast, the distances for other objects in the same samples containing the requested object and for objects in samples without the requested object were relatively small. This confirms that these successful methods achieve effective unlearning by modifying the latent feature space of the requested object while maintaining the features of other objects.

Observation 2: Even among successful methods, we observed cases where the latent space feature distances for retained objects between the original and unlearned models were unexpectedly large. This is likely due to the inherent complexity of generative models, which can introduce random fluctuations in the latent space. These variations highlight the stochastic nature of generative processes and suggest that some unintended noise may affect the representation of retained objects.

Observation 3: Interestingly, we found little difference in latent space results between the image reconstruction and image synthesis tasks. This suggests a shared latent feature behavior across these tasks, despite their differences in objectives and output. However, further analysis is needed to better understand this consistency and its implications for unlearning in generative models.

5.4 Indirect Leakage Test

We then investigate whether object unlearning sufficiently removes knowledge of the object from nearby scene graphs related to the original object. To this end, we introduce three query variants. In each of these variants, we modify the category (label) of an object in the scene graph and query the SG2I model with this altered scene graph G' to determine whether the removed object might inadvertently reappear in the generated output. The three query types are: (1) *Fuzzy Query Attack*: Replace the unlearned object's category label in the scene graph with a more general or ambiguous label (e.g., replacing "man" with "person"). (2) *Neighbor Query Attack*: Modify the category label of an object in the scene graph adjacent

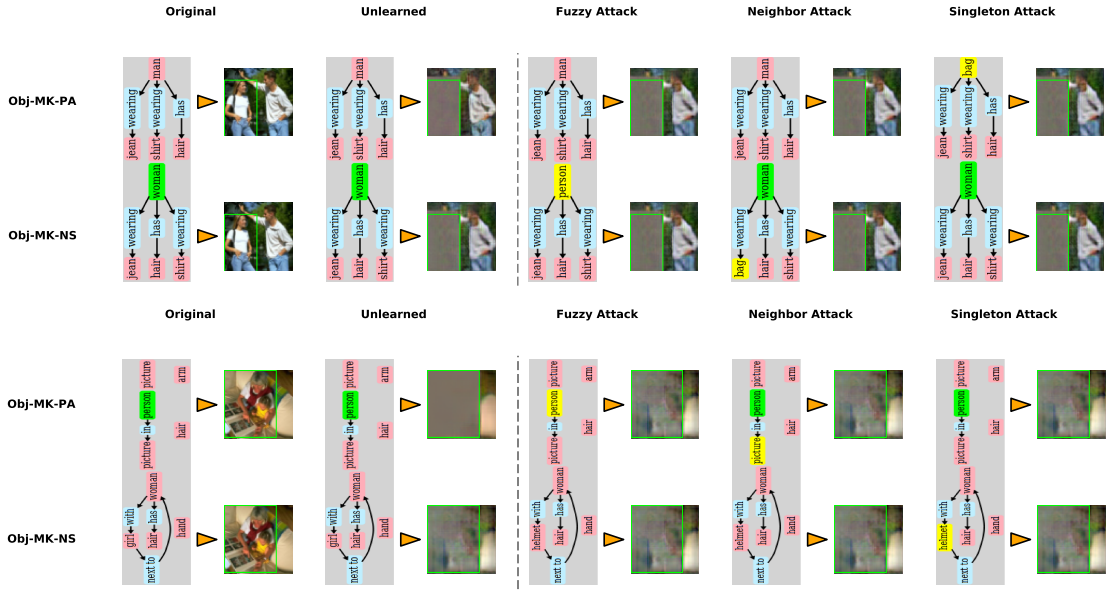


Figure 6: Indirect leakage tests on both the image reconstruction (upper) and generation (lower) task. The modified object label in the scene graph is highlighted in 'yellow boxes'. It is clear that none of these three attacks can successfully make the model leak information about objects that have been unlearned.

to, but not directly connected to, the unlearned object node. (3) *Singleton Query Attack*: Modify the category label of an isolated object in the scene graph, unconnected to the unlearned object. Here, we specifically selected Obj-MK-PA and Obj-MK-NS, as they demonstrated the best performance in the previous experiments.

Figure 6 visualize the outcomes across these attacks. The resulting generation remains consistently lacking, signifying that the object learning is robust to small alterations in the scene graphs. In other words, these query strategies failed to bypass the unlearning process or indirectly infer the unlearned object. This finding highlights the robustness of the unlearning process: objects that have been intentionally removed remain secure, and users cannot manipulate input queries to reveal forgotten content.

This robustness suggests that, in most practical scenarios, models with unlearning applied are unlikely to expose forgotten objects, even under query-based probing. However, we suspect that more targeted and advanced in-training methods might be needed to intentionally make the model leak unlearned objects. Investigating these methods and their effects will be a key focus of future research.

6 Discussion

Our work pioneers fine-grained object unlearning using scene graphs, but several challenges and future directions remain. First, unlearning effectiveness varies significantly with model scale — small models may over-forget while large models exhibit resistance due to stronger generalization. Second, increasing architectural complexity in generative pipelines raises new questions about where and how to apply unlearning interventions. Third, although our framework supports extensibility to multimodal scenarios and privacy-preserving unlearning requests, robust verification metrics and defense against adversarial threats remain open concerns. Notably, while our research is not grounded in core computer vision model development, we view this work as a conceptual and methodological starting point. We encourage the vision community to further explore such *entity-level unlearning leveraging structured representations* within more advanced and emerging generative architectures (e.g., transformer-based or token diffusion-based frameworks), where new opportunities and challenges await.

7 Conclusion

In this paper, we introduce a novel framework for object unlearning, specifically addressing the limitations of current unlearning approaches on handling granular unlearning request. Unlike traditional sample or feature unlearning methods, our scene graph-based approach provides a targeted unlearning mechanism that selectively removes sensitive objects while preserving the utility of other, non-requested elements in the data. We validate the effectiveness of this framework through extensive evaluations on image reconstruction and synthesis tasks, demonstrating its superior ability to obscure unlearned objects without compromising the overall quality of the generated images. By leveraging influence functions to approximate the unlearning process, we mitigate the computational costs typically associated with generative models. Our findings highlight the importance of fine-grained unlearning in addressing user's varying data removal requests, while preserving the integrity and utility of the original dataset. This work lays the foundation for more precise unlearning methods and paves the way for future research aimed at enhancing privacy protections in MLaaS platforms without sacrificing model utility.

Ethical Considerations

Our study involves the use of images that may contain human faces, representative of the private personal information that is to be unlearned from a given image generation model. As these datasets are public, with a long history of works having used these images for research, there are no new ethical implications for the humans within the dataset. The ultimate goal of this research is to develop privacy enhancing technologies, which in the long term provides tools that may be use to correct breaches in individual privacy thereby satisfying "Beneficence" and "Justice". Our work does not contain any user participants, and the outcome of this research influences benefits all individuals equally, to allow large model holders more effective methods to forget private data upon request. As such "Respect for Persons" is satisfied.

References

[1] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine Unlearning. In *SP*. IEEE, 141–159.

[2] California Consumer Privacy Act (CCPA) 2020. California Consumer Privacy Act (CCPA). https://oag.ca.gov/privacy/ccpa. Accessed: 2020-02-14.

[3] Yinzhi Cao and Junfeng Yang. 2015. Towards Making Systems Forget with Machine Unlearning. In *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 463–480.

[4] Wenhan Chang, Tianqing Zhu, Yufeng Wu, and Wanlei Zhou. 2024. Zero-shot Class Unlearning via Layer-wise Relevance Analysis and Neuronal Path Perturbation. *arXiv preprint arXiv:2410.23693* (2024).

[5] Wenhan Chang, Tianqing Zhu, Heng Xu, Wenjian Liu, and Wanlei Zhou. 2024. Class machine unlearning for complex data via concepts inference and data poisoning. *arXiv preprint arXiv:2405.15662* (2024).

[6] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. 2023. A Comprehensive Survey of Scene Graphs: Generation and Application. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 1 (2023), 1–26. doi:10.1109/TPAMI.2021.3137605

[7] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2022. Graph unlearning. In *Proc. ACM SIGSAC Conference on Computer and Communications Security*. 499–513.

[8] Jiali Cheng, George Dasoulas, Huan He, Chirag Agarwal, and Marinka Zitnik. 2023. GNNDelete: A General Unlearning Strategy for Graph Neural Networks. In *Proc. International Conference on Learning Representations*.

[9] Children's Online Privacy Protection Rule (COPPA) 1998. Children's Online Privacy Protection Rule (COPPA). https://www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/childrens-online-privacy-protection-rule. Accessed: 2020-02-14.

[10] Helisa Dhamo, Azade Farshad, Iro Laina, Nassir Navab, Gregory D. Hager, Federico Tombari, and Christian Rupprecht. 2020. Semantic Image Manipulation Using Scene Graphs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 5212–5221. doi:10.1109/CVPR42600.2020.00526

[11] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. 2023. Erasing Concepts from Diffusion Models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 2426–2436. doi:10.1109/ICCV51070.2023.00230

[12] General Data Protection Regulation (GDPR) 2018. General Data Protection Regulation (GDPR). https://gdpr-info.eu/. Accessed: 2020-02-14.

[13] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. 2021. Adaptive machine unlearning. *Advances in Neural Information Processing Systems* 34 (2021), 16319–16330.

[14] Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. 2022. Federated unlearning: How to efficiently erase a client in fl? *arXiv preprint arXiv:2207.05521* (2022).

[15] Hongsheng Hu, Shuo Wang, Tian Dong, and Minhui Xue. 2024. Learn What You Want to Unlearn: Unlearning Inversion Attacks against Machine Unlearning. *CoRR* abs/2404.03233 (2024). arXiv:2404.03233 doi:10.48550/ARXIV.2404.03233

[16] Miao Hua, Lijie Liu, Ziyang Cheng, Qian He, Bingchuan Li, and Zili Yi. 2021. FaceEraser: Removing Facial Parts for Augmented Reality. *arXiv preprint arXiv:2109.10760* (2021).

[17] National Cancer Institute. 2024. Targeted Therapy for Cancer. https://www.cancer.gov/about-cancer/treatment/types/targeted-therapies. Accessed 15 May 2025.

[18] Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image Generation From Scene Graphs. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 1219–1228. doi:10.1109/CVPR.2018.00133

[19] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3668–3678.

[20] Zhifeng Kong and Kamalika Chaudhuri. 2023. Data Redaction from Pre-trained GANs. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2023, Raleigh, NC, USA, February 8-10, 2023*. IEEE, 638–677. doi:10.1109/SATML54575.2023.00048

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.* 123, 1 (2017), 32–73. doi:10.1007/S11263-016-0981-7

[22] Hongsheng Li, Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Xia Zhao, Syed Afaq Ali Shah, and Mohammed Bennamoun. 2024. Scene Graph Generation: A comprehensive survey. *Neurocomputing* 566 (2024), 127052. doi:10.1016/J.NEUCOM.2023.127052

[23] Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. 2021. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*. IEEE, 1–10.

[24] Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. 2024. Machine Unlearning in Generative AI: A Survey. *CoRR* abs/2407.20516 (2024).

[25] Yueming Lyu, Yue Jiang, Ziwen He, Bo Peng, Yunfan Liu, and Jing Dong. 2023. 3D-Aware Adversarial Makeup Generation for Facial Privacy Protection. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 11 (2023), 13438–13453. doi:10.1109/TPAMI.2023.3290175

[26] Gaurav Mittal, Shubham Agrawal, Anuva Agarwal, Sushant Mehta, and Tanya Marwah. 2019. Interactive Image Generation Using Scene Graphs. In *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net. https://openreview.net/forum?id=SJx-SULKOV

[27] Saemi Moon, Seunghyuk Cho, and Dongwoo Kim. 2024. Feature Unlearning for Pre-trained GANs and VAEs. In *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*, Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (Eds.). AAAI Press, 21420–21428. doi:10.1609/AAAI.V38I19.30138

[28] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. 2020. Variational bayesian unlearning. *Advances in Neural Information Processing Systems* 33 (2020), 16025–16036.

[29] Chao Pan, Eli Chien, and Olgica Milenkovic. 2023. Unlearning graph classifiers with limited data resources. In *Proceedings of the ACM Web Conference 2023*. 716–726.

[30] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2337–2346. doi:10.1109/CVPR.2019.00244

[31] Proposal for an ePrivacy Regulation 2019. Proposal for an ePrivacy Regulation. https://ec.europa.eu/digital-single-market/en/proposal-eprivacy-regulation. Accessed: 2020-02-14.

[32] Anwar Said, Tyler Derr, Mudassir Shabbir, Waseem Abbas, and Xenofon Koutsoukos. 2023. A survey of graph unlearning. *arXiv preprint arXiv:2310.02164* (2023).

[33] Kaede Shiohara, Xingchao Yang, and Takafumi Taketomi. 2023. BlendFace: Re-designing Identity Encoders for Face-Swapping. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 7600–7610. doi:10.1109/ICCV51070.2023.00702

[34] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.

[35] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. 2022. On the Necessity of Auditable Algorithmic Definitions for Machine Unlearning. In *USENIX Security Symposium*. USENIX Association, 4007–4022.

[36] Subarna Tripathi, Anahita Bhiwandiwalla, Alexei Bastidas, and Hanlin Tang. 2019. Using Scene Graph Context to Improve Image Generation. *CoRR* abs/1901.03762 (2019). arXiv:1901.03762 http://arxiv.org/abs/1901.03762

[37] Cheng-Long Wang, Mengdi Huai, and Di Wang. 2023. Inductive Graph Unlearning. In *USENIX Security Symposium*. USENIX Association, 3205–3222.

[38] Jing Wang, Jun Wang, Jianhou Gan, and Juxiang Zhou. 2021. Face Privacy Protection Based on Attribute Manipulation. In *ICIT 2021: IoT and Smart City, Guangzhou, China, December 22 - 25, 2021*. ACM, 185–188. doi:10.1145/3512576.3512609

[39] Zhibo Wang, He Wang, Shuaifan Jin, Wenwen Zhang, Jiahui Hu, Yan Wang, Peng Sun, Wei Yuan, Kaixin Liu, and Kui Ren. 2023. Privacy-preserving Adversarial Facial Features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 8212–8221. doi:10.1109/CVPR52729.2023.00794

[40] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. 2023. Machine Unlearning of Features and Labels. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society. https://www.ndss-symposium.org/ndss-paper/machine-unlearning-of-features-and-labels/

[41] Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. 2023. GIF: A General Graph Unlearning Strategy via Influence Function. In *Proceedings of the ACM Web Conference 2023*. 651–661.

[42] Kun Wu, Jie Shen, Yue Ning, Ting Wang, and Wendy Hui Wang. 2023. Certified Edge Unlearning for Graph Neural Networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2606–2617.

[43] Lin Yuan, Linguo Liu, Xiao Pu, Zhao Li, Hongbo Li, and Xinbo Gao. 2022. PRO-Face: A Generic Framework for Privacy-preserving Recognizable Obfuscation of Face Images. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, João Magalhães, Alberto Del Bimbo, Shin'ichi Satoh, Nicu Sebe, Xavier Alameda-Pineda, Qin Jin, Vincent Oria, and Laura Toni (Eds.). ACM, 1661–1669. doi:10.1145/3503161.3548202

[44] Eric J. Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. 2023. Forget-Me-Not: Learning to Forget in Text-to-Image Diffusion Models. *CoRR* abs/2303.17591 (2023). arXiv:2303.17591 doi:10.48550/ARXIV.2303.17591

[45] Peng-Fei Zhang, Guangdong Bai, Zi Huang, and Xin-Shun Xu. 2022. Machine Unlearning for Image Retrieval: A Generative Scrubbing Approach. In *ACM Multimedia*. ACM, 237–245.

[46] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*. Computer Vision Foundation / IEEE Computer Society, 586–595.

[47] Yihua Zhang, Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jiancheng Liu, Xiaoming Liu, and Sijia Liu. 2024. UnlearnCanvas: A Stylized Image Dataset to Benchmark Machine Unlearning for Diffusion Models. *CoRR* abs/2402.11846 (2024). arXiv:2402.11846 doi:10.48550/ARXIV.2402.11846

[48] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. 2019. Image Generation From Layout. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 8584–8593. doi:10.1109/CVPR.2019.00878