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

Anonymous Author(s)
Affiliation
Address
email

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 separate 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.

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 [27], 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 [27], videos [37], medical images [39], and amino-acid sequences [9]. 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 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

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

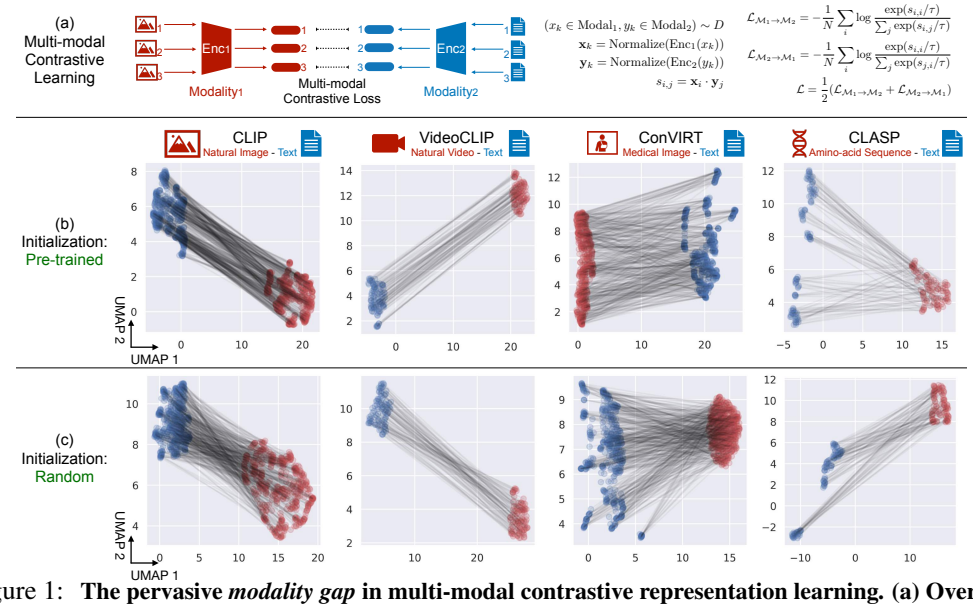


Figure 1: **The pervasive *modality gap* in multi-modal contrastive representation learning. (a) Overview of multi-modal contrastive learning.** Paired inputs from two modalities (e.g., image-caption) are sampled from the dataset and embedded into the hypersphere using two different encoders. The loss 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.

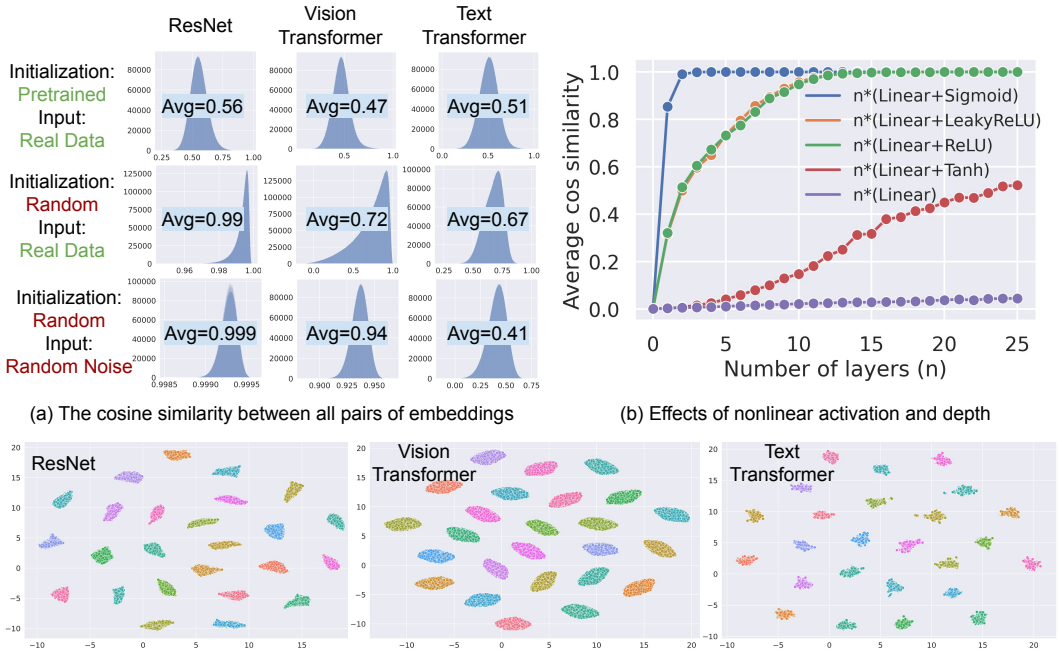
any pair of embedding vectors with high probability, thereby creating more narrow cones in deeper architectures. We further prove that different random initializations of model weights result in different cones. Interestingly, increasing the modality gap in models like CLIP can improve its downstream 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.
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



(a) The cosine similarity between all pairs of embeddings

(b) Effects of nonlinear activation and depth

(c) UMAP visualization of embeddings of 25 randomly initialized models on **real data** (color indicates random seed)

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.

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)[1] on MSCOCO Caption [6]. 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.

In the literature, the cone effect has been observed in the language representations from language models (e.g., BERT) [10]. A common explanation is that the *unbalanced* distribution of word frequencies biased the *optimization* [12, 23]. However, we found that the cone effect still exists in models with random weights (Figure 2 (a)). 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[2], 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.

[1] 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

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

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[3]. In addition, although all 3 models incorporate normalization layers such as batch norm [16] and layer norm [3] 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 [29]. 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, 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 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 $\phi(\mathbf{x}) := (\phi(x_1), \dots, \phi(x_k))^T = (\max(x_1, 0), \dots, \max(x_k, 0))^T$ for a multivariate input $\mathbf{x} = (x_1, \dots, x_k)^T \in \mathbb{R}^k$ 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 $[k] := \{1, \dots, k\}$ 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*

[3] 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.

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 C, we elaborate on the relationship between β and the cosine similarity, and provide a detailed statement of the Theorem.

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 [39, 37, 24]. 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:

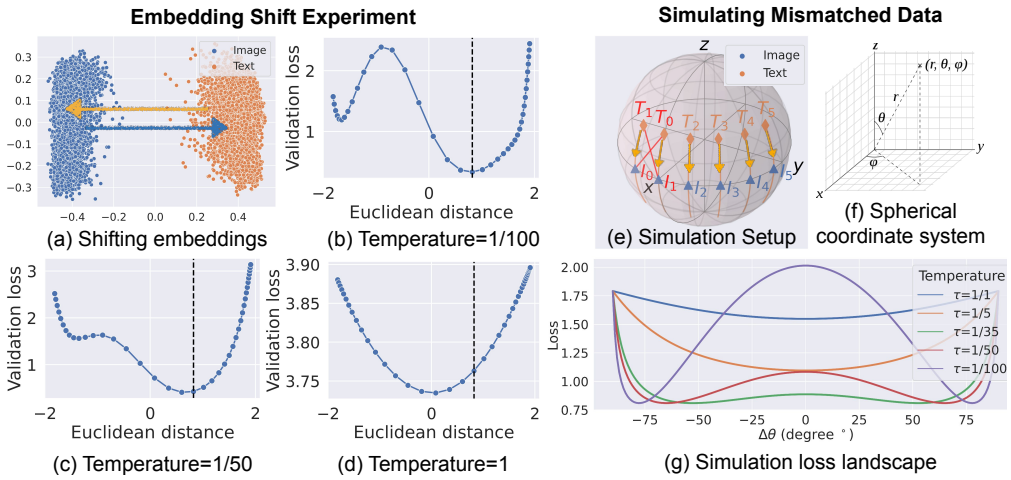


Figure 3: **Contrastive learning preserves modality gap. (a) Embedding shift experiment.** To probe the loss landscape of CLIP, we manually shift the image embeddings and text embeddings towards closing the gap. **(b-d) The loss landscapes under different temperatures.** Y axis indicates the contrastive 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). **(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).

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

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[4] 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 $\|\vec{\Delta}_{\text{gap}}\| = 0.82$.

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

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

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).

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 [21]. 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.

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.

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 [15]), and optical character recognition (SVHN, HatefulMemes [20]). 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 [27, Sec. 7.1]. We performed zero-shot evaluations on CLIP (ViT-B/32) on the evaluation set of the FairFace dataset [19], 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 2). 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)[5]. 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

[5] [27] 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

vision [5, 14, 4, 7], language [28, 13], and graph [38, 26]. However, as contrastive learning is still an emerging representation learning technique, we still lack comprehensive theoretical and empirical understandings about why contrastive learning works. [35] 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. [33] 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 [39, 37, 24, 17, 9]. It has garnered tremendous interest and excitement as a framework for data integration. These models are often pre-trained with contrastive loss [32], as [27] 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 [30, 36]. There have been recent efforts in formulating images and text embeddings as metric learning [11], multilabel classification [18], n-gram language learning [22], and captioning [8]. Research into how the modality gap phenomenon generalizes to the multi-modal representations obtained by these alternative methods is an interesting 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 [10, 12, 23]. Given that isotropy has both theoretical and empirical benefits for static embeddings [25], the extent of anisotropy in contextualized representations is surprising [10]. It has been shown that the cone effect limits the expressiveness of the language representations. Post-processing methods [23, 31, 2, 25] or modified training objective [12, 34, 13] alleviate the cone effect and improve downstream performance. Existing work attributes the cone effect to the *optimization* under unbalanced word frequencies distribution [12, 23]. 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 shwon 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 [33]. Our study provides a complementary understanding of the multi-modal representation space.

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.

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

[4] 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.

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

[6] 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.

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

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

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

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

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

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

[17] 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.

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

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

[20] 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.

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

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

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

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

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

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

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

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

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

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

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

[37] 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.

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

[39] 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]