Déjà vu Memorization in Vision–Language Models

Bargav Jayaraman
FAIR, Meta
California, USA
bargav@meta.com

Chuan Guo
FAIR, Meta
California, USA
chuanguo@meta.com

Kamalika Chaudhuri
FAIR, Meta
California, USA
kamalika@meta.com

Abstract

Vision-Language Models (VLMs) have emerged as the state-of-the-art representation learning solution, with myriads of downstream applications such as image classification, retrieval and generation. A natural question is whether these models memorize their training data, which also has implications for generalization. We propose a new method for measuring memorization in VLMs, which we call *déjà vu memorization*. For VLMs trained on image-caption pairs, we show that the model indeed retains information about individual objects in the training images beyond what can be inferred from correlations or the image caption. We evaluate *déjà vu* memorization at both sample and population level, and show that it is significant for OpenCLIP trained on as many as 50M image-caption pairs. Finally, we show that text randomization considerably mitigates memorization while only moderately impacting the model's downstream task performance. The code is available here: https://github.com/facebookresearch/VLMDejaVu.

1 Introduction

Vision-Language Models (VLMs) have emerged as the state-of-the-art solution for learning representations from images and text data, with a number of downstream applications such as image generation [Ramesh et al., 2021, 2022, Yu et al., 2022b], retrieval [Wang et al., 2015, Cao et al., 2016, Zhang et al., 2021, Baldrati et al., 2022], captioning [Mokady et al., 2021], and classification. At the same time, large foundation models are known to memorize and retain information about their training data [Carlini et al., 2019, Meehan et al., 2023, Carlini et al., 2023], and hence, a natural question is whether these Vision-Language Models *memorize* as well. If so, this raises questions about generalizability of these models. We investigate whether Vision-Language Models retain information about their training data beyond the bounds of generalization.

The main challenge in measuring memorization is designing a measurement technique that can tease apart memorization from spurious correlations. For example, for an image of a black swan on water, a representation learning model may learn to predict *black swan* given the background *water* if either: *(i)* it retains extra information about the training image, or, *(ii)* if most of the examples in the training corpus with water also involve black swans. The first kind constitutes as memorization whereas the second kind is spurious correlation. This uncoupling of memorization from spurious correlation is particularly complicated for VLMs. Unlike generative models, VLMs as well as other representation learning models lack decoders that can directly generate images or text; therefore, what the model learns about its training data has to be detected more subtly.

Prior work has looked into this problem for image-only representation models [Meehan et al., 2023] by measuring whether the model can predict the foreground of an image (e.g, black swan) beyond simple correlations based simply on its background (e.g, water). However, such simple solutions do not apply here. VLMs have two separate modalities – text and image, and the data sets used to train and evaluate them are considerably more complex than the simple foreground-background structure

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

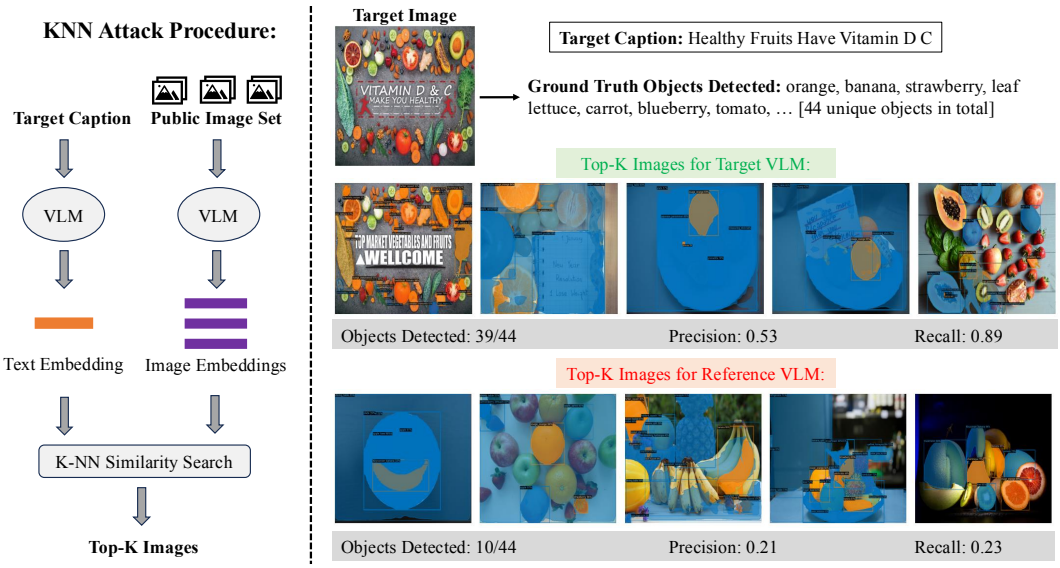


Figure 1: An example where a CLIP [Radford et al., 2021] model trained on a 40M subset of a Shutterstock data set exhibits *déjà vu* memorization of objects present in a training image. Public set is a separate collection of 20M images from Shutterstock that has no overlap with the training set. The objects annotated in orange are true positives, i.e., the ones present in the target image, and the objects annotated in blue are false positives. Our test recovers significantly more memorized objects for the target VLM (trained on the target image) compared to the reference VLM (not trained on the target image). Additional qualitative examples can be found in Figure 11 in the appendix.

of ImageNet (see Figure 6 for an example). A consequence is that the image and text modalities can interact and transfer information in these models in subtly complex ways, making measurement significantly more challenging.

In this work, we propose a new method for measuring memorization in VLMs (depicted in Figure 1). Given a target image caption, we use the VLM to encode the caption and retrieve relevant image samples from a *public set* of images. Our test is based on the key insight that if an image-text pair is memorized by a VLM, then the retrieved images would resemble the training image to a significantly higher amount of detail than what is predictable from either the text caption or simple correlation. Formally, given a text-image pair, we retrieve an image from the model based on an embedding of its text description, and we measure what fraction of ground-truth objects in the original image also co-occur in the retrieved image. Then, to determine whether this happens simply due to correlation, we measure how this compares with the same statistic obtained from a similar VLM which does not have this image-text pair in its training data. Combining these two steps gives us a measurement method that we call VL-Déjà-Vu.

We evaluate our test on CLIP [Radford et al., 2021] models trained on subsets of Shutterstock and a filtered version of LAION (filtered LAION) with varying number of training samples. We find that even at training data set sizes where CLIP generalizes well, there is a significant degree of model memorization as depicted by our metrics (see Section 4). Finally, we explore mitigation measures that reduce information leakage in Section 5. We find that text masking significantly mitigates déjà vu memorization at a marginal cost to the model utility. We note that there could be other effective mitigations but were not explored due to the computational limitations.

Contributions. Our main contributions are as follows.

- We propose VL-Déjà-Vu—a new way of measuring memorization in VLMs by measuring what fraction of ground-truth objects in an image can be predicted from its text description for a training image-text pair.
- Based on this measurement technique, we propose both (a) an individual sample-level test to detect memorization for individual text-image pairs and (b) an aggregate population-level test for a Vision-Language Model.

- We use our VL-Déjà-Vu test to evaluate memorization in CLIP, and show that memorization does occur for VLMs trained using a number of different training set sizes and regularization parameter values, even for settings where the model generalizes well.
- Finally, we explore mitigation measures, and demonstrate that among a number of different ways to train CLIP, random masking of text serves to significantly reduce *déjà vu* memorization.

2 Background

Vision-Language models [Radford et al., 2021, Li et al., 2022, Yu et al., 2022a, Li et al., 2023, Xu et al., 2023] are multi-modal models whose core function is to map image-text pairs into a pair of representations that are semantically relevant. These embeddings can then be used for downstream tasks such as image classification, captioning, retrieval and generation. VLMs are composed of a *vision block*, consisting of a convolutional network or a vision transformer, and a *text block*, consisting of a transformer, that produce image and text embeddings respectively from input image-text pairs. Given a trained vision-language model f, and an image-text pair $z = \langle z_{img}, z_{txt} \rangle$, we denote the corresponding image and text embeddings as $f(z_{img})$ and $f(z_{txt})$.

We consider VLMs that involve contrastive pre-training; in other words, during training, the model learns to minimize the distance between the image and text embeddings of the matching pairs in the training set D_{tr} while maximizing the distance of the mismatched pairs. The most commonly used contrastive loss is the InfoNCE loss [van den Oord et al., 2018] given as follows:

$$L = -\log \frac{\exp(f(z^i_{img})^\intercal f(z^i_{txt})/\tau)}{\sum_j \exp(f(z^i_{img})^\intercal f(z^j_{txt})/\tau)} \tag{1}$$

where τ is the temperature and $z^j, \forall j \neq i$ are negative examples to contrast against. In practice, for each positive example z^i, we use all other examples in a training batch as negative examples. The most popular VLM of this type is CLIP (Contrastive Language-Image Pre-Training; Radford et al. [2021]), trained on an undisclosed data set, which achieves competitive out-of-the-box performance across many transfer learning tasks. OpenCLIP [Ilharco et al., 2021] has released an open-source implementation of CLIP, and showed that training on a filtered LAION dataset [Schuhmann et al., 2021] can achieve comparable performance to the original CLIP model. Our work investigates memorization in OpenCLIP.

Memorization in ML models. It is well-known that machine learning models can memorize their training data in ways that enable data extraction. This phenomenon has been studied for both language [Carlini et al., 2019, 2021, Zanella-Béguelin et al., 2020, Jayaraman et al., 2022] and vision [Somepalli et al., 2023, Carlini et al., 2023, Sablayrolles et al., 2018, Meehan et al., 2023] models. However, all these works only consider the uni-modal setting, and as such the impact of this phenomenon is not clear in the multi-modal settings. Moreover, almost all the prior studies (except Meehan et al. [2023]) focus on generative models – language or vision – where measuring memorization is easier because of the presence of a decoder.

Similar to Meehan et al. [2023], we investigate the setting of representation learning models, where we do not have a decoder and instead only have access to an encoder. Although unlike Meehan et al. [2023], who considered vision models that capture the relationship between representation of the background of an image (such as water) and the label of its foreground object (such as black swan), we consider settings where the models are trained on more complex data sets that have multiple objects in any given image. Such a simple foreground-background measurement does not directly apply to our setting of Vision Language Models where the two modalities may leak training data in more subtle and complicated ways. Our work builds upon their test, and extends it to VLMs. A more detailed background discussion can be found in Appendix B.

3 *Déjà vu* Memorization for Vision-Language Models

Déjà vu memorization happens when a foundation model retains information about individual training data points beyond what is expected by simple correlation, and allows the recovery of such information during inference time. An example is when an image representation learning model

can confidently predict the foreground of a training image based simply on its background [Meehan et al., 2023], while similar predictions cannot be made for test images.

In the context of Vision-Language Models, however, measuring *déjà vu* memorization is not as simple, due to the presence of multiple modalities as well as the complex nature of the training data. Compared to ImageNet, VLMs are trained on vastly more semantically rich data sets with many more objects as well as complicated captions, which may not capture everything in the image – see Figure 6 for an example. This means that the text and image modalities can interact and transfer information in subtly complex ways, making measurement significantly more challenging.

To resolve this challenge, we instead propose to measure whether the ground truth objects in an image can be predicted from the representation of its caption. We rely on the intuition that the caption of an image typically does not include all its objects, and hence high confidence recovery of this level of detail implies some form of memorization. If this prediction can be done significantly more accurately when the image is in the training set of a model than when it is in the test, then the image-text pair is being memorized by the said model.

Definition 1 (*Déjà vu* Memorization) *A vision-language model f suffers from déjà vu memorization if it retains specific information about the individual training images that allows the recovery of objects present in the training images. In other words, for a target image-text pair $z = \langle z_{img}, z_{txt} \rangle$, more unique objects can be recovered from z_{img} given z_{txt} when z is present in f's training set compared to when it is not.*

This is possible due to the model's ability to encode the individual objects in the image embeddings, which is in turn reflected in the corresponding text embeddings when the model minimizes the contrastive loss during training. Next we will discuss how we quantify this phenomenon using two separate models (a target and a reference) as well as a nearest neighbor test.

3.1 Measurement Methodology

Since VLMs are meant to capture general correlations between images and their text captions, our goal is to differentiate the recovery of ground-truth objects due to *déjà vu* memorization from dataset-level correlations alone. As a motivating example, consider the use of CLIP in a cross-modal retrieval task, where images are retrieved from a web-scale database given text. We wish to capture the degree of surprise in the retrieval result when the model memorizes training captions, i.e. how many objects can the model recover beyond dataset-level correlation? To enable this evaluation for a given image-text pair $z = \langle z_{img}, z_{txt} \rangle$, we use two separate VLMs f_A and f_B that are trained on randomly sampled but disjoint data sets A and B respectively. z lies in the training set of *exactly one* of these models, and hence by comparing the outputs of the two models, we can infer whether z was memorized. We do a k-nearest neighbor test using a separate public set of images as described in Algorithm 1 and find the subset of images that are closest to z in the representation space. We then decode the objects present in these images. For this we use an object detector to provide ground-truth annotations for measuring the precision and recall of object recovery. We note that while there will always be some bias when using object detectors, human or automated, this bias should not affect our evaluation when considering the gap between the two models. This is because the object detector is not trained on the same training set as the VLM, hence any incurred bias should be independent of the trained VLMs.

3.2 Metrics

Our memorization metrics are built bottom-up from our notion of deja vu memorization for VLMs. We start from fine-grained *sample-level metrics* to more aggregate *population-level metrics*. The k-nearest neighbor test in Algorithm 1 shows how to obtain predictions of the ground-truth objects given an image; we next use these predictions to develop the population-level and sample-level memorization metrics. For our evaluation, we adopt the precision, recall and F-score metrics from the information retrieval literature to quantify the fraction of objects memorized by the models.

Sample-level metrics. At the sample level, we evaluate the fraction of ground-truth objects memorized by the target model from a given training image–text pair $z = \langle z_{img}, z_{txt} \rangle$. To do this, we run the nearest neighbor test on both the target and reference models, f_A and f_B, to obtain their

Algorithm 1 k-Nearest Neighbor Test

Setup Phase

1: Sample two disjoint data sets A and B consisting of image–text pairs of the form $z = \langle z_{img}, z_{txt} \rangle$ and train models f_A and f_B on the respective data sets.
2: Sample a separate public set of images P that is disjoint from the images in A and B.
3: For each image $z^i_{img} \in P$, obtain the corresponding image embeddings from both the models, $f_A(z^i_{img})$ and $f_B(z^i_{img})$.

Testing Phase

4: Sample a record from A set, $z = \langle z_{img}, z_{txt} \rangle \in A$, and obtain the corresponding text embeddings from both the models, $f_A(z_{txt})$ and $f_B(z_{txt})$.
5: Obtain k public images $N_A \subseteq P$ and $N_B \subseteq P$ closest to z_{txt} in the embedding space for f_A and f_B respectively.
6: Evaluate the gap between the fraction of ground-truth objects detected in the sets N_A and N_B.

respective neighbor sets N_A and N_B as per Algorithm 1. We then calculate the *precision*, *recall* and *F-score* values when identifying the ground truth objects present in z_{img} using N_A and N_B and report the gap between the respective values for both the models. A positive gap corresponds to the target model memorizing the training sample and the magnitude of the gap indicates the degree of memorization. The precision, prec, and recall, recall, are given by the following equations ($\forall i \in \{A, B\}$):

$$\mathsf{prec}(z, f_i) = \frac{\#\text{ unique objects in } N_i \cap z_{img}}{\#\text{ unique objects in } N_i}, \quad \mathsf{recall}(z, f_i) = \frac{\#\text{ unique objects in } N_i \cap z_{img}}{\#\text{ unique objects in } z_{img}}. \tag{2}$$

F-score is the harmonic mean of precision and recall.

Population-level metrics measure what fraction of the training data is memorized by a model. For proper measurement, we propose three metrics: *population precision gap* (PPG), *population recall gap* (PRG) and *AUC gap* (AUCG). Given the notations defined in Algorithm 1, the population precision gap is the the fraction of data points from A where f_A has a higher precision in identifying the ground truth objects than f_B minus the fraction of data points where f_B has a higher precision in identifying the ground truth objects than f_A. If no memorization occurs, models f_A and f_B should be interchangeable and hence this gap is zero. Formally,

$$\mathsf{PPG} = \frac{1}{|A|}\Big(|\{z \in A : \mathsf{prec}(z, f_A) > \mathsf{prec}(z, f_B)\}| - |\{z \in A : \mathsf{prec}(z, f_A) < \mathsf{prec}(z, f_B)\}|\Big), \tag{3}$$

where $|A|$ denotes the size of the set A and $\mathsf{prec}(z, f_A)$ measures the precision of object prediction on z given the model f_A as defined in Equation 2. We define the population recall gap similarly:

$$\mathsf{PRG} = \frac{1}{|A|}\Big(|\{z \in A : \mathsf{recall}(z, f_A) > \mathsf{recall}(z, f_B)\}| - |\{z \in A : \mathsf{recall}(z, f_A) < \mathsf{recall}(z, f_B)\}|\Big). \tag{4}$$

We also visualize the fine-grained cumulative recall distribution of both the models over the training set as shown in Figure 3. This gives us a better understanding of what fraction of objects are recovered overall. We then measure the difference between the two distributions (i.e., for f_A and f_B) to simplify this information into a single quantity we call AUC gap.

While both the population-level and sample-level metrics rely on the precision and recall functions, they have subtle differences. First, population-level metrics measure the aggregate memorization over the entire training set whereas sample-level metrics measure the memorization in individual training samples. Second, population-level metrics rely on binary tests to differentiate between the target and reference models and as such do not capture the magnitude of the gap between the models as is done by the sample-level metrics. We define both sets of metrics to capture the memorization at different granular levels and to be actionable in a meaningful way, thereby allowing the model developers to fine-tune the models to mitigate the memorization risk.

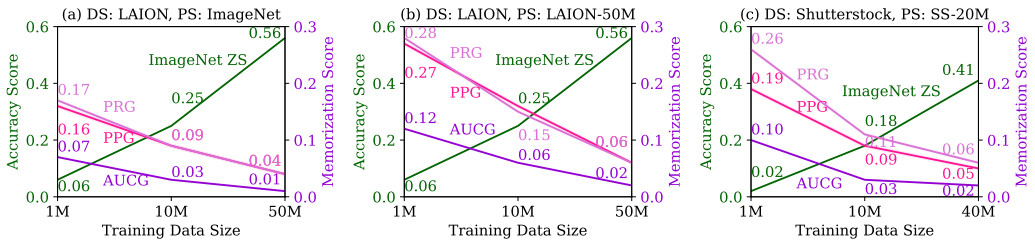


Figure 2: Utility and *déjà vu* memorization of ViT-B-32 CLIP models with varying training set sizes. Model utility is quantified in terms of ImageNet zero-shot accuracy. Population-level memorization of models is measured using the metrics defined in Section 3.2 over various public sets *(a)*: training set sampled from filtered LAION and ImageNet is used as public set. *(b)*: training set sampled from filtered LAION and a holdout filtered LAION-50M set is used as public set. *(c)*: training set sampled from Shutterstock and a holdout SS-20M set is used as public set. For the memorization metrics, we report the *mean* $\pm$ *std* values (*std* ≤ 0.003) over 100 repetitions of randomly sampling 10% of records with replacement.

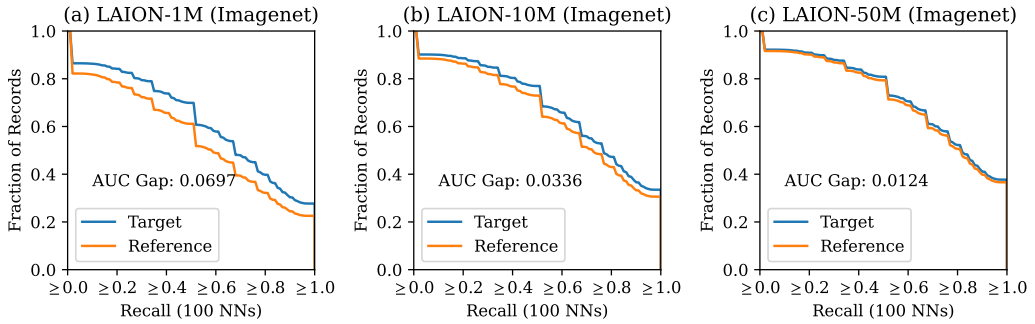


Figure 3: Object recall distribution of target and reference models trained on filtered LAION data set for 200 epochs with different training sizes. ImageNet is used as the public set for kNN test.

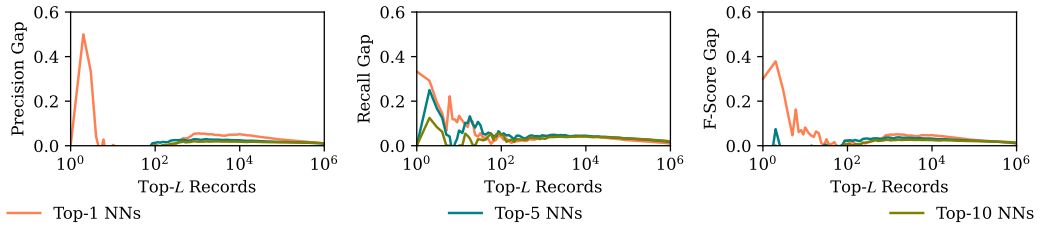
4 Evaluating *Déjà vu* Memorization

We next apply the metrics designed in Section 3.1 to determine if CLIP memorizes training data. Specifically, we seek to answer the following two research questions:

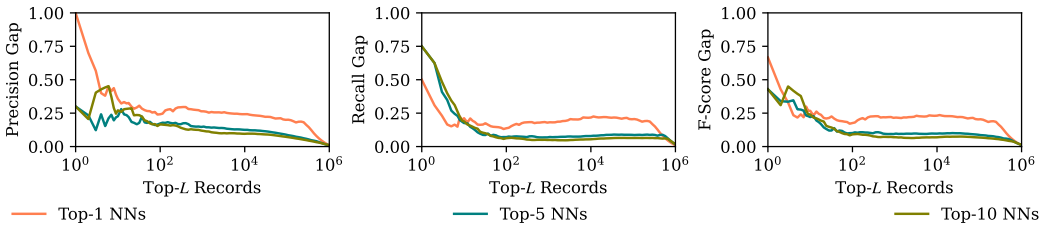
1. How does déjà vu memorization vary with training set size and number of training epochs?
2. Are all training data points memorized uniformly?

Models and datasets. We train OpenCLIP from scratch on different datasets, including Shutterstock (a privately licensed data set of 239M image-captions pairs) and ⟨filtered LAION [Radenovic et al., 2023] + COCO [Lin et al., 2014]⟩. We sample up to 50M image-text pairs from the data sets and train OpenCLIP models with ViT-B-32 architecture. For Shutterstock experiments, we consider a separate set of 20M samples from Shutterstock (called SS-20M), with no overlap with the training sets, as public set. For the filtered LAION experiments, we consider two public sets: (a) a separate subset of 50M samples from filtered LAION (called filtered LAION-50M) with no overlap with the training sets, and (b) the entire ImageNet training set [Deng et al., 2009]. More details on the experiment setup and how we obtain data subsets can be found in Appendix C.

Model utility. As mentioned above (and also discussed in detail in Appendix C), we trained models with different training set sizes consisting of 1M/10M/50M image-text pairs from filtered LAION and 1M/10M/40M image-text pairs from Shutterstock. We use zero-shot performance on ImageNet to evaluate the utility of these models. Figure 2 shows the zero-shot accuracy on ImageNet. Additional utility benchmarks across various ARO (Attribution, Relation, and Order) tasks [Yuksekgonul et al., 2023] can be found in Figure 7 in the appendix.



(a) Records are sorted w.r.t. the minimum embedding distance between target caption and public images.



(b) Records are sorted w.r.t. the decreasing number of correct object predictions for target model.

Figure 4: Sample-level memorization gap between target and reference models when predicting top-10 objects for different top-L records. Models are trained on disjoint 10M subsets of filtered LAION data set for 200 epochs and ImageNet public set is used for the KNN test. The model exhibits very strong *déjà vu* memorization on a small subset of samples, as indicated by the large precision/recall/F-score gaps when L is small.

4.1 Measuring Population-Level Memorization

For quantifying population-level memorization, we measure the gap between the object recall distributions for the target and reference models. If there were no memorization, we would observe virtually no gap between the two distributions, i.e. AUCG = 0. Figure 3 shows the object recall distribution gap between the target and reference models trained on filtered LAION for varying training set sizes when ImageNet is used as the public set. When the training set size is small (e.g. 1M as shown in the left-most figure), there is a higher *déjà vu* memorization due to the models overfitting on the training set. The gap decreases as the training set size increase from 1M up to 50M, confirming that the models begin to generalize better. Note that the memorization is still significant for models trained on 10M data set. We consider this setting for further experiments as this is a typical training set size for many foundation models in practice [Ilharco et al., 2021]. For instance, it is common to train CLIP models on the 12M Conceptual Captions data set [Sharma et al., 2018] or the 15M subset of the YFCC data set [Thomee et al., 2016].

Apart from the AUCG (AUC gap) metric, we also quantify the gap in terms of the PPG (population precision gap) and PRG (population recall gap) metrics. Recall that a positive value for these metrics indicates memorization and the magnitude indicates the degree of memorization. Figure 2 shows the PPG, PRG and AUCG metric values for models trained on filtered LAION and Shutterstock with different training set sizes; using ImageNet and filtered LAION-50M public sets for the filtered LAION models and SS-20M public set for the Shutterstock models. Recall that the public sets have no overlap with the model training sets. While the absolute metric values are different for different public sets, the trend remains the same: memorization decreases with increasing training set size as the models begin to generalize better. In Section 5, we explore various approaches to reduce this memorization.

4.2 Measuring Sample-Level Memorization

While the population-level metrics like AUCG, PPG and PRG show evidence of memorization, they do not pinpoint which training images are more vulnerable. We sort the training data in decreasing order of memorization to show the subset of most vulnerable records. To do this, we explore several sorting metrics. The most straightforward metric is the distance between the training text embedding and the nearest neighbour public image embeddings obtained using Algorithm 1. The records for which the public image embeddings are the closest are more easily memorized by the model.

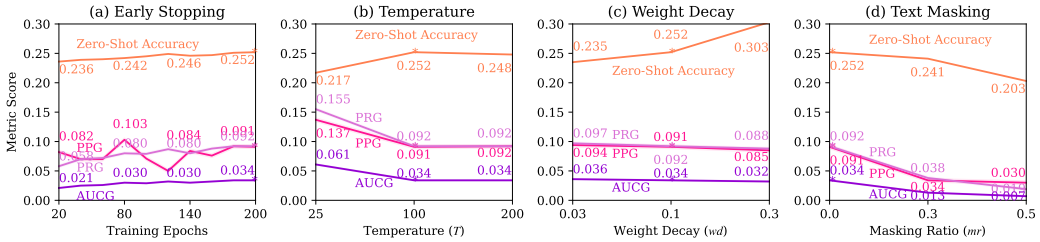


Figure 5: Effect of mitigation on ViT-B-32 OpenCLIP models trained on 10M subset of filtered LAION. Memorization evaluation is done using ImageNet as public set. Default setting is highlighted with asterisk. For the memorization metrics, we report the *mean* $\pm$ *std* values (*std* ≤ 0.003) over 100 repetitions of randomly sampling 10% of records with replacement. Among these mitigations, text masking has the best trade-off that reduces memorization without sacrificing utility.

Compared to the population-level memorization, where we keep the experiments parameter-free to the best extent, at the sample-level we want to focus on more fine-grained leakage so we choose top-10 object labels to measure the gap instead of predicting all the objects.

Figure 4a shows the precision, recall and F-score gaps between the target and reference models for varying top-k records sorted with respect to this distance metric where ImageNet is used as the public set. As shown, the gaps can be greater than 0.3 for top-1 and top-10 records. We also tried sorting the records in the decreasing order of the number of objects correctly identified using the target model with the nearest neighbor test. Figure 4b shows the precision, recall and F-score gaps for the records sorted using this metric. We see that the gap can become very significant for the top-1 and top-10 records. Although this metric requires access to the ground truth labels, this is still useful to visualize the worst case examples. Results for sample-level memorization with filtered LAION-50M public set show a similar trend and can be found in Section D.1. Sample-level memorization results for Shutterstock experiments can be found in Appendix E.

Key Observations. We show *déjà vu* memorization at both population and sample levels. At the population-level, where we measure the aggregate memorization of model over the training set, we find that the memorization decreases with an increase in the training set size. This could be attributed to improved model generalization. At the sample-level, we note that the model memorizes disproportionately—a subset of training image-text pairs are memorized more than the others.

5 Mitigation

How can we mitigate *déjà vu* memorization in VLMs? Since it presumably happens due to the model overfitting on training data, it is likely that regularization techniques may be able to mitigate it. We investigate the impact of four regularization techniques on *déjà vu* memorization.

1. *Early stopping* is a common technique for regularizing neural networks where model training is ended prematurely. It is effective due to the observation that models begin to overfit on the training set when they are trained for more epochs.
2. *Temperature* is the contrastive loss parameter that controls how close the text and image embeddings can get during the model training. Changing the temperature parameter has a regularization effect for SSL as observed by Meehan et al. [2023].
3. *Weight decay*, also known as L_2 regularization, is a standard ML regularization technique.
4. To reduce overfitting along the text and image modalities in VLMs, we look at additional regularization through *text randomization*, where we randomly mask a fraction of the text tokens during training. We control the fraction of text tokens masked using a *masking ratio* parameter.

In the following we present results when ImageNet is used as the public set for the nearest neighbor test. Results for the filtered LAION-50M public set can be found in Section D.2. Since Shutterstock memorization trends are similar to those of filtered LAION, we only explore filtered LAION settings for mitigation.

5.1 Early Stopping

It is widely believed that deep learning models begin to overfit on the training data as the number of training epochs increases. It is thus a good practice to early stop the training as soon as the model utility on a hold-out test set stagnates or begins to drop. However this is often not the case for SSL models. It is not uncommon to observe that the zero-shot accuracy of SSL models keeps improving as the models are trained for longer [Meehan et al., 2023]. Regardless, we still explore early stopping as a mitigation mechanism. As shown in Figure 5, training the CLIP model for more epochs leads to better zero-shot accuracy, but at the same time, *déjà vu* memorization also increases. This is in line with our hypothesis above. Even when we early stop the model at 20 epochs (10% of the default parameter value of 200 epochs), the memorization risk is not completely mitigated although the absolute values are lower.

5.2 Temperature Scaling

Temperature, or logit scale, controls how close the text and image embeddings can get during training. Smaller values allow for the multi-modal embeddings to get closer, and as a consequence the CLIP contrastive loss drops quickly, whereas larger values regularize the loss but may lead to training instability as noted by Radford et al. [2021]. The default value in OpenCLIP implementation is set to 100. We vary this value between 25, 100 and 200. As shown in Figure 5, decreasing the temperature (T) from 100 to 25 decreases the model's zero-shot classification accuracy on ImageNet from 25.2% to 21.7% and also increases the memorization as indicated by the increase in the PPG, PRG and AUCG metrics. This is due to the decrease in the distance between the text and image embeddings for the training data which could potentially lead to model overfitting. Increasing the temperature to 200 moderately impacts the model's zero-shot classification accuracy and the memorization leakage remains more or less the same.

5.3 Weight Decay

Weight decay directly controls the model overfitting, with larger values corresponding to stronger regularization. The default value is set to 0.1 and we vary it between 0.03, 0.1 and 0.3. As expected, decreasing the weight decay wd from 0.1 to 0.03 decreases the model's zero-shot classification accuracy and also worsens the leakage due to memorization as shown in Figure 5. Interestingly, increasing the weight decay to 0.3 significantly improves the model's zero-shot accuracy. We believe that the default value of 0.1 is not optimal for the 10M training set size as it was set based on the model training for larger data sizes (possibly on the entire filtered LAION data set). With 0.3 weight decay, we observe a consistent decrease in the population memorization leakage, as shown by the PPG, PRG and AUCG values for $wd = 0.3$ in Figure 5, but the values are still significantly high. We also explored setting weight decay to 0.01 and 1.0, but they either adversely impacted the model utility or severely increased memorization. Thus while tuning wd does not completely mitigate memorization, we can get a reasonable trade-off in the neighbourhood of $wd = 0.3$.

5.4 Text Randomization

During model training, the CLIP models increase the cosine similarity between the matching image-caption pairs while simultaneously decreasing the cosine similarity between mismatched pairs to reduce the contrastive loss. While it is common to augment the training images to reduce overfitting, the text captions are not randomized. This could lead to the model overfitting on the text captions when minimizing the contrastive loss. To avoid this, we propose text randomization as a defense. For COCO subset of the training set, we randomly choose one out of the five captions for each image per epoch during training. For filtered LAION subset, we randomly mask a fraction of caption tokens since only a single caption is available per image in the filtered LAION data set. We vary the masking ratio between 0 (no masking), 0.3 and 0.5 (randomly mask half of the tokens).

We find this defense to work the best in mitigating déjà vu memorization but at the cost of ImageNet zero-shot accuracy. As shown in Figure 5, using a masking ratio of 0.3 reduces the ImageNet zero-shot accuracy from 25.2% (in the default case when $mr = 0.0$) to 24.1%, but at the same time this significantly reduces memorization. The PPG metric reduces from 9.1% to 3.4%, and the PRG metric reduces from 9.2% to 3.8%. Moreover, the recall CDF gap (AUCG) also reduces from 0.034 to 0.013. Further increasing the masking ratio to 0.5 mitigates the risk even more. PPG reduces to

3.0%, PRG reduces to 1.9%, and AUCG reduces to only 0.007. However, we note that text masking has a positive impact on ARO benchmark utility as shown in Figure 7. This is because masking avoids overfitting on specific text tokens making the models less likely to behave like bag-of-words. Thus text masking achieves the best utility trade-offs. We would expect a significant drop in the model utility if we further increase mr since the captions would have considerably less information.

Key Observations. We study the impact of tuning four regularization parameters: number of training epochs, temperature, weight decay and masking ratio. We find that early stopping reduces memorization but at the cost of model utility. Increasing the temperature increases the model zero-shot accuracy and decreases memorization up to a certain threshold, beyond which the model utility begins to decrease. Surprisingly, we find that the default value of 100 already gives the optimal results. Similar to temperature, increasing the weight decay increases the model utility and decreases the memorization up to a certain threshold. We find 0.3 weight decay to achieve the best results for a model trained over 10M data. We observe a sharp decrease in model utility beyond this value. Text masking seems to be most effective in mitigating memorization. Increasing the masking ratio decreases memorization and also decreases the model utility. Masking ratio of 0.3 achieves a good trade-off by significantly reducing memorization while only moderately impacting the model utility.

6 Discussion

Prior works have mainly shown memorization in the uni-modal setting: either for the language models [Carlini et al., 2019] or for vision models [Meehan et al., 2023]. We have demonstrated that even in the complex multi-modal setting, ML models suffer from memorization. Moreover, while prior works have only evaluated memorization for small training data sizes (typically on the scale of 1 million or less), we show memorization on a wide scale, from 1 million to 50 million training set size. Our experiments show that while the population-level memorization metrics decrease with increase in the training set size, there remain strongly memorized examples as exemplified by the sample-level memorization where the model disproportionately memorizes a subset of records.

Careful tuning of right hyper-parameters can, however, mitigate this memorization risk. We propose a suite of metrics to quantify *déjà vu* memorization in hope of guiding ML practitioners to train models in a safe way. These metrics not only quantify the risk in a meaningful and interpretable manner, but are also sensitive to the tuning of the mitigation parameters, thereby aiding the practitioners in choosing the right model hyper-parameter values that achieve a good utility-risk trade-off.

Below we discuss some detailed discussions and limitations of our work.

Not applicable to out-of-box models. Since our tests require access to two models, *target* and *reference*, along with the underlying training set, we note that this can not be directly applied to measure memorization in out-of-the-box pre-trained models as there is no reference model for such cases. We leave this case as a future work.

Distinguishing memorization from learning. A model can memorize and generalize (or learn) at the same time. This can happen at a sub-population level, where the model memorizes rare concepts and generalizes to common concepts, or even at a sample level, where memorization is required for learning rare concepts as theorized in Feldman [2020]. *Déjà vu* memorization is meant to go beyond this, and instead examine when a model that is trained on an image with a generic caption (i.e., they do not describe the image in high detail), memorizes many small details about the associated image (i.e., what objects are present in the image) when given the caption. In other words, we define *déjà vu* memorization as what can be inferred about the training image from its caption beyond simple correlations, which can happen through both learning and memorization in the traditional sense.

Extending beyond objects. While our approach is also applicable to annotations that go beyond objects, this is not in the scope of this work. Even in this setting, the prior state-of-art approach [Meehan et al., 2023] only considers a single object label per image (ImageNet) and none of the prior works consider a. multimodal setting, b. large training size sizes, and c. multiple objects per image.

Relation to Overfitting. *Déjà vu* memorization measures overfitting at a more granular level—instead of a binary decision, it measures to what *degree* the model overfits a training sample.

Acknowledgements

We thank Diane Bouchacourt for helpful feedback. We would also like to thank Amro Abbas for helping in obtaining the de-duplicated version of filtered LAION data set and Evgenia Rusak for helping with the OpenCLIP implementation.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv:2303.09540*, 2023.

Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Conditioned and Composed Image Retrieval Combining and Partially Fine-Tuning CLIP-Based Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022.

Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S Yu. Deep Visual-Semantic Hashing for Cross-Modal Retrieval. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX Security Symposium (USENIX Security 19)*, 2019.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, et al. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting Training Data from Diffusion Models. In *32nd USENIX Security Symposium (USENIX Security 23)*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009.

Rachna Dhamija and Adrian Perrig. Déjà Vu–A User Study: Using Images for Authentication. In *9th USENIX Security Symposium (USENIX Security 00)*, 2000.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021. URL `https://doi.org/10.5281/zenodo.5143773`.

Bargav Jayaraman and David Evans. Are Attribute Inference Attacks Just Imputation? *arXiv:2209.01292*, 2022.

Bargav Jayaraman, Esha Ghosh, Melissa Chase, Sambuddha Roy, Wei Dai, and David Evans. Combing for Credentials: Active Pattern Extraction from Smart Reply. *arXiv:2207.10802*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2017.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-Training for Unified Vision-Language Understanding and Generation. In *International Conference on Machine Learning*. PMLR, 2022.

Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014.

Casey Meehan, Florian Bordes, Pascal Vincent, Kamalika Chaudhuri, and Chuan Guo. Do SSL Models Have Déjà Vu? A Case of Unintended Memorization in Self-supervised Learning. *arXiv:2304.13850*, 2023.

Ron Mokady, Amir Hertz, and Amit H Bermano. ClipCap: CLIP Prefix for Image Captioning. *arXiv:2111.09734*, 2021.

Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. Filtering, Distillation, and Hard Negatives for Vision-Language Pre-Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

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 *International conference on machine learning*. PMLR, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In *International Conference on Machine Learning*. PMLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv:2204.06125*, 2022.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Déjà Vu: an empirical evaluation of the memorization properties of ConvNets. *arXiv:1809.06396*, 2018.

Christoph Schuhmann, Richard Vencu, Romain, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. *arXiv:2111.02114*, 2021.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018. URL https://aclanthology.org/P18-1238.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The New Data in Multimedia Research. *Commun. ACM*, 2016. doi: 10.1145/2812802. URL https://doi.org/10.1145/2812802.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748*, 2018.

Daixin Wang, Peng Cui, Mingdong Ou, and Wenwu Zhu. Deep Multimodal Hashing with Orthogonal Regularization. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023.

Kaiyu Yang, Jacqueline Yau, Li Fei-Fei, Jia Deng, and Olga Russakovsky. A study of face obfuscation in imagenet. In *International Conference on Machine Learning (ICML)*, 2022.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022a.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *arXiv:2206.10789*, 2022b.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=KRLUvxh8uaX`.

Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. Analyzing Information Leakage of Updates to Natural Language Models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, 2020.

Peng-Fei Zhang, Yang Li, Zi Huang, and Hongzhi Yin. Privacy Protection in Deep Multi-Modal Retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting Twenty-thousand Classes using Image-level Supervision. In *ECCV*, 2022.

A License of the assets

A.1 License for the code

The licensing information for OpenCLIP [Ilharco et al., 2021] can be found at https://github.com/mlfoundations/open_clip/blob/main/LICENSE. We use the code from Meehan et al. [2023] for memorization quantification, the licensing information can be found at https://github.com/facebookresearch/DejaVu?tab=License-1-ov-file#readme. For object annotations, we use Detic [Zhou et al., 2022], the licensing information can be found at https://github.com/facebookresearch/Detic/blob/main/LICENSE.

A.2 License for the data sets

We use ImageNet [Yang et al., 2022] for which the license can be found at https://www.image-net.org/download.php. We use a filtered version of LAION [Radenovic et al., 2023] (which we call filtered LAION) for which licensing information can be found at https://github.com/facebookresearch/diht/blob/main/LICENSE. The licensing information for the MS COCO data set [Lin et al., 2014] that we use can be found at https://cocodataset.org/#termsofuse. We also use Shutterstock data set which is a private licensed data set consisting of 239M image-caption pairs.

B Background and Related Work

Foundation models, such as large language models, have been long known to memorize their training data in ways that enable easy extraction. For example, a line of work [Carlini et al., 2019, 2021, Zanella-Béguelin et al., 2020, Jayaraman et al., 2022] has shown that large language models exactly memorize sequences of text tokens from the training data, and these text tokens can be extracted. Somepalli et al. [2023], Carlini et al. [2023] showed that diffusion models can generate images that are semantically and stylistically similar to training images or even their near-exact copies under certain circumstances. However, almost all prior studies that demonstrate this kind of memorization focus on generative models – language or vision – where measuring memorization is easier because of the presence of a decoder. In contrast, our work is concerned with representation learning models, where we simply have an encoder.

Sablayrolles et al. [2018] study déjà vu [1] memorization in neural networks and show that it is possible to infer whether an image or a subset of images was used in model training. Our work is also closely related to Meehan et al. [2023], which measures déjà vu memorization in image representation models. They show that given the representation of the background of an image, (such as water), the label of its foreground object (such as black swan) can be predicted reliably. Moreover, this prediction is significantly more accurate for images in the training set of a model, thus showing that the models memorize their training data beyond the bounds of spurious correlation. However, such a simple foreground-background measurement does not directly apply to the more complex, multi-modal Vision Language Models where the two modalities may leak training data in more subtle and complicated ways. Our work builds upon their test, and extends it to VLMs.

Finally, there has been a body of work on empirical measurement of privacy, and broadly speaking, there are three main kinds of attacks. In membership inference [Shokri et al., 2017], the goal is to determine if a specific data point was used to train a model. In attribute inference [Yeom et al., 2018], the goal is to infer unknown features or attributes of a data point based on a model trained on this or similar points. Finally, training data reconstruction attacks [Fredrikson et al., 2015] aim to recover one or more training data points given a model and some auxiliary information. Our work falls within the purview of attribute inference. However, unlike most attribute inference attacks which

[1] We note that many prior works have used the term "déjà vu" in different contexts. Dhamija and Perrig [2000] use this to refer to the ability of humans to recognize images, and they use it as a proxy for password-based authentication. Sablayrolles et al. [2018] denote déjà vu to essentially mean membership inference, where they test if a model *remembers* if an image was used in training. Meehan et al. [2023] refer to déjà vu as the ability of inferring foreground objects from vision models given a background patch of pixels. We use this term to refer to a vision–language model's ability to recall the individual objects in the training images.

were shown to be forms of statistical imputation [Jayaraman and Evans, 2022], our tests directly measure how much more effective attribute inference can be when a data point is in the training set of a model.

C Detailed Experiment Setup

For our experiments we use OpenCLIP [Ilharco et al., 2021] to train the models. For filtered LAION experiments, we train models over subsets of filtered LAION [Radenovic et al., 2023] and MS-COCO [Lin et al., 2014] data sets. For Shutterstock experiments, we train models over various subsets of Shutterstock data set, a privately licensed dataset of 239M image-captions pairs.

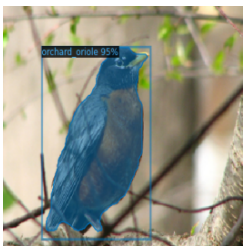
Obtaining Data Splits. As discussed in Algorithm 1, our test requires disjoint training sets A and B to train the models f_A and f_B respectively, and additionally we require a public set P, that has no overlap with A and B, for our nearest neighbor search. Moreover, for our tests to be meaningful we need to remove duplicate image–caption pairs otherwise the kNN test becomes trivial if the same sample is also present in the public set and as a result we would overestimate memorization. Conversely, if the same sample is present in both A and B sets, then it is harded to distinguish the outputs of two models and we would underestimate memorization. This type of duplication is common in internet-scraped data sets such as filtered LAION and Shutterstock. We perform semantic deduplication over filtered LAION data set using the procedure of Abbas et al. [2023] to obtain 220M deduplicated image–caption pairs. The Shutterstock data set has different type of duplicates— multiple images are present with same verbatim captions. So we do a simpler yet effective deduplication by considering only one unique image per caption. This reduces the overall data set size to around 103M image-caption pairs.

- For filtered LAION experiments: To obtain the two non-overlapping training sets for filtered LAION experiments, we sample 40K image–text pairs from COCO data set and 1M/10M/50M image–text pairs from filtered LAION data set to form A set. We do the same from the remaining pool of data to obtain the B set. Since the COCO part of the A and B sets is insignificant compared to the filtered LAION portion of the sets, we only count the filtered LAION portion size for simplicity when we say we sample 1M/10M/50M training sets. To obtain the filtered LAION-50M public set, we sample 50M pairs from the remaining pool of deduplicated filtered LAION which has 120M pairs (after removing the largest A and B sets from the original 220M data set). We include most of the results on this public set in Appendix D. Since this data set may contain human faces, we perform face-blurring on all the sets. We also take the 1.28M images from ImageNet data set [Deng et al., 2009] and perform face-blurring to form our ImageNet public set.

- For Shutterstock experiments: We take the caption-level deduplicated data set consisting of 103M image–caption pairs and randomly split it into 40M + 40M + 20M sets. The first two 40M sets are used to obtain the 1M/10M/40M A and B sets respectively. The last 20M set is used as the public set. A small portion of the remaining 3M data is used as a hold-out set for hyper-parameter tuning during model training.

Model Hyper-Parameter Settings. We use the ViT-B-32 CLIP model architecture consisting of around 151M trainable parameters and train the models for 200 epochs using Adam [Kingma and Ba, 2017] optimizer with cosine learning rate scheduler and a learning rate of 0.0005. For filtered LAION experiments, we use 256 Nvidia Quadro GP100 GPUs with 16GB VRAM to train the models in parallel with an effective batch size of 16 384. We set the weight decay to 0.1 and use 1000 warmup steps for the learning rate scheduler. For Shutterstock experiments, we use 32 Nvidia A100 GPUs with 80GB VRAM to train the models in parallel with an effective batch size of 32 768. We set the weight decay to 0.2 and warmup to 2000 steps. All the model training runs use 512GB RAM and the training time scales with the data size: training on 10M data size takes around 2 days and training on 50M data size takes around 10 days to complete. All other hyper-parameters are set to the default value as used in OpenCLIP; we do an ablation study on the impact of temperature and weight decay in Section 5.

Obtaining Object Annotations. For quantitative evaluation of our nearest neighbor tests, we require detailed object annotations for the A, B and P sets. Both Shutterstock and filtered LAION



Label:
Orchard Oriole

(a) ImageNet Sample.

Caption:
Several kitchen workers making dishes in commercial kitchen.
Labels:
Catsup Bottle, Pot, Hat, Plate, Dinner Napkin, Finger Bowl, Soda Can, Bottle, Dripping Pan, Cup, Work Shirt, Bowl, Apron, Person, Belt, Pan.

(b) COCO Sample.

Figure 6: Comparing images from ImageNet and COCO data sets. The ImageNet images only have single label per image but COCO images have complex scenes with multiple object labels. Additionally, COCO images have accompanying text captions. Label annotations with bounding boxes are highlighted in blue for both the images.

data sets only have image captions and no object annotations. ImageNet originally has only one object annotation per image, as shown in Figure 6. Hence, we use an open-source annotation tool, called Detic [Zhou et al., 2022], to obtain multiple fine-grained object annotations per image for all our data sets. This tool can annotate all the 21K ImageNet objects. Detic uses a default threshold of 0.5 to identify object bounding boxes (i.e., any bounding box that has more than 0.5 confidence is considered for annotation). For Shutterstock we use 0.3 threshold as the 0.5 threshold results in nearly 17% images with no annotations. For all other data sets, we use the default value of 0.5. Even though COCO has multiple object annotations, its class label space is small (i.e., only 80 unique classes). Hence we use Detic on COCO to extend its annotations and to make the label annotations consistent across all the data sets we use. Figure 1 shows the sample images with multiple object annotations obtained using Detic.

Limitations in Experimental Evaluation. We find that the object annotation tool, Detic [Zhou et al., 2022], is not always accurate. For instance, the tool often classifies a 'polar bear' as 'jaguarundi'. However, our experiments rely on the relative gap in the object detection between the target and reference models and as such are robust to these inaccuracies as long as the annotations are consistent across the images. For instance, if the 'polar bear' is classified as 'jaguarundi' across all the public set images, the gap between the 'polar bear' detection accuracy of target and reference models, based on our nearest neighbor test, will remain consistent. While the absolute numbers in our quantitative tests may vary based on the object annotation tool used, our experimental observations would not change.

D Additional Results with filtered LAION-50M

In Section 4, we discussed the memorization results considering the ImageNet as the public set for our nearest neighbor test. Here we discuss the results with a much larger filtered LAION-50M data set as the public set. While the overall trend remains the same as with the ImageNet, with a richer public set, we are able to achieve a larger memorization gap for our models.

D.1 Sample-Level Memorization

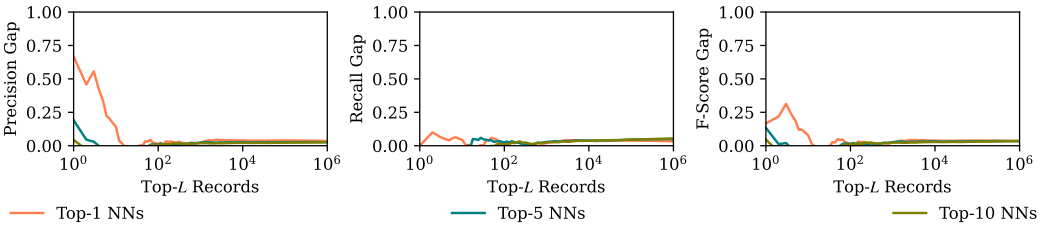
Similar to the sample-level evaluation for ImageNet public set in Section 4.2, we evaluate the gap in precision, recall and F-scores of top-k records sorted with respect to the minimum embedding distance when considering filtered LAION-50M as the public set for the nearest neighbor test. Figure 8a shows the memorization gap of the top-k records. We note a greater precision gap with top-1 nearest neighbor when compared to the case where ImageNet was used as a public set (see Figure 4a). However, the recall gap is lower with this public set. These variations could be due to the nature of the public set— many filtered LAION images have few or no annotations. This does not mean that the sample-level memorization risk is lower. As shown in Figure 8b, the memorization gap is much higher for this public set when we sort the records in the decreasing order of the number of correct predictions made by the target model using the nearest neighbor test. This corroborates

Figure 7: ARO benchmark accuracy comparison for various models. Top figure compares the accuracy of various baseline models on three compositional reasoning tasks: Visual Genome Attribution, Visual Genome Relation and COCO Order. The pretrained CLIP models are trained on 400M private dataset, whereas our LAION and Shutterstock models are trained on smaller subsets of the respective datasets. Our models are comparable to the pretrained CLIP models in VG Relation and Attribution Tasks. The middle figures show the impact of temperature (left) and weight decay (right) on the ARO accuracy for our models trained on 10M subset of filtered LAION dataset. As shown, the default parameter values (shown by asterisk) achieve the best values for most cases. The bottom figures show the impact of text masking on ARO accuracy for our models trained on 10M subsets of filtered LAION (left) and Shutterstock (right) datasets. Our text masking does not deteriorate the model utility, and in fact further boosts ARO accuracy for COCO ordering task. This is because text masking avoids overfitting on specific text tokens. Thus, unlike the unmitigated CLIP models, the mitigated models are less likely to behave like bag-of-words.

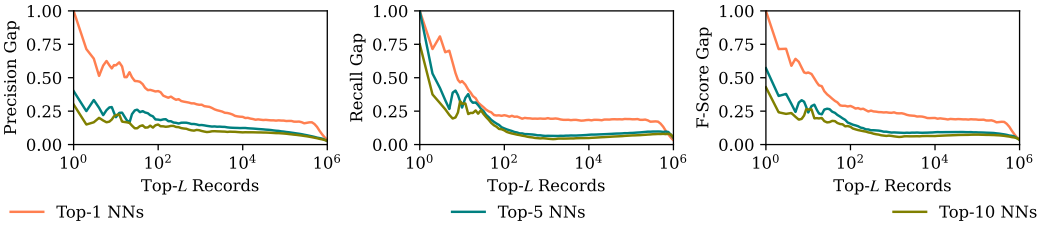
our population-level memorization results in Figure 2 where we find a higher memorization gap with filtered LAION-50M public set.

D.2 Mitigation

We observe similar trends for mitigation with different regularization parameters as with the ImageNet case. Figure 9 shows the impact of different parameters on the memorization. Since the filtered LAION-50M public set is much larger than the ImageNet public set, the overall memorization values are higher due to the public set nearest neighbors being more representative of the target image, and thus capturing more objects. However, the trend remains the same. Increasing the temperature decreases the memorization, but the default value of 100 is close to optimal as the trade-off between memorization and model utility is the best. Increasing the weight decay improves



(a) Records are sorted w.r.t. the minimum embedding distance between target caption and public images.



(b) Records are sorted w.r.t. the decreasing number of correct object predictions for target model.

Figure 8: Sample-level memorization gap between target and reference models when predicting top-10 objects for different top-L records. Models are trained on disjoint 10M subsets of filtered LAION data set for 200 epochs and filtered LAION-50M public set is used for the KNN test.

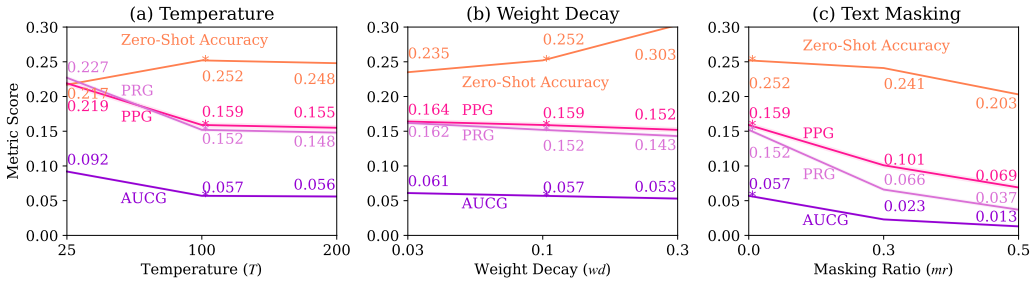
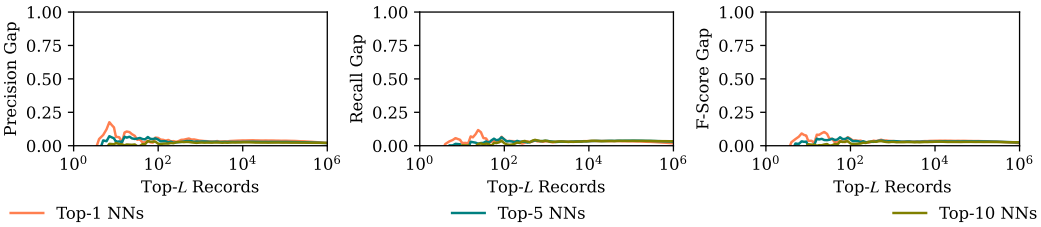


Figure 9: Effect of parameter tuning on ViT-B-32 CLIP models trained on 10M subset of filtered LAION for 200 epochs. Memorization evaluation is done using filtered LAION-50M as public set. Default setting is highlighted with asterisk. For the memorization metrics, we report the *mean* $\pm$ *std* values (*std* ≤ 0.003) over 100 repetitions of randomly sampling 10% of records with replacement.

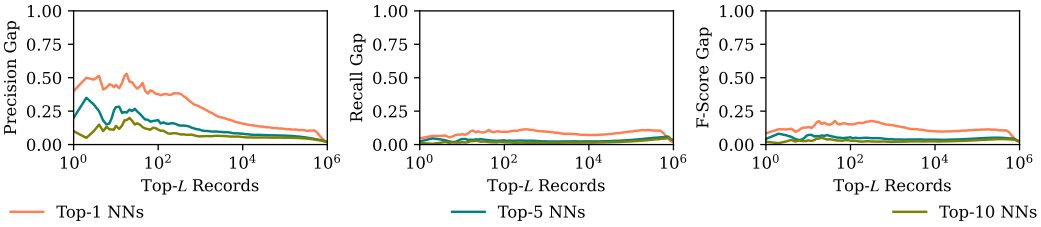
the model utility (indicated by the zero-shot accuracy) and decreases memorization. Weight decay of 0.3 gives near optimal trade-offs. Further increasing wd to 1.0 results in a drastic decrease in model utility, and thus we do not include the results. Increasing the masking ratio from 0 to 0.5 significantly reduces the memorization but at the cost of model utility. While the optimal value of mr would depend on the application and how much tolerance on the model utility loss is acceptable, we find that $mr = 0.3$ achieves a significant reduction in memorization while only moderately impacting the zero-shot accuracy, as shown in Figure 9. Any further increase in mr beyond 0.5 would greatly sacrifice the model utility and thus is not recommended.

E Additional Results with Shutterstock

Similar to the sample-level evaluation for models trained on filtered LAION data set in Section 4.2, we evaluate the gap in precision, recall and F-scores of top-k records sorted with respect to the minimum embedding distance for models trained on Shutterstock data set when considering SS-20M as the public set for the nearest neighbor test. Figure 10a shows the memorization gap of the top-k records. We note smaller precision and recall gaps for this data set. This is due to two reasons: (a) nature of the data set— Shutterstock data set has many similar images even after we



(a) Records are sorted w.r.t. the minimum embedding distance between target caption and public images.



(b) Records are sorted w.r.t. the decreasing number of correct object predictions for target model.

Figure 10: Sample-level memorization gap between target and reference models when predicting top-10 objects for different top-L records. Models are trained on disjoint 10M subsets of Shutterstock data set for 200 epochs and SS-20M public set is used for the KNN test.

do the caption-level deduplication (see Appendix C) so even the referece model performs well on this data set, and (b) the model hyper-parameter settings for this training set size is possibly sub-optimal— the model zero-shot accuracy on ImageNet seems to be the highest at 20 epochs (18.16%) and it slightly decreases till 200 epochs (17.49%) when trained on 10M subset of Shutterstock data. Figure 10b shows the memorization gap when we sort the records in the decreasing order of the number of correct predictions made by the target model using the nearest neighbor test. As expected this gap is much higher than the previous case. Overall, the trends are similar to the filtered LAION experiments.

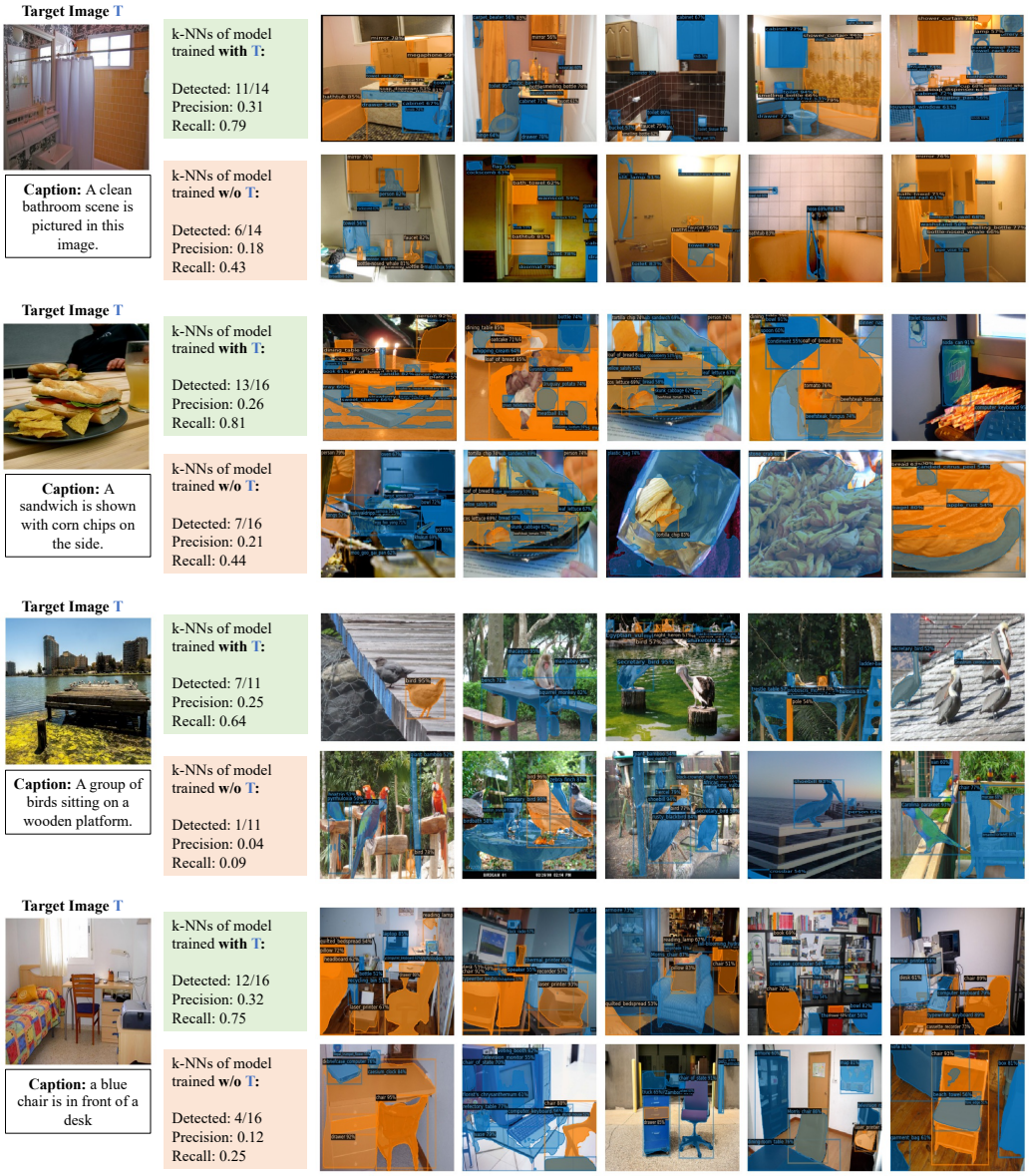


Figure 11: Additional examples showing *déjà vu* memorization. Target images are from COCO training set and the public images are from ImageNet data set. The objects annotated in orange are true positives, i.e., the ones present in the target image, and the objects annotated in blue are false positives.

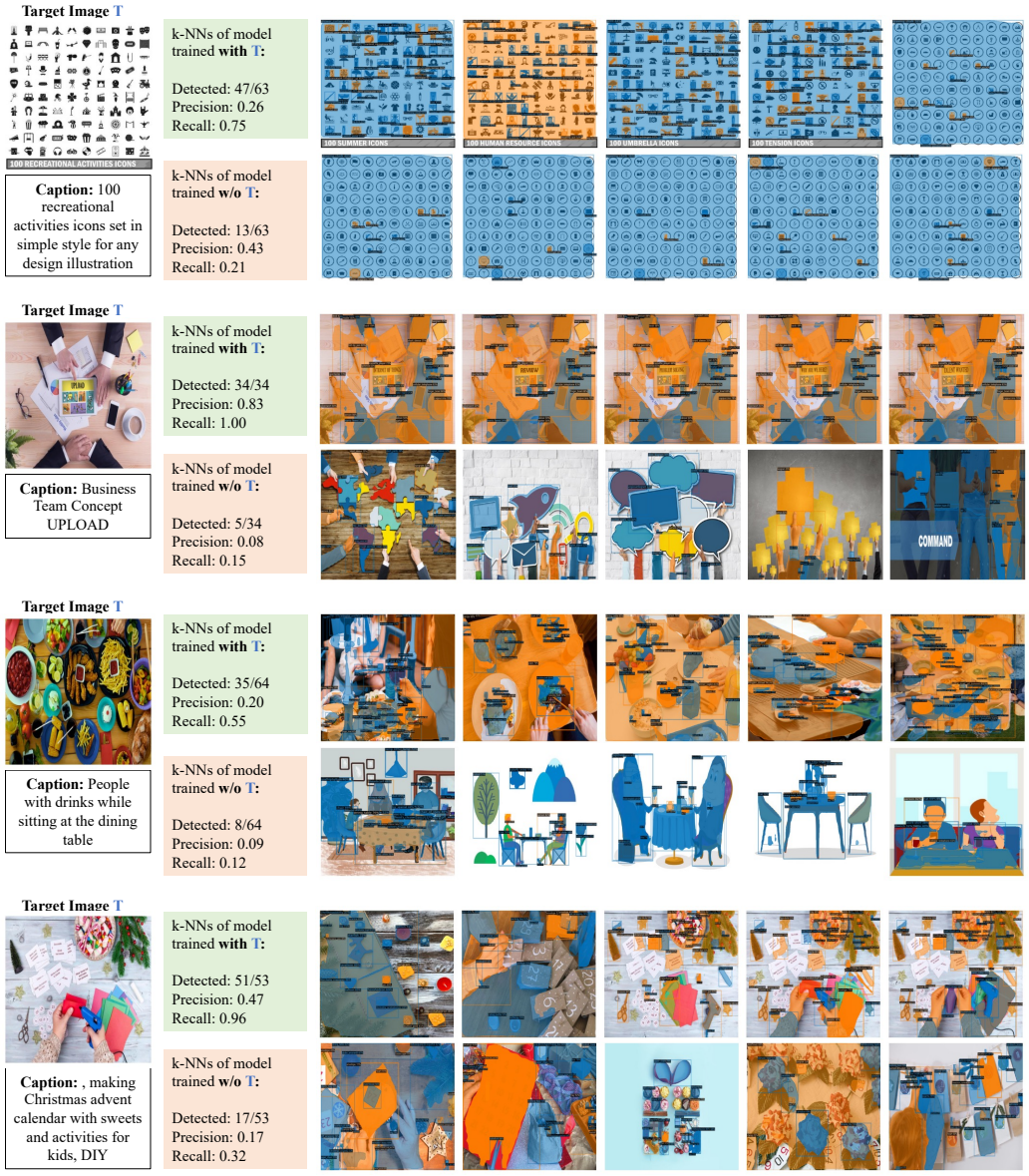


Figure 12: Additional examples showing *déjà vu* memorization. Target images are from Shutterstock training set and the public images are from SS-20M public set. The objects annotated in orange are true positives, i.e., the ones present in the target image, and the objects annotated in blue are false positives.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction about the deja vu memorization for VLMs are accurately reflected in the paper's contributions. Namely, this is the first memorization study in the VLM space and that we propose a novel way to quantify this memorization. We also explore various mitigations and show which ones work and which do not, although we note the fact that there could be other effective mitigations but were not explored due to the computational limitations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mention the limitations about the number of possible mitigation strategies explored in the work in Section 1. We also discuss the experimental limitations of our work due to the usage of automatic object detection tool in Appendix C. Finally, in Section 6 we mention that our work can not be directly applied to quantify memorization in out-of-box pre-trained models as our work requires access to a reference model.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There are no theoretical results in the paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We discuss all the model hyper-parameter settings and how we obtain the different data set splits for our experiments in detail in Appendix C, along with a discussion of what computational resources were used for the experiments. We also describe how we obtain the object annotations for all the images. We point out to which opensource repository we use for training our models. We also describe which data sets were used in our experiments and we cite the relevant sources where possible. We made the memorization quantification code publicly available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: While the code is publicly available, the data sets used cannot be made public due to the licensing and general nature of the data. We do cite all the relevant data sources wherever possible so the readers can refer to those sources. Moreover, we provide all the necessary parameter settings to replicate the results of the paper, possibly on some publicly available licensed data sets.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We explain all the necessary training details in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For our population-level memorization metrics, we report the mean and standard deviation values over 100 repetitions of randomly sampling 10% of records with replacement. However, since the standard deviation is less than 0.003, these error bars are not visible in the plots, such as Figure 2, Figure 5, and Figure 9.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss all the computational resources and time taken to train the models in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We ensure NeurIPS Code of Ethics are followed, especially with the usage of data sets. For public data sets that contain human faces, such as filtered LAION, COCO and ImageNet, we blur all the faces prior to training the models or for any other analysis.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We demonstrate that multi-modal models can memorize the objects present in the training images. While our intent is to make this risk more transparent to aid researchers get a deeper understanding of the memorization issue inherent in representation learning models, an adversary could potentially use this to launch attacks on CLIP-style models. Although such an adversary would need non-trivial amount of background information for a successful attack. For instance, the adversary would need at least access to two models such that exactly one of the two is trained on a target image-text pair. They would also need access to the underlying training data for the target model. We discuss the approaches that are effective at mitigating this risk.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: All our models are trained from scratch and we do not release these models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original code base of OpenCLIP and Detic that we use for training the models and object annotation respectively. We also include the licensing information for all the code and data sets we use in Appendix A. We included the same in our code that is publicly released.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We released the code base which is well documented. We do not release models or data sets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Crowdsourcing was not used.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects were used so this is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.