Supplementary Material for
"PDV: Prompt Directional Vectors for Zero-shot Composed Image Retrieval"

Osman Tursun[1], Sinan Kalkan[2], Simon Denman[1] and Clinton Fookes[1]
[1]Queensland University of Technology
[2]Middle East Technical University
{osman.tursun,s.denman,c.fookes}@qut.edu.au, skalkan@metu.edu.tr

S1. Additional Experiments and Results

S1.1. Regarding Hyper-parameter Tuning

In this work, we introduce three parameters: α_I, α_T, and β. Once their semantic roles are understood, manually adjusting them becomes intuitive and straightforward:

$\alpha_T \rightarrow$	Influence of the text prompt on the semantic content of the reference image
$\alpha_I \rightarrow$	Influence of the text prompt on the visual content of the reference image
$\beta \rightarrow$	Controls trade-off between visual and semantic content

We also examined the potential for automatic tuning. However, due to several uncertain factors—such as prompt quality, baseline performance on the target dataset, and user expectations—it remains highly challenging to optimise all three parameters automatically. Nevertheless, we find that α_I can be tuned automatically to reduce the gap between PDV-I and PDV-T. In the following subsections, we discuss the manual adjustment of each parameter and the automatic tuning of α_I.

S1.1.1 Tuning α_T: Influence of Text Prompt

Among these parameters, the most important is α_T, which primarily controls the influence of the text prompt. If a user observes that the semantic changes in the top retrieved results are insufficient, α_T should be increased; conversely, if the changes are too strong, it should be decreased. For example, in the top case of Figure S1, when $\alpha_T = 1$, the skirt does not yet display clear white stripes. Increasing α_T produces results with more distinct white stripes. In contrast, in the middle and bottom examples of Figure S1, the retrieved results with $\alpha_T = 1$ are already valid. Further increasing α_T in these cases makes the semantic changes too strong, leading the method to return invalid results.

S1.1.2 Manual Tuning α_I: Influence of Text Prompt

The role of α_I is similar to that of α_T. It also controls the strength of the prompt, but in this case, the composition is with the original visual embedding $\Psi_I(I_{ref})$. When $\alpha_I = 0$, PDV-I reduces to content-based image retrieval. From the ablation results shown in Figure S9, we observe that setting $\alpha_I = 1$ is generally safe, as most methods achieve consistent improvements when α_I is increased from -0.5. Nevertheless, further increases in α_I can also be beneficial. Users should continue increasing α_I when the top retrieved results are overly similar to the reference image and fail to incorporate the semantic concepts specified in the prompt. For instance, in the top and bottom examples of Figure S2, when $\alpha_I > 1$, the top-1 retrieval results successfully present the semantic elements described in the user prompt.

S1.1.3 Tuning β: Fusion Factor

The parameter β is used in PDV-F, which fuses PDV-I and PDV-T. Its value ranges from 0 to 1. When $\beta = 1$, PDV-F is equivalent to PDV-T, and when $\beta = 0$, it reduces to PDV-I. From the ablation results shown in Figure S10, we observe that most methods achieve improved performance when β lies between 0.6 and 0.9. Beyond performance optimization, β also plays a crucial role in balancing retrieval characteristics. As illustrated in Figure S3, lower β values ($\beta < 0.5$) emphasize visual similarity, producing top results that closely resemble the reference image I_{ref} in terms of appearance. In contrast, higher β values ($\beta > 0.5$) prioritize semantic alignment, incorporating conceptual elements described in the text prompt. This provides fine-grained control over whether the retrieval system favors visual fidelity or semantic relevance.

S1.1.4 Automatically Tuning α_I

Based on the experimental results, we observe that PDV-T consistently outperforms PDV-I. If the features of PDV-I, denoted as $\Phi_{\text{PDV-I}}$, are more closely aligned with those of

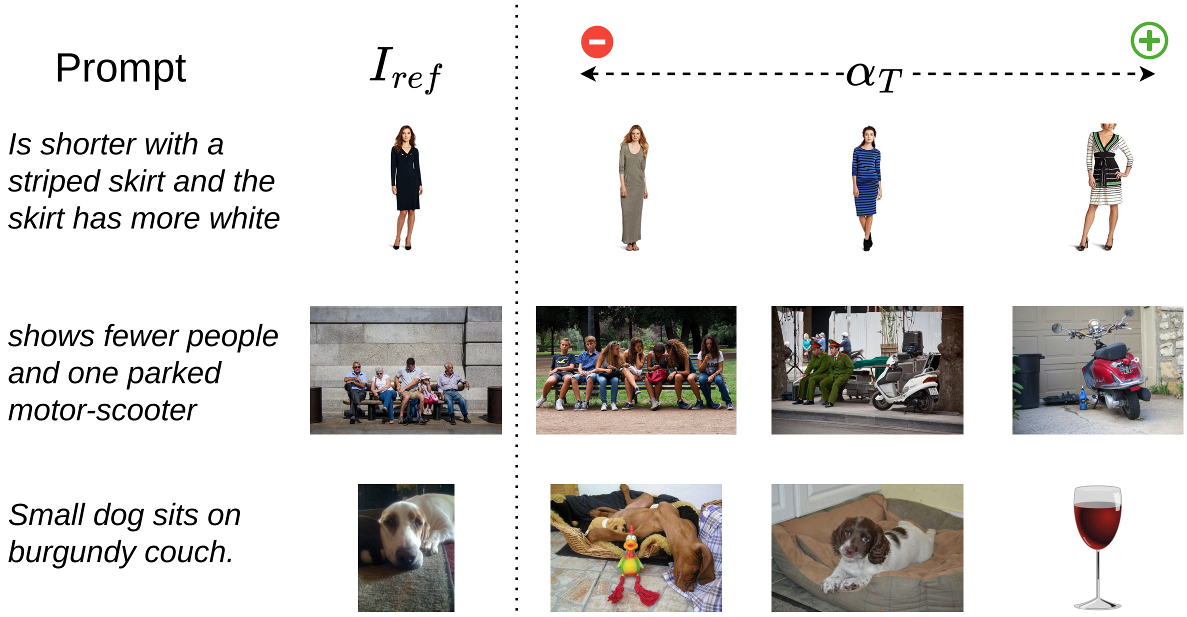

Prompt
I_{ref}
α_T
Is shorter with a striped skirt and the skirt has more white
shows fewer people and one parked motor-scooter
Small dog sits on burgundy couch.

Figure S1. Qualitative results of PDV-T showing the effect of different α_T values. For each query, we display the top-1 retrieval result for three different α_T settings. The middle result uses $\alpha_T = 1$ (baseline), the left result uses a smaller α_T value, and the right result uses a larger α_T value. All α_T values are within the range $[-0.5, 2]$.

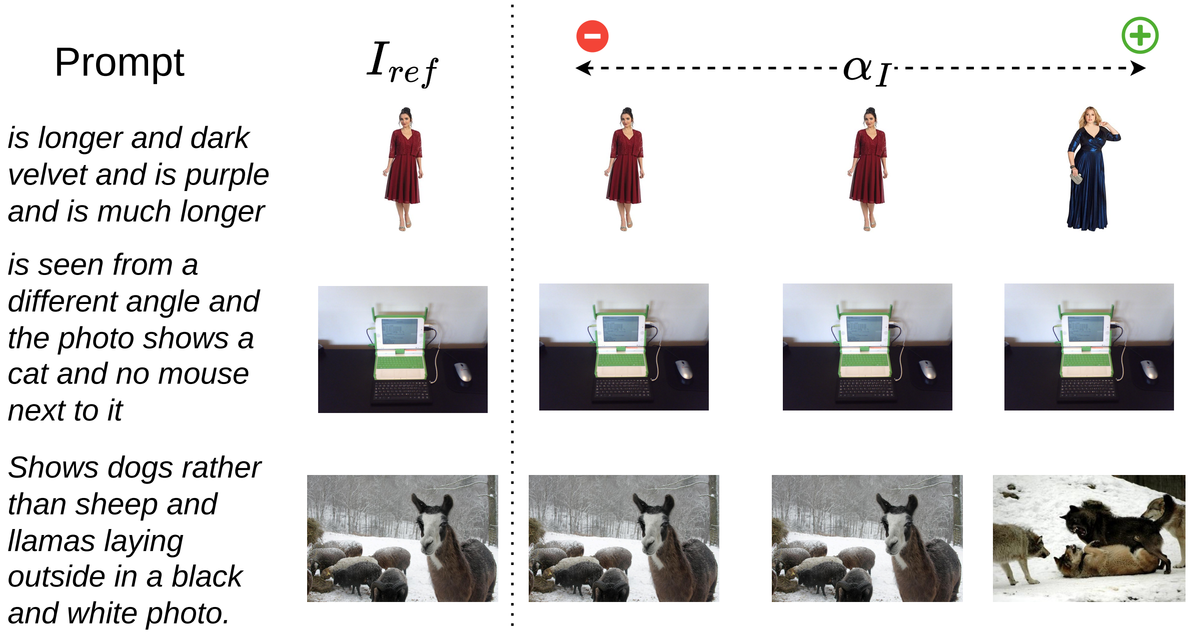

Prompt
I_{ref}
α_I
is longer and dark velvet and is purple and is much longer
is seen from a different angle and the photo shows a cat and no mouse next to it
Shows dogs rather than sheep and llamas laying outside in a black and white photo.

Figure S2. Qualitative results of PDV-I showing the effect of different α_I values. For each query, we display the top-1 retrieval result for three different α_I settings. The middle result uses $\alpha_I = 1$ (baseline), the left result uses a smaller α_I value, and the right result uses a larger α_I value. All α_I values are within the range $[-0.5, 2]$.

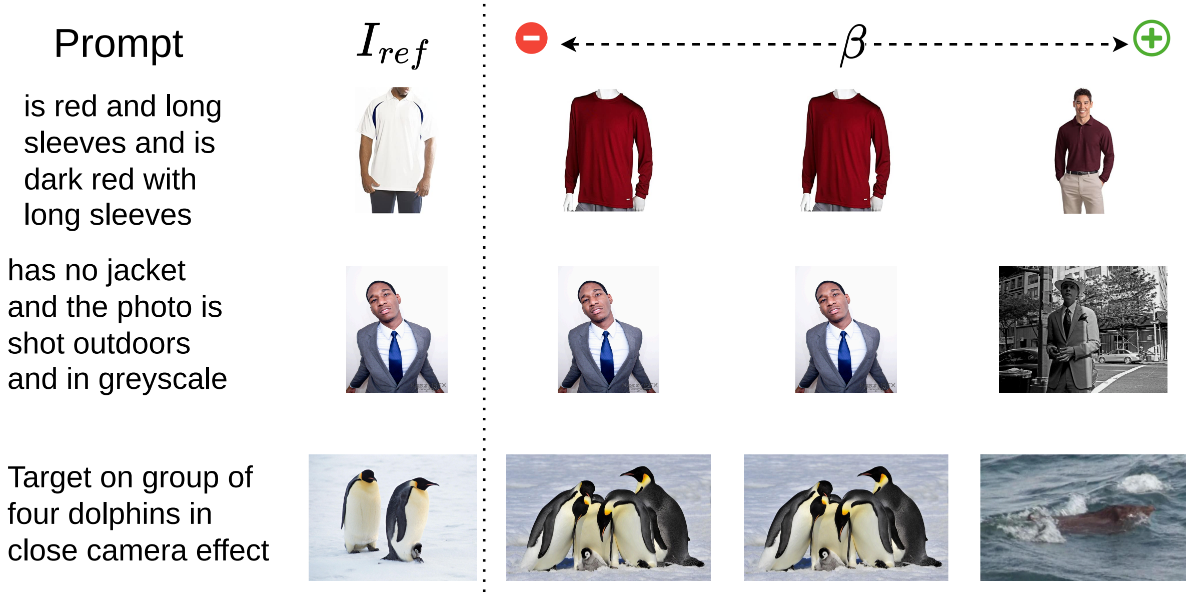

Prompt
I_{ref}
β
is red and long sleeves and is dark red with long sleeves
has no jacket and the photo is shot outdoors and in greyscale
Target on group of four dolphins in close camera effect

Figure S3. Qualitative results of PDV-F illustrating the effect of different β values. For each query, we show the top-1 retrieval result under three settings: $\beta = 0$ (left), $\beta = 0.5$ (middle), and $\beta = 1$ (right).

PDV-T, $\Phi_{\text{PDV-T}}$, the performance of PDV-I can approach that of PDV-T. Motivated by this observation, we tune the parameter α_I (while keeping $\Phi_{\text{PDV-T}}$ fixed) to minimize the ℓ_2 distance between $\Phi_{\text{PDV-I}}$ and $\Phi_{\text{PDV-T}}$, as expressed in Equation S1. To determine the optimal value of α_I, we employ the Nelder–Mead optimization method [5], which is a derivative-free and straightforward approach, making it particularly convenient to implement.

$$\alpha_I = \arg\min_{\alpha} \mathcal{L}(\Phi_{\text{PDV-T}}, \Phi_{\text{PDV-I}}(\alpha)), \qquad \text{(S1)}$$

To evaluate the effectiveness of the proposed method, we fix $\alpha_T = 1$, making PDV-T equivalent to the baseline, and compare the performance of tuned α_I against the fixed setting $\alpha_I = 1$. We conduct experiments on the FashionIQ dataset using three different methods with multiple backbone architectures. As shown in Table S1, our approach successfully determines customized α_I values for each setting. In all cases, R@50 shows consistent improvements over the baseline, with the largest gain of 23% achieved by CIReVL with the ViT-B/32 backbone on the Toptee subset. R@10 also improves steadily in most scenarios, particularly with the CIReVL method. However, for Pic2Word, R@10 decreases by 3.42% on the Dress subset.

S1.2. Efficient Retrieval with PDV

PDV is designed to enhance the retrieval performance of baseline methods in a subsequent search, triggered when an initial query fails, without incurring the high computational cost typically associated with iterative search processes.

The computational bottleneck in Zero-Shot Composed Image Retrieval (ZS-CIR) systems stems from two primary operations: feature extraction and similarity ranking.

Regarding feature extraction, the cost is dictated by the model employed. Recent ZS-CIR approaches rely on large vision-language models, whose feature extraction overhead is significant, as detailed in Table S2. In contrast, PDV generates new features for subsequent trials by building upon the embeddings from the initial retrieval. This process involves only efficient scalar multiplications and matrix additions, making its per-trial feature extraction cost nearly negligible. The only substantial computational overhead is the one-time initial calculation of the reference image embedding, $\Psi_I(I_{ref})$.

The cost of similarity ranking, on the other hand, is primarily a function of the feature dimension and the gallery size. While the feature dimension is fixed by the base model, the gallery size can be reduced for subsequent searches. To improve efficiency, we integrate a simple filtering strategy with PDV: items whose distance from the query exceeds a predefined threshold are removed from the gallery for subsequent ranking. While not unique to PDV, we believe this is the first discussion of such an optimization in a ZS-CIR context.

We evaluated this approach on the FashionIQ dataset using two baseline methods, CIReVL and Pic2Word. As shown in Table S3, for CIReVL, a threshold of 0.8 filters out over 80% of the gallery items while degrading the R@50

Backbone	**Method**	α_I	**Shirt**		**Dress**		**Toptee**	
			R@10	R@50	R@10	R@50	R@10	R@50
ViT-B/32	SEARLE	1.57/1.65/1.58	15.68%	15.62%	20.04%	20.51%	14.52%	11.18%
	CIReVL	1.92/2.24/2.02	25.65%	18.82%	24.95%	16.86%	27.81%	23.11%
ViT-L/14	CIReVL	1.90/2.16/1.95	12.96%	9.58%	15.12%	10.50%	17.70%	12.80%
	Pic2Word	1.47/1.46/1.48	0.00%	3.09%	-3.42%	2.07%	0.00%	0.74%
	SEARLE	1.67/1.82/1.73	5.84%	10.62%	16.57%	9.41%	6.15%	9.86%
ViT-G/14	CIReVL	1.50/1.63/1.53	10.43%	6.61%	19.15%	14.30%	17.51%	12.14%

Table S1. Performance differences with automatic α_I tuning compared to the fixed setting $\alpha_I = 1$ on the FashionIQ datasets. The α_I column reports the tuned values for the Shirt, Dress, and Toptee subsets.

Method	**Feature Extraction Time (Sec.)**	
	Initial	Retrial
Pic2Word	0.02	0.02
+ PDV	0.03	0.00
LinCIR	0.02	0.02
+ PDV	0.03	0.00
KEDs	0.03	0.04
CIReVL (2 Captions)	1.23	1.23
+ PDV	1.24	0.00
LDRE (20 Captions)	17.30	17.30
+ PDV	17.31	0.00

Table S2. Comparison of computation efficiency of the baseline ZS-CIR approaches on NVIDIA A100 GPU.

metric by at most 1.31%. For Pic2Word, a threshold of 0.75 filters out over 68% of the gallery with a maximum R@50 decrease of 3.73%. These results demonstrate that PDV, combined with gallery filtering, offers a highly effective trade-off, significantly accelerating retrieval speed while maintaining competitive accuracy.

S1.3. ϕ Angles of the Baselines

In Section 3.3 of the main paper, we discussed that PDV's performance is highly correlated with baseline performance and provided theoretical justification through simulation results as shown in Figure S4. We have introduced a new parameter ϕ, which is the angle between the calculated prompt directional vector Δ_{PDV} and the ground truth prompt directional vector Δ_{GT}. When ϕ is small, adjusting the parameter α can effectively reduce θ which is the angle between the target embedding vector $\Psi_I(I_{target})$ and the composed embedding vector $\Psi_T(\mathcal{F}(I_{ref}, P))$.

Here, we present the actual ϕ values for three baseline methods—CIReVL, Pic2Word, and SEARLE—using different backbones across three subdatasets of the FashionIQ dataset. Figure S5 presents the ϕ values for three baseline methods—CIReVL, Pic2Word, and SEARLE—across the FashionIQ subdatasets. While we observed the expected trend of stronger models exhibiting smaller ϕ values, we were surprised to find that the state-of-the-art CIReVL baseline maintains a large ϕ angle of approximately $65°$. This finding suggests that PDV has not yet been evaluated with optimal baseline models, despite already yielding considerable gains with CIReVL. We therefore anticipate that PDV will be even more effective when paired with future baselines that achieve a smaller ϕ (ideally $< 60°$), the regime where, according to Figure S4b, our method is most impactful.

S1.4. Additional Quantitative Results

In this supplementary section, we present additional quantitative results that were omitted from the main paper due to space constraints.

S1.4.1 Ablation Analysis

While Figure 3 in the main paper illustrates the effects of scaling factor α and fusion factor β on Recall@5 performance across various PDV applications, Figures S8, S9, and S10 present complementary results for Recall@10 and Recall@50 metrics.

The Recall@10 and Recall@50 results demonstrate consistent trends with the Recall@5 findings presented in the main paper, thus validating our conclusions across multiple evaluation metrics.

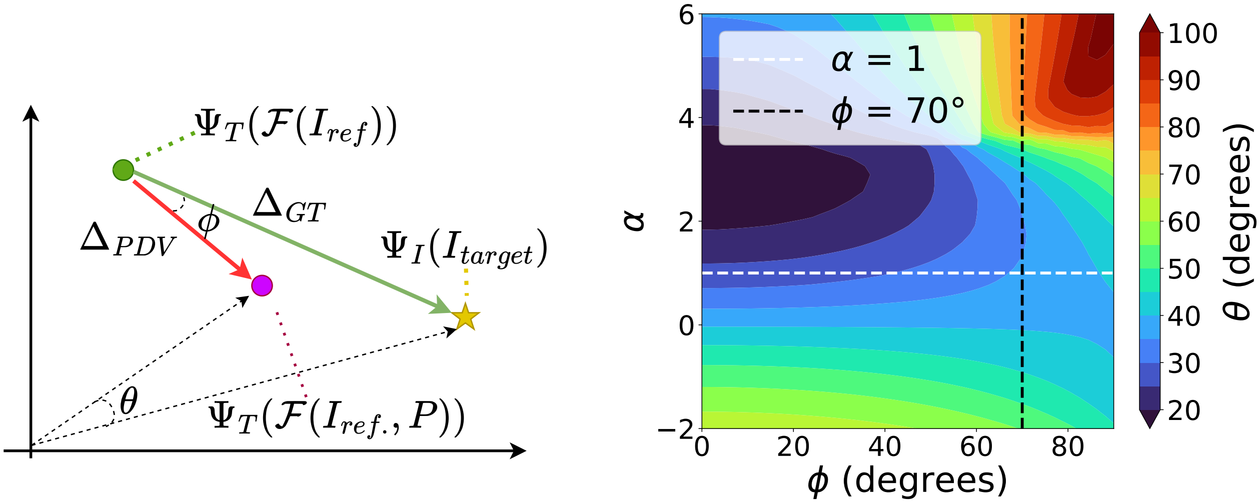
S1.4.2 PDV-I Results

We also provide additional PDV-I results achieved on the validation set of the FashionIQ dataset, as shown in Tables S5 and S6. PDV-I also achieved significant improvements over existing approaches that directly leverage image embeddings for retrieval.

Lastly, we provide a detailed visualization of the impact of α/β scaling on top-5 retrieval results. Figure S7 illustrates the performance of CIReVL with the ViT-B-32 CLIP model across three different datasets.

Backbone	Method	Dataset	Threshold	Filtered Ratio	Change in R@10	Change in R@50
ViT-B/32	CIReVL + PDV-F	Toptee	0.8	90.85%	-2.88%	-1.31%
		Dress	0.8	82.31%	0.00%	-0.73%
		Shirt	0.8	87.96%	-1.22%	-1.31%
ViT-L/14	Pic2Word + PDV-F	Toptee	0.75	85.62%	-1.24%	-0.94%
		Dress	0.75	68.79%	0.00%	0.12%
		Shirt	0.75	88.36%	-1.91%	-3.73%

Table S3. Performance changes of PDV-F after applying filtering to the initial retrieval results on the FashionIQ dataset.



(a) Visualisation of θ and ϕ.

(b) θ vs. α and ϕ

Figure S4. A visualization of how scaling α affects the similarity between the composed embedding and the target embedding.

S1.4.3 Zero-shot methods with PDV Vs. Supervised Methods

We compared state-of-the-art zero-shot composed image retrieval (ZS-CIR) methods enhanced with PDV against supervised methods on the FashionIQ and CIRR datasets. Our evaluation included early supervised methods such as TIRG [7] and ARTEMIS citedelmas2022artemis, as well as recent state-of-the-art approaches like CCIN [6] and SPRC [1]. The comparative results presented in Table S4 demonstrate that PDV achieves remarkably competitive performance against supervised methods. The PDV-based approaches significantly outperform early supervised baselines, with substantial improvements over TIRG (41.90% vs 14.13% R@10 on FashionIQ Dress) and ARTEMIS [3] (41.90% vs 25.68%). While PDV methods do not quite match the performance of recent state-of-the-art approaches like CCIN and SPRC, the performance gap remains relatively modest—typically within 7-9 percentage points on FashionIQ and approximately 8-9 points on CIRR's mean score (72.85% vs 81.66% for CCIN). This narrow performance gap is particularly noteworthy given that ZS-CIR methods with PDV operate without human-annotated training data. These results suggest that unsupervised approaches are reaching a level of effectiveness that positions them as viable alternatives to supervised methods.

S2. PDV Algorithm and Code

The PDV algorithm is given in Algorithm 1, and the code is shown in Figure S6. The implementation of PDV is very intuitive, and it could be easily integrated with any ZS-CIR approaches.

Algorithm 1 Calculate PDV Features

```
function CalculatePDVFeatures($\mathbf{f}_{\text{text}}$, $\mathbf{f}_{\text{text\_composed}}$, $\mathbf{f}_{\text{image}}$, $\alpha_i$, $\alpha_t$, $\beta$)
    $\mathbf{f}_{\text{text}} \leftarrow \text{normalize}(\mathbf{f}_{\text{text}})$
    $\mathbf{f}_{\text{text\_composed}} \leftarrow \text{normalize}(\mathbf{f}_{\text{text\_composed}})$
    $\mathbf{f}_{\text{image}} \leftarrow \text{normalize}(\mathbf{f}_{\text{image}})$
    $\mathbf{pdv} \leftarrow \mathbf{f}_{\text{text\_composed}} - \mathbf{f}_{\text{text}}$
    $\mathbf{f}_{\text{PDVI}} \leftarrow \mathbf{f}_{\text{image}} + \alpha_i \cdot \mathbf{pdv}$
    $\mathbf{f}_{\text{PDVT}} \leftarrow \mathbf{f}_{\text{text}} + \alpha_t \cdot \mathbf{pdv}$
    $\mathbf{f}_{\text{PDVF}} \leftarrow (1-\beta) \cdot \mathbf{f}_{\text{PDVI}} + \beta \cdot \mathbf{f}_{\text{PDVT}}$
    return $\text{normalize}(\mathbf{f}_{\text{PDVF}})$
end function
```

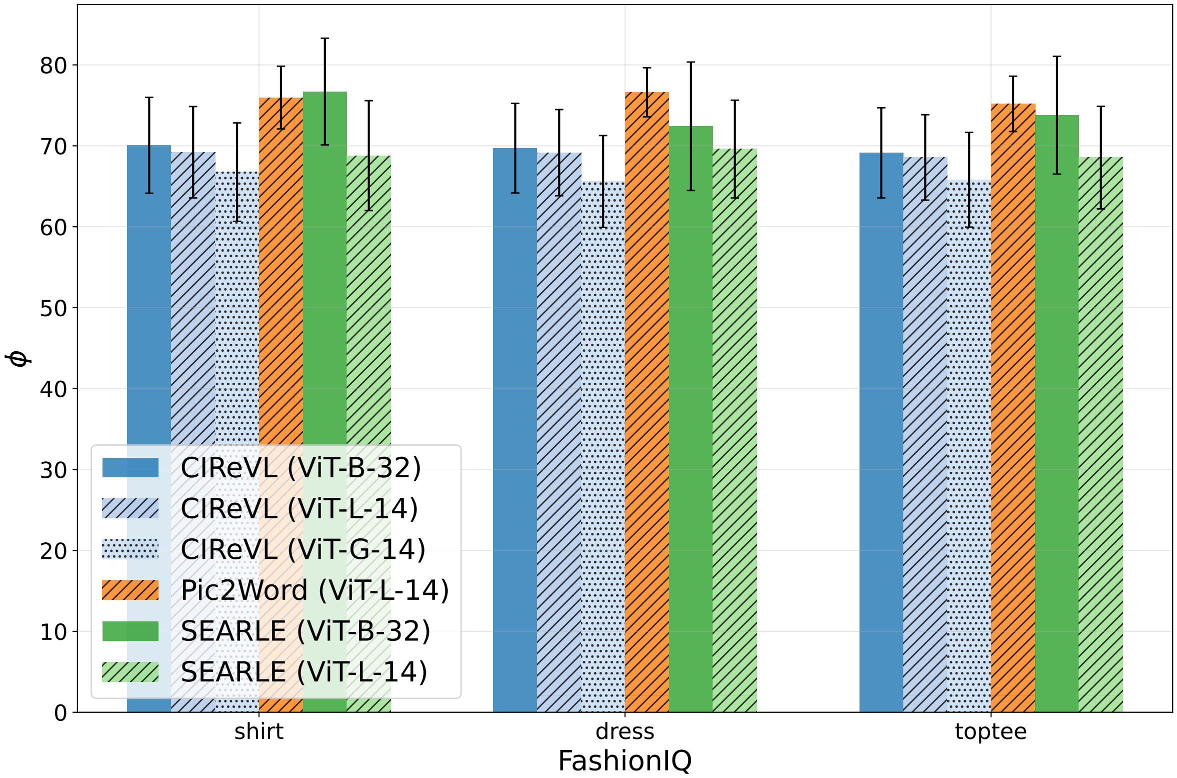


Figure S5. Performance Comparison: ϕ Angles by Method and Model Across FashionIQ Datasets

Method	FashionIQ							CIRR				
	Dress		Shirt		Toptee		Mean					
	R@10	R@50	R@10	R@50	R@10	R@50		R@1	R@5	R@10	R@50	Mean
TIRG [7]	14.13	34.61	13.10	30.91	14.79	34.37	23.66	14.61	48.37	64.08	90.03	54.27
ARTEMIS [3]	25.68	51.05	21.57	44.13	25.89	55.06	37.68	16.96	46.10	61.31	87.73	53.03
CLIP4CIR [2]	33.81	59.40	39.99	60.45	41.41	65.37	50.03	38.53	69.98	81.86	95.93	71.58
CompoDiff [4]	40.65	57.14	36.87	57.39	43.93	61.17	49.53	22.35	54.36	73.41	91.77	60.47
TG-CIR [8]	45.22	69.66	52.60	72.52	56.14	77.10	58.05	45.25	78.29	87.16	97.30	77.00
SPRC [1]	48.83	72.09	53.83	74.14	**58.13**	**78.58**	64.27	51.96	82.12	89.74	97.69	80.37
CCIN [6]	**49.38**	**72.58**	**55.93**	**74.14**	57.93	77.56	**64.59**	**53.41**	**84.05**	**91.17**	**98.00**	**81.66**
CIReVL + PDV	41.90	58.19	40.70	62.82	48.09	67.77	53.25	38.15	67.93	77.90	92.77	69.19
LDRE + PDV	-	-	-	-	-	-	-	42.51	72.22	81.71	94.94	72.85

Table S4. Comparison of PDV with supervised methods on FashionIQ and CIRR datasets.

References

[1] Yang Bai, Xinxing Xu, Yong Liu, Salman Khan, Fahad Khan, Wangmeng Zuo, Rick Siow Mong Goh, and Chun-Mei Feng. Sentence-level prompts benefit composed image retrieval. *arXiv preprint arXiv:2310.05473*, 2023. 5, 6

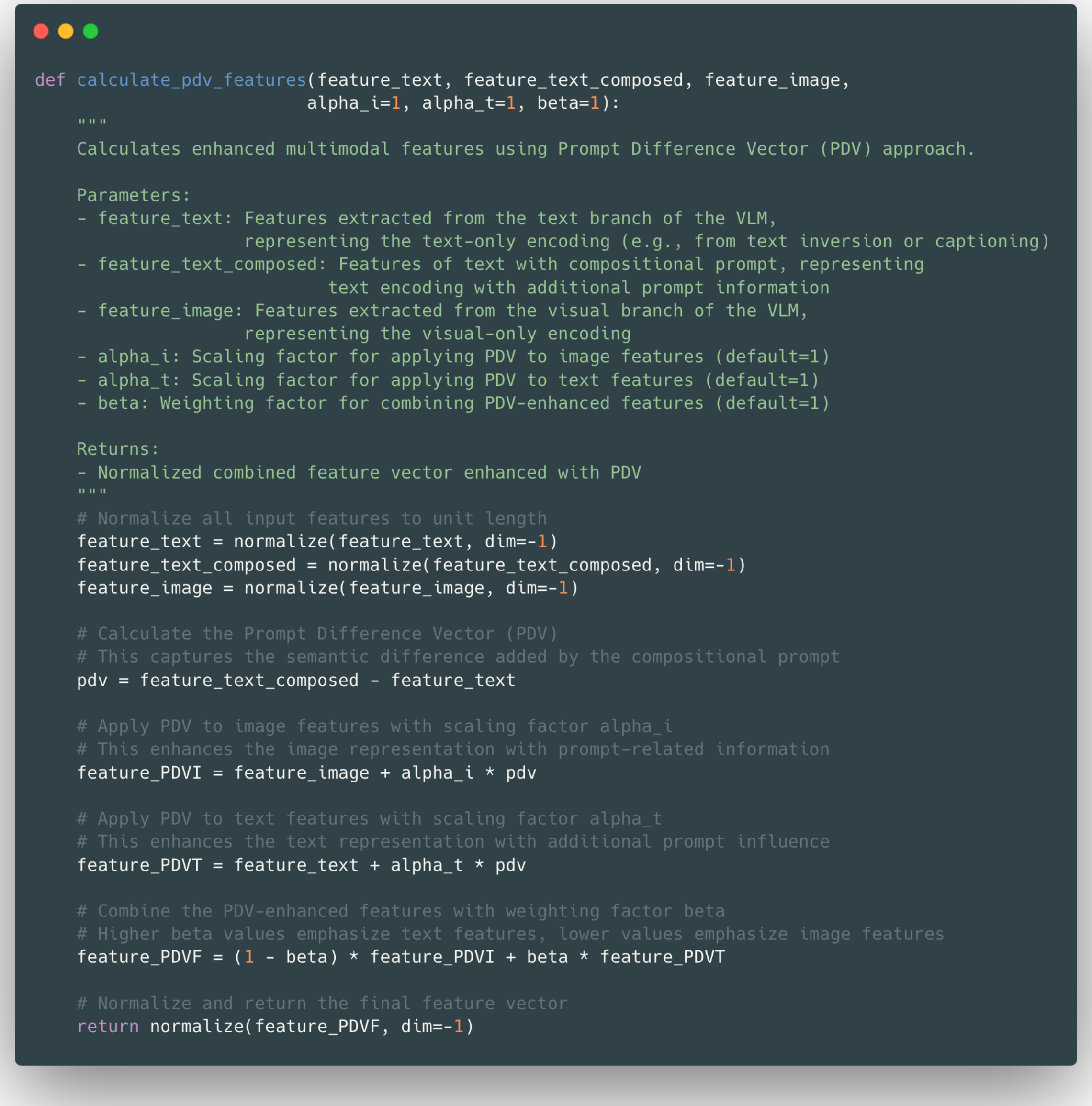
[2] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Conditioned and composed image retrieval combining and partially fine-tuning clip-based features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4959–4968, June 2022. 6

[3] Ginger Delmas, Rafael Sampaio de Rezende, Gabriela Csurka, and Diane Larlus. Artemis: Attention-based retrieval with text-explicit matching and implicit similarity. *arXiv preprint arXiv:2203.08101*, 2022. 5, 6

[4] Geonmo Gu, Sanghyuk Chun, Wonjae Kim, HeeJae Jun, Yoohoon Kang, and Sangdoo Yun. Compodiff: Versatile composed image retrieval with latent diffusion. *arXiv preprint arXiv:2303.11916*, 2023. 6

[5] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965. 3

[6] Likai Tian, Jian Zhao, Zechao Hu, Zhengwei Yang, Hao Li, Lei Jin, Zheng Wang, and Xuelong Li. Ccin: Compositional conflict identification and neutralization for composed image retrieval. In *Proceedings of the IEEE/CVF Conference*



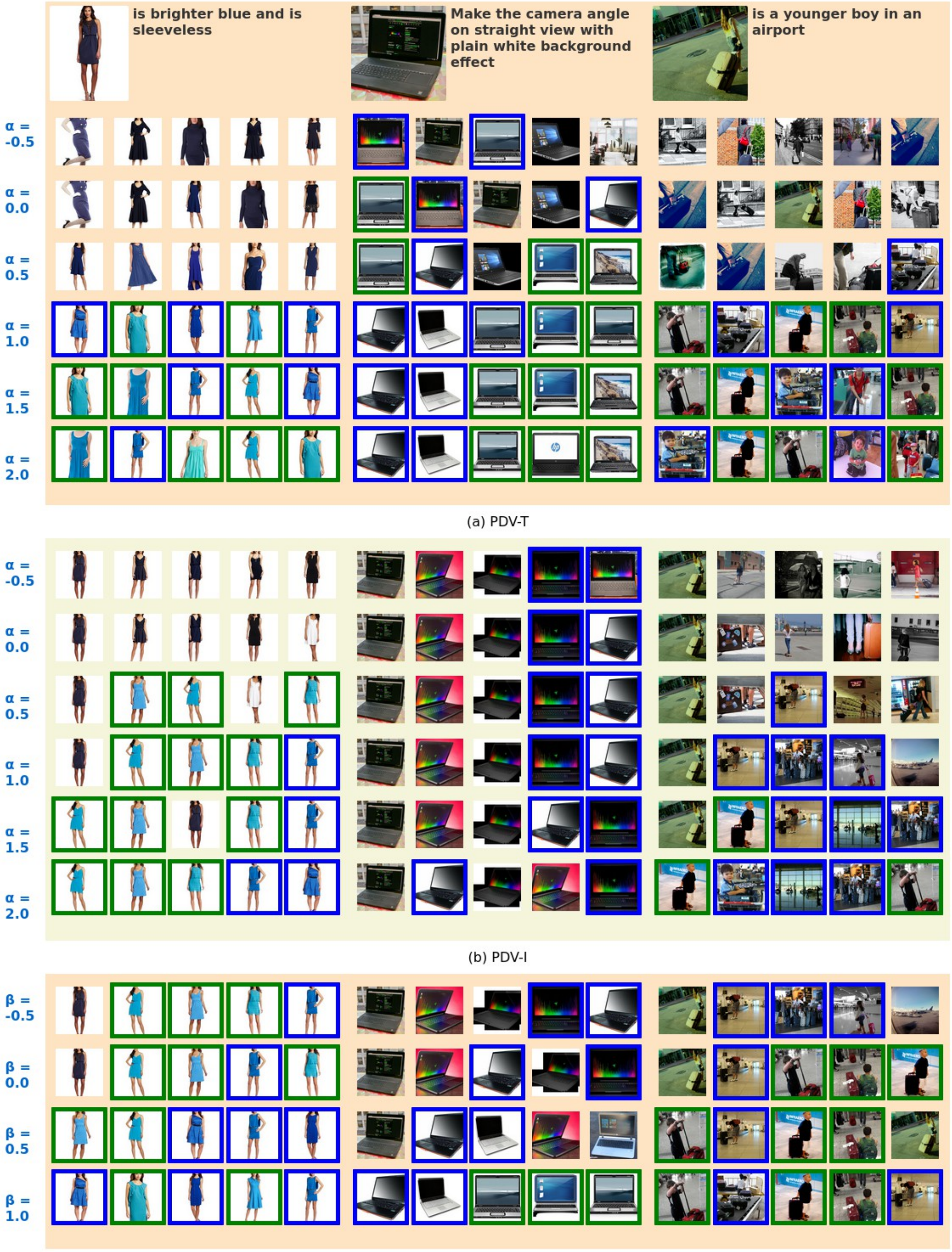
```
def calculate_pdv_features(feature_text, feature_text_composed, feature_image,
                          alpha_i=1, alpha_t=1, beta=1):
    """
    Calculates enhanced multimodal features using Prompt Difference Vector (PDV) approach.

    Parameters:
    - feature_text: Features extracted from the text branch of the VLM,
                    representing the text-only encoding (e.g., from text inversion or captioning)
    - feature_text_composed: Features of text with compositional prompt, representing
                            text encoding with additional prompt information
    - feature_image: Features extracted from the visual branch of the VLM,
                    representing the visual-only encoding
    - alpha_i: Scaling factor for applying PDV to image features (default=1)
    - alpha_t: Scaling factor for applying PDV to text features (default=1)
    - beta: Weighting factor for combining PDV-enhanced features (default=1)

    Returns:
    - Normalized combined feature vector enhanced with PDV
    """
    # Normalize all input features to unit length
    feature_text = normalize(feature_text, dim=-1)
    feature_text_composed = normalize(feature_text_composed, dim=-1)
    feature_image = normalize(feature_image, dim=-1)

    # Calculate the Prompt Difference Vector (PDV)
    # This captures the semantic difference added by the compositional prompt
    pdv = feature_text_composed - feature_text

    # Apply PDV to image features with scaling factor alpha_i
    # This enhances the image representation with prompt-related information
    feature_PDVI = feature_image + alpha_i * pdv

    # Apply PDV to text features with scaling factor alpha_t
    # This enhances the text representation with additional prompt influence
    feature_PDVT = feature_text + alpha_t * pdv

    # Combine the PDV-enhanced features with weighting factor beta
    # Higher beta values emphasize text features, lower values emphasize image features
    feature_PDVF = (1 - beta) * feature_PDVI + beta * feature_PDVT

    # Normalize and return the final feature vector
    return normalize(feature_PDVF, dim=-1)
```

Figure S6. Python function for calculating PDV features.

Fashion-IQ			Shirt		Dress		Toptee		Average	
Backbone	Method	α_I	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50
ViT-B/32	Image-only †	-	6.92	14.23	4.46	12.19	6.32	13.77	5.90	13.37
	Text-only †	-	19.87	34.99	15.42	35.05	20.81	40.49	18.70	36.84
	Image + Text †	-	13.44	26.25	13.83	30.88	17.08	31.67	14.78	29.60
	SEARLE + **PDV-I**	2	18.25	31.84	18.49	39.17	21.32	37.74	19.35	36.25
	CIReVL + **PDV-I**	2	28.95	45.88	29.00	49.13	34.22	56.09	30.72	50.37

Table S5. PDV-I performance on FashionIQ val datasets. † denotes that numbers are taken from the original paper.

on Computer Vision and Pattern Recognition (CVPR), pages 3974–3983, June 2025. 5, 6


is brighter blue and is sleeveless
Make the camera angle on straight view with plain white background effect
is a younger boy in an airport
α = -0.5
α = 0.0
α = 0.5
α = 1.0
α = 1.5
α = 2.0
β = -0.5
β = 0.0
β = 0.5
β = 1.0

(a) PDV-T

(b) PDV-I

(C) PDV-F

Figure S7. Visualisation of the impact of α/β scaling on top-5 retrieval results. CIReVL with ViT-B-32 Clip model is the baseline method used. Representative examples with prompts from three datasets: FashionIQ (left), CIRR (middle), and CIRCO (right) are shown at the top. **Green** and **blue** bounding boxes indicate true positives and near-true positives, respectively.

[7] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recog-*

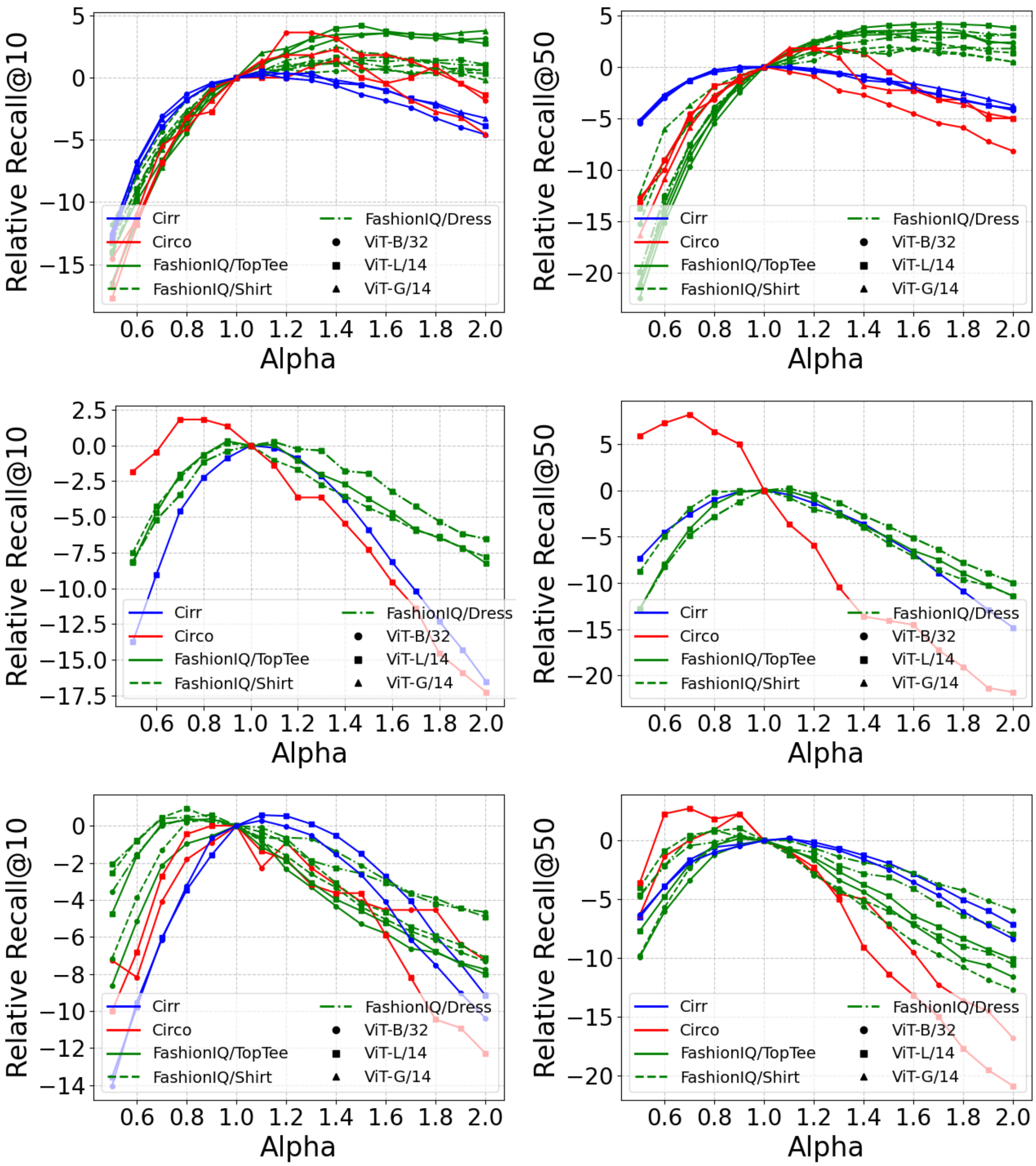


Figure S8. PDV-T: Impact of α scaling on Recall@10 (left) and Recall@50 (right) performance. Results shown for three baseline methods: CIReVL (top), Pic2Word (middle) and SEARLE (bottom).

nition, pages 6439–6448, 2019. 5, 6

[8] Haokun Wen, Xian Zhang, Xuemeng Song, Yinwei Wei, and Liqiang Nie. Target-guided composed image retrieval. In *Proceedings of the 31st ACM international conference on multimedia*, pages 915–923, 2023. 6

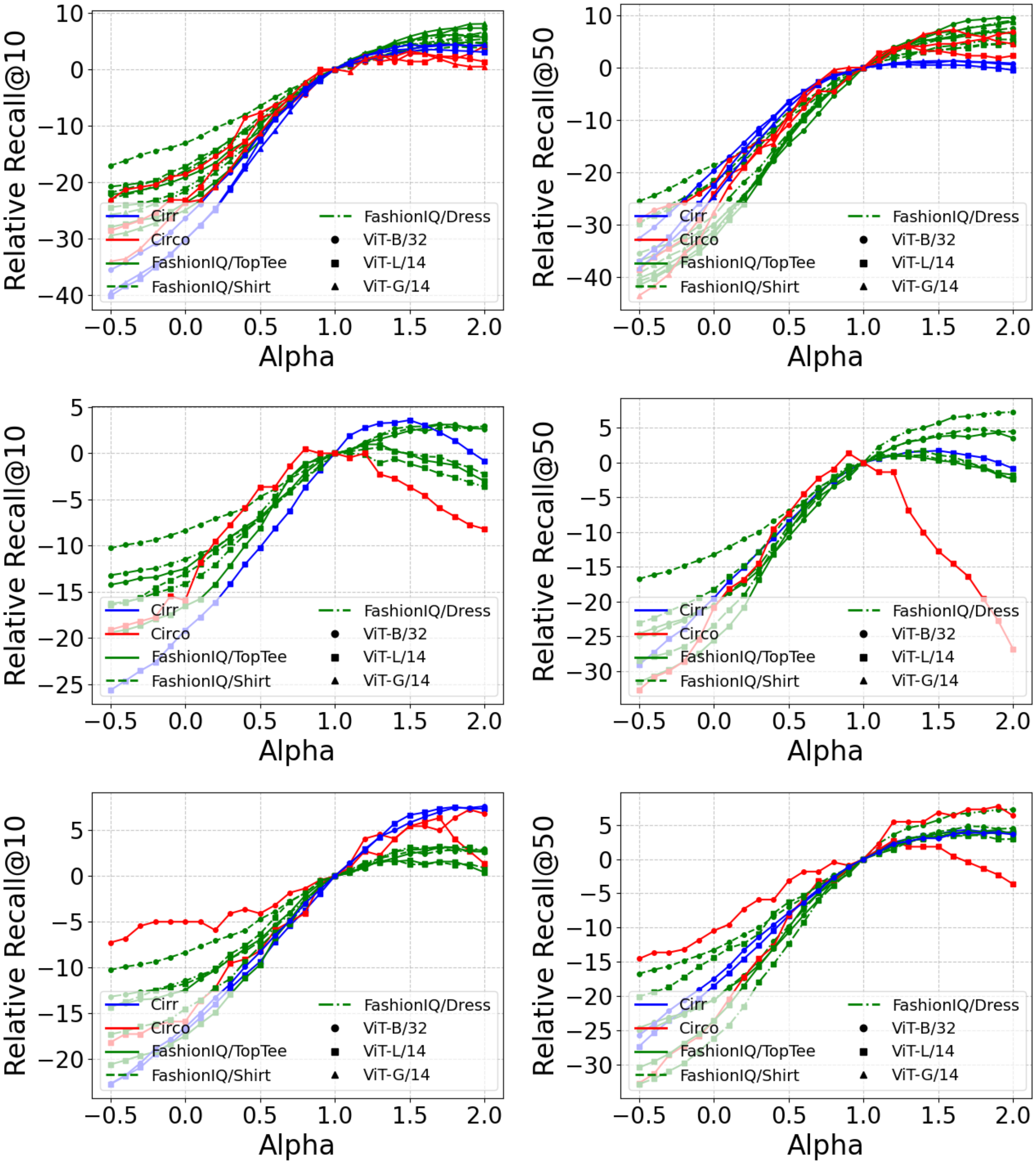


Figure S9. PDV-I: Impact of α scaling on Recall@10 (left) and Recall@50 (right) performance. Results shown for three baseline methods: CIReVL (top), Pic2Word (middle) and SEARLE (bottom).

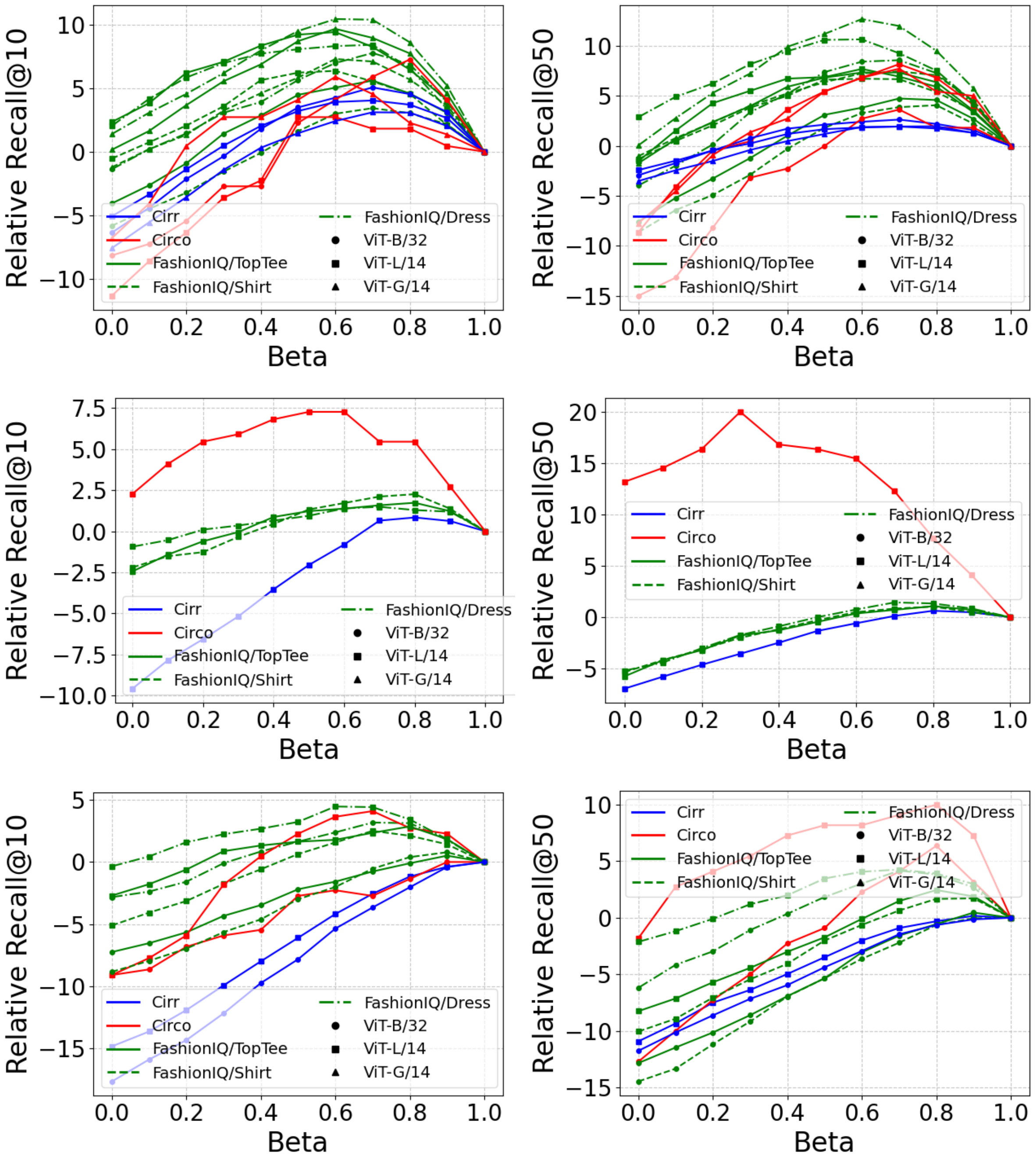


Figure S10. PDV-F: Impact of β scaling on Recall@10 (left) and Recall@50 (right) performance. Results shown for three baseline methods: CIReVL (top), Pic2Word (middle) and SEARLE (bottom).

Dataset			**CIRCO**				**CIRR**						
	Metric		mAP@k				Recall@k				R_s@k		
Arch	Method	α_I	k=5	k=10	k=25	k=50	k=1	k=5	k=10	k=50	k=1	k=2	k=3
ViT-B/32	Image-only †	-	1.34	1.60	2.12	2.41	6.89	22.99	33.68	59.23	21.04	41.04	60.31
	Text-only †	-	2.56	2.67	2.98	3.18	21.81	45.22	57.42	81.01	62.24	81.13	90.70
	Image + Text †	-	2.65	3.25	4.14	4.54	11.71	35.06	48.94	77.49	32.77	56.89	74.96
	SEARLE + **PDV-I**	1.5	4.77	5.23	6.31	6.82	16.65	42.53	55.16	81.42	44.68	67.78	82.94
	CIReVL + **PDV-I**	2.0	**10.29**	**10.80**	**12.23**	**12.93**	**27.18**	**56.53**	**67.76**	**87.64**	**59.81**	**79.59**	**90.15**
	LDRE + **PDV-I**	2.0	8.00	8.88	10.06	10.72	23.37	51.21	63.69	85.57	55.57	76.63	88.15

Table S6. PDV-I performance on CIRCO and CIRR test datasets. Note that the image-only approach utilizes the visual embedding of the reference image, whereas the text-only approach employs the text embedding of the prompt. **Bold** = best results, underline = second best. †Numbers from original paper.