Personalizing Retrieval using Joint Embeddings; or "the Return of Fluffy"

Bruno Korbar
Visual Geometry Group
University of Oxford
Oxford, UK
korbar@robots.ox.ac.uk

Andrew Zisserman
Visual Geometry Group
University of Oxford
Oxford, UK
az@robots.ox.ac.uk

***Abstract*—The goal of this paper is to be able to retrieve images using a compound query that combines object instance information from an image, with a natural text description of what that object is doing or where it is. For example, to retrieve an image of 'Fluffy the unicorn (specified by an image) on someone's head'. To achieve this we design a mapping network that can 'translate' from a local image embedding (of the object instance) to a text token, such that the combination of the token and a natural language query is suitable for CLIP style text encoding, and image retrieval. Generating a text token in this manner involves a simple training procedure, that only needs to be performed once for each object instance. We show that our approach of using a trainable mapping network, termed π-map, together with *frozen* CLIP text and image encoders, improves the state of the art on two benchmarks designed to assess personalized retrieval.**

***Index Terms*—video search, personalization, retrieval.**

I. Introduction

Large-scale pre-trained vision-language models (VLMs) alleviated the need for training task-specific models due to their emerging capability for both intra- and cross-modal retrieval. By enforcing the alignment of text and images, these models allow us to classify objects and scenes, retrieve relevant images given a textual description, and even spatially locate specific objects in an image. However, in practical uses, we are often interested in searching for a specific "thing" in an image. On our phones, we may have hundreds of images of dogs, but we may only be interested in one specific dog – our dog "Chia". Searching our library for "My dog Chia with a stick", since the VLMs have no knowledge of our dog Chia, might return either a generic dog or, for example, chia seeds. But what if we want to 'teach' a VLM what "my dog Chia" refers to? Given the name of the dog and a few template images, can we 'teach' a VLM to recognise our dog?

In prior work, this problem has been referred to as the "personalization" of VLMs [3], [28].

The great advantage of achieving this personalization is that we then can deploy the compositional power of the VLM, and search for "my dog Chia" carrying out various activities and in different environments simply by writing our query as a natural language sentence, as illustrated in Fig 1. Our approach is inspired by the language model's almost infinite expressively; given a specific-enough query, the large language model used in most popular VLMs *should* be able to synthesise information necessary for better text-to-image retrieval. Therefore, one could argue that the task of personalization might be expressed as learning that 'my dog Chia' corresponds to 'an adorable[1] 4-month old blue-eyed husky mix with grey inverse mask and white socks and features, about 10 inches high'.

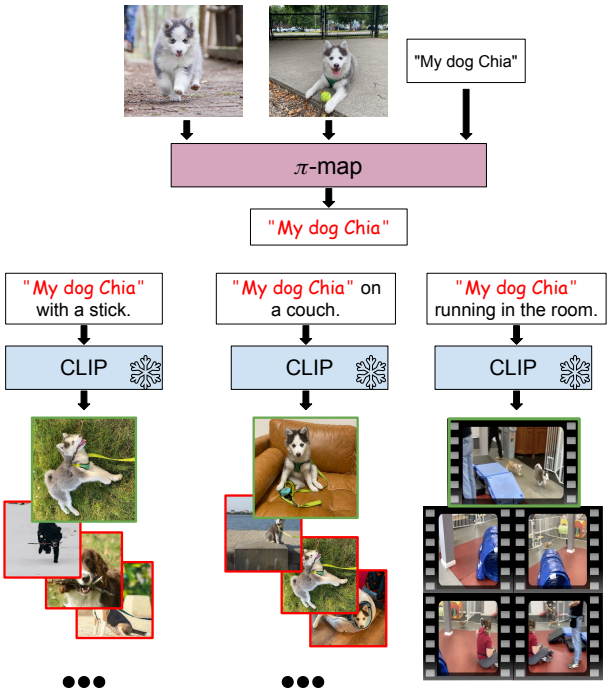


Fig. 1: Given a few example images of an instance, our π-map model learns a personalised text embedding for the instance ('"My dog Chia"). This text embedding can then be composed with free form text queries to search amongst a dataset of images or within video frames.

Our approach trains a 'translation' network that can map from a few example template images of the object of interest to a suitable text embedding. The text embedding is then used in query sentences for the personalized search for that object. We are not the first to attempt this (for example, personalization is the goal of the paper "this is my unicorn, fluffy" [3]), and

[1] not strictly relevant

our solution builds on those of others, but our method has fewer requirements than prior work, and demonstrates superior retrieval performance.

In terms of requirements: we are able to use *frozen* CLIP image and text encoders (whereas previous work fine-tuned the text encoder [11]); and by using a *local image embedding* we require fewer and less diverse training images than prior work for the personalization – avoiding the failing of learning the context of the image background rather than the foreground object of interest [3]. Furthermore, we leverage a LLM's expressivity to automatically generate caption augmentations in the language domain. Also, unlike previous work [28], training does not require retrieval from a large dataset, so it is efficient.

In terms of performance, we demonstrate superior retrieval performance compared to previous methods over **two** standard benchmark datasets: this-is-my' [28] and 'DeepFashion2' [3], [6].

II. Related Work

a) Methods for translating between image and text embeddings.: Translation between the modalities of VLMs is a well explored topic related to the task of personalization. Mokady *et al*. [15] show that a single mapping network can translate encoding from images to the text model. They fully finetune the text encoder. Alayrac *et al*. propose training adapter models that map a visual input to the LLM domain using a model dubbed 'Perceiver Resampler'. With such mappings, they only train adapter layers within a LLM [1]. Li *et al*. [12] devise an even more efficient model ('Q-former') and a two-stage training method that translates any arbitrary large vision transformer into the domain of LLMs with no need for additional adapter layers. These methods have became a de-facto choice for tasks such as retrieval [1], [4], [7], and for visual question answering [1], [13], [27].

An inherent discrepancy between the text and image embeddings has also been a subject of extensive study. Nukrai *et al*. show that noise injection during the CLIP training process helps alleviate the 'modality gap' [16]. Schrodi *et al*. show that this modality gap can be attributed to as little as two dimensions within each embedding [19].

b) Test-time adaptation.: The task of test-time adaptation (TTA) and various fine-tuning approaches are closely related to the personalization of VLMs. The goal of TTA is to leverages the unlabeled data that arrives at test time by adapting either the forward pass or parameters of the model according to some proxy task [2], [18]. While in the task of personalization we aim to preserve model's capabilities and only specialise it for one or two instances, the task of TTA generally requires a distribution shift of an entire model. Zhao *et al*. [30] show that VLMs can be adapted to out-of-distribution samples using reinforcement learning from CLIP's feedback. Gao *et al*. [5] show that a feature adapter can replace the need for fine-tuning VLMs. Wortsman *et al*. [26] present a robust method of fine-tuning VLMs to adapt to the test time data.

c) Personalization methods for Joint Embedding Retrieval.: Korbar and Zisserman [11] have explored how VLMs textual encoder can be augmented to associate a given face embedding with the corresponding name and use either interchangeably to retrieve relevant videos. This method relies on having strong face embeddings and is, therefore, limited to the domain of faces. Wang *et al*. [25] demonstrated that expert embeddings from [11] can be replaced by a method that finds the closest generic prompt embedding to the novel class. They learn an 'expert' prompt which is a function of the generic prompt. They focus on novel class discovery rather than on learning instance-specific attributes. Cohen *et al*. [3] proposed extending VLM's language encoder's vocabulary with a newly learned token which represents a specific instance. Their method assumes a clean, manually annotated dataset of specific instances, which are seldom available. Yeh *et al*. [28] learn a database of common traits of a given category (a process they dub "meta-personalization") and then learn a specific personalised embedding as a weighted combination of global category features. While this approach does not need a large example database (as general-category objects can be discovered automatically), it is limited to the number of common category traits it can store.

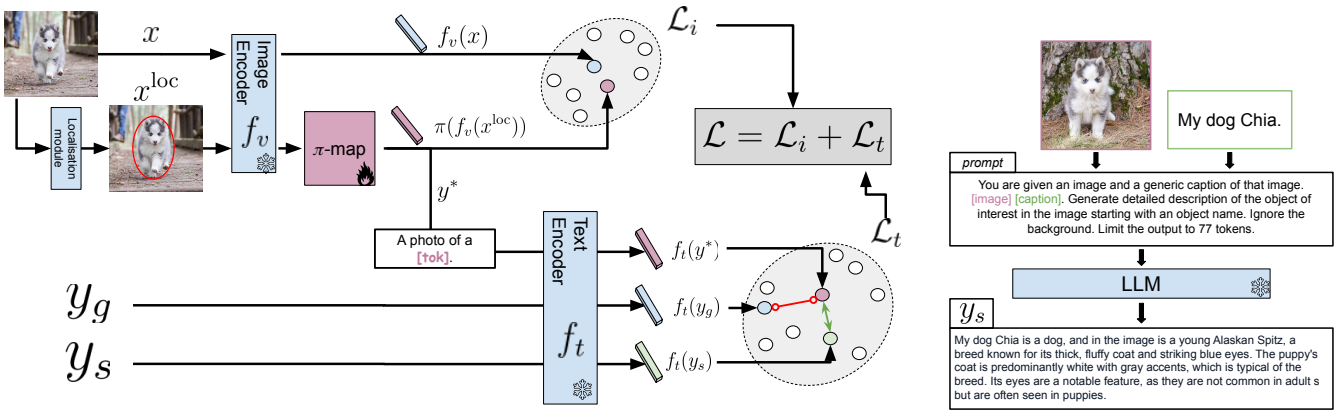
d) Compound retrieval.: In compound text-to-image retrieval – retrieval over multiple semantic axes, the focus is on specificity over each axis. Ventura *et al*. developed a large-scale compound retrieval benchmark collected automatically by mining web-video captions [24]. Zhong *et al*. [31] present a compound retrieval image dataset containing the axis 'person' and 'scene', while Korbar and Zisserman [11] present a video benchmark containing the axis of 'person', 'action', and 'scene'.

e) Conditional retrieval.: Similarly, a task of conditional retrieval seeks to retrieve a particular version of an image given constrained parameters, e.g. given a daytime image of Eiffel tower and a text prompt "at night", the task is to retrieve a nighttime image. Khartik *et al*. [10] demonstrate a method of prompt adaptation using large language models. Gu *et al*. [8] show superior performance on the same task using latent representation of diffusion models. While this task resembles personalisation in that it aims to retrieve a particular verison of an instance, there is no requirement to name that instance.

III. Method

This section introduces our personalized retrieval method, describes our proposed **P**ersonalised **I**mage **E**mbedding Mapping model (PIE-map or π-map), explains the training procedure, and compares our approach to related work.

Overview: Given a few example images of a specific object, our method generates a personalized embedding (a unique text token) through a brief one-time training process using π-map (top part of Fig. 1). With this personalized embedding (e.g., "My dog Chia"), queries such as "My dog Chia playing in the park" are formed by combining the personalized token with additional descriptive text tokens. The CLIP text encoder processes this combined query into an embedding, which then



(a) (b)

Fig. 2: (a) **Generating a text token, y^*, for a specific object instance**. The token y^* is obtained by fine-tuning the π-map given an image x of the instance and a specific text description y_s. The π-map is fine-tuned such that the text embedding of y^* is close to the text embedding of the specific description y_s but away from the text embedding of the the generic class description y_g. Also, as a regularization, y^* is close to the original image embedding. The total loss is a linear combination of text embedding loss, $\mathcal{L}_t$, and the image embedding loss, $\mathcal{L}_i$. (b) Caption augmentation using an LLM [22]

retrieves relevant images from a dataset by ranking them according to their embedding similarity. An illustration of this can be seen at the bottom of Fig. 1.

Notation: Let x be an example (template) image of the object instance, y^* is the personalized text token we seek denoting a specific instance ('my dog Chia'), y_g is text denoting a generic object category ('dog'), and y_s is text describing the specific instance. Let $f_t(\cdot)$ and $f_v(\cdot)$ be text and image encoders of a CLIP VLM respectively [17]. The mapping is denoted as π.

Fine-tuning π-map to obtain the text token y^*: The π-map transforms visual embeddings into representations that align closely with text embeddings; essentially converting visual data into text-like "words". To aim is to obtain a text token that represents the specific object instance (of Chia) from the example image, but is 'distinct' from the embedding of the general category (dogs).

The template image x is used to obtain (i) a localized image embedding $f_v(x^{loc})$, and (ii) a detailed text description y_s (as described below). Personalised text token y^* is then obtained by minimising the following objective function for :

$$\mathcal{L} = (1-\alpha)\mathcal{L}_t + \alpha\mathcal{L}_i$$

where $\mathcal{L}_t$ is a contrastive loss in the text embedding space that ensures that text embedding of 'an image of y^*' is close to the text embedding of the detailed description y_s, but far from the text embedding of the generic object category ('dog'), y_g. $\mathcal{L}_i$ is a contrastive regularization loss in the image embedding space that ensures that the text embedding of y^* is close to the image embedding of the original image x. α is a loss balancing hyperparameter (determined by line search to be $\alpha = 0.25$).

These losses are illustrated in Fig. 2, and described in detail in the following subsection.

Inference: Once the token y^* has been obtained by fine-tuning π-map (and this only needs to be done once for each instance), then it is appended to the rest of the text of a query and passed through the text encoder. We measure the similarity between the query embedding, f_t, and the dataset of visual features (the image embedding of each image, f_v) and rank based on this scalar product. An overview can be seen in Fig 1.

A. *Model details*

Architecture: We use a three-layer Multi-Layer Perceptron (MLP) featuring residual connections. Additionally, we introduce two learnable conditioning embedding vectors that influence the outputs of the first two MLP layers.

These conditioning vectors guide the image-to-text embedding transformation, encouraging the embeddings to emphasize dimensions where the most significant discrepancies between image and text embeddings occur. As noted by [19], image and text embeddings are primarily distinguished by two key dimensions. Removing these one or two principal components would render the embeddings indistinguishable between modalities.

Thus, our model amplifies these critical dimensions similarly to the way attention mechanisms operate [23]. Specifically, the outputs from each of the first two MLP layers are multiplied by conditioning vectors, strategically enhancing the values of the embedding's most relevant components before undergoing a final linear projection. This ensures the model precisely focuses on dimensions essential for differentiating between visual and textual embeddings.

Obtaining the localised embedding. By definition, $f_v(x)$ is a global embedding. Therefore, it is sensitive to the image background and context. Say all photos of 'my dog Chia' come from a forest. The model would then be biased to all images of a dog in a forest and might completely miss 'Chia' in the street. We alleviate this issue by using $f_v(x^{loc})$ – a localised version of the embedding.

To localise the embeddings, we build on a technique by Shtedritski *et al*. [20] who demonstrated that drawing a red ellipse around the area of interest focuses the semantic image embedding to the region within it, and Sun *et al*. [21] who showed that such visually augmented image can be used as a localised embedding for downstream tasks.

Inspired by this work, we obtain an image with a red 'circle' (an ellipse) by using a pre-trained language-guided detector to detect objects in the image [14]. We empirically demonstrate that adding a red circle around the instance to the template images during training increases performance and reduces the number of template images we need to form a personalised embedding.

Obtaining a detailed text description: We augment the caption y_s automatically by passing an image x^{loc} and a prompt (see illustration in Fig. 2b) to a large language model [22] to form a detailed text description. Previous work by Schrodi *et al*. [19] has shown that a more expressive caption can diminish the information imbalance between text and image embeddings

Image loss $\mathcal{L}_i$: The local image embedding is mapped to the text input using π-map to obtain $\pi(f_v(x^{loc}))$ which becomes the basis of our personalized token y^*. While most of the training is done in the text domain, we do not want $\pi(f_v(x^{loc}))$ to collapse to an encoding of a word 'dog'. Therefore we keep a regularization loss $\mathcal{L}_i$ which keeps $\pi(f_v(x^{loc}))$ and $f_v(x)$ close. Formally we use contrastive loss formulation:

$$\mathcal{L}_i(\pi(f_v(x^{loc}))) = -log(\frac{d(\pi(f_v(x^{loc})), f_v(x))}{\sum_{n \neq x \in \mathbb{B}} d(\pi(f_v(x^{loc})), f_v(n)}) \quad (1)$$

where $\mathbb{B}$ is a randomly sampled training minibatch, a distance metric is given by $d(a,b) = exp(\frac{a^T b/\tau}{||a^T||_2||b||_2/\tau})$, and $\tau = 0.07$ is a temperature hyperparameter. Intuitively, the regularisation loss $\mathcal{L}_i$ ensures that the projected embedding $\pi(f_v(x^{loc}))$ does not drift from the original embedding and thus retains its semantic information.

Text loss $\mathcal{L}_t$: This is based on three text prompts: a generic text prompt ("A photo of a dog", y_g), a specific detailed text prompt ("My dog chia is an alaskan spitz...", y_s), and the learned π-map embedding y^*. These text embeddings y_g, y_s, and $y^* = \pi(f_v(x^{loc}))$ are then passed through CLIP text encoder f_t. Since we are only training a single embedding, we want to specialise it by making it close to y_s (learning the semantic correspondence to the detailed information), while making it less sensitive to the more general class y_g (therefore forcing our model to extract more specialised information). We achieve this by optimising a contrastive objective $\mathcal{L}_t$ which ensures that $y^* = f_t(\pi(f_v(x^{loc}))$ is similar in the embedding space to $f_t(y_s)$ while being away from $f_t(y_g)$.

$$\mathcal{L}_t(\pi(f_v(x^{loc})), y_s) = -log\frac{d(f_t(y^*), f_t(y_s))}{\sum_{y_i \in \mathbb{N}} d(f_t(y^*), f_t(y_i))} \quad (1)$$

where $\mathbb{N}$ is a set of negative examples comprising all other specific and generic captions in $\mathbb{B}$ not equal to y_s.

B. Discussion: relation to previous methods

Compared to the CLIP-PAD approach of [11], we do not train the language encoder but instead use frozen versions of CLIP's image and language encoders, training only a separate module for the image-to-text translation mapping π-map. This is a great advantage as π-map can simply be 'plugged in' to existing deployments of CLIP for retrieval.

Both PALVARA method of [3] and personalization approach of [28] learn direct text-replacement from images; [3] from set encoding, and [28] by learning a linear combination of known features. This means that *(a)* during the personalization stage, both of these are limited to learning from concepts they already 'know'. [28] can only represent instances that can be expressed by the linear combination of their meta categories and [3] only personalises tokens learned from object detection datasets. π-map can on another hand be trained on top of an arbitrary CLIP model directly as its pre-training stage is general. It also means that *(b)*, querying using an image token directly is impossible as [28] requires a text prompt and [3]'s prompts are fixed after the personalization stage. Because we learn direct mapping from image to text, any image can be used as a query by mapping it into the text embedding space.

C. Implementation details

VLM details: For fair comparison with prior work ([3], [28]), we use OpenAI's CLIP (VIT-B/16) [17].

Pre-training π-map: In order to initialize model and bring the modalities closer together, we pre-train π-map on ImageNet by minimizing symmetric cross-entropy loss (following [17]) between $f_v(x)$ and $f_t(\pi(f_v(x))$. Intuitively, we are trying to 'teach' π-map to map an image of a 'dog', to that of the text encoding of 'dog'. This is done only once.

Initialise π-map's conditioning vectors: We found that initialisation of conditioning vectors matters. To initialise the vectors, we first compute the image embeddings of template images and corresponding text caption, compute the absolute pairwise difference between embedding dimensions, and finally find the two dimensions with the maximum abs. difference. We use this difference vector, zeroing out the largest and second largest dimension respectively, before taking the softmax to initialise the first and second vectors respectively. The illustration of our model can be seen in Fig 2 on the right. The effect of this initialisation scheme can be seen in Table I.

Localisation details: To obtain localised image x^{loc}, we pass the original image and its general category to a pre-trained language-guided detection model GroundingDINO ('GroundingDINO-B') [14]. For a text prompt, GroundingDINO returns coordinate of the bounding boxes of the object. We superimpose an ellipse onto the image that passes through a centre of each side of the bounding box.

Obtaining a detailed text description: In order to augment the caption, we forward the prompt defined in Fig 2b and feed it to the REKA-Core model [22].

Training details: We pre-train the model using a batch size of 256 and a learning rate of $3e-4$ for 10 epochs. For

personalization, we train the model for 50 epochs on 'this-is-my' dataset, and for 80 epochs on 'DeepFashion2' with a cosine annealing learning rate starting at $1e-4$ with 200 steps of linear warmup. All training is done with AdamW optimiser. In practice, a training run for learning 15 personalised tokens on 'this-is-my' training set takes about 54 minutes, or 3.6 minutes per-token on a single A4000 chip.

Using multiple example images. When multiple example images are present, we randomly sample one to generate detailed description y_s, and keep it fixed for training across all template images. To generate the final embedding y^*, we average the π-map projection of them.

Extension to videos. Keen-eyed readers would have noticed that x has thus far been described as an image, but one of the datasets contains video training data. For such cases, we subsample a training query video to 10 frames uniformly, localise the specific instance if applicable, and encode the frames using visual encoder f_v. We then average embeddings to get $f_v(x)$ and $f_v(x^{loc})$.

IV. Datasets and Evaluation Measures

In this section, we describe the datasets used for evaluating our personalization method, as well as the evaluation measures used for each of them.

A. This-is-my

Yeh *et al*. proposed 'this-is-my' [28] for personalised text-to-*video* retrieval. The dataset consists of 104 training segments, 683 evaluation segments, and 30 test segments annotated with ten general categories (e.g. 'dog') and 15 specific categories (e.g. 'my dog Biscuit'). We use it for method development and downstream performance evaluation.

Evaluation procedure: In order to evaluate our model, we extract CLIP image features from 30 test segments, uniformly sampling frames at 1fps following the prescribed protocol in [9], [17]. We embed the textual query using CLIP, with its tokenizer trained to recognise each of the 30 instances. We define similarity for a given video as the maximum dot product between the query and all video features. For the *generic* setting ('An image of y^*') and report mean average precision (mAP) and mean reciprocal rank (MRR) following [28]. For the *context*ualised setting ('A photo of y^* in the car park'), there is only one correct match, hence we report MRR and recall-at-5 (R@5). For a fair comparison with SOTA, once the training hyperparameters are set, we train the model on both the train and eval set as [28] train their model on both (eval set is referred to as 'personalisation' dataset in their work). The embedding is formed by using 5 randomly sampled and localised frames from an eval video.

B. DeepFashion2

Cohen *et al*. [3] proposed a modified version of DeepFashion2 [6] for personalization purposes. They curated a dataset of 653 training and 221 evaluation images that have assigned one of 50 [CONCEPT] tags: e.g. 'a white skirt', 'a short dress', etc. For the evaluation images, they collect in-context captions such as 'The [CONCEPT] is facing a glass store display' (short caption) or 'White cabinets, some with open drawers, are alongside and behind the [CONCEPT]' (long caption). Overall, 50 total concepts are contained in the dataset.

Evaluation procedure: As DeepFashion2 is an image dataset, we simply encode each image with a CLIP visual model, and follow the same evaluation protocol as for the 'this-is-my' dataset otherwise. We follow the benchmark setting from [28] and use five images to form the embedding.

V. Results

In this section, we present the results of our method. Sec V-A presents various ablation studies taken into account while designing the model. We then compare our trained model with state-of-the-art (SOTA) on the personalisation datasets described in Sec IV: 'this-is-my' [28] (in Sec. V-B), and 'DeepFashion2' [6] (in Sec V-C) datasets. In the supplementary, we demonstrate that our model can work in a feed-forward fashion on a identity-specific retrieval dataset. Finally, we discuss our findings and limitations in Sec. V-D. Qualitative results can be found in Figure 3.

A. Ablation study

In this section, we evaluate our design choices on the evaluation section of the 'this-is-my' dataset. As we want to obtain finer-grained insight into our model's performance, we compute what we call true R@5 (or tR@5, defined as a number of correct examples retrieved in top-5 over a number of all positives) and precision-at-5 (P@5; the proportion of positive examples retrieved in top-5). Note that maximum theoretical tR@5=40.3.

Table I (a) shows that our model significantly outperforms naive CLIP baselines. Table I (b) explores the effects of our modelling choices discussed in Sec. III on the downstream performance. It is notable that caption augmentation plays a significant role in achieving good results ($+10.9$ precision points). While localisation plays only a minor role in the overall result, it allows us to achieve similar results with a lower number of frames (Tbl. II).

B. this-is-my

Results on this-is-my datasets are reported in Table III. Our model using an image (randomly selected and held out from the training set) comfortably outperforms all other methods. It is notable that, although the text encoding is computed using an average of the template training images, using an image as a query as opposed to the text yields better results.

C. DeepFashion2

Our results on DeepFashion2 [6] show marginal improvement over previous methods. DeepFashion2 is also the only dataset where caption augmentation did, in fact, cause adverse effects (54.7/78.2 with and 55.1/78.9 w/o). We hypothesise this

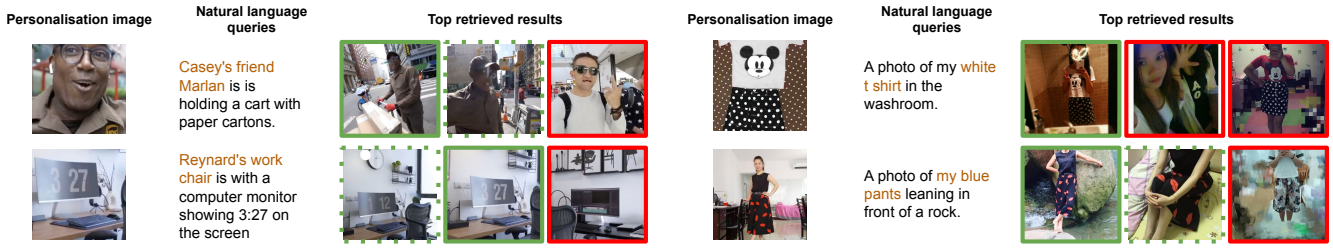


Fig. 3: A qualitative sample of contextual retrieval sorted from left to right from this-is-my [28] and DeepFashion2 [6] datasets. Green and red rectangles correspond to the correctly and incorrectly retrieved segments/images. Dotted green line shows correctly retrieved instances but in wrong setting.

TABLE I: Ablations on eval split of 'this-is-my' [28] dataset.

Method	tR@5 (max 40.3)	P@5
text – generic	12.3	56.1
text – specific	11.8	58.3
image	15.3	63.5
text + image	18.1	67.7
ours	33.7	87.2

(a) Baseline results. Results in grey denote CLIP [17] baseline.

Ablation	tR@5 (max 40.3)	P@5
Ours	33.7	87.2
w/o Reg Loss	29.1	79.1
w/o Caption Augmentation	26.0	76.3
w/o Localisation	31.9	83.5
w/o Pre-Training	27.7	77.6
w/o Init-Scheme	32.9	86.2

(b) Ablating various model components.

TABLE II: The performance of the method depends on a number of query images. Using local features reduces the amount of template images necessary. Results on the eval split of 'this-is-my' [28] dataset.

#template imgs	with localisation		no localisation	
	tR@5 (max 40.3)	P@5	tR@5 (max 40.3)	P@5
1	31.9	84.5	28.6	77.8
3	33.7	87.2	30.4	81.9
5	33.6	87.2	31.9	83.5
10	33.2	87.0	32.4	84.8

is due to the relative simplicity of the object (e.g. 'white skirt') when compared to more complex descriptions of humans or particular objects in 'this-is-my' dataset. Full results can be seen in Table IV.

D. Discussion and Limitations

We demonstrate that our model learns image-to-text mapping with less examples and achieving higher performance than all other personalization methods. One limitation shared with most previous work is that the model has to be fine-tuned to learn the token for each instance (though this only has to be done once). While our method could in theory address this issue, we reserve these experiments for future work. Furthermore, although our model is much class agnostic in comparison to previous work, there is still bias and lack of cross-domain robustness due underlying use of CLIP [20].

TABLE III: Results on the test set of 'this-is-my' dataset [28]. Included baseline use either text features or linear combination of image and text CLIP features for retrieval. 'txt' denotes queries in plain text as seen in Fig. 3. '*' denotes our reproduction of the baseline.

Method	Context		Generic	
	MRR	R@5	mAP	MRR
CLIP baseline [28] (txt)	30.8	36.7	16.6	44.2
CLIP baseline [28] (img+txt)	20.9	23.3	51.7	82.9
CLIP* [17] (img+txt)	24.3	28.9	52.4	83,4
Thisismy [28]	42	50.7	56.4	87.4
Ours	**43.1**	**52.0**	**58.4**	**88.3**

TABLE IV: Results on 'DeepFashion2' dataset [6], personalization split as defined by [3]. Note that [3] use ViT-B/32 instead of ViT-B/16. Results in grey denote CLIP [17] baseline. 'txt' denotes queries in plain text as seen in Fig. 3.

Method	Context		Generic	
	MRR	R@5	mAP	MRR
txt	21.2	23.4	9.0	17.5
img	14.5	17.6	20.9	43.9
img + txt	21.0	26.9	21.7	43.6
PALVARA [3]	28.4	39.2	-	-
this-is-my [28]	38.4	51.4	53.4	77.7
Ours	38.5	51.8	54.7	78.2

VI. Conclusion

We present a conceptually simple and effective method for learning personalised tokens in VLMs using image-to-text mapping called π-map. It is highly capable in personalised text-to-image, text-to-video, and image-to-image retrieval, outperforming all prior work on three personalization benchmarks while requiring only a few examples to fully personalise the embedding. In future work, we hope to expand on our method and develop a new, larger benchmark for personalised retrieval.

REFERENCES

[1] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.

[2] M. Alfarra, H. Itani, A. Pardo, M. Ramazanova, J. C. Perez, M. Müller, B. Ghanem *et al.*, "Evaluation of test-time adaptation under computational time constraints," in *Forty-first International Conference on Machine Learning*, 2024.

[3] N. Cohen, R. Gal, E. A. Meirom, G. Chechik, and Y. Atzmon, ""this is my unicorn, fluffy": Personalizing frozen vision-language representations," in *European Conference on Computer Vision (ECCV)*, 2022.

[4] M. Dzabraev, M. Kalashnikov, S. Komkov, and A. Petiushko, "Mdmmt: Multidomain multimodal transformer for video retrieval," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3354–3363.

[5] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, "Clip-adapter: Better vision-language models with feature adapters," *International Journal of Computer Vision*, vol. 132, no. 2, pp. 581–595, 2024.

[6] Y. Ge, R. Zhang, L. Wu, X. Wang, X. Tang, and P. Luo, "A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," *CVPR*, 2019.

[7] S. K. Gorti, N. Vouitsis, J. Ma, K. Golestan, M. Volkovs, A. Garg, and G. Yu, "X-pool: Cross-modal language-video attention for text-video retrieval," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5006–5015.

[8] G. Gu, S. Chun, W. Kim, H. Jun, Y. Kang, and S. Yun, "Compodiff: Versatile composed image retrieval with latent diffusion," *Transactions on Machine Learning Research*, 2024, expert Certification. [Online]. Available: https://openreview.net/forum?id=mKtlzW0bWc

[9] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt, "Openclip," Jul. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5143773

[10] S. Karthik, K. Roth, M. Mancini, and Z. Akata, "Vision-by-language for training-free compositional image retrieval," *International Conference on Learning Representations (ICLR)*, 2024.

[11] B. Korbar and A. Zisserman, "Personalised clip or: how to find your vacation videos," in *BMVC*, 2022.

[12] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *ICML*, 2023.

[13] S. Li, Y. Du, J. B. Tenenbaum, A. Torralba, and I. Mordatch, "Composing ensembles of pre-trained models via iterative consensus," *arXiv preprint arXiv:2210.11522*, 2022.

[14] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[15] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," *arXiv preprint arXiv:2111.09734*, 2021.

[16] D. Nukrai, R. Mokady, and A. Globerson, "Text-only training for image captioning using noise-injected clip," *arXiv preprint arXiv:2211.00575*, 2022.

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[18] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer, 2010, pp. 213–226.

[19] S. Schrodi, D. T. Hoffmann, M. Argus, V. Fischer, and T. Brox, "Two effects, one trigger: On the modality gap, object bias, and information imbalance in contrastive vision-language representation learning," *arXiv preprint arXiv:2404.07983*, 2024.

[20] A. Shtedritski, C. Rupprecht, and A. Vedaldi, "What does clip know about a red circle? visual prompt engineering for vlms," in *ICCV*, 2023.

[21] S. Sun, R. Li, P. Torr, X. Gu, and S. Li, "Clip as rnn: Segment countless visual concepts without training endeavor," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 13 171–13 182.

[22] R. Team, A. Ormazabal, C. Zheng, C. de Masson d'Autume, D. Yogatama, D. Fu, D. Ong, E. Chen, E. Lamprecht, H. Pham, I. Ong, K. Aleksiev, L. Li, M. Henderson, M. Bain, M. Artetxe, N. Relan, P. Padlewski, Q. Liu, R. Chen, S. Phua, Y. Yang, Y. Tay, Y. Wang, Z. Zhu, and Z. Xie, "Reka core, flash, and edge: A series of powerful multimodal language models," 2024. [Online]. Available: https://arxiv.org/abs/2404.12387

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] L. Ventura, A. Yang, C. Schmid, and G. Varol, "Covr: Learning composed video retrieval from web video captions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, 2024, pp. 5270–5279.

[25] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy *et al.*, "Dualprompt: Complementary prompting for rehearsal-free continual learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 631–648.

[26] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong *et al.*, "Robust fine-tuning of zero-shot models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 7959–7971.

[27] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, "Zero-shot video question answering via frozen bidirectional language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 124–141, 2022.

[28] C.-H. Yeh, B. Russell, J. Sivic, F. C. Heilbron, and S. 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*, 2023, pp. 19 123–19 132.

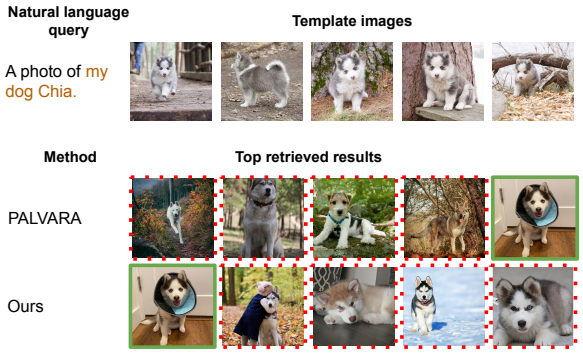
[29] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," 2023. [Online]. Available: https://arxiv.org/abs/2303.15343

[30] S. Zhao, X. Wang, L. Zhu, and Y. Yang, "Test-time adaptation with CLIP reward for zero-shot generalization in vision-language models," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=kIP0duasBb

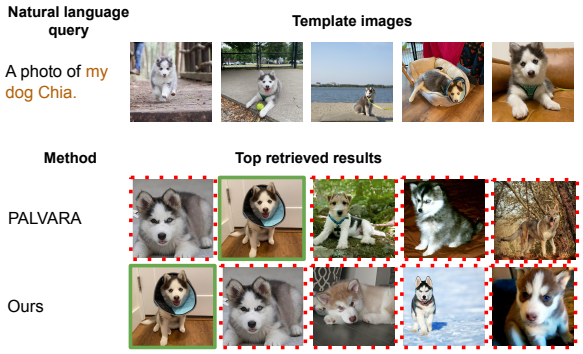
[31] Y. Zhong, R. Arandjelović, and A. Zisserman, "Faces in places: Compound query retrieval," in *BMVC*, 2016.

APPENDIX

We investigate the quantitative importance of using local features for learning personalised embeddings in Tbl. 1 and Tbl. 2 of the main paper. To demonstrate the importance qualitatively, we learn personalised embedding with 5 different images of the dog 'Chia' using our method and those of PALVARA [3]. We obtain 40 images with Google image search to use as hard negatives (prompts used: 'a dog in a forest' and an 'a small husky in a forest'), and display top-5 in Figure 4. Our method shows resilience to the type of background features, while PALVARA [3] seems to exploit additional biases (such as background) in the images.



(a)



(b)

Fig. 4: Importance of using localised features: learning personalised features for 'My dog Chia' from two different sets of template images. In *a)* all template images come from the same time and place, while in *b)*, the images are varied. Our method ranks the correct image first on both occasions, while PALVARA [3] remains sensitive to the diversity of the template images.

In the main paper, we present results only using OpenAI's CLIP (VIT-B/16) [17] in order to compare fairly with state-of-the-art methods. To demonstrate that our method can work on various CLIP variants pre-trained on different datasets in a 'plug-and-play' fashion, we use fixed setup described in the main body using personalised embeddings without image queries, and experiment with various CLIP variants provided by [9]. Our results (Table V) demonstrate that our method can be applied in a plug-and-play fashion. Most variants deviate only slightly from the baseline model (within 1.1 recall point on 'this-is-my' and 'CiA' datasets).

TABLE V: Exploration of performance using various CLIP variants. All results are computed using text-only queries on the test sets of this-is-my [28], CiA [11], and DeepFashion2 [6] datasets.

CLIP Variant	ThisIsMy Context		CiA		DeepFashion2 Context	
	MRR	R@5	R@1	R@5	MRR	R@5
VIT-B/16 [17]	42.1	50.9	64.9	81.2	38.3	51.2
ViT-B/32 [17]	41.6	50.4	64.7	81.1	38.3	51.0
ViT-L/14 [9]	42.4	51.0	65.4	81.4	38.6	51.5
ViT-H/14 [9]	42.7	51.5	65.5	81.4	38.9	52.0
ViT-SO400M/14 (siglip) [9], [29]	43.4	52.0	66.0	82.3	39.4	53.7

(a) 'this-is-my' dataset [28].

(b) DeepFashion2 dataset [6].

Fig. 5: Examples from our evaluation datasets.

Samples from the evaluation datasets are given in Fig. 5.