Improving Personalized Search with Regularized Low-Rank Parameter Updates

Supplementary Material

6. Results on ConCon-Chi TEST-UNSEEN split

In order to compare to the baselines reported in the original ConCon-Chi paper [28], we report results on the full TEST split, which contains 3 validation concepts and 17 unseen concepts. However unlike zero-shot methods like SEARLE, we use these 3 validation concepts to select the λ regularization hyperparameter. We evaluate on the TEST-UNSEEN split in Tab. 9, which excludes these validation concepts. Our results verify that our accuracy gains hold for the concepts for which λ was not tuned.

Method	Context			Concept-only	
	mRR	mAP	recall@1	mRR	mAP
SEARLE	43.88	30.73	33.49	96.67	61.94
Ours	**46.17**	**31.99**	**36.29**	**100.00**	**70.65**

Table 9. Performance on the TEST-UNSEEN split of ConCon-Chi.

7. Standard Error on DeepFashion2

We report the mean and standard error over 5 runs with different random seeds on the DeepFashion2 test set in Tab. 10 with 5 randomly selected train images for each concept per run.

8. Ablation Validation Split Results & Hyperparameters

We provide the ConCon-Chi validation split results and the value for the regularization weight hyperparameter λ for the ablations reported in the main paper: LoRA rank (Tab. 11, LoRA layers (Tab. 12), and LoRA parameters (Tab. 13). We performed our search for the value of λ resulting in convergence to the highest accuracy for each setting on the validation split. We selected our final model setting (rank=1, layers=12, parameters=V, $\lambda = 0.35$) based on the results of these ablations on the validation split.

9. Comparison to Yeh *et al.* [38]

Yeh *et al.* [38] propose a textual inversion approach for PerVL that meta-learns a per-class basis on large scale data, over which the V^* tokens for new concepts are learned as a linear combination. Both the V^* token and basis are updated at personalization time. Differently from the original PerVL setting [7], the tokens for all concepts in the dataset are learned *jointly*, with the vision-text contrastive loss using images of the other concepts as hard negatives and an additional text-text contrastive loss pushing apart the text embeddings for different concepts. We exclude their method from our main comparisons since this is a different setting than that followed by prior methods. Using the other concepts as hard negatives gives the method an advantage at retrieval time since the retrieval database is composed of images of the concepts in the dataset. For DeepFashion2 in particular, where the concepts are all clothing items and many are visually similar, using the other concepts as negatives helps the model distinguish its representation of each concept from visually similar concepts that will appear in the retrieval database.

To adapt our method to this setting where hard negatives are provided, we create an additional objective that pushes personal textual queries for the concept being learned away from the image embeddings of other concepts in CLIP space. Specifically we define a *negative loss*, $\mathcal{L}_{neg}$, as a negative MSE loss:

$$\mathcal{L}_{\text{neg}} = -\frac{1}{N_c}\sum_{i=1}^{N_c}\left(\frac{\psi'_{T,c}(q_i)}{||\psi'_{T,c}(q_i)||_2} - \frac{\psi_I(I_i^n)}{||\psi_I(I_i^n)||_2}\right)^2 \quad (9)$$

where for each iteration, $\{I_i^n\}$ consists of N_c sampled training images containing a concept that is **not** concept c. We alter Eq. 6 (main text) to be:

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{neg}} + \lambda\mathcal{L}_{\text{reg}} \quad (10)$$

Note that this training objective differs from Yeh *et al.*, which uses a set of contrastive losses between the concepts during joint training. We introduce $\mathcal{L}_{\text{neg}}$ as a means of accomodating hard negatives with minimal changes to our existing training objective and setting.

Quantitative Comparison We provide a quantitative comparison on DeepFashion2 in this setting in Tab. 14. We use the ViT-B/32 backbone for these experiments and set $\lambda_{\text{neg}} = 1$ and $\lambda_{\text{reg}} = 0.1$. Without having the other concepts as hard negatives, our method naturally has lower concept-only performance, as it does not have the advantage of hard negatives to disambiguate between similar concepts. With the addition of negatives, we achieve similar concept-only performance to Yeh *et al.*, and much higher context performance. These results demonstrate that our method better balances personal knowledge and generic knowledge than Yeh *et al.*'s textual inversion based method.

Method	Arch.	Context		Concept-only	
		mRR	recall@5	mRR	mAP
Adapter	ViT-B/32	5.9 ± 0.7	-	-	-
COLLIE [32]	ViT-B/32	7.9 ± 0.7	-	-	-
Text Only	ViT-B/32	17.6 ± 0.0	-	-	-
AvgIm + Text	ViT-B/32	18.8 ± 0.4	-	-	-
PALAVRA [7]	ViT-B/32	28.4 ± 0.7	39.2 ± 1.3	-	-
SEARLE [4]	ViT-B/32	21.90 ± 0.39	27.15 ± 0.57	25.97 ± 0.80	12.74 ± 0.48
Ours	ViT-B/32	$\mathbf{34.82 \pm 0.52}$	$\mathbf{44.88 \pm 1.17}$	$\mathbf{59.26 \pm 1.64}$	$\mathbf{28.75 \pm 0.74}$
SEARLE [4]	ViT-L/14	27.62 ± 0.26	34.12 ± 0.39	32.07 ± 0.90	16.17 ± 0.62
Ours	ViT-L/14	$\mathbf{40.72 \pm 0.27}$	$\mathbf{51.31 \pm 0.78}$	$\mathbf{65.96 \pm 0.36}$	$\mathbf{35.07 \pm 0.65}$

Table 10. Results from Tab. 1 (main text, comparison on the DeepFashion2 test set) with standard error reported over 5 runs.

LoRA rank	Reg. weight	Context (Single-Concept)			Concept-only		VLM cap
		mRR	mAP	r@1	mRR	mAP	r@10
r=2	λ=2	52.71	37.30	41.43	**100.00**	57.21	**52.61**
r=4	λ=6	52.51	37.20	**42.45**	**100.00**	**57.54**	52.50
r=8	λ=24	52.52	37.20	**42.45**	**100.00**	**57.54**	52.51
r=16	λ=100	52.62	37.34	41.45	**100.00**	57.53	52.48
r=1	λ=0.35	**52.75**	**37.82**	41.51	**100.00**	57.49	52.47

Table 11. Validation split performance and regularization weight for ablation of LoRA rank on ConCon-Chi. For each rank, we sweep over different values for λ and report the best-performing value.

Layer(s)	Reg. weight	Context (Single-Concept)			Concept-only		VLM cap
		mRR	mAP	r@1	mRR	mAP	r@10
11,12	λ=2	52.42	37.36	**42.40**	**100.00**	56.99	**52.66**
10,11,12	λ=4	52.03	37.32	41.45	**100.00**	57.46	52.56
all layers	λ=40	44.45	32.46	34.91	83.33	53.23	52.37
1	λ=1	43.39	32.68	33.96	83.33	54.19	52.21
12	λ=0.35	**52.75**	**37.82**	41.51	**100.00**	**57.49**	52.47

Table 12. Validation split performance and regularization weight for ablation of LoRA layers on ConCon-Chi. For each layer set, we sweep over different values for λ and report the best-performing value.

Param(s)	Reg. weight	Context (Single-Concept)			Concept-only		VLM cap
		mRR	mAP	r@1	mRR	mAP	r@10
Q	λ=0	23.17	15.09	13.21	38.89	8.77	52.15
K	λ=0	19.82	14.93	9.43	2.36	5.81	52.11
O	λ=100	51.22	33.69	**42.45**	83.33	51.69	52.62
Q,K,V,O	λ=500	51.14	33.86	**42.45**	83.33	51.90	**52.66**
Q,V	λ=2	**53.04**	37.76	42.40	**100.0**	56.66	52.63
MLP1	λ=50	44.01	28.45	33.96	**100.0**	48.05	51.64
MLP2	λ=200	50.57	33.12	38.68	**100.0**	49.81	51.25
final proj	λ=700	52.42	35.77	39.62	**100.0**	53.91	51.09
V	λ=0.35	52.75	**37.82**	41.51	**100.00**	**57.49**	52.47

Table 13. Validation split performance and regularization weight for ablation of LoRA parameters on ConCon-Chi. For each parameter set, we sweep over different values for λ and report the best-performing value.

Method	Context		Concept-only	
	mRR	recall@5	mRR	mAP
Yeh *et al.*	34.4 ± 0.7	45.2 ± 1.1	69.3 ± 1.8	40.0 ± 1.0
Ours	34.82 ± 0.52	44.88 ± 1.17	59.26 ± 1.64	28.75 ± 0.74
Ours + negs	$\mathbf{42.23 \pm 0.23}$	$\mathbf{52.57 \pm 0.35}$	$\mathbf{69.66 \pm 0.98}$	$\mathbf{40.65 \pm 0.59}$

Table 14. Comparison to Yeh *et al.* [38], which uses the other concepts as hard negatives during training. We include our method in the original setting (Ours), and our method adapted to also use negatives (Ours + negs). All results use the ViT-B/32 architecture and report mean and standard error over 5 runs.

# Train Imgs	Method	Context			Concept-only	
		mRR	mAP	recall@1	mRR	mAP
0	*Coarse (class name)*	24.21	16.83	14.48	-	-
	Discriminative†	43.16	30.16	31.92	-	-
	Rich†	40.58	27.65	29.98	-	-
1	PALAVRA	34.39 ± 1.68	22.56 ± 1.29	24.59 ± 1.94	-	-
	Pic2Word	37.15 ± 1.76	25.23 ± 1.20	26.35 ± 1.85	-	-
	SEARLE	41.07 ± 0.92	28.16 ± 0.55	31.16 ± 0.94	-	-
	Ours	$\mathbf{44.68 \pm 0.61}$	$\mathbf{30.99 \pm 0.48}$	$\mathbf{34.45 \pm 0.55}$	$\mathbf{98.83 \pm 1.62}$	$\mathbf{65.10 \pm 0.96}$
5	PALAVRA [7]	35.99	23.59	26.75	-	-
	Pic2Word [31]	38.62	26.39	27.68	-	-
	SEARLE [4]	43.93	30.74	33.49	**100.00**	61.68
	Ours	**46.33**	**32.33**	**36.16**	**100.00**	**68.71**

Table 15. Comparison to prior work on the ConCon-Chi benchmark, including the single training image setting. For single image training, we report the mean and standard deviation. Our approach achieves state-of-the-art results in both the 1-image and 5-image settings. † indicates oracle descriptions.

10. Single Training Image Experiments on ConCon-Chi

The original ConCon-Chi paper [28] also reports results where only a single training image is used per concept. We report results for our method in this setting in Tab. 15. We use the same hyperparameters as our main ConCon-Chi experiments where all 5 training images per concept are used. We report the mean and standard deviation over each of the 5 training images. Our method performs best in the single-image setting, and our single-image method even outperforms the other methods when they use all 5 training images. This result demonstrates the effectiveness of POLAR even with a single training image per concept.

Method	Iters	Personalization time (ms)
Text. Inv. (1 tok)	50	1597.62
Ours	50	219.54
Text. Inv. (1 tok)	500	15961.97
Ours	500	1940.34

Table 16. Total personalization time for a concept in milliseconds of our method vs. textual inversion.

11. Personalization Time Analysis

POLAR is fast to personalize and does not require pretraining. For all experiments in Section 4 (main text), we optimize for 500 iterations to ensure all variants converge; however for our main method setting (rank=1, layers=12, params=V, λ=0.35), our model converges within 50 iterations. We provide runtime analysis in Tab. 16, showing the full personalization time of our ViT-L/14-based method with 5 training images for a concept on a single NVIDIA V100 GPU. We report the personalization time for both 50 iterations and 500 iterations. Because we backpropagate only through the final layer of the text encoder, our method is significantly faster to optimize than traditional textual inversion.

12. Additional Implementation Details

DeepFashion2. We train our ViT-B/32 model for 50 iterations, and our ViT-L/14 model for 200 iterations. We use the Adam optimizer with learning rate 0.001. We use the token "sks" as V^*.

ConCon-Chi. We train our ViT-L/14 model for 500 iterations. We use the Adam optimizer with learning rate 0.001. We do not append the classname to V^*, because the classnames are less likely to be aligned with the concept. For example, several concepts have the classname "puppet" as they are animal-like objects created from household materials, but this is unlikely to align with CLIP's concept of "puppet" based on its pretraining. We use the token "sks" as V^*.

13. Evaluation of General Knowledge

VLM Captions. To generate the captions for calculating our VLM caption recall@10 metric, we prompt LLaVA-1.5-7B [22] with the image and the prompt "Caption this image in 1-2 sentences." To assess noise in the captions, we manually checked 100 of the captions, finding 88 accurate, 10 with minor errors, and 2 wrong. The metric is intended to assess the performance delta from original CLIP, so a noisy caption equally affects both methods. We choose a permissive threshold of r@10 because the ground truth is determined as the single image from which the caption is generated, but ConCon-Chi has multiple similar images. Our method performs similarly to CLIP across different thresholds, as shown in Tab. 17.

Evaluation on general retrieval task. We also evaluate

Method	r@1	r@5	r@10	r@50
Original CLIP	13.27	39.62	52.69	78.07
Ours	13.39	39.61	52.62	78.07

Table 17. Evaluation with different recall thresholds for our VLM caption metric.

retention of general knowledge by performing general image retrieval on Flick30k [39] with the parameter update for a concept applied. We report results in Tab. 18, showing parity with original CLIP.

Method	r@1	r@5	r@10
Original CLIP	67.76	89.78	94.26
Ours	68.16	89.79	94.43

Table 18. Evaluation on the Flick30k general image retrieval task.

Evaluation with ConCon-Chi discriminative captions. The ConCon-Chi dataset also includes *discriminative* descriptions for each concept, which are human-annotated text descriptions that differentiate the concepts from one another (*e.g.*, "bird sprayer puppet"). These descriptions provide an oracle baseline for the benchmark. We also evaluate retention of general knowledge by evaluating image retrieval on ConCon-Chi where each personal concept's place in the image caption annotations is replaced by the concept's discriminative description. Results are provided in Tab. 19, showing similar performance to original CLIP.

Method	r@1	r@5	r@10
Original CLIP	31.92	55.17	66.51
Ours	31.62	54.76	66.00

Table 19. Evaluation on ConCon-Chi general image retrieval using discriminative concept descriptions in captions.

14. Comparison to Weight Decay

We regularize our personalized parameter updates via the $||A_{L,c}||_2 = 1$ constraint and imposing a squared-L_2 penalty on $B_{L,c}$. This strategy is similar to weight decay, which also encourages learning small weights, but differs in two key aspects. First, weight decay is typically applied to all parameters, while we only impose a penalty on the size of $B_{L,c}$. Second, weight decay is implemented differently, directly subtracting a portion of the weights during the optimizer update. Tab. 20 compares our regularization scheme to simply using weight decay with the Adam optimizer (with a tuned value of 1e-4) and the AdamW optimizer with default hyperparameters. These results show that simply using Adam/AdamW struggles both with learning the concept (due to applying weight decay to $A_{L,c}$) and retaining general knowledge.

Method	Context (Single-Concept)			Concept-only		VLM cap
	mRR	mAP	r@1	mRR	mAP	r@10
Adam + wd	47.58	32.34	38.64	100.00	65.59	51.20
AdamW + wd	49.61	34.08	39.46	97.50	59.72	51.24
Ours	**51.64**	**36.73**	**41.77**	**100.00**	**68.71**	**52.62**

Table 20. Comparison of our regularization strategy with optimizer weight decay.

15. Generalization of Ablations to DeepFashion2

While we report our main ablations on the ConCon-Chi dataset, we observe similar trends on DeepFashion2. Tab. 21 shows ablating the parameters on which the LoRA is learned on DeepFashion2 for a single run of 5 training images. We see similar results to ConCon-Chi (Tab. 6), with the value transform performing best.

Params	Context		Concept-only	
	mRR	r@5	mRR	mAP
K	23.97	29.41	13.51	00.08
O	35.15	44.80	58.51	30.55
Q,V	36.36	47.96	60.21	32.60
Q,K,V,O	35.37	45.34	60.09	32.12
V	**41.35**	**49.32**	**65.48**	**35.02**

Table 21. Ablation of LoRA parameters on DeepFashion2.

16. Limitations

Like existing approaches in the space of personalized generation that use a fixed V^* token in place of new concepts, we experience sensitivity to the choice of V^*. Similar to prior work [14, 19, 24] we find unique single tokens to be the most effective, and we use the token for "sks" in our main experiments. We observe that selecting a V^* for which CLIP likely has a strong existing representation (*e.g.*, "dog") makes it more challenging to successfully teach the model the new personalized meaning with limited parameter updates. Future work may explore dynamically determining hyperparameters such as the rank of the LoRA update and the regularization weight for different choices of V^* to eliminate this sensitivity and allow referral to concepts in natural language without the substitution of V^*.

Additionally, by updating only the text encoder ψ_T and not the image encoder ψ_I, our performance is inherently bounded by the frozen image encoder's ability to capture distinguishing visual details. While this choice makes sense practically for our task setting (the image features for all images in the retrieval database can be precomputed by regular

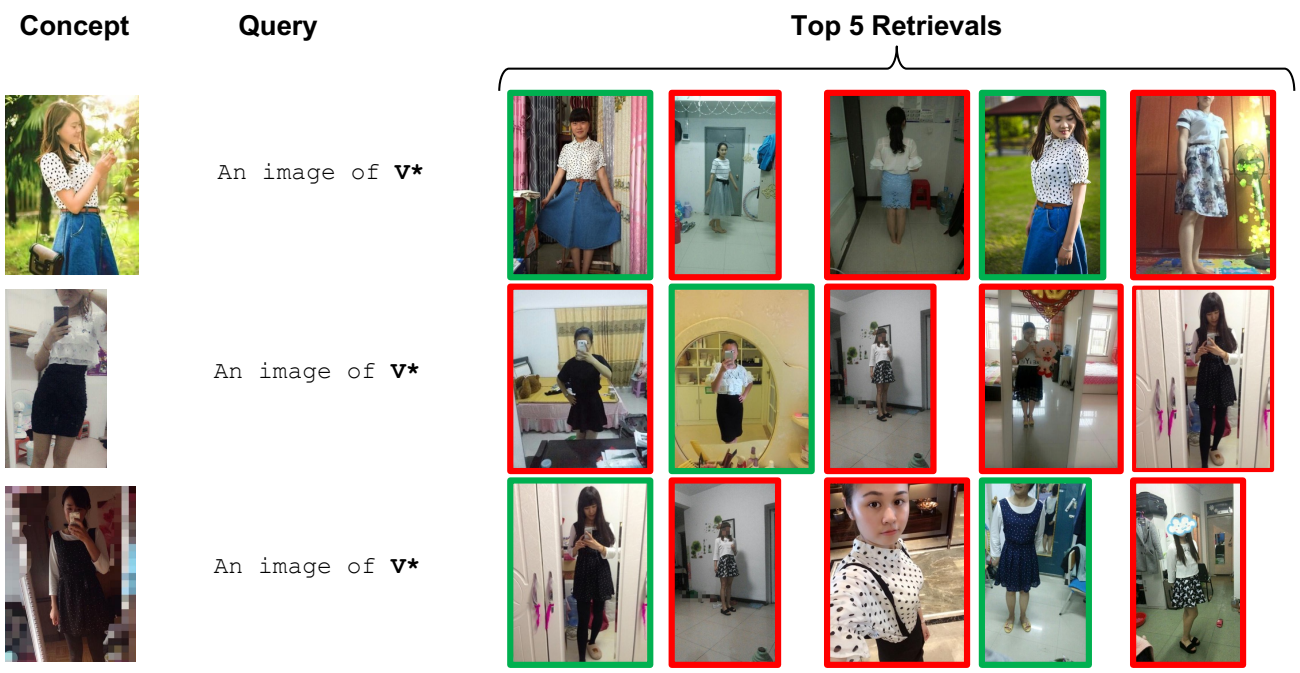

Figure 5. Our method sometimes struggles to differentiate between concepts of the same class with similar visual attributes such as color and pattern. We show concept-only queries from DeepFashion2 where such failures occur, with correct retrievals shown in green and incorrect retrievals shown in red. In row 1, the model retrieves other outfits that also have a white shirt and blue skirt, but the pattern of the shirt differs from the correct concept (*e.g.*, polka dot *vs*. striped). In row 2, the model fails to disambiguate between black skirts of different shapes. In row 3 where the concept has a black and white polka-dot pattern, the model retrieves some incorrect concepts that also have a black and white polka-dot pattern.

CLIP and then the incoming textual queries are encoded by ψ'_{T}), our approach may struggle to differentiate between visually similar concepts such as different people or objects of the same class. Some works on related tasks avoid this issue by using domain-specific specialized models such as facial feature detectors for personal concepts [1, 18]. However our focus is on minimally adapting CLIP without introducing additional domain-specific models. We show cases where our model fails to distinguish between visually-similar concepts in Fig 5.