PERSONALIZED REPRESENTATION FROM PERSONALIZED GENERATION

Anonymous authors
Paper under double-blind review

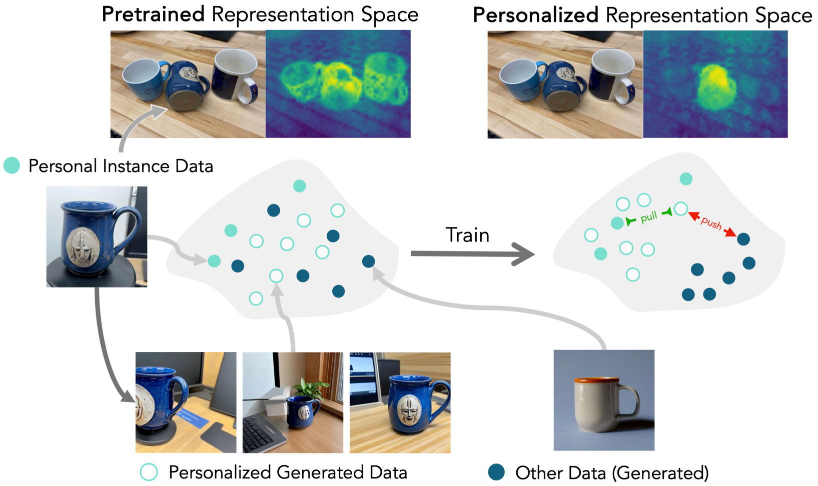
Pretrained Representation Space
Personalized Representation Space
● Personal Instance Data
Train
pull
push
○ Personalized Generated Data
● Other Data (Generated)

Figure 1: **Learning personalized representations from limited real data.** In this paper we explore whether and how synthetic data can be used to train a personalized representation. Given a few real images of an instance, we generate novel images and contrastively fine-tune a general-purpose pretrained model to learn a personalized representation, useful for diverse downstream tasks.

ABSTRACT

Modern vision models excel at general purpose downstream tasks. It is unclear, however, how they may be used for personalized vision tasks, which are both fine-grained and data-scarce. Recent work has successfully applied synthetic data to general-purpose representation learning, while advances in T2I diffusion models have enabled the generation of personalized images from just a few real examples. Here, we explore a potential connection between these ideas, and formalize the challenge of using personalized synthetic data to learn *personalized representations*, which encode knowledge about an object of interest and may be flexibly applied to any downstream task relating to the target object. We introduce an evaluation suite for this challenge, including reformulations of two existing datasets and a novel dataset explicitly constructed for this purpose, and propose a contrastive learning approach that makes creative use of image generators. We show that our method improves personalized representation learning for diverse downstream tasks, from recognition to segmentation, and analyze characteristics of image generation approaches that are key to this gain.

1 INTRODUCTION

Representation learning in computer vision seeks to learn general-purpose encodings for objects or semantic concepts that may be flexibly applied to downstream tasks such as recognition and semantic segmentation. In recent years we have seen a surge of interest in *personalized vision* – where a user can easily develop customized models for objects of their personal interest, e.g., a model capable of detecting their pet dog in personally-collected images (Zhang et al., 2023; Cohen et al., 2022; Nitzan et al., 2022). Among other benefits, personalized systems can keep data private; preferably these models are trained locally, without needing to share user data to a centralized repository, or access other users' data. The personalized setting has two critical challenges. First, it is data-scarce: Curated data collection is time-consuming and expensive; a user would ideally need only provide a few examples of their object to obtain a personalized model. Second, it can be extremely fine-grained; e.g., recognizing an individual dog as opposed to recognizing the category "dog".

While modern vision models have proven successful for general-purpose tasks, adapting their representations to fine-grained problems with scarce labeled data remains challenging (Zhang et al., 2024; Radford et al., 2021; Cohen et al., 2022; Stevens et al., 2023). As shown in Figure 1, we contrast *general-purpose representations* with the notion of a *personalized representation*: a specialized representation space that encodes the knowledge about an instance of interest needed for a variety of downstream personalized tasks. In this paper, we ask: **Is it possible to learn a personalized representation from only a few real images of a single instance?**

Works such as (Tian et al., 2023b) have shown that, when intelligently paired with contrastive objectives, synthetic data can enable learning strong *general-purpose* visual representations. Other works have investigated *personalized generation* (Gal et al., 2023; Ruiz et al., 2022), but do not extend to representation learning. Our work targets the combination of these ideas: can personalized generation provide effective synthetic data for training *personalized representations*? We explore what makes for useful generative data augmentation for personalized representation learning and how to best learn from that data. We evaluate our learned representations for four downstream tasks: classification, retrieval, detection, and segmentation, and find that performance universally improves.

In summary, our contributions are the following:

- **Personalized representations** trained with synthetic data, using only three real examples of an instance, **significantly outperform pretrained counterparts** across datasets, backbones, and downstream tasks.
- We introduce new mechanisms for evaluating personalized representations, including **PODS – Personal Object Discrimination Suite – a new dataset** of 100 personal objects under specific distribution shifts, and reformulations of existing instance-level datasets.
- Leveraging **additional resources can significantly improve** personalized representations. While pretrained T2I models are key to achieving the best performance, comparable results can be obtained with **fewer computational resources**.
- Different generators introduce **unique biases/limitations** that affect learned representations.

2 RELATED WORKS

Personalized visual generation. Early efforts to personalize generated images attempted to edit specific people or styles given user inputs with Generative Adversarial Networks (GANs) (Bau et al., 2019; Roich et al., 2021; Alaluf et al., 2021; Dinh et al., 2022; Nitzan et al., 2022). Recent efforts focus on T2I diffusion models, usually learning a unique identifier for a target object given a few images. Textual Inversion (Gal et al., 2023) freezes a pretrained generative model then learns a unique and personal text token for the object, which can be conditioned on for generation. NeTI (Alaluf et al., 2023) enhances expressivity and editability by learning different token embeddings for each diffusion timestep and U-Net layer. DreamBooth (Ruiz et al., 2022) fine-tunes the entire T2I model to produce more accurate images of the target concept. CustomDiffusion (Kumari et al., 2022) instead fine-tunes a subset of model weights, and enables joint training over multiple concepts. Follow-up works to these have sought to improve the efficiency and accuracy of personalized generations (Ruiz et al., 2023; Arar et al., 2023; Wei et al., 2023; Han et al., 2023; Guan et al.), e.g., finetuning-free personalization methods that reduce computational cost (Shi et al., 2024; Chen et al., 2024; Huang et al., 2024; Ma et al., 2024).

Personalized recognition and representations. Personalized vision involves tailoring vision models to user-specific concepts and preferences. PerSAM (Zhang et al., 2023) extends the Segment-Anything Model (Kirillov et al., 2023a) to segment user-specified objects with a few example images and masks. Personalization has also been explored for image captioning (Wang et al., 2023; Chunseong Park et al., 2017; Park et al., 2018), pose estimation (Nguyen et al., 2024b), and image retrieval via textual inversion: finding a mapping of images to text tokens (Saito et al., 2023;

Karthik et al., 2023; Baldrati et al., 2023; Cohen et al., 2022; Yeh et al., 2023). Among the textual inversion works, PALAVARA (Cohen et al., 2022) enables personalization for both global and dense vision tasks but relies on large-scale captioned data for the inversion process. In contrast, our approach requires only a few images, without annotations, from the user. Recent concurrent works have also personalized vision-language models for tasks like VQA and object recognition (Nguyen et al., 2024a; Alaluf et al., 2024). Unlike these prior methods, we personalize general-purpose vision backbones using a self-supervised framework over generated data, achieving strong performance across both image-level tasks (e.g., retrieval) and dense prediction tasks (e.g., detection and segmentation) without the need for large-scale data.

Re-Identification. Personalized recognition is closely related to re-identification, in which a model is tasked with recognizing objects (Sun et al., 2004) or faces (Turk & Pentland, 1991) of the same identity. Early works in Re-ID explored metric learning on hand-crafted features (Ojala et al., 2002; Gray & Tao, 2008; Zhao et al., 2017);later methods learned deep metrics with supervised/unsupervised signals (He et al., 2021; Taigman et al., 2014; Schroff et al., 2015). Recent metric learning works use large curated datasets to train on thousands of unique instances of a certain category (typically humans (Zheng et al., 2015; Yadav & Vishwakarma, 2024) or vehicles (Liu et al., 2016; Amiri et al., 2024)). While our work involves training features with contrastive losses, we focus on personalizing pre-trained features for a single instance with a few images.

Training on synthetic data. Training on synthetic data has been extensively investigated to tackle issues like privacy preservation, data imbalance, and data scarcity (Sakshaug & Raghunathan, 2010; Tanaka & Aranha, 2019; Khan et al., 2019; Jahanian et al., 2021; Tucker et al., 2020). Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have further unleashed such potential in zero-shot settings (He et al., 2022b), few-shot settings (He et al., 2022b; Trabucco et al., 2023; Lin et al., 2023), out-of-distribution scenarios (Sariyildiz et al., 2023; Bansal & Grover, 2023; Jung et al., 2024), and supervised classification (Yeo et al., 2024; Kupyn & Rupprecht, 2024). These works note the importance of the classifier-free guidance scale (Sariyildiz et al., 2023; Tian et al., 2023b) and prompt selection (Lei et al., 2023), and propose post-processing filtering (He et al., 2022b) when using off-the-shelf T2I models. Alternatively, (Azizi et al., 2023) and (Yuan et al., 2023) fine-tune diffusion models on ImageNet and show improved classification performance when supplementing real with synthetic data. Similarly, (Zhou et al., 2023; Trabucco et al., 2023) invert training images as conditions for generating new synthetic images. Other studies address data-imbalance (Shin et al., 2023), domain shifts (Yuan et al., 2022), scaling synthetic data (Fan et al., 2023), and applications to various tasks, including segmentation (Wu et al., 2023), general-purpose representation learning (Tian et al., 2023b;a), and CLIP training (Hammoud et al., 2024).

3 METHODS

This paper tackles two questions: how to achieve personalized visual representation by leveraging generative models, and what factors are essential to producing highly effective training data. In Section 3.1 we formalize the personalized representation task. Our three-stage method is then illustrated in Figure 2. We prepare a personalized generator from a few target instance images (Section 3.2) and produce synthetic personalized data (Section 3.3). We then train a personalized representation on the generated data with a contrastive objective (Section 3.4). Lastly, we consider scenarios with additional annotations and data, and how to incorporate them to enhance personalization.

3.1 FORMALIZING THE PERSONALIZED REPRESENTATION CHALLENGE.

We assume access to a small dataset of real images $\mathcal{D}_R$ of a specific object c, and the generic category c_{pr} of the object. We use a generative model $g_\theta(x)$ to synthesize a novel dataset $\mathcal{D}_S$ of images of c and train a personalized representation by adapting a general purpose vision encoder f_ϕ.

We assume we are only provided images c (which we also denote as an *instance*) for training our personalized representation. We evaluate on global and local downstream tasks. Note that we evaluate *instance* performance (e.g., one v. all classification, detection, etc). This differs from many previous works that focus on generating synthetic data for closed-set k-way classification (Shin et al., 2023; He et al., 2022b).

3.2 PERSONALIZED SYNTHETIC DATA GENERATION

We generate personalized data from $\mathcal{D}_R$ using Stable Diffusion 1.5, a T2I model, as our generator g_θ. We adapt g_θ using DreamBooth (Ruiz et al., 2022) to generate novel images of c when conditioned on an identifier token.

A T2I diffusion model g_θ generates images given an initial noise latent $\epsilon \sim \mathcal{N}(0, 1)$ and a conditioning text embedding $\hat{y} = \Gamma_\omega(y)$ where Γ_ω is a text encoder, and y is a user-provided prompt. Given a ground-truth image x and the text embedding $\hat{c_{pr}}$ of the generic semantic category c_{pr}, DreamBooth fine-tunes g_θ using the loss:

$$\mathbb{E}_{x,\hat{y},\epsilon,\epsilon',t}[w_t||g_\theta(\alpha_t x + \sigma_t \epsilon, \hat{y}) - x||_2^2] \\ +\lambda w_{t'}||g_\theta(\alpha_{t'} x_{pr} + \sigma_{t'} \epsilon', \hat{c_{pr}}) - x_{pr}||_2^2],$$

where x_{pr} is an image synthesized with the pre-trained generator conditioned on $\hat{c_{pr}}$, t is the timestep, and variables α_t, σ_t, and w_t relate to the noise schedule and sampling quality. The first loss term is a reconstruction loss on x, and the second term is a prior preservation loss on x_{pr}. The two loss terms are weighted by λ. Following standard implementations, we also fine-tune Γ_ω with the same loss. For further details, refer to (Ruiz et al., 2022).

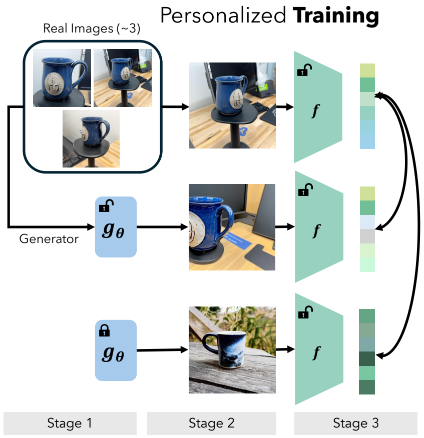

Figure 2: **Personalized Representation Training Pipeline.** Our three-stage training method: 1) Generative Model Training 2) Synthetic Data Generation 3) Contrastive LoRA Fine-Tuning.

While there are several alternative methods for personalized generation (Gal et al., 2023; Alaluf et al., 2023), we focus on DreamBooth, which has been shown to maintain highest fidelity to fine details (Alaluf et al., 2023).

3.3 CONTROLLING GENERATED DATASET ATTRIBUTES

Prior work has observed that fidelity to the target subject and diversity of generated data are both important factors (Sariyildiz et al., 2023). T2I models offer several mechanisms of injecting diversity into generated outputs, allowing us to explore the relationship between these attributes and the quality of learned personalized representations.

Classifier-Free Guidance (CFG). A common way of injecting diversity for diffusion models is modifying the CFG (Ho & Salimans, 2022) at inference, which controls how strongly the generation adheres to the conditioning prompt. We experiment with CFG $\in \{4.0, 5.0, 7.5\}$.

LLM-generated captions. As seen in (He et al., 2022b) and (Dunlap et al., 2023), off-the-shelf Large Language Models such as T5 (Raffel et al., 2023) can be leveraged to generate text-prompts for each object. Following prior works, we generate image captions with GPT-4 (OpenAI, 2023), ensuring that they introduce rich context descriptions in addition to describing the target object. For example, if the object is a shirt, an LLM-generated prompt could be "`a shirt on a coat hook`", or "`a person wearing a shirt at a street market`". For further details, refer to the Appendix.

3.4 REPRESENTATION LEARNING FROM SYNTHETIC DATA.

Given $(\mathcal{D}_R, \mathcal{D}_S)$ of instance c, we personalize f_ϕ via fine-tuning. Critical to representation learning is having both positive and negative examples. We obtain positives from $\mathcal{D}_S$. We generate negatives $\tilde{\mathcal{D}_S}$ by prompting the pretrained g_θ (Stable Diffusion 1.5) with the generic object category: "`a photo of` c_{pr}".

Given $(x, x_+, x_0, ..., x_N)$ where $x \in \mathcal{D}_\mathcal{R}$, $x_+ \in \mathcal{D}_\mathcal{S}$, $x_i \in \tilde{\mathcal{D}_\mathcal{S}}$ for $i = 0, \ldots, N$, we extract f_ϕ features as a concatenation of the `CLS` token and average-pooled final-layer patch-embeddings. We then finetune f_ϕ using the infoNCE loss,

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\text{sim}(\mathbf{x}_0, \mathbf{x}_+)/\tau)}{\sum_{i=1}^{N} \exp(\text{sim}(\mathbf{x}_0, \mathbf{x}_i)/\tau)}.$$

This loss pushes together the representations of real and synthetic images of c, and pushes apart representations of c and other objects. We also experiment with alternate contrastive/non-contrastive losses in the Appendix. We fine-tune via Low-Rank Adaptation (LoRA), which is more parameter-efficient than full fine-tuning (Hu et al., 2021).

3.5 ALTERNATIVES TO DREAMBOOTH

Real data baseline. A simple baseline is to contrastively fine-tune using only the available real data $\mathcal{D}_R$ as positives. Here, we still use a large pool of negatives.

Comparisons enabled by extra resources. With so few real images, there may be benefit in expending "upfront effort" to collect further labels and data. A user might annotate their images, download internet-available data, or even capture more images of the target object. We aim to understand if, in such settings with extra annotations and real data, there are still benefits to using generated data versus computationally cheaper alternatives.

Segmentation masks: First, we consider collecting segmentation masks of $\mathcal{D}_R$. This enables a simple, cheap generative model: *Cut-and-Paste*. Here the generator samples independently from foregrounds containing the target object (carved from $\mathcal{D}_R$) and generic backgrounds from a T2I model. Details in A.3.3.

With masks, we can also improve DreamBooth generations by enabling *masked DreamBooth training* and *filtering*. Fine-tuning g_θ can be affected by signals such as shared backgrounds. To minimize such overfitting, we mask out the gradients for background pixels during DreamBooth training, as in (Zhang et al., 2023). This enables more diverse generations with better prompt adherence. We also use masks to filter generated datasets. Using a perceptual metric (Fu et al., 2023) and perSAM Zhang et al. (2023) we predict a mask for the generated image and measure the similarity to masked training images, filtering out those below a threshold. Details in A.3.2.

Internet-available real data: In a second case, a user may download open-source real datasets; these can provide a source of *real negatives* and *real backgrounds* for Cut/Paste, avoiding the computational cost of image generation and enabling comparison to real-only approaches.

Extra real positives: Finally, a user may expand $\mathcal{D}_R$ by physically collecting extra real target images. This also provides an expanded set of images for Cut/Paste and DreamBooth generation.

4 EXPERIMENTS

4.1 DATASETS

Evaluating our personalized representations necessitates instance-level datasets with multiple tasks, across various real-world scenarios. To satisfy this criteria, we reformulate two existing datasets – DeepFashion2 (Ge et al., 2019) (focused on shirts) and DogFaceNet (Mougeot et al., 2019) (focused on dogs) – and introduce a new dataset, PODS (Personal Object Discrimination Suite). PODS features common personal and household objects, enabling instance-level evaluation across classification, retrieval, detection, and segmentation tasks. To assess robustness and generalization, DF2 and Dogs provide in-the-wild test images, and DF2 and PODS include test sets designed with distribution shifts. All datasets are split such that for each object there are exactly 3 training images and at least 3 test images. We summarize our datasets and procedures below; for additional details and qualitative examples of the datasets, refer to the Appendix.

DeepFashion2 (DF2) is a large-scale fashion dataset with 873K Commercial-Consumer clothes pairs for instance-level retrieval, detection, and segmentation. We use the Consumer-to-Shop Clothes retrieval benchmark, which matches gallery images of clothing items to in-the-wild consumer images, thus encoding a train-test distribution shift. Out of 13 clothing categories, we select

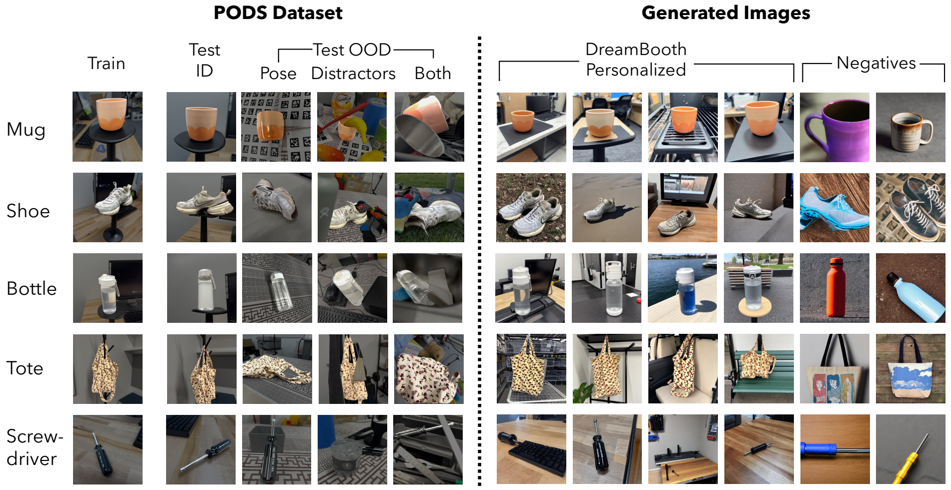

Figure 3: **(left) Examples of instances from our new PODS dataset.** We showcase one example instance from each of the five object categories, displaying images from both the training and various test splits. We dim the surrounding scene, highlighting the instance of interest. This masking technique is not applied to our dataset images or during training. **(right) We show example generated images from Dreambooth** (LLM, cfg 5), which we use as positives in our representation learning finetuning.

the *shirts* category as our focus. We subselect a set of 169 shirts, after filtering out categories which lack sufficient numbers of gallery images.

DogFaceNet (Dogs) is a dog identification dataset, containing 8600 images of 209 dogs. DogFaceNet includes multiple unique dogs of the same breed, making the dataset more challenging. We subselect 80 dogs with sufficient numbers of images, and split the images into a train and test set. To support evaluation of segmentation and detection, we manually annotate the dataset with masks.

Our new dataset: PODS contains 100 unique objects across 5 every-day categories (mugs, screwdrivers, shoes, bags, waterbottles). Each object is captured in four scenes with varying conditions and vantage points. The train set contains 3 images of each object, displayed in a canonical pose with full visibility of key identifying features such as logos. The test set contains 80-100 images of each object, captured in four scenes: one in-distribution (ID) and three out-of-distribution (OOD). We show examples of each type of scene in Figure 3. The ID scene is taken in the same conditions as the training images. OOD scenes include one scene with *pose variation*, one scene with *distractor objects*, and one scene with *both variations*. All OOD scenes are against differing backgrounds from the ID scenes.

The dataset supports evaluation across 4 tasks: classification, retrieval, detection, segmentation. Each test image is associated with the target instance label. From each test scene, 3 randomly-selected images are additionally annotated with the bounding box and segmentation mask of the displayed object. Masks are manually annotated using TORAS (Kar et al., 2021) and SAM (Kirillov et al., 2023b); detection bounding boxes are extracted from the masks. We expect the PODS dataset to be a meaningful benchmark for personalized representation and instance-level detection research, and a valuable resource for the personalized generation community.

4.2 TRAINING

We fine-tune a vision backbone f_ϕ on sets of $(x, x_+, x_0, ...x_N)$ where the anchor x is drawn from the 3 real positive images, the positive x_+ is drawn from the pool of synthetic positives, and x_i for $i = 0, \ldots, N$ are drawn from the synthetic negatives. We apply the following data augmentations to all images: random rotations, horizontal flips, and resized crops. We experiment with several state-of-the-art backbones: DINOv2-ViT B/14 (Oquab et al., 2023), CLIP-ViT B/16, (Radford et al., 2021), and MAE-ViT B/16 (He et al., 2022a).

Each dataset is randomly divided class-wise into a validation set (30 classes), and test set (size depending on the dataset). Using the validation set we sweep over key training parameters: # synthetic

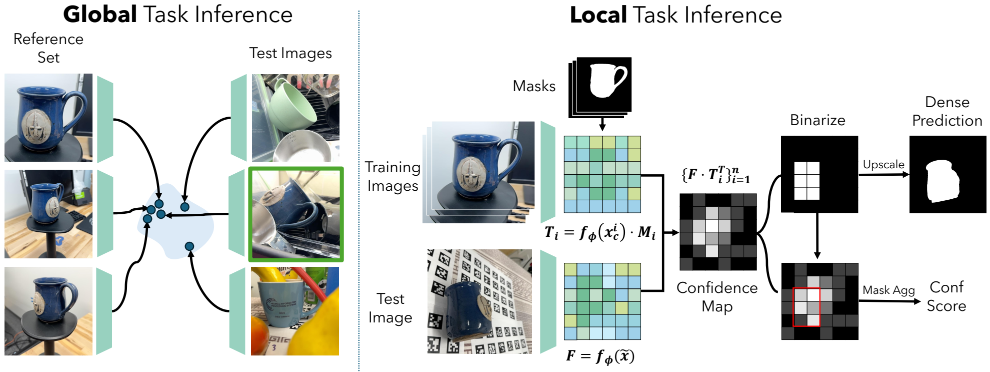

Figure 4: **Inference Pipelines**. We visualize the global (classification, retrieval) and local (detection, segmentation) evaluation pipelines. Global inference uses cosine similarity between CLS embeddings, while local inference extracts patch features with spatial information.

positives, # anchor-positive pairs, and the choice of loss function. Based on our validation experiments, we LoRA finetune with the infoNCE loss for 2 epochs over 4500 anchor-positive pairs, drawn from 450 synthetic positives and 1000 synthetic negatives. For validation results and further training details, refer to the Appendix.

4.3 EVALUATION

We evaluate personalized representations across one v. all tasks that require *global understanding*, and *the ability to localize* with respect to the target object. Due to the few-shot nature of our task, we evaluate representations directly, without training task-specific heads. We summarize our evaluations below and in Figure 4.

Classification. For a particular instance c, given a test image $\tilde{x}$, a frozen encoder f_ϕ, and training images $x_i^c \in \mathcal{D}_R$ we compute the maximum cosine similarity between the `CLS` tokens of $f_\phi(\tilde{x})$ and $f_\phi(x_c^i)$; this is taken as the prediction confidence. Samples with confidence above some threshold t are taken as positives. Thus we report the Area under the Precision-Recall Curve (PR-AUC), which is a threshold-free metric.

Retrieval. We use our test set as the "query" set, and $\mathcal{D}_R$ as the "retrieval" set. We compute the cosine similarity between the `CLS` token of $f_\phi(\tilde{x})$ and those of the images in $\mathcal{D}_R$. We score the resultant ranking with the NDCG metric (Jeunen et al., 2024).

Segmentation. We compute the average cosine similarity between the patch embeddings of $f_\phi(\tilde{x})$ and those of $f_\phi(x_c^i)$ to generate a local confidence map, where x_c^i are masked to the target, following the procedure of (Zhang et al., 2023). We then apply binarization directly to the confidence map using Otsu's thresholding method (Otsu et al., 1975) and upscale to the image dimensions to yield a segmentation prediction. We report the standard mask AP metric (Deng et al., 2024) and F1 scores, given the high imbalance between positives and negatives in the test sets.

Detection. We apply the same procedure as segmentation, and extract a bounding box prediction by drawing a box around the predicted mask. We obtain a confidence score for the box by averaging over the confidence map within the box region. We report the standard AP metric and the F1 score.

For each task, we compare the performance of our learned personalized representations to pretrained models. Note that we do not train prediction heads, due to the lack of real training data in our setting – rather, we use these evaluations to probe what our personalized features learn about the target object, compared to pretrained features.

5 RESULTS AND DISCUSSION

5.1 PERSONALIZED REPRESENTATIONS IMPROVE OVER PRETRAINED REPRESENTATIONS

We LoRA-tune three backbones (DINOv2, CLIP, MAE) and evaluate the personalized representations on four tasks. We sweep over synthetic datasets with different levels of diversity by varying the CFG and usage of LLM-generated prompts as described in Section 3.3, and select the best for each

	Classification			Retrieval			Detection			Segmentation		
	PODS	DF2	Dogs	PODS	DF2	Dogs	PODS	DF2	Dogs	PODS	DF2	Dogs
DINOv2	31.0	14.4	83.1	71.7	36.3	89.4	13.0	4.4	12.0	11.8	5.1	11.4
DINOv2-P (DB)	42.9 ↑	34.9 ↑	81.9 ↓	76.8 ↑	64.4 ↑	94.9 ↑	13.7 ↑	8.2 ↑	16.5 ↑	13.0 ↑	8.7 ↑	16.8 ↑
CLIP	28.7	12.7	36.5	64.4	34.7	58.0	0.2	2.8	7.4	0.1	3.0	6.7
CLIP-P (DB)	44.2 ↑	25.6 ↑	65.4 ↑	72.2 ↑	51.3 ↑	80.4 ↑	1.6 ↑	4.0 ↑	10.6 ↑	0.6 ↑	4.4 ↑	10.2 ↑
MAE	8.4	5.2	11.3	35.0	25.8	33.6	0.0	0.2	0.1	0.0	1.1	1.1
MAE-P (DB)	15.8 ↑	12.1 ↑	30.3 ↑	30.3 ↓	23.2 ↓	42.9 ↑	0.1 ↑	0.3 ↑	0.2↑	0.1 ↑	1.2 ↑	1.2 ↑

Table 1: **Performance of personalized v. pretrained representations across backbones, tasks, and datasets.** We compare personalized and pretrained backbones with access to only 3 real images, and no extra data/annotations. For each backbone we report results for the best-performing synthetic dataset (chosen using the validation set), averaged over 4 seeds. Personalized representations (-P) largely outperform pretrained representations across all tasks. Full results across synthetic datasets, and error over seeds, are in the Appendix.

Method	Real Backgrounds	Real Negs	Classification			Retrieval			Detection			Segmentation		
			PODS	DF2	Dogs	PODS	DF2	Dogs	PODS	DF2	Dogs	PODS	DF2	Dogs
Real Imgs	-	✗	35.9	27.6	82.2	65.6	50.8	92.9	12.5	6.4	14.5	11.6	7.2	14.3
	-	✓	33.7	27.6	82.1	61.6	50.7	92.8	13.1	6.5	15.0	11.5	7.6	14.9
Cut/Paste	✗	✗	57.1	48.2	84.3	83.1	68.4	93.9	17.5	11.0	15.5	14.8	12.5	15.3
	✓	✓	58.8	46.3	88.0	78.2	65.5	95.7	19.5	10.3	14.2	15.9	11.5	13.4
Masked DB	-	✗	55.8	47.1	83.8	82.2	69.2	94.3	18.5	11.0	16.5	**16.6**	12.6	16.5
	-	✓	55.1	43.1	84.9	76.5	68.1	94.2	19.1	10.7	14.1	15.8	12.6	13.3
Combined	✗	✗	59.4	**51.1**	85.3	**85.5**	**71.8**	95.0	18.4	**12.2**	**17.3**	15.3	**14.0**	**17.6**
	✓	✓	**61.5**	49.3	**88.7**	81.0	70.5	**96.2**	**21.1**	12.1	15.1	17.6	13.9	14.3

Table 2: **Comparisons across data augmentation methods.** We compare DINOv2-P trained with different augmentation strategies, including those requiring extra annotations/data. Training with synthetic data improves performance significantly over training with the limited real-image dataset; combined Masked DreamBooth + Cut/Paste performs best in all cases. Significant boosts are also achievable more cheaply when incorporating internet-available real data with Cut/Paste.

backbone based on validation performance. In Table 1 we compare the performance of pretrained and personalized backbones on DF2, Dogs, and PODS, using the best synthetic dataset for each backbone (full results in the Appendix).

Personalized models (-P in Table 1) boost performance in 33/36 cases. We observe improvements – often substantial – in nearly every combination of backbone and task besides MAE retrieval. For example, averaged across datasets, DINOv2 detection improves by 44%, DINOv2 retrieval by 30%, and CLIP classification by 88% relative to pretrained models. Across all three datasets, personalized models boost performance both for global tasks requiring semantic understanding, and dense tasks requiring localization. We also visualize dense prediction maps for multi-object images in Figure 5 and the Appendix, showing that personalized patch features better localize the target object.

5.2 WHAT MAKES FOR THE BEST TRAINING DATA?

In the previous section, we show that personalizing representations significantly boost personalized task performance. Here, we compare data generation approaches, some leveraging additional resources, to investigate tradeoffs between computational cost and performance.

We examine incorporating segmentation masks of $\mathcal{D}_R$ (see Section 3.4). This enables a cheap baseline method, Cut-and-Paste (CP), and improvements to DreamBooth via masked training and filtering (Masked DB). We also test sampling from a combined pool of CP and Masked DB images (Combined). Our CP, Masked DB, and Combined pools are each 450 images for fair comparison. We also ablate the use of generated negatives and generated CP backgrounds by using open-source images as real alternatives; this enables comparison to cheap methods that use only real data. Our results are shown in Table 2.

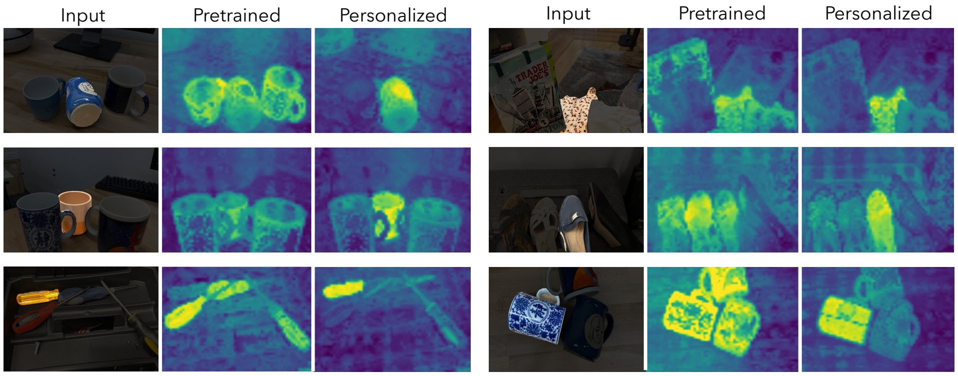

Figure 5: **Qualitative Results.** Each triplet shows the test image (left), dense prediction maps for pretrained DINOv2 (center), and personalized (right). Prediction maps are computed via patchwise embedding similarity between the test and localized train images following Figure 4. Personalized representations distinctly localize the target instance, unlike pretrained embeddings. For visualization only, the personalized instance is highlighted in the test images but this is not applied during training or inference.

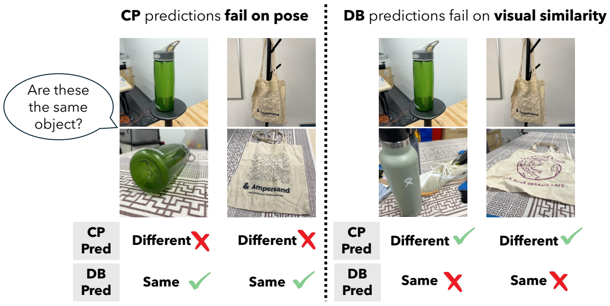

Figure 7: **DreamBooth vs Cut and Paste Model Failures.** We show object pairs where DB-personalized and CP-personalized models differ most in predictions.

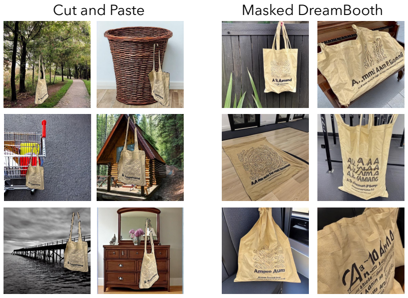

Figure 8: **(left)** CP limitations include pose and realism. **(right)** Masked DB struggles with fine-grained details.

Performance improves significantly from incorporating masks; results in Table 2 out-perform DINOv2-P results in Table 1, obtained without masks. However, incorporating real negatives and real Cut/Paste backgrounds does not consistently improve performance over their generative counterparts. Moreover, the best synthetic data methods outperform all real-image-only methods, with the Combined pool performing best across all datasets/tasks. Comparing Tables 2 and 3 also show some tradeoff between efficiency and performance. The high performance of the Combined pool indicates that learned models provide valuable knowledge for personalized data augmentation. However, sampling CP with real backgrounds performs similarly to Masked DB alone, showing that strong performance can still be achieved with an efficient alternative.

Scaling real positives. We manually capture additional training images for 25 PODS instances (5 per category) with different backgrounds, poses, and lighting. We compare training DINOv2-P on $\mathcal{D}_R$, and on synthetic $\mathcal{D}_S$ generated from $\mathcal{D}_R$ at sizes $|\mathcal{D}_R| = 3$ and $|\mathcal{D}_R| = 20$ using the Combined method. Performance increases as $|\mathcal{D}_R|$ increases, saturating at $|\mathcal{D}_R| = 15$, likely due to limitations in the diversity of the additional real data. Expanding diversity further could improve scaling but requires significant manual effort. Synthetic augmentation remains effective as $|\mathcal{D}_R|$ scales (27% gain with 3 real images, 8% gain with 20). As generative models improve, the ability to complement real datasets is expected to grow.

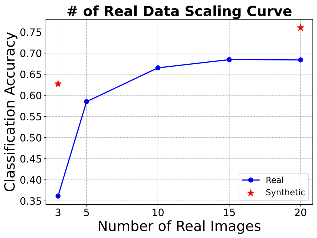

Figure 6: **Real and synthetic data scaling curve for a subset of PODS.**

5.3 How do different datasets affect representations?

As seen in Table 2, Masked DB and CP achieve similar performance. Here, we show that they exhibit distinct strengths and limitations. We analyze divergence cases in high-confidence predictions of DB- and CP-trained representations, revealing consistent failure patterns. DB-trained models excel at pose generalization but often confuse visually similar instances. Conversely, CP models are more robust to distractors but falter when encountering unfamiliar poses. We show examples in Figure 6. This trend is also quantitatively shown (Figure 9). In the PODS Distractors split, CP models outperform DB models by 7%, whereas in the Pose split, DB models surpass CP models by 6.4%.

In Figure 6 we trace these attributes of learned representations to biases/limitations in DB and CP generations. The main DB limitation is difficulty in preserving fine-grained object details, resulting in images that only loosely resemble the target characteristics, even with filtering. These inaccuracies likely propagate to the learned representation, compromising its fine-grained discriminative ability. Conversely, CP maintains perfect object fidelity but restricts pose variability, potentially leading to pose overfitting.

We also study how fidelity and diversity affect performance in Figure 12. Without filtering, DB datasets lie at extremes (high-diversity, low-fidelity or vice versa). Both the Masked DB and CP datasets, achieve a better balance, and thus higher performance.

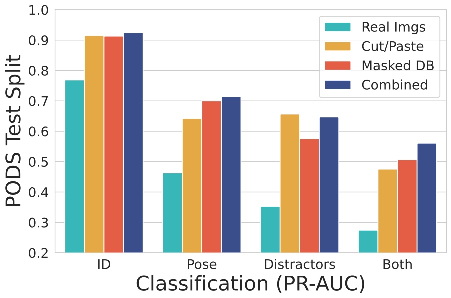

Figure 9: **DB, CP, and Combined performance on PODS test splits.** DB outperforms CP on the Pose split, and underperforms it on the Distractors split, whereas Combined performs best on both. These results support our qualitative observations.

5.4 Applications

Our thresholding evaluation for dense tasks allows direct probing of personalized patch features, however does not achieve state-of-the-art results. We show that our personalized representations can be easily integrated into perSAM – a practical existing pipeline – to improve its performance (Zhang et al., 2023). Instead of using a pretrained backbone to extract keypoint proposals and generate confidence scores, we use our personalized backbone. Segmentation performance improves from 5.6% F1 score (DINOv2) to 10.9 % (DINOv2-P) on DF2, 19.8% to 24.4% on Dogs, and 21.6% to 25.3% on PODS. Full results in the Appendix.

6 Limitations and Conclusions

6.1 Limitations

Our pipeline has a potential for high computational cost, as seen in Table 3, particularly due to fine-tuning and T2I generation stages. We defer this limitation to future research in generative model efficiency. Furthermore, by training on data generated by T2I models we may inherit their biases and limitations, as shown in 5.1. When employing them for data generation, we must be mindful of the relevant ethical concerns around deployment and equitable use.

6.2 Conclusions

We leverage generative models to adapt general-purpose representation spaces to personalized ones, using *very few* real examples of a *single* instance. We quantitatively and qualitatively show that personalized representations consistently boost performance across downstream tasks. Moreover, we also study computationally cheaper alternatives that leverage additional resources, and show that combining different generation methods may enable further improvements. We release our new dataset, PODS, and new splits and annotations for two existing instance-level datasets, offering comprehensive benchmarks for future work.

In the future, generative models will continue to get faster, cheaper, more accurate, controllable and potentially less biased. Our work is not limited to existing generation techniques. We are excited by the potential of learning personalized representations in this way, and envision a possibility that allows users to have ownership over their own models and data.

CODE OF ETHICS

The authors have read the code of ethics and we acknowledge that we adhere to the code presented.

ETHICS STATEMENT

Research into personalized representations has the potential for both positive and negative societal impacts, which we discuss below.

One potential application of such an approach, and more broadly of all instance-level recognition and re-identification work, is surveillance technology. Surveillance has the potential to infringe upon human right to privacy depending on it's use and intent. In this work we choose focus on non-human instance recognition applications and datasets, as opposed to human facial re-identification or vehicle re-identification applications.

Personalized representation learning also has potential positive impacts on human privacy. By removing the need for access to real images from other instances, we can build methods for people to have personalized AI models without having to share their data to a central repository or to have access to other's data during personalization, beyond what is already contained in a pretrained T2I model. This could enable users to choose to keep personal data siloed, similar to motivation for federated learning settings, while still benefiting from personalized training. However, it does not overcome current challenges around lack of informed consent for personal data to be used during the training of large-scale T2I models in the first place, alongside potential copyright infringement for created content, which is of increasing concern and debate within our community Li et al. (2024); Duan et al. (2023).

Additionally, this has the potential to make progress on the democratization of representation learning by give more power to a user. By removing the requirement to have access to external data, we reduce data storage and access resources to benefit from such a system. Because as few as three positive examples are all that is needed for a user to hand-select, this empowers users to develop personalized representations for instances of their own interest simply and with minimal effort. However, as discussed in the limitations section 6.1, the current computational overhead of our proposed method makes it infeasible for most people to easily make use of. This points to the potential for future work in increasing the efficiency of such methods, and will also benefit from progress in efficient deep learning hardware.

The current computational cost of the method has an additional negative dimension, which is the power and water usage by the GPUs needed to train and run inference with our current method. AI has become an increasingly large portion of global power use, water use, and carbon emmissions Luccioni et al. (2023). Inefficient methods exacerbate this effect.

REPRODUCIBILITY

We have uploaded an anonymized zip file containing the source code for our work, including the necessary metadata to reproduce our results, such as the LLM-generated captions used for dataset creation. The codebase features an end-to-end pipeline for data generation, training personalized models, and conducting inference and evaluation: https://drive.google.com/file/d/1eZpNe00YL4FoOG2RGSmdKtlZquGlsidD/view?usp=sharing. In the appendix, we provide detailed hyperparameters and outlines for data generation (DreamBooth finetuning, Cut and Paste) and personalization training (parameters for LoRA and other model training hyperparameters), Additionally, we include a reproducible description of our evaluation pipeline. We also present extensive ablations and full results in the appendix for transparency, justifying each design choice so that other researchers can replicate our findings. We also intend to release our dataset, PODS, and the reformulated DeepFashion2 and DogFaceNet datasets.

REFERENCES

Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *CVPR*, 2021.

Yuval Alaluf, Elad Richardson, Gal Metzer, and Daniel Cohen-Or. A neural space-time representation for text-to-image personalization. *TOG*, 2023.

Yuval Alaluf, Elad Richardson, Sergey Tulyakov, Kfir Aberman, and Daniel Cohen-Or. Myvlm: Personalizing vlms for user-specific queries. *arXiv preprint arXiv:2403.14599*, 2024.

Ali Amiri, Aydin Kaya, and Ali Seydi Keceli. A comprehensive survey on deep-learning-based vehicle re-identification: Models, data sets and challenges. *arXiv preprint arXiv:2401.10643*, 2024.

Moab Arar, Rinon Gal, Yuval Atzmon, Gal Chechik, Daniel Cohen-Or, Ariel Shamir, and Amit H. Bermano. Domain-agnostic tuning-encoder for fast personalization of text-to-image models. In *SIGGRAPH Asia*, 2023.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo. Zero-shot composed image retrieval with textual inversion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15338–15347, 2023.

Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. *arXiv preprint arXiv:2302.02503*, 2023.

David Bau, Hendrik Strobelt, William S. Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *TOG*, 2019.

Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6593–6602, 2024.

Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. Attend to you: Personalized image captioning with context sequence memory networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 895–903, 2017.

Niv Cohen, Rinon Gal, Eli A Meirom, Gal Chechik, and Yuval Atzmon. “this is my unicorn, fluffy”: Personalizing frozen vision-language representations. In *ECCV*, 2022.

Xueqing Deng, Qihang Yu, Peng Wang, Xiaohui Shen, and Liang-Chieh Chen. Coconut: Modernizing coco segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21863–21873, 2024.

Tan M Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *CVPR*, 2022.

Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In *International Conference on Machine Learning*, pp. 8717–8730. PMLR, 2023.

Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E. Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation, 2023.

Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. *arXiv preprint arXiv:2312.04567*, 2023.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *NeurIPS*, 2023.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023.

Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo. A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. *CVPR*, 2019.

Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008.

Shanyan Guan, Yanhao Ge, Ying Tai, Jian Yang, Wei Li, and Mingyu You. Hybridbooth: Hybrid prompt inversion for efficient subject-driven generation.

Hasan Abed Al Kader Hammoud, Hani Itani, Fabio Pizzati, Philip Torr, Adel Bibi, and Bernard Ghanem. Synthclip: Are we ready for a fully synthetic clip training? *arXiv preprint arXiv:2402.01832*, 2024.

Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris N. Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. In *ICCV*, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022a.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022b.

Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. In *ICCV*, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Qihan Huang, Siming Fu, Jinlong Liu, Hao Jiang, Yipeng Yu, and Jie Song. Resolving multi-condition confusion for finetuning-free personalized image generation. *arXiv preprint arXiv:2409.17920*, 2024.

Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *arXiv preprint arXiv:2106.05258*, 2021.

Olivier Jeunen, Ivan Potapov, and Aleksei Ustimenko. On (normalised) discounted cumulative gain as an off-policy evaluation metric for top-n recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1222–1233, 2024.

Kyuheon Jung, Yongdeuk Seo, Seongwoo Cho, Jaeyoung Kim, Hyun-seok Min, and Sungchul Choi. Dalda: Data augmentation leveraging diffusion model and llm with adaptive guidance scaling. *arXiv preprint arXiv:2409.16949*, 2024.

Amlan Kar, Seung Wook Kim, Marko Boben, Jun Gao, Tianxing Li, Huan Ling, Zian Wang, and Sanja Fidler. Toronto annotation suite. `https://aidemos.cs.toronto.edu/toras`, 2021.

Shyamgopal Karthik, Karsten Roth, Massimiliano Mancini, and Zeynep Akata. Vision-by-language for training-free compositional image retrieval. *arXiv preprint arXiv:2310.09291*, 2023.

Shahroz Khan, Erkan Gunpinar, Masaki Moriguchi, and Hiromasa Suzuki. Evolving a psychophysical distance metric for generative design exploration of diverse shapes. *Journal of Mechanical Design*, 2019.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *ICCV*, 2023a.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023b.

Nupur Kumari, Bin Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *CVPR*, 2022.

Orest Kupyn and Christian Rupprecht. Dataset enhancement with instance-level augmentations. *arXiv preprint arXiv:2406.08249*, 2024.

Shiye Lei, Hao Chen, Sen Zhang, Bo Zhao, and Dacheng Tao. Image captions are natural prompts for text-to-image models. *arXiv preprint arXiv:2307.08526*, 2023.

Zhangheng Li, Junyuan Hong, Bo Li, and Zhangyang Wang. Shake to leak: Fine-tuning diffusion models can amplify the generative privacy risk. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 18–32. IEEE, 2024.

Shaobo Lin, Kun Wang, Xingyu Zeng, and Rui Zhao. Explore the power of synthetic data on few-shot object detection. In *CVPR*, 2023.

Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *CVPR*, 2016.

Alexandra Sasha Luccioni, Yacine Jernite, and Emma Strubell. Power hungry processing: Watts driving the cost of ai deployment? *arXiv preprint arXiv:2311.16863*, 2023.

Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–12, 2024.

Guillaume Mougeot, Dewei Li, and Shuai Jia. A deep learning approach for dog face verification and recognition. In *PRICAI 2019: Trends in Artificial Intelligence*, 2019.

Thao Nguyen, Haotian Liu, Yuheng Li, Mu Cai, Utkarsh Ojha, and Yong Jae Lee. Yo'llava: Your personalized language and vision assistant. *arXiv preprint arXiv:2406.09400*, 2024a.

Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Yinlin Hu, Renaud Marlet, Mathieu Salzmann, and Vincent Lepetit. Nope: Novel object pose estimation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17923–17932, 2024b.

Yotam Nitzan, Kfir Aberman, Qiurui He, Orly Liba, Michal Yarom, Yossi Gandelsman, Inbar Mosseri, Yael Pritch, and Daniel Cohen-Or. Mystyle: A personalized generative prior. *TOG*, 2022.

Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 2002.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023.

Nobuyuki Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11 (285-296):23–27, 1975.

Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. Towards personalized image captioning via multimodal memory networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):999–1012, 2018.

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 *ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *TOG*, 2021.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. *ArXiv*, 2023.

Kuniaki Saito, Kihyuk Sohn, Xiang Zhang, Chun-Liang Li, Chen-Yu Lee, Kate Saenko, and Tomas Pfister. Pic2word: Mapping pictures to words for zero-shot composed image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19305–19314, 2023.

Joseph W. Sakshaug and Trivellore E. Raghunathan. Synthetic data for small area estimation. In *Privacy in Statistical Databases*, 2010.

Mert Bulent Sariyildiz, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *CVPR*, 2023.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8543–8552, 2024.

Joonghyuk Shin, Minguk Kang, and Jaesik Park. Fill-up: Balancing long-tailed data with generative models. *arXiv preprint arXiv:2306.07200*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Samuel Stevens, Jiaman Wu, Matthew J Thompson, Elizabeth G. Campolongo, Chan Hee Song, David Carlyn, Li Dong, Wasila M. Dahdul, Charles Stewart, Tanya Y. Berger-Wolf, Wei-Lun Chao, and Yu Su. Bioclip: A vision foundation model for the tree of life. *arXiv preprint arXiv:2311.18803*, 2023.

Carlos C Sun, Glenn S Arr, Ravi Prakash Ramachandran, and Stephen G Ritchie. Vehicle reidentification using multidetector fusion. *IEEE TITS*, 2004.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

Fabio Henrique Kiyoiti Dos Santos Tanaka and Claus Aranha. Data augmentation using gans. *arXiv preprint arXiv:1904.09135*, 2019.

Yonglong Tian, Lijie Fan, Kaifeng Chen, Dina Katabi, Dilip Krishnan, and Phillip Isola. Learning vision from models rivals learning vision from data. *arXiv preprint arXiv:2312.17742*, 2023a.

Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *arXiv preprint arXiv:2306.00984*, 2023b.

Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023.

Allan Tucker, Zhenchen Wang, Ylenia Rotalinti, and Puja Myles. Generating high-fidelity synthetic patient data for assessing machine learning healthcare software. *npj Digital Medicine*, 2020.

Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *CVPR*, 1991.

Xuan Wang, Guanhong Wang, Wenhao Chai, Jiayu Zhou, and Gaoang Wang. User-aware prefix-tuning is a good learner for personalized image captioning. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pp. 384–395. Springer, 2023.

Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *ICCV*, 2023.

Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. *arXiv preprint arXiv:2303.11681*, 2023.

Ankit Yadav and Dinesh Kumar Vishwakarma. Deep learning algorithms for person re-identification: sate-of-the-art and research challenges. *Multimedia Tools and Applications*, 83 (8):22005–22054, 2024.

Chun-Hsiao Yeh, Bryan Russell, Josef Sivic, Fabian Caba Heilbron, and Simon Jenni. Meta-personalizing vision-language models to find named instances in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19123–19132, 2023.

Teresa Yeo, Andrei Atanov, Harold Benoit, Aleksandr Alekseev, Ruchira Ray, Pooya Esmaeil Akhoondi, and Amir Zamir. Controlled training data generation with diffusion models. *arXiv preprint arXiv:2403.15309*, 2024.

Jianhao Yuan, Francesco Pinto, Adam Davies, Aarushi Gupta, and Philip Torr. Not just pretty pictures: Text-to-image generators enable interpretable interventions for robust representations. *arXiv preprint arXiv:2212.11237*, 2022.

Jianhao Yuan, Jie Zhang, Shuyang Sun, Philip Torr, and Bo Zhao. Real-fake: Effective training data synthesis through distribution matching. *arXiv preprint arXiv:2310.10402*, 2023.

Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Xianzheng Ma, Hao Dong, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. *arXiv preprint arXiv:2305.03048*, 2023.

Yunhua Zhang, Hazel Doughty, and Cees GM Snoek. Low-resource vision challenges for foundation models. *arXiv preprint arXiv:2401.04716*, 2024.

Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*, 2017.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

Yongchao Zhou, Hshmat Sahak, and Jimmy Ba. Training on thin air: Improve image classification with generated data. *arXiv preprint arXiv:2305.15316*, 2023.

A DATASETS

A.1 DF2 AND DOGS REFORMULATION

In the paper, we reformulate two datasets: DeepFashion2 and DogFaceNet.

DeepFashion2. In this section, we identify the training split that we subselected from DeepFashion2 to enable reproducibility of our results. DeepFashion2 is a large-scale retrieval dataset with a public train-validation-test split. Each split contains its own query/customer and gallery set. We sample our training set from the gallery set of the validation split, and we sample our test set from the consumer set of the validation split. Each image in validation split has a six-digit identifier and an annotation file containing the information. We organize the data into instance categories; we define the training and testing images for an instance as gallery/consumer images depicting the same clothing item of the same style.

Below we provide metadata of our subselected dataset and include in our submission the exact training and test data split.

- **Clothing Item Category:** Short-Sleeve Tops, category id 1
- **Unique Instances Selected:** 169
- **Total # Training Images:** 507
- **# of Training Images per Instance:** 3
- **Total # Test Images:** 2924
- **Range of Test Images per Instance:** [4, 24]

Dogs. Our Dogs dataset reformulates the DogFaceNet_large split from the datasets released with DogFaceNet for dog re-identification studies. Since the original dataset was published with instance-level splits, we perform our own splitting of the dataset to fit our personalized learning setting. Due to the nature of data collection (images of dogs collected from sequential footage), we had to pay careful attention to the possibility of data poisoning between the train and test set. The procedure we followed for data splitting is as follows:

1. Filtered DogFaceNet dog classes to keep classes with above 10 images per instance
2. Performed a random train-test split for every instance, keeping 3 images for train and rest for test
3. Manually inspected every instance in the dataset to remove data poisoning. This entailed looking through the training and test data and making sure that no test images were from the same sequential footage as the train data. When such images were discovered, they were removed from the test set.
4. After data-cleanup, we removed instances with less then 4 remaining test images.

The above procedure resulted in 80 total dog classes. Below is the metadata of our subselected dataset. Similar to DF2, we include in our submission the exact training and test data split.

- **Unique Instances Selected:** 80
- **Total # Training Images:** 240
- **# of Training Images per Instance:** 3
- **Total # Test Images:** 1218
- **Range of Test Images per Instance:** [6, 38]

Dataset Examples. In Fig. 10 and 11 we show examples of training and test images for the Dogs and DF2 datasets. Notably, the test sets cover a wide range of diverse in-the-wild scenarios, including different contexts, positions, backgrounds, camera angles, occlusions, and lightings.

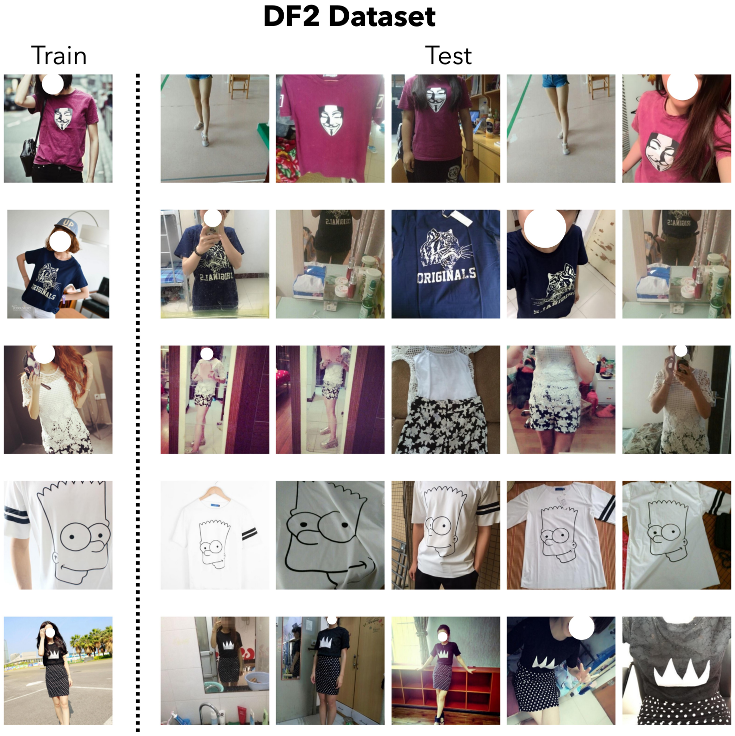

Figure 10: **DF2 train/test examples.** Training images are of models wearing clothes, and test images are from consumers. Images and classes are randomly sampled.

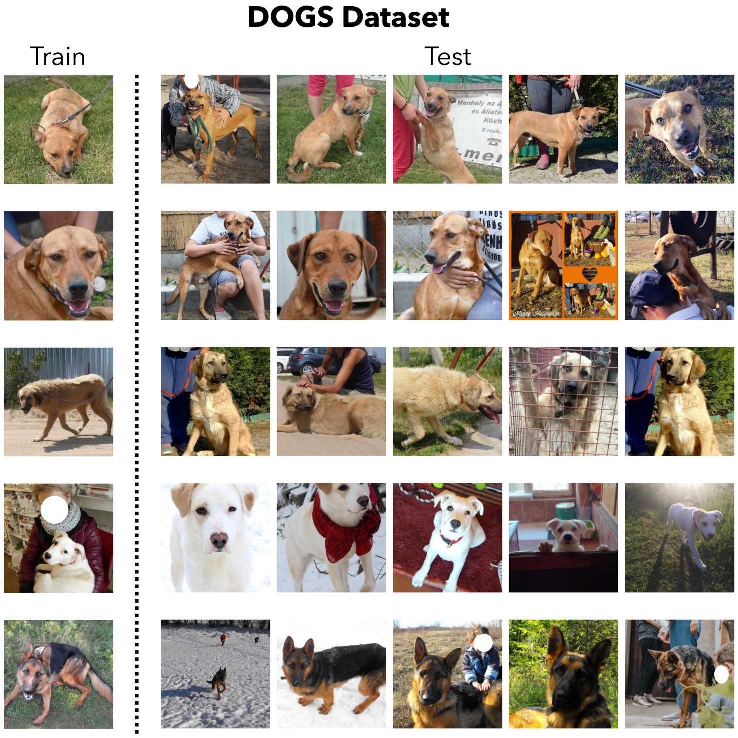

Figure 11: **Dogs train/test images.** Images and classes are randomly sampled.

Licenses for existing assets The datasets we use are released under the following licenses:

DeepFashion2: MIT
DogFaceNet Dataset: Non-Commercial, Research-Only

A.2 PODS

A.2.1 PODS OVERVIEW

- **Unique Instances:** 100
- **Total # Training Images:** 300
- **# of Training Images per Instance:** 3
- **Total # Test Images:** 10991
- **Range of Test Images per Instance:** [72, 202]
- **Total # Test Images with Dense Annotations:** 1200
- **Range of Test Images per Instance with Dense Annotations:** [12, 12]

A.2.2 PODS CREATION

PODS includes images of 100 objects; 20 each from five categories: mugs, screwdrivers, tote bags, shoes, and water bottles. We choose these categories to cover a range of personal, everyday objects, both rigid and deformable.

Scenes. Every object is captured in 4 scenes. We describe each scene through 4 attributes: the *background* of the scne, the *pose* of the target object, the presence of *distractor objects*, and the visibility of the object's *key identifying features* (such as a mug's logo).

Below is a detailed description of each scene.

- **Train/In-distribution:**
    - *Background:* The object is against a plain office background.
    - *Pose*: The object is located on a flat surface, upright in its ”canonical pose”.
    - *Distractors:* No similarly-sized distractor objects are nearby, or in the clear foreground.
    - *Identifying features:* Key identifying features (i.e. logo, handles, etc) clearly visible.
- **Distractors:**
    - *Background:* The object is against different background from the Train scene.
    - *Pose*: Object is upright in its ”canonical” pose.
    - *Distractors:* Object is surrounded by 2-5 distractor objects of varying sizes, located in both the foreground and background. These can act as potential occluders.
    - *Identifying features:* Key identifying features may not be fully visible (for instance, a mug may be occluded by distractors such that the logo is only partially visible.)
- **Pose:**
    - *Background:* The object is against different background from the Train scene.
    - *Pose*: Object is in a different pose from the training scene.
    - *Distractors:* No similarly-sized distractor objects nearby, or in the clear foreground.
    - *Identifying features:* Key identifying features may not be fully visible (for instance, a mug may be turned so that its logo is only partially visible.)
- **Both:**
    - *Background:* The object is against different background from the Train scene.
    - *Pose*: Object is in a different pose from the training scene.
    - *Distractors:* Object is surrounded by 2-5 distractor objects of varying sizes, located in both the foreground and background. These can act as potential occluders.

  - *Identifying features:* Key identifying features may not be fully visible (for instance, a mug may be turned or occluded so that its logo is only partially visible.)
- **Multi-object:**
  - *Background:* The objects are against a different background from the Train. scene.
  - *Distractors:* The 20 objects are randomly split into 4 groups of 5 objects; the objects themselves serve as distractors.
  - *Pose:* Half of the objects in the scene are not in their canonical poses.
  - *Identifying features*: Key identifying features may not be fully visible (for instance, a mug may be turned or occluded so that its logo is only partially visible.)

For the final (hardest) scene, we attempt to capture images that mimic expected in-the-wild settings for each category. For instance, ”Both” scenes for shoes are captured outdoors, with sports equipment as distractors. Similarly mugs are captured on a drying rack, with other wares as distractors.

Capture. We capture all images on an iPhone 15 Pro. For each scene, we use the PolyCam app to capture a video. The app automatically extracts $\tilde{2}0$ frames to be exported. We capture each video in three 120-180 degree views, each at a different vantage point: Level with the object, above the object (camera looking down), below the object (camera looking up). Note that this results in images where the object is not centered, occluded by distractor objects, or out of focus in the background; these are useful as hard positives.

Splits and annotation. For each object, we manually inspect the Training scene and extract three training images, taken at level with the target object, and roughly equally spaced throughout the camera trajectory. The rest of the Training scene images are relegated to the test set, and serve as the in-distribution split. Images from the other three scenes serve as the three out-of-distribution (OOD) splits.

Annotation. We record a unique identifier for each object, and label every test image with the identifier of the object in that image. These image-level labels are used for classification and retrieval.

For each object, we randomly choose 3 images from each test scene to annotate with segmentation masks and bounding boxes. Thus there are 12 images total per object with dense annotations. To annotate with masks, we first use Grounding-SAM Kirillov et al. (2023a) to generate mask proposals. We then manually inspect every image. For images with incorrect generated masks, we manually annotate using TORAS (Kar et al., 2021).

A.3 SYNTHETIC DATA GENERATION

A.3.1 LLM-GENERATED CAPTIONS

We generated 100 instance-relevant prompts for image generation, which we used at inference time for all generated images. Here we present 30 caption examples from the caption that we used for each dataset. We attach the full caption set to our supplement.

PODS Dataset

- Mugs
  1. A `<new1>` mug on a wooden desk
  2. A `<new1>` mug in a cozy living room
  3. A `<new1>` mug on a windowsill
  4. A `<new1>` mug in a breakfast nook
  5. A `<new1>` mug on a bedside table
  6. A `<new1>` mug in a sink full of dishes
- Bottles
  1. A `<new1>` bottle on a picnic table

2. A `<new1>` bottle in a backpack pocket
3. A `<new1>` bottle on a yoga mat
4. A `<new1>` bottle in a car cup holder
5. A `<new1>` bottle on a nightstand
6. A `<new1>` bottle in a bicycle holder

- Screwdrivers
  1. A `<new1>` screwdriver in a toolbox
  2. A `<new1>` screwdriver on a wooden workbench
  3. A `<new1>` screwdriver in a carpenter's tool belt
  4. A `<new1>` screwdriver on a garage shelf
  5. A `<new1>` screwdriver in a utility drawer
  6. A `<new1>` screwdriver on a metal shelf
- Totes (Bags)
  1. A `<new1>` bag in a car trunk
  2. A `<new1>` bag on a park bench
  3. A `<new1>` bag in a shopping cart
  4. A `<new1>` bag on a library shelf
  5. A `<new1>` bag in a gym locker
  6. A `<new1>` bag on a wooden table
- Shoes
  1. A `<new1>` shoe in the rain
  2. A `<new1>` shoe on a sandy beach
  3. A `<new1>` shoe in a gym locker
  4. A `<new1>` shoe on a staircase
  5. A `<new1>` shoe in a laundry basket
  6. A `<new1>` shoe on a wooden floor

DF2 Dataset

1. A person wearing a `<new1>` shirt at a park
2. A `<new1>` shirt on a mannequin
3. A person wearing a `<new1>` shirt at a party
4. A `<new1>` shirt on a clothesline
5. A person wearing a `<new1>` shirt at a concert
6. A `<new1>` shirt on a chair
7. A person wearing a `<new1>` shirt at a café
8. A `<new1>` shirt on a laundry basket
9. A person wearing a `<new1>` shirt at a stadium
10. A `<new1>` shirt on a hook
11. A person wearing a `<new1>` shirt at a bus stop
12. A `<new1>` shirt on a drying rack
13. A person wearing a `<new1>` shirt at a gym
14. A `<new1>` shirt on a shelf
15. A person wearing a `<new1>` shirt at a picnic
16. A `<new1>` shirt on a table
17. A person wearing a `<new1>` shirt at a restaurant
18. A `<new1>` shirt on a suitcase

19. A person wearing a `<new1>` shirt at home
20. A `<new1>` shirt on a couch
21. A person wearing a `<new1>` shirt at a school
22. A `<new1>` shirt on a man's back
23. A person wearing a `<new1>` shirt at a train station
24. A `<new1>` shirt on a floor
25. A person wearing a `<new1>` shirt at a wedding
26. A `<new1>` shirt on a counter
27. A person wearing a `<new1>` shirt at a library
28. A `<new1>` shirt on a washing machine
29. A person wearing a `<new1>` shirt at work
30. A `<new1>` shirt on a clothes rack

Dogs Dataset

1. A `<new1>` dog in the park
2. A `<new1>` dog at the vet
3. A `<new1>` dog in a car
4. A `<new1>` dog at the groomer
5. A `<new1>` dog on a walk
6. A `<new1>` dog in the snow
7. A `<new1>` dog at the lake
8. A `<new1>` dog in the backyard
9. A `<new1>` dog at the `<new1>` dog park
10. A `<new1>` dog in a sweater
11. A `<new1>` dog in a bed
12. A `<new1>` dog at the farm
13. A `<new1>` dog in the woods
14. A `<new1>` dog in a kennel
15. A `<new1>` dog at a barbecue
16. A `<new1>` dog on a hike
17. A `<new1>` dog in a crate
18. A `<new1>` dog at a birthday party
19. A `<new1>` dog in a puddle
20. A `<new1>` dog at the groomer
21. A `<new1>` dog in a costume
22. A `<new1>` dog in a car ride
23. A `<new1>` dog at the beach
24. A `<new1>` dog in the city
25. A `<new1>` dog in a training class
26. A `<new1>` dog in the mountains
27. A `<new1>` dog at the playground
28. A `<new1>` dog in the backyard
29. A `<new1>` dog in a pool
30. A `<new1>` dog at a picnic

A.3.2 MASKED DREAMBOOTH - FILTERING

We apply automatic filtering to the Masked DreamBooth pipeline as an additional data-processing step that we can take when masks are available, to ensure high-quality generated data. We use the masks to extract a bounding box for the object of interest in the training image, and embed it using DreamSim (Fu et al., 2023), a perceptual similarity metric. Similarly, at every test-image prediction, we apply perSAM to generate a test-mask, and embed the masked test image with DreamSim. A cosine similarity is computed between the train masked embedding and the test masked embedding, and an empirically chosen threshold is used to filter out the data below a certain threshold value. For DF2 and PODS, the threshold was 0.6 and for Dogs it was 0.55.

A.3.3 CUT AND PASTE

To generate cut and paste images, we first require a small subset of training images and their corresponding masks, which we use to extract the foreground object. For the background, we generate 600 unique background scenes following the same set of LLM-prompts used to generate the diverse DreamBooth images. To use them for background generation, we removed the `"<new1>"` specification from every prompt: e.g. `"photo of a <new1> at the beach"` becomes `"photo of a beach"` and addressed proposition inconsistencies afterwards. We then randomly resized the masked foreground image to a scale between 0.3 and 1.3 times the original image size, and pasted it onto the background-generated images at a randomly selected location within the image.

A.3.4 RUNTIMES

We report the wall-clock runtimes of synthetic data generation methods, using a single NVIDIA A100 GPU, in Table 3. Per-Image generation times are taken as an average over 50 generations. Per-Dataset times indicate the time to generate a 450-image dataset with a batch size of 1. We do not take into account the time to download open-source datasets (e.g., real backgrounds for Cut/Paste), as this is highly context dependent.

Method	Fine-tuning (min)	Generation per-Image (sec)	Generation per-Dataset (min)
DreamBooth (no filtering)	3.8	0.98	7.35
DreamBooth (w/ filtering)	3.8	1.83	13.7-152.5
Cut/Paste (real BG)	-	0.06	0.45
Cut/Paste (generated BG)	-	1.04	7.8

Table 3: **Runtime for synthetic data generation.** The fastest method is Cut/Paste with real backgrounds, which does not require a T2I model.

B METHODS

B.1 TRAINING

We use the following hyperparameters to LoRA fine-tune each backbone:

- Learning rate: 0.0003
- Batch size: 16
- LoRA rank: 16
- LoRA alpha: 0.5
- LoRA dropout: 0.3

B.2 EVALUATION

Classification: We take all the available test examples for each instance of interest as our test set. We evaluate each trained personalized embedding space in a one-vs-all binary classification setting with respect to the rest of our test data, using a standard few-shot learning setup. Given a test image $\tilde{x}$ we use a frozen vision encoder, f, to obtain embeddings of $\tilde{x}$. We then compute the maximum cosine similarity between $f(\tilde{x})$ and any real image in $\mathcal{D}_R$, and take this as the prediction confidence. Samples with a confidence above some threshold t are taken as positives; our evaluation metric is thus the Area under the Precision-Recall Curve, which is a threshold-free metric. The performance for a dataset is computed as an average over the PR-AUC for each learned embedding space. We compare performance between our learned personalized embedding spaces and non-personalized models (i.e. pretrained DINOv2).

Retrieval: We take text images as a query set and all available reference training images as a retrieval set. Given a test image $\tilde{x}$ we use a frozen vision encoder, f, to obtain embeddings of $\tilde{x}$. We then compute the maximum cosine similarity between $f(\tilde{x})$ and every image in the retrieval set, $\mathcal{D}_R$, and take this as the prediction ranking for the retrieval task. We score the resultant ranking with the standard NDCG metric, as it considers both relevant and position of all retrieved items, unlike F1 and MRR metrics.

Segmentation: To be able to perform localization, we use the encoder's patch embeddings. First, we obtain target local features by computing the masked embedding of a training image. We then compute the cosine similarity between the target local features and the patch embeddings of the target test image to generate a local confidence map, where high confidence regions correspond to localization probability of the object. We then apply binarization directly to the local confidence map with Otsu's thresholding method, and upscale it to the image dimensions to yield a segmentation prediction. We score this prediction by taking the aggregate local confience map values in the predicted mask region. We evaluate our segmentation task using the standard mask AP metric, and also report f1 scores given the fact that our test sets are highly imbalanced (there are significantly more negatives than objects of interest in the test set).

Detection: For detection, we apply the same pipeline as segmentation to obtain a local confidence map and a binarized mask. We extract the bounding box prediction from the prediction by drawing a box around the boundaries segmentation mask. We score this prediction by taking the aggregate local confience map values in the predicted bounding box region. We evaluate our detection task using the standard AP metric, and also report f1 score.

C ADDITIONAL EXPERIMENTS

C.1 HYPERPARAMETER AND LOSS FUNCTION ABLATIONS

We conduct ablations of the key training parameters of our method: # anchor-positive pairs, total # generated positives, and choice of loss function. We do so on the validation set of DF2, using the Masked Dreambooth dataset (without filtering); these were chosen arbitrarily, and intended to be representative of trends that we might expect to see across other datasets.

	# Synthetic Imgs	# Anchor-Pos Pairs	**Classification**	**Retrieval**
DINOv2	-	-	12.0	35.9
DINOv2-P	300	300	27.2	51.2
	300	600	35.9	58.7
	300	1500	38.0	61.1
	300	3000	38.1	**62.1**
	300	6000	**38.5**	61.7
	300	15000	37.7	61.3
	300	30000	36.9	60.0

Table 4: **Ablation on the number of generated positives.**

Number of anchor-positive pairs. Given a fixed number of synthetic and real images, we can potentially sample many combinations of anchors and positives for contrastive learning. We thus sweep over different ratios of generated positives to sampled anchor-positive pairs. We fix the size of $\mathcal{D}_S$, the pool of generated positives, to 300 (arbitrarily chosen) and sample increasing numbers of anchor-positive pairs from this pool for training. We find that performance on the DF2 validation set plateaus at a 1:10 ratio, and subsequently use this ratio in all of our main experiments. Results are shown in Table 4.

	# Synthetic Imgs	# Anchor-Pos Pairs	**Classification**	**Retrieval**
DINOv2	-	-	12.0	35.9
DINOv2-P	3	30	12.8	36.0
	30	300	28.6	35.9
	150	1500	36.8	60.8
	300	3000	38.1	**62.1**
	450	4500	**38.7**	61.9
	600	6000	38.3	60.8

Table 5: **Ablation on the number of anchor-positive pairs.**

Number of generated positives. We ablate the size of $\mathcal{D}_S$, the pool of generated positives. To isolate the effect of the positive pool size, we fix ratio between the size of $\mathcal{D}_S$ and the number of anchor-positive pairs that are sampled from ($\mathcal{D}_R$, $\mathcal{D}_S$). We fix this ratio to 1:10, as this was found to be best in the previous ablation. We find that performance plateaus at 450 generated positives and thus use 450 as the size of $\mathcal{D}_S$ in all of our main experiments. Results are shown in Table 5.

Loss function. We evaluate DINOv2-P trained on the Masked DreamBooth dataset (CFG 5) with three contrastive loss functions, and one non-contrastive loss. For the Cross-Entropy experiment, we add a linear layer with a sigmoid that projects the output feature vector to a single prediction scalar (1 indicating the target object, 0 for any negative). We evaluate on the validation set of DF2 and find that InfoNCE leads to the best results, and that contrastive losses overall perform better than Cross-Entropy. Results are shown in Table 6.

	Loss Function	**Classification**	**Retrieval**
DINOv2	-	12.0	35.9
DINOv2-P	InfoNCE	**36.5**	**63.3**
	InfoNCE (Multi-Positive)	27.5	28.0
	Hinge	29.4	48.4
	Cross-Entropy	24.9	37.0

Table 6: **Loss function ablation.**

C.2 FULL EVALUATIONS ACROSS SYNTHETIC DATASETS

We ablate components of our method that contribute to the diversity of generated datsets, in particular the CFG parameter, and the use of LLM-generated prompts. In Tables 7-8 we show results across all tested synthetic datasets, backbones, and downstream tasks. We highlight the synthetic datasets that lead to the best performance for each backbone (as determined by average performance across the PODS/DF2/Dogs validation sets). For these, we report the minimum/maximum performance across four seeds, with the averages shown in Table 1. Notably, LLM-generated prompts significantly improve performance on global tasks, however have little impact on dense task performance.

C.3 EVALUATION ON PERSAM

Here, we show that our personalized representations can be plugged into state-of-the-art pipelines for personal tasks. We experiment with PerSAM Zhang et al. (2023). In the PerSAM method, an image encoder is used to extract patch features of one or more training images, which are compared

Model	CFG	LLM	Classification			Retrieval		
			PODS	DF2	Dogs	PODS	DF2	Dogs
DINOv2	-	-	28.1	14.4	83.1	69.6	36.3	89.4
DINOv2-P	4	×	41.6	33.8	80.0	71.4	59.3	91.9
	5	×	41.9	32.8	80.2	70.9	58.6	91.7
	7.5	×	40.8	32.2	81.6	70.2	57.6	92.3
	4	✓	45.7	36.4	81.0	78.9	63.1	94.2
	5	✓	(46.4, 47.5)	(35.2, 36.1)	(81.3, 82.0)	(79.1, 80.46	(62.3, 63.1)	(93.9, 94.5)
	7.5	✓	42.6	36.0	81.7	68.8	61.9	94.1
CLIP	-	-	26.7	12.7	36.4	61.4	34.7	58.1
CLIP-P	4	×	41.1	25.0	57.6	59.6	45.7	72.1
	5	×	40.0	24.0	57.7	59.2	44.5	71.6
	7.5	×	38.2	24.2	57.6	58.1	43.7	72.0
	4	✓	(45.2, 45.9)	(27.2, 27.4)	(62.1, 63.3)	(69.9, 70.6)	(51.1, 51.4)	(79.8, 80.4)
	5	✓	45.9	26.4	64.6	70.5	50.4	80.3
	7.5	✓	42.9	24.9	66.0	60.3	48.7	81.0
MAE	-	-	8.7	5.2	11.2	34.6	25.8	33.6
MAE-P	4	×	11.7	12.5	25.6	27.4	23.9	36.7
	5	×	11.3	12.4	26.3	27.1	23.4	36.8
	7.5	×	11.1	12.2	27.0	26.9	22.9	37.3
	4	✓	(13.5, 14.0)	(10.9, 11.1)	(30.1, 30.3)	(29.5, 30.0)	(23.1, 23.6)	(42.5, 42.8)
	5	✓	12.9	10.8	20.8	29.1	22.3	34.3
	7.5	✓	10.1	9.7	32.2	26.2	21.4	44.8

Table 7: **Performance of personalized v. pretrained representations on global tasks.** We report results for all generated synthetic datasets, ablating both CFG and the use of LLM-generated prompts. The best dataset for each backbone (selected using validation performance) is highlighted in yellow, with the min/max performance over 4 seeds reported.

Model	CFG	LLM	Detection (mAP)			Detection (F1)			Segmentation (mAP)			Segmentation (F1)		
			PODS	DF2	Dogs	PODS	DF2	Dogs	PODS	DF2	Dogs	PODS	DF2	Dogs
DINOv2	-	-	11.3	5.2	11.0	11.6	6.6	12.8	13.2	4.3	11.8	15.1	5.6	15.1
DINOv2-P	4	×	12.9	8.7	16.0	12.8	11.0	18.1	14.6	7.8	16.1	15.9	9.5	19.7
	5	×	12.8	8.5	15.3	12.7	10.6	17.5	14.5	7.8	15.5	15.8	9.3	19.3
	7.5	×	13.0	8.5	15.6	12.9	10.5	17.7	14.3	7.5	15.7	15.7	9.0	19.0
	4	✓	12.5	9.1	15.9	11.8	11.0	18.4	13.2	8.7	16.0	14.0	10.3	19.8
	5	✓	(12.3, 13.3)	(9.1, 9.4)	(15.7, 16.6)	(12.0, 12.9	(11.0, 11.5)	(18.6, 19.3)	(13.9, 14.1)	(8.8, 9.2)	(16.0, 16.7)	(14.6, 14.9	(10.4, 10.9)	(20.2, 20.7)
	7.5	✓	12.8	9.6	16.1	12.7	11.7	18.5	14.9	8.8	16.3	16.3	10.5	20.2
CLIP	-	-	0.1	3.1	6.5	0.1	3.8	7.0	0.3	3.2	7.6	0.3	3.9	8.6
CLIP-P	4	×	0.4	4.4	7.3	0.2	5.3	8.1	1.0	4.7	8.7	0.5	5.2	9.9
	5	×	0.5	4.2	7.3	0.2	5.1	8.1	1.0	4.6	8.7	0.6	5.0	9.8
	7.5	×	0.5	4.2	7.8	0.2	5.1	8.5	1.0	4.4	9.1	0.6	4.9	10.2
	4	✓	(0.4, 0.6)	(4.4, 4.5)	(8.4, 8.8)	(0.2, 0.3)	(5.2, 5.4)	(9.3, 9.7)	(0.9, 1.0)	(4.4, 4.6)	(9.5, 10.0)	(0.6, 0.6)	(5.0, 5.2)	(10.9, 11.3)
	5	✓	0.6	4.4	9.2	0.3	5.3	10.0	0.9	4.6	10.4	0.5	5.2	11.7
	7.5	✓	0.2	4.5	9.7	0.1	5.3	10.5	0.6	4.7	10.6	0.4	5.3	11.9
MAE	-	-	0.1	1.0	1.1	0.1	1.6	1.7	0.1	0.2	0.2	0.0	0.4	0.3
MAE-P	4	×	0.2	1.1	1.1	0.2	1.8	1.5	0.0	0.3	0.2	0.0	0.5	0.3
	5	×	0.2	1.1	1.0	0.2	1.7	1.5	0.0	0.3	0.2	0.0	0.4	0.3
	7.5	×	0.2	1.1	1.0	0.2	1.7	1.5	0.1	0.3	0.2	0.0	0.4	0.3
	4	✓	(0.2, 0.2)	(1.2, 1.3)	(0.9, 1.0)	(0.2, 0.2)	(1.8, 2.0)	(1.5, 1.6)	(0.0, 0.0)	(0.3, 0.3)	(0.2, 0.3)	(0.0, 0.0)	(0.5, 0.5)	(0.4, 0.4)
	5	✓	0.3	1.2	1.0	0.3	1.8	1.6	0.1	0.3	0.2	0.0	0.5	0.3
	7.5	✓	0.2	1.2	0.9	0.2	1.9	1.5	0.0	0.3	0.2	0.0	0.5	0.4

Table 8: **Performance of personalized v. pretrained representations on dense tasks.** We report results for all generated synthetic datasets, ablating both CFG and the use of LLM-generated prompts. The best dataset for each backbone (selected using validation performance) is highlighted in yellow, with the min/max performance over 4 seeds reported.

to the patch features of a test image to extract a confidence map. The confidence map is then used to generate keypoint proposals to prompt SAM, and guide the attention map of the SAM decoder (refer to Zhang et al. (2023) for details). We evaluate PerSAM with DINOv2, and DINOv2-P as the image encoder, across DF2, Dogs, and PODS. Our results are shown in Table 9; we find that DINOv2-P improves personalized segmentation performance over DINOv2 over all datasets. Note that we use DINOv2-P trained with the best-performing dataset for the DINOv2 backbone, highlighted in Tables 7-8.

		Segmentation (mAP)	Segmentation (F1)
PODS	DINOv2	17.1	21.6
	DINOv2-P	21.5 ↑	25.3↑
DF2	DINOv2	5.2	6.8
	DINOv2-P	10.9 ↑	12.8 ↑
Dogs	DINOv2	15.4	19.8
	DINOv2-P	19.5 ↑	24.4 ↑

Table 9: **Application to PerSAM**

C.4 DIVERSITY AND FIDELITY ANALYSIS

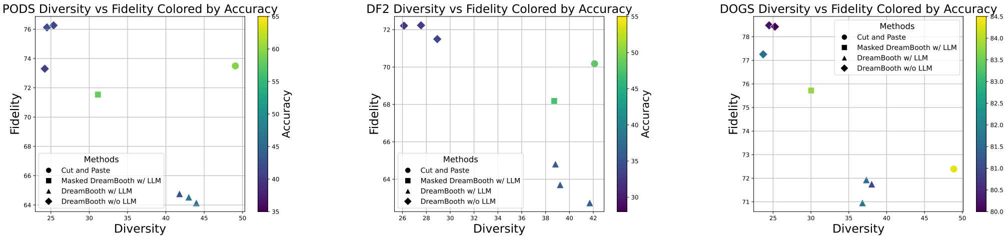

Figure 12: **Diversity and fidelity plot colored by accuracy** across PODS, DF2 and Dogs synthetic datasets. Note that the fidelity metric may be influenced by the background features appearing in the cropped image, resulting in a reduced fidelity score for some samples.

To better understand the characteristics of our generated datasets, we perform a diversity and fidelity analysis in Figure 12. We measure fidelity by computing the DreamSim cosine similarity between synthetic images and the mean embedding of the reference real images, both cropped around the object of interest. To measure diversity, we compute the pairwise similarity between DreamSim features of the synthetic images in each dataset.

Notably, optimal accuracy is achieved when both fidelity and diversity are sufficiently balanced—too much fidelity at the expense of diversity, or too much diversity with low fidelity, both lead to degraded performance. Maintaining appropriate levels of both results in the best performance, and this can be achieved in our work through Masked DreamBooth or Cut and Paste generative methods.

D QUALITATIVE RESULTS

D.1 DENSE PREDICTION VISUALIZATIONS

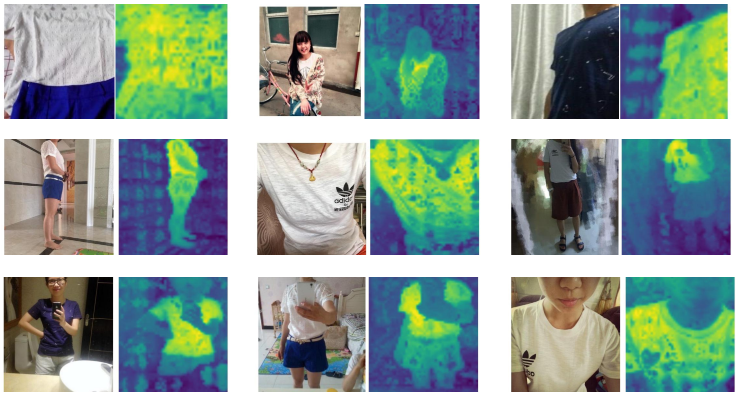

Figure 13: **DF2 Dense Predictions** Images and classes are randomly sampled

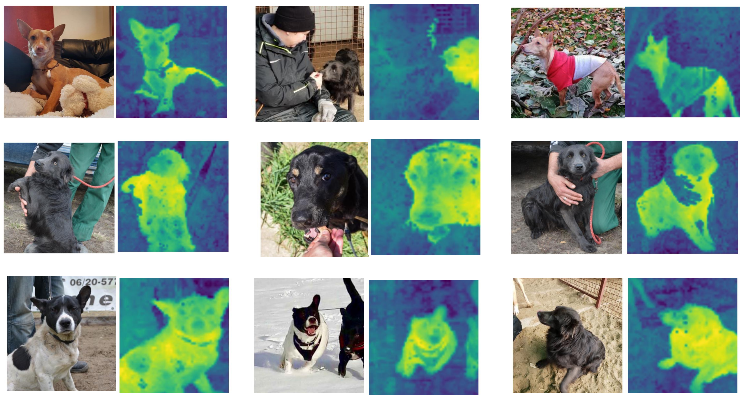

Figure 14: **Dogs Dense Predictions** Images and classes are randomly sampled

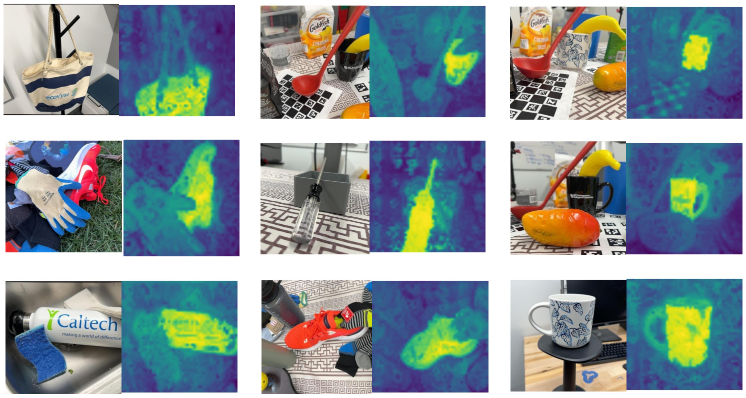

Figure 15: **PODS Dense Predictions** Images and classes are randomly sampled

D.2 CHALLENGING EXAMPLES

We visualize examples of hard negatives and hard positives from the Dogs dataset to better understand the capabilities of personalized representations. For a given query image x of a target object, we identify hard positives as the k positives with the lowest DINOv2 cosine similarity to x (often the dog in a very different setting or position), and hard negatives as the k negatives with the highest DINOv2 cosine similarity to x (often different dogs of the same breed). We denote DINOv2-P as successful on a hard negative (positive) if it has a lower (higher) cosine similarity to x than DINOv2.

We show randomly selected examples of hard positives/negatives in the Dogs dataset in Figures 16 and 17 respectively. DINOv2-P is typically successful on hard positives; the cosine similarity between positive pairs nearly always increases, even in cases with significant differences (lighting, pose, etc) from the query image. However, we also identify several cases in which the cosine similarity between query images and hard negatives also increases. This is a failure case that may be induced by noisy positives in the synthetic training dataset, leading the personalized representation to associate the target object with spurious features.

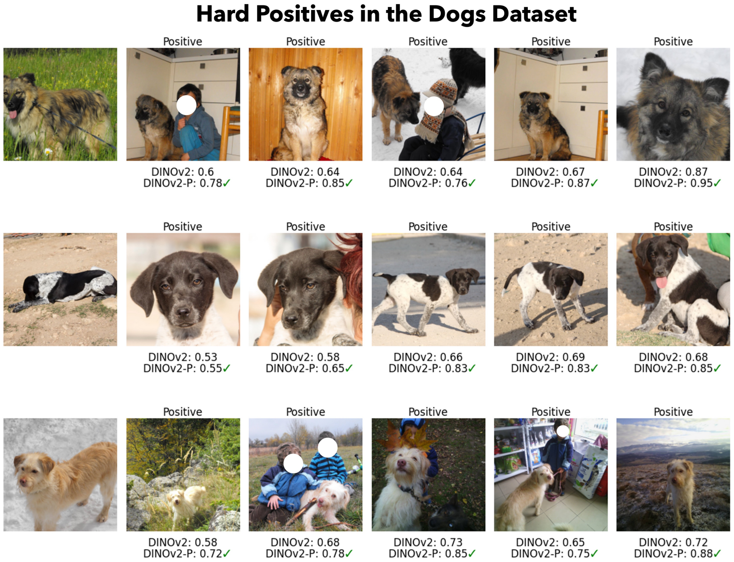

Figure 16: **Dogs Hard Positives.** Given images of a target dog (leftmost of each row) we identify the positive test images with the lowest DINOv2 similarity to the query. DINOv2-P cosine similarity typically increases, even for cases with significant differences in setting, camera angle, occlusion, etc.

D.3 DREAMBOOTH GENERATED DATA

We present qualitative examples for the following datasets:

- PODS DreamBooth without LLM 18, PODS DreamBooth with LLM 19, PODS Masked DreamBooth with LLM + Filtering 19, PODS negatives 21
- Dogs DreamBooth without LLM 22, Dogs DreamBooth with LLM 23, Dogs Masked DreamBooth with LLM + Filtering 24 and Dogs negatives 25

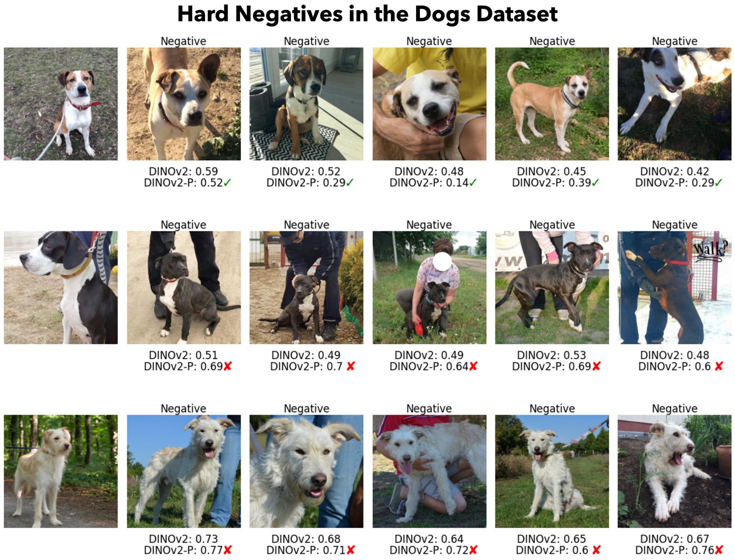

Figure 17: **Dogs Hard Negatives.** Given images of a target dog (leftmost of each row) we identify the negative test images with the highest DINOv2 similarity to the query. In some cases DINOv2-P cosine similarity decreases (top row) however we also identify failure cases where cosine similarity increases (second/third rows).

PODS Generated Images

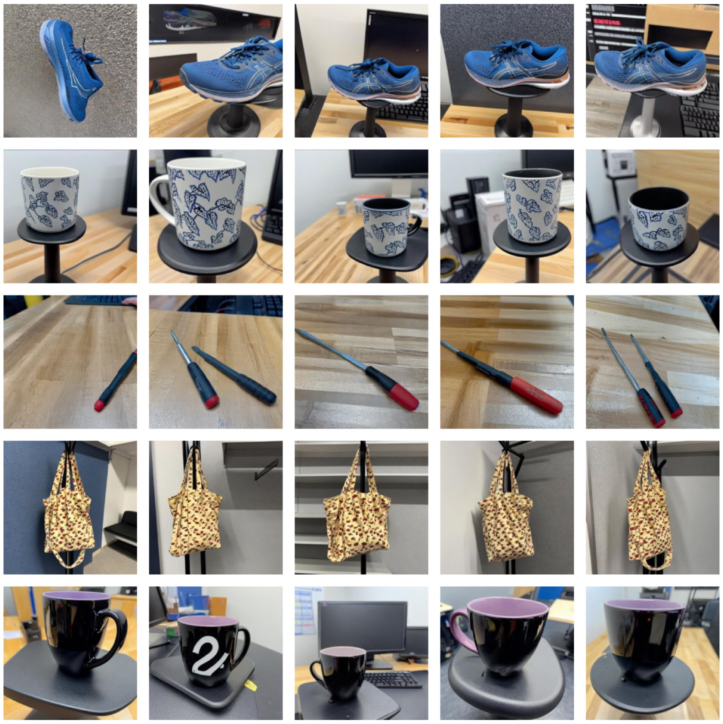

Figure 18: **PODS Generated Images - DreamBooth without LLM prompting.** Images and classes are randomly sampled

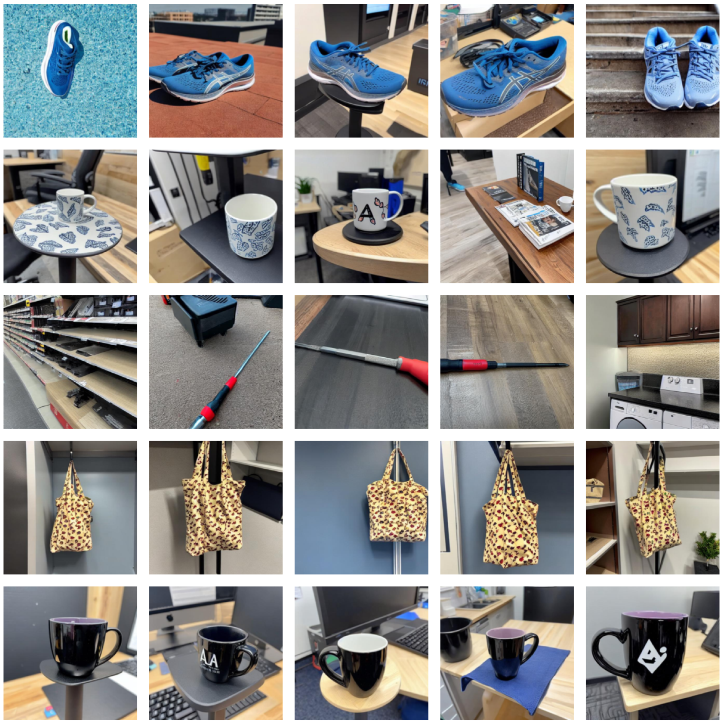

Figure 19: **PODS Generated Images - DreamBooth with LLM prompting.** Images and classes are randomly sampled

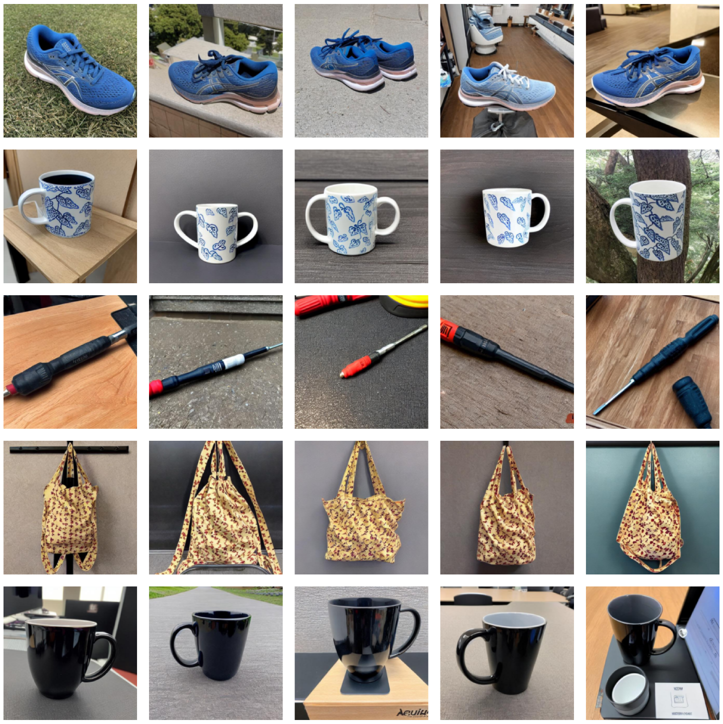

Figure 20: **PODS Generated Images - DreamBooth with LLM, Masking and Filtering.** Images and classes are randomly sampled

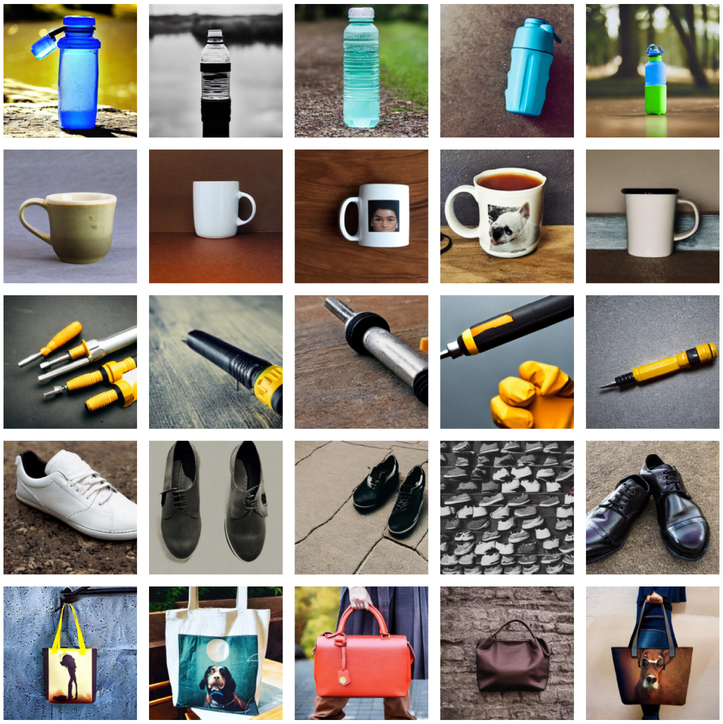

Figure 21: **PODS Generated Negatives** Images and classes are randomly sampled. Each row are sampled negatives for each object category.

Dogs Generated Images

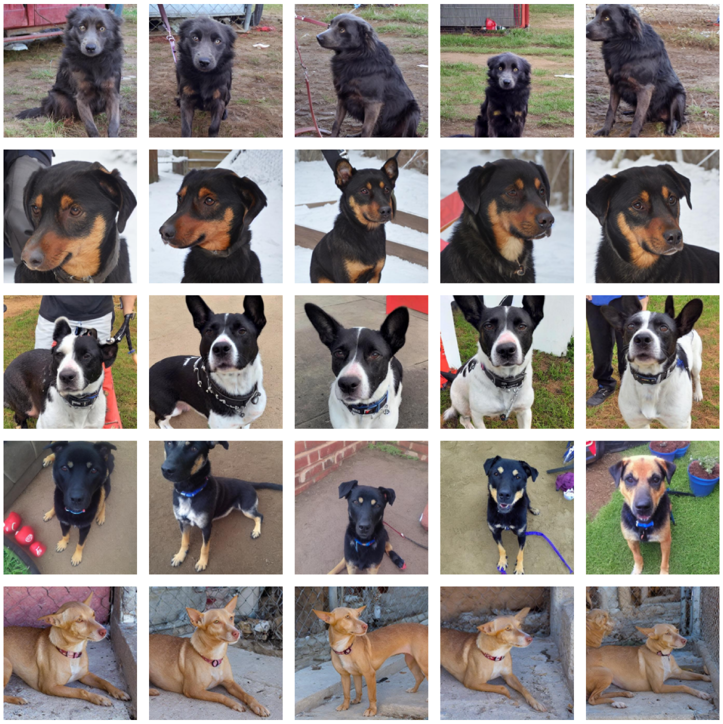

Figure 22: **Dogs Generated Images - DreamBooth without LLM prompting.** Images and classes are randomly sampled

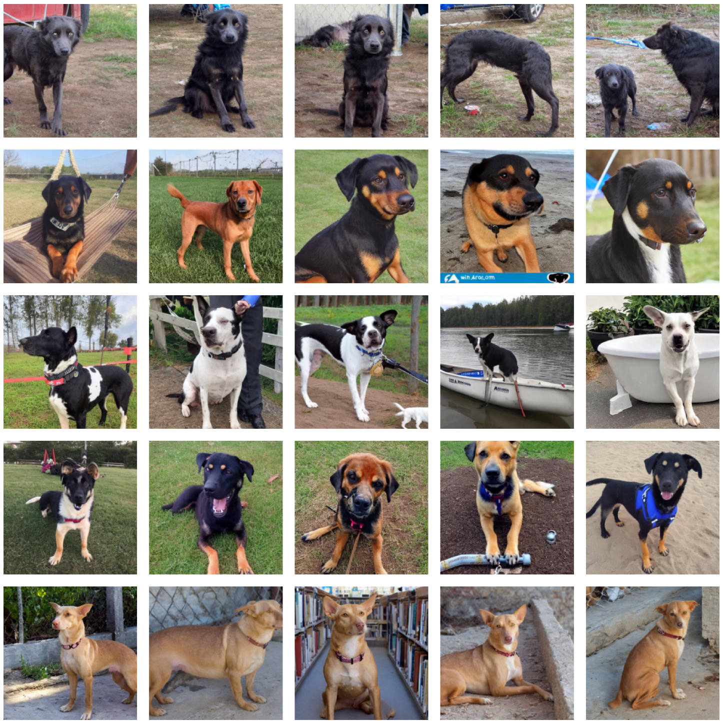

Figure 23: **Dogs Generated Images - DreamBooth with LLM prompting.** Images and classes are randomly sampled

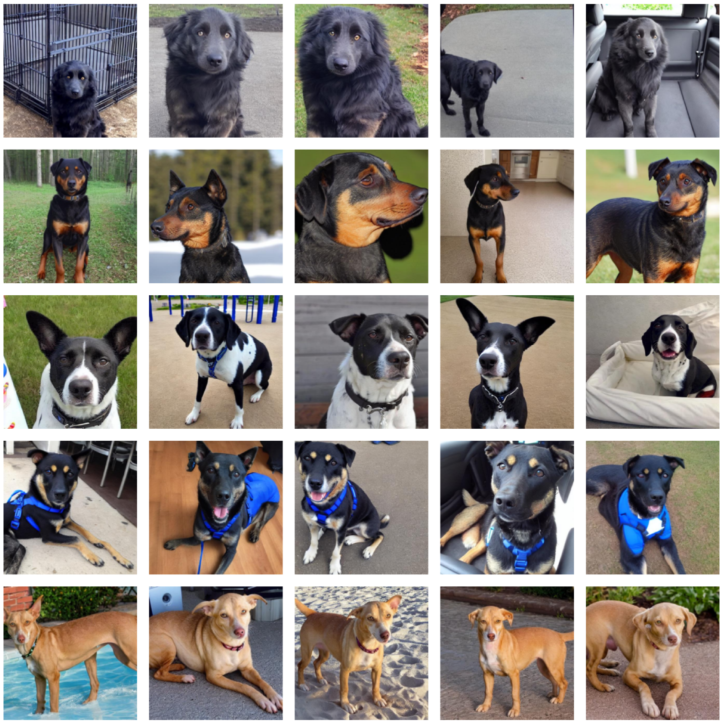

Figure 24: **Dogs Generated Images - DreamBooth with LLM, Masking and Filtering.** Images and classes are randomly sampled

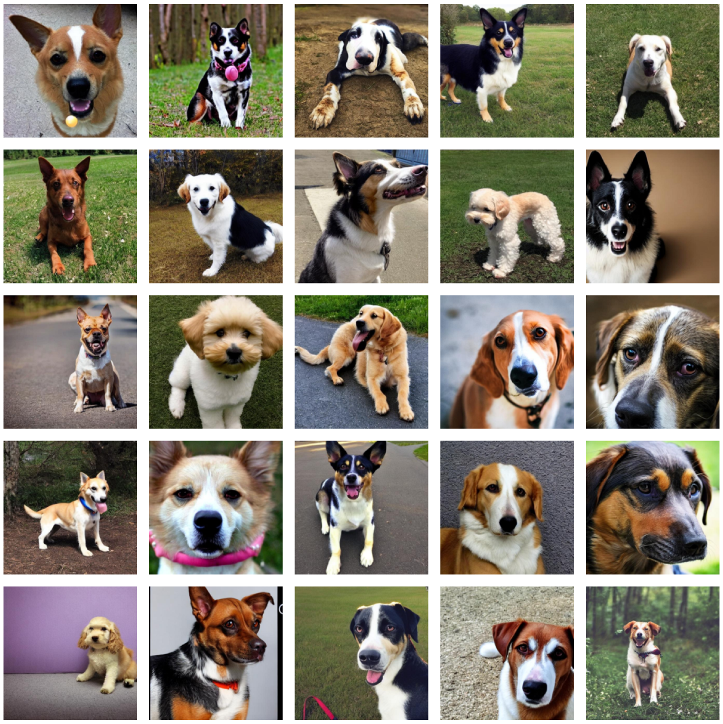

Figure 25: **Dogs Generated Negatives** Images and classes are randomly sampled