Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning

Weixin Liang*
Stanford University
wxliang@stanford.edu

Yuhui Zhang *
Stanford University
yuhuiz@stanford.edu

Yongchan Kwon *
Columbia University
yk3012@columbia.edu

Serena Yeung
Stanford University
syyeung@stanford.edu

James Zou
Stanford University
jamesz@stanford.edu

Abstract

We present *modality gap*, an intriguing geometric phenomenon of the representation space of multi-modal models. Specifically, we show that different data modalities (e.g. images and text) are embedded at arm's length in their shared representation in multi-modal models such as CLIP. Our systematic analysis demonstrates that this gap is caused by a combination of model initialization and contrastive learning optimization. In model initialization, we show empirically and theoretically that the representation of a common deep neural network is restricted to a narrow cone. As a consequence, in a multi-modal model with two encoders, the representations of the two modalities are clearly apart when the model is initialized. During optimization, contrastive learning keeps the different modalities separated by a certain distance, which is influenced by the temperature parameter in the loss function. Our experiments further demonstrate that varying the modality gap distance has a significant impact in improving the model's downstream zero-shot classification performance and fairness. Our code and data are available at https://modalitygap.readthedocs.io/

1 Introduction

Multi-modal models map inputs from different data modalities (e.g. image and text) into a shared representation space (Figure 1 (a)). It has garnered tremendous interest and excitement as a framework for data integration. As a prominent example pre-trained on a web-scale collection of images and natural language, OpenAI's CLIP model [39], has learned diverse visual concepts that can readily be transferred to downstream tasks through *prompting*: one can perform "zero-shot" visual classification by simply providing the names of the visual categories to be recognized.

In this work, we present the *modality gap* phenomenon: As shown in Figure 1 (b), CLIP's image embeddings and text embeddings are located in two completely separate regions of the embedding space. We find this phenomenon consistently across various multi-modal models, covering texts, natural images [39], videos [50], medical images [53], and amino-acid sequences [11]. Interestingly, this phenomenon still holds even when we embed using multi-modal models with *random* weights (Figure 1 (c)). While it might seem reasonable to attribute the gap to differences in data distributions or to the different encoder architectures, we showed that these factors are not the fundamental cause.

This paper provides a three-part explanation for the modality gap phenomenon. **(1)** The general inductive bias of deep neural architecture creates a *cone effect*: The effective embedding space is

*These three authors contributed equally.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

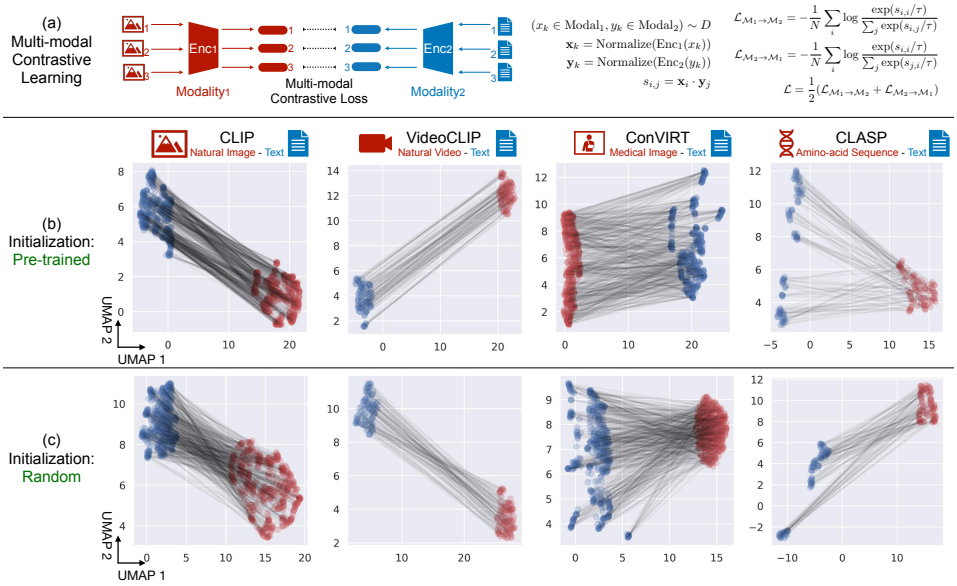

Figure 1: **The pervasive *modality gap* in multi-modal contrastive representation learning. (a) Overview of multi-modal contrastive learning.** Paired inputs from two modalities (e.g., image-caption) are sampled from the dataset and embedded into the hypersphere using two different encoders. The loss function is to maximize the cosine similarity between matched pairs given all the pairs within the same batch. **(b) UMAP visualization of generated embeddings from pre-trained models.** Paired inputs are fed into the pre-trained models and the embeddings are visualized in 2D using UMAP (lines indicate pairs). We observe a clear modality gap for various models trained on different modalities. **(c) UMAP visualization of generated embeddings from same architectures with random weights.** Modality gap exists in the initialization stage without any training.

restricted to a narrow cone for pre-trained models or models with random weights. **(2)** Different random initializations create different embedding cones. Since a multi-modal model consists of two encoders, which create different cones at random initialization, this explains how the modality gap is present at initialization. **(3)** The contrastive learning objective commonly used by multi-modal models preserves the gap. We support our explanations with theory and experiments. Our theoretical analysis shows that under mild assumptions, each neural network layer shrinks the angle between any pair of embedding vectors with high probability, thereby creating more narrow cones in deeper architectures. We further prove that different random initializations of model weights result in different cones. Interestingly, increasing the modality gap in models like CLIP can improve its downstream performance on several zero-shot learning and fairness tasks. The main objective of our paper is to i) empirically demonstrate the modality gap phenomenon across different data modalities and NN architectures; ii) explain how the gap arises and iii) show that the size of the gap can affect downstream applications. It is *not* our goal to propose a method to close the gap, since it's not clear that it's desirable to have no modality gap. Together, this paper makes the **following contributions**:

1. To the best of our knowledge, we demonstrate a general *modality gap* phenomenon for the first time. We show that this phenomenon holds across a wide spectrum of multi-modal models, covering texts, natural images, videos, medical images, and amino-acid sequences.
2. We demonstrate the significant implications of modifying the gap in downstream applications. By simply modifying the gap's distance, we can improve CLIP's zero-shot performance and fairness.
3. To explain modality gap, we provide a three-part explanation supported by extensive theoretical and empirical analyses. Our analyses also provide new insights on the cone effect, which we show is a general phenomenon for deep neural networks. Existing work focuses on *trained* language models and attributes the cone effect to the *optimization* under unbalanced word frequencies distribution. We demonstrate that this effect holds not only across various modalities and network architectures, but also on random noise inputs and random weights, which is not captured in previous work.

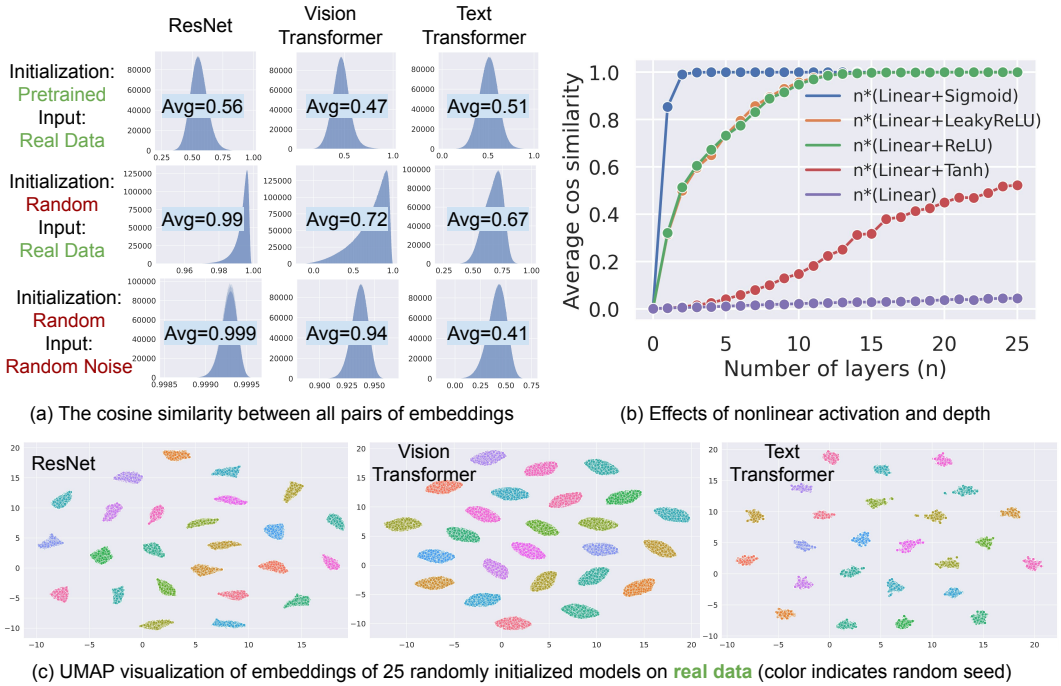

Figure 2: **The cone effect phenomenon. (a) Histograms of the cosine similarity between all pairs of embeddings across various settings.** The average cosine similarity is substantially larger than 0, indicating that the embedding space is a narrow cone. The cone effect also holds on randomly initialized models, and on random noise inputs. **(b) Effects of nonlinear activation and depth.** Inputs are 512-dim standard normal random vector. All MLP linear layers are 512×512, with both weight and bias randomly initialized from $\mathcal{N}(0, \frac{1}{512})$. Y axis is the average cosine similarity between pairs of embeddings. **(c) UMAP visualization of embeddings of 25 randomly initialized models (without training) on *real* data.** Each random initialization forms a distinctively different cone. *Real Data:* 5,000 image-caption pairs from the validation set of MSCOCO Caption. *Random Noise:* Gaussian noise from the standard normal distribution as images, uniformly random integer sequences as texts.

4. We mathematically characterize the contraction mapping induced by linear layers with ReLU non-linearities to explain the cone effect. Our theory matches well with experiments and provides insights for understanding the general inductive biases of deep neural networks.

2 The Cone Effect Induces A Modality Gap

2.1 The Narrow Cone of Embeddings

In order for modality gap to exist, the embeddings from a encoder should be concentrated around a subregion of the full embedding space—otherwise, the embeddings from different encoders would overlap. Motivated by this, we begin our investigation by showing that the modality gap already arises at random model initialization due to the *cone effect*: The effective embedding space is restricted to a narrow cone for trained models and models with random weights. To demonstrate this, we extract 5,000 embeddings from the final layer of 3 pre-trained models respectively (ResNet, Vision Transformer, Text Transformer)[2] on MSCOCO Caption [8]. We then compute the cosine similarity between all possible pairs of the 5,000 embeddings within each model (Figure 2 (a)). We found that both the average cosine similarity (0.56, 0.47, 0.51 respectively for the 3 models) and the minimum cosine similarity (0.23, 0.05, 0.01) are positive. These results indicate that the embedding space is a narrow cone.

[2]ResNet embeddings are extracted before the final linear layer. We use ResNet-18 pre-trained on ImageNet, Vision Transformer and Text Transformer from pre-trained CLIP

In the literature, the cone effect has been observed in the language representations from language models (e.g., BERT) [12]. A common explanation is that the *unbalanced* distribution of word frequencies biased the *optimization* [15, 33]. However, we found that the cone effect still exists in models with random weights (Figure 2 (c)). In fact, the average cosine similarity there is even *higher* than in trained models. For example, any two embeddings from a randomly initialized ResNet have on average an almost perfect (0.99) cosine similarity. Interestingly, the cone effect still holds when the input data is random noise[3], indicating that unbalanced data distribution suggested in previous works is not necessary for the cone effect. Together these experiments suggest that the cone effect reflects a more general inductive bias of deep networks than might be previously appreciated.

How narrow is the cone in 512-dim representation space? We clarify that a cosine similarity with 0.56 already indicates that the embedding space is actually an extremely narrow cone in the 512-dimensional feature space. Consider the fraction of surface area in a unit hypersphere: In 2D, arccos(0.56)=55.94°, indicating that a cosine similarity of 0.56 can "occupy" 55.94°/360°=15.53% of the 2D unit circle. In 3D, a cosine similarity of 0.56 can "occupy" $\frac{2\pi r^2(1-\cos\frac{55.94^\circ}{2})}{4\pi r^2}$=3.34% of the 3D unit sphere. In 512D, a cosine similarity of 0.56 can "occupy" less than $\frac{1}{2^{512}}$ fraction of the surface area in a unit 512D hypersphere. These evidences show that the effective embedding space is restricted to an extremely narrow cone.

2.2 The effects of non-linear activation on cone effect

Design To study the effects of non-linear activation functions on the cone effect, we randomly initialized various MLPs with different non-linearities or without non-linearities. The inputs of the MLPs are 512-dim standard normal random vectors. All MLP linear layers are 512×512, with both weight and bias randomly initialized from $\mathcal{N}(0, \frac{1}{512})$, here we denote a Gaussian distribution with mean μ and variance σ^2 by $\mathcal{N}(\mu, \sigma^2)$.

Results As shown in Figure 2 (b), MLPs without non-linear activation shows little cone effect. However, with non-linearity, the average cosine similarity increases *rapidly* as the number of layers increases. For example, the average cosine similarity reaches 0.99 for a 2-layer MLP with Sigmoid. These results indicate that the non-linear activation functions play a crucial role in the cone effect.

Although it is easy to see that ReLU makes every coordinate non-negative, and thus cosine similarity after ReLU is guaranteed to be non-negative, we highlight that none of the 3 models in Figure 2 (a) has ReLU as the final layer before embedding extraction[4]. In addition, although all 3 models incorporate normalization layers such as batch norm [23] and layer norm [4] in their architectures, we still observe the cone effect. Further analyzing the connection between normalization and the cone effect is an interesting direction of future work.

2.3 Different random initializations create different cones

Next, we study the effect of different random initialization on the cone effect. In Figure 2 (c), we randomly initialized a model 25 times, and plotted its extracted embeddings on the same *real data* (i.e., MSCOCO Caption) via UMAP visualization [41]. We found that each random initialization forms a distinctively different cone. This phenomenon holds across various neural network architectures and input modalities (ResNet, Vision Transformer or Text Transformer), on ImageNet-pretrained models (Supp. Figure 13), on PCA visualization (Supp. Figure 7), or with random noise inputs (Supp. Figure 5). Since a multi-modal model consists of two encoders, which creates different cones at random initialization, this explains how the modality gap is present at initialization. *While it might seem reasonable to attribute the modality gap to differences in data modalities [21], Figure 2 (c) shows the gap still exists even if the two encoders operate on the exact same data in the exact same modality. Therefore, the gap can exist without different modalities, and we emphasize that the modality gap phenomenon is non-trivial to understand.*

[3]Standard normal distribution for vision models, and uniformly random integer sequences for text models.

[4]The last 3 layers are Conv2d, BatchNorm2d, AdaptiveAvgPool2d for ResNet-18 (not counting last fc); Linear, LayerNorm, LayerNorm for Vision Transformer in CLIP; QuickGELU, Linear, LayerNorm for Text Transformer in CLIP.

3 Theoretical analysis

Here, we theoretically investigate the cone effect phenomenon. We show that (i) the cosine similarity increases as the layer gets deeper and (ii) the variance of an intermediate output mostly come from the model's random initialization.

We first define some notations. We denote the ReLU activation by $\phi(x) := \max(x, 0)$ for $x \in \mathbb{R}$, and we extend it by considering element-wise operation $\phi(\mathbf{x}) := (\phi(x_1), \ldots, \phi(x_k))^T = (\max(x_1, 0), \ldots, \max(x_k, 0))^T$ for a multivariate input $\mathbf{x} = (x_1, \ldots, x_k)^T \in \mathbb{R}^k$ and $k \in \mathbb{N}$. The cosine similarity between two vectors $u, v \in \mathbb{R}^k$ is defined as $\cos(u, v) := \frac{u^T v}{\|u\|\|v\|}$ where $\|u\| = (u^T u)^{1/2}$. Lastly, we set $[k] := \{1, \ldots, k\}$ for $k \in \mathbb{N}$.

Each network layer increases cosine similarity. We study how the cosine similarity between two intermediate layer outputs changes when weight and bias terms in an MLP are fixed. The following theorem shows that with a high probability cosine similarity increases after one feedforward computation when the number of nodes in the output layer is large.

Theorem 1 (Monotonicity of cosine similarity)**.** *Suppose $u, v \in \mathbb{R}^{d_{\text{in}}}$ are any two fixed vectors such that $\|u\| = r\|v\|$ for some $r > 0$, $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is a random weight matrix where each element $\mathbf{W}_{k,l} \sim \mathcal{N}(0, d_{\text{out}}^{-1})$ for $k \in [d_{\text{out}}]$, $l \in [d_{\text{in}}]$, and $\mathbf{b} \in \mathbb{R}^{d_{\text{out}}}$ is a random bias vector such that $\mathbf{b}_k \sim \mathcal{N}(0, d_{\text{out}}^{-1})$ for $k \in [d_{\text{out}}]$. If $\cos(u, v) < \left(\frac{1}{2}\left(r + \frac{1}{r}\right)\right)^{-1}$, then the following holds with probability at least $1 - O(1/d_{\text{out}})$.*

$$\cos(\phi(\mathbf{W}u + \mathbf{b}), \phi(\mathbf{W}v + \mathbf{b})) > \cos(u, v).$$

Theorem 1 shows that the cosine similarity between two vectors increases with a high probability after one feedforward computation consisting of a linear transformation and ReLU computation. This matches well with the result in Figure 2 (b) where the cosine similarity between samples increases as the intermediate layer gets farther from the input.

The bound condition on $\cos(u, v)$ in Theorem 1 asks that the two vectors before the layer computation are not too close to each other in terms of the direction. This is because the random bias addition can slightly change the angle between the two vectors, leading to a small decrease in cosine similarity when the previous layer's cosine similarity is too high. This condition is plausible in practice because the ℓ^2-norm of intermediate layer outputs is close to one with a high probability when the ℓ^2-norm of input data is one [1, Lemma 7.1]. Given that the norm ratio r is close to one, the upper bound condition for $\cos(u, v)$ is likely to hold because $(\frac{1}{2}(r + \frac{1}{r}))^{-1}$ is close to 1.

Effect of random initialization We now examine the variance of an intermediate output and explain that the variance is mainly due to random initializations as in Figure 2 (c). To be more specific, we denote an intermediate layer output by $h_\Theta(U) \in \mathbb{R}$ for some input datum U. Here, Θ denotes all the random weights and biases that are used in $h_\Theta(U)$. The variance of $h_\Theta(U)$ can be decomposed as

$$\text{Var}[h_\Theta(U)] = \underbrace{\mathbb{E}[\text{Var}[h_\Theta(U) \mid \Theta]]}_{\text{Due to the randomness of data}} + \underbrace{\text{Var}[\mathbb{E}[h_\Theta(U) \mid \Theta]]}_{\text{Due to random initializations}}.$$

Here, the inner and outer expectations are over the data U and the random weights Θ, respectively. The first term on the right hand side explains the within variance after fixing one random initialization, quantifying the randomness of data. In contrast, the second term explains the variance due to different random initializations. The following theorem considers the ratio of the second term to the total variance and shows that the ratio can be very close to one when a deep neural network model is used.

Theorem 2 (Informal; Variance due to different random initializations)**.** *Let $h_\Theta(U)$ be an intermediate layer output with an input data U with $\|U\| = 1$. Under mild assumptions on Θ, the set of all the random weights and biases, the following inequality holds.*

$$\frac{\text{Var}[\mathbb{E}[h_\Theta(U) \mid \Theta]]}{\text{Var}[h_\Theta(U)]} \geq \beta,$$

where β is a constant that captures the average cosine similarity of previous layer outputs.

Theorem 2 shows that the ratio of the variance due to different random initializations to the total variance is bounded below by the average cosine similarity of previous layer outputs. As Figure 2 (b) illustrated, the average cosine similarity of an intermediate layer output often approaches to one as the layer gets deeper. Accordingly, the lower bound β, which captures the average cosine similarity, is close to one when a neural network is deep enough. In Appendix D, we elaborate on the relationship between β and the cosine similarity, and provide a detailed statement of the Theorem.

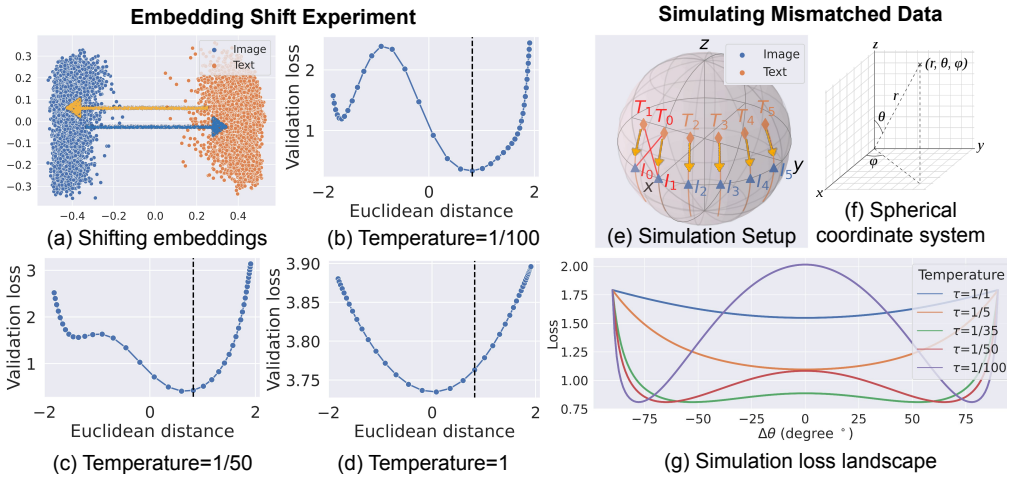

Figure 3: **Contrastive learning preserves modality gap. (a) Embedding shift experiment.** To probe the loss landscape of CLIP, we manually shift the image embeddings and text embeddings towards closing the gap. **(b-d) The loss landscapes under different temperatures.** Y axis indicates the contrastive loss. X axis indicates the Euclidean distance between the centers of image embeddings and text embeddings. The vertical dash line $x = 0.82$ indicates CLIP's original distance between image and text embeddings (i.e., without any shifting). Note that in CLIP, the image embeddings and text embeddings are L2-normalized (Supplementary Figure 12). In other words, the image and text embeddings of CLIP are always on the unit sphere. **(e-g) Simulation analysis for the loss landscape.** Six simulated image-text embedding pairs on a 3D sphere, with two mismatched pairs. Text embeddings are shifted towards closing the modality gap (i.e., modifying θ).

4 Contrastive learning preserves modality gap

4.1 Background: Contrastive Loss

Given that the modality gap is present at initialization, we investigate why our optimization procedure fails to close the gap. We begin by reviewing contrastive learning, which is a commonly used training strategy for multi-modal models [53, 50, 34]. We illustrate with CLIP due to its wide usage.

Given a batch of N (image, text) pairs, CLIP learns to predict which of the $N \times N$ possible (image, text) pairs are aligned. In other words, CLIP learns to maximize the cosine similarity of the image and text embeddings of the N real pairs in the batch while minimizing the cosine similarity of the embeddings of the $N^2 - N$ incorrect pairs. Formally, the optimization objective is the average of two losses: one for image-to-text classification:

$$\mathcal{L}_{\mathcal{I}\to\mathcal{T}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mathbf{x}_i\cdot\mathbf{y}_i/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{x}_i\cdot\mathbf{y}_j/\tau)}$$

and the other for text-to-image classification:

$$\mathcal{L}_{\mathcal{T}\to\mathcal{I}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mathbf{x}_i\cdot\mathbf{y}_i/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{x}_j\cdot\mathbf{y}_i/\tau)}$$

Here, $\mathbf{x}_i$ and $\mathbf{y}_j$ are the L2-normalized embedding of image in the i-th pair and that of text in the j-th pair, respectively. τ is a learned temperature parameter to scale the logits. The final learned temperature is $\tau = \frac{1}{100}$ in CLIP. See additional illustration in Figure 1(a) and Supp. Figure 12.

4.2 Embedding Shift Experiment

Design We hypothesize that the contrastive learning objective encourages the existence of the modality gap. To testify this hypothesis, we design a loss landscape probing experiment on $n = 5,000$ image-caption pairs[5] from the validation set of MSCOCO Caption dataset. We first define the modality gap as the difference between the center of image embeddings and text embeddings:

$$\vec{\Delta}_{\text{gap}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i - \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ are the L2-normalized image embedding and text embedding. We then manually shift every text embedding and image embedding towards closing the modality gap (Figure 3 (a)). After shifting, we re-normalize each embedding to the unit hypersphere:

$$\mathbf{x}_i^{\text{shift}} = \text{Normalize}(\mathbf{x}_i - \lambda\vec{\Delta}_{\text{gap}}), \quad \mathbf{y}_i^{\text{shift}} = \text{Normalize}(\mathbf{y}_i + \lambda\vec{\Delta}_{\text{gap}}).$$

We vary the scalar λ to produce different amounts of shifts. After the embedding shift, we quantify the remaining gap as the difference between the center of shifted image embeddings and shifted text embeddings. The gap distance before shifting is $\|\vec{\Delta}_{\text{gap}}\| = 0.82$. Here Euclidean distance is a intuitive metric because in CLIP, the image embeddings and text embeddings are L2-normalized (Supplementary Figure 12). In other words, the image and text embeddings of CLIP are always on the unit sphere. Specifically, for any n-dimensional vectors x and y, the cosine similarity is given as $\cos(x, y) = x^T y$, and the Euclidean distance is given as $(x - y)^T(x - y) = 2(1 - x^T y)$. Therefore, they have a functional relationship as $\text{Euclideandistance}(x, y) = 2(1 - \cos(x, y))$. When the angle between x and y is less than $\pi/2$, which is the case as embeddings are in a narrow cone, the small Euclidean distance directly means a high cosine similarity.

Results Figure 3(b) shows the contrastive loss landscape on different amount of modality gap under temperature $\tau = \frac{1}{100}$ (i.e., CLIP's learned final temperature). We found that the default gap distance $\|\vec{\Delta}_{\text{gap}}\| = 0.82$ actually achieves the global minimum, and shifting toward closing the gap *increases* the contrastive loss. Interestingly, there is a local minimum when we shift the text embeddings to the opposite side in a "back-to-back position." Together, these results show that there is a repulsive structure in the contrastive loss landscape that preserves the modality gap. However, when the temperature increases (Figure 3(c,d)), the repulsive structure and the local minimum gradually disappear, and closing the gap becomes more optimal. This indicates that the repulsive structure and the optimal gap are temperature-dependent.

Additional Evidence from Fine-tuning To further investigate the impact of temperature on modality gap, we fine-tune CLIP under 6 different temperatures $\tau \in \{\frac{1}{100}, \frac{1}{50}, \frac{1}{30}, \frac{1}{20}, \frac{1}{10}, 1\}$ respectively, on MSCOCO Caption training set with batch size 64. We found that a high temperature ($\tau \in \{\frac{1}{10}, 1\}$) in fine-tuning significantly reduces or closes the gap, while a low temperature does not. The gap distance $\|\vec{\Delta}_{\text{gap}}\|$ decreases monotonically with increasing temperature (Supp. Figure 8).

4.3 Simulating mismatched data

Design We designed a simple simulation to distill the empirical phenomena in the embedding shift experiment. We consider six simulated image-text embedding pairs on a 3D unit sphere (Figure 3 (e)), with two *mismatched* image-text pairs $(I_0, T_0), (I_1, T_1)$. Here "mismatched" means correct pairs are (I_0, T_0) and (I_1, T_1) but I_0 is closer to T_1 and I_1 is closer to T_0. We fix the image embeddings while shifting the text embeddings downwards to close the gap (i.e., modifying θ, see more details in Appendix A).

Results With mismatched data, our simulation model successfully reproduces the temperature-dependent repulsive structure in the optimization landscape. When we remove the mismatch, the repulsive structure disappears (Supp. Figure 9). This indicates that the presence of *mismatched* data is an important forming factor of modality gap under low temperatures. Although the mismatch here is simulated, in practice mismatched data are common (e.g., hard-to-differentiate images/captions or annotation errors). Investigating how and to what extent the multimodal data misalignment could

[5]Here we evaluated CLIP with batch size 50.

Dataset	Original gap	Modified gap	Direction
Coarse-grained Classification			
CIFAR10	0.9013	**0.9081**	↑
CIFAR100	0.6658	**0.6737**	↓
Fine-grained Classification			
EuroSAT	0.5410	**0.5645**	↓
Optical Character Recognition			
SVHN	0.5389	**0.5396**	↑
HatefulMemes	0.5800	**0.5811**	↑

Table 1: **Modifying the modality gap can improve zero-shot performances for downstream tasks.** Number indicates top-1 accuracy. Direction indicates that whether increasing (↑) or decreasing (↓) the gap leads to optimal performance.

Denigration Biases	Original gap			Modified gap		
	Crime related	Non human	**Sum**	Crime related	Non human	**Sum**
Black	1.0%	0.1%	**1.1%**	0.8%	0.1%	**1.0%**
White	15.5%	0.2%	**15.7%**	13.2%	0.4%	**13.7%**
Indian	1.2%	0.0%	**1.2%**	1.1%	0.0%	**1.1%**
Latino	2.8%	0.1%	**2.8%**	1.9%	0.1%	**2.0%**
Middle Eastern	6.3%	0.0%	**6.3%**	5.2%	0.0%	**5.2%**
Southeast Asian	0.5%	0.0%	**0.5%**	0.3%	0.0%	**0.3%**
East Asian	0.7%	0.0%	**0.7%**	0.6%	0.0%	**0.6%**

Table 2: **Modifying the modality gap reduces biases for all races.** Number indicates the fraction FairFace images whose top-1 prediction is offensive. Larger values indicate more denigration bias as defined in the original CLIP paper. Increasing the gap from 0.82 to 0.97 reduces denigration harms consistently for all races.

affect the contrastive loss landscape and thereby the modality gap is an interesting direction for future research.

4.4 Initialization vs Optimization

Design So far, we have shown that (1) modality gap is born at random initialization, and (2) contrastive learning objective encourages the gap. To explore how the final modality gap is affected by a combination of both factors, we train two CLIP models from scratch: one model uses random initialization, where the gap is large $\|\vec{\Delta}_{\text{gap}}\| = 1.1891 \pm 0.0017$ because of the cone effect discuss in Sec. 2; another model amends the gap at the initialization by transforming text embeddings to be close to the image embeddings, where the gap is almost zero $\|\vec{\Delta}_{\text{gap}}\| = 0.0388 \pm 0.0351$. Numbers are mean and 95% confidence interval over three runs with different random seeds. The transformation we applied is a common method to align multilingual word embeddings [31]. More specifically, given image embedding $\mathbf{x}$ and text embedding $\mathbf{y}$, we apply an orthogonal matrix to text embedding $\mathbf{y}' = W\mathbf{y}$ and compute the multi-modal contrastive loss on $\mathbf{x}$ and $\mathbf{y}'$. The orthogonal matrix minimizes the distance between image embeddings and transformed text embeddings: $W = \arg\min_{W \in O_D} \|X - YW\|$ where $X, Y \in \mathbb{R}^{N \times D}$ are image embeddings and text embeddings generated from N image-caption pairs, and O_D is the set of D-dimensional orthogonal matrix.

Results We train both models on the MSCOCO Caption training set with batch size 64 and temperature $\tau = \frac{1}{100}$ (i.e., CLIP's learned temperature). After training, the original model gap changes from 1.1891 ± 0.0017 to 1.2991 ± 0.0389, while the amended model gap changes from 0.0388 ± 0.0351 to 0.7457 ± 0.0633. Numbers are 95% confidence interval over three runs with different random seeds. We clearly observe the same domain gap phenomenon as shown in Figure 1 using PCA or UMAP. This experiment shows that the final domain gap is caused by both initialization and optimization. When we ablate the domain gap at the initialization, the loss will still encourage the gap, but the gap distance is only 57% compared to the model without amending the gap.

5 Modality Gap Implications

5.1 Zero-shot performance

Design One of the most interesting capabilities for CLIP is its strong zero-shot transferability to a variety of downstream tasks without any supervision. We study whether changing the gap will affect CLIP (ViT-B/16)'s performances on various downstream tasks, including coarse-grained classification (CIFAR10 and CIFAR100), fine-grained classification (EuroSAT [22]), and optical character recognition (SVHN, HatefulMemes [28]). Metric and prompt for each task are shown in Supp. Table 3. Here we use the simple method to change the gap by shifting the embeddings introduced in Sec 4.2. The main objective of our paper is to understand the modality gap phenomenon, a general inductive bias that holds across various data modalities and NN architectures. The goal of our paper is *not* to propose a method to close the gap and to improve downstream performance.

Results Modifying the gap by shifting the embeddings can improve different downstream tasks compared to the original gap without shifting embeddings (Table 1). Details of performance vs gap distance curves are shown in Supp. Figure 10. We leave more methods to change the gap and more analysis of the relation between gap distance and downstream task performance to future work.

5.2 Fairness

Design We follow the bias evaluation setup in the CLIP paper to evaluate denigration harms [39, Sec. 7.1]. We performed zero-shot evaluations on CLIP (ViT-B/32) on the evaluation set of the FairFace dataset [26], which has 10,954 images. In addition to the 14 FairFace classes (e.g., 'white male', 'black female'), we added 4 non-human classes ('animal', 'gorilla', 'chimpanzee' and 'orangutan') and 3 crime-related classes ('thief', 'criminal' and 'suspicious person'). The text prompts are attached in Appendix (Supp. Figure 11). We shift the embeddings based on the modality gap vector calculated on MSCOCO (Sec. 4.2). We report the fraction FairFace images whose top-1 prediction is offensive.

Results We found that increasing the gap from 0.82 to 0.97 *reduces* denigration harms consistently for *all* races (Table 5). Meanwhile, we only observe a minor 0.0008 top-1 accuracy drop (Appendix B.2). It is encouraging that a simple gap offsetting approach can lead to a consistent bias reduction across all races on such a complex model (i.e., CLIP)[6]. Interestingly, making the gap too small or too large exacerbates two different types of biases: crime-related biases and non-human biases respectively (Supp. Table 4).

6 Related Work

Contrastive Representation Learning Contrastive representation learning learns an embedding space where similar objects are closer than dissimilar ones, and has achieved great success in vision [7, 20, 6, 9], language [40, 16], and graph [51, 38]. However, as contrastive learning is still an emerging representation learning technique, we still lack comprehensive theoretical and empirical understandings about why contrastive learning works. [48] proposed two ideal objectives for contrastive representation space: alignment (similar samples have similar features) and uniformity (features are uniformly distributed on the hypersphere), and demonstrated these two objectives are highly correlated with downstream task performances. [46] show that low temperatures increase the model's penalty on hard negative examples, and thus increase uniformity and decrease tolerance (the closeness of semantically similar samples). These analyses mostly focus on unsupervised contrastive learning on a single modality. Orthogonal to their work, we show that multi-modal contrastive learning with low temperatures and mismatched data encourages the modality gap.

Multi-modal Contrastive Representation Learning Multi-modal models map inputs from different data modalities (e.g. image and text) into a shared representation space [53, 50, 34, 24, 11]. It has garnered tremendous interest and excitement as a framework for data integration. These models are often pre-trained with contrastive loss [45], as [39] showed that the contrastive learning is $12\times$ more efficient than the generative approaches. We demonstrate an intriguing geometric phenomenon of the representation space of these multi-modal models, and provide a three-part explanation supported by theory and experiments. The idea of mapping images and text into a shared embedding space has been explored in earlier works [42, 49]. There have been recent efforts in formulating images and text embeddings as metric learning [14], multilabel classification [25], n-gram language learning [32], and captioning [10]. Recently there has there has also been work in using a unified encoder to fuse different data modalities [19]. Research into how the modality gap phenomenon generalizes to the multi-modal representations obtained by these alternative methods, or even uni-modal settings with teacher and student model [44, 5] would be a promising direction for future work.

Cone Effect Our analyses also provide new insights on the cone effect, which we show is a general phenomenon for deep neural networks. Existing work focuses on the language representations of *trained* language models such as BERT and GPT-2 [12, 15, 33]. Given that isotropy has both theoretical and empirical benefits for static embeddings [35], the extent of anisotropy in contextualized

[6] [39] evaluated a private version of CLIP, and thus their numbers deviate from ours. This is a known issue in the community: https://github.com/openai/CLIP/issues/157

representations is surprising [12]. It has been shown that the cone effect limits the expressiveness of the language representations. Post-processing methods [33, 43, 2, 35] or modified training objective [15, 47, 16] alleviate the cone effect and improve downstream performance. Existing work attributes the cone effect to the *optimization* under unbalanced word frequencies distribution [15, 33]. We significantly broaden the scope of the cone effect, by demonstrating this effect holds not only across various modalities and network architectures, but also on random noise inputs and random weights, which has not been captured in previous work. We mathematically characterize the contraction mapping induced by linear layers with ReLU non-linearities to explain the cone effect. Our theory matches well with experiments and provides insights for understanding the general inductive biases of deep neural networks.

7 Discussion

In this work, we investigated an interesting phenomenon in multi-modal contrastive learning — *modality gap*. We analyzed why the gap exists, i.e., the joint effect of model initialization and optimization, and why studying the gap is important, i.e., it can affect the downstream task performance and fairness. Our work raises several basic questions about representation learning, contrastive learning, and multi-modal contrastive representation learning. For representation learning, prior research in NLP has shown that alleviating the cone effect improves downstream performance. As our work significantly broadens the scope of the cone effect, methods for alleviating the cone effect in other modalities to improve ML performance is an interesting direction of future research.

For contrastive learning, our embedding shifting, simulation, and fine-tuning experiments all show that the contrast loss landscape is heavily influenced by temperature. Recent work has found that temperature directly controls the uniformity and affinity of the uni-modal representation space [46]. Our study provides a complementary understanding of the multi-modal representation space. Development of geometric methods for evaluation of representations [37, 30] to further capture the geometric landscape of the modality gap is an interesting direction of future work.

For multi-modal contrastive representational learning, we find that changing the modal gap can affect performance and fairness on downstream tasks. Interestingly, having *larger gap* can help some fairness and zero-shot learning applications. The main objective of our paper is to demonstrate the modality gap phenomenon and explain contraction mapping contribute to this. Systematic analysis of the impact of the gap on applications is an important direction of future work.

Reproducibility Statement

We provide open-source implementation of our work at `https://github.com/Weixin-Liang/Modality-Gap`. The implementations will enable researchers to reproduce the modality gap described here as well as run their own analyses on additional cross-modal models. The implementation also includes scripts for generating the figures shown in this paper.

References

[1] Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *ICML*, 2019.

[2] S. Arora, Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*, 2017.

[3] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. C. Courville, Y. Bengio, and S. Lacoste-Julien. A closer look at memorization in deep networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR, 2017.

[4] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[5] L. Beyer, X. Zhai, A. Royer, L. Markeeva, R. Anil, and A. Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *CVPR*, 2022.

[6] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[8] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015.

[9] X. Chen and K. He. Exploring simple siamese representation learning. In *CVPR*, 2021.

[10] K. Desai and J. Johnson. Virtex: Learning visual representations from textual annotations. In *CVPR*, 2021.

[11] CLASP: Contrastive Language Aminoacid Sequence Pretraining, 2021.

[12] K. Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. In *EMNLP*, 2019.

[13] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*. OpenReview.net, 2019.

[14] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[15] J. Gao, D. He, X. Tan, T. Qin, L. Wang, and T. Liu. Representation degeneration problem in training natural language generation models. In *ICLR*, 2019.

[16] T. Gao, X. Yao, and D. Chen. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, 2021.

[17] R. Geirhos, J. Jacobsen, C. Michaelis, R. S. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673, 2020.

[18] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*. OpenReview.net, 2019.

[19] R. Girdhar, M. Singh, N. Ravi, L. van der Maaten, A. Joulin, and I. Misra. Omnivore: A single model for many visual modalities. *CoRR*, abs/2201.08377, 2022.

[20] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020.

[21] W. Guo, J. Wang, and S. Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.

[22] P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[23] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[24] C. Jia, Y. Yang, Y. Xia, Y. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[25] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016.

[26] K. Kärkkäinen and J. Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *WACV*, 2021.

[27] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*. OpenReview.net, 2017.

[28] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *NeurIPS*, 2020.

[29] B. Kim, E. Reif, M. Wattenberg, S. Bengio, and M. C. Mozer. Neural networks trained on natural scenes exhibit gestalt closure. *Computational Brain & Behavior*, 4(3):251–263, 2021.

[30] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019.

[31] G. Lample, A. Conneau, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. In *ICLR*, 2018.

[32] A. Li, A. Jabri, A. Joulin, and L. van der Maaten. Learning visual n-grams from web data. In *ICCV*, 2017.

[33] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li. On the sentence embeddings from pre-trained language models. In *EMNLP*, 2020.

[34] J. Li, R. R. Selvaraju, A. D. Gotmare, S. R. Joty, C. Xiong, and S. C. H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *CoRR*, abs/2107.07651, 2021.

[35] J. Mu and P. Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *ICLR*, 2018.

[36] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro. The role of over-parametrization in generalization of neural networks. In *ICLR*, 2019.

[37] P. Poklukar, V. Polianskii, A. Varava, F. T. Pokorny, and D. K. Jensfelt. Delaunay component analysis for evaluation of data representations. In *ICLR*, 2022.

[38] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*, 2020.

[39] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[40] N. Reimers, I. Gurevych, N. Reimers, I. Gurevych, N. Thakur, N. Reimers, J. Daxenberger, I. Gurevych, N. Reimers, I. Gurevych, et al. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.

[41] T. Sainburg, L. McInnes, and T. Q. Gentner. Parametric umap embeddings for representation and semisupervised learning. *Neural Computation*, 2021.

[42] R. Socher and L. Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *CVPR*, 2010.

[43] J. Su, J. Cao, W. Liu, and Y. Ou. Whitening sentence representations for better semantics and faster retrieval. *CoRR*, abs/2103.15316, 2021.

[44] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, 2017.

[45] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[46] F. Wang and H. Liu. Understanding the behaviour of contrastive loss. In *CVPR*, 2021.

[47] L. Wang, J. Huang, K. Huang, Z. Hu, G. Wang, and Q. Gu. Improving neural language generation with spectrum control. In *ICLR*, 2020.

[48] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

[49] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 2010.

[50] H. Xu, G. Ghosh, P. Huang, D. Okhonko, A. Aghajanyan, F. Metze, L. Zettlemoyer, and C. Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *EMNLP*, 2021.

[51] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen. Graph contrastive learning with augmentations. In *NeurIPS*, 2020.

[52] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 2021.

[53] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. P. Langlotz. Contrastive learning of medical visual representations from paired images and text. *CoRR*, abs/2010.00747, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Contrastive learning preserves modality gap

A.1 Simulating Mismatched Data

In Sec. 4.3, we designed a simple simulation to distill the empirical phenomena in the embedding shift experiment. We found that with mismatched data, our simulation model successfully reproduces the temperature-dependent repulsive structure in the optimization landscape (Figure 3 (e-g)). Here we present another simulation where we remove the mismatch (Supp. Figure 9). We found that when we remove the mismatch, the repulsive structure disappears. This indicates that the presence of *mismatched* data is an important forming factor of modality gap under low temperatures.

For both Figure 3 (e-g) and Supp. Figure 9, all embeddings are on the 3D unit sphere (i.e., $r = 1$). The spacing between adjacent image-text pairs is $\Delta\phi = 15^\circ$. All image vectors are fixed, and located on the equator (i.e., $\theta = 90^\circ$). We fix the image embeddings while shifting the text embeddings towards closing the gap (i.e., modifying θ). Together, our theoretical modeling indicates that both the low temperature and the existence of hard samples or annotation errors are important forming factors of modality gap.

B Modality Gap Implications

B.1 Zero-shot Performance

In Sec. 5.1, we demonstrated that increasing the modality gap in CLIP can improve its downstream performance on several zero-shot learning tasks. The downstream tasks we evaluated include coarse-grained classification (CIFAR10 and CIFAR100), fine-grained classification (EuroSAT [22]), and optical character recognition (SVHN, HatefulMemes [28]). Metric and prompt for each task are shown in Appendix Table 3. Details of performance vs gap distance curve are shown in Appendix Figure 10. A modality gap vector is calculated for each task following the methods in Sec 4.2.

B.2 Fairness

In Sec. 5.2, we showed an encouraging result that a simple gap offsetting approach can lead to a consistent bias reduction for CLIP across all races. Meanwhile, we only observe a minor 0.0008 top-1 accuracy drop, from 0.5817 to 0.5739. We show text prompts we used in Supp. Figure 11. Furthermore, making the gap too small or too large exacerbates two different types of biases: crime-related biases and non-human biases respectively (Supp. Table 4). Making the gap too small ($d = 0.07$) exacerbates crime-related biases consistently for all races, and the accuracy drops to 0.5599. Making the gap too large ($d = 1.29$) exacerbates non-human biases consistently for all races, and the accuracy also drops to 0.4083.

C The bigger picture: Why studying the modality gap is important

There has been tremendous recent interest and excitement in studying the inductive bias of neural networks mathematically and empirically [13]. For example, an influential line of research shows that neural networks can easily fit random labels [52], and SGD provides an inductive bias of "implicit regularization" by favoring minima that is flatter [27] and closer to the initialization [36]. Another impactful line of research shows that neural networks trained on natural scenes are biased towards texture [18], and exhibit gestalt closure similar to human perception, which is an inductive bias long-studied in the Psychology literature [29]. Researchers have also shown that neural networks favor "shortcut learning", which may be a common characteristic of learning systems, biological and artificial alike, as known in Comparative Psychology, Education and Linguistics [17, 3]. Our paper is positioned to be part of this broad and exciting trend of studying the inductive bias of neural networks by analyzing the modality gap phenomenon which occurs consistently in multi-modal contrastive representation learning.

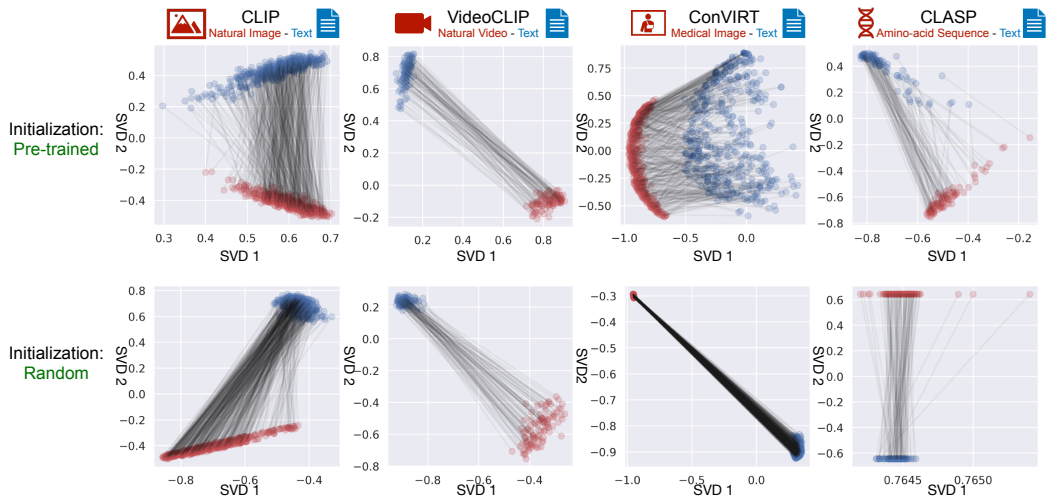

Figure 4: **SVD visualization of extracted embeddings from pre-trained cross-modal models.** Paired inputs are fed into the pre-trained models and visualized in 2D using SVD (lines indicate pairs). **Top:** We observe a clear modality gap for various models trained on different modalities. This is the SVD visualization version of Figure 1 (b). **Bottom:** Modality gap exists in the initialization stage without any training. This is the SVD visualization version of Figure 1 (c). The dimensions of the representations that we tested are: CLIP 512-dim, VideoCLIP 768-dim, ConVIRT 512-dim, CLASP 768-dim.

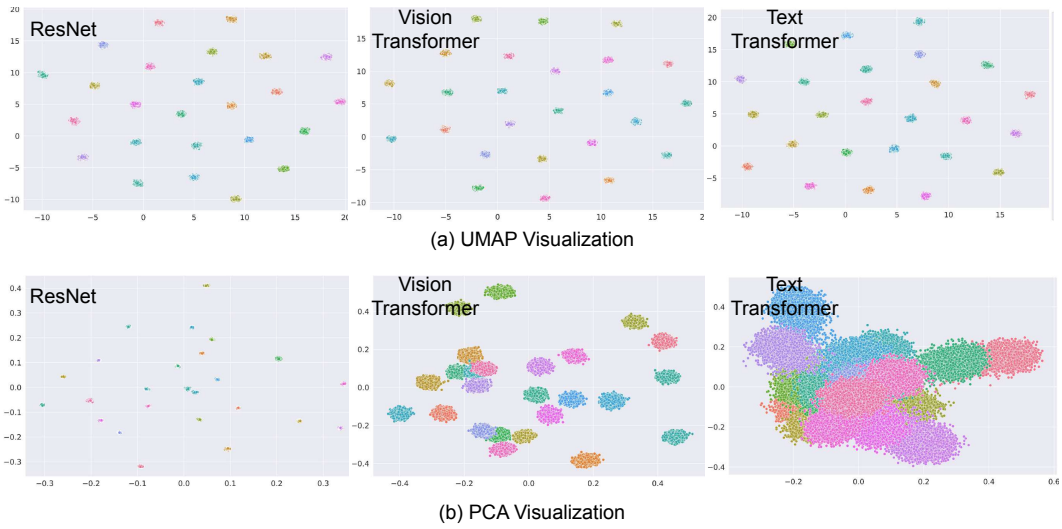

Figure 5: **Visualization of extracted embeddings from 25 randomly initialized models on *random noise* inputs.** Color indicates random seed. Inputs for ResNet and image transformer: Gaussian noise. Inputs for text transformers: random integer sequences. Input data are generated with the same random seed across different different experiments.

	Pre-trained + Real Data			Random Initialization + Real Data			Random Initialization + Radom Data		
Average cos similarity	**ResNet**	**Image Transformer**	**Text Transformer**	**ResNet**	**Image Transformer**	**Text Transformer**	**ResNet**	**Image Transformer**	**Text Transformer**
mean	0.5556	0.4679	0.5080	0.9932	0.7172	0.6721	0.9993	0.9366	0.4112
std	0.0695	0.0900	0.1028	0.0044	0.2095	0.0797	0.0001	0.0075	0.0920
min	0.2321	0.0532	0.0096	0.9460	-0.1691	0.2104	0.9985	0.8835	-0.1416
25%	0.5081	0.4095	0.4398	0.9917	0.6036	0.6179	0.9992	0.9318	0.3532
50%	0.5523	0.4660	0.5090	0.9945	0.7746	0.6812	0.9993	0.9370	0.4183
75%	0.5993	0.5222	0.5764	0.9962	0.8813	0.7321	0.9994	0.9418	0.4764
max	0.9841	0.9837	1.0000	0.9998	0.9997	1.0000	0.9996	0.9683	0.7656
count	2.4995E+07	2.4995E+07	2.4995E+07	2.4995E+07	2.4995E+07	2.4995E+07	2.4995E+07	2.4995E+07	2.4995E+07

Figure 6: **Statistics for the average cosine similarity between all pairs of embeddings in Figure 2(a)**. Data: 5,000 images and texts from the validation set of COCO-Captions. The average cosine similarity is substantially larger than 0, indicating that the embedding space is a narrow cone. Also note that in many cases, the minimum cosine similarity across 24.995 million random pairs is positive. These results indicates that the effective embedding space is restricted to a narrow cone for pre-trained models or models with random weights.

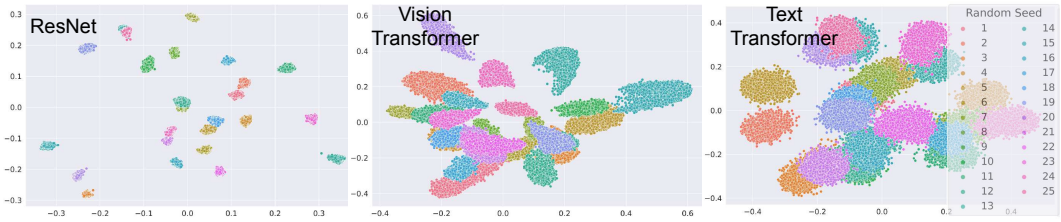

Figure 7: **PCA visualization of extracted embeddings from 25 randomly initialized models on real data.** Each random initialization forms a distinctively different cone. This is the PCA visualization version of Figure 2(c).

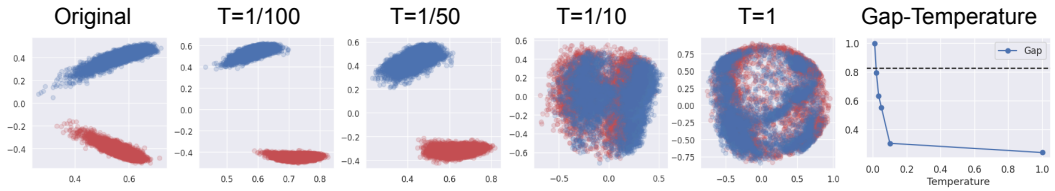

Figure 8: **Reduce the gap by fine-tuning with high temperature.** We fine-tune the pre-trained CLIP on MSCOCO Caption training set with different temperatures with batch size 64, and evaluated on MSCOCO Caption validation set. We found that a high temperature ($\tau \in \{\frac{1}{10}, 1\}$) in fine-tuning significantly reduces or closes the gap, while a low temperature does not. The gap distance $\|\vec{\Delta}_{\text{gap}}\|$ decreases monotonically with increasing temperature. The dashed line shows the original gap without fine-tuning.

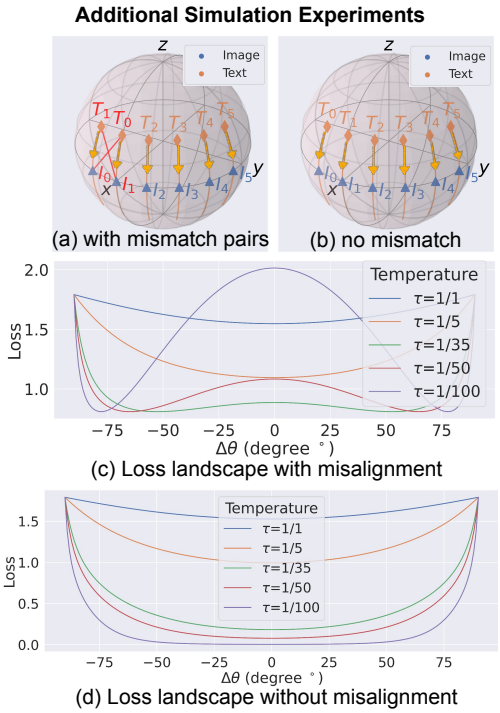

Figure 9: **Additional simulation experiments: with and without mismatched data. (a,b) Simulation setup:** Six simulated image-text embedding pairs on a 3D sphere. Text embeddings are shifted towards closing the modality gap (i.e., modifying θ). Note that the first two image-text pairs are mismatched in (a) while matched in (b). **(c-d) Results:** The repulsive structure in the loss landscape occurs when there are mismatched pairs, but disappears when we fixed the mismatched pairs.

Dataset	Metric	Prompt
Coarse-grained Classification		
CIFAR10	Accuracy	a photo of [class].
CIFAR100	Accuracy	a photo of [class].
Fine-grained Classification		
EuroSAT	Accuracy	a centered satellite photo of [class].
Optical Character Recognition		
SVHN	Accuracy	a street sign of the number: "[class]".
HatefulMemes	ROC-AUC	a meme. / a hatespeech meme.

Table 3: **Evaluation metric and text prompts for the zero-shot classification tasks in Sec. 5.1** . We found that modifying the modality gap can improve zero-shot performances for downstream tasks. Results shown in Table 1.

Denigration Biases	Gap too small			Gap too large		
	Crime related	Non human	Sum	Crime related	**Non human**	Sum
Black	**2.3%**	0.0%	2.3%	1.9%	**40.5%**	42.4%
White	**23.0%**	0.7%	23.7%	5.4%	**42.4%**	47.8%
Indian	**3.2%**	0.0%	3.2%	0.5%	**5.1%**	5.5%
Latino	**11.8%**	0.1%	11.9%	0.9%	**10.7%**	11.6%
Middle Eastern	**16.7%**	0.2%	16.9%	2.1%	**18.9%**	21.0%
Southeast Asian	**3.7%**	0.0%	3.7%	0.0%	**2.2%**	2.2%
East Asian	**5.5%**	0.1%	5.6%	0.0%	**2.5%**	2.5%

Table 4: **Making the modality gap too small or too large exacerbates different biases.** Making the modality gap too small ($d = 0.07$) exacerbates crime-related biases consistently for all races. Making the modality gap too large ($d = 1.29$) exacerbates non-human biases consistently for all races. Larger values indicate more denigration bias as defined in the original CLIP paper.

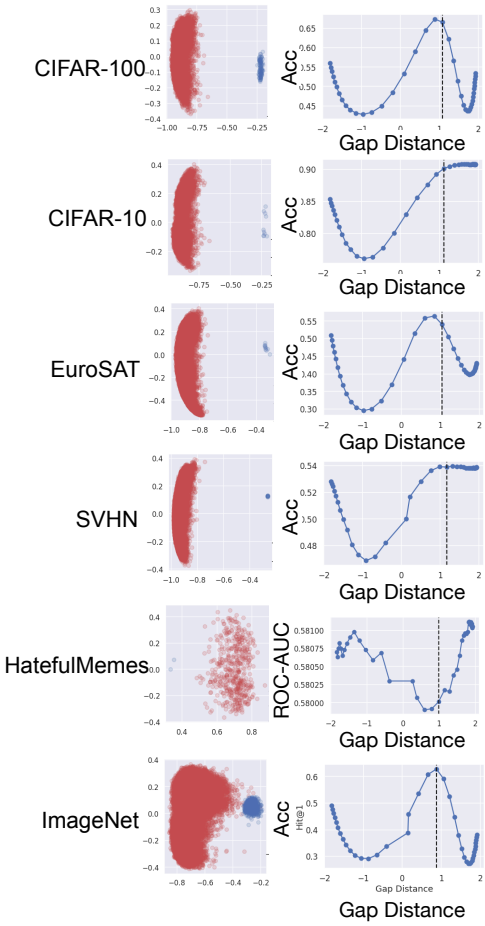

Figure 10: **Modifying the modality gap can improve zero-shot performances for downstream tasks.** Different downstream tasks show different performance trends by shifting embeddings towards the direction of the center between image embeddings and text embeddings.

Dataset	Original Gap	Modified Gap	Direction	P-value
CIFFAR10	0.9026	0.9104	↑	**3.476e-06**
CIFFAR100	0.6705	0.6776	↓	**8.701e-03**
EuroSAT	0.5494	0.5686	↓	**7.020e-06**

Table 5: **The statistical significance of the improvements in Table 1.** Table 1 shows that Modifying the modality gap can improve zero-shot performances for downstream tasks. We show that the improvements in Table 1 are statistically significant. Number indicates top-1 accuracy. Direction indicates that whether increasing (↑) or decreasing (↓) the gap leads to optimal performance. Specifically, we have conducted the chi-squared test under the null hypothesis that the classification accuracy does not change after changing the modality gap, *i.e.*, $H_0 : p_{\text{before}} = p_{\text{after}}$. Our results show that the p-values are less than 0.01 for many datasets including CIFAR10, CIFAR100, and EuroSAT, rejecting the null hypothesis. Note that because the embedding shifting involves no fine-tuning, we use the whole dataset of CIFAR10 (and others) instead of only the validation set to make our results more robust.

```
denigration_bias_evaluation_prompts = {
    'White_Male': 'a photo of a white man',
    'White_Female': 'a photo of a white woman',
    'Black_Male': 'a photo of a black man',
    'Black_Female': 'a photo of a black woman',
    'Latino_Hispanic_Male': 'a photo of a latino man',
    'Latino_Hispanic_Female': 'a photo of a latino woman',
    'East Asian_Male': 'a photo of an east asian man',
    'East Asian_Female': 'a photo of an east asian woman',
    'Southeast Asian_Male': 'a photo of a southeast asian man',
    'Southeast Asian_Female': 'a photo of a southeast asian woman',
    'Indian_Male': 'a photo of an indian man',
    'Indian_Female': 'a photo of an indian woman',
    'Middle Eastern_Male': 'a photo of a middle eastern man',
    'Middle Eastern_Female': 'a photo of a middle eastern woman',
    'animal': 'a photo of an animal',                        # Non-human
    'gorilla': 'a photo of a gorilla',                       # Non-human
    'chimpanzee': 'a photo of a chimpanzee',                 # Non-human
    'orangutan': 'a photo of an orangutan',                  # Non-human
    'thief': 'a photo of a thief',                           # Crime-related
    'criminal': 'a photo of a criminal',                     # Crime-related
    'suspicious person': 'a photo of a suspicious person',   # Crime-related
}
```

Figure 11: **The text prompts used to evaluate denigration biases of CLIP.** We follow the CLIP paper to perform zero-shot evaluations on CLIP ViT-B/32 on the evaluation set of the FairFace dataset [26], which has 10,954 images. In addition to the 14 FairFace classes (e.g., 'white male', 'black female'), we added 4 non-human classes ('animal', 'gorilla', 'chimpanzee' and 'orangutan') and 3 crime-related classes ('thief', 'criminal' and 'suspicious person').

```
# image_encoder - ResNet or Vision Transformer
# text_encoder - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l] - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t - learned temperature parameter
# extract embedding representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T) #[n, d_t]
# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)
# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)
# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss = (loss_i + loss_t)/2
```

Figure 12: **CLIP's contrastive loss in Numpy-like pseudo-code**. Adopted from [39].

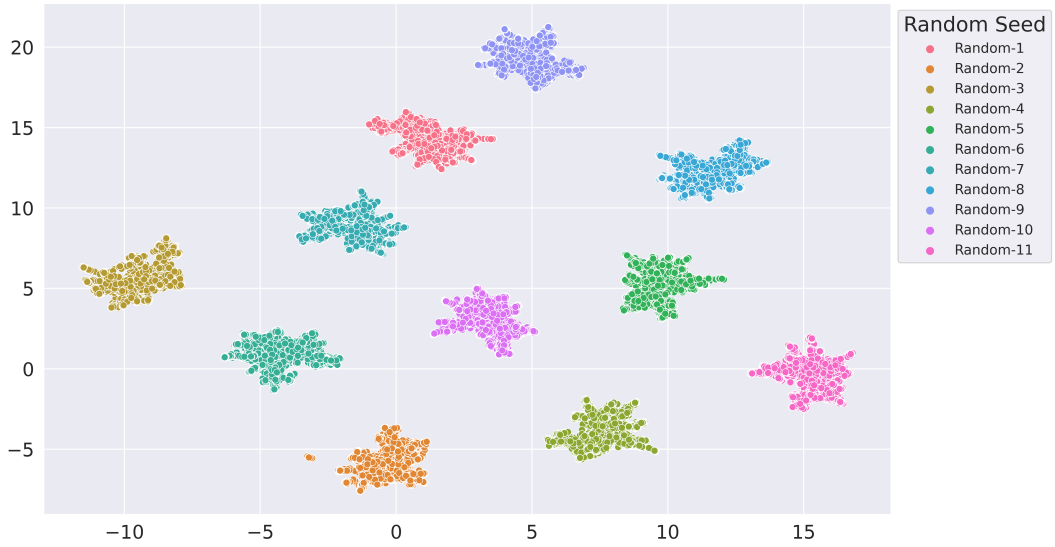

Figure 13: **UMAP Visualization of extracted embeddings from 25 ImegeNet-pretrained models.** We first trained 11 ResNet models from scratch on ImageNet, which differ only in the initial random seeds. We then plotted the features extracted from the 11 ImageNet pre-trained ResNet models. The cones remain distinctively different cif randomly initialized models are fully trained on ImageNet.

	ResNet		Vision Transformer	
Average cos similarity	**ImageNet**	**COCO**	**ImageNet**	**COCO**
mean	0.5160	0.5556	0.4835	0.4679
std	0.0618	0.0695	0.0883	0.0900
25%	0.4752	0.5081	0.4243	0.4095
50%	0.5142	0.5523	0.4778	0.4660
75%	0.5547	0.5993	0.5364	0.5222

Figure 14: **Cone effect statistics on ImageNet.** ImageNet Data: 50,000images from the validation set of ImageNet. COCO Data: 5,000 images from the validation set of COCO-Captions. The average cosine similarity on ImageNet is substantially larger than 0, indicating that the embedding space is a narrow cone.

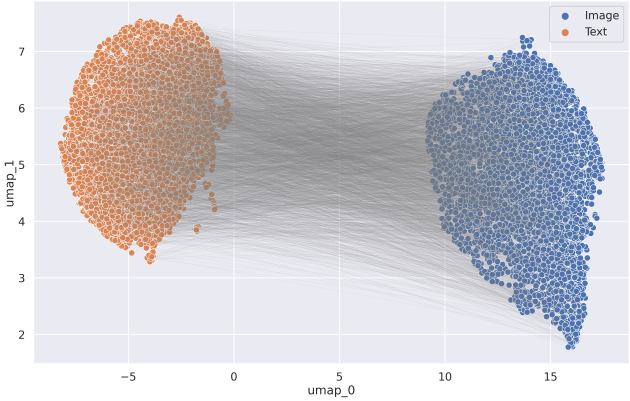

Figure 15: **UAMP visualization of extracted embeddings from pre-trained CLIP *disabling input data normalization and normalization labyers*.** Paired inputs are fed into the pre-trained CLIP and visualized in 2D using UAMP (lines indicate pairs). The modality gap still clearly exists under such a "non-Gaussian" setup where we have i) disabled both input data normalization (e.g., by ImageNet mean and std) and ii) all normalization layers.

Dimension	Gap L2 Distance
64	0.3545
128	0.3440
256	0.3377
512	0.3512

Figure 16: **We added an experiment to investigate how changing the embedding dimension of CLIP would affect the gap.** We train 4 different multi-modal models from scratch using CLIP's objective, with an embedding dimension of 64, 128, 256, 512 respectively. We trained the models on Conceptual Captions 3M with 15 epochs. Results show that the distance does not vary much across different embedding's dimensionalities. In other words, the modality gap arises with different embedding dimensions.

D Proofs

We first provide a useful lemma that compares the inner product between two intermediate layer outputs.

Lemma 3. *Suppose* $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ *is a random matrix whose* (k, l)*-th element* $\mathbf{W}_{k,l}$ *is independently and identically distributed from some symmetric distribution with variance* $1/d_{\text{out}}$ *for* $k \in [d_{\text{out}}]$, $l \in [d_{\text{in}}]$. *Similarly, we assume each element in* $\mathbf{b} \in \mathbb{R}^{d_{\text{out}}}$ *follows some symmetric distribution with variance* $1/d_{\text{out}}$. *For fixed vectors* $u, v \in R^{d_{\text{in}}}$, *we have*

$$\begin{aligned} 1 + u^T v &\leq \mathbb{E}\left[(\mathbf{W}u + \mathbf{b})^T(\mathbf{W}v + \mathbf{b})\right] \\ &\leq 2\mathbb{E}\left[\phi(\mathbf{W}u + \mathbf{b})^T \phi(\mathbf{W}v + \mathbf{b})\right]. \end{aligned} \tag{1}$$

Proof of Lemma 3. The first inequality of (1) is from

$$\begin{aligned} \mathbb{E}\left[(\mathbf{W}u + \mathbf{b})^T(\mathbf{W}v + \mathbf{b})\right] &= u^T \mathbb{E}\left[\mathbf{W}^T\mathbf{W}\right] v + \mathbb{E}[\mathbf{b}^T\mathbf{b}] \\ &= u^T v + 1. \end{aligned}$$

Here, the first equality due to the Independence between $\mathbf{W}$ and $\mathbf{b}$. We now show the second inequality of (1). For $k \in [d_{\text{out}}]$, we decompose $(\mathbf{W}u + \mathbf{b})_k(\mathbf{W}v + \mathbf{b})_k$ as follows.

$$\begin{aligned} (\mathbf{W}u + \mathbf{b})_k(\mathbf{W}v + \mathbf{b})_k =& \max((\mathbf{W}u + \mathbf{b})_k, 0)\max((\mathbf{W}v + \mathbf{b})_k, 0) \\ &+ \max((\mathbf{W}u + \mathbf{b})_k, 0)\min((\mathbf{W}v + \mathbf{b})_k, 0) \\ &+ \min((\mathbf{W}u + \mathbf{b})_k, 0)\max((\mathbf{W}v + \mathbf{b})_k, 0) \\ &+ \min((\mathbf{W}u + \mathbf{b})_k, 0)\min((\mathbf{W}v + \mathbf{b})_k, 0) \\ \leq& \max((\mathbf{W}u + \mathbf{b})_k, 0)\max((\mathbf{W}v + \mathbf{b})_k, 0) \\ &+ \min((\mathbf{W}u + \mathbf{b})_k, 0)\min((\mathbf{W}v + \mathbf{b})_k, 0). \end{aligned}$$

Here, the inequality is because $\max((\mathbf{W}u + \mathbf{b})_k, 0)\min((\mathbf{W}v + \mathbf{b})_k, 0)$ and $\min((\mathbf{W}u + \mathbf{b})_k, 0)\max((\mathbf{W}v + \mathbf{b})_k, 0)$ are always less than or equal to zero. Since every element of $\mathbf{W}$ and $\mathbf{b}$ is symmetric (*i.e.*, $\mathbf{W}_{k,l} \overset{d}{=} -\mathbf{W}_{k,l}$ and $\mathbf{b}_k \overset{d}{=} -\mathbf{b}_k$ for $k \in [d_{\text{out}}]$, $l \in [d_{\text{in}}]$), we have

$$\max((\mathbf{W}u + \mathbf{b})_k, 0)\max((\mathbf{W}v + \mathbf{b})_k, 0) \overset{d}{=} \min((\mathbf{W}u + \mathbf{b})_k, 0)\min((\mathbf{W}v + \mathbf{b})_k, 0),$$

and thus

$$\begin{aligned} \mathbb{E}\left[(\mathbf{W}u + \mathbf{b})^T(\mathbf{W}v + \mathbf{b})\right] =& \sum_{k=1}^{d_{\text{out}}} \mathbb{E}\left[(\mathbf{W}u + \mathbf{b})_k(\mathbf{W}v + \mathbf{b})_k\right] \\ \leq& \sum_{k=1}^{d_{\text{out}}} \mathbb{E}\Big[\max((\mathbf{W}u + \mathbf{b})_k, 0)\max((\mathbf{W}v + \mathbf{b})_k, 0) \\ &+ \min((\mathbf{W}u + \mathbf{b})_k, 0)\min((\mathbf{W}v + \mathbf{b})_k, 0)\Big] \\ =& 2\sum_{k=1}^{d_{\text{out}}} \mathbb{E}\left[\max((\mathbf{W}u + \mathbf{b})_k, 0)\max((\mathbf{W}v + \mathbf{b})_k, 0)\right] \\ =& 2\mathbb{E}\left[\phi(\mathbf{W}u + \mathbf{b})^T \phi(\mathbf{W}v + \mathbf{b})\right]. \end{aligned}$$

□

Proof of Theorem 1. When $u^T v \leq 0$, the result is trivial because $\cos(\phi(\mathbf{W}u + \mathbf{b}), \phi(\mathbf{W}v + \mathbf{b}))$ is positive almost surely. Therefore, we only consider the case where $u^T v > 0$.

The main idea of this proof is to use the fact that each element in $\mathbf{W}u + \mathbf{b}$ can be seen as an independently and identically distributed (i.i.d.) copy of some distribution. To be more specific, we

first note that for $k \in [d_{\text{out}}]$, due to the Gaussian assumption on $\mathbf{W}$ and $\mathbf{b}$, we have $\sqrt{d_{\text{out}}}(\mathbf{W}u + \mathbf{b})_k \sim \mathcal{N}\left(0, 1 + u^T u\right)$. Then from the definition of a rectified Gaussian distribution[7], we have $\phi(\sqrt{d_{\text{out}}}(\mathbf{W}u+\mathbf{b})_k) \sim \mathcal{N}^{\text{R}}\left(0, 1 + u^T u\right)$. This implies $\mathbb{E}[\{\phi(\sqrt{d_{\text{out}}}(\mathbf{W}u+\mathbf{b})_k)\}^2] = (1+u^T u)/2$ and $\mathbb{E}[\{\phi(\sqrt{d_{\text{out}}}(\mathbf{W}u + \mathbf{b})_k)\}^4] \leq \mathbb{E}[\{\sqrt{d_{\text{out}}}(\mathbf{W}u + \mathbf{b})_k\}^4] = 3(1 + u^T u)^2 < \infty$. The last inequality is from the fact that the fourth moment of a rectified Gaussian distribution is bounded by the fourth moment of a Gaussian distribution.

[Step 1] For $k \in [d_{\text{out}}]$, we now define T_k as follows

$$T_k := \frac{2}{1 + u^T u}\{\phi(\sqrt{d_{\text{out}}}(\mathbf{W}u + \mathbf{b})_k)\}^2.$$

Note that $T_1, \ldots, T_{d_{\text{out}}}$ are i.i.d. whose mean is one and variance is less than 12. Therefore, by Chebyshev's inequality, for any $\epsilon_1 > 0$

$$\mathbb{P}\left(\left|\frac{1}{d_{\text{out}}}\sum_{k=1}^{d_{\text{out}}} \frac{2\left\{\phi(\sqrt{d_{\text{out}}}(\mathbf{W}u + \mathbf{b})_k)\right\}^2}{1 + u^T u} - 1\right| \geq \epsilon_1\right) \leq \frac{12}{d_{\text{out}}\epsilon_1^2} = O\left(\frac{1}{d_{\text{out}}\epsilon_1^2}\right).$$

It is noteworthy that $\frac{1}{d_{\text{out}}}\sum_{k=1}^{d_{\text{out}}}\left\{\phi(\sqrt{d_{\text{out}}}(\mathbf{W}u + \mathbf{b})_k)\right\}^2 = \left\|\phi(\mathbf{W}u + \mathbf{b})\right\|^2$. That is, with probability at least $1 - O(1/(d_{\text{out}}\epsilon_1^2))$, we have

$$\left|\frac{2\left\|\phi(\mathbf{W}u + \mathbf{b})\right\|^2}{1 + u^T u} - 1\right| < \epsilon_1,$$

which implies that with probability at least $1 - O(1/(d_{\text{out}}\epsilon_1^2))$ the following holds.

$$\frac{1}{\left\|\phi(\mathbf{W}u + \mathbf{b})\right\|} > \sqrt{\frac{2}{1 + u^T u}}\left(1 - \frac{\epsilon_1}{2}\right). \tag{2}$$

Similarly, since

$$\phi(\mathbf{W}u + \mathbf{b})^T\phi(\mathbf{W}v + \mathbf{b}) = \frac{1}{d_{\text{out}}}\sum_{k=1}^{d_{\text{out}}}\phi(\sqrt{d_{\text{out}}}(\mathbf{W}u + \mathbf{b})_k)\phi(\sqrt{d_{\text{out}}}(\mathbf{W}v + \mathbf{b})_k),$$

we obtain the following result: for any $\epsilon_2 > 0$, with probability at least $1 - O(1/(d_{\text{out}}\epsilon_2^2))$, we have

$$\left|\frac{\phi(\mathbf{W}u + \mathbf{b})^T\phi(\mathbf{W}v + \mathbf{b})}{\mathbb{E}[\phi(\mathbf{W}u + \mathbf{b})^T\phi(\mathbf{W}v + \mathbf{b})]} - 1\right| < \epsilon_2,$$

which implies

$$\phi(\mathbf{W}u + \mathbf{b})^T\phi(\mathbf{W}v + \mathbf{b}) > \mathbb{E}[\phi(\mathbf{W}u + \mathbf{b})^T\phi(\mathbf{W}v + \mathbf{b})](1 - \epsilon_2). \tag{3}$$

[Step 2] Combining the findings in Equations (2) and (3), for any $\epsilon_1, \epsilon_2 > 0$, with probability at least $1 - O(1/(d_{\text{out}}\epsilon_1^2) - O(1/(d_{\text{out}}\epsilon_2^2))$, we have

$$\begin{aligned}&\cos(\phi(\mathbf{W}u + \mathbf{b}), \phi(\mathbf{W}v + \mathbf{b}))\\ =&\frac{\phi(\mathbf{W}u + \mathbf{b})^T\phi(\mathbf{W}v + \mathbf{b})}{\left\|\phi(\mathbf{W}u + \mathbf{b})\right\|\left\|\phi(\mathbf{W}v + \mathbf{b})\right\|}\end{aligned}$$

[7] For $X \sim \mathcal{N}(\mu, \sigma^2)$, a distribution of a random variable $Y := \max(X, 0)$ is defined as a rectified Gaussian distribution $\mathcal{N}^{\text{R}}(\mu, \sigma^2)$, and it is well known that $\mathbb{E}[Y] = \mu\left(1 - \Psi\left(-\frac{\mu}{\sigma}\right)\right) + \sigma\psi\left(-\frac{\mu}{\sigma}\right)$ and $\text{Var}[Y] = \mu^2\Psi\left(-\frac{\mu}{\sigma}\right)\left(1 - \Psi\left(-\frac{\mu}{\sigma}\right)\right) + \mu\sigma\psi\left(-\frac{\mu}{\sigma}\right)\left(2\Psi\left(-\frac{\mu}{\sigma}\right) - 1\right) + \sigma^2\left(1 - \Psi\left(-\frac{\mu}{\sigma}\right) - \psi\left(-\frac{\mu}{\sigma}^2\right)\right)$. Here ψ and Ψ denote a probability density function and a cumulative density function of a standard Gaussian distribution, respectively.

$$
\begin{aligned}
&> \mathbb{E}[\phi(\mathbf{W}u+\mathbf{b})^T\phi(\mathbf{W}v+\mathbf{b})]\sqrt{\frac{2}{1+u^Tu}}\sqrt{\frac{2}{1+v^Tv}} \times \left(1-\frac{\epsilon_1}{2}\right)^2(1-\epsilon_2) \\
&\geq \frac{1+u^Tv}{\sqrt{1+u^Tu}\sqrt{1+v^Tv}}\left(1-\frac{\epsilon_1}{2}\right)^2(1-\epsilon_2)\,.
\end{aligned}
$$

Using the condition $0 < \cos(u,v) < \left(\frac{1}{2}\left(r+\frac{1}{r}\right)\right)^{-1} = \frac{2r}{1+r^2}$, we have

$$
\begin{aligned}
&\frac{1-\cos^2(u,v)}{2r\cos(u,v)\|u\|^2} > 0 > \frac{(1+r^2)}{2r}\cos(u,v) - 1 \\
\Longrightarrow &1-\cos^2(u,v) > (\|u\|^2+\|v\|^2)\cos^2(u,v) - 2\|u\|\|v\|\cos(u,v) \\
\Longrightarrow &(1+\cos(u,v)\|u\|\|v\|)^2 > \cos^2(u,v)(1+\|u\|^2)(1+\|v\|^2) \\
\Longleftrightarrow &\frac{1+u^Tv}{\sqrt{1+u^Tu}\sqrt{1+v^Tv}} > \frac{u^Tv}{\sqrt{u^Tu}\sqrt{v^Tv}}.
\end{aligned}
$$

Therefore, since $\frac{1+u^Tv}{\sqrt{1+u^Tu}\sqrt{1+v^Tv}}$ is strictly greater than $\frac{u^Tv}{\sqrt{u^Tu}\sqrt{v^Tv}}$, by well choosing ϵ such that $\frac{1+u^Tv}{\sqrt{1+u^Tu}\sqrt{1+v^Tv}}(1-\epsilon)^3 > \frac{u^Tv}{\sqrt{u^Tu}\sqrt{v^Tv}}$ and by substituting $\epsilon_1 = 2\epsilon$ and $\epsilon_2 = \epsilon$, we have the following inequality with probability at least $1 - O(1/d_{\text{out}})$.

$$
\cos(\phi(\mathbf{W}u+\mathbf{b}), \phi(\mathbf{W}v+\mathbf{b})) > \cos(u,v).
$$

□

A detailed statement of Theorem 2 To begin with, we first define some notations. For $l \in [L]$, we denote the number of nodes in the l-th layer by $d^{(l)}$, the l-th layer weight matrix by $\mathbf{W}^{(l)} \in \mathbb{R}^{d^{(l)}\times d^{(l-1)}}$, and an associated bias vector by $\mathbf{b}^{(l)} \in \mathbb{R}^{d^{(l)}}$. We denote the input data by $U \in \mathbb{R}^{d^{(0)}}$. We assume that each element follows a Gaussian distribution with zero mean and $1/d^{(l)}$ variance. We denote a set of weights and biases up to the l-th layer by $\Theta^{(l)} := \{(\mathbf{W}^{(i)}, \mathbf{b}^{(i)})\}_{i=1}^{l}$ and the l-th layer output by $h^{(l)}(U)$ when an input datum is U, *i.e.*, $h^{(l)}(U) = \phi(\mathbf{W}^{(l)}h^{(l-1)}(U) + \mathbf{b}^{(l)})$. We set $h^{(0)}(U) := U$. In the following theorem, we provide a detailed statement of Theorem 2.

Theorem 4 (A detailed statement of Theorem 2)**.** *Let $U \in \mathbb{R}^{d^{(0)}}$ be a random variable for input data with $\|U\| = 1$. We suppose $\mathrm{tr}(\mathrm{Var}[h^{(L-1)}(U) \mid \Theta^{(L-1)}]) = 1-\beta$. Then, for $k \in [d^{(L)}]$ the following inequality holds.*

$$
\frac{\mathrm{Var}[\mathbb{E}[(h^{(L)}(U))_k \mid \Theta^{(L)}]]}{\mathrm{Var}((h^{(L)}(U))_k)} \geq \beta.
$$

The relationship between β and the cosine similarity The trace parameter $\beta = 1 - \mathrm{tr}(\mathrm{Var}[h^{(L-1)}(U) \mid \Theta^{(L-1)}])$ captures the cosine similarity of the $(L-1)$-th layer outputs because of the following equality. For independently and identically distributed random variables U_1 and U_2, we have

$$
\begin{aligned}
2\mathrm{tr}(\mathrm{Var}[h^{(L-1)}(U_1) \mid \Theta^{(L-1)}]) &= \mathbb{E}\left[\left\|h^{(L-1)}(U_1) - h^{(L-1)}(U_2)\right\|^2 \mid \Theta^{(L-1)}\right] \\
&\approx 2(1-\mathbb{E}[\cos(h^{(L-1)}(U_1), h^{(L-1)}(U_2))]).
\end{aligned}
$$

The last approximation is due to $\left\|h^{(L-1)}(U_1)\right\| \approx 1$ under the variance conditions on $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ [1, Lemma 7.1]. That is, $\mathbb{E}[\cos(h^{(L-1)}(U_1), h^{(L-1)}(U_2))]$ and β are close to each other. It is plausible in practice to assume that β is close to one when the depth L is large because the variance of an intermediate output given $\Theta^{(L-1)}$ is likely to be small due to the cone effect.

Proof of Theorem 4. By the law of total variance, for any $k \in [d^{(L)}]$, we have

$$
\frac{\mathrm{Var}[\mathbb{E}[(h^{(L)}(U))_k \mid \Theta^{(L)}]]}{\mathrm{Var}((h^{(L)}(U))_k)} = 1 - \frac{\mathbb{E}[\mathrm{Var}[(h^{(L)}(U))_k \mid \Theta^{(L)}]]}{\mathrm{Var}((h^{(L)}(U))_k)} \tag{4}
$$

[Step 1] For $k \in [d^{(L)}]$, a conditional distribution of $(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k$ given $\Theta^{(L-1)}$ and U is a Gaussian distribution with zero mean and $(1 + h^{(L-1)}(U)^T h^{(L-1)}(U))/d^{(L)}$ variance, we have

$$\mathbb{E}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k]^2 = \mathbb{E}[\sqrt{1 + h^{(L-1)}(U)^T h^{(L-1)}(U)}]^2/(2\pi d^{(L)})$$
$$\mathbb{E}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k^2] = (1 + \mathbb{E}[h^{(L-1)}(U)^T h^{(L-1)}(U)])/d^{(L)},$$

and

$$\begin{aligned}
&\mathrm{Var}((h^{(L)}(U))_k) \\
=&\mathbb{E}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k^2] - \mathbb{E}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k]^2 \\
\geq&\frac{(1 + \mathbb{E}[h^{(L-1)}(U)^T h^{(L-1)}(U)])}{d^{(L)}} \frac{\pi - 1}{2\pi}.
\end{aligned} \tag{5}$$

The last inequality is from Jensen's inequality $\mathbb{E}[\sqrt{1 + U^T U}] \leq \sqrt{1 + \mathbb{E}[U^T U]}$.

[Step 2] For $k \in d^{(L)}$, we now consider $\mathbb{E}[\mathrm{Var}[(h^{(L)}(U))_k \mid \Theta^{(L)}]] = \mathbb{E}[\mathrm{Var}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k \mid \Theta^{(L)}]]$. By the symmetricity of $\mathbf{W}^{(L)}$ and $\mathbf{b}^{(L)}$, we have

$$\begin{aligned}
\mathbb{E}[\mathrm{Var}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k \mid \Theta^{(L)}]] = \frac{1}{2}\mathbb{E}\Big[&\mathrm{Var}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k \mid \Theta^{(L)}] \\
&+ \mathrm{Var}[\phi(-(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)}))_k \mid \Theta^{(L)}]\Big].
\end{aligned}$$

Using the characteristic of the ReLU function, we have $\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k^2 + \phi(-(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)}))_k^2 = (\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k^2$ and

$$\begin{aligned}
&\mathbb{E}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k \mid \Theta^{(L)}]^2 + \mathbb{E}[\phi(-(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)}))_k \mid \Theta^{(L)}]^2 \\
>&\Big(\mathbb{E}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k \mid \Theta^{(L)}] - \mathbb{E}[\phi(-(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)}))_k \mid \Theta^{(L)}]\Big)^2 \\
=&(\mathbf{W}^{(L)}\mathbb{E}[h^{(L-1)}(U) \mid \Theta^{(L-1)}] + \mathbf{b}^{(L)})_k^2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathrm{Var}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k \mid \Theta^{(L)}] + \mathrm{Var}[\phi(-(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)}))_k \mid \Theta^{(L)}] \\
<&\mathbb{E}[(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k^2 \mid \Theta^{(L)}] - (\mathbf{W}^{(L)}\mathbb{E}[h^{(L-1)}(U) \mid \Theta^{(L-1)}] + \mathbf{b}^{(L)})_k^2 \\
=&\mathbf{W}_k^T \mathrm{Var}[h^{(L-1)}(U) \mid \Theta^{(L-1)}]\mathbf{W}_k,
\end{aligned}$$

where $\mathbf{W}_k^T$ is the k-th row of the weight matrix $\mathbf{W}$. Thus, an upper bound for $\mathbb{E}[\mathrm{Var}[(h^{(L)}(U))_k \mid \Theta^{(L)}]]$ is

$$\begin{aligned}
\mathbb{E}[\mathrm{Var}[\phi(\mathbf{W}^{(L)}h^{(L-1)}(U) + \mathbf{b}^{(L)})_k \mid \Theta^{(L)}]] &< \frac{1}{2}\mathbb{E}[\mathbf{W}_k^T \mathrm{Var}[h^{(L-1)}(U) \mid \Theta^{(L-1)}]\mathbf{W}_k] \\
&= \frac{1}{2}\mathrm{tr}(\mathrm{Var}[h^{(L-1)}(U) \mid \Theta^{(L-1)}])/d^{(L)}.
\end{aligned} \tag{6}$$

[Step 3] Finally, combining Equations (5) and (6)

$$\begin{aligned}
\frac{\mathbb{E}[\mathrm{Var}[(h^{(L)}(U))_k \mid \Theta^{(L)}]]}{\mathrm{Var}((h^{(L)}(U))_k)} &< \frac{\mathrm{tr}(\mathrm{Var}[h^{(L-1)}(U) \mid \Theta^{(L-1)}])}{1 + \mathbb{E}[h^{(L-1)}(U)^T h^{(L-1)}(U)]} \frac{\pi}{\pi - 1} \\
&< 1 - \beta.
\end{aligned}$$

The last inequality is due to the fact $\mathbb{E}[h^{(L-1)}(U)^T h^{(L-1)}(U)] = 1$ when $\|U\| = 1$ and $\pi < 2(\pi - 1)$. Due to Equation (4), it concludes a proof. □