STiTch: Semantic Transition and Transportation in Collaboration for Training-Free Zero-Shot Composed Image Retrieval

Supplementary Material

Appendix Overview

The supplementary material is organized into the following sections:

A. Algorithm of STiTch Process

We summarize the detailed inference algorithm of STiTch in Alg. 1.

Algorithm 1: Inference algorithm of STiTch.

Input: reference image x, text modification m, target image database $\mathrm{Y} = \{y_n\}_{n=1}^{N}$, a pre-trained CLIP model, and a pre-trained MLLMs. The number of query times K, and the number of image augmentations M.
Output: The retrieval score $p(y|x, m)$ over all target images.
Querying: Complete the input prompts with x and m, and query MLLM K times to collect the descriptions P_t from Eq. 2.
Transition: Calculate $\Delta \boldsymbol{m}$ in Eq. 3 by feeding m into CLIP text encoder, and then obtain the transferred P_t from Eq. 4.
Alignment: Collect Q_y in Eq. 5 by augmenting target image y_n for $M-1$ times, and then calculate $\mathcal{L}_{P_t,Q_{y_n}}$ from Eq. 6. image y_n in Y Calculate $\mathcal{L}_{P_t,Q_{y_n}}$ according to **Alignment** step.
Return Calculate the retrieval score from Eq. 9.

B. Additional Comparative Results.

We in this section included more comprehensive comparisons with more methods across various architectures on all datasets presented in Tab.6, Tab.7, and Tab.8. It should be noted that in Tab.7, the notation (*) indicates that we reproduced the experiments using the OpenAI weights, and the (†) indicates that we reproduced the experiments using the OpenCLIP weights, respectively. From these comparisons, our approach outperforms all the baselines in most cases, showing the efficiency of STiTch's three operations.

C. Impacts of Different MLLMs

Like previous works that employ MLLMs to analyze multimodal inputs and generate target descriptions, we specify Qwen2-VL-7B as the MLLM in earlier experiments. Here, we further explore the performance of STiTch with different MLLMs. Specifically, we report the results on Qwen2-VL-2B, Qwen2-VL-7B, LLaVA-Next-7B, and GPT-4o(mini) in Tab. 12. The results show that our STiTch can be applied to MLLMs with different architectures and that the performance improves as the number of MLLM's parameters increases. This demonstrates the potential of STiTch in flexibility and scalability, as it serves as a plug-and-play pipeline that can seamlessly integrate with various MLLMs. Indeed, we observe that different MLLMs can lead to variations in the generated captions and thus impact retrieval results. This observation further supports our core motivation: rather than re-training or fine-tuning the large models, we aim to design a framework that maximizes retrieval effectiveness given any off-the-shelf MLLM.

In addition to Tab.4 that ablates each module on Qwen-7B, we also report the results with another MLLM GPT-4o(mini) in Tab.11. The ablations on two MLLMs can show the real efficiency of STiTch's modules: (1) *Strategic Synergy Over Raw MLLM Power*: The highest mAP@k values (e.g., 38.93 @k=5, 44.46 @k=50) occur when both Transition and Transportation are enabled. This indicates that STiTch's strength lies in its systematic collaboration of strategies rather than relying solely on MLLM capabilities. Even with the same MLLM (e.g., GPT-4o(mini)), disabling either strategy reduces performance (e.g., Transportation only yields 36.61 @k=5; Transition only yields 37.50 @k=5), confirming that STiTch actively improves task-specific reasoning. (2) *Modular Adaptability*: The results implies STiTch's strategies are architecture-agnostic. While the choice of MLLM impacts absolute performance, the framework's relative gains from Transition+Transportation collaboration remain consistent.

D. Additional Ablation Experiments

D.1. Impacts of the bidirectional distance.

To conduct a more comprehensive analysis of the impacts of the bidirectional distance, we supplemented experiments with STiTch under different backbones using CT distance and OT distance as alignment strategy in Tab.9 and Tab.10. The results show that CT outperforms OT, highlighting the advantages of bidirectional fine-grained alignment.

Table 6. Performance comparison on CIRCO and CIRR datasets. Both ViT-B and ViT-L are loaded from OpenAI official weights, while ViT-G is loaded from OpenCLIP.

CIRCO + CIRR →			**CIRCO**				**CIRR**					
	Metric		mAP@k				Recall@k			$\text{Recall}_{\text{Subset}}$@k		
Arch	Method	Train	k=5	k=10	k=25	k=50	k=1	k=5	k=10	k=1	k=2	k=3
ViT-B/32	PALAVRA	✓	4.61	5.32	6.33	6.80	16.62	43.49	58.51	41.61	65.30	80.94
	SEARLE	✓	9.35	9.94	11.13	11.84	24.00	53.42	66.82	54.89	76.60	88.19
	CIReVL	✗	14.94	15.42	17.00	17.82	23.94	52.51	66.00	60.17	80.05	90.19
	LDRE	✗	17.96	18.32	20.21	21.11	25.69	55.13	69.04	60.53	80.65	90.70
	OSrCIR	✗	18.04	19.17	20.94	21.85	25.42	54.54	68.19	62.31	80.86	91.13
	SEIZE	✗	19.04	19.64	21.55	22.49	**27.47**	**57.42**	70.17	65.59	**84.48**	92.77
	STiTch(Ours)	✗	**20.26**	**21.01**	**23.01**	**24.04**	25.83	55.25	**70.20**	**65.64**	83.60	**92.80**
ViT-L/14	Pic2Word	✓	8.72	9.51	10.64	11.29	23.90	51.70	65.30	53.76	74.46	87.08
	SEARLE	✓	11.68	12.73	14.33	15.12	24.24	52.48	66.29	53.76	75.01	88.19
	LinCIR	✓	12.59	13.58	15.00	15.85	25.04	53.25	66.68	57.11	77.37	88.89
	Context-I2W	✓	13.04	14.62	16.14	17.16	25.60	55.10	68.50	-	-	-
	CIReVL	✗	18.57	19.01	20.89	21.80	24.55	52.31	64.92	59.54	79.88	89.69
	LDRE	✗	23.35	24.03	26.44	27.50	26.53	55.57	67.54	60.43	80.31	89.90
	OSrCIR	✗	23.87	25.33	27.84	28.97	**29.45**	57.68	69.86	62.12	81.92	91.10
	SEIZE	✗	24.98	25.82	28.24	28.35	28.65	57.16	69.23	62.22	84.05	92.34
	STiTch(Ours)	✗	**25.55**	**26.27**	**28.81**	**29.99**	28.87	**57.97**	**69.90**	**65.22**	**84.10**	**92.37**
ViT-G/14	Pic2Word	✓	5.54	5.59	6.68	7.12	30.41	58.12	69.23	68.92	85.45	93.04
	SEARLE	✓	13.20	13.85	15.32	16.04	34.80	64.07	75.11	68.72	84.70	93.23
	LinCIR	✓	19.71	21.01	23.13	24.18	35.25	64.72	76.05	63.35	82.22	91.98
	CIReVL	✗	26.77	27.59	29.96	31.03	34.65	64.29	75.06	67.95	84.87	93.21
	LDRE	✗	31.12	32.24	34.95	36.03	36.15	66.39	77.25	68.82	85.66	93.76
	OSrCIR	✗	30.47	31.14	35.03	36.59	37.26	67.25	77.33	69.22	85.28	93.55
	SEIZE	✗	32.46	33.77	36.46	37.55	38.87	69.42	79.42	**74.15**	89.23	95.71
	STiTch (Ours)	✗	**34.40**	**35.56**	**38.07**	**40.02**	**39.23**	**69.95**	**79.56**	73.56	**89.50**	**95.86**

D.2. Impacts of caption number and augmentation views.

Moreover, for clarity, we have provided the specific values corresponding to Fig.3 in the main text and supplemented the results of ablation experiments under different architectures, which can be found in Tab.13. It is evident that compared to a single caption (k=1), multiple captions can provide richer multi-modal knowledge to better understand the implicit input, leading to more accurate descriptions.

D.3. Hyper-parameters Study

We report a sensitivity analysis of α in Tab.14. The results show that STiTch exhibits moderate sensitivity to α, with performance being non-monotonic. Specifically, values in the range of $0.3 - 0.5$ yield optimal results, while overly small or large values degrade performance. This confirms the effectiveness of treating modification as a transition vector, as it helps mitigate biases between MLLM-generated captions and images. For practical use, in accuracy-critical tasks (e.g., CIRCO), we suggest $\alpha \leq 0.5$ to avoid over-modification; In recall-critical tasks (e.g., CIRR), starting with $\alpha = 0.4$ is reasonable. For new datasets, a grid search within $[0.3, 0.5]$ could be conducted, selecting the optimal α based on validation performance tailored to the application's specific needs.

E. More Visualization

For a more comprehensive qualitative analysis, we present the visualization results of GeneCIS datasets about the task of focus in Fig. 7. It illustrated that the original generated descriptions indeed introduce visual noise while our STiTch often focuses on the correct object, leading to higher CIR performance.

F. Further Comparison with SEIZE

We observe that both SEIZE [44] and our STiTch generate multiple captions and apply the semantic calibration process. However, these two models are different from each other in terms of caption generation, semantic calibration strategy, and retrieval score calculation: (1) **Two-Stage Generation vs. One-Stage Generation**: SEIZE first generates N captions for the reference image using a captioner and then modifies them according to the input modification text via an LLM. In contrast, our STiTch directly employs an MLLM to generate N captions for the composed input, eliminating information loss from two-stage approaches. Moreover, the efficiency comparison in Tab. 5 shows that two-stage generation methods are time-consuming, which may limit their applicability in real-time scenarios. (2) **Similarity Space vs. Embedding Space**: SEIZE refines the final retrieval score

Table 7. Performance comparison on Fashion-IQ datasets. Both ViT-B and ViT-L are loaded from OpenAI official weights, while ViT-G is loaded from OpenCLIP. (*) denotes we rerun the experiments on the OpenAI weights, and (†) denotes we rerun the experiments on the OpenCLIP weights.

Fashion-IQ →			**Shirt**		**Dress**		**Toptee**		**Average**	
Arch	Method	Train	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50
ViT-B/32	PALAVRA	✓	21.49	37.05	17.25	35.94	20.55	38.76	19.76	37.25
	SEARLE	✓	24.44	41.61	18.54	39.51	25.70	46.46	22.89	42.53
	CIReVL	✗	28.36	47.84	25.29	46.36	31.21	53.85	28.28	49.35
	CIReVL*	✗	22.03	37.00	13.34	30.14	18.97	38.19	18.11	35.11
	CIReVL†	✗	27.72	46.12	22.01	41.60	30.09	52.22	26.60	46.64
	OSrCIR	✗	31.16	51.13	29.35	50.37	36.51	58.71	32.34	53.40
	OSrCIR*	✗	22.77	40.87	17.01	37.04	20.75	41.00	20.18	39.60
	OSrCIR†	✗	32.83	52.06	29.75	51.91	36.31	58.24	32.96	54.07
	SEIZE	✗	29.38	47.97	25.37	46.84	32.07	54.78	28.94	49.35
	STiTch(Ours)	✗	25.22	44.16	18.59	40.16	25.97	47.61	23.26	43.98
ViT-G/14	Pic2Word	✓	33.17	50.39	25.43	47.65	35.24	57.62	31.28	51.89
	SEARLE	✓	36.46	55.35	28.16	50.32	39.83	61.45	34.81	55.71
	LinCIR	✓	46.76	65.11	38.08	60.88	50.48	71.09	45.11	65.69
	CIReVL	✗	29.85	51.07	27.07	49.53	35.80	56.14	32.19	52.36
	CIReVL*	✗	31.65	49.07	23.90	43.13	32.53	53.19	29.36	48.46
	CIReVL†	✗	32.63	50.05	25.09	45.12	34.42	55.12	30.71	50.10
	OSrCIR	✗	38.65	54.71	33.02	54.78	41.04	61.83	37.57	57.11
	OSrCIR*	✗	36.56	55.45	30.69	53.25	40.13	61.30	35.79	56.67
	OSrCIR†	✗	37.39	56.92	30.59	53.50	39.72	61.04	35.79	57.15
	SEIZE	✗	43.60	65.42	39.61	61.02	45.94	71.12	43.05	65.85
	STiTch(Ours)	✗	39.48	56.59	35.04	56.74	42.86	64.95	39.12	59.43

Table 8. Performance comparison on GeneCIS datasets. **Both ViT-B and ViT-L are loaded from openai official weights, while ViT-G is loaded from openclip**.

GeneCIS →			**Focus Attribute**			**Change Attribute**			**Focus Object**			**Change Object**			**Average**
Arch	Method	Train	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1
ViT-B/32	SEARLE	✓	18.9	30.6	41.2	13.0	23.8	33.7	12.2	23.0	33.3	13.6	23.8	33.3	14.4
	CIReVL	✗	17.9	29.4	40.4	14.8	25.8	35.8	14.6	24.3	33.3	16.1	27.8	37.6	15.9
	OSrCIR	✗	19.4	32.7	42.8	16.4	27.7	38.1	15.7	25.7	35.8	18.2	30.1	39.4	17.4
	STiTch(Ours)	✗	**21.1**	**35.0**	**45.5**	**17.9**	**29.9**	**40.4**	**16.4**	**28.5**	**38.9**	**18.3**	**30.1**	**39.5**	**18.4**
ViT-L/14	SEARLE	✓	17.1	29.6	40.7	16.3	25.2	34.2	12.0	22.2	30.9	12.0	24.1	33.9	14.4
	LinCIR	✓	16.9	30.0	41.5	16.2	28.0	36.8	8.3	17.4	26.2	7.4	15.7	25.0	12.2
	Context-I2W	✓	17.2	30.5	41.7	16.4	28.3	37.1	8.7	17.9	26.9	7.7	16.0	25.4	12.7
	CIReVL	✗	19.5	31.8	42.0	14.4	26.0	35.2	12.3	21.8	30.5	17.2	28.9	37.6	15.9
	OSrCIR	✗	**20.9**	33.1	44.5	17.2	28.5	37.9	15.0	23.6	34.2	18.4	30.6	38.3	17.9
	SEIZE	✗	20.5	33.4	45.0	17.6	28.9	38.5	15.4	25.6	36.2	18.7	30.9	39.8	18.1
	STiTch(Ours)	✗	20.3	**34.6**	**46.4**	**18.3**	**29.8**	**41.6**	**16.8**	**28.5**	**38.4**	**18.8**	**31.0**	**40.3**	**18.6**
ViT-G/14	LinCIR	✓	19.1	33.0	42.3	17.6	30.2	38.1	10.1	19.1	28.1	7.9	16.3	25.7	13.7
	CIReVL	✗	20.5	34.0	44.5	16.1	28.6	39.4	14.7	25.2	33.0	18.1	31.2	41.0	17.4
	OSrCIR	✗	22.7	36.4	47.0	17.9	30.8	42.0	16.9	28.4	36.7	**21.0**	**33.4**	**44.2**	19.6
	SEIZE	✗	**22.9**	36.2	47.3	18.6	31.4	42.7	18.2	28.8	37.6	19.6	33.0	43.5	19.8
	STiTch(Ours)	✗	21.9	**36.4**	**47.9**	**19.6**	**31.9**	**42.8**	**20.2**	**30.3**	**39.6**	19.7	33.2	43.4	**20.4**

by directly changing the cosine score. Our STiTch aims to refine the generated captions in the CLIP embedding space. (3) **Point-to-Point vs. Set-to-Set**: SEIZE represents the final global caption feature by employing the average pooling on captions, and then measures similarity with candidates via cosine similarity. Our STiTch, however, models the captions as a discrete distribution and then develops a transportation-aware set-to-set metric to calculate the distances.

For experiments, STiTch demonstrates more pronounced improvements on CIRR and CIRCO (Tab.6) as well as GeneCIS (Tab.8), where the modification descriptions are richer and require more faithful semantic modeling. Such settings align well with STiTch's one-stage caption generation and transportation-aware set-to-set metric, which jointly

Table 9. Ablation results of different alignment strategies on CIRCO and CIRR datasets.

CIRCO + CIRR →		**CIRCO**				**CIRR**					
Metric		mAP@k				Recall@k			$\text{Recall}_{\text{Subset}}$@k		
Arch	Method	k=5	k=10	k=25	k=50	k=1	k=5	k=10	k=1	k=2	k=3
ViT-B/32 (OpenAI)	w/CT	20.26	21.01	23.01	24.04	25.83	55.18	68.22	65.64	83.60	92.80
	w/OT	19.81	20.48	22.33	23.31	24.72	53.66	66.77	65.23	83.64	92.46
ViT-B-32 (OpenCLIP)	w/CT	28.21	28.99	31.31	32.53	32.56	62.10	73.86	70.00	86.60	94.87
	w/OT	23.94	24.79	27.01	28.09	31.37	60.97	72.92	69.54	85.76	94.27
ViT-L/14 (OpenAI)	w/CT	25.55	26.27	28.81	29.99	28.87	57.97	69.90	65.22	84.10	92.37
	w/OT	24.76	25.85	28.52	29.67	28.05	57.01	69.64	65.71	83.74	92.17
ViT-L-14 (OpenCLIP)	w/CT	32.31	33.33	36.32	37.49	35.04	65.57	76.41	71.52	88.00	94.65
	w/OT	30.23	31.22	34.13	35.20	34.39	64.48	76.17	71.98	88.34	94.68

Table 10. Ablation results of different alignment strategies on GeneCIS dataset.

GeneCIS →		**Focus Attribute**			**Change Attribute**			**Focus Object**			**Change Object**			**Average**
Arch	Method	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1
ViT-B/32 (OpenAI)	w/CT	21.1	35.0	45.5	17.9	29.9	40.4	16.4	28.5	38.9	18.3	30.1	39.5	18.4
	w/OT	20.5	34.1	44.9	17.5	30.2	40.1	16.3	28.0	38.3	18.3	29.9	39.4	18.2
ViT-B-32 (OpenCLIP)	w/CT	20.8	33.6	44.2	17.6	29.4	39.7	17.4	30.7	40.4	19.6	33.6	44.2	18.9
	w/OT	20.2	33.4	43.8	17.1	29.4	39.0	17.0	29.7	39.9	20.0	33.4	43.8	18.6
ViT-L/14 (OpenAI)	w/CT	20.3	34.6	46.4	18.3	29.8	41.6	16.8	28.5	38.4	18.8	31.0	40.3	18.6
	w/OT	20.6	34.5	45.8	18.0	29.2	40.3	16.8	27.9	38.2	18.8	30.2	40.1	18.5
ViT-L-14 (OpenCLIP)	w/CT	20.5	33.8	44.2	18.1	29.0	40.2	18.5	29.4	39.1	19.9	32.9	42.8	19.3
	w/OT	20.3	33.4	44.4	17.6	28.8	39.9	17.9	28.6	38.1	19.6	33.1	42.3	18.9

Table 11. Ablation results on the transition and transportation modules. All results are conducted on CIRCO datasets with GPT-4o(mini).

Strategy		mAP@k			
Transition	*Transportation*	k=5	k=10	k=25	k=50
✗	✗	35.49	37.05	40.02	41.28
✓	✗	37.50	39.10	42.24	43.50
✗	✓	36.61	38.10	41.11	42.38
✓	✓	**38.93**	**40.14**	**43.18**	**44.46**

preserve semantic diversities and model transitions in embedding space more effectively than two-stage caption–editing pipelines. On Fashion-IQ, however, STiTch performs below SEIZE. This is primarily due to the overly simple modification text in Fashion-IQ (*e.g.*, "is solid white", "is a lighter color"), which provides limited semantic signal and therefore offers suboptimal guidance for our transition vectors. SEIZE, by contrast, relies on a pre-trained captioning model to generate reference captions, a strategy more compatible with Fashion-IQ's simplified language. However, as shown in Tab.7, this approach incurs noticeably higher inference-time computational cost, with STiTch achieving nearly a 3× speed-up over SEIZE.

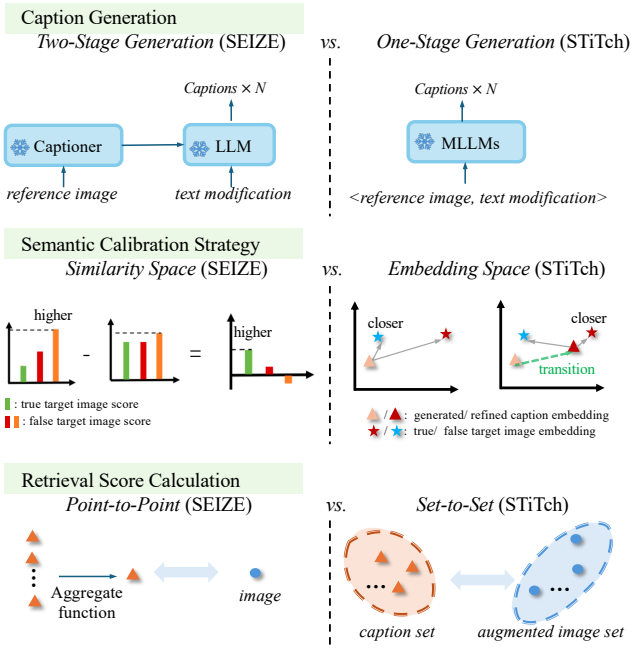

Figure 6. Further comparison between SEIZE and STiTch in terms of caption generation, semantic calibration strategy, and retrieval score calculation (zoom-in for more details).

Table 12. Performance comparison on CIRCO and CIRR datasets with various MLLMs.

CIRCO + CIRR →	**CIRCO**				**CIRR**					
Metric	mAP@k				Recall@k			$\text{Recall}_{\text{Subset}}$@k		
Method	k=5	k=10	k=25	k=50	k=1	k=5	k=10	k=1	k=2	k=3
Qwen-2B	22.49	23.64	25.90	26.95	26.05	53.28	65.59	64.53	82.46	91.25
Qwen-7B	25.55	26.27	28.81	29.99	28.87	57.97	69.90	65.22	84.10	92.37
LLaVA-Next (Mistral-7B)	24.17	24.73	27.03	28.11	26.97	55.10	66.92	65.01	82.75	91.40
GPT-4o(mini)	25.68	26.50	29.16	30.30	28.59	58.13	69.99	66.15	84.98	92.86

Table 13. Ablation study on CIRCO and CIRR datasets with different number of image augmentation on CLIP-B/32 and fix the number of description to 5.

CIRCO + CIRR →	**CIRCO**				**CIRR**					
Metrics	mAP@k				Recall@k			$\text{Recall}_{\text{Subset}}$@k		
Num	k=5	k=10	k=25	k=50	k=1	k=5	k=10	k=1	k=2	k=3
1	19.73	19.89	21.68	22.63	25.16	53.59	66.46	63.88	82.87	92.19
5	20.19	20.84	22.70	23.73	25.25	54.00	67.40	64.46	83.61	92.39
10	20.26	21.01	23.01	24.04	25.83	55.25	68.22	65.64	83.60	92.80
25	20.60	21.37	23.55	24.55	25.64	55.45	68.87	65.71	84.41	92.46
50	21.01	21.62	23.74	24.79	**26.15**	**55.78**	**69.16**	**66.17**	**84.74**	93.06
100	**21.96**	**22.51**	**24.60**	**25.62**	25.67	55.69	69.08	65.45	84.36	**93.08**

Table 14. Sensitivity analysis of α on Qwen2-VL-7B and ViT-B/32 on CIRCO and CIRR datasets (default $\alpha = 0.45$ in our main manuscript).

CIRCO + CIRR →	**CIRCO**				**CIRR**				
Metrics	mAP@k				Recall@k			$\text{Recall}_{\text{Subset}}$@k	
α value	k=5	k=10	k=25	k=50	k=1	k=5	k=10	k=1	k=2
0.1	18.37	19.09	20.77	21.76	23.28	49.98	62.36	64.05	83.21
0.2	19.74	20.49	22.34	23.32	24.63	52.46	65.45	64.89	83.40
0.3	21.71	22.36	24.33	25.26	25.35	54.12	67.28	65.49	83.74
0.4	20.73	21.37	23.33	24.37	25.81	55.37	68.34	65.23	83.67
0.45	20.26	21.01	23.01	24.04	25.83	55.25	68.22	65.64	83.60
0.5	21.47	22.47	24.46	25.46	26.02	55.45	68.58	64.82	83.49
0.6	19.77	20.45	22.55	23.48	25.62	55.40	68.22	63.64	83.13
0.7	19.05	20.18	22.11	23.20	25.11	54.65	68.22	63.62	82.68

G. STiTch In-Context Learning Details

We utilize an in-context learning method in Fig.8. To achieve ZS-CIR, each sample uses the same placeholder "`<image_url>`" instead of an actual reference image URL. By providing several example outputs, the model is able to understand the required reasoning process without an actual reference image. This approach ensures efficient reasoning in a zero-sample setting. Each text requires the model to focus on a specific object and provide a detailed description. This helps the model understand the key elements in the image and how they relate to each other. We use uniform placeholders `<image_url>` and `<reference_image_url>` to ensure that the input and output formats are consistent for easy model processing.

H. Limitations and Future Work

Although our method achieves strong performance, there remain several directions for future exploration. First, while the visual augmentation applied to target images is lightweight and performed offline, our STiTch requires approximately $M - 1$ times more memory than others. We leave memory optimization as future work, with potential directions including online strategies or coarse-to-fine retrieval. Second, when the query image depicts a complex scene involving multiple objects or relationships, and the accompanying modification text provides insufficient detail, our STiTch may focus on the wrong or ambiguous object, leading to unexpected captions. This limitation is consistent with issues observed in prior CIReVL [17] and OSrCIR [36] models. Moreover, current benchmarks suffer from a false-negative problem. As noted in [27], each (reference image, modification) pair in FashionIQ can correspond to multiple valid target images, yet only one is annotated as ground truth. Consequently, semantically correct retrieval results may be unfairly penalized under existing evaluation protocols. We leave these challenges as promising directions for future research.

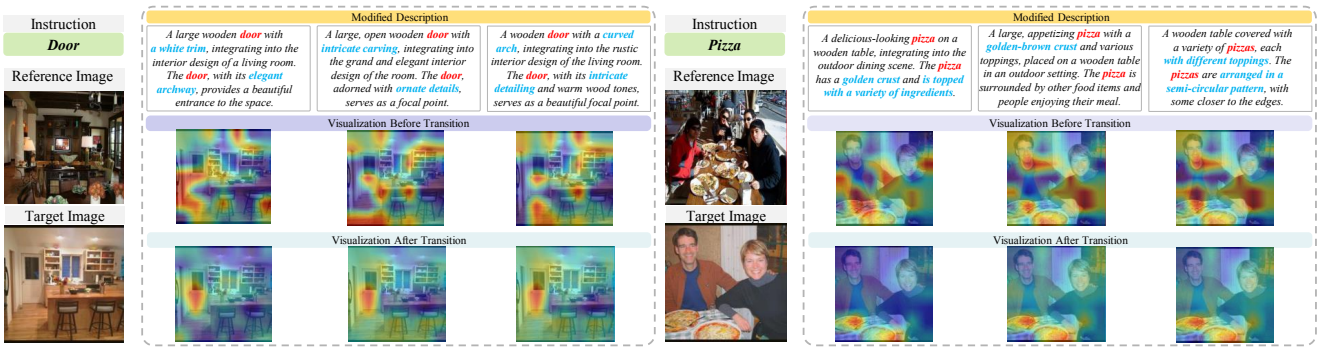

Figure 7. Visualization of the GeneCIS dataset on the 'Focus Object' task. Heatmaps before and after the transition on target image are shown. Captions generated by MLLMs often contain irrelevant visual noise (*blue* text), while the STT model effectively suppresses such noise and highlights the correct focus object (*red text*).

Example 1

<Input>

{

"Reference Image 1": <image_url>, (each sample uses the same placeholder "<image_url>" instead of an actual image URL)

"Text modification 1": "**Instruction:** Focus cardboard and its arrangement with other objects in the image. Describe cardboard with details. **Edited Description:** A flattened piece of cardboard placed at the base of a disorganized pile. The cardboard, with its worn and creased surface, shows frayed edges and slight tears, indicating frequent use or exposure to the elements. It serves as a foundation for the objects stacked above, including bright blue garbage bags, a broken white plastic chair, and parts of a large appliance."

"Reference Image 2": <image_url>, (each sample uses the same placeholder "<image_url>" instead of an actual image URL)

"Text modification 2": "**Instruction:** Focus wood floor and its arrangement with other objects in the image. Describe wood floor with details. **Edited Description:** A polished wood floor with a warm, rich tone that adds to the cozy ambiance of the room. The floor's natural grain is partially covered by a soft, beige area rug placed in the center, creating a harmonious balance between texture and color. Surrounding the rug, the floor extends under the dark brown sofa and armchair, complementing their earthy tones. The floor seamlessly integrates with the room's arrangement, supporting the furniture and decor, such as the side tables and potted plants, enhancing the overall inviting atmosphere."

"Reference Image 3": <reference_image_url_1>,

"Text modification 3": " **Instruction:** Focus **<relative_caption>** and its arrangement with other objects in the image. Describe **<relative_caption>** with details. **Edited Description:** "

}

<Response>

{

"Target Image Description": "A large wooden door with a white trim, integrating into the interior design of a living room. The door, with its elegant archway, provides a beautiful entrance to the space."

}

Example 2

<Input>

{

"Reference Image 1": <image_url>, (each sample uses the same placeholder "<image_url>" instead of an actual image URL)

"Text modification 1": "**Instruction:** Focus cardboard and its arrangement with other objects in the image. Describe cardboard with details. **Edited Description:** A flattened piece of cardboard placed at the base of a disorganized pile. The cardboard, with its worn and creased surface, shows frayed edges and slight tears, indicating frequent use or exposure to the elements. It serves as a foundation for the objects stacked above, including bright blue garbage bags, a broken white plastic chair, and parts of a large appliance."

"Reference Image 2": <image_url>, (each sample uses the same placeholder "<image_url>" instead of an actual image URL)

"Text modification 2": "**Instruction:** Focus wood floor and its arrangement with other objects in the image. Describe wood floor with details. **Edited Description:** A polished wood floor with a warm, rich tone that adds to the cozy ambiance of the room. The floor's natural grain is partially covered by a soft, beige area rug placed in the center, creating a harmonious balance between texture and color. Surrounding the rug, the floor extends under the dark brown sofa and armchair, complementing their earthy tones. The floor seamlessly integrates with the room's arrangement, supporting the furniture and decor, such as the side tables and potted plants, enhancing the overall inviting atmosphere."

"Reference Image 4": <reference_image_url>,

"Text modification 4": " **Instruction:** Focus **<relative_caption>** and its arrangement with other objects in the image. Describe **<relative_caption>** with details. **Edited Description:** "

}

<Response>

{

"Target Image Description": "A delicious-looking pizza on a wooden table, integrating into the outdoor dining scene. The pizza has a golden crust and is topped with a variety of ingredients."

}

Figure 8. Examples of our in-context learning on GeneCIS dataset. Each sample uses the same placeholder "`<image_url>`" instead of an actual reference image URL.

.