

Zero-Shot Class Unlearning in CLIP with Synthetic Samples

Alexey Kravets
 University of Bath, UK
 ak3095@bath.ac.uk

Vinay Namboodiri
 University of Bath, UK
 vpn22@bath.ac.uk

Abstract

Machine unlearning is a crucial area of research. It is driven by the need to remove sensitive information from models to safeguard individuals' right to be forgotten under rigorous regulations such as GDPR. In this work, we focus on unlearning within CLIP, a dual vision-language encoder model trained on a massive dataset of image-text pairs using contrastive loss. To achieve forgetting we expand the application of Lipschitz regularization to the multimodal context of CLIP. Specifically, we ensure the smoothing of both visual and textual embeddings associated with the class intended to be forgotten relative to the perturbation introduced to the samples from that class. Additionally, importantly, we remove the necessity for real forgetting data by generating synthetic samples through gradient ascent maximizing the target class. Our forgetting procedure is iterative, where we track accuracy on a synthetic forget set and stop when accuracy falls below a chosen threshold. We employ a selective layers update strategy based on their average absolute gradient value to mitigate over-forgetting. We validate our approach on several standard datasets and provide thorough ablation analysis and comparisons with previous work. Full code is released [here](#).

1. Introduction

Machine unlearning [19], as opposed to machine learning, is emerging as a significant area of research. Firstly, with the rise of stringent regulations such as GDPR¹, there's a growing demand to remove sensitive information from models thereby upholding individuals' right to be forgotten. Secondly, unlearning is crucial for enhancing security in scenarios vulnerable to model inversion attacks [1] or data poisoning [2]. These attacks aim to manipulate the behavior of the model, making unlearning a valuable defense mechanism. Lastly, the accumulation of outdated information can lead to accuracy deterioration in models over time. To counteract this, models must possess the

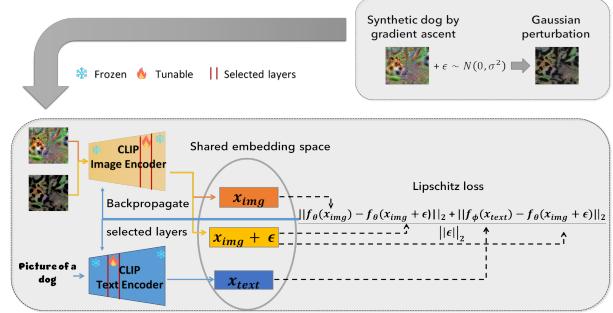


Figure 1. Overview of the approach. First, we generate synthetic images of a class to forget by gradient ascent. Then, we perform a Gaussian perturbation of the images and pass the original and perturbed images through CLIP image encoder and textual description of the class through CLIP textual encoder. As image and text are projected into a shared embedding space, we can utilize the final representation of the perturbed image as a perturbed representation of text "Picture of a dog". Finally, we apply the Lipschitz regularization and backpropagate to selected layers by importance to visual and textual encoders.

ability to forget obsolete data and integrate new information into the learning process effectively [15].

Challenges of Unlearning CLIP: CLIP [16] and other related vision-language models can be used for image and text-based tasks without being retrained. The unlearning for such models is especially challenging due to the following three reasons: a) they have separate visual and textual encoders. Thus, even if we unlearn using the image representation, the textual modality may still be able to generalise. For instance, if we want to forget the concept of a specific breed of dog - Chihuahua and are able to use a method to unlearn in the image representation, the model may still be able to caption the dog Chihuahua as the information may be retained in the textual representation b) we do not have access to the data used to train the CLIP model². Techniques that require availability of samples for unlearning are thus not applicable c) CLIP model has large number

¹<https://gdpr-info.eu/art-17-gdpr/>

²<https://github.com/openai/CLIP/issues/127>

of parameters and thus retraining the model is unfeasible. In our work, we explicitly design a method that overcomes all these challenges.

Devising a solution: In order to solve the problem, we need a basic principle that we can use for unlearning. The most straightforward method is to retrain the original model without using the specific data that is intended to be forgotten. As mentioned, for a foundational model like CLIP, that is computationally too expensive. The other approach could be to use random sampling of data to be retained and some samples of the data to be forgotten and using a technique like amnesiac forgetting [9] to adapt the CLIP model. As a CLIP model spans lot of concepts, that is not practical. We next consider methods that rely only on samples of data to be forgotten. A recent method [7] shows that by retraining the model by perturbing the samples to be forgotten, one can achieve unlearning. We follow this approach and aim to extend it to address all the challenges associated with unlearning in CLIP.

The first step in using the perturbation approach is to determine the type of perturbation to apply. We compare Lipschitz with other embedding perturbation methods and find that Lipschitz perturbation performs well. Consequently, we adopt this approach for unlearning. Next, we tackle the challenge of needing real data samples for forgetting. Since we lack access to the actual data used in training the model, we address this problem by generating synthetic image samples using gradient ascent [17]. We further solve the problem of unlearning with the dual modalities. We finally need to solve the computational challenge of unlearning. We do so by selectively changing the weights of only a few layers instead of retraining the whole model. We pursue an iterative unlearning approach that allows us to precisely control the amount of unlearning required to forget the specific class while retaining the information for all other classes.

Overview of our approach: We provide an overview of our approach in Fig. 1. Let us consider that a specific class needs to be forgotten, such as the class of dog *Chihuahua*. Our approach involves generating synthetic samples of this class using gradient ascent. We then use iterative forgetting through Lipschitz regularization using the synthetic image samples and do the same for the textual representation through the joint image-text representation. Our approach does not rely on any real training data to forget. As our solution does not rely on real samples to perform unlearning it is zero-shot.

2. Related Work

Multimodal Unlearning Multimodal forgetting remains under-explored in the literature. Authors in [3] introduce multimodal unlearning defining it by three key properties: modality decoupling, unimodal knowledge retention, and multimodal knowledge retention. The methodology in-

volves optimizing a multimodal model through three losses to effectively unlearn forgotten data while preserving the knowledge of retained data, satisfying these properties. However, this methodology cannot be applied to CLIP due to its non-parametric fusion of modalities. Also the method requires training data for knowledge retention. In [21], authors attempt to induce forgetting in Stable Diffusion (SD) through attention steering. This process entails minimizing cross-attention maps from the Stable Diffusion model between latent input features and textual embeddings of concepts intended for forgetting. This disentangles textual associations from image associations of the target concepts. Similarly for the SD model, authors in [8] utilize the inverse of the energy-based composition [11] to guide generation probability away from conditional towards the unconditional prediction of concepts to be forgotten by negating the predicted noise associated with the forget concept. The techniques used for generative models are not applicable to dual-encoder models such as CLIP. To the best of our knowledge, none of the existing multimodal unlearning methods specifically address forgetting in CLIP.

Machine Unlearning with Generated Data In [18], the authors generated anti-samples for the classes meant to be forgotten by error maximization, which is the reverse of the minimization process employed during training. These anti-samples exhibit patterns opposite to those of the sample classes. They also train using data samples from the training data classes to be remembered. During the CLIP forgetting procedure, we also generate synthetic data. However, our approach diverges from that in [18] as our method employs loss minimization to generate synthetic data with same patterns rather than opposite patterns of the forget data. The challenge induced by the approach [18] is that a delicate balance needs to be struck between retraining with classes to be remembered vs anti-samples of classes that need to be forgotten. Our approach uses regularization and synthetic samples in a more efficient manner.

Zero-shot Machine Unlearning Authors in [4] have taken the approach in [18] described previously a step further by eliminating the dependence on real data from the classes meant to be retained. They utilize a synthesis approach similar to that in [18] for generating forget data, while the retain data are synthesized using an error minimization procedure. This solution is, therefore, zero-shot as it does not rely on any real data. Note that the method is not as practical as our approach, as for general models like CLIP, there is no explicit class that needs to be retained, but, the general capabilities needs to be retained. The work that is closest to ours is the work by Foster *et al.* [7]. They perform forgetting via local Lipschitz regularization on unimodal vision models. They do not require the

retain data but rely on the real data to be forgotten for training which does not make them completely zero-shot. We extend their method to multimodal CLIP model and eliminate the need for actual data. We provide comparisons with both [7] and [4] and show that our proposed method does perform better for CLIP while being more practical than either of these methods.

3. Preliminaries

3.1. CLIP Dual Encoder Model

CLIP is a dual encoder model that consists of visual and textual components. The visual component processes images and extracts their features while the textual component processes textual descriptions and encodes them into a fixed-length vector representation. Due to the contrastive pre-training during which CLIP learns to associate images and text in a shared embedding space, CLIP is able to perform a variety of zero-shot tasks, including classification.

For classification, given N classes, they are encoded within a contextual prompt such as "A photo of a {class}" with the CLIP textual encoder f_ϕ . This results in the classifier weight matrix $W \in \mathbb{R}^{N \times d}$, where d represents the embedding dimension. When presented with a test image I_i , it is encoded using the CLIP image encoder f_θ :

$$T_i = f_\theta(I_i), T_i \in R^{1 \times d} \quad (1)$$

Following this encoding step, the dot product between the matrix W and the embedded image T_i is computed to get the zero-shot classification logits for the image I_i :

$$\text{CLIPlogits}_i = T_i W^T, \text{CLIPlogits}_i \in R^{1 \times N} \quad (2)$$

3.2. Local Lipschitz Regularization

In order to unlearn samples for a particular class, we rely on an existing technique. Our method is based on work by Foster *et. al.* [7] that utilizes the concept of Lipschitz continuity for forgetting. The idea is to locally perturb an input image that needs to be forgotten by a Gaussian noise and minimizing the ratio between the change in the outputs for the perturbed and unperturbed images and change in the input. This regularization was first proposed by Yoshida and Miyato [20] as a means for obtaining generalization. However, [7] observed that using sufficient Gaussian perturbation, the learnt response for the particular input is unlearnt. Formally, given some Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ of the same dimensionality of the input image x we minimize:

$$\ell = \mathbb{E} \left(\frac{\|f_\theta(x) - f_\theta(x + \epsilon)\|_2}{\|\epsilon\|_2} \right) \quad (3)$$

The expectation is approximated by averaging over N perturbations. We have also evaluated direct regularization on

the embedding loss and observed the Lipschitz regularization to be better.

The Lipschitz regularization while being a useful way for unlearning, can be a weak signal not removing enough class information from the model. Our approach therefore relies on assuring that we ensure the unlearning of the class by validating on a set of examples synthesized for the class and using iterative unlearning for a good performance.

4. Method

4.1. Setting

Given the model's training set denoted as D , in a standard machine unlearning setting we identify two subsets of D : data to retain D_r and data to forget D_f . However, our setting differs because the training data for CLIP have not been made publicly available. Consequently, we are unable to determine whether a specific data sample was used for training. Therefore, we cannot verify the forgetting of a particular sample or compare the forgetting performance to a retrained model on D_r excluding D_f . Even if we had access to the training data, achieving the latter would be infeasible due to the substantial computational resources required to retrain large-scale models like CLIP.

4.2. Extending Local Lipschitz Regularization to CLIP

Authors in [7] utilize local Lipschitz regularization exclusively on vision models. Our experiments reveal that updating solely the vision branch is insufficient for CLIP to forget a selected class and adjustments to both vision and text branches are necessary. Adapting Lipschitz regularization to a dual encoder CLIP model poses a challenge as there is no direct method to perturb discrete language tokens with Gaussian noise. One potential approach involves directly modifying tokens, however, determining the degree of noise introduced to the input becomes ambiguous. Since the final layer embedding from both the CLIP vision and language branches are mapped to a shared image-text space, we can avoid perturbing the text directly. Instead, we use the perturbed image as a proxy for the perturbed text and compute Lipschitz regularization for the text branch in the same manner as for the image branch. We thus define our loss objective for both vision and text branches as follows:

$$\ell = \mathbb{E} \left(\frac{\|f_\theta(\mathbf{x}_{img}) - f_\theta(\mathbf{x}_{img} + \epsilon)\|_2}{\|\epsilon\|_2} \right) \quad (4)$$

$$+ \frac{\|f_\phi(\mathbf{x}_{text}) - f_\theta(\mathbf{x}_{img} + \epsilon)\|_2}{\|\epsilon\|_2} \quad (5)$$

where f_θ is the image encoder and f_ϕ is the text encoder outputting the last layers embedding, x_{img} is the image sample and x_{text} its corresponding class wrapped in

the contextual prompt. As the CLIP objective ensures that the shared embeddings for image and text are close, requiring the text embedding to be close to the perturbed image embedding is a valid unlearning regularization. The expectation in the equation 5 is approximated with Monte Carlo using N perturbations for each sample.

4.3. Synthetic Forget Samples

We create synthetic forget samples by performing gradient ascent to maximize the target class [17]. Starting from random noise sample we perform the following update until the prediction on x is of a desired class:

$$x = x + \alpha \frac{\partial L(x, y)}{\partial x} \quad (6)$$

Where α is the learning rate, y the desired target class and L the loss function. We then use these synthetic forget samples to update the weights in CLIP using local Lipschitz regularization loss. These synthetic samples do not have the appearance of the sample class (examples in the appendix) due to the simple approach we use for generation, however, these samples suffice for unlearning a class.

4.4. Layers Update Based on the Average Gradient

We have observed that updating all parameters of CLIP results in excessive forgetting. Therefore, we perform a selective update of layers based on their importance to the samples we aim to forget. To determine this importance, we calculate the average absolute gradient value of the layers and update a specific number of layers in both the vision and text branches during each iteration. Results of the ablation on forgetting with all the parameters are shown in Tab. 7.

4.5. Stopping Criteria

To achieve gradual forgetting, we begin with a low value of σ and a small number of layers to update in CLIP. During forgetting we monitor the accuracy on the synthetic training samples and stop the forgetting process when the accuracy falls below a predefined threshold. If during forgetting the accuracy of the training synthetic data does not drop we increase both the amount of noise σ and the number of layers for a more aggressive forgetting.

5. Experiments

5.1. Comparable Methods

As our approach is the first to be proposed for unlearning in CLIP, there are no direct comparable methods available. Therefore, for this we have adapted a number of methods to provide a fair evaluation of our approach. These are given as follows:

Embedding regularization loss (Emb) Similarly to the method outlined before we perturb the inputs with a Gaussian noise but this time, instead of Lipschitz regularization we utilize embeddings regularization that is defined as follows:

$$M = \mathbb{E}(\|f_\theta(x_{img}) - f_\theta(x_{img} + \epsilon)\|_2 + \quad (7)$$

$$\|f_\phi(x_{text}) - f_\theta(x_{img} + \epsilon)\|_2) + \quad (8)$$

$$\alpha \cdot \|f_\theta(x_{img}) + f_\phi(x_{text})\|_2 \quad (9)$$

As only synthetic forget data is used it is a zero-shot method like ours (denoted *ZS* in the results Tab. 1).

Amnesiac forgetting with synthetic data (Amns) We adapt the approach shown in [9] to the multimodal setting fine-tuning CLIP with the contrastive loss used to train the CLIP model. We replace the labels corresponding to the forget class randomly with a different label using synthetic data. To keep it zero-shot we do not use the retain data but only train with the forget data.

Error Minimization-Maximization Noise (EMMN) We adapt the method in [4] to multimodal setting learning retain and forget samples through loss minimization and maximization respectively and train the model on them. As the method does not require any real data it is zero-shot.

Unimodal Lipschitz (ULip) We perform forgetting only on the visual encoder of CLIP as in [7] using image perturbation and local Lipschitz regularization. We run the method using **real** data to forget. As it requires real data it is not completely zero-shot (denoted *semi ZS* in Tab. 1).

Amnesiac forgetting with real data including retain data (AmnsRetain) Similarly to *Amns* [9] described earlier we replace the labels corresponding to the forget class randomly with a different label using **real** data. This time we include the retain real data from the dataset to which the label to forget belongs to. As this method uses the data to retain it is not zero-shot (denoted *not ZS* in Tab. 1).

5.2. Datasets

We assess CLIP’s forgetting using four high-quality fine-grained datasets: Caltech101 [6] consists of images belonging to 101 distinct categories containing examples of objects or scenes. StanfordCars [13] includes images of cars categorized into different makes and models. Oxford-Flowers [14] comprises images of flowers from 102 species while StanfordDogs [12] contains images of dogs categorized into different breeds. These datasets comprise images spanning various categories with minimal overlap between them, thus we do not need to filter for similar classes to the forget class across different datasets during evaluation.

5.3. Implementation Details

We perform our experiments on CLIP with ResNet50 [10] as visual encoder. We have observed that allowing all parameters in the vision encoder to be chosen for update during the forgetting process results in poor performance on other classes. Therefore, we restrict parameter updates to the attention layers in the RN50 vision encoder while all parameters in the text encoder are eligible for the update selection. Fig. 4 in the appendix illustrates the top 25 most frequently updated layers of the RN50 model. Finally, we generate 64 synthetic samples and stop the forgetting process when their accuracy goes below 0.1. Experiments with ViT [5] and other implementation details can be found in the appendix.

5.4. Evaluation

As we mentioned previously, we are unable to compare CLIP performance on a forget class to the retrained version of the model without the forget data as we effectively do not know whether a certain sample was used to train CLIP due to the data being not open sourced. However, even if the data were open sourced the computational power required to retrain such a big model that relies on a huge amount of data for its zero-shot capabilities would be a challenge. Therefore, after the forgetting procedure we will evaluate CLIP’s classification performance on the selected class for forgetting, the remaining classes from that dataset and the classification performance on the remaining three datasets. It’s important to highlight that we aim for the accuracy on the forget class to be as low as possible, while maintaining similar accuracy levels on the remaining classes of dataset the forget class belongs to and all other datasets compared to before the application of the forgetting procedure.

5.5. Results

Comparison across different methods In Tab. 1 we present the results of different forgetting methods averaging across three classes on four selected datasets. Full results for all classes and methods can be found in the appendix. *Method* column refers to the method used for the experiments. *Forgetting Type* refers to whether the method is zero-shot (ZS), indicating that no real data was used; semi zero-shot (semi ZS), where real data were used for forgetting; and not zero-shot (not ZS), where both retain and forget data were real. *Dataset* column specifies the dataset from which the class to be forgotten was selected, while the *Avg. Target Class acc.* denote the average accuracy on the target class before (BF) and after (AF) forgetting. *Avg. Other Classes acc.* indicates the average accuracy on the other classes in the dataset, excluding the class to be forgotten. The final eight columns display the results on the remaining datasets reported both before and after forgetting. We observe that

our forgetting procedure, referred to as *Lip* has proven successful as indicated by a notable decrease in accuracy for the targeted classes, often approaching zero. Conversely, the accuracy for the remaining classes and other datasets remain largely unaffected after applying the forgetting procedure. In comparison, the Embedding loss referred to as *Emb* appears to be more aggressive than Lipschitz, not only erasing target knowledge but also impacting knowledge about other classes. On the other hand *Amns* method has less forgetting power often resulting in not enough drop in accuracy of the target class and at the same time when forgetting was relatively successful results in over-forgetting on not targeted classes. The *AmnsRetain* approach, which utilizes real data for both retention and forgetting, although not directly comparable to our method being not zero-shot, enables CLIP to forget the target class. We also observe that the accuracy on *Avg. Other Classes acc. AF* is often much higher than *BF* because of the classes to retain used for fine-tuning the model to regain the knowledge lost during forgetting. However, we note that datasets to which the forget class does not belong, and whose data was not used for retention, perform less effectively compared to our method. This demonstrates how large models like CLIP where we do not have information about the training data and classes suffer from drop in the accuracy on classes not included in the retain data. Therefore, our method not only competes in forgetting without using any real data and any retention data but also surpasses *AmnsRetain* in terms of maintaining accuracy on other datasets. The *EMMN* method, while often facilitating forgetting of the target class experiences significant decrease in accuracy, both on the not targeted classes of the dataset from which the forget class was picked and on other datasets. Finally, *ULip* is not only not powerful enough to forget the target class but it also destroys knowledge not related to the target class resulting in a substantial drop on other classes of both related and unrelated datasets to the forget class. We attribute this phenomenon to asymmetric forgetting where attempting to erase knowledge in only one encoder disrupts the connection between the two encoders and consequently affects the projection in the shared embedding space.

Comparison of our method with real and synthetic data We present results of our method with real and synthetic data on three classes for four different datasets in Tabs. 3 and 2 respectively. We see that forgetting yields similar results for synthetic and real data.

5.6. Verification of Forgetting Success

The accuracy achieved on synthetic forget data should serve as a measure of how effectively the model has forgotten a class. We find that for this indicator to be consistent the probability of the predicted class on synthetic samples

Table 1. Forgetting results. We compare our method (Lip) to five other methods averaging across three classes for four selected datasets. We aim to minimize the Avg. *Target Class acc.* AF while maintaining Avg. *Other Classes acc.* AF and other datasets at a similar level to that before forgetting (BF).

Method	Dataset	Forgetting Type	Avg. Target		Avg. Other		Avg.		Avg.		Avg.	
			Class acc.	Classes acc.	StanfordCars	StanfordDogs	Caltech101	OxfordFlowers	BF	AF	BF	AF
Lip (Ours)	StanfordCars	ZS	0.397	0.056	0.558	0.551	-	-	0.517	0.513	0.857	0.86
Emb	StanfordCars	ZS	0.397	0.087	0.558	0.536	-	-	0.517	0.51	0.857	0.85
Amns [9]	StanfordCars	ZS	0.397	0.357	0.558	0.498	-	-	0.517	0.505	0.857	0.863
EMMN [4]	StanfordCars	ZS	0.397	0.0	0.558	0.054	-	-	0.517	0.043	0.857	0.424
ULip [7]	StanfordCars	semi ZS	0.397	0.127	0.558	0.457	-	-	0.517	0.502	0.857	0.848
AmnsRetain [9]	StanfordCars	not ZS	0.397	0.04	0.558	0.711	-	-	0.517	0.509	0.857	0.881
Lip	StanfordDogs	ZS	0.593	0.048	0.516	0.516	0.558	0.558	-	-	0.857	0.866
Emb	StanfordDogs	ZS	0.593	0.261	0.516	0.479	0.558	0.554	-	-	0.857	0.836
Amns	StanfordDogs	ZS	0.593	0.327	0.516	0.465	0.558	0.556	-	-	0.857	0.848
EMMN	StanfordDogs	ZS	0.593	0.0	0.516	0.053	0.558	0.107	-	-	0.857	0.493
ULip	StanfordDogs	semi ZS	0.593	0.429	0.516	0.47	0.558	0.539	-	-	0.857	0.842
AmnsRetain	StanfordDogs	not ZS	0.593	0.044	0.516	0.663	0.558	0.521	-	-	0.857	0.838
Lip	Caltech101	ZS	0.839	0.081	0.857	0.865	0.558	0.557	0.517	0.52	-	-
Emb	Caltech101	ZS	0.839	0.131	0.857	0.83	0.558	0.546	0.517	0.501	-	-
Amns	Caltech101	ZS	0.838	0.33	0.857	0.834	0.558	0.553	0.517	0.502	-	-
EMMN	Caltech101	ZS	0.839	0.0	0.857	0.397	0.558	0.097	0.517	0.081	-	-
ULip	Caltech101	semi ZS	0.839	0.666	0.857	0.854	0.558	0.56	0.517	0.509	-	-
AmnsRetain	Caltech101	not ZS	0.839	0.0	0.857	0.925	0.558	0.526	0.517	0.505	-	-
Lip	OxfordFlowers	ZS	0.848	0.0	0.659	0.645	0.558	0.557	0.517	0.51	0.857	0.868
Emb	OxfordFlowers	ZS	0.848	0.442	0.659	0.625	0.558	0.553	0.517	0.505	0.857	0.85
Amns	OxfordFlowers	ZS	0.848	0.388	0.659	0.592	0.558	0.54	0.517	0.487	0.857	0.835
EMMN	OxfordFlowers	ZS	0.848	0.0	0.659	0.121	0.558	0.121	0.517	0.112	0.857	0.676
ULip	OxfordFlowers	semi ZS	0.848	0.691	0.659	0.59	0.558	0.549	0.517	0.488	0.857	0.845
AmnsRetain	OxfordFlowers	not ZS	0.848	0.059	0.659	0.922	0.558	0.553	0.517	0.51	0.857	0.866

need to be close to the probability of the real samples, otherwise there might be some discrepancy. For example, in Tab. 4 in the 1st row 2009 Bentley Arnage Sedan class maintains a high accuracy on *Synth. train* data despite *Target Class*

Table 2. Forgetting results with **synthetic** data using Lipschitz loss. We show the forgetting results on three classes for four different datasets.

Dataset	Class name	Target		Other		Target		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers			
		Class acc.	Classes acc.	Class acc.	Classes acc.	Synt.	Real	BF	AF	BF	AF	BF	AF	BF	AF	BF	AF
StanfordCars	Pekinese	0.705	0.066	0.515	0.514	0.045	0.558	0.559	-	-	0.857	0.867	0.661	0.658			
StanfordCars	toy poodle	0.574	0.033	0.516	0.506	0.022	0.558	0.565	-	-	0.857	0.867	0.661	0.647			
StanfordCars	Scotch terrier	0.3	0.047	0.517	0.516	0.047	0.083	0.558	0.557	-	-	0.857	0.865	0.661	0.66		
StanfordCars	2009 Spyker C8 Coupe	0.262	0.024	0.559	0.553	0.0	0.0	-	-	0.517	0.518	0.857	0.865	0.661	0.66		
StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.143	0.558	0.544	0.0	0.0	-	-	0.517	0.502	0.857	0.845	0.661	0.638		
StanfordCars	2011 Ford Ranger SuperCab	0.524	0.0	0.558	0.553	0.0	0.0	-	-	0.517	0.52	0.857	0.869	0.661	0.661		
Caltech101	euphonium	0.789	0.0	0.858	0.868	0.016	0.0	0.558	0.557	0.517	0.52	-	-	0.661	0.658		
Caltech101	minaret	0.826	0.043	0.857	0.863	0.0	0.067	0.558	0.558	0.517	0.515	-	-	0.661	0.661		
Caltech101	platypus	0.9	0.2	0.857	0.866	0.062	0.286	0.558	0.557	0.517	0.524	-	-	0.661	0.653		
OxfordFlowers	gazania	0.957	0.0	0.658	0.649	0.062	0.0	0.558	0.559	0.517	0.513	0.857	0.869	-	-		
OxfordFlowers	tree mallow	1.0	0.0	0.658	0.643	0.047	0.0	0.558	0.557	0.517	0.51	0.857	0.869	-	-		
OxfordFlowers	trumpet creeper	0.588	0.0	0.661	0.643	0.047	0.083	0.558	0.557	0.517	0.503	0.857	0.866	-	-		

Table 3. Forgetting results with **real** data using Lipschitz loss. We show the forgetting results on three classes for four different datasets.

Dataset	Class name	Target		Other		Target		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers		
		Class acc.	Classes acc.	Class acc.	Classes acc.	Synt.	Real	BF	AF	BF	AF	BF	AF	BF	AF	BF
StanfordCars	Pekinese	0.705	0.066	0.515	0.514	0.045	0.558	0.563	-	-	0.857	0.873	0.661	0.654		
StanfordCars	toy poodle	0.574	0.033	0.516	0.506	0.022	0.558	0.565	-	-	0.857	0.871	0.661	0.646		
StanfordCars	Scotch terrier	0.5	0.016	0.517	0.509	0.043	0.558	0.562	-	-	0.857	0.862	0.661	0.654		
StanfordCars	2009 Spyker C8 Coupe	0.262	0.024	0.559	0.517	0.059	-	-	0.517	0.509	0.857	0.849	0.661	0.642		
StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.024	0.558	0.557	0.0	-	-	0.517	0.52	0.857	0.866	0.661	0.655		
StanfordCars	2011 Ford Ranger SuperCab	0.524	0.048	0.558	0.561	0.029	-	-	0.517	0.521	0.857	0.87	0.661	0.658		
Caltech101	euphonium	0.789	0.0	0.858	0.853	0.0	0.558	0.549	0.517	0.495	-	-	0.661	0.625		
Caltech101	minaret	0.826	0.043	0.857	0.865	0.026	0.558	0.563	0.517	0.51	-	-	0.661	0.657		
Caltech101	platypus	0.9	0.5	0.857	0.866	0.235	0.558	0.559	0.517	0.517	-	-	0.661	0.652		
OxfordFlowers	gazania	0.957	0.0	0.658	0.65	0.026	0.558	0.561	0.517	0.515	0.857	0.866	-	-		
OxfordFlowers	tree mallow	1.0	0.353	0.658	0.652	0.0	0.558	0.56	0.517	0.512	0.857	0.861	-	-		
OxfordFlowers	trumpet creeper	0.588	0.059	0.661	0.658	0.069	0.558	0.56	0.517	0.515	0.857	0.864	-	-		

Table 4. Forgetting verification discrepancy examples.

Discrepancy	Dataset	Class name	Target		Other		Target		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers			
			Class acc.	Classes acc.	Class acc.	Classes acc.	Synt.	Real	train valid.	BF	AF	BF	AF	BF	AF	BF	AF	
Discrepancy	Caltech101	revolver	0.96	0.92	0.856	0.862	0.016	0.875	0.558	0.552	0.517	0.507	-	-	0.661	0.632		
Discrepancy	StanfordCars	2009 Bentley Arnage Sedan	0.692	0.051	0.557	0.538	0.922	0.0	-	-	0.517	0.527	0.857	0.875	0.661	0.644		
No Discrepancy	Caltech101	revolver	0.96	0.16	0.856	0.855	0.016	0.0	0.558	0.524	0.517	0.48	-	-	0.661	0.604		
No Discrepancy	StanfordCars	2009 Bentley Arnage Sedan	0.692	0.026	0.557	0.543	0.031	0.0	-	-	0.517	0.511	0.857	0.852	0.661	0.655		

acc. AF dropping to 5%. Conversely, in the 2nd row we observe that for *revolver* class the accuracy on *Synt. train* subset is low, yet forgetting was not successful as indicated by the true validation accuracy *Target Class acc.* AF *Real valid.*. It turns out that for *revolver* class the probability of the generated samples is too low while for 2009 Bentley Arnage Sedan is too high. Thus, by generating synthetic samples with class predicted probability closer to that of real samples we reduce the discrepancy as demonstrated in the last two rows of Tab.4. Note that generating higher probability synthetic samples compared to probability of real samples would still suffice to forget the class but not to verify the forgetting success. A simple alternative is to rely on a small real validation subset of the class to forget or if available, stored past probabilities of the class prediction. This verification can be conducted by a user if they are unwilling to share their data with the company.

5.7. Forgetting on Multiple Classes

We assess forgetting on multiple classes in Tab. 5. *Other Classes acc.* represents the performance on the remaining classes before forgetting (BF) and after forgetting (AF) respectively on the dataset specified in *Dataset*. We observe that after forgetting multiple classes, the accuracy on remaining classes and other datasets remains relatively high on StanfordDog and Caltech101 while reducing slightly more on other two datasets. This is due to the 1.5% loss in accuracy on *2010 Dodge Ram Pickup 3500 Crew Cab* and 2% on *trumpet creeper* alone as shown in Tab. 2 leading to more forgetting subsequently on other classes. This does not happen on StanfordDogs and Caltech101 datasets where the reduction in performance after forgetting on not targeted classes is low. Indeed, replacing *2010 Dodge Ram Pickup 3500 Crew Cab* with *2012 Rolls-Royce Ghost Sedan* that has a smaller reduction in performance on other classes after forgetting as shown in Tab.6, forgetting on multiple classes also improves. These are shown on the last row in Tab.6. In general, it is normal to expect that as the number of classes to forget increases, the accuracy on other classes will gradually decrease, as demonstrated in our experiments, albeit at a relatively slow rate. Additional analysis in the Appendix.

6. Ablations & Additional Tasks

In this section we present ablations and results on additional tasks. Extra ablations can be found in the Appendix.

6.1. Textual Loss Ablation

In Tab. 7 we compare ULip (*ExtraTextLoss* ablation type) method and our LIP method (*Original* ablation type) using synthetic data. The only difference between these methods is the inclusion of an additional textual loss in the LIP method. The results demonstrate the critical importance of incorporating both visual and textual losses for ef-

Table 5. Forgetting multiple classes with generated data using Lipschitz loss.

Dataset	Classes	Avg. Target		Other		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
		Classes acc.		Classes acc.		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
		BF	AF	BF	AF	BF	AF	BF	AF	BF	AF	BF	AF
StanfordCars	Pekinese,toy poodle,Scotch terrier	0.593	0.09	0.517	0.507	0.558	0.547	-	-	0.857	0.865	0.661	0.633
StanfordCars	2009 Spyker C8 Coupe, 2010 Dodge Ram Pickup 3500 Crew Cab, 2011 Ford Ranger SuperCab	0.397	0.2	0.558	0.519	-	-	0.517	0.482	0.857	0.84	0.661	0.607
Caltech101	cuckoo,minear,platypus	0.839	0.125	0.857	0.869	0.558	0.549	0.517	0.515	-	-	0.661	0.633
OxfordFlowers	gazania,tree mallow,trumpet creeper	0.848	0.0	0.661	0.609	0.558	0.552	0.517	0.498	0.857	0.863	-	-

Table 6. Forgetting other examples using Lipschitz loss.

Forgetting Data	Dataset	Class name	Target		Other		Target		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
			Class acc.		Classes acc.		Class acc.		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
			BF	AF	BF	AF	Synt.	Real	BF	AF	BF	AF	BF	AF	BF	AF
Generated	StanfordCars	2012 Chevrolet Avalanche Crew Cab	0.622	0.044	0.557	0.494	0.281	0.0	-	-	0.517	0.501	0.857	0.87	0.661	0.633
Generated	StanfordCars	2012 Rolls-Royce Ghost Sedan	0.526	0.07	0.557	0.557	0.078	0.15	-	-	0.520	0.527	0.857	0.862	0.66	0.658
Generated	StanfordCars	2009 Spyker C8 Coupe, 2012 Rolls-Royce Ghost Sedan, 2011 Ford Ranger SuperCab	0.397	0.12	0.558	0.535	-	-	0.517	0.501	0.857	0.862	0.661	0.631	-	-

fective forgetting in CLIP, as ULip forgetting with synthetic data proves to be highly ineffective.

Table 7. Different ablations. *Original*: results with our method (Lip) from Tab.1. *AllParamsVary*: forgetting on synthetic data allowing all parameters to vary. *OneIter*: forgetting on synthetic data with a single epoch. *ExtraTextLoss*: ULip forgetting on synthetic data.

Ablation Type	Method	Dataset	Avg. Target		Avg. Other		Avg.		Avg.		Avg.			
			Class acc.	BF AF	Classes acc.	StanfordCars	StanfordDogs	Caltech101	OxfordFlowers	BF AF	BF AF	BF AF		
Original	Lip	StanfordCars	0.397	0.056	0.558	0.551	-	-	0.517	0.513	0.857	0.86	0.661	0.653
Original	Lip	StanfordDogs	0.593	0.048	0.516	0.516	0.558	0.558	-	-	0.857	0.866	0.661	0.655
Original	Lip	Caltech101	0.839	0.081	0.857	0.865	0.558	0.557	0.517	0.52	-	-	0.661	0.657
Original	Lip	OxfordFlowers	0.848	0.0	0.659	0.645	0.558	0.557	0.517	0.51	0.857	0.868	-	-
ExtraTextLoss	ULip	StanfordCars	0.397	0.222	0.558	0.545	-	-	0.517	0.504	0.857	0.849	0.661	0.648
ExtraTextLoss	ULip	StanfordDogs	0.593	0.534	0.516	0.512	0.558	0.558	-	-	0.857	0.856	0.661	0.651
ExtraTextLoss	ULip	Caltech101	0.839	0.859	0.857	0.855	0.558	0.557	0.517	0.515	-	-	0.661	0.655
ExtraTextLoss	ULip	OxfordFlowers	0.848	0.809	0.659	0.654	0.558	0.556	0.517	0.513	0.857	0.857	-	-
AllParamsVary	Lip	StanfordCars	0.397	0.0	0.558	0.532	-	-	0.517	0.018	0.857	0.114	0.661	0.008
AllParamsVary	Lip	StanfordDogs	0.593	0.0	0.516	0.009	0.558	0.013	-	-	0.857	0.074	0.661	0.011
AllParamsVary	Lip	Caltech101	0.839	0.0	0.857	0.073	0.558	0.014	0.517	0.012	-	-	0.661	0.013
AllParamsVary	Lip	OxfordFlowers	0.848	0.0	0.659	0.019	0.558	0.017	0.517	0.013	0.857	0.086	-	-
OneIter	Lip	StanfordCars	0.397	0.008	0.558	0.556	-	-	0.517	0.524	0.857	0.87	0.661	0.665
OneIter	Lip	StanfordDogs	0.593	0.201	0.516	0.527	0.558	0.56	-	-	0.857	0.87	0.661	0.666
OneIter	Lip	Caltech101	0.845	0.405	0.857	0.866	0.558	0.56	0.517	0.521	-	-	0.661	0.662
OneIter	Lip	OxfordFlowers	0.848	0.189	0.659	0.653	0.558	0.562	0.517	0.519	0.857	0.867	-	-

6.2. Varying All the Parameters

When all parameters are allowed to vary without employing selective forgetting, CLIP tends to overforget. These findings are shown in Tab.7 for *AllParamsVary* ablation type. Therefore, greater control over the varying parameters is necessary in CLIP to achieve forgetting. This differs from the approach used with vision models in [7], where all parameters were allowed to vary. This discrepancy can be attributed to the size of the CLIP model, which contains a vast amount of information unlike the smaller models trained on limited data in [7], as well as the structural differences that necessitate more controlled updates.

6.3. Forgetting Using One Iteration

In [7] authors used a single epoch for forgetting. In Tab. 7 in *OneIter* ablation type we can clearly see that this is often not enough to forget as the model most of the time still maintains a very high accuracy on the class to forget. Thus, multiple iterations are required to achieve a desirable level of forgetting.

6.4. Synthetic Images Generation CuPL Template

We test how the text templates of the generated samples affect performance. We generate synthetic samples using templates from CuPL³, e.g. for the "Pekinese" class one example is "*The image is of a small, brown and white Pekinese dog with long, flowing fur.*". We generate 64 synthetic samples as before and for each generated sample a random description of the class from CuPL is used. CuPL descriptions involve not only the class itself but also features of the

³https://github.com/sarahpratt/CuPL/tree/main/all_prompts

class containing thus more information additionally to the class name. The results are shown in Tab. 8. We see that by changing the template of the synthetic samples generation the forgetting is still successful in breaking the image-text association for the class. However, because of additional features in the template that might be shared among other classes such as *"flowing fur"* the remaining accuracy slightly decreases. This makes us conclude that forgetting can be sensitive to the template used to generate synthetic samples. Note that the evaluation is still done using the standard template.

Table 8. Aggregated forgetting results. We perform forgetting on synthetic samples generated with randomly selected CuPL templates.

Method Dataset	Avg. Target		Avg. Other		Avg.		Avg.		Avg.			
	Class acc.		Classes acc.		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
	BF	AF	BF	AF	BF	AF	BF	AF	BF	AF	BF	AF
Lip StanfordCars	0.397	0.032	0.558	0.525	-	-	0.517	0.494	0.857	0.844	0.661	0.635
Lip StanfordDogs	0.593	0.066	0.516	0.478	0.558	0.532	-	-	0.857	0.838	0.661	0.604
Lip Caltech101	0.839	0.014	0.857	0.859	0.558	0.548	0.517	0.508	-	-	0.661	0.637
Lip OxfordFlowers	0.848	0.039	0.659	0.575	0.558	0.525	0.517	0.48	0.857	0.831	-	-

6.5. Variation of Templates for Evaluation

In the following experiments we test the sensitivity of the models after forgetting to the change in the evaluation template. We use the synthetic samples generated with a standard template but evaluate using three different templates: *"We can see a {class} in this image"*, *"This is a representation of {class}"*, *"There is evidence of a {class} in the picture"*. This shows how sensitive for evaluation the model is to the change in template after forgetting. Note that because the evaluation template changed, so did the zero-shot classification accuracy before forgetting on CLIP. The results are shown in Tab. 9 where we observe that even after changing the evaluation template forgetting is still valid across the classes we have forgotten.

Table 9. Aggregated forgetting results. We aggregate across three different evaluation templates to assess sensitivity of models after forgetting to the change in evaluation template.

Method Dataset	Avg. Target		Avg. Other		Avg.		Avg.		Avg.			
	Class acc.		Classes acc.		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
	BF	AF	BF	AF	BF	AF	BF	AF	BF	AF	BF	AF
Lip StanfordCars	0.272	0.048	0.493	0.494	-	-	0.412	0.413	0.81	0.802	0.518	0.512
Lip StanfordDogs	0.306	0.103	0.416	0.397	0.492	0.491	-	-	0.81	0.795	0.518	0.504
Lip Caltech101	0.897	0.211	0.809	0.809	0.492	0.499	0.412	0.422	-	-	0.518	0.547
Lip OxfordFlowers	0.698	0.187	0.518	0.51	0.492	0.493	0.412	0.41	0.81	0.805	-	-

6.6. Additional Tasks

We evaluate our method for the retrieval task in addition to classification. The retrieval task involves three scenarios: text retrieval from the image input, image retrieval from the text input, and image retrieval from the image input. Since the classification task can be viewed as text retrieval from

Table 10. Retrieval task results showing precision@k for k of 1, 5, 10.

Retrieval Type	Model	Precision@1 (↓)	Precision@5 (↓)	Precision@10 (↓)
IfT	CLIP original Avg.	0.833	0.683	0.583
IfT	CLIP forget Avg.	0.08	0.23	0.191
IfI	CLIP original Avg.	0.417	0.367	0.317
IfI	CLIP forget Avg.	0.333	0.367	0.325

an image, we present aggregated results for the other two retrieval tasks in Fig. 10, with full results provided in the Appendix.

Image Retrieval from Text Input (IfT) We create a database from 4 datasets and perform image retrieval task from the database given a text input. We evaluate on precision@k metric measuring the proportion of retrieved items that are relevant among top K retrieved items. This indicates the accuracy of the retrieved results. We perform our experiments with k of 1, 5 and 10. Note that the lower the precision@k the better. We see in Tab. 10 that the model is most of the times unable to retrieve images from input text.

Image Retrieval from Image Input (IfI) Similarly to above, but now we test image-image retrieval. We observe in Tab. 10 that image representation for the forget objects is mainly untouched and the model is able to find also forget classes. These results indicate that forgetting is achieved breaking the multimodal link but unimodal information still remains in the model. This was surprising and we ask whether our forgetting is successful given these results? Therefore, we checked whether the original CLIP is able to retrieve images from image input for classes the model is unable to classify, i.e. has classification accuracy of 0. In Appendix H we show that it is the case the model can still identify similar features and shapes of objects without knowing the textual class. Thus, we conclude that for class forgetting, breaking text-image association is sufficient.

7. Conclusions

In this work we have successfully achieved class forgetting without losing knowledge on other classes in the multimodal setting of CLIP. Our experiments were conducted on four standard datasets, demonstrating that forgetting can be achieved based solely on the textual class names by generating synthetic samples of the class, without dependence on real data, thus achieving true zero-shot forgetting. Our forgetting process is iterative where we increase the number of layers to update and the strength of perturbations based on the reduction in accuracy of synthetic training data.

References

- [1] A methodology for formalizing model-inversion attacks — ieee conference publication — ieee xplore. 1
- [2] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *ICML*, 03 2013. 1
- [3] Jiali Cheng and Hadi Amiri. Multimodal machine unlearning, 11 2023. 2
- [4] 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. 2, 3, 4, 6
- [5] 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. 5
- [6] 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. 4
- [7] Jack Foster, Kyle Fogarty, Stefan Schoepf, Cengiz Öztireli, and Alexandra Brintrup. Zero-shot machine unlearning at scale via lipschitz regularization. 02 2024. 2, 3, 4, 6, 7
- [8] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models, 06 2023. 2
- [9] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning, 10 2020. 2, 4, 6
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 12 2015. 5
- [11] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *NeurIPS*, 07 2022. 2
- [12] 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. 4
- [13] 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, page 554–561, 2013. 4
- [14] M. Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 4
- [15] 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. 1
- [16] 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, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 8748–8763. PMLR, 18–24 Jul 2021. 1
- [17] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2013. 2, 4
- [18] 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. 2
- [19] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1), aug 2023. 1
- [20] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *ICLR*, 05 2017. 3
- [21] 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. 2

A. ResNet Full Results

Table 11. Forgetting results with ResNet visual encoder. We compare our methods with five others on three classes for four selected datasets.

Method	Dataset	Class name	Target Class acc.		Other Classes acc.		Target Class acc.		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
			BF	AF	BF	AF	Synt.	Real train valid.	BF	AF	BF	AF	BF	AF	BF	AF
Lip	StanfordDogs	Pekinese	0.705	0.066	0.515	0.514	0.062	0.0	0.558	0.559	-	-	0.857	0.867	0.661	0.658
Lip	StanfordDogs	toy poodle	0.574	0.033	0.516	0.518	0.031	0.0	0.558	0.559	-	-	0.857	0.867	0.661	0.647
Lip	StanfordDogs	Scotch terrier	0.5	0.047	0.517	0.516	0.047	0.083	0.558	0.557	-	-	0.857	0.865	0.661	0.66
Lip	StanfordCars	2009 Spyker C8 Coupe	0.262	0.024	0.559	0.553	0.0	0.0	-	-	0.517	0.518	0.857	0.865	0.661	0.66
Lip	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.143	0.558	0.544	0.0	0.0	-	-	0.517	0.502	0.857	0.845	0.661	0.638
Lip	StanfordCars	2011 Ford Ranger SuperCab	0.524	0.0	0.558	0.555	0.0	0.0	-	-	0.517	0.52	0.857	0.869	0.661	0.661
Lip	Caltech101	euphonium	0.789	0.0	0.588	0.568	0.016	0.0	0.558	0.557	0.517	0.52	-	-	0.661	0.658
Lip	Caltech101	minaret	0.826	0.043	0.857	0.863	0.0	0.067	0.558	0.556	0.517	0.515	-	-	0.661	0.661
Lip	Caltech101	platypus	0.9	0.2	0.857	0.866	0.062	0.286	0.558	0.558	0.517	0.524	-	-	0.661	0.653
Lip	OxfordFlowers	gazania	0.957	0.0	0.658	0.649	0.062	0.0	0.558	0.559	0.517	0.513	0.857	0.869	-	-
Lip	OxfordFlowers	tree mallow	1.0	0.0	0.658	0.643	0.047	0.0	0.558	0.557	0.517	0.51	0.857	0.869	-	-
Lip	OxfordFlowers	trumpet creeper	0.588	0.0	0.661	0.643	0.047	0.083	0.558	0.557	0.517	0.503	0.857	0.866	-	-
Emb	StanfordDogs	Pekinese	0.705	0.361	0.515	0.484	0.0	0.318	0.558	0.559	-	-	0.857	0.84	0.661	0.633
Emb	StanfordDogs	toy poodle	0.574	0.361	0.516	0.481	0.0	0.217	0.558	0.553	-	-	0.857	0.832	0.661	0.613
Emb	StanfordDogs	Scotch terrier	0.5	0.062	0.517	0.472	0.031	0.083	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	0.0	0.0	-	-	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	0.047	0.0	-	-	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	0.0	0.125	-	-	0.517	0.509	0.857	0.852	0.661	0.654
Emb	Caltech101	euphonium	0.789	0.263	0.588	0.833	0.0	0.385	0.558	0.548	0.517	0.506	-	-	0.661	0.616
Emb	Caltech101	minaret	0.826	0.13	0.857	0.827	0.0	0.133	0.558	0.54	0.517	0.507	-	-	0.661	0.639
Emb	Caltech101	platypus	0.9	0.0	0.857	0.829	0.047	0.0	0.558	0.549	0.517	0.49	-	-	0.661	0.597
Emb	OxfordFlowers	gazania	0.957	0.739	0.658	0.632	0.0	0.875	0.558	0.551	0.517	0.503	0.857	0.849	-	-
Emb	OxfordFlowers	tree mallow	1.0	0.353	0.658	0.612	0.094	0.583	0.558	0.554	0.517	0.504	0.857	0.849	-	-
Emb	OxfordFlowers	trumpet creeper	0.588	0.235	0.661	0.632	0.0	0.167	0.558	0.555	0.517	0.508	0.857	0.853	-	-
Amns	StanfordDogs	Pekinese	0.705	0.459	0.515	0.486	0.0	0.409	0.558	0.561	-	-	0.857	0.847	0.661	0.65
Amns	StanfordDogs	toy poodle	0.574	0.492	0.516	0.423	0.016	0.261	0.558	0.55	-	-	0.857	0.839	0.661	0.628
Amns	StanfordDogs	Scotch terrier	0.5	0.031	0.517	0.488	0.0	0.083	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	0.016	0.0	-	-	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	0.047	0.222	-	-	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	0.078	0.375	-	-	0.517	0.507	0.857	0.868	0.661	0.656
Amns	Caltech101	euphonium	0.789	0.316	0.588	0.856	0.016	0.385	0.558	0.557	0.517	0.519	-	-	0.661	0.655
Amns	Caltech101	minaret	0.826	0.174	0.857	0.813	0.031	0.267	0.558	0.546	0.517	0.493	-	-	0.661	0.591
Amns	Caltech101	platypus	0.9	0.5	0.857	0.832	0.0	0.571	0.558	0.555	0.517	0.495	-	-	0.661	0.634
Amns	OxfordFlowers	gazania	0.957	0.87	0.658	0.595	0.391	0.875	0.558	0.557	0.517	0.489	0.857	0.834	-	-
Amns	OxfordFlowers	tree mallow	1.0	0.0	0.658	0.598	0.094	0.0	0.558	0.511	0.517	0.476	0.857	0.843	-	-
Amns	OxfordFlowers	trumpet creeper	0.588	0.294	0.661	0.584	0.0	0.25	0.558	0.554	0.517	0.494	0.857	0.828	-	-
AmnsRetain	StanfordDogs	Pekinese	0.705	0.049	0.515	0.667	-	-	0.558	0.531	-	-	0.857	0.835	0.661	0.61
AmnsRetain	StanfordDogs	toy poodle	0.574	0.082	0.516	0.663	-	-	0.558	0.521	-	-	0.857	0.831	0.661	0.605
AmnsRetain	StanfordDogs	Scotch terrier	0.5	0.0	0.517	0.659	-	-	0.558	0.511	-	-	0.857	0.847	0.661	0.616
AmnsRetain	StanfordCars	2009 Spyker C8 Coupe	0.262	0.071	0.559	0.719	-	-	-	-	0.517	0.513	0.857	0.886	0.661	0.623
AmnsRetain	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.0	0.558	0.712	-	-	-	-	0.517	0.504	0.857	0.878	0.661	0.62
AmnsRetain	StanfordCars	2011 Ford Ranger SuperCab	0.524	0.048	0.558	0.704	-	-	-	-	0.517	0.51	0.857	0.879	0.661	0.622
AmnsRetain	Caltech101	euphonium	0.789	0.0	0.588	0.926	-	-	0.558	0.537	0.517	0.506	-	-	0.661	0.624
AmnsRetain	Caltech101	minaret	0.826	0.0	0.857	0.927	-	-	0.558	0.516	0.517	0.501	-	-	0.661	0.63
AmnsRetain	Caltech101	platypus	0.9	0.0	0.857	0.923	-	-	0.558	0.526	0.517	0.508	-	-	0.661	0.654
AmnsRetain	OxfordFlowers	gazania	0.957	0.0	0.658	0.931	-	-	0.558	0.554	0.517	0.512	0.857	0.865	-	-
AmnsRetain	OxfordFlowers	tree mallow	1.0	0.0	0.658	0.931	-	-	0.558	0.552	0.517	0.503	0.857	0.872	-	-
AmnsRetain	OxfordFlowers	trumpet creeper	0.588	0.176	0.661	0.903	-	-	0.558	0.553	0.517	0.515	0.857	0.861	-	-
EMMN	StanfordDogs	Pekinese	0.705	0.033	0.515	0.168	-	-	0.558	0.38	-	-	0.857	0.846	0.661	0.489
EMMN	StanfordDogs	toy poodle	0.574	0.082	0.516	0.21	-	-	0.558	0.376	-	-	0.857	0.852	0.661	0.482
EMMN	StanfordDogs	Scotch terrier	0.5	0.0	0.517	0.138	-	-	0.558	0.374	-	-	0.857	0.836	0.661	0.467
EMMN	StanfordCars	2009 Spyker C8 Coupe	0.262	0.024	0.559	0.173	-	-	-	-	0.517	0.312	0.857	0.815	0.661	0.378
EMMN	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.0	0.558	0.18	-	-	-	-	0.517	0.331	0.857	0.8	0.661	0.4
EMMN	StanfordCars	2011 Ford Ranger SuperCab	0.524	0.5	0.558	0.19	-	-	-	-	0.517	0.298	0.857	0.801	0.661	0.386
EMMN	Caltech101	euphonium	0.789	0.0	0.588	0.701	-	-	0.558	0.426	0.517	0.365	-	-	0.661	0.467
EMMN	Caltech101	minaret	0.826	0.043	0.857	0.694	-	-	0.558	0.419	0.517	0.39	-	-	0.661	0.464
EMMN	Caltech101	platypus	0.9	0.5	0.857	0.687	-	-	0.558	0.422	0.517	0.375	-	-	0.661	0.453
EMMN	OxfordFlowers	gazania	0.957	0.0	0.658	0.314	-	-	0.558	0.429	0.517	0.441	0.857	0.858	-	-
EMMN	OxfordFlowers	tree mallow	1.0	0.0	0.658	0.267	-	-	0.558	0.413	0.517	0.42	0.857	0.836	-	-
EMMN	OxfordFlowers	trumpet creeper	0.588	0.941	0.661	0.375	-	-	0.558	0.44	0.517	0.446	0.857	0.856	-	-
ULip	StanfordDogs	Pekinese	0.705	0.705	0.515	0.506	-	-	0.558	0.563	-	-	0.857	0.863	0.661	0.656
ULip	StanfordDogs	toy poodle	0.574	0.082	0.516	0.395	-	-	0.558	0.494	-	-	0.857	0.81	0.661	0.611
ULip	StanfordDogs	Scotch terrier	0.5	0.5	0.517	0.509	-	-	0.558	0.561	-	-	0.857	0.853	0.661	0.657
ULip	StanfordCars	2009 Spyker C8 Coupe	0.262	0.119	0.559	0.389	-	-	-	-	0.517	0.492	0.857	0.825	0.661	0.63
ULip	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.405	0.071	0.558	0.497	-	-	-	-	0.517	0.51	0.857	0.859	0.661	0.643
ULip	StanfordCars	2011 Ford Ranger SuperCab	0.524	0.19	0.558	0.486	-	-	-	-	0.517	0.504	0.857	0.86	0.661	0.644
ULip	Caltech101	euphonium	0.789	0.316	0.588	0.849	-	-	0.558	0.556	0.517	0.5	-	-	0.661	0.64
ULip	Caltech101	minaret	0.826	0.783	0.857	0.855	-	-	0.558	0.564	0.517	0.514	-	-	0.661	0.659
ULip	Caltech101	platypus	0.9	0.9	0.857	0.858	-	-	0.558	0.561	0.517	0.513	-	-	0.661	0

B. Additional Ablations

B.1. Perturbation to Text Embedding

We analyze how adding noise to the text embeddings, which are in continuous space, rather than using image input changes the forgetting results. Comparing the results in Tab. 12 for *TextEmbPeturb* ablation type and *Original* we see that perturbing textual embedding makes forgetting slightly worse, which is likely due to the fact that perturbing textual embeddings is less meaningful compared to image pixels or discrete text tokens.

Table 12. Ablations on perturbation to text embedding.

Ablation Type	Dataset	Avg. Target		Avg. Other		Avg.		Avg.		Avg.			
		Class acc.	Classes acc.	BF	AF	BF	AF	BF	AF	BF	AF		
Original	StanfordCars	0.397	0.056	0.558	0.551	-	-	0.517	0.513	0.857	0.86	0.661	0.653
Original	StanfordDogs	0.593	0.048	0.516	0.516	0.558	0.558	-	-	0.857	0.866	0.661	0.655
Original	Caltech101	0.839	0.081	0.857	0.865	0.558	0.557	0.517	0.52	-	-	0.661	0.657
Original	OxfordFlowers	0.848	0.0	0.659	0.645	0.558	0.557	0.517	0.51	0.857	0.868	-	-
TextEmbPeturb	StanfordCars	0.397	0.056	0.558	0.55	-	-	0.517	0.513	0.857	0.859	0.661	0.652
TextEmbPeturb	StanfordDogs	0.593	0.267	0.516	0.509	0.558	0.557	-	-	0.857	0.864	0.661	0.648
TextEmbPeturb	Caltech101	0.839	0.114	0.857	0.866	0.558	0.558	0.517	0.515	-	-	0.661	0.652
TextEmbPeturb	OxfordFlowers	0.848	0.02	0.659	0.631	0.558	0.553	0.517	0.503	0.857	0.867	-	-

C. Error Analysis Multiple Classes

In Tab. 5 of the main paper, we presented the performance after forgetting across multiple classes. Notably, on the *OxfordFlowers* dataset, the accuracy decrease was more pronounced compared to other datasets when forgetting was applied, indicating higher sensitivity. This observation held true for single class forgetting as well. Therefore, we examined how close in terms of their position in the sorted list of logits scores the correct prediction of the original model and incorrect prediction of the forget model were before and after forgetting. If their proximity in terms of position and logits scores was significant, it would suggest that the model was already uncertain about those predictions. Consequently, a minor, targeted forgetting may have resulted in a subtle change in logits scores, swapping the predictions.

We conducted this analysis on the model after selected classes forgetting specified in Tab. 5 column *Classes*. Following the forgetting of three classes from *Caltech101* datasets, there were 106 incorrect predictions after forgetting compared to the model’s performance before forgetting on the *OxfordFlowers* dataset. Among these, we found that in 88 instances (83.02%), the correct class in an ordered set of logits predicted by CLIP shifted by one place. This means that instead of the correct class having the highest score, it ranked second-highest after forgetting. At the same time, to be included in those 88 instances the new highest incorrect prediction of the forget model must have been the second highest class in the original model’s logits.

To explain this procedure with an example, consider a given image of a *Poodle* for which the original model predicted the classes in the following way: *Poodle*, *Labrador*, *Spaniel* ordered by logits assumed to be 15, 14.9, 12. Here, *Poodle* was the correct prediction with the highest score, as we only examined cases where the original model was correct and assessed how that changed after forgetting. After the forgetting process, the new model predicts: *Labrador*, *Poodle*, *Spaniel* with sorted scores 15.1, 14.9, 11.8. In this case, the correct prediction (*Poodle*) moved from the highest to second highest score, i.e., one step away from its original position, and the difference in scores is 0.02 (corresponding to *One-Step Avg. Score* in Tab. 13). Also, the new incorrect prediction with the highest score, *Labrador*, was the second highest prediction in the original model logits, thus this example would be included in calculating the *One-Step %*.

Table 13. Sensitivity Analysis on OxfordFlowers dataset.

Model	One-Step %	Δ One-Step Avg. Score	Two-Step %	Δ Two-Step Avg. Score
StanfordDogs	83.02	0.014	93.4	0.014
StanfordCars	83.87	0.015	93.54	0.016
Caltech101	83.33	0.015	93.33	0.017

Similarly, looking at shifts of up to two places, where the correct class ranked either second or third highest, we found that 99 cases (93.4%) moved away by a maximum of two places from the correct prediction.

For *StanfordDogs*, there were 120 incorrect labels. In 100 cases (83.33%), the correct class moved away by one place, and in 112 cases (93.33%), it shifted by a maximum of two places.

For *StanfordCars*, we took the model after forgetting on classes shown on the last row of Tab. 6. There were 124 incorrect labels. In 104 cases (83.87%), the correct class moved away by one place, and in 116 cases (93.54%), it shifted by a maximum of two places.

These results are summarized in Table 13. The Δ *One-Step Avg. Score* represents the standardized average change in logits scores between the model’s new incorrect and correct predictions. Similarly, for Δ *Two-Step Avg. Score*, considering a shift of two places.

This analysis indicates that the greater than expected drop in accuracy on the *OxfordFlowers* dataset is attributed to the model’s original uncertainty about those cases. Thus, despite our demonstrated ability to precisely target the parameters for forgetting the target class, the model’s initial uncertainty contributes to the relatively more pronounced decrease in accuracy.

D. Synthetic Images Visualization

In Fig. 2 we can see an example of a synthetic image for a class from four different datasets we tested on. These synthetic samples do not have a clear appearance of the sample class but are enough for unlearning the class.

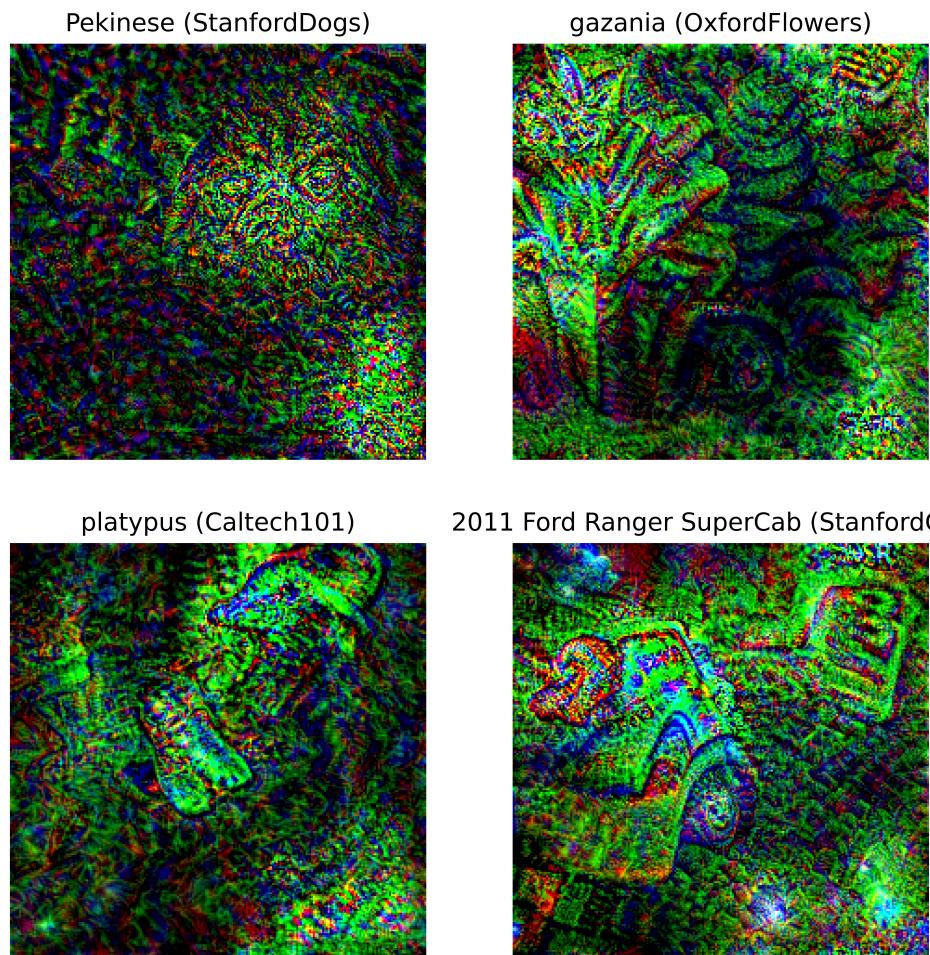


Figure 2. Synthetic images examples.

E. Predictions Before and After Forgetting on the Target Class

In the Fig. 3 we present examples of the model’s predictions before (BF) and after forgetting (AF). It’s evident that post-forgetting, the model still predicts classes that closely resemble the correct ones, indicating that its general understanding of similar classes remains intact. This suggests that our method effectively targets specific knowledge of the model to remove detailed knowledge about the target class while preserving broader knowledge.

BF: 2011 Ford Ranger SuperCab
AF: 2009 Dodge Ram Pickup 3500 Quad Cab



BF: Pekinese
AF: Shih



BF: gazania
AF: marigold



BF: platypus
AF: beaver



Figure 3. Predictions of the model before (BF) and after forgetting (AF) with the prediction BF representing the target class to forget.

F. ViT Results

Results for CLIP with ViT-B/16 visual encoder are included in Tab. 14. In general we observe that forgetting with ViT compared to ResNet architecture is harder and the class accuracy after forgetting, even if decreases, it often does not go to zero. This difference may be attributed to the fact that ViT learns more fine-grained representations of the input images compared to ResNet thanks to its self-attention mechanism. As a result, specific classes could be encoded in a more intricate manner within ViT’s learned representations, making them harder to unlearn without affecting other aspects of the model’s knowledge. When it comes to other methods for comparison the conclusions are similar to the ones highlighted for ResNet. Aggregated results are shown in Tab. 15

Table 14. Forgetting results with ViT-B/16 visual encoder. We compare our methods with five others on three classes for four selected datasets.

Method	Dataset	Class name	Target Class acc.		Other Classes acc.		Target Class acc.		StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
			BF	AF	BF	AF	Synt. Real train valid.	BF	AF	BF	AF	BF	AF	BF	AF	BF
Lip	StanfordDogs	Pekinese	0.787	0.377	0.59	0.601	0.094	0.273	0.655	0.656	-	-	0.933	0.934	0.708	0.708
Lip	StanfordDogs	toy poodle	0.607	0.033	0.591	0.593	0.0	0.0	0.655	0.639	-	-	0.933	0.932	0.708	0.707
Lip	StanfordDogs	Scotch terrier	0.625	0.016	0.591	0.582	0.219	0.0	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	0.484	0.222	-	-	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	0.062	0.0	-	-	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	0.984	0.125	-	-	0.591	0.59	0.933	0.933	0.708	0.713
Lip	Caltech101	euphonium	1.0	0.158	0.933	0.935	0.0	0.154	0.655	0.653	0.591	0.597	-	-	0.708	0.706
Lip	Caltech101	minaret	0.913	0.87	0.933	0.932	0.031	1.0	0.655	0.649	0.591	0.59	-	-	0.708	0.709
Lip	Caltech101	platypus	1.0	0.7	0.933	0.93	0.031	0.857	0.655	0.653	0.591	0.595	-	-	0.708	0.711
Lip	OxfordFlowers	gazania	1.0	0.0	0.705	0.7	0.297	0.0	0.655	0.642	0.591	0.587	0.933	0.935	-	-
Lip	OxfordFlowers	tree mallow	0.765	0.176	0.707	0.699	0.172	0.0	0.655	0.65	0.591	0.596	0.933	0.933	-	-
Lip	OxfordFlowers	trumpet creeper	0.588	0.059	0.709	0.705	0.781	0.0	0.655	0.644	0.591	0.581	0.933	0.932	-	-
Emb	StanfordDogs	Pekinese	0.787	0.213	0.59	0.601	0.031	0.227	0.655	0.656	-	-	0.933	0.934	0.708	0.708
Emb	StanfordDogs	toy poodle	0.607	0.0	0.591	0.472	0.0	0.0	0.655	0.621	-	-	0.933	0.931	0.708	0.696
Emb	StanfordDogs	Scotch terrier	0.625	0.0	0.591	0.481	0.141	0.0	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	0.016	0.0	-	-	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	0.078	0.0	-	-	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	0.203	0.0	-	-	0.591	0.542	0.933	0.92	0.708	0.699
Emb	Caltech101	euphonium	1.0	0.368	0.933	0.935	0.0	0.385	0.655	0.652	0.591	0.594	-	-	0.708	0.709
Emb	Caltech101	minaret	0.913	0.826	0.933	0.933	0.016	0.867	0.655	0.635	0.591	0.583	-	-	0.708	0.711
Emb	Caltech101	platypus	1.0	0.6	0.933	0.861	0.0	0.429	0.655	0.539	0.591	0.376	-	-	0.708	0.547
Emb	OxfordFlowers	gazania	1.0	0.0	0.705	0.705	0.141	0.0	0.655	0.645	0.591	0.593	0.933	0.933	-	-
Emb	OxfordFlowers	tree mallow	0.765	0.0	0.707	0.577	0.172	0.0	0.655	0.58	0.591	0.501	0.933	0.903	-	-
Emb	OxfordFlowers	trumpet creeper	0.588	0.0	0.709	0.569	0.0	0.0	0.655	0.406	0.591	0.472	0.933	0.88	-	-
Amns	StanfordDogs	Pekinese	0.787	0.623	0.59	0.366	0.031	0.545	0.655	0.581	-	-	0.933	0.896	0.708	0.609
Amns	StanfordDogs	toy poodle	0.607	0.033	0.591	0.234	0.0	0.0	0.655	0.57	-	-	0.933	0.899	0.708	0.482
Amns	StanfordDogs	Scotch terrier	0.625	0.0	0.591	0.473	0.062	0.042	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	0.0	0.0	-	-	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	0.0	0.0	-	-	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	0.094	0.25	-	-	0.591	0.516	0.933	0.916	0.708	0.596
Amns	Caltech101	euphonium	1.0	1.0	0.933	0.901	0.406	0.923	0.655	0.648	0.591	0.57	-	-	0.708	0.639
Amns	Caltech101	minaret	0.913	0.739	0.933	0.774	0.094	0.6	0.655	0.336	0.591	0.257	-	-	0.708	0.366
Amns	Caltech101	platypus	1.0	0.8	0.933	0.868	0.078	0.714	0.655	0.566	0.591	0.507	-	-	0.708	0.594
Amns	OxfordFlowers	gazania	1.0	0.913	0.705	0.518	0.062	0.875	0.655	0.586	0.591	0.514	0.933	0.908	-	-
Amns	OxfordFlowers	tree mallow	0.765	0.824	0.707	0.484	0.094	0.75	0.655	0.593	0.591	0.513	0.933	0.91	-	-
Amns	OxfordFlowers	trumpet creeper	0.588	0.765	0.709	0.578	0.094	0.75	0.655	0.627	0.591	0.55	0.933	0.92	-	-
AmnsRetain	StanfordDogs	Pekinese	0.787	0.0	0.59	0.706	-	-	0.655	0.473	-	-	0.933	0.829	0.708	0.49
AmnsRetain	StanfordDogs	toy poodle	0.607	0.0	0.591	0.709	-	-	0.655	0.475	-	-	0.933	0.847	0.708	0.507
AmnsRetain	StanfordDogs	Scotch terrier	0.625	0.062	0.591	0.679	-	-	0.655	0.468	-	-	0.933	0.848	0.708	0.488
AmnsRetain	StanfordCars	2009 Spyker C8 Coupe	0.429	0.0	0.656	0.768	-	-	-	-	0.591	0.475	0.933	0.88	0.708	0.573
AmnsRetain	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548	0.0	0.656	0.742	-	-	-	-	0.591	0.494	0.933	0.89	0.708	0.549
AmnsRetain	StanfordCars	2011 Ford Ranger SuperCab	0.81	0.0	0.654	0.721	-	-	-	-	0.591	0.46	0.933	0.891	0.708	0.539
AmnsRetain	Caltech101	euphonium	1.0	0.0	0.933	0.945	-	-	0.655	0.493	0.591	0.46	-	-	0.708	0.556
AmnsRetain	Caltech101	minaret	0.913	0.0	0.933	0.938	-	-	0.655	0.494	0.591	0.468	-	-	0.708	0.532
AmnsRetain	Caltech101	platypus	1.0	0.0	0.933	0.956	-	-	0.655	0.53	0.591	0.493	-	-	0.708	0.543
AmnsRetain	OxfordFlowers	gazania	1.0	0.0	0.705	0.944	-	-	0.655	0.563	0.591	0.522	0.933	0.916	-	-
AmnsRetain	OxfordFlowers	tree mallow	0.765	0.0	0.707	0.934	-	-	0.655	0.558	0.591	0.528	0.933	0.913	-	-
AmnsRetain	OxfordFlowers	trumpet creeper	0.588	0.0	0.709	0.956	-	-	0.655	0.558	0.591	0.542	0.933	0.913	-	-
EMMN	StanfordDogs	Pekinese	0.787	0.0	0.59	0.376	-	-	0.558	0.278	-	-	0.857	0.828	0.661	0.432
EMMN	StanfordDogs	toy poodle	0.607	0.0	0.591	0.373	-	-	0.558	0.308	-	-	0.857	0.836	0.661	0.446
EMMN	StanfordDogs	Scotch terrier	0.625	0.125	0.591	0.347	-	-	0.558	0.265	-	-	0.857	0.813	0.661	0.436
EMMN	StanfordCars	2009 Spyker C8 Coupe	0.429	0.0	0.656	0.188	-	-	-	-	0.517	0.116	0.857	0.614	0.661	0.148
EMMN	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548	0.476	0.656	0.184	-	-	-	-	0.517	0.13	0.857	0.56	0.661	0.126
EMMN	StanfordCars	2011 Ford Ranger SuperCab	0.81	0.0	0.654	0.175	-	-	-	-	0.517	0.111	0.857	0.594	0.661	0.136
EMMN	Caltech101	euphonium	1.0	0.105	0.933	0.783	-	-	0.558	0.352	0.517	0.297	-	-	0.661	0.45
EMMN	Caltech101	minaret	0.913	0.348	0.933	0.817	-	-	0.558	0.36	0.517	0.315	-	-	0.661	0.485
EMMN	Caltech101	platypus	1.0	0.4	0.933	0.838	-	-	0.558	0.345	0.517	0.294	-	-	0.661	0.484
EMMN	OxfordFlowers	gazania	1.0	0.0	0.705	0.44	-	-	0.558	0.308	0.517	0.312	0.857	0.832	-	-
EMMN	OxfordFlowers	tree mallow	0.765	0.0	0.707	0.445	-	-	0.558	0.33	0.517	0.288	0.857	0.829	-	-
EMMN	OxfordFlowers	trumpet creeper	0.588	0.059	0.709	0.413	-	-	0.558	0.312	0.517	0.31	0.857	0.828	-	-
ULip	StanfordDogs	Pekinese	0.787	0.836	0.59	0.569	-	-	0.655	0.651	-	-	0.933	0.933	0.708	0.709
ULip	StanfordDogs	toy poodle	0.607	0.098	0.591	0.508	-	-	0.655	0.646	-	-	0.933	0.927	0.708	0.694
ULip	StanfordDogs	Scotch terrier	0.625	0.828	0.591	0.531	-	-	0.655	0.649	-	-	0.933	0.931	0.708	0.704
ULip	StanfordCars	2009 Spyker C8 Coupe	0.429	0.048	0.656	0.494	-	-	-	-	0.591	0.591	0.933	0.932	0.708	0.716
ULip	StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548	0.0	0.656	0.515	-	-	-	-	0.591	0.598	0.933	0.931	0.708	0.713
ULip	StanfordCars	2011 Ford Ranger SuperCab	0.81	0.048	0.654	0.247	-	-	-	-	0.591	0.581	0.933	0.927	0.708	0.704
ULip	Caltech101	euphonium	1.0	1.0	0.933	0.919	-	-	0.655	0.645	0.591	0.59	-	-	0.708	0.705
ULip	Caltech101	minaret	0.913	0.435	0.933	0.923	-	-	0.655	0.655	0.591	0.59	-	-	0.708	0.707
ULip	Caltech101	platypus	1.0	1.0	0.933	0.907	-	-	0.655	0.65	0.591	0.571	-	-	0.708	0.694
ULip	OxfordFlowers	gazania	1.0	0.0												

Table 15. Aggregated forgetting results with ViT-B/16 visual encoder. We compare our method (Lip) to five other methods averaging across three classes for four selected datasets. We aim to minimize the Avg. *Target Class acc.* AF while maintaining Avg. *Other Classes acc.* AF and other datasets at a similar level to that before forgetting (BF).

Method	Dataset	Forgetting Type	Avg. Target	Avg. Other		Avg.		Avg.		Avg.		Avg.		
			Class acc.	Classes acc.	StanfordCars	StanfordCars	StanfordDogs	StanfordDogs	Caltech101	OxfordFlowers	BF	AF	BF	AF
Lip	StanfordCars	ZS	0.595 0.159	0.656 0.642	- -	0.591 0.584	0.933 0.932	0.708 0.707						
Emb	StanfordCars	ZS	0.595 0.0	0.656 0.557	- -	0.591 0.508	0.933 0.921	0.708 0.69						
Amns	StanfordCars	ZS	0.595 0.143	0.656 0.18	- -	0.591 0.398	0.933 0.876	0.708 0.51						
EMMN	StanfordCars	ZS	0.595 0.159	0.656 0.182	- -	0.591 0.119	0.933 0.589	0.708 0.137						
ULip	StanfordCars	semi ZS	0.595 0.032	0.656 0.419	- -	0.591 0.59	0.933 0.93	0.708 0.711						
AmnsRetain	StanfordCars	not ZS	0.595 0.0	0.656 0.744	- -	0.591 0.476	0.933 0.887	0.708 0.554						
Lip	StanfordDogs	ZS	0.673 0.142	0.591 0.592	0.655 0.647	- -	0.933 0.935	0.708 0.709						
Emb	StanfordDogs	ZS	0.673 0.071	0.591 0.518	0.655 0.632	- -	0.933 0.93	0.708 0.699						
Amns	StanfordDogs	ZS	0.673 0.219	0.591 0.358	0.655 0.59	- -	0.933 0.901	0.708 0.572						
EMMN	StanfordDogs	ZS	0.673 0.042	0.591 0.365	0.655 0.284	- -	0.933 0.826	0.708 0.438						
ULip	StanfordDogs	semi ZS	0.673 0.588	0.591 0.536	0.655 0.649	- -	0.933 0.93	0.708 0.702						
AmnsRetain	StanfordDogs	not ZS	0.673 0.021	0.591 0.698	0.655 0.472	- -	0.933 0.841	0.708 0.495						
Lip	Caltech101	ZS	0.971 0.576	0.933 0.935	0.655 0.652	0.591 0.594	- -	0.708 0.709						
Emb	Caltech101	ZS	0.971 0.598	0.933 0.91	0.655 0.609	0.591 0.517	- -	0.708 0.656						
Amns	Caltech101	ZS	0.971 0.846	0.933 0.848	0.655 0.517	0.591 0.445	- -	0.708 0.533						
EMMN	Caltech101	ZS	0.971 0.284	0.933 0.813	0.655 0.352	0.591 0.302	- -	0.708 0.473						
ULip	Caltech101	semi ZS	0.971 0.812	0.933 0.916	0.655 0.65	0.591 0.584	- -	0.708 0.702						
AmnsRetain	Caltech101	not ZS	0.971 0.0	0.933 0.946	0.655 0.506	0.591 0.474	- -	0.708 0.544						
Lip	OxfordFlowers	ZS	0.784 0.078	0.707 0.702	0.655 0.645	0.591 0.588	0.933 0.933	- -						
Emb	OxfordFlowers	ZS	0.784 0.0	0.707 0.617	0.655 0.543	0.591 0.522	0.933 0.906	- -						
Amns	OxfordFlowers	ZS	0.784 0.834	0.707 0.527	0.655 0.602	0.591 0.526	0.933 0.913	- -						
EMMN	OxfordFlowers	ZS	0.784 0.02	0.707 0.433	0.655 0.317	0.591 0.304	0.933 0.83	- -						
ULip	OxfordFlowers	semi ZS	0.784 0.02	0.707 0.529	0.655 0.64	0.591 0.554	0.933 0.913	- -						
AmnsRetain	OxfordFlowers	not ZS	0.784 0.0	0.707 0.945	0.655 0.56	0.591 0.531	0.933 0.914	- -						

Table 16. Forgetting with real data using Lipschitz loss ViT-B/16 visual encoder.

Dataset	Class name	Target	Other	Target	StanfordCars		StanfordDogs		Caltech101		OxfordFlowers	
		Class acc.	Classes acc.	Class acc.	BF	AF	BF	AF	BF	AF	BF	AF
StanfordDogs	Pekinese	0.787 0.016	0.59 0.607	0.0	0.655	0.657	-	-	0.933	0.932	0.708	0.709
StanfordDogs	toy poodle	0.607 0.131	0.591 0.581	0.022	0.655	0.64	-	-	0.933	0.94	0.708	0.715
StanfordDogs	Scotch terrier	0.625 0.047	0.591 0.56	0.085	0.655	0.64	-	-	0.933	0.937	0.708	0.709
StanfordCars	2009 Spyker C8 Coupe	0.429 0.238	0.656 0.652	0.176	-	-	0.591	0.591	0.933	0.933	0.708	0.708
StanfordCars	2010 Dodge Ram Pickup 3500 Crew Cab	0.548 0.143	0.656 0.637	0.088	-	-	0.591	0.598	0.933	0.938	0.708	0.713
StanfordCars	2011 Ford Ranger SuperCab	0.81 0.81	0.654 0.654	0.794	-	-	0.591	0.593	0.933	0.933	0.708	0.71
Caltech101	euphonium	1.0 0.0	0.933 0.935	0.031	0.655	0.656	0.591	0.596	-	-	0.708	0.707
Caltech101	minaret	0.913 0.739	0.933 0.928	0.711	0.655	0.644	0.591	0.593	-	-	0.708	0.719
Caltech101	platypus	1.0 0.9	0.933 0.934	0.647	0.655	0.656	0.591	0.599	-	-	0.708	0.714
OxfordFlowers	gazania	1.0 0.087	0.705 0.711	0.077	0.655	0.653	0.591	0.601	0.933	0.934	-	-
OxfordFlowers	tree mallow	0.765 0.294	0.707 0.7	0.172	0.655	0.652	0.591	0.601	0.933	0.941	-	-
OxfordFlowers	trumpet creeper	0.588 0.118	0.709 0.713	0.241	0.655	0.648	0.591	0.602	0.933	0.939	-	-

Table 17. Forgetting on multiple classes using Lipschitz loss ViT-B/16 visual encoder.

Dataset	Classes	Avg. Target		Other		StanfordCars	StanfordDogs	Caltech101	OxfordFlowers				
		Classes acc.	Classes acc.	BF	AF								
StanfordDogs	Pekinese,toy poodle,Scotch terrier 2009 Spyker C8 Coupe, 2010 Dodge Ram Pickup 3500 Crew Cab,	0.672	0.251	0.589	0.584	0.655	0.644	-	0.933	0.939	0.708	0.713	
StanfordCars	2011 Ford Ranger SuperCab	0.595	0.3	0.656	0.625	-	-	0.591	0.576	0.933	0.928	0.708	0.699
Caltech101	euphonium,minaret,platypus	0.971	0.498	0.932	0.929	0.655	0.634	0.591	0.589	-	-	0.708	0.709
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	-	-

G. Forgetting Algorithm

Algorithm 1 CLIP forgetting

Require: CLIP image and text encoders: f_θ, f_ϕ
Require: Textual class to forget: x_{text} , Learning rate: α ,
Require: Initial number of visual and textual layers to update: $InitV_{up}, InitT_{up}$
Require: Increase steps sigma: s , visual layers: v , textual layers: t
Require: Number of perturbations: N , Initial Gaussian perturbation: σ
Require: Accuracy stopping threshold: $GoalAcc$, Total increase steps: I
Require: Optimizer: $optim(\theta, \phi, lr = \alpha)$

▷ Function that generates synthetic training samples by gradient ascent as in Eq. 5
 $X \leftarrow SyntheticImagesGen(x_{text})$

for n in $range(I)$ **do**

- for** x_{img} in X **do**

 - $\ell = 0$
 - for** i in $range(N)$ **do**

 - Sample $\epsilon \sim \mathcal{N}(0, \sigma^2)$
 - $x'_{img} = x_{img} + \epsilon$
 - $k = \frac{\|f_\theta(x_{img}) - f_\theta(x'_{img})\|_2 + \|f_\phi(x_{text}) - f_\theta(x'_{img})\|_2}{\|\epsilon\|_2}$
 - $\ell = \ell + k$

 - end for**
 - $\ell = \ell/N$

▷ Update $InitV_{up}$ and $InitT_{up}$ most important layers in visual and textual branch respectively
 $\theta, \phi \leftarrow SelectiveUpdate(optim\{\Delta_{\theta, \phi}\ell\}, InitV_{up}, InitT_{up})$

$Acc \leftarrow EvalAcc(X)$ ▷ Accuracy on synthetic training samples

if $Acc < GoalAcc$ **then**

 - return** θ, ϕ

else

 - ▷ Increase parameters to reduce more aggressively the accuracy on synthetic samples
 - $InitV_{up} \leftarrow InitV_{up} + v$
 - $InitT_{up} \leftarrow InitT_{up} + t$
 - $\sigma \leftarrow \sigma + s$

end if

end for

end for

return θ, ϕ

H. Additional Tasks Full results and Further Investigation

H.1. Further Investigation Image-Image Retrieval

In the main paper, among the additional tasks we tested the image-image retrieval on the model after forgetting. We surprisingly found that even after forgetting the model is still able to retrieve images of the class it has forgotten starting from an input image. We speculated that in image retrieval, the model can still identify similar features and shapes of objects without actually recognizing or knowing the specific class they belong to. To confirm our hypothesis we conduct image-image retrieval on the **original** CLIP model on classes it predicts with **zero classification accuracy**. The results that confirm our hypothesis are shown in Tab. 18.

Table 18. Image retrieval from image input results on classes with zero classification accuracy.

Model Type	Class	Classification Accuracy	Precision@1	Precision@5	Precision@10
CLIP original	Appenzeller (StanfordDogs)	0	1.0	0.4	0.2
CLIP original	Pembroke (StanfordDogs)	0	1.0	0.6	0.4
CLIP original	Cardigan (StanfordDogs)	0	0.0	0.2	0.2
CLIP original	2010 Chevrolet HHR SS (StanfordCars)	0	1.0	0.4	0.4
CLIP original	2009 HUMMER H2 SUT Crew Cab (StanfordCars)	0	1.0	0.6	0.7
CLIP original	english marigold (OxfordFlowers)	0	1.0	0.8	0.6
CLIP original	colt's foot (OxfordFlowers)	0	1.0	0.8	0.7
CLIP original	cape flower (OxfordFlowers)	0	1.0	1.0	1.0

H.2. Full results

Table 19. 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	Scotch terrier	0.0	0.4	0.4
CLIP forget	toy poodle	0.0	0.0	0.1
CLIP forget	Pekinese	0.0	0.0	0.2
CLIP forget	2009 Spyker C8 Coupe	1.0	0.8	0.8
CLIP forget	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.0	0.1
CLIP forget	2011 Ford Ranger SuperCab	1.0	0.6	0.4
CLIP forget	euphonium	1.0	1.0	0.6
CLIP forget	minaret	1.0	1.0	0.9
CLIP forget	platypus	1.0	1.0	0.5
CLIP forget	gazania	1.0	0.2	0.4
CLIP forget	tree mallow	0.0	0.0	0.2
CLIP forget	trumpet creeper	0.0	0.2	0.2
CLIP forget Mean	-	0.5	0.433	0.4

Table 20. Image retrieval from image 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	0.2	0.3
CLIP original	toy poodle	1.0	0.4	0.5
CLIP original	Pekinese	0.0	0.0	0.0
CLIP original	2009 Spyker C8 Coupe	1.0	0.4	0.3
CLIP original	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.0	0.1
CLIP original	2011 Ford Ranger SuperCab	0.0	0.2	0.4
CLIP original	euphonium	1.0	0.8	0.9
CLIP original	minaret	1.0	1.0	1.0
CLIP original	platypus	1.0	0.6	0.5
CLIP original	gazania	1.0	1.0	0.7
CLIP original	tree mallow	1.0	1.0	0.6
CLIP original	trumpet creeper	1.0	1.0	0.7
CLIP original Mean	-	0.667	0.55	0.5
CLIP forget	Scotch terrier	0.0	0.2	0.3
CLIP forget	toy poodle	1.0	0.4	0.5
CLIP forget	Pekinese	0.0	0.0	0.0
CLIP forget	2009 Spyker C8 Coupe	1.0	0.4	0.2
CLIP forget	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.0	0.1
CLIP forget	2011 Ford Ranger SuperCab	0.0	0.4	0.3
CLIP forget	euphonium	1.0	0.8	0.9
CLIP forget	minaret	1.0	1.0	0.9
CLIP forget	platypus	1.0	0.6	0.5
CLIP forget	gazania	1.0	1.0	0.7
CLIP forget	tree mallow	1.0	1.0	0.7
CLIP forget	trumpet creeper	1.0	1.0	0.7
CLIP forget Mean	-	0.667	0.567	0.483

Table 21. 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	Scotch terrier	0.0	0.0	0.0
CLIP forget	toy poodle	1.0	0.2	0.1
CLIP forget	Pekinese	0.0	0.0	0.0
CLIP forget	2009 Spyker C8 Coupe	0.0	0.8	0.5
CLIP forget	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.2	0.3
CLIP forget	2011 Ford Ranger SuperCab	0.0	0.0	0.0
CLIP forget	euphonium	0.0	0.8	0.8
CLIP forget	minaret	0.0	0.4	0.2
CLIP forget	platypus	0.0	0.2	0.2
CLIP forget	gazania	0.0	0.0	0.0
CLIP forget	tree mallow	0.0	0.2	0.2
CLIP forget	trumpet creeper	0.0	0.0	0.0
CLIP forget Mean	-	0.08	0.23	0.191

Table 22. Image retrieval from image 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	0.0	0.0	0.0
CLIP original	toy poodle	0.0	0.0	0.1
CLIP original	Pekinese	0.0	0.0	0.0
CLIP original	2009 Spyker C8 Coupe	1.0	0.4	0.3
CLIP original	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.0	0.0
CLIP original	2011 Ford Ranger SuperCab	0.0	0.2	0.3
CLIP original	euphonium	0.0	0.4	0.2
CLIP original	minaret	1.0	0.8	0.6
CLIP original	platypus	0.0	0.2	0.2
CLIP original	gazania	1.0	0.8	0.6
CLIP original	tree mallow	1.0	0.8	0.8
CLIP original	trumpet creeper	1.0	0.8	0.7
CLIP original Mean	-	0.417	0.367	0.317
CLIP forget	Scotch terrier	0.0	0.0	0.0
CLIP forget	toy poodle	0.0	0.0	0.1
CLIP forget	Pekinese	0.0	0.0	0.0
CLIP forget	2009 Spyker C8 Coupe	1.0	0.4	0.3
CLIP forget	2010 Dodge Ram Pickup 3500 Crew Cab	0.0	0.0	0.1
CLIP forget	2011 Ford Ranger SuperCab	0.0	0.2	0.3
CLIP forget	euphonium	0.0	0.4	0.2
CLIP forget	minaret	0.0	0.8	0.6
CLIP forget	platypus	0.0	0.2	0.2
CLIP forget	gazania	1.0	0.8	0.6
CLIP forget	tree mallow	1.0	0.8	0.8
CLIP forget	trumpet creeper	1.0	0.8	0.7
CLIP forget Mean	-	0.333	0.367	0.325

I. Additional Figures and Implementation Details

Implementation Details We use Adam optimizer with learning rate of 5e-05, weight decay of 0.2 and betas parameters of 0.9 and 0.98. We set the number of perturbed samples N to 25. Initial σ is 0.1 increasing to 2. Initially, layers we allow to vary are 5 for both visual and textual encoders and increasing to 8 and 20 respectively if we need more forgetting according to the remaining synthetic samples accuracy.

Additional Figures In the vision encoder, the weights of the values, queries, and output projections are updated most frequently, whereas in the text encoder the MLP output projection weights are updated most often.

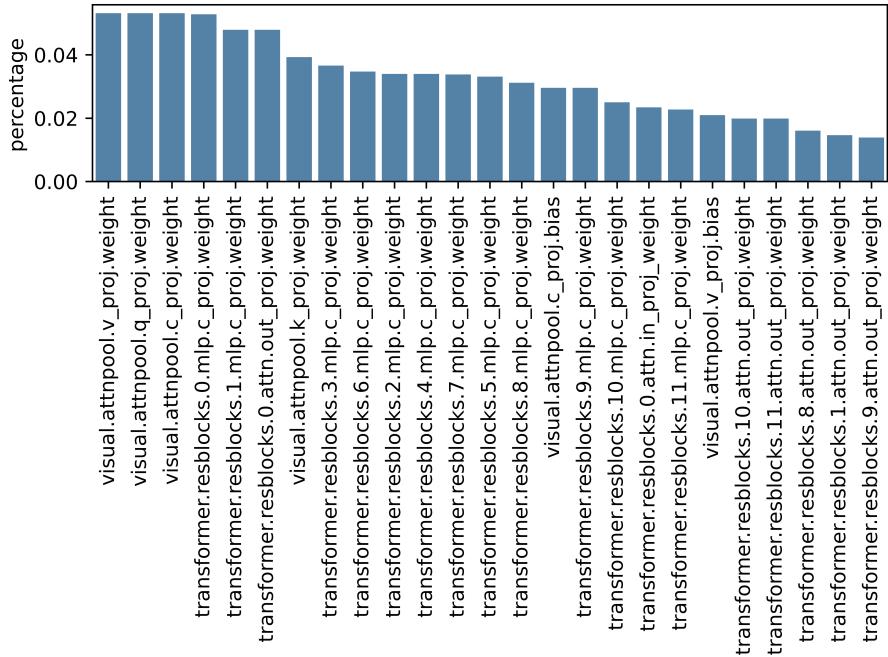


Figure 4. **Selected layers for forgetting.** The figure shows the top 25 most frequent updated layers during forgetting process across selected classes and datasets.

J. Limitations

One limitation is that it is hard to assess how well the model will perform on other classes after unlearning. Looking at *2012 Chevrolet Avalanche Crew Cab* class in Tab. 6, even if forgetting is quite successful, 6% of accuracy is lost on other classes of *StanfordCars*. Note that knowledge about classes not related to cars remained fairly close to that before forgetting. Our iterative procedure can help control this trade-off between unlearning and retaining knowledge.