A. Proof of Theorem 3.1

To ease the reading, we first restate Theorem 3.1 and then provide the proof.

Theorem 3.1. For a pair of modality encoders $g_T(\cdot)$ and $g_V(\cdot)$, if the multi-modal features $Z_T = g_T(X_T)$ and $Z_V = g_V(X_V)$ are perfectly aligned in the feature space, i.e., $Z_T = Z_V$, then $\inf_h \mathbb{E}_p[\ell_{\text{CE}}(h(Z_T, Z_V), Y)] - \inf_{h'} \mathbb{E}_p[\ell_{\text{CE}}(h'(X_T, X_V), Y)] \geq \Delta_p$.

Proof of Theorem 3.1. Consider the joint mutual information $I(Z_T, Z_V; Y)$. By the chain rule, we have the following decompositions:

$$\begin{aligned} I(Z_T, Z_V; Y) &= I(Z_T; Y) + I(Z_V; Y \mid Z_T) \\ &= I(Z_V; Y) + I(Z_T; Y \mid Z_V). \end{aligned}$$

However, since Z_T and Z_V are perfectly aligned, $I(Z_V; Y \mid Z_T) = I(Z_T; Y \mid Z_V) = 0$, which means $I(Z_T, Z_V; Y) = I(Z_V; Y) = I(Z_T; Y)$. On the other hand, by the celebrated data-processing inequality, we know that

$$I(Z_T; Y) \leq I(X_T; Y), \quad I(Z_V; Y) \leq I(X_V; Y).$$

Hence, the following chain of inequalities holds:

$$\begin{aligned} I(Z_T, Z_V; Y) &= \min\{I(Z_T; Y), I(Z_V; Y)\} \\ &\leq \min\{I(X_T; Y), I(X_V; Y)\} \\ &\leq \max\{I(X_T; Y), I(X_V; Y)\} \\ &\leq I(X_T, X_V; Y), \end{aligned}$$

where the last inequality follows from the fact that the joint mutual information $I(X_T, X_V; Y)$ is at least as large as any one of $I(X_T; Y)$ and $I(X_V; Y)$. Therefore, due to the variational form of the conditional entropy, we have

$$\begin{aligned} &\inf_h \mathbb{E}_p[\ell_{\text{CE}}(h(Z_T, Z_V), Y)] - \inf_{h'} \mathbb{E}_p[\ell_{\text{CE}}(h'(X_T, X_V), Y)] \\ &= H(Y \mid Z_T, Z_V) - H(Y \mid X_T, X_V) \\ &= I(X_T, X_V; Y) - I(Z_T, Z_V; Y) \\ &\geq \max\{I(X_T; Y), I(X_V; Y)\} - \min\{I(X_T; Y), I(X_V; Y)\} \\ &= \Delta_p. \end{aligned}$$

■

B. Additional Results

B.1. Two-tower-based models

Visualization of constructing latent structures: To better understand the effect of constructing latent modality structures, we visualize the effect of our method on the latent space in Fig. 5. Note that all our methods achieve performance gain regardless of size of the modality gap, which complys with Section 3 and Theorem 3.1.

Table 5. Downstream tasks performance on fusion-based models.

Method	VQA		NLVR^2		SNLI-VE	
	test-dev	test-std	dev	test-P	val	test
ALBEF [32]	73.38	73.52	78.36	79.54	79.69	79.91
CODIS [13]	73.15	73.29	78.58	**79.92**	79.45	80.13
OURS$_{\text{All}}$	74.12	74.16	**80.18**	79.80	79.62	**80.23**
OURS$_{\text{Sep}}$	73.52	73.59	79.05	79.76	**79.95**	79.61
OURS$_{\text{Br}}$	**74.26**	**74.36**	78.70	79.36	79.86	79.95
OURS$_{\text{GC}}$	73.90	73.87	78.96	79.53	79.82	80.16

Table 6. Ablation study on zero-shot image-text retrieval performance on Flickr30K with model pre-trained on COCO.

Method	Text Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
ALBEF [32]	58.4	83.2	89.5	44.5	69.8	78.0
CODIS [13]	62.7	87.0	92.3	49.0	74.1	82.9
OURS$_{\text{Sep}}$	**66.0**	**88.2**	**93.9**	50.4	76.2	83.7
OURS$_{\text{Br}}$	65.4	88.1	93.1	**50.8**	**77.1**	**84.4**
OURS$_{\text{GC}}$	64.3	87.5	92.3	50.5	75.9	83.3

Analysis of selecting regularizers: As we have illustrated the effect of each regularizer in the latent space. Together with the insights from our theoretical results and both quantitative and qualitative evidences, we conclude the following:

1) The deep feature separation regularizer is ideal for *downstream tasks with fine-tuning stages*. As shown by our Theorem 3.1, feature separation helps to preserve the modality-specific information, which could be helpful in downstream tasks with proper fine-tuning. From Tab. 3, feature separation achieves remarkable performance on linear probing (extra supervision) tasks.

2) The Brownian bridge regularizer is most effective when *the distribution shift between the training and downstream tasks is small*. It aims to bridge two modalities and in Fig. 5(d) it indeed reduced the modality gap and constrained features into one region. Thus, in-distribution tasks (*i.e.*, test features fall into the same region as training data in latent space) benefit most. Better results on ImageNet-related tasks (closer to training distribution) but worse ones on CIFAR-related task in Tab. 1 and Tab. 2 confirm this.

3) The geometric consistency regularizer is most robust to *distribution shifts in downstream tasks*. It regulates the geometric shape of two modalities as shown in Fig. 5(e), which could potentially preserve the consistency under distribution shifts. Results in Tab. 2 shows the regularizer to be robust under natural distribution shifts with superior results.

4) Based on the discussion above, we conclude that the choice among these three regularizers is often *dataset/task-specific*. Each regularizer enforces different inductive bias to the structure of the feature space. Hence, more regularizers do not necessarily lead to better performance. We also conducted additional experiments in Tab. 5 to show

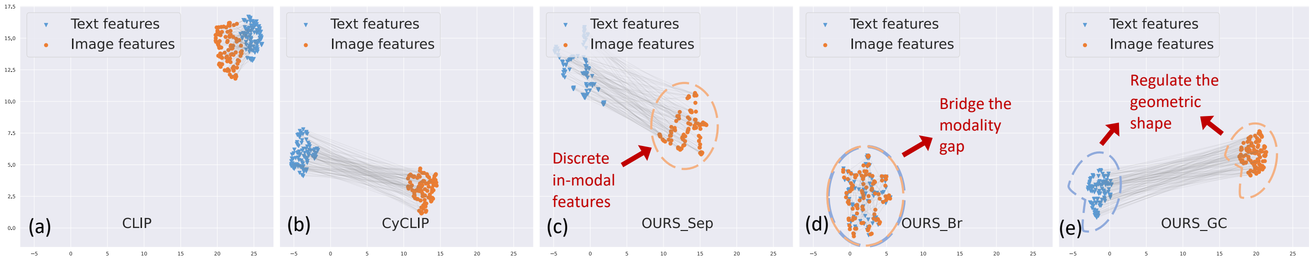
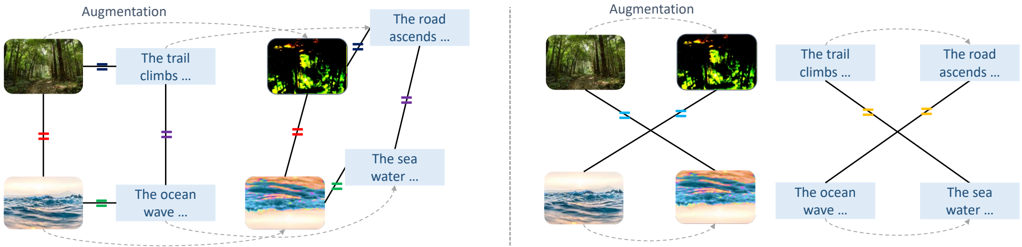


Figure 5. Visualization of constructing latent modality structures. Each line connects the positive image-text feature pair.



Geometric consistency on augmented features – Inter-modal design

Geometric consistency on augmented features – In-modal Design

Figure 6. Visualization of constructing latent modality structures. Each line connects the positive image-text feature pair.

Table 7. Comparison of different design choices for geometric consistency on augmented features.

Method	Text Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
Inter-modal	**64.3**	**87.5**	**92.3**	**50.5**	**75.9**	**83.3**
In-modal	61.7	85.8	91.8	48.0	74.7	82.5

that combining all regularizers cannot always outperform the single regularizer baseline. Overall, we suggest to use one regularizer for simplicity and efficiency, though combining them could be beneficial in some cases.

Visualization of experimental results: To better demonstrate the effectiveness of our proposed methods, we visualize our experimental results on two-tower-based framework (*e.g.* CLIP) in Fig. 7 and Fig. 8. Our methods show significant improvement on most of the tasks.

B.2. Fusion-based models

Results of using all regularizers together: We provide additional results on using all three regularizers. The results are shown in Tab. 5. While using all the regularizations together leads to performance gain, all our regularization methods improve the performance when used individually.

Results on small scale experiments: We evaluate zero-shot image-text retrieval on smaller scale experiments by pretaining on COCO and evaluate on Flickr30 [71]. As shown in Tab. 6, results indicate that all three regularizations improve the performance, while text retrieval benefits most from deep feature separation regularization and image retrieval task benefits most from Brownian bridge regularization.

Results on other geometric consistency loss design: We also explored other possible designs as in Fig. 6. We use COCO for pretraining and Flickr30 for testing. In the main paper, we apply the inter-modal design for geometric consistency on augmented features. As shown in Tab. 7, such inter-modal design has better performance.

C. Implementation Details

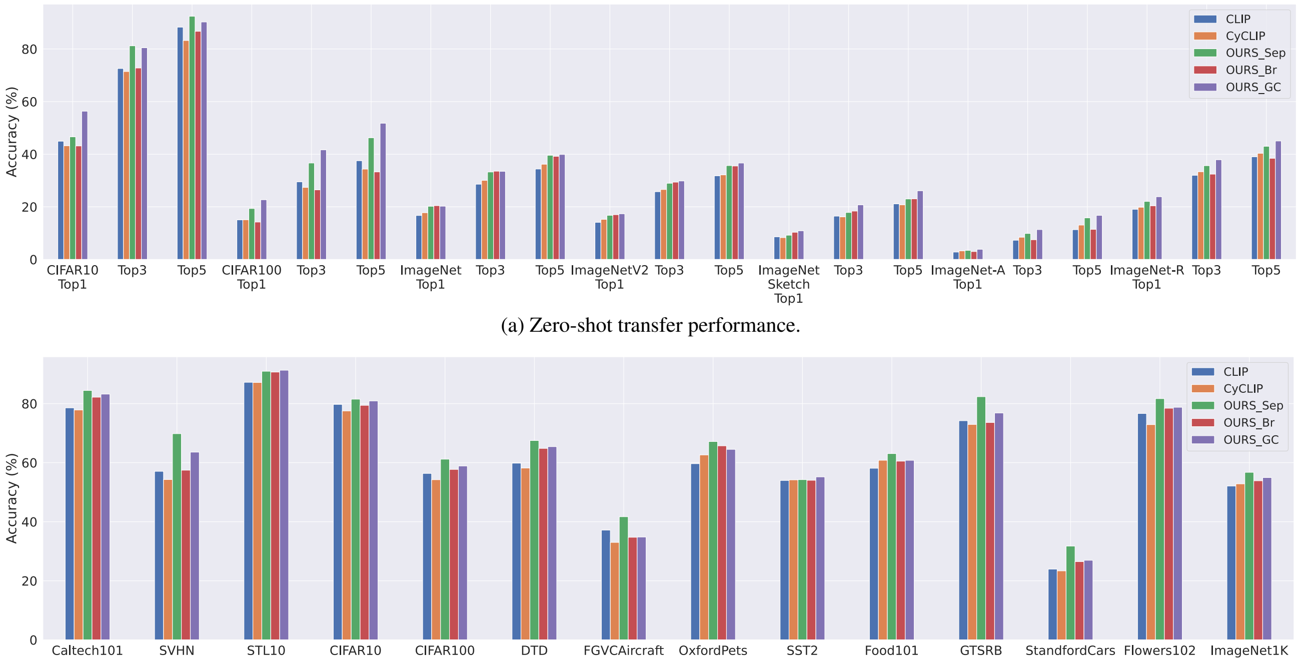
C.1. Two-tower based models.

We follow the same code base and hyper-parameters setting as CyCLIP [18] except for number of GPUs. We train the model from scratch on 64 NVIDIA A100 GPUs and train for 64 epochs. Our batch size is 128 and feature dimension is 1024. We use an initial learning rate of $5e^{-4}$ with cosine scheduling. We warm-up the model for 10000 steps. We evaluate the model trained to the last epoch for our method.

C.2. Fusion based models.

We follow the codebase and hyper-parameter setting as [13, 32] except for number of GPUs. We train all the models on 16 NVIDIA A100 GPUs. During the pre-train stage, we train with the pre-training tasks for 30 epochs. AdamW [37] optimizer is used along with weight decay of 0.02, batch size of 512, learning rate initially of 1e-5. We warm up the learning rate to 1e-4 after 1000 iterations and follow the cosine decay. The input size for pre-training task is 256 and the input sizes for downstream tasks are 384.

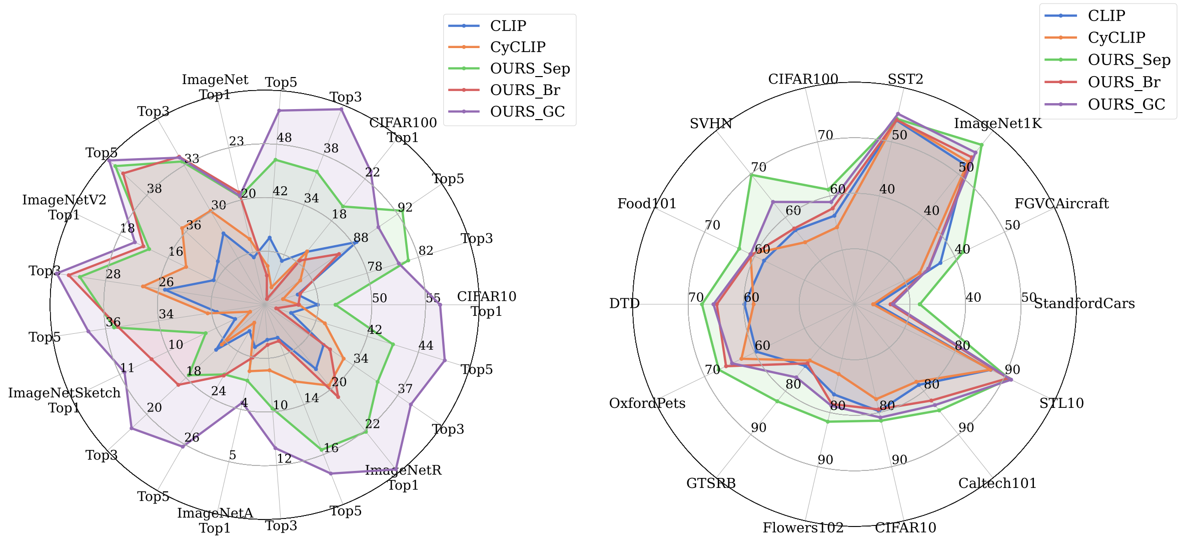
Reproducibility We follow the standard practice to fix the random seed to ensure that all our results are reproducible. The source code will be public.



(a) Zero-shot transfer performance.

(b) Linear-probing performance.

Figure 7. Visualization of Two-tower-based methods (*e.g*. CLIP) performance. Each color represents a different approach.



(a) Zero-shot Performance.

(b) Linear Probing Performance.

Figure 8. Visualization of Two-tower-based methods (*e.g*. CLIP) performance. Each axis represents the performance on a dataset with a certain metric. Each color represents different approach. The larger area that one approach covers, the better overall performance.