Zero-shot CLIP Class Unlearning via Text-image Space Adaptation

Anonymous authors
Paper under double-blind review

Abstract

Efficient machine unlearning has attracted significant interest due to the high computational cost of retraining models from scratch whenever data needs to be forgotten. This need arises from data privacy regulations, the necessity to update outdated information, and the possibility to enhance model robustness and security.

In this paper we address class unlearning in vision-language CLIP model. Modern unlearning methods for CLIP have demonstrated that zero-shot forgetting is achievable by generating synthetic data and fine-tuning both visual and textual encoders with a regularization loss. Our approach shows that unlearning in CLIP can be accomplished in a zero-shot manner without any visual data by adapting the shared vision-text space of CLIP, thereby making the unlearning process more efficient. Our method delivers superior results, demonstrating strong performance and complete forgetting, regardless of the visual encoder used in CLIP. Furthermore, we explore what exactly is being targeted by the unlearning algorithm discovering some interesting properties of CLIP features.

1 Introduction

Machine unlearning Xu et al. (2023) involves removing specific data points from a trained model without a full retraining, ensuring that the influence of this data is entirely removed. This process aims to make the model behave similarly to one that was never exposed to the data points that were removed. The importance of machine unlearning has been highlighted in contexts of data privacy laws like the General Data Protection Regulation (GDPR) which stipulates the right to be forgotten [1]. Users have the right to request the removal of their data from all systems including those embedded in the models. Mere deletion of data from storage is inadequate in such cases as model's weights still retain information about that data. Machine unlearning is also crucial in scenarios involving the update of outdated information Oren & Keromytis (2015) and has also shown to be useful in model security and robustness against adversarial attacks Xi Wu (2016); Biggio et al. (2013).

In this work we specifically address class unlearning in CLIP Radford et al. (2021), aiming to break the association between visual and textual representations of the forget class. CLIP is a vision-language model widely recognized in the computer vision community. It has been extensively adopted across various real-world applications, including robotics control Shridhar et al. (2021), zero-shot object tracking Solawetz (2021), content moderation Ahmed et al. (2023), image classification Radford et al. (2021) and retrieval Sain et al. (2023) among others. Given CLIP's broad application and influence, ensuring that it can unlearn specific data is crucial particularly when sensitive or proprietary information is embedded within the model. If CLIP has inadvertently learned to recognize such information during training it can propagate this knowledge across various applications posing significant ethical and legal risks. Furthermore, the lack of access to its original training data[2] makes it hard to identify what information is embedded in CLIP. This lack of access to the training data also complicates unlearning as many existing unlearning techniques rely on the availability of the training data Cheng & Amiri (2023); Fan et al. (2023); Foster et al. (2024).

---

[1] https://gdpr-info.eu/art-17-gdpr/

[2] https://github.com/openai/CLIP/issues/127

Unlearning in CLIP is challenging due to the following reasons: a) we do not have access to the original data used for training CLIP. Thus, any retraining of CLIP to achieve unlearning is not feasible b) CLIP is a large parameter model. Even if we did obtain access to data that needs to be forgotten from CLIP, fine-tuning CLIP would be challenging. To the best of our knowledge, only one study Kravets & Namboodiri (2024) has addressed zero-shot unlearning in CLIP. This study demonstrates unlearning in a zero-shot manner without requiring any real data. They further indicate that changing weights in both the visual and textual encoders is necessary to forget a specific class. In contrast, we demonstrate that unlearning in CLIP can be achieved by modifying only a small part of the textual encoder responsible for projecting the textual representation of the class into the shared image-text embedding space. Moreover, their approach relies on synthetic samples which can be time-consuming to generate making the unlearning process relatively slow. As our approach can unlearn only based on the textual representation, it does not require any sample generation.

We recognize that at its core, the contrastive learning for CLIP aims to obtain a joint embedding space for the image and textual representation. Hence, we explicitly use projection of the textual representation to achieve forgetting. Our approach uses a direct optimization of a loss function to modify the text representation projection matrix. While doing so, we need to ascertain the gap between image and text representations is modified only for a select set of classes that we desire to be unlearned while maintaining the gap between image and text representations for the classes that need to be retained. Once we do this for the text representations, we observe that we achieve unlearning for the image-text classes that need to be forgotten and preserve the image-text correspondence for the other classes. After the optimization process is completed, an image for a retained class would still be close to the corresponding text representation. However, for the class that is unlearned, the image representation would be the same as the initial representation but the textual representation would be differentas it has been explicitly modified for this class. For the optimization we apply an adaptation technique, low-rank adaptation (LoRA) Hu et al. (2021), to find the minimum change in the text projection matrix optimizing a loss function that ensures that the change is such that the representation of the non-forget classes is retained while altering the representation of the forget class.

We do a **performance** comparison in Tab. 1 showing that our method both outperforms the previous methods and is more robust to different visual encoders achieving perfect forgetting with both ViT Dosovitskiy et al. (2020) and ResNet He et al. (2015). We analyse through ablations the importance of retain and forget **loss components** in Section 7.2 and how **forget class projection place** in the image-text space affects the unlearning ability of the model in Section 7.5. We find that retaining the knowledge of non-forget classes requires the inclusion of semantically similar classes, which can be generated using a large language model (LLM). This is because projecting the forget class to a different space primarily affects the closest classes in the image-text embedding space, thus, it is important to preserve this part of the space, while non-semantically similar classes are retained without explicit inclusion. We conduct a thorough ablation analysis on the how the **number of semantically similar classes** affects performance in Section 7.3. Additionally, in Section 7.4 we assess how including semantically different classes affects performance.

We investigate how unlearning happens. In Section 6.3 we show that there exist some **"magic" neurons** that the unlearning algorithm targets. These weights are such that changing them decreases the dot product for the class to forget the most leading to a change in the class prediction. Furthermore, in Appendix D we show that there is a positive relation between the **difficulty of unlearning a class** and the Frobenious norm in the matrix of weights change. An overview of our approach is provided in Fig 1. Our **contributions** are summarized as follows:

- We improve current state-of-the-art CLIP unlearning keeping it zero-shot as shown in Section 6.1.
- We unlearn without generating synthetic visual data improving efficiency. In contrast to previous work we show that no visual data is required to unlearn and textual data is enough (Section 4).
- We provide a thorough analysis to understand our method. We show that there exist some "magic" neurons that our method targets to achieve unlearning and why it does that in Section 6.3. We also show that the Frobenius norm correlates with the difficulty of unlearning in Appendix D.
- A detailed analysis provided in Section 7 validates the choices made in our method and the generalizability of our method.

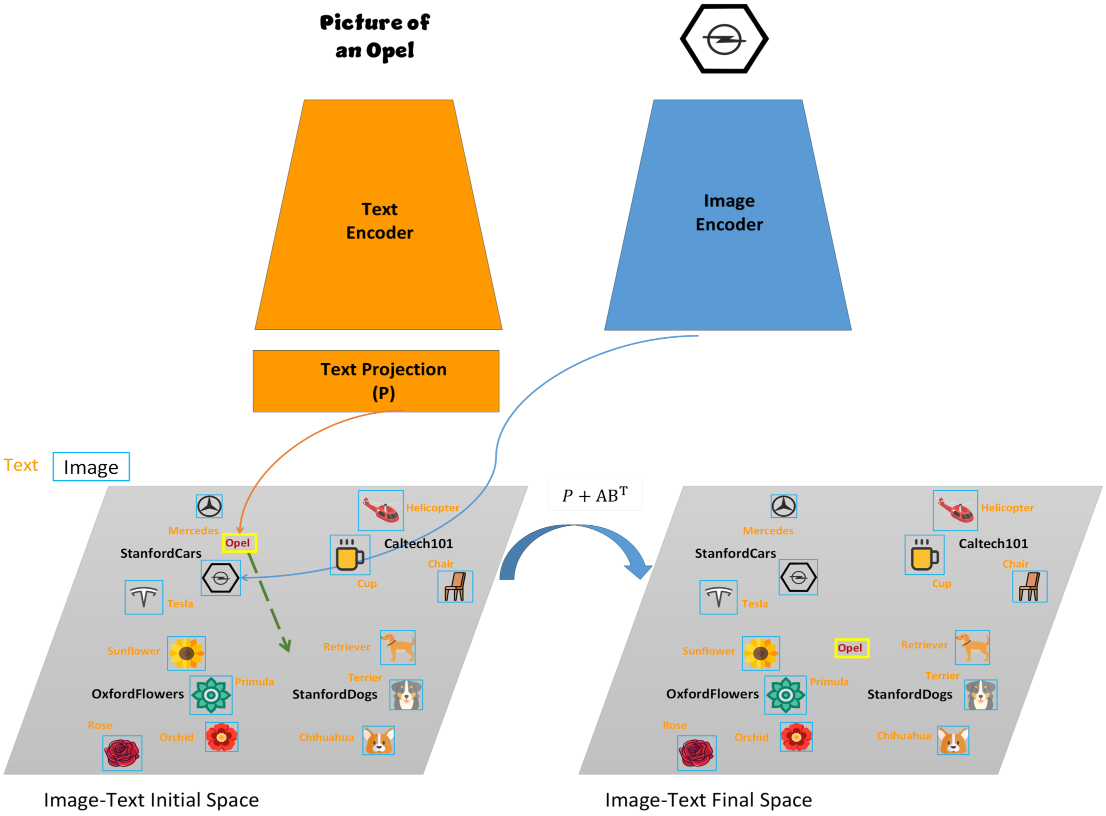


Figure 1: **Overview of the approach.** We utilize LoRA to adapt the projection matrix of the textual representation into the shared image-text space. We ensure the representation for the non forget classes is retained while altering it for the class to forget. To maintain the representation of the non forget classes we generate some semantically similar classes using an LLM. On the other hand, the forget class is projected into the empty token representation in the image-text space. In the figure we illustrate the forgetting for the *Opel* class.

2 Related Work

Machine Unlearning Machine unlearning can be categorized into data-based and model-based unlearning methods Xu et al. (2023). Data-based unlearning involves manipulating the training data, such as randomly relabeling data related to the classes to be forgotten Graves et al. (2020); Felps et al. (2020), generating anti-samples with visual patterns opposite to the class to be forgotten and labeling them with the forget class label Tarun et al. (2022), or replacing the data with various transformations that facilitate unlearning Cao & Yang (2015); Shibata & Mitsuzumi (2021).

Model-based unlearning methods, on the other hand, act directly on the model weights. Some methods use the Hessian to perform a Newton update on the converged weights in the opposite direction of gradient descent, thereby increasing the loss for the forget data Guo et al. (2019); Golatkar et al. (2021); Sekhari et al. (2021). Other methods involve pruning weights based on their saliency Fan et al. (2023) or on the contribution that certain features channel play in order to predict a class using Term Frequency-Inverse Document Frequency (TF-IDF) Wang et al. (2022). Additionally, some model-based methods use a generation-discriminator setting optimizing the original model until the predicted outputs on the forget classes and unseen classes become indistinguishable to the discriminator Chen et al. (2021). Another approach is knowledge distillation, where the student model is trained to mimic the original teacher model's output except for the classes to be forgotten Chundawat et al. (2023).

Multimodal Unlearning The authors in Li et al. (2024a) propose a method to forget visual recognition of concepts using a single image for multimodal models. They begin by creating a multifaceted fine-tuning dataset aimed at aligning the forget concept with unseen concepts, assigning it a new visual description, decoupling factual knowledge about it, and preserving unrelated knowledge. The model is then fine-tuned with this data using a dual masked KL-divergence loss. Similarly, the authors in Cheng & Amiri (2023) achieve unlearning with a three-term loss function designed to ensure modality decoupling, unimodal knowledge retention, and multimodal knowledge retention. Both methods require real training examples and are not applicable to the CLIP dual encoder model.

In contrast, Zhang et al. (2023) and Gandikota et al. (2023a) achieve class forgetting in the Diffusion model by modifying cross-attention mechanisms, thereby disrupting the associations between visual and textual representations of the concepts to be forgotten. Similarly, Gandikota et al. (2023b) act on cross-attention in the Diffusion model optimizing the key and value matrices mapping target concepts to a new one while preserving some other concepts. This technique allows editing, debiasing and erasure of concepts in the Diffusion model. Since CLIP does not utilize cross-attention, these methods are not directly applicable.

To the best of our knowledge, Kravets & Namboodiri (2024) is the first paper to address zero-shot unlearning in CLIP by applying Lipschitz regularization. This approach guides the embedding of the visual and textual representations of the forget class towards a perturbed embedding, breaking the visual-textual association while retaining knowledge of other classes. They achieve unlearning in a zero-shot manner by generating synthetic visual data, thus eliminating the need for real examples. The method involves modifying weights in both visual and textual encoders. In contrast, we show that simply updating the projection matrix from the text to image space is sufficient to achieve unlearning still in a zero-shot manner without requiring any real nor synthetic images and unlearning more efficiently as synthetic image generation requires time.

Model Adaptation Model adapters are task-specific modules added to a pre-trained model to enable it to efficiently adapt to new downstream task without retraining the entire model. Houlsby et al. (2019) inserted sequentially a small multilayer perceptron (MLP) layer between the layers of a pre-trained BERT model while freezing the original pre-trained parameters. Similarly, authors in Chen et al. (2022) added MLP layers but in parallel to the original frozen MLP connecting them in a residual fashion showing superiority compared to the sequential adaptation. Pfeiffer et al. (2021) proposed a unified framework for training and sharing adapters across various tasks. Hu et al. (2021) introduced Low-Rank Adaptation (LoRA) technique which injects trainable low-rank matrices to learn task-specific information without altering the original pre-trained weights significantly.

We utilize adapters, and specifically LoRA to fine-tune the text projection matrix into the shared image-text embedding space in order to unlearn a class in CLIP.

3 Preliminaries

CLIP CLIP is a multimodal model that understands both visual and textual inputs. It has been trained with a contrastive loss, which helps it learn to represent similar images and their textual descriptions closely in a shared image-text embedding space while keeping dissimilar ones apart. Contrastive training enables CLIP to perform various tasks, such as classification and retrieval, in a zero-shot manner. CLIP has a dual encoder architecture, with separate encoders for images and text. The textual encoder is a Transformer Vaswani et al. (2017) neural network that ends with a projection matrix, which projects the textual representations into a shared vision-text embedding space. This projection matrix is the key in our forgetting method.

Setup for Machine Unlearning Given a trained model and its training data D, the goal of machine unlearning is to forget the information about some selected data points used during training denoted as forget set $D \subseteq D_f$ while retaining the information about other data denoted as retain set $D_r = D \setminus D_f$. The general objective for unlearning can be defined as an optimization problem minimizing the following loss:

$$\mathcal{L} = \mathcal{L}_{\text{forget}}(D_f) + \alpha \mathcal{L}_{\text{retain}}(D_r) \tag{1}$$

Training data for CLIP, D, consists of both visual and textual data. Previous research Kravets & Namboodiri (2024) has shown that to unlearn we also require both visual and textual examples, where visual can also be synthetic examples. In contrast, we show that textual data are enough for unlearning. Our aim for multimodal CLIP unlearning is to achieve modality decoupling, which involves breaking the associations between visual and textual representations for the class to forget.

4 Method

Loss Our method relies on an optimization approach that directly considers an explicit low-rank adaptation of the text projection matrix into the shared image-text representation space. The main principles we use are that the image-text representation space should be minimally changed for the classes that are to be retained and should be changed for the classes that are to be forgotten in a systematic manner such that unlearning is achieved. We also want the low-rank transformation matrices to be sparse to increase efficiency. These requirements lead to direct terms in our optimization approach. Note that similar optimization approaches for unlearning have been used previously for instance in LLM unlearning Li et al. (2024b) and Stable Diffusion concept editing Gandikota et al. (2023b). However, our contribution lies not exactly in the optimization approach. Rather, our main contribution is a straightforward method to achieve unlearning in CLIP by using constraints on the text project matrix representation. This enables us to achieve forgetting in CLIP without requiring the actual data used for training (which is not available for CLIP). Further, our approach differs in being a low-rank adaptation that provides an explicit unlearning to be achieved as the change in projection is known precisely through the low-rank projection adaptation. As we only change one matrix we can track what is being changed by our algorithm - we find that there exist some "magic" neurons that the method targets in a specific manner to unlearn. This analysis is provided in in Section 6.3. Also, we show in Appendix D that there is a positive relation between the difficulty of unlearning a class and the Frobenious norm in the matrix of weights change.

Given CLIP textual encoder f_θ that encodes input text into vector representation and P the projection matrix that projects this representation into the image-text shared space, we optimize the following loss:

$$\begin{aligned}\mathcal{L} = \lambda_1 &\left\| \left(P_{[n\times m]} + A_{[n\times r]} B^\top_{[m\times r]}\right) f_\theta(X_r)_{[m\times m]} - P_{[n\times m]} f_\theta(X_r)_{[m\times m]} \right\|_2^2 + \\ \lambda_2 &\left\| \left(P_{[n\times m]} + A_{[n\times r]} B^\top_{[m\times r]}\right) f_\theta(X_f)_{[m\times m]} - F \right\|_2^2 + \\ \lambda_3 &\left\| A_{[n\times r]} B^\top_{[m\times r]} - \mathbf{0}_{[n\times m]} \right\|_2^2\end{aligned} \tag{2}$$

Where X_r are the textual classes to retain, X_f textual classes to forget and F is the new representation of the forget class that we discuss below. We aim to optimize the variation of P using LoRA, with $A_{[n\times r]}$ and $B_{[m\times r]}$ being low-rank matrices. The first component of the loss ensures that classes to retain are maintained close to their original position in the embedding space. The second component modifies the projection matrix P such that the class to forget is projected into a new position of the image-text space F. The third component ensures that this is done with minimum modification to P.

We cannot include all classes seen by CLIP during its contrastive pre-training in X_r since these are unknown. However, we find that including semantically similar classes to the forget class suffices to keep the representations of all the retain textual classes we tested on fairly untouched. Indeed, it's important to include semantically similar classes because when forgetting a class we perturb the space around that class which affects representation of similar classes, thus preserving those ensures that only the forget class is projected to a different part of the image-text space while retaining classes that were close to it in the embedding space. As we show in the ablations in Section 7.3, retaining any type of classes is not useful as it reduces the performance on classes of the dataset the forget class was picked from. To generate semantically similar classes, we use a large language model (LLM) using a prompt *"Generate semantically similar classes to {class}"*. These are shown in the Appendix G.

To determine where to project the forget class, denoted F in Eq. 2, we use the empty token representation. In the ablations in Section. 7.5 we tested other variations such as a random projection and a perturbed representation of the forget class, which lead to slightly worse results.

Determination of the Loss Parameters We fix λ_1 and λ_3 while λ_2 is is determined iteratively. At each iteration, we assess the reduction in the second component of the loss to evaluate whether the change in the projection matrix P is sufficient to project the forget class to the new chosen vector. We start from a fixed λ_2 and increase it in small steps until the reduction in the second loss component exceeds 0.75% of its initial value. Additional implementation details are described in the Appendix E.

5 Experiments

5.1 Comparable Methods

There exist only one directly comparable method on CLIP unlearning, while other are adapted from other methods. We only compare our approach to zero-shot methods that do not require any real data to unlearn.

Lipschitz CLIP Unlearning (Lip) To unlearn specific classes authors in Kravets & Namboodiri (2024) locally perturb both image and text representation of the forget class by a Gaussian noise and minimize the Lipschitz regularization loss updating both the encoders. The method is zero-shot because instead of the original images synthetic images are utilized that are generated by gradient ascent.

Embedding regularization loss (Emb) Similar to the above, instead of Lipschitz regularization loss a simple difference between embeddings with L1 regularization term is used.

Amnesiac forgetting with synthetic data (Amns) The approach from Graves et al. (2020) is adapted to a multimodal setting by fine-tuning CLIP using the same contrastive loss employed in its initial training. In this approach, the labels of the class to forget are randomly replaced with different labels using synthetic data. To maintain zero-shot setting, data from the classes to retain are not utilized and solely data for the class to forget are employed to unlearn.

Error Minimization-Maximization Noise (EMMN) The approach from Chundawat et al. (2023) is adapted to multimodal setting learning retain and forget samples through loss minimization and maximization respectively and training the model on these samples.

5.2 Datasets

Following Kravets & Namboodiri (2024) we evaluate CLIP's forgetting capabilities on four high-quality, fine-grained datasets: Caltech101 Fei-Fei et al. (2007) contains images from 101 distinct categories, each representing various objects or scenes. StanfordCars Krause et al. (2013) contains images of cars of different makes and models. OxfordFlowers Nilsback & Zisserman (2008) includes images of flowers of 102 different classes. StanfordDogs Khosla et al. (2011) comprises 120 classes of dogs of different species.

5.3 Evaluation

Ideally, to assess the forgetting procedure we should compare against the retrained model without the forget class. However, as CLIP training data are unknown and even if they were open sourced the computational power required to assess against a retrained model would be prohibitive, we adopt a similar logic to Kravets & Namboodiri (2024) in order to assess how well the class has been forgotten. We want the accuracy on the forget class to be as low as possible while maintaining the accuracy on other classes to a similar level before forgetting. As we need to compare different quantities such as the drop in accuracy of the forget class, the remaining accuracy of the dataset the class was picked from and remaining accuracy on other datasets we create an aggregated metrics for an easier comparison. Given the normalized reduction in the accuracy of

Table 1: Main forgetting results. We compare our method to five other methods averaging across three classes for four selected datasets.

Method	**Model**	**Dataset**	**Avg. Target Class acc.**		**Avg. Other Classes acc.**		**Avg. StanfordCars**		**Avg. StanfordDogs**		**Avg. Caltech101**		**Avg. OxfordFlowers**		**Avg. Score (↓)**
			BF	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	
Ours	RN50	StanfordCars	0.397	0.0	0.558	0.55	-	-	0.517	0.51	0.857	0.855	0.661	0.657	**0.007**
Lip	RN50	StanfordCars	0.397	0.056	0.558	0.551	-	-	0.517	0.513	0.857	0.86	0.661	0.653	0.034
Emb	RN50	StanfordCars	0.397	0.087	0.558	0.536	-	-	0.517	0.51	0.857	0.85	0.661	0.649	0.06
Amns	RN50	StanfordCars	0.397	0.357	0.558	0.498	-	-	0.517	0.505	0.857	0.863	0.661	0.653	0.208
EMMN	RN50	StanfordCars	0.397	0.0	0.558	0.054	-	-	0.517	0.043	0.857	0.424	0.661	0.069	0.644
Ours	RN50	StanfordDogs	0.593	0.0	0.516	0.509	0.558	0.554	-	-	0.857	0.856	0.661	0.653	**0.007**
Lip	RN50	StanfordDogs	0.593	0.048	0.516	0.516	0.558	0.558	-	-	0.857	0.866	0.661	0.655	0.018
Emb	RN50	StanfordDogs	0.593	0.261	0.516	0.479	0.558	0.554	-	-	0.857	0.836	0.661	0.621	0.121
Amns	RN50	StanfordDogs	0.593	0.327	0.516	0.465	0.558	0.556	-	-	0.857	0.848	0.661	0.643	0.138
EMMN	RN50	StanfordDogs	0.593	0.0	0.516	0.053	0.558	0.107	-	-	0.857	0.493	0.661	0.107	0.594
Ours	RN50	Caltech101	0.839	0.0	0.857	0.859	0.558	0.56	0.517	0.513	-	-	0.661	0.658	**0.002**
Lip	RN50	Caltech101	0.839	0.081	0.857	0.865	0.558	0.557	0.517	0.52	-	-	0.661	0.657	0.021
Emb	RN50	Caltech101	0.839	0.131	0.857	0.83	0.558	0.546	0.517	0.501	-	-	0.661	0.618	0.061
Amns	RN50	Caltech101	0.838	0.33	0.857	0.834	0.558	0.553	0.517	0.502	-	-	0.661	0.627	0.102
EMMN	RN50	Caltech101	0.839	0.0	0.857	0.397	0.558	0.097	0.517	0.081	-	-	0.661	0.13	0.602
Ours	RN50	OxfordFlowers	0.848	0.0	0.659	0.651	0.558	0.558	0.517	0.515	0.857	0.858	-	-	**0.003**
Lip	RN50	OxfordFlowers	0.848	0.0	0.659	0.645	0.558	0.557	0.517	0.509	0.857	0.868	-	-	0.008
Emb	RN50	OxfordFlowers	0.848	0.442	0.659	0.625	0.558	0.553	0.517	0.5	0.857	0.85	-	-	0.122
Amns	RN50	OxfordFlowers	0.848	0.388	0.659	0.592	0.558	0.54	0.517	0.487	0.857	0.835	-	-	0.135
EMMN	RN50	OxfordFlowers	0.848	0.0	0.659	0.121	0.558	0.121	0.517	0.112	0.857	0.676	-	-	0.519
Ours	ViT-B/16	StanfordCars	0.595	0.0	0.656	0.642	-	-	0.591	0.591	0.933	0.934	0.708	0.703	**0.006**
Lip	ViT-B/16	StanfordCars	0.595	0.159	0.656	0.642	-	-	0.591	0.584	0.933	0.932	0.708	0.707	0.06
Emb	ViT-B/16	StanfordCars	0.595	0.0	0.656	0.557	-	-	0.591	0.508	0.933	0.921	0.708	0.69	0.066
Amns	ViT-B/16	StanfordCars	0.595	0.143	0.656	0.18	-	-	0.591	0.398	0.933	0.876	0.708	0.51	0.327
EMMN	ViT-B/16	StanfordCars	0.595	0.159	0.656	0.182	-	-	0.591	0.119	0.933	0.589	0.708	0.137	0.592
Ours	ViT-B/16	StanfordDogs	0.673	0.0	0.591	0.582	0.655	0.653	-	-	0.933	0.93	0.708	0.697	**0.008**
Lip	ViT-B/16	StanfordDogs	0.673	0.142	0.591	0.592	0.655	0.647	-	-	0.933	0.935	0.708	0.709	0.045
Emb	ViT-B/16	StanfordDogs	0.673	0.071	0.591	0.518	0.655	0.632	-	-	0.933	0.93	0.708	0.699	0.056
Amns	ViT-B/16	StanfordDogs	0.673	0.219	0.591	0.358	0.655	0.59	-	-	0.933	0.901	0.708	0.572	0.209
EMMN	ViT-B/16	StanfordDogs	0.673	0.042	0.591	0.365	0.655	0.284	-	-	0.933	0.826	0.708	0.438	0.301
Ours	ViT-B/16	Caltech101	0.971	0.0	0.933	0.932	0.655	0.653	0.591	0.574	-	-	0.708	0.699	**0.009**
Lip	ViT-B/16	Caltech101	0.971	0.576	0.933	0.935	0.655	0.652	0.591	0.594	-	-	0.708	0.709	0.12
Emb	ViT-B/16	Caltech101	0.971	0.598	0.933	0.91	0.655	0.609	0.591	0.517	-	-	0.708	0.656	0.182
Amns	ViT-B/16	Caltech101	0.971	0.846	0.933	0.848	0.655	0.517	0.591	0.445	-	-	0.708	0.533	0.334
EMMN	ViT-B/16	Caltech101	0.971	0.284	0.933	0.813	0.655	0.352	0.591	0.302	-	-	0.708	0.473	0.341
Ours	ViT-B/16	OxfordFlowers	0.784	0.0	0.707	0.705	0.655	0.654	0.591	0.584	0.933	0.933	-	-	**0.004**
Lip	ViT-B/16	OxfordFlowers	0.784	0.078	0.707	0.702	0.655	0.645	0.591	0.588	0.933	0.933	-	-	0.026
Emb	ViT-B/16	OxfordFlowers	0.784	0.0	0.707	0.617	0.655	0.543	0.591	0.522	0.933	0.906	-	-	0.089
Amns	ViT-B/16	OxfordFlowers	0.784	0.834	0.707	0.527	0.655	0.602	0.591	0.526	0.933	0.913	-	-	0.307
EMMN	ViT-B/16	OxfordFlowers	0.784	0.02	0.707	0.433	0.655	0.317	0.591	0.304	0.933	0.83	-	-	0.305

the class to forget A_{cl} and normalized reduction in the accuracy on the remaining classes for the N examined datasets $A_{\{ds\}}$, we calculate the **Average Score** metrics as:

$$\text{Avg. Score} = \frac{1}{N+1}((1 - A_{cl}) + \sum_{ds} A_{\{ds\}}) \tag{3}$$

Best methods will have a **small** average score. During evaluation we use the standard template *A photo of a {class}*, however in the ablations we evaluate the forget model with other templates to test the robustness to different evaluation templates.

6 Results

6.1 Comparison against other forgetting methods

In Tab.1 we present the aggregated forgetting results across 3 classes for each dataset and across different methods with RN50 and ViT-B/16 visual encoders respectively. Granular results are found in the appendix. *Method* column indicates the forgetting method used, *Dataset* column indicates the dataset from which the

Table 2: Forgetting on multiple classes with RN50 and ViT-B/16 models.

Method	**Model**	**Dataset**	**Classes**	**Avg. Target Classes acc.**		**Other Classes acc.**		**StanfordCars**		**StanfordDogs**		**Caltech101**		**OxfordFlowers**		**Avg. Score (↓)**
				BF	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	
Lip	RN50	StanfordDogs	Pekinese,toy poodle,Scotch terrier	0.591	0.091	0.515	0.507	0.558	0.547	-	-	0.857	0.865	0.661	0.633	0.046
Lip	RN50	StanfordCars	2009 Spyker C8 Coupe, 2010 Dodge Ram Pickup 3500 Crew Cab, 2011 Ford Ranger SuperCab	0.397	0.222	0.56	0.519	-	-	0.517	0.482	0.857	0.84	0.661	0.607	0.16
Lip	RN50	Caltech101	euphonium,minaret,platypus	0.827	0.125	0.858	0.869	0.558	0.549	0.517	0.515	-	-	0.661	0.633	0.042
Lip	RN50	OxfordFlowers	gazania,tree mallow,trumpet creeper	0.86	0.0	0.656	0.609	0.558	0.552	0.517	0.498	0.857	0.863	-	-	0.023
Ours	RN50	StanfordDogs	Pekinese,toy poodle,Scotch terrier	0.591	0.0	0.515	0.499	0.558	0.54	-	-	0.857	0.854	0.661	0.629	0.023
Ours	RN50	StanfordCars	2009 Spyker C8 Coupe, 2010 Dodge Ram Pickup 3500 Crew Cab, 2011 Ford Ranger SuperCab	0.397	0.0	0.56	0.53	-	-	0.517	0.499	0.857	0.85	0.661	0.654	0.021
Ours	RN50	Caltech101	euphonium,minaret,platypus	0.827	0.0	0.858	0.863	0.558	0.551	0.517	0.499	-	-	0.661	0.655	0.011
Ours	RN50	OxfordFlowers	trumpet creeper,gazania,tree mallow	0.86	0.0	0.656	0.627	0.558	0.554	0.517	0.502	0.857	0.856	-	-	0.016
Lip	ViT-B/16	StanfordDogs	Pekinese,toy poodle,Scotch terrier	0.672	0.251	0.589	0.584	0.655	0.644	-	-	0.933	0.939	0.708	0.713	0.08
Lip	ViT-B/16	StanfordCars	2009 Spyker C8 Coupe, 2010 Dodge Ram Pickup 3500 Crew Cab, 2011 Ford Ranger SuperCab	0.595	0.3	0.656	0.625	-	-	0.591	0.576	0.933	0.928	0.708	0.699	0.119
Lip	ViT-B/16	Caltech101	euphonium,minaret,platypus	0.971	0.498	0.932	0.929	0.655	0.634	0.591	0.589	-	-	0.708	0.709	0.11
Lip	ViT-B/16	OxfordFlowers	trumpet creeper,gazania,tree mallow	0.807	0.31	0.705	0.68	0.655	0.613	0.591	0.551	0.933	0.929	-	-	0.111
Ours	ViT-B/16	StanfordDogs	Pekinese,toy poodle,Scotch terrier	0.672	0.0	0.589	0.557	0.655	0.624	-	-	0.933	0.92	0.708	0.668	0.035
Ours	ViT-B/16	StanfordCars	2009 Spyker C8 Coupe, 2010 Dodge Ram Pickup 3500 Crew Cab, 2011 Ford Ranger SuperCab	0.595	0.0	0.656	0.633	-	-	0.591	0.586	0.933	0.931	0.708	0.693	0.013
Ours	ViT-B/16	Caltech101	euphonium,minaret,platypus	0.962	0.0	0.932	0.929	0.655	0.65	0.591	0.558	-	-	0.708	0.685	0.02
Ours	ViT-B/16	OxfordFlowers	trumpet creeper,gazania,tree mallow	0.807	0.0	0.705	0.682	0.655	0.649	0.591	0.578	0.933	0.929	-	-	0.014

class to be forgotten was picked, *Avg. Target Class acc.* denotes the average accuracy on the target class while *Avg. Other Classes acc.* the avergae accuracy on other classes from the dataset of the forget class before (BF) and after (AF) forgetting. Finally, the final eight columns represent the results on the remaining datasets reported both before and after forgetting.
For both the models we observe superiority in terms of the average score of our method that is able to achieve a good balance between forgetting on the target class and retaining information about not forget classes. Specifically, our method is able to remove completely the information about the forget classes from the model while other methods, apart from EMNN that however overforgets other classes, usually retain some information. *Lip* is the most competitive with our method that sometimes achieves better accuracy on the other classes retaining however some information about the forget class. Furthermore, comparing to *Lip*, which often struggles to forget well with a ViT visual encoder, our method is more robust and forgets well independently on the visual encoder used. Full results can be found in the appendix.

6.2 Forgetting on Multiple Classes

In Tab. 2 we show the results for *Lip* and *Our* methods when performing forgetting on multiple classes for RN50 and ViT-B/16 visual encoders respectively. Our method shows its superiority in terms of the average score also in this case. Again, our method is able to completely forget all the targeted classes while still maintaining high accuracy across not forget classes while *Lip*, especially for with the ViT-B/16 visual encoder retains substantial information on the forget classes. Indeed, *Lip* is less consistent across different architectures while our method is able to maintain this consistency both in case of single class and multiple classes forgetting.

6.3 Understanding the Unlearning

Thanks to the simplicity of our method that only modifies one matrix we can closely examine what happens during the unlearning process. Specifically, we analyze which neurons in the projection matrix undergo the most significant changes following the unlearning procedure. This is done by looking at the absolute value of the AB^T matrix that represents the changes in the text projection matrix. Recall that all the projection matrix does is projecting the hidden textual representation into the text-image shared embedding space: from 512 into 1024 dimensions for CLIP with ResNet50 visual encoder and from 512 into 512 dimensions with ViT-B/16. We observe that there are some "magic" neurons that the algorithm modifies more indicating that these textual features need to change the most to forget a class while preserving the other classes. For example, for ResNet50 such neurons are in column 222 of the the weight projection change matrix $(AB^T)_{[512,1024]}$ as can be seen in Fig. 2 that shows on the x axis the column where most change occurred and on the y axis the sum of absolute values of the changes in that column.

It turns out this is not a random selection; plotting textual features across different classes and datasets, shown in Fig. 3, reveals that feature 222, which the unlearning algorithm targets, has the largest value. Thus, changing this feature is the easiest way for the model to unlearn a class, decreasing the dot product between the visual and textual features for that class. The corresponding visual feature 222 is also negative across images, so the network increases the value of the textual feature 222, decreasing the dot product for that class causing a change in model's prediction. The presence of such neurons and the algorithm's targeted modification is intriguing. A similar phenomenon occurs with the ViT-B/16 as seen in the same figure.

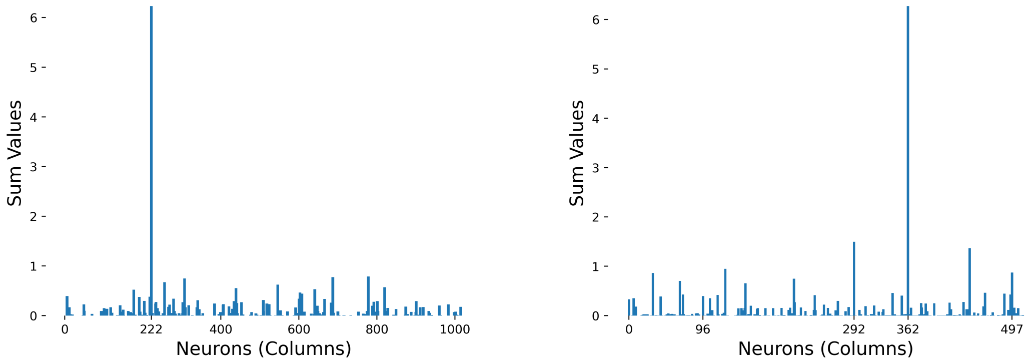


Figure 2: Sum of absolute values of neurons in different columns of the textual projection matrix change. On the **left** for RN50 visual encoder, on the **right** for ViT-B/16 visual encoder.

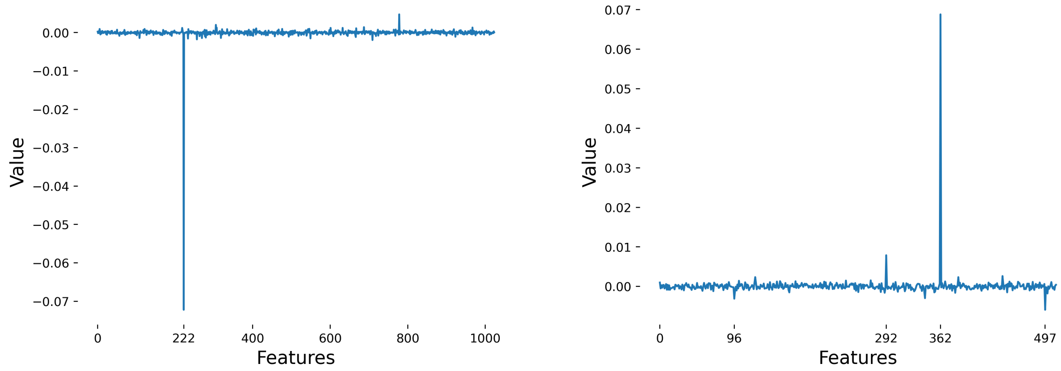


Figure 3: Textual features values averaged across different classes. On the **left** for RN50 visual encoder, on the **right** for ViT-B/16 visual encoder.

We do a similar analysis for Lipschitz forgetting, where tracking changes at different weight levels becomes challenging due to the modification of many layers. However, we can observe the alterations in final visual

Table 3: Aggregated results across different evaluation templates. We aggregate across 3 evaluations template to assess sensitivity of the models after forgetting to the change in the evaluation template.

Model	**Dataset**	**Avg. Target Class acc.**		**Avg. Other Classes acc.**		**Avg. StanfordCars**		**Avg. StanfordDogs**		**Avg. Caltech101**		**Avg. OxfordFlowers**		**Avg. Score** (↓)
		BF	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	
RN50	StanfordCars	0.272	0.0	0.493	0.488	-	-	0.415	0.415	0.81	0.811	0.519	0.518	0.003
RN50	StanfordDogs	0.306	0.0	0.416	0.405	0.492	0.493	-	-	0.81	0.809	0.519	0.518	0.007
RN50	Caltech101	0.879	0.029	0.81	0.81	0.492	0.488	0.415	0.415	-	-	0.519	0.517	0.01
RN50	OxfordFlowers	0.698	0.0	0.518	0.513	0.492	0.491	0.415	0.415	0.81	0.81	-	-	0.004
ViT-B/16	StanfordCars	0.497	0.0	0.623	0.618	-	-	0.516	0.514	0.88	0.882	0.61	0.61	0.003
ViT-B/16	StanfordDogs	0.532	0.0	0.516	0.504	0.622	0.617	-	-	0.88	0.88	0.61	0.607	0.008
ViT-B/16	Caltech101	0.97	0.011	0.879	0.88	0.622	0.621	0.516	0.513	-	-	0.61	0.61	0.004
ViT-B/16	OxfordFlowers	0.667	0.0	0.609	0.604	0.622	0.62	0.516	0.513	0.88	0.88	-	-	0.004

and textual features. Interestingly, when examining features that change the most with Lipschitz forgetting, we observe the same pattern as with our method for the textual features. In contrast, this behavior is not seen for visual features where different features undergo more significant changes for different images.

7 Ablations & Additional Tasks

7.1 Variation of Templates for Evaluation

In these experiments we test how sensitive the model after forgetting is to the evaluation template and whether when changing it the model is still able to retrieve the forget class. Following Kravets & Namboodiri (2024) we evaluate using the following three templates: *"We can see a {class} in this image"*, *"This is a representation of {class}"*, *"There is evidence of a {class} in the picture"*. Note that by changing the evaluation template also the accuracy of zero-shot CLIP changes.

In Tab. 3 we observe that forgetting is robust to the change in the evaluation template as the model is still unable to retrieve the forget class and maintains a high accuracy of the not forget classes relatively to the model before forgetting.

7.2 Loss Components Ablation

In this subsection we assess how important are the loss components to the forgetting procedure. For this, we first set λ_1 to 0 and then λ_3 to 0. Results are shown in Tab. 4 where we observe that both components are important for forgetting that achieve the best results in terms of the average score when all the components are included. The average score drops more when λ_1 is excluded from the loss.

Table 4: Ablations on loss components.

Method	**Model**	**Avg. Target Class acc.**		**Avg. Other Classes acc.**		**Avg. StanfordCars**		**Avg. StanfordDogs**		**Avg. Caltech101**		**Avg. OxfordFlowers**		**Avg. Score** (↓)
		BF	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	
All loss terms	RN50	0.669	0.0	0.648	0.642	0.558	0.557	0.517	0.513	0.857	0.857	0.661	0.656	**0.005**
Excluding λ_1	RN50	0.669	0.01	0.648	0.625	0.558	0.558	0.517	0.511	0.857	0.863	0.661	0.644	0.017
Excluding λ_3	RN50	0.669	0.004	0.648	0.641	0.558	0.553	0.517	0.509	0.857	0.855	0.661	0.655	0.01
All loss terms	ViT-B/16	0.756	0.0	0.722	0.715	0.655	0.653	0.591	0.583	0.933	0.932	0.708	0.7	**0.008**
Excluding λ_1	ViT-B/16	0.756	0.185	0.722	0.694	0.655	0.653	0.591	0.587	0.933	0.935	0.708	0.702	0.061
Excluding λ_3	ViT-B/16	0.756	0.001	0.722	0.711	0.655	0.636	0.591	0.578	0.933	0.928	0.708	0.694	0.019

7.3 Number of Classes to Retain

In this section we evaluate how varying the number of classes to preserve affects the forgetting results. In Tab. 5 we observe that reducing the number of classes to preserve the average score drops, but the results are still relatively robust even when only 10% classes to retain are used - the most sensitive to the reduction in retain classes is the dataset the forget class was picked from. All the generated retain classes can be found in the appendix, and on average we generate 100 semantically similar classes to the forget class for each dataset.

Table 5: Ablations on the number of classes to retain.

% Number of Classes	Model	Avg. Target Class acc.		Avg. Other Classes acc.		Avg. StanfordCars		Avg. StanfordDogs		Avg. Caltech101		Avg. OxfordFlowers		Avg. Score (↓)
		BF	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	
All	RN50	0.669	0.0	0.648	0.642	0.558	0.556	0.517	0.509	0.857	0.856	0.661	0.655	**0.007**
50%	RN50	0.669	0.0	0.648	0.636	0.558	0.554	0.517	0.51	0.857	0.856	0.661	0.653	0.01
10%	RN50	0.669	0.0	0.648	0.623	0.558	0.558	0.517	0.51	0.857	0.856	0.661	0.653	0.013
All	ViT-B/16	0.756	0.0	0.722	0.712	0.655	0.646	0.591	0.576	0.933	0.93	0.708	0.695	**0.015**
50%	ViT-B/16	0.756	0.0	0.722	0.704	0.655	0.65	0.591	0.576	0.933	0.931	0.708	0.698	0.015
10%	ViT-B/16	0.756	0.0	0.722	0.688	0.655	0.652	0.591	0.581	0.933	0.932	0.708	0.7	0.016

7.4 Retaining Classes from Forget Class Dataset

In our main experiments we used semantically similar classes (*SemSim*) to the forget class for the retain loss component. In Tab. 6 we compare the effects of using actual classes from the dataset the forget class was picked from, denoted as Cls_r, and when using semantically different classes (*SemDiff*). Overall, we find that semantically similar classes are crucial for maintaining high *Other Classes acc.*. When forgetting, the original projection matrix is altered in a way that perturbs the space near the forget class more leading to a greater reduction in accuracy for semantically similar classes which are closer in the image-text embedding space compared to different classes, where the space is less affected. Using actual classes (Cls_r) performs the best, but similarly to semantically similar classes generated by a large language model. In contrast, using semantically different classes, taken from the Food101 Bossard et al. (2014) dataset, results in the worst outcomes, especially for *Other Classes acc.* while the accuracy of the classes not semantically similar to the forget class (i.e. other test datasetes) is maintained without explicitly including them.

Table 6: Ablations with actual (Cls_r), semantically similar (*SemSim*) and different (*SemDiff*) classes

Type of Retained Classes	Model	Avg. Target Class acc.		Avg. Other Classes acc.		Avg. StanfordCars		Avg. StanfordDogs		Avg. Caltech101		Avg. OxfordFlowers		Avg. Score (↓)
		BF	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	
Cls_r	RN50	0.669	0.0	0.648	0.644	0.558	0.558	0.517	0.51	0.857	0.856	0.661	0.659	**0.004**
SemSim	RN50	0.669	0.0	0.648	0.642	0.558	0.557	0.517	0.513	0.857	0.857	0.661	0.656	0.005
SemDiff	RN50	0.669	0.0	0.648	0.61	0.558	0.558	0.517	0.511	0.857	0.86	0.661	0.655	0.016
Cls_r	ViT-B/16	0.756	0.0	0.722	0.718	0.655	0.652	0.591	0.583	0.933	0.931	0.708	0.702	**0.006**
SemSim	ViT-B/16	0.756	0.0	0.722	0.715	0.655	0.653	0.591	0.583	0.933	0.932	0.708	0.7	0.008
SemDiff	ViT-B/16	0.756	0.004	0.722	0.68	0.655	0.651	0.591	0.584	0.933	0.934	0.708	0.703	0.018

7.5 Forget Class Projection

We evaluate the importance of where to project the forget classes in the shared image-text space. We test different variations like projecting into a random vector and a perturbed embedding of the forget concepts comparing them to the empty token projection used in our main experiments. In Tab 7 we observe that projection space is less important as similar results are achieved when we project into different parts of the space.

7.6 Retrieval Task

We additionally evaluate CLIP on the retrieval task after unlearning. Following Kravets & Namboodiri (2024), we evaluate retrieval of image from text input. We evaluate retrieval creating a database from the four datasets we used in our main experiments. We use precision@k metric for k of 1, 5 and 10, which measures the proportion of relevant items among the top K retrieved results. Lower precision@k indicates better performance. These results are displayed in Tab 8 where we compare the *original*, *Lip* and our method aggregating across all the classes and datasets. Our method achieves best performance also on the retrieval task. Full results in the Appendix C.

Table 7: Ablations on projection. We perform an ablation study projecting into the empty token vector (*EmptyToken proj*), random vector sampled from Gussiaan distribution (*Random proj*) and perturbed embedding of the forget class (*Perturbed proj*)

Method	**Model**	**Avg. Target Class acc.**		**Avg. Other Classes acc.**		**Avg. StanfordCars**		**Avg. StanfordDogs**		**Avg. Caltech101**		**Avg. OxfordFlowers**		**Avg. Score (↓)**
		BF	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	
EmptyToken proj	RN50	0.669	0.0	0.648	0.642	0.558	0.557	0.517	0.513	0.857	0.857	0.661	0.656	**0.005**
Random proj	RN50	0.669	0.0	0.648	0.642	0.558	0.557	0.517	0.511	0.857	0.856	0.661	0.655	0.006
Perturbed proj	RN50	0.669	0.0	0.648	0.642	0.558	0.557	0.517	0.511	0.857	0.856	0.661	0.655	0.006
EmptyToken proj	ViT-B/16	0.756	0.0	0.722	0.715	0.558	0.653	0.517	0.583	0.857	0.932	0.661	0.7	**0.008**
Random proj	ViT-B/16	0.756	0.0	0.722	0.713	0.558	0.652	0.517	0.58	0.857	0.931	0.661	0.698	0.01
Perturbed proj	ViT-B/16	0.756	0.0	0.722	0.713	0.558	0.652	0.517	0.58	0.857	0.931	0.661	0.698	0.01

Table 8: Aggregated across all datasets and classes image retrieval from text input results showing precision@k for k of 1, 5 and 10 with RN50 and ViT-B/16 visual encoders.

Model	**Precision@1 (↓)**	**Precision@5 (↓)**	**Precision@10 (↓)**
RN50 (original)	0.833	0.683	0.583
RN50 (Lip)	0.08	0.23	0.191
RN50 (Ours)	**0.0**	**0.017**	**0.008**
ViT-B/16 (original)	0.833	0.717	0.667
ViT-B/16 (Lip)	0.5	0.433	0.4
ViT-B/16 (Ours)	**0.0**	**0.0**	**0.0**

7.7 Unlearning is Specific

The datasets in our main experiments include semantically similar classes that often share words. Our procedure effectively breaks the exact textual and visual association for the forget class but doesn't break the association for similar words to the forget class as evidenced by the high accuracy on other classes from the same dataset that are often similar in meaning to the forget class. For instance, when removing the class information for *toy poodle*, classes *miniature poodle* and *standard poodle*, which share the word *poodle*, maintain accuracy comparable to that before unlearning. To eliminate synonyms and similar words, the forgetting procedure would need to be repeated for those terms. We see this as a feature rather than a limitation, as it enables the preservation of as much information as possible during the forgetting process. This approach provides precise control over which information to forget, including synonyms and similar words if necessary.

We also show in the Appendix F that the new classes predicted by the unlearned model are close to the correct ones, which is a further indication that our method targets specific knowledge of the model while preserving its general understanding.

8 Conclusions

In this work we demonstrated that it is possible to unlearn a class in the CLIP model without altering the original visual encoder, thereby eliminating the need to generate synthetic data. A learned adaptation to the projection matrix of the textual encoder, which projects textual representations into the image-text embedding space, is sufficient for class unlearning. Furthermore, we show that the representation of semantically similar classes can be affected during unlearning, reducing their accuracy. Therefore, it is crucial to include semantically similar classes that are close in the embedding space in the loss function to retain their information.

References

Syed Hammad Ahmed, Shengnan Hu, and Gita Sukthankar. The potential of vision-language models for content moderation of children's videos, 2023. URL `https://arxiv.org/abs/2312.03936`.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *ICML*, 03 2013. URL `https://arxiv.org/abs/1206.6389`.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. *Computer Vision – ECCV 2014*, pp. 446–461, 2014. doi: 10.1007/978-3-319-10599-4_29.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning, 05 2015. URL `https://ieeexplore.ieee.org/document/7163042`.

Kongyang Chen, Yao Huang, and Yiwen Wang. Machine unlearning via gan, 11 2021. URL `https://arxiv.org/abs/2111.11869`.

S. Chen, C. GE, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 16664–16678, 2022.

Jiali Cheng and Hadi Amiri. Multimodal machine unlearning, 11 2023. URL `https://arxiv.org/abs/2311.12047`.

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Trans. Info. Forensics and Security*, 18:2345–2354, 2023. doi: 10.1109/TIFS.2023.3265506. URL `https://arxiv.org/abs/2201.05629`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CVPR*, 10 2020.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106:59–70, 04 2007. doi: 10.1016/j.cviu.2005.09.012.

Daniel L. Felps, Amelia D. Schwickerath, Joyce D. Williams, Trung N. Vuong, Alan Briggs, Matthew Hunt, Evan Sakmar, David D. Saranchak, and Tyler Shumaker. Class clown: Data redaction in machine unlearning at enterprise scale, 12 2020. URL `http://arxiv.org/abs/2012.04699v1`.

Jack Foster, Kyle Fogarty, Stefan Schoepf, Cengiz Öztireli, and Alexandra Brintrup. Zero-shot machine unlearning at scale via lipschitz regularization. 02 2024. doi: 10.48550/arXiv.2402.01401. URL `https://arxiv.org/abs/2402.01401`.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models, 06 2023a. URL `https://arxiv.org/abs/2303.07345`.

Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. *arXiv preprint arXiv:2308.14761*, 2023b.

Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks, 06 2021. URL `https://arxiv.org/abs/2012.13431`.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning, 10 2020. URL `https://arxiv.org/abs/2010.10981`.

Chuan Guo, Tom Goldstein, Awni Hannun, and van . Certified data removal from machine learning models. *arXiv (Cornell University)*, 11 2019. doi: 10.48550/arxiv.1911.03030.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 12 2015. URL `https://arxiv.org/abs/1512.03385`.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2790–2799, 09–15 Jun 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv:2106.09685 [cs]*, 10 2021. URL `https://arxiv.org/abs/2106.09685`.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops*, ICCVW '13, pp. 554–561, 2013. ISBN 9781479930227.

A. Kravets and V. Namboodiri. Zero-shot class unlearning in clip with synthetic samples. 2024.

Jiaqi Li, Qianshan Wei, Chuanyi Zhang, Guilin Qi, Miaozeng Du, Yongrui Chen, and Sheng Bi. Single image unlearning: Efficient machine unlearning in multimodal large language models, 2024a. URL `https://arxiv.org/abs/2405.12523`.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, and Ann-Kathrin Dombrowski. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024b.

M. Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, 2008. URL `https://www.semanticscholar.org/paper/Automated-Flower-Classification-over-a-Large-Number-Nilsback-Zisserman/02b28f3b71138a06e40dbd614abf8568420ae183`.

Yossef Oren and Angelos D. Keromytis. Attacking the internet using broadcast digital television. *ACM Transactions on Information and System Security*, 17:1–27, 04 2015. doi: 10.1145/2723159. URL `http://www.cs.columbia.edu/~angelos/Papers/2014/redbutton-usenix-sec14.pdf`.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 487–503, April 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763. PMLR, 18–24 Jul 2021.

Aneeshan Sain, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. Clip for all things zero-shot sketch-based image retrieval, fine-grained or not, 2023. URL `https://arxiv.org/abs/2303.13440`.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning, 07 2021. URL `https://arxiv.org/abs/2103.03279`.

Takashi Shibata and Yu Mitsuzumi. Learning with selective forgetting, 2021. URL `https://www.ijcai.org/proceedings/2021/0137.pdf`.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

Solawetz. Object tracking with clip, 2021. URL `https://blog.roboflow.com/zero-shot-object-tracking`.

Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Trans. Neural Net. and Learn. Systems*, 07 2022. URL `https://arxiv.org/abs/2111.08947`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL `https://arxiv.org/abs/1706.03762`.

Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning, 01 2022. URL `https://arxiv.org/abs/2110.11794`.

Somesh Jha Jeffrey F. Naughton Xi Wu, Matthew Fredrikson. A methodology for formalizing model-inversion attacks | ieee conference publication | ieee xplore, 2016. URL `https://ieeexplore.ieee.org/document/7536387`.

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1), aug 2023. ISSN 0360-0300. doi: 10.1145/3603620. URL `https://doi.org/10.1145/3603620`.

Eric Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. 03 2023. doi: 10.48550/arXiv.2303.17591. URL `https://arxiv.org/abs/2303.17591`.

A ResNet Full Results

Table 9: Forgetting results with RN50 visual encoder. We compare our methods with four others on three classes for four selected datasets.

Method	Dataset	Class name	Target Class acc.		Other Classes acc.		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
			BF	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**
Ours	StanfordDogs	Pekinese	0.705	0.0	0.515	0.51	0.558	0.559	-	-	0.857	0.853	0.661	0.659
Ours	StanfordDogs	toy poodle	0.574	0.0	0.516	0.507	0.558	0.56	-	-	0.857	0.857	0.661	0.644
Ours	StanfordDogs	Scotch terrier	0.5	0.0	0.517	0.509	0.558	0.543	-	-	0.857	0.859	0.661	0.656
Ours	StanfordCars	2009 Spyker C8 Coupe	0.262	0.0	0.559	0.56	-	-	0.517	0.509	0.857	0.858	0.661	0.658
Ours	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.0	0.558	0.542	-	-	0.517	0.512	0.857	0.851	0.661	0.654
Ours	StanfordCars	2011 Ford Ranger SuperCab	0.524	0.0	0.558	0.549	-	-	0.517	0.509	0.857	0.856	0.661	0.658
Ours	Caltech101	euphonium	0.789	0.0	0.858	0.861	0.558	0.561	0.517	0.512	-	-	0.661	0.66
Ours	Caltech101	minaret	0.826	0.0	0.857	0.857	0.558	0.557	0.517	0.519	-	-	0.661	0.653
Ours	Caltech101	platypus	0.9	0.0	0.857	0.86	0.558	0.56	0.517	0.507	-	-	0.661	0.661
Ours	OxfordFlowers	gazania	0.957	0.0	0.658	0.646	0.558	0.555	0.517	0.514	0.857	0.859	-	-
Ours	OxfordFlowers	tree mallow	1.0	0.0	0.658	0.646	0.558	0.56	0.517	0.514	0.857	0.856	-	-
Ours	OxfordFlowers	trumpet creeper	0.588	0.0	0.661	0.661	0.558	0.56	0.517	0.516	0.857	0.861	-	-
Lip	StanfordDogs	Pekinese	0.705	0.066	0.515	0.514	0.655	0.559	-	-	0.933	0.867	0.708	0.658
Lip	StanfordDogs	toy poodle	0.574	0.033	0.516	0.518	0.655	0.559	-	-	0.933	0.867	0.708	0.647
Lip	StanfordDogs	Scotch terrier	0.5	0.047	0.517	0.516	0.655	0.557	-	-	0.933	0.865	0.708	0.66
Lip	StanfordCars	2009 Spyker C8 Coupe	0.262	0.024	0.559	0.553	-	-	0.591	0.518	0.933	0.865	0.708	0.66
Lip	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.143	0.558	0.544	-	-	0.591	0.502	0.933	0.845	0.708	0.638
Lip	StanfordCars	2011 Ford Ranger SuperCab	0.524	0.0	0.558	0.555	-	-	0.591	0.52	0.933	0.869	0.708	0.661
Lip	Caltech101	euphonium	0.789	0.0	0.858	0.868	0.655	0.557	0.591	0.52	-	-	0.708	0.658
Lip	Caltech101	minaret	0.826	0.043	0.857	0.863	0.655	0.556	0.591	0.515	-	-	0.708	0.661
Lip	Caltech101	platypus	0.9	0.2	0.857	0.866	0.655	0.558	0.591	0.524	-	-	0.708	0.653
Lip	OxfordFlowers	gazania	0.957	0.0	0.658	0.649	0.655	0.559	0.591	0.513	0.933	0.869	-	-
Lip	OxfordFlowers	tree mallow	1.0	0.0	0.658	0.643	0.655	0.557	0.591	0.51	0.933	0.869	-	-
Lip	OxfordFlowers	trumpet creeper	0.588	0.0	0.661	0.643	0.655	0.557	0.591	0.503	0.933	0.866	-	-
Emb	StanfordDogs	Pekinese	0.705	0.361	0.515	0.484	0.558	0.559	-	-	0.857	0.84	0.661	0.633
Emb	StanfordDogs	toy poodle	0.574	0.361	0.516	0.481	0.558	0.553	-	-	0.857	0.832	0.661	0.613
Emb	StanfordDogs	Scotch terrier	0.5	0.062	0.517	0.472	0.558	0.551	-	-	0.857	0.837	0.661	0.617
Emb	StanfordCars	2009 Spyker C8 Coupe	0.262	0.024	0.559	0.529	-	-	0.517	0.508	0.857	0.841	0.661	0.639
Emb	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.119	0.558	0.542	-	-	0.517	0.512	0.857	0.857	0.661	0.654
Emb	StanfordCars	2011 Ford Ranger SuperCab	0.524	0.119	0.558	0.539	-	-	0.517	0.509	0.857	0.852	0.661	0.654
Emb	Caltech101	euphonium	0.789	0.263	0.858	0.833	0.558	0.548	0.517	0.506	-	-	0.661	0.616
Emb	Caltech101	minaret	0.826	0.13	0.857	0.827	0.558	0.54	0.517	0.507	-	-	0.661	0.639
Emb	Caltech101	platypus	0.9	0.0	0.857	0.829	0.558	0.549	0.517	0.49	-	-	0.661	0.597
Emb	OxfordFlowers	gazania	0.957	0.739	0.658	0.632	0.558	0.551	0.517	0.503	0.857	0.849	-	-
Emb	OxfordFlowers	tree mallow	1.0	0.353	0.658	0.612	0.558	0.554	0.517	0.504	0.857	0.849	-	-
Emb	OxfordFlowers	trumpet creeper	0.588	0.235	0.661	0.632	0.558	0.555	0.517	0.508	0.857	0.853	-	-
Amns	StanfordDogs	Pekinese	0.705	0.459	0.515	0.486	0.558	0.561	-	-	0.857	0.847	0.661	0.65
Amns	StanfordDogs	toy poodle	0.574	0.492	0.516	0.423	0.558	0.55	-	-	0.857	0.839	0.661	0.628
Amns	StanfordDogs	Scotch terrier	0.5	0.031	0.517	0.488	0.558	0.559	-	-	0.857	0.859	0.661	0.651
Amns	StanfordCars	2009 Spyker C8 Coupe	0.262	0.143	0.559	0.516	-	-	0.517	0.51	0.857	0.854	0.661	0.646
Amns	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.429	0.558	0.49	-	-	0.517	0.5	0.857	0.868	0.661	0.658
Amns	StanfordCars	2011 Ford Ranger SuperCab	0.524	0.5	0.558	0.489	-	-	0.517	0.507	0.857	0.868	0.661	0.656
Amns	Caltech101	euphonium	0.789	0.316	0.858	0.856	0.558	0.557	0.517	0.519	-	-	0.661	0.655
Amns	Caltech101	platypus	0.9	0.5	0.857	0.832	0.558	0.555	0.517	0.495	-	-	0.661	0.634
Amns	Caltech101	minaret	0.826	0.174	0.857	0.813	0.558	0.546	0.517	0.493	-	-	0.661	0.591
Amns	OxfordFlowers	gazania	0.957	0.87	0.658	0.595	0.558	0.557	0.517	0.489	0.857	0.834	-	-
Amns	OxfordFlowers	tree mallow	1.0	0.0	0.658	0.598	0.558	0.511	0.517	0.476	0.857	0.843	-	-
Amns	OxfordFlowers	trumpet creeper	0.588	0.294	0.661	0.584	0.558	0.554	0.517	0.494	0.857	0.828	-	-
EMMN	StanfordDogs	Pekinese	0.787	0.0	0.59	0.376	0.655	0.278	-	-	0.933	0.828	0.708	0.432
EMMN	StanfordDogs	toy poodle	0.607	0.0	0.591	0.373	0.655	0.308	-	-	0.933	0.836	0.708	0.446
EMMN	StanfordDogs	Scotch terrier	0.625	0.125	0.591	0.347	0.655	0.265	-	-	0.933	0.813	0.708	0.436
EMMN	StanfordCars	2009 Spyker C8 Coupe	0.429	0.0	0.656	0.188	-	-	0.591	0.116	0.933	0.614	0.708	0.148
EMMN	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548	0.476	0.656	0.184	-	-	0.591	0.13	0.933	0.56	0.708	0.126
EMMN	StanfordCars	2011 Ford Ranger SuperCab	0.81	0.0	0.654	0.175	-	-	0.591	0.111	0.933	0.594	0.708	0.136
EMMN	Caltech101	euphonium	1.0	0.105	0.933	0.783	0.655	0.352	0.591	0.297	-	-	0.708	0.45
EMMN	Caltech101	minaret	0.913	0.348	0.933	0.817	0.655	0.36	0.591	0.315	-	-	0.708	0.485
EMMN	Caltech101	platypus	1.0	0.4	0.933	0.838	0.655	0.345	0.591	0.294	-	-	0.708	0.484
EMMN	OxfordFlowers	gazania	1.0	0.0	0.705	0.44	0.655	0.308	0.591	0.312	0.933	0.832	-	-
EMMN	OxfordFlowers	tree mallow	0.765	0.0	0.707	0.445	0.655	0.33	0.591	0.288	0.933	0.829	-	-
EMMN	OxfordFlowers	trumpet creeper	0.588	0.059	0.709	0.413	0.655	0.312	0.591	0.31	0.933	0.828	-	-

B ViT-B/16 Full Results

Table 10: Forgetting results with ViT-B/16 visual encoder. We compare our methods with four others on three classes for four selected datasets.

Method	**Dataset**	**Class name**	**Target Class acc.**		**Other Classes acc.**		**StanfordCars**		**StanfordDogs**		**Caltech101**		**OxfordFlowers**	
			BF	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**	**BF**	**AF**
Ours	StanfordDogs	Pekinese	0.787	0.0	0.59	0.586	0.655	0.653	-	-	0.933	0.933	0.708	0.692
Ours	StanfordDogs	toy poodle	0.607	0.0	0.591	0.581	0.655	0.651	-	-	0.933	0.932	0.708	0.7
Ours	StanfordDogs	Scotch terrier	0.625	0.0	0.591	0.58	0.655	0.654	-	-	0.933	0.924	0.708	0.698
Ours	StanfordCars	2009 Spyker C8 Coupe	0.429	0.0	0.656	0.643	-	-	0.591	0.592	0.933	0.935	0.708	0.701
Ours	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548	0.0	0.656	0.646	-	-	0.591	0.591	0.933	0.932	0.708	0.703
Ours	StanfordCars	2011 Ford Ranger SuperCab	0.81	0.0	0.654	0.639	-	-	0.591	0.589	0.933	0.934	0.708	0.703
Ours	Caltech101	euphonium	1.0	0.0	0.933	0.93	0.655	0.651	0.591	0.56	-	-	0.708	0.692
Ours	Caltech101	minaret	0.913	0.0	0.933	0.934	0.655	0.654	0.591	0.588	-	-	0.708	0.705
Ours	Caltech101	platypus	1.0	0.0	0.933	0.932	0.655	0.654	0.591	0.573	-	-	0.708	0.701
Ours	OxfordFlowers	gazania	1.0	0.0	0.705	0.702	0.655	0.652	0.591	0.583	0.933	0.932	-	-
Ours	OxfordFlowers	tree mallow	0.765	0.0	0.707	0.703	0.655	0.653	0.591	0.58	0.933	0.933	-	-
Ours	OxfordFlowers	trumpet creeper	0.588	0.0	0.709	0.709	0.655	0.656	0.591	0.59	0.933	0.933	-	-
Lip	StanfordDogs	Pekinese	0.787	0.377	0.59	0.601	0.655	0.656	-	-	0.933	0.934	0.708	0.708
Lip	StanfordDogs	toy poodle	0.607	0.033	0.591	0.593	0.655	0.639	-	-	0.933	0.932	0.708	0.707
Lip	StanfordDogs	Scotch terrier	0.625	0.016	0.591	0.582	0.655	0.647	-	-	0.933	0.938	0.708	0.713
Lip	StanfordCars	2009 Spyker C8 Coupe	0.429	0.262	0.656	0.639	-	-	0.591	0.581	0.933	0.93	0.708	0.7
Lip	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548	0.048	0.656	0.634	-	-	0.591	0.58	0.933	0.933	0.708	0.708
Lip	StanfordCars	2011 Ford Ranger SuperCab	0.81	0.167	0.654	0.653	-	-	0.591	0.59	0.933	0.933	0.708	0.713
Lip	Caltech101	euphonium	1.0	0.158	0.933	0.935	0.655	0.653	0.591	0.597	-	-	0.708	0.706
Lip	Caltech101	minaret	0.913	0.87	0.933	0.932	0.655	0.649	0.591	0.59	-	-	0.708	0.709
Lip	Caltech101	platypus	1.0	0.7	0.933	0.936	0.655	0.653	0.591	0.595	-	-	0.708	0.711
Lip	OxfordFlowers	gazania	1.0	0.0	0.705	0.7	0.655	0.642	0.591	0.587	0.933	0.935	-	-
Lip	OxfordFlowers	tree mallow	0.765	0.176	0.707	0.699	0.655	0.65	0.591	0.596	0.933	0.933	-	-
Lip	OxfordFlowers	trumpet creeper	0.588	0.059	0.709	0.705	0.655	0.644	0.591	0.581	0.933	0.932	-	-
Emb	StanfordDogs	Pekinese	0.787	0.213	0.59	0.601	0.655	0.656	-	-	0.933	0.934	0.708	0.708
Emb	StanfordDogs	toy poodle	0.607	0.0	0.591	0.472	0.655	0.621	-	-	0.933	0.931	0.708	0.696
Emb	StanfordDogs	Scotch terrier	0.625	0.0	0.591	0.481	0.655	0.617	-	-	0.933	0.926	0.708	0.695
Emb	StanfordCars	2009 Spyker C8 Coupe	0.429	0.0	0.656	0.479	-	-	0.591	0.392	0.933	0.908	0.708	0.659
Emb	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548	0.0	0.656	0.626	-	-	0.591	0.59	0.933	0.934	0.708	0.713
Emb	StanfordCars	2011 Ford Ranger SuperCab	0.81	0.0	0.654	0.565	-	-	0.591	0.542	0.933	0.92	0.708	0.699
Emb	Caltech101	euphonium	1.0	0.368	0.933	0.935	0.655	0.652	0.591	0.594	-	-	0.708	0.709
Emb	Caltech101	minaret	0.913	0.826	0.933	0.933	0.655	0.635	0.591	0.583	-	-	0.708	0.711
Emb	Caltech101	platypus	1.0	0.6	0.933	0.861	0.655	0.539	0.591	0.376	-	-	0.708	0.547
Emb	OxfordFlowers	gazania	1.0	0.0	0.705	0.705	0.655	0.645	0.591	0.593	0.933	0.933	-	-
Emb	OxfordFlowers	tree mallow	0.765	0.0	0.707	0.577	0.655	0.58	0.591	0.501	0.933	0.903	-	-
Emb	OxfordFlowers	trumpet creeper	0.588	0.0	0.709	0.569	0.655	0.406	0.591	0.472	0.933	0.88	-	-
Amns	StanfordDogs	Pekinese	0.787	0.623	0.59	0.366	0.655	0.581	-	-	0.933	0.896	0.708	0.609
Amns	StanfordDogs	toy poodle	0.607	0.033	0.591	0.234	0.655	0.57	-	-	0.933	0.899	0.708	0.482
Amns	StanfordDogs	Scotch terrier	0.625	0.0	0.591	0.473	0.655	0.618	-	-	0.933	0.908	0.708	0.626
Amns	StanfordCars	2009 Spyker C8 Coupe	0.429	0.0	0.656	0.058	-	-	0.591	0.242	0.933	0.808	0.708	0.361
Amns	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548	0.214	0.656	0.166	-	-	0.591	0.436	0.933	0.904	0.708	0.572
Amns	StanfordCars	2011 Ford Ranger SuperCab	0.81	0.214	0.654	0.315	-	-	0.591	0.516	0.933	0.916	0.708	0.596
Amns	Caltech101	euphonium	1.0	1.0	0.933	0.901	0.655	0.648	0.591	0.57	-	-	0.708	0.639
Amns	Caltech101	minaret	0.913	0.739	0.933	0.774	0.655	0.336	0.591	0.257	-	-	0.708	0.366
Amns	Caltech101	platypus	1.0	0.8	0.933	0.868	0.655	0.566	0.591	0.507	-	-	0.708	0.594
Amns	OxfordFlowers	gazania	1.0	0.913	0.705	0.518	0.655	0.586	0.591	0.514	0.933	0.908	-	-
Amns	OxfordFlowers	tree mallow	0.765	0.824	0.707	0.484	0.655	0.593	0.591	0.513	0.933	0.91	-	-
Amns	OxfordFlowers	trumpet creeper	0.588	0.765	0.709	0.578	0.655	0.627	0.591	0.55	0.933	0.92	-	-
EMMN	StanfordDogs	Pekinese	0.787	0.0	0.59	0.376	0.655	0.278	-	-	0.933	0.828	0.708	0.432
EMMN	StanfordDogs	toy poodle	0.607	0.0	0.591	0.373	0.655	0.308	-	-	0.933	0.836	0.708	0.446
EMMN	StanfordDogs	Scotch terrier	0.625	0.125	0.591	0.347	0.655	0.265	-	-	0.933	0.813	0.708	0.436
EMMN	StanfordCars	2009 Spyker C8 Coupe	0.429	0.0	0.656	0.188	-	-	0.591	0.116	0.933	0.614	0.708	0.148
EMMN	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548	0.476	0.656	0.184	-	-	0.591	0.13	0.933	0.56	0.708	0.126
EMMN	StanfordCars	2011 Ford Ranger SuperCab	0.81	0.0	0.654	0.175	-	-	0.591	0.111	0.933	0.594	0.708	0.136
EMMN	Caltech101	euphonium	1.0	0.105	0.933	0.783	0.655	0.352	0.591	0.297	-	-	0.708	0.45
EMMN	Caltech101	minaret	0.913	0.348	0.933	0.817	0.655	0.36	0.591	0.315	-	-	0.708	0.485
EMMN	Caltech101	platypus	1.0	0.4	0.933	0.838	0.655	0.345	0.591	0.294	-	-	0.708	0.484
EMMN	OxfordFlowers	gazania	1.0	0.0	0.705	0.44	0.655	0.308	0.591	0.312	0.933	0.832	-	-
EMMN	OxfordFlowers	tree mallow	0.765	0.0	0.707	0.445	0.655	0.33	0.591	0.288	0.933	0.829	-	-
EMMN	OxfordFlowers	trumpet creeper	0.588	0.059	0.709	0.413	0.655	0.312	0.591	0.31	0.933	0.828	-	-

C Additional Tasks Full results

Table 11: Image retrieval from text input results showing precision@k for k of 1, 5 and 10 using RN50 model

Model Type	**Class**	**Precision@1**	**Precision@5**	**Precision@10**
CLIP original	Scotch terrier	1.0	0.2	0.2
CLIP original	toy poodle	1.0	0.6	0.5
CLIP original	Pekinese	1.0	0.8	0.6
CLIP original	2009 Spyker C8 Coupe	1.0	0.6	0.5
CLIP original	2010 Dodge Ram Pickup 3500 Crew Cab	1.0	0.2	0.2
CLIP original	2011 Ford Ranger SuperCab	0.0	0.2	0.2
CLIP original	euphonium	1.0	1.0	1.0
CLIP original	minaret	1.0	1.0	1.0
CLIP original	platypus	1.0	1.0	0.6
CLIP original	gazania	1.0	1.0	1.0
CLIP original	tree mallow	0.0	0.8	0.7
CLIP original	trumpet creeper	1.0	0.8	0.5
CLIP original Mean	-	0.833	0.683	0.583
CLIP forget (Lip)	Scotch terrier	0.0	0.0	0.0
CLIP forget (Lip)	toy poodle	1.0	0.2	0.1
CLIP forget (Lip)	Pekinese	0.0	0.0	0.0
CLIP forget (Lip)	2009 Spyker C8 Coupe	0.0	0.8	0.5
CLIP forget (Lip)	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.2	0.3
CLIP forget (Lip)	2011 Ford Ranger SuperCab	0.0	0.0	0.0
CLIP forget (Lip)	euphonium	0.0	0.8	0.8
CLIP forget (Lip)	minaret	0.0	0.4	0.2
CLIP forget (Lip)	platypus	0.0	0.2	0.2
CLIP forget (Lip)	gazania	0.0	0.0	0.0
CLIP forget (Lip)	tree mallow	0.0	0.2	0.2
CLIP forget (Lip)	trumpet creeper	0.0	0.0	0.0
CLIP forget Mean (Lip)	-	0.08	0.23	0.191
CLIP forget (Ours)	Scotch terrier	0.0	0.0	0.0
CLIP forget (Ours)	toy poodle	0.0	0.0	0.0
CLIP forget (Ours)	Pekinese	0.0	0.0	0.0
CLIP forget (Ours)	2009 Spyker C8 Coupe	0.0	0.0	0.0
CLIP forget (Ours)	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.0	0.0
CLIP forget (Ours)	2011 Ford Ranger SuperCab	0.0	0.2	0.1
CLIP forget (Ours)	euphonium	0.0	0.0	0.0
CLIP forget (Ours)	minaret	0.0	0.0	0.0
CLIP forget (Ours)	platypus	0.0	0.0	0.0
CLIP forget (Ours)	gazania	0.0	0.0	0.0
CLIP forget (Ours)	tree mallow	0.0	0.0	0.0
CLIP forget (Ours)	trumpet creeper	0.0	0.0	0.0
CLIP forget Mean (Ours)	-	**0.0**	**0.017**	**0.008**

Table 12: Image retrieval from text input results showing precision@k for k of 1, 5 and 10 using ViT-B/16 model

Model Type	**Class**	**Precision@1**	**Precision@5**	**Precision@10**
CLIP original	Scotch terrier	0.0	0.0	0.1
CLIP original	toy poodle	1.0	0.8	0.7
CLIP original	Pekinese	1.0	0.4	0.5
CLIP original	2009 Spyker C8 Coupe	1.0	0.8	0.8
CLIP original	2010 Dodge Ram Pickup 3500 Crew Cab	1.0	0.6	0.5
CLIP original	2011 Ford Ranger SuperCab	1.0	0.8	0.5
CLIP original	euphonium	1.0	1.0	1.0
CLIP original	minaret	1.0	1.0	1.0
CLIP original	platypus	1.0	1.0	0.9
CLIP original	gazania	1.0	1.0	1.0
CLIP original	tree mallow	0.0	0.4	0.4
CLIP original	trumpet creeper	1.0	0.8	0.6
CLIP original Mean	-	0.833	0.717	0.667
CLIP forget (Lip)	Scotch terrier	0.0	0.4	0.4
CLIP forget (Lip)	toy poodle	0.0	0.0	0.1
CLIP forget (Lip)	Pekinese	0.0	0.0	0.2
CLIP forget (Lip)	2009 Spyker C8 Coupe	1.0	0.8	0.8
CLIP forget (Lip)	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.0	0.1
CLIP forget (Lip)	2011 Ford Ranger SuperCab	1.0	0.6	0.4
CLIP forget (Lip)	euphonium	1.0	1.0	0.6
CLIP forget (Lip)	minaret	1.0	1.0	0.9
CLIP forget (Lip)	platypus	1.0	1.0	0.5
CLIP forget (Lip)	gazania	1.0	0.2	0.4
CLIP forget (Lip)	tree mallow	0.0	0.0	0.2
CLIP forget (Lip)	trumpet creeper	0.0	0.2	0.2
CLIP forget Mean (Lip)	-	0.5	0.433	0.4
CLIP forget (Ours)	Scotch terrier	0.0	0.0	0.0
CLIP forget (Ours)	toy poodle	0.0	0.0	0.0
CLIP forget (Ours)	Pekinese	0.0	0.0	0.0
CLIP forget (Ours)	2009 Spyker C8 Coupe	0.0	0.0	0.0
CLIP forget (Ours)	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.0	0.0
CLIP forget (Ours)	2011 Ford Ranger SuperCab	0.0	0.0	0.0
CLIP forget (Ours)	euphonium	0.0	0.0	0.0
CLIP forget (Ours)	minaret	0.0	0.0	0.0
CLIP forget (Ours)	platypus	0.0	0.0	0.0
CLIP forget (Ours)	gazania	0.0	0.0	0.0
CLIP forget (Ours)	tree mallow	0.0	0.0	0.0
CLIP forget (Ours)	trumpet creeper	0.0	0.0	0.0
CLIP forget Mean (Ours)	-	**0.0**	**0.0**	**0.0**

D Interpreting Difficulty of Unlearning a Class

We can directly analyze the low-rank adaptation change in the projection matrix to understand the difficulty of unlearning a certain class. Specifically, we examine the Frobenius norm of the adaptation matrix. Our hypothesis is that a larger Frobenius norm means that a greater modification in the projection matrix is required to successfully forget a target class. Such greater modification will make maintaining other classes accuracy on the similar level harder and thus the average score metrics (the lower the better) will increase as well.

The Frobenius norm of a matrix A is defined as:

$$\|A\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2} \tag{4}$$

We now plot the Frobenius norm of the projection change matrix alongside the average score metrics for the two networks. To ensure more robust statistical results, we include 30 randomly sampled classes:

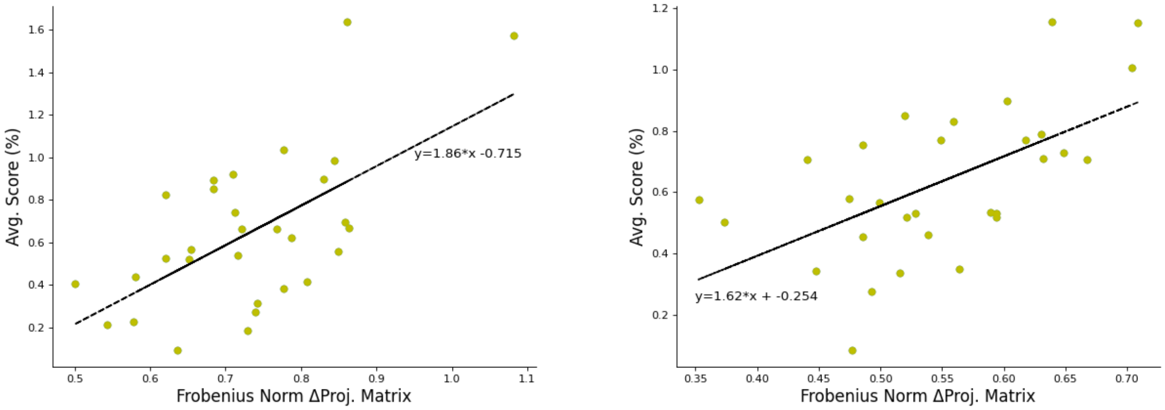


Figure 4: Interpreting the difficulty of unlearning a class by looking at the Frobenius norm of the projection change matrix. Figure on the **left** shows the results for the RN50 model and on the **right** for the ViT-B/16 model.

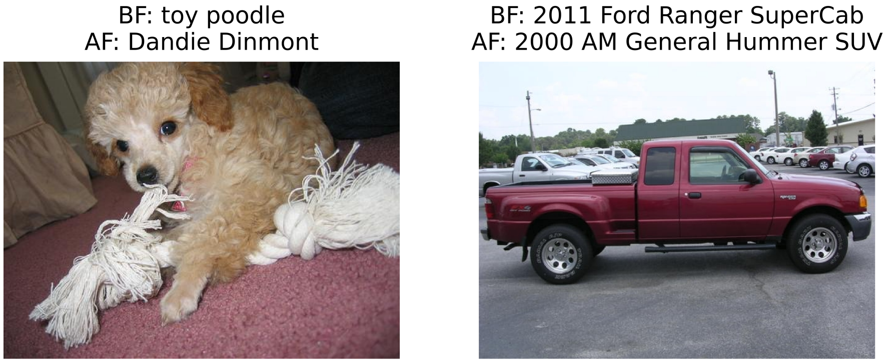
From Fig. 4 we can see that there is a positive relation between the Frobenius norm of the change in the projection matrix and the average score metrics confirming our hypothesis. The correlation between the Frobenius norm of the change is 0.62 and 0.59 respectively for RN50 and ViT-B/16 models showing that the relation between the two is significant.

E Implementation Details

For both the models we use the λ_1 of 0.3, λ_3 of 1 and a varying λ_2 with initial value of 1.1 incrementing by 0.05 until the reduction in the second loss component exceeds 0.75% of its initial value. We optimize the low-ranking matrices A and B of rank r of 5 for 2000 iterations using Adam optimizer with learning rate of 0.01 and saving the weights that achieve the minimum loss. We use an empty template with only the name of the class when unlearning.

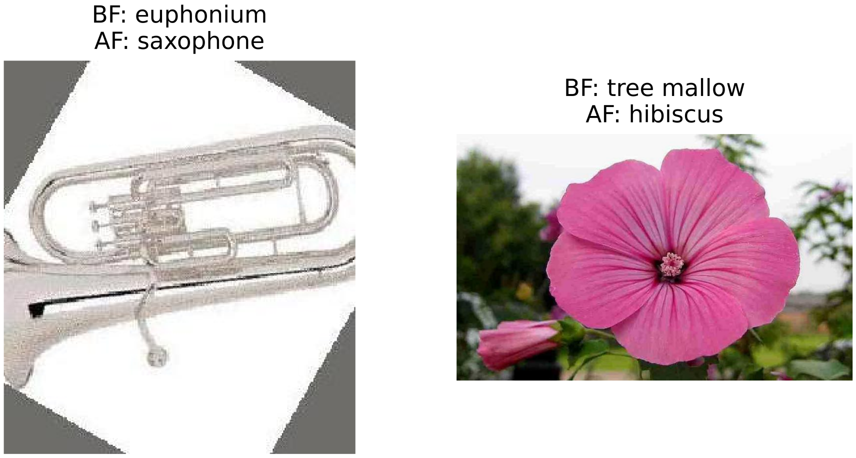
F Predictions Before and After Unlearning on the Forget Class

In Fig. 5 we show examples of the model's predictions before (BF) and after forgetting (AF). We observe that the new classes predicted by the unlearned model are close to the correct ones indicating that our method targets specific knowledge of the model while preserving its general understanding.



BF: toy poodle
AF: Dandie Dinmont

BF: 2011 Ford Ranger SuperCab
AF: 2000 AM General Hummer SUV



BF: euphonium
AF: saxophone

BF: tree mallow
AF: hibiscus

Figure 5: Predictions before (BF) and after forgetting (AF) with the prediction BF representing the target class to forget.

G Generated Semantically Similar Classes to Preserve

Here we show the list of semantically similar classes generated by an LLM with a prompt *Generate semantically similar classes to {class}*.

StanfordDogs:

Shih Tzu, Lhasa Apso, Maltese, Havanese, Bichon Frise, Yorkshire Terrier, Pomeranian, Cavalier King Charles Spaniel, Papillon, Japanese Chin, Brussels Griffon, Miniature Schnauzer, West Highland White Terrier, Cairn Terrier, Norfolk Terrier, Norwich Terrier, Tibetan Spaniel, Tibetan Terrier, Silky Terrier, Affenpinscher, Chinese Crested, Italian Greyhound, Toy Manchester Terrier, Toy Fox Terrier, Australian Terrier, Border Terrier, Dandie Dinmont Terrier, Sealyham Terrier, Skye Terrier, Welsh Terrier, Lakeland Terrier, Jack Russell Terrier, Parson Russell Terrier, Rat Terrier, Bedlington Terrier, Manchester Terrier, Fox Terrier, Wire Fox Terrier, Smooth Fox Terrier, Irish Terrier, Glen of Imaal Terrier, Kerry Blue Terrier, Soft Coated Wheaten Terrier, Bull Terrier, Miniature Bull Terrier, Boston Terrier, French Bulldog, English Bulldog, American Bulldog, Boxer, Pug, Miniature Pinscher, German Pinscher, Doberman Pinscher, Great Dane, Mastiff, Bullmastiff, Neapolitan Mastiff, Dogue de Bordeaux, Rottweiler, Saint Bernard, Bernese Mountain Dog, Greater Swiss Mountain Dog, Newfoundland, Leonberger, Tibetan Mastiff, Chihuahua, Poodle, Miniature Poodle, Standard Poodle, Shetland Sheepdog, Collie, Border Collie, Australian Shepherd, Australian Cattle Dog, Old English Sheepdog, Bearded Collie, Briard, Welsh Corgi, Cardigan Welsh Corgi, Pembroke Welsh Corgi, American Eskimo Dog, Alaskan Malamute, Siberian Husky, Samoyed, Shiba Inu, Akita, Basenji, Beagle, Bloodhound, Basset Hound, Dachshund, Coonhound, Foxhound, Whippet, Greyhound, Saluki, Afghan Hound, Borzoi, Irish Wolfhound, Scottish Deerhound

StanfordCars:

Chevrolet Silverado 1500,GMC Sierra 1500,Toyota Tundra,Nissan Titan,Ram 1500,Ford F-150,Honda Ridgeline,Chevrolet Colorado,GMC Canyon,Toyota Tacoma,Nissan Frontier,Jeep Gladiator,Ford Maverick,Hyundai Santa Cruz,Chevrolet Silverado 2500HD,GMC Sierra 2500HD,Ford F-250 Super Duty,Ram 2500,Chevrolet Silverado 3500HD,GMC Sierra 3500HD,Ford F-350 Super Duty,Ram 3500,Chevrolet Silverado 4500HD,Ford F-450 Super Duty,GMC Sierra 4500HD,Ram 4500,Chevrolet Silverado 5500HD,Ford F-550 Super Duty,GMC Sierra 5500HD,Ram 5500,Ford F-650,Ford F-750,International CV Series,Mitsubishi Fuso Canter,Isuzu N-Series,Hino 268,Freightliner M2 106,Peterbilt 220,Kenworth T270,Ram ProMaster,Ford Transit,Mercedes-Benz Sprinter,Nissan NV,Chevrolet Express,GMC Savana,Ram ProMaster City,Ford Transit Connect,Nissan NV200,Chevrolet Colorado ZR2,Toyota Tacoma TRD Pro,Jeep Wrangler Rubicon,Ford Ranger Tremor,Ram Rebel,Chevrolet Silverado Trail Boss,GMC Sierra AT4,Ford F-150 Raptor,Nissan Titan XD,Toyota Tundra TRD Pro,Chevrolet Avalanche,Honda Element,Ford Explorer Sport Trac,Lincoln Mark LT,Cadillac Escalade EXT,Hummer H2 SUT,Chevrolet SSR,Subaru Baja,Dodge Dakota,Mazda B-Series,Mitsubishi Raider,Suzuki Equator,Isuzu i-Series,Ford Courier,Volkswagen Amarok,Peugeot Landtrek,Fiat Fullback,Renault Alaskan,Mercedes-Benz X-Class,SsangYong Musso,Great Wall Steed,Mahindra Scorpio Getaway,Tata Xenon,Holden Colorado,HSV Maloo,Ford Falcon Ute,Chevrolet S-10,Ford Ranger Raptor,RAM 1200,Toyota Hilux,Chevrolet LUV,Ford Courier,Mazda BT-50,Mitsubishi Triton,Nissan Navara,Isuzu D-Max,Volkswagen Tarok,Jeep Comanche

Caltech101:

Accordion, Bagpipes, Banjo, Bassoon, Cello, Clarinet, Cornet, Double Bass, Drum Set, Flute, French Horn, Guitar, Harp, Mandolin, Marimba, Oboe, Piano, Saxophone, Sitar, Sousaphone, Tambourine, Trombone, Trumpet, Tuba, Ukulele, Viola, Violin, Xylophone, Zenko Drum, Glockenspiel, Concertina, Hurdy-Gurdy, Lute, Melodica, Piccolo, Pipe Organ, Recorder, Theremin, Triangle, Bass Drum, Cabasa, Castanets, Claves, Conga Drum, Cowbell, Djembe, Guiro, Kalimba, Maracas, Shekere, Sleigh Bells, Snare Drum, Talking Drum, Timpani, Vibraslap, Whip, Washboard, Zephyr Organ, Zither, Azimuth Marker, Bell Tower, Belfry, Cathedral Spire, Church Steeple, Dome, Gazebo, Lighthouse, Obelisk, Pagoda, Watchtower, Water Tower, Windmill, Cairn, Cenotaph, Column, Monolith, Obelisk, Pavilion, Pyramid, Stupa, Totem Pole, Triumphal Arch, Rotunda, Spire, Tower, Ziggurat, Amphibian, Anteater, Armadillo, Barramundi, Basilisk, Beaver, Capybara, Chameleon, Coatimundi, Echidna, Gecko, Gila Monster, Iguana, Komodo Dragon, Koala, Marsupial, Mole, Monotreme, Newt, Numbat, Opossum, Pangolin, Platypus, Quokka, Quoll, Salamander, Shrew, Skink, Sloth, Sugar Glider, Tasmanian Devil, Tree Frog, Tuatara, Wombat, Anhinga, Auk, Bittern, Booby, Cormorant, Crane, Curlew, Egret, Flamingo, Frigatebird, Gannet, Grebe, Heron, Ibis, Jaeger, Kestrel, Kingfisher, Kittiwake, Loon, Oystercatcher, Pelican, Petrel, Puffin, Rail, Razorbill, Sandpiper, Shearwater, Skua, Snipe, Tern, Turnstone, Wader, Whimbrel, Woodcock, Meerkat, Mongoose, Pangolin, Platypus, Potto, Puffin, Quokka, Quoll, Raccoon, Red Panda, Ringtail, Skunk, Sloth, Sugar Glider, Tasmanian Devil, Tenrec, Tree Shrew, Wombat, Zebra Finch, Zebu, Zonkey, Zorilla, Zygodont

OxfordFlowers:

Rose, Tulip, Lily, Daisy, Sunflower, Orchid, Marigold, Lavender, Daffodil, Chrysanthemum, Carnation, Hibiscus, Iris, Peony, Poppy, Lotus, Bluebell, Magnolia, Gardenia, Jasmine, Azalea, Camellia, Geranium, Hyacinth, Petunia, Zinnia, Begonia, Cosmos, Foxglove, Freesia, Gladiolus, Hollyhock, Lilac, Narcissus, Snapdragon, Sweet Pea, Verbena, Violet, Wisteria, Aster, Anemone, Gaura, Bachelor's Button, Bellflower, Buttercup, Calla Lily, Canna, Protea, Columbine, Coreopsis, Delphinium, Gaillardia, Primula, Heliotrope, Impatiens, Kalanchoe, Lantana, Morning Glory, Nasturtium, Pansy, Phlox, Plumeria, Primrose, Ranunculus, Rhododendron, Scabiosa, Sedum, Stock, Tithonia, Trillium, Tuberose, Wallflower, Yarrow, Yucca, Amaryllis, Bougainvillea, Bromelia, Angelonia, Armeria, Balloon Flower, Ballmoss, Bee Balm, Black-eyed Susan, Bleeding Heart, Borage, Browallia, Candytuft, Clematis, Cleome, Cockscomb, Coral Bells, Corydalis, Crocosmia, Cyclamen, Diascia, Dusty Miller, Echinacea, Euphorbia, Four O'Clock, Gazania, Geum