Abstract

The goal of multimodal alignment is to learn a single latent space that is shared between multimodal inputs. The most powerful models in this space have been trained using massive datasets of paired inputs and large-scale computational resources, making them prohibitively expensive to train in many practical scenarios. We surmise that existing unimodal encoders pre-trained on large amounts of unimodal data should provide an effective bootstrap to create multimodal models from unimodal ones at much lower costs. We therefore propose FuseMix, a multimodal augmentation scheme that operates on the latent spaces of arbitrary pre-trained unimodal encoders. Using FuseMix for multimodal alignment, we achieve competitive performance – and in certain cases outperform state-of-the-art methods – in both image-text and audio-text retrieval, with orders of magnitude less compute and data: for example, we outperform CLIP on the Flickr30K text-to-image retrieval task with \( \sim 600 \times \) fewer GPU days and \( \sim 80 \times \) fewer image-text pairs. Additionally, we show how our method can be applied to convert pre-trained text-to-image generative models into audio-to-image ones. Code is available at: https://github.com/layer6ai-labs/fusemix.

1. Introduction

Recent advances in multimodal machine learning have unlocked unprecedented capabilities across a wide array of understanding-based [47, 48] and generation-based [22, 46, 49, 54] applications, some of which have even garnered mainstream attention [1, 72, 73, 102]. Of particular interest to us for this work is multimodal alignment [28, 30, 36], which we alternatively refer to as multimodal fusion, wherein the goal is to learn a single latent space that is shared between inputs of various modalities. Recent successes in multimodal fusion have been largely driven by large-scale training regimes requiring many GPUs, and often relying on datasets of billions of multimodal pairs [40, 70, 105]. This presents a cost that is unacceptable for many practical scenarios where access to compute is limited and where multimodal data is scarce [56, 91]. It is thus of paramount importance to design efficient frameworks that can democratize research in multimodal fusion.

In this work, our key insight is that off-the-shelf unimodal encoders that have been pre-trained on large amounts of unimodal data already encode rich semantics that should provide an effective bootstrap for multimodal fusion. We introduce FuseMix, a simple and easy-to-implement data augmentation scheme for multimodal fusion inspired by mixup [109], where we share the mixing coefficient across modalities. We show that by aligning the latent spaces of existing pre-trained unimodal encoders using FuseMix, we obtain highly competitive fused multimodal models, which in certain cases even outperform state-of-the-art methods in both image-text and audio-text retrieval tasks, all while using orders of magnitude less compute and data. For example, we
use $\sim 600 \times$ less compute ($\sim 5^1$ vs. $\sim 3000^2$ GPU days) and $\sim 80 \times$ less image-text pairs ($\sim 5M$ vs. $\sim 400M$) than CLIP [70] to perform multimodal fusion, yet are still able to outperform it in recall for the text-to-image retrieval task on the Flickr30K test set [104], see Figure 1. Moreover, in settings with access to limited multimodal pairs, we show that dataset quality and diversity are important properties to increase downstream performance. Finally, we further demonstrate the applicability of our FuseMix fusion framework for audio-to-image generation [28].

2. Related Work

Multimodal Learning. The overarching objective of multimodal learning is to build universal models that can jointly perceive data of various modalities [7, 27, 50, 51, 82, 95, 96, 98, 106, 111]. Said modalities can range from data streams including but not limited to image, text, audio, and video. A standard approach to building multimodal models is to train them end-to-end on data paired across all modalities of interest [3, 15, 47, 48, 59, 79, 81]. However, this approach generally does not scale well since training large-scale multimodal models from scratch can quickly become very compute and data intensive. A more practical approach is to instead bootstrap from pre-trained unimodal networks. Yet, several works in this vein still perform backpropagation through the pre-trained networks [1, 14, 22, 49, 54, 63, 88], which incurs significant overhead due to the large size of the underlying unimodal networks; this problem is bound to be exacerbated as the size of networks increases.

More related to our setting are multimodal models that focus on learning a single shared latent space wherein multiple modalities can be jointly encoded (i.e. multimodal alignment). This line of work was pioneered by CLIP [70] and ALIGN [36], which use a dual-encoder architecture trained with a contrastive objective to jointly embed texts and images. CoCa [105] adds an autoregressive image captioning term to the contrastive objective, which they find improves performance. 3T [40] instead aligns the text and image encoders with the latent space of a pre-trained classifier. LiT [108] uses a frozen pre-trained image classifier as the image encoder, and aligns a text encoder with it. Despite their successes, all of these works train one or both encoders from scratch, requiring expensive gradient computations spanning many GPUs. They also use internet-scale datasets consisting of image-text pairs ranging in quantity from 400M to 5B pairs, and these datasets are often not made publicly available. Moreover, several works have extended CLIP to include other modalities such as video [25, 29, 60] and audio [30], but they require fine-tuning CLIP to achieve good performance. Similarly, other audio-text fusion methods [19, 61, 99] require fine-tuning of the underlying encoders and additional training data. Finally, ImageBind [28] learns a shared latent space across six modalities using a contrastive objective with images as an anchor modality, which they achieve by jointly training several modality encoders from scratch. In contrast to all these works, we prioritize computational and data efficiency by using frozen pre-trained unimodal encoders, by leveraging minimal multimodal paired data, and by ensuring all our experiments require no more than a single GPU of compute.

Data Augmentation. Historically, data augmentations were introduced in an effort to synthetically increase dataset size and diversity [44, 76]: this is exactly our goal, as we operate in a setting with relatively scarce paired multimodal data. In the natural image domain, common augmentations include horizontal flips, random crops, and color jitter [6, 13], which were designed to leave semantic information unchanged. However, designing such augmentations in any given domain requires expert knowledge of which transformations preserve semantic information. For example, naively applying color jitter on the medical image domain can destroy the most relevant information for tasks like cancer classification [75, 80]. Furthermore, handcrafted augmentation schemes typically do not readily transfer to other modalities. This effect is evidenced by the scarcity of modality-agnostic augmentation schemes, despite recent efforts therein such as random projections [80] and randomized quantization [97]. We note that while input masking has been successfully applied in various modalities, expert knowledge is still required to determine an appropriate masking strategy for each modality individually [21, 33, 35, 87]. Given these challenges, it is unsurprising that data augmentations are not as well studied for multimodal learning [31]. In our work, we propose a multimodal augmentation scheme that operates on latent space and is inspired by mixup [109].

3. Problem Setting and Background

3.1. Multimodal Fusion as Alignment

In this work, we define multimodal fusion from the perspective of alignment. Alignment is the task of learning a single latent space that is shared between multimodal inputs. Formally, given any two data modalities $X$ and $Y$ (e.g. images and texts), we aim to learn two networks, $f_X: X \rightarrow S$ and $f_Y: Y \rightarrow S$, that embed each respective modality into a shared latent space $S$.

Recently, contrastive learning has emerged as a prevalent objective for multimodal alignment [36, 47, 70, 108]. It aims to learn a joint latent space wherein semantically
similar multimodal inputs in ambient space are encoded to
nearby points, while semantically dissimilar inputs are en-
bedded further apart. To this end, contrastive learning re-
quires access to semantically similar multimodal inputs in
the form of positive pairs (e.g. images and their correspon-
ding text captions), as well as access to semantically dis-
similar negative pairs (e.g. unrelated images and texts). There-
fore, we must assume there is a way to obtain samples of
positive pairs from the joint distribution over modalities X
and Y given by p_{X,Y}. Negative pairs are often obtained
by sampling from the product of marginal distributions of
each modality, p_X and p_Y. With access to a positive pair
(x, y) \sim p_{X,Y} and negative pairs (x_i^-, y_i^-) \sim p_X p_Y
for i = 1, \ldots, M, contrastive learning in the context of mul-
timodal alignment leverages the InfoNCE loss [66]:
\begin{equation}
L \left( f_X, f_Y; x, y, \{y_i^-\}^{M}_{i=1} \right) \triangleq \frac{\log e^{f_X(x) f_Y(y)/\tau}}{\tau} + \sum_{i=1}^{M} e^{f_X(x) f_Y(y_i^-)/\tau},
\end{equation}
where a \cdot b \triangleq \frac{a \cdot b}{\|a\|_2 \cdot \|b\|_2} denotes cosine similarity and \tau > 0
is either a fixed or learnable scalar temperature parameter.
The final objective is then given by a symmetric version
of the InfoNCE objective:
\begin{equation}
L_{sym} \left( f_X, f_Y \right) \triangleq \mathbb{E} \left[ \frac{1}{2} L \left( f_X, f_Y; x, y, \{y_i^-\}^{M}_{i=1} \right) + \frac{1}{2} L \left( f_Y, f_X; x, y, \{x_i^-\}^{M}_{i=1} \right) \right],
\end{equation}
where the expectation is taken with respect to the positive
pair (x, y) \sim p_{X,Y} and the M negative pairs (x_i^-, y_i^-) \sim p_X p_Y.

We note that formulating alignment through contrastive
learning has been shown to enable zero-shot transfer to vari-
ous multimodal downstream tasks [28, 30, 56, 70], and has
also been shown to improve performance in general multi-
modal settings, including understanding-based [47, 48] and
generation-based [16, 46, 49] tasks. This formulation also
admits theoretical motivations from the perspective of mul-
ti modal information maximization [4, 47, 66, 85].

3.2. Mixup
Mixup [109] is a general-purpose data augmentation rou-
tine for supervised learning. Its premise is simple: given
pairs (x, l) and (\hat{x}, \hat{l}) of data (i.e. x and \hat{x}) and their corre-
sponding labels (i.e. l and \hat{l}), it constructs augmented sam-
ple pairs by taking the convex combinations \hat{x} \triangleq \lambda x + (1 - \lambda) \hat{x}
\footnote{While sampling independently from the marginals could technically result in a semantically related (positive) pair, the probability of this happening in practice is extremely small (e.g. a random text describing an image will rarely properly describe a different random image), thus justifying this procedure to obtain negative pairs.}
and \hat{l} \triangleq \lambda l + (1 - \lambda) \hat{l}, where \lambda \in (0, 1) is an inter-
polation coefficient most commonly sampled from a Beta
distribution B(\alpha, \beta) with hyperparameters \alpha, \beta > 0. The
loss used to train the model is then optimized on the aug-
mented data/label pairs rather than the original ones. Sub-
sequent works have motivated mixup from the perspective
of robustness and generalization [110], as well as calibration
[83]. Variations on mixup have extended the method to
contrastive learning where labels are unavailable [90], but
can be created by proxy [45]. Recently, in the context of mul-
timodal learning, So et al. [77] proposed a mixup strategy
using spherical interpolations to fine-tune CLIP, but this
method requires a shared latent space that is already aligned
and is not readily applicable in our setting.

4. Motivation
Despite recent successes, the prototypical paradigm for
multimodal fusion exhibits critical bottlenecks rooted in
large computational and data overhead, as well as a lack of
modularity. In this section, we discuss these bottlenecks:

\textbf{Computational Burden.} Recent advances in deep
learning have shown that model scale is a key driver of per-
formance and downstream capabilities [8, 17, 37, 55, 107].
Although increasing model scale can greatly benefit perfor-
ance, the required computational cost to train such
models also increases commensurately, and is unattainable
for many machine learning practitioners and researchers.
In the context of multimodal models, these effects are
more prominent as computational requirements are gener-
ally compounded. For example, in our setting of multi-
modal fusion, it is common to jointly train both f_X and f_Y
[28, 36, 40, 70, 105]. This means that backpropagation is
now required through two networks that must both be held
in memory. Moreover, as we increase the scale of each net-
work, the number of parameters requiring expensive gradi-
ent computations quickly accumulates. We therefore aim
to prioritize computational considerations to design an effi-
cient framework for multimodal fusion.

\textbf{Scarcity of High-Quality Paired Data.} Sourcing mul-
timodal paired data is a necessary step in most multimodal
applications. This step amounts to obtaining paired sam-
pies from the joint distribution over modalities as (x, y) \sim
p_{X,Y}. However, in practice, high-quality paired data is of-
ten scarce and expensive to obtain. Typically, this is ei-
ther due to a lack of readily available paired data across all
modalities of interest [28], or due to noisy samples stem-
mimg from large amounts of uninformative and weakly-
labeled data pairs [48, 91]. On the other hand, high-quality
samples of unimodal data from the corresponding marginal
distributions of each modality, x \sim p_X and y \sim p_Y, are
relatively cheap and easy to amass in large quantities. This
is because unimodal data can be collected without any label
pairings while still providing informative intrinsic supervi-
sory signals, as evidenced by successes in self-supervised learning [6, 13, 21, 32, 33]. As such, we aim to defray the cost of sourcing multimodal paired data by leveraging more readily available unimodal signals.

**Tight Coupling From End-to-End Fusion.** While jointly training \( f_X \) and \( f_Y \) from scratch for multimodal fusion may produce a semantically meaningful shared latent space, the resulting networks are tightly coupled. This means that modifying any aspect of either network typically requires completely re-training both networks end-to-end. This presents a challenging bottleneck in practice, as research advancements in each underlying modality cannot be incorporated by end-to-end multimodal fusion without re-training \( f_X \) and \( f_Y \), incurring significant computational, data, and environmental costs [78]. Our goal is therefore to design a plug-and-play framework for multimodal fusion such that individual components can be easily replaced with minimal overhead, allowing multimodal models to keep pace with unimodal improvements.

**5. Method**

In this section we present our framework for multimodal fusion, which aims to address the key considerations of computational and data efficiency, as well as modularity (Sec. 5.1). We also introduce a multimodal augmentation scheme on latent space called FuseMix to facilitate multimodal fusion (Sec. 5.2). Our entire pipeline is depicted in Figure 2.

**5.1. Towards Efficient Multimodal Fusion**

As a first step, we take our two encoders as \( f_X = h_X \circ g_X \) and \( f_Y = h_Y \circ g_Y \). That is, we define \( g_X : \mathcal{X} \rightarrow Z_X \) and \( g_Y : \mathcal{Y} \rightarrow Z_Y \), where \( Z_X \) and \( Z_Y \) are intermediate latent spaces. We then have \( h_X : Z_X \rightarrow S \) and \( h_Y : Z_Y \rightarrow S \), which we hereafter refer to as fusion adapters. Our key insight here is to take both \( g_X \) and \( g_Y \) as pre-trained unimodal encoders which we keep frozen throughout, and treat our fusion adapters \( h_X \) and \( h_Y \) as learnable heads for multimodal fusion. This design offers several advantages:

**Computational Improvements.** We can now equivalently rewrite the alignment loss from Equation 1 as \( \mathcal{L}(h_X, h_Y; g_X(x), g_Y(y), \{g_Y(y_i^-)\}_{i=1}^M) \). This allows us to express the contrastive objective in Equation 2 as an expectation with respect to: positive pairs of encodings \((g_X(x), g_Y(y))\), whose distribution is induced by pushing positive pairs \((x, y) \sim p_{X,Y}\) through the encoders \(g_X\) and \(g_Y\); and negative pairs of encodings \((g_X(x_i^-), g_Y(y_i^-))\), whose distribution is analogously obtained from negative pairs on the ambient spaces. Importantly, since this expectation is taken with respect to a distribution which depends only on the frozen \(g_X\) and \(g_Y\), but not on the trainable \(h_X\) and \(h_Y\), the unimodal encoders \(g_X\) and \(g_Y\) are not used in any gradient computations. In other words, since the unimodal encoders are only needed to provide samples on latent space, not for backpropagation, we can simply pre-compute these samples and then discard the unimodal encoders while training. This step ensures that we do not need to store large encoders in memory during multimodal fusion, which significantly reduces computational requirements. The only parameters stored in memory during fusion are those of the learnable fusion adapters which are extremely lightweight compared to the unimodal encoders.

In fact, in all of our experiments, we only require a single GPU at every step.

**Paired Data Efficiency.** By setting \( Z_X \) and \( Z_Y \) as the latent spaces of pre-trained unimodal encoders, we can directly benefit from the rich modality-specific semantics that they already encode. Learning this information from
scratch might be redundant for multimodal fusion, so leveraging pre-trained unimodal encoders can be an effective bootstrap to reduce the need for large-scale multimodal paired data. We can interpret this effect as a form of distillation from a unimodal latent space into a joint space for which contrastive objectives have been shown to be effective [40, 84]. In other words, leveraging pre-trained unimodal encoders for multimodal fusion should require less paired data than training end-to-end from scratch.

**Plug-and-Play Framework.** We highlight that our modular approach to multimodal fusion is agnostic to both the choice of unimodal encoders \(g_X\) and \(g_Y\) and to the underlying modalities \(\mathcal{X}\) and \(\mathcal{Y}\). Importantly, by combining arbitrary pre-trained unimodal encoders, we can decouple unimodal learning from multimodal fusion. Therefore, as the development of unimodal encoders continues to advance, we can easily and efficiently leverage new unimodal encoders for multimodal fusion in a plug-and-play manner.

### 5.2. FuseMix: Multimodal Latent Mixup

Given our aim of performing multimodal fusion with minimal samples of paired data, it would seem intuitive to also leverage data augmentations to generate synthetic multimodal pairs \((\hat{x}, \hat{y}) \in \mathcal{X} \times \mathcal{Y}\). However, constructing semantically meaningful data augmentations directly on the ambient spaces \(\mathcal{X}\) and \(\mathcal{Y}\) is generally challenging due to the heterogeneity of multimodal data [31]. On the other hand, we note that \(Z_X\) and \(Z_Y\) provide a more homogeneous alternative since they are both intermediate latent spaces of pre-trained unimodal encoders. Additionally, they already encode semantic information that can be beneficial for creating meaningful data augmentations.

As such, we introduce a simple yet effective multimodal augmentation scheme on latent space that is agnostic to both the involved modalities and the choice of unimodal encoders. Our approach, which we call FuseMix, is inspired by mixup [109], in that augmented samples are generated from random convex combinations. In particular, we take linear interpolations between samples in both \(Z_X\) and \(Z_Y\). Importantly, since both latent spaces are taken from pre-trained unimodal encoders, we should expect linear interpolations to be more semantically meaningful than when carried out on ambient space, as is typically done in mixup [45, 90, 109]. We note that this idea of semantic interpolations in latent space is reminiscent of latent space arithmetic that has a well-established history [24, 28, 62, 69].

However, na"{i}vely mixing random samples in each latent space would only produce augmented pairs of latents \((\tilde{z}_x, \tilde{z}_y) \in Z_X \times Z_Y\) where \(\tilde{z}_x\) and \(\tilde{z}_y\) are unrelated to one another. Therefore, we want to impose some structure on how interpolations are performed across modalities to ensure that we can construct semantically meaningful augmented pairs. To achieve this we take any two existing positive multimodal pairs \((z_x, z_y) \overset{\Delta}{=} (g_X(x), g_Y(y))\) and \((\tilde{z}_x, \tilde{z}_y) \overset{\Delta}{=} (g_X(\tilde{x}), g_Y(\tilde{y}))\), where \((x, y), (\tilde{x}, \tilde{y}) \overset{i.i.d.}{\sim} p_{X,Y}\), and construct a corresponding augmentation \((\hat{z}_x, \hat{z}_y)\) as

\[
(\hat{z}_x, \hat{z}_y) \overset{\Delta}{=} \lambda (z_x, z_y) + (1 - \lambda) (\tilde{z}_x, \tilde{z}_y),
\]

where \(\lambda \in (0,1)\) is the shared interpolation coefficient. Sharing \(\lambda\) across modalities ensures that the resulting augmentation is semantically consistent, meaning \(\hat{z}_x\) and \(\hat{z}_y\) still form a valid positive pair. In practice, we can of course similarly apply FuseMix to obtain interpolations of negative pairs in such a way that the result remains a negative pair.

Finally, our version of Equation 2 on the intermediate latent spaces with FuseMix is given by

\[
L_{\text{FuseMix}}^{\text{sym}}(h_X, h_Y) \overset{\Delta}{=} \mathbb{E} \left[ \frac{1}{2} \mathcal{L}(h_X; h_Y; \tilde{x}_x, \tilde{z}_x, \{\tilde{z}_y^\tau\}_{\tau=1}^M) \right. \\
+ \frac{1}{2} \mathcal{L}(h_Y; h_X; \tilde{z}_y, \tilde{x}_y, \{\tilde{x}_x^\tau\}_{\tau=1}^M),
\]

where the expectation is taken with respect to: the positive pairs \((z_x, z_y)\) and \((\tilde{z}_x, \tilde{z}_y)\) used to obtain the augmented positive pair \((\hat{z}_x, \hat{z}_y)\); the negative pairs \(\{(\tilde{z}_x^\tau, \tilde{z}_y^\tau)\}_{\tau=1}^M\); and \(\{(\tilde{z}_x^\tau, \tilde{z}_y^-)\}_{\tau=1}^M\) used to obtain the augmented negative pairs \(\{(\hat{z}_x^\tau, \hat{z}_y^-)\}_{\tau=1}^M\); and \(\lambda \sim \mathcal{B}(\alpha, \beta)\). We note that our FuseMix fusion algorithm can be implemented very easily. Given pre-computed samples of multimodal latent pairs (see Sec. 5.1), setting the batch size \(B \overset{\Delta}{=} M + 1\), and taking \(\alpha = \beta\), the simplicity of our method is illustrated in Algorithm 1, requiring only a few lines of code.
6. Experiments

In our experiments, we consider the image-text and audio-text modality pairings. We start by describing details of our implementation and then we perform experimental analysis to evaluate our framework and provide insights on key components of multimodal fusion.

6.1. Implementation Details

Unimodal Latent Extraction. Since an important consideration of our method is to minimize computational requirements, we only use a single 32GB NVIDIA V100 GPU for all of our experiments. This is possible for us because, as mentioned in Sec. 5.1, we can pre-compute the latents from pre-trained unimodal encoders so that the underlying encoders can be discarded thereafter. Additionally, we can extract the latents for each modality one at a time to ensure that no more than one encoder must be loaded at once. Importantly, these steps allow us to consider large-scale encoders on the order of billions of parameters which would generally not be feasible for end-to-end fusion on a single GPU. We mainly consider Transformer-based [89] unimodal encoders, and extract low-dimensional latents from the penultimate layer of either the [CLS] token if it exists, or the mean-pooled token otherwise.

Multimodal Latent Fusion. We parameterize our fusion adapters as lightweight MLPs using an inverted bottleneck architecture following previous work [5, 53, 86]. Each MLP consists of residual blocks followed by a final projection layer of dimension 512 by default to embed each modality into a shared space. We highlight that since our fusion adapters are operating on low-dimensional latents, the computational cost to train them is minimal, and despite training on a single GPU, we can use large batch sizes (up to $B = 20K$ on our V100 GPU), which has been shown to benefit contrastive learning [13, 32, 85, 94, 100]. Finally, we note that in all of our experiments, unless otherwise stated, we use $L_{\text{sym}}$ as our sole objective for multimodal fusion. More details on the MLP architecture and hyperparameters can be found in Appendix A and B.

Training Datasets. We rely on common multimodal datasets for training. Specifically, following previous works [15, 47–49], we leverage the image-text pairs from human-annotated datasets (COCO [52] and Visual Genome [41]), and web datasets (SBU Captions [68] and Conceptual Captions 3M [74]), amounting to 5M total pairs. In order to remain data-efficient, we note that we intentionally avoid internet-scale datasets like the ones used in several recent works [36, 40, 70, 105], as these are orders of magnitude larger than our collated dataset. Similarly, to remain data-efficient for the audio-text regime, we only leverage the AudioCaps [38] and Clotho [23] train sets which provide 50K and 15K human-annotated audio-text pairs, respectively.

6.2. Cross-Modal Retrieval Performance

To assess the quality of multimodal alignment learned from FuseMix fusion, we follow previous works [19, 36, 39, 40, 70, 99] and evaluate our method using the downstream task of cross-modal retrieval. In particular, for the image-text pairing, we evaluate downstream performance on the Flickr30K [104] and COCO [52] test sets, and for the audio-text pairing, we evaluate our method on the AudioCaps [38] and Clotho [23] test sets. In our experiments, we use subscriptions to specify which pre-trained unimodal encoders were used for bootstrapping. In terms of image encoders, we consider both DINOv2 [67] (D) and UNICOM [2] (U) since, as of the time of writing, they are two of the top-ranked visual recognition models as measured by the ImageNet [18] linear probing benchmark. On the text side, we use the MTEB [64] text embedding benchmark to select two encoders with demonstrably semantic latent spaces, namely BGE [101] (B) and E5 [92] (E). Finally, on the audio side we utilize the commonly used HTS-AT [9] (H) and the recent Whisper [71] (W) encoders. In practice, we actually use the concatenation of the latents from these two encoders (W&H), similar to [19]. We emphasize that given the plug-and-play nature of our method, as better unimodal encoders become available, we can quickly and cheaply incorporate them into our framework. We report results across all combinations of these encoders in Table 1 and Table 2.

For image-text retrieval, we highlight that our method is highly competitive and sometimes able to outperform various state-of-the-art methods which are trained on orders of magnitude more paired data and that require substantially more than a single GPU of compute for fusion. Moreover, we find that the combination of two of the most recent models, DINOv2+BGE, achieves the highest performance, highlighting the benefits of a plug-and-play approach that can leverage the most recent advancements. We also note that when our method and CLIP [91] are both only trained on pairs from Conceptual Captions 3M, we outperform CLIP by a notable margin, demonstrating that FuseMix is an effective strategy for fusion on lower data regimes. Similarly, for audio-text retrieval we outperform all other methods trained on similar data, and can compete with methods that use orders of magnitude more paired data.

6.3. Evaluating Dataset Efficiency

As mentioned in Sec. 4, sourcing multimodal data pairs across all modalities of interest can be costly, especially in scarce data regimes. In practical settings, it is therefore natural to wonder how one should allocate efforts to construct a dataset for multimodal fusion that would maximize performance. We aim to answer this question by characterizing and quantifying three key properties of datasets, namely quantity, quality, and diversity. For dataset quantity, we take an existing dataset and uniformly subsample vari-
Finally, for dataset quality, we consider human-annotated datasets to be of higher-quality, and web datasets to be of lower-quality. For dataset quality, we consider numbers of pairs to measure the effect of quantity on downstream performance. For dataset quality, we consider human-annotated datasets to be of higher-quality, and web datasets to be of lower-quality. Finally, for dataset diversity, we can rely on determinantal point processes (DPPs) \cite{10, 42, 43}. For a given dataset, DPPs return subsets of points of a pre-specified size that are maximally diverse (see Appendix D for details). Applied to our setting, we use DPPs on an existing dataset to obtain diverse subsets of various sizes and then compare the performance against uniformly sampled subsets of the corresponding sizes.

Our results are shown in Figure 3. We observe that increased quantity of data improves performance in lower data regimes as expected (Figure 3a). However, the quality of the underlying dataset also has a very strong effect, as has been similarly observed in other work \cite{48, 91}. In fact, in Figure 3b, we find that 6× the number of image-text pairs from the web are required to match the performance of using higher quality human-annotated pairs. Interestingly, in Figure 3c we find that with access to limited data, sourcing image-text pairs that are maximally diverse provides substantial improvements of up to nearly 40% compared to se-

Table 1. Results of image-text retrieval on the Flickr30K 1K and COCO 5K test sets. The top section of the table contains fusion methods trained with internet-scale data, while the bottom section contains methods using much fewer image-text pairs. All our results use the largest available version of the underlying unimodal encoders. Refer to Sec. 6.2 for the definition of the subscripts.

<table>
<thead>
<tr>
<th>Method</th>
<th># (image, text)</th>
<th>Flickr30K (1K test set)</th>
<th>COCO (5K test set)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>text → image</td>
<td>R@1</td>
<td>R@5</td>
</tr>
<tr>
<td>internet-scale</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILIP \cite{103}</td>
<td>300M</td>
<td>75.0</td>
<td>93.4</td>
</tr>
<tr>
<td>CLIP \cite{70}</td>
<td>400M</td>
<td>68.7</td>
<td>90.6</td>
</tr>
<tr>
<td>ALIGN \cite{36}</td>
<td>1B</td>
<td>75.7</td>
<td>93.8</td>
</tr>
<tr>
<td>LIT \cite{108}</td>
<td>4B</td>
<td>66.5</td>
<td>-</td>
</tr>
<tr>
<td>3T \cite{40}</td>
<td>5B</td>
<td>72.1</td>
<td>-</td>
</tr>
<tr>
<td>low-data regime</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLIP \cite{91}</td>
<td>3M</td>
<td>54.3</td>
<td>84.1</td>
</tr>
<tr>
<td>FuseMix(D,B)</td>
<td>3M</td>
<td>59.9</td>
<td>86.4</td>
</tr>
<tr>
<td>FuseMix(W,B)</td>
<td>5M</td>
<td>66.3</td>
<td>88.9</td>
</tr>
<tr>
<td>FuseMix(U,B)</td>
<td>5M</td>
<td>60.3</td>
<td>87.7</td>
</tr>
<tr>
<td>FuseMix ESA</td>
<td>5M</td>
<td>68.8</td>
<td>90.9</td>
</tr>
<tr>
<td>FuseMix W &amp; ESA</td>
<td>5M</td>
<td>71.2</td>
<td>91.4</td>
</tr>
</tbody>
</table>

Table 2. Results of audio-text retrieval on the AudioCaps 1K and Clotho 1K test sets. The top section of the table contains fusion methods trained with internet-scale data, while the bottom section contains methods using much fewer audio-text pairs. All our results use the largest available version of the underlying unimodal encoders. Refer to Sec. 6.2 for the definition of the subscripts.

<table>
<thead>
<tr>
<th>Method</th>
<th># (audio, text)</th>
<th>AudioCaps (1K test set)</th>
<th>Clotho (1K test set)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>text → audio</td>
<td>R@1</td>
<td>R@5</td>
</tr>
<tr>
<td>internet-scale</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAION-CLAP \cite{99}</td>
<td>700K</td>
<td>36.2</td>
<td>70.3</td>
</tr>
<tr>
<td>VALOR \cite{11}</td>
<td>6.5M</td>
<td>40.1</td>
<td>73.9</td>
</tr>
<tr>
<td>ONE-PEACE \cite{83}</td>
<td>2.4M</td>
<td>42.5</td>
<td>77.5</td>
</tr>
<tr>
<td>VAST \cite{12}</td>
<td>27M</td>
<td>52.0</td>
<td>76.8</td>
</tr>
<tr>
<td>low-data regime</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ML-ACT \cite{61}</td>
<td>50K/15K</td>
<td>33.9</td>
<td>69.7</td>
</tr>
<tr>
<td>MMT \cite{39}</td>
<td>50K/15K</td>
<td>36.1</td>
<td>72.0</td>
</tr>
<tr>
<td>CLAP-HTSAT \cite{19}</td>
<td>70K</td>
<td>34.7</td>
<td>70.2</td>
</tr>
<tr>
<td>LAION-CLAP \cite{99}</td>
<td>65K</td>
<td>36.7</td>
<td>70.9</td>
</tr>
<tr>
<td>FuseMix ([W,H,B])</td>
<td>50K/15K</td>
<td>41.3</td>
<td>76.9</td>
</tr>
<tr>
<td>FuseMix ([W,H,B])</td>
<td>65K</td>
<td>43.1</td>
<td>77.4</td>
</tr>
</tbody>
</table>
6.4. Audio-to-Image Generation

We consider the recently proposed task [28] of generating images given audio prompts. The aim is to repurpose an existing text-to-image generative model to be conditioned on audio in lieu of text. Girdhar et al. [28] achieved this using a private reimplementation of DALLE-2 [72]. We opt to use FuseMix to perform this task while only using publicly available models: we use GLIDE\(^5\) [65], a text-conditioned diffusion model which leverages CLIP\(^6\) [70] to condition on text. We apply our method to align the latent space of Whisper into the latent space of CLIP to endow GLIDE with audio-conditioning capabilities (see details in Appendix B). In Figure 4, we provide examples of generated samples using various sounds. While we omit quantitative analysis for this task due to a lack of suitable metrics, we provide a qualitative comparison of each sample with a corresponding sample generated from the original text-conditioned GLIDE using a text prompt that is semantically equivalent to the audio prompt. For example, for the sound of a cat meowing, we compare with the text prompt “a photo of a cat meowing”. We find it noteworthy that conditioning GLIDE on audio in lieu of text, Girdhar et al. [28] achieved this using modal fusion with encoders only accessible via an API. Whispers into the latent space of CLIP to endow GLIDE with audio-conditioning capabilities (see details in Appendix B).

6.5. Ablations

Effect of Unimodal Encoder Size. Given the plug-and-play nature of our method, we would hope that larger underlying unimodal encoders would be beneficial for multimodal fusion. We study this effect by evaluating downstream performance for various sizes of encoders. We consider the following combinations: DINOv2 ViT-S/14 & BGE Small; DINOv2 ViT-B/14 & BGE Base; and DINOv2 ViT-G/14 & BGE Large, referred to as S, B, and L, respectively, in Figure 5a. As shown, scaling the unimodal encoders translates to improved downstream performance.

Effect of Batch Size. As mentioned in Sec. 6.1, since training our fusion adapters requires minimal compute, we can use larger batch sizes even on a single GPU. In Figure 5b, we see that our method can benefit from more negative samples in the contrastive objective, which is consistent with findings in previous work [13, 32, 85].

Effect of Data Augmentations. In Figure 5c, we evaluate the importance of data augmentations and compare our proposed FuseMix with other modality-agnostic data augmentation schemes, namely Gaussian noise and random quantization [97]. We note that data augmentations generally seem beneficial in our setting, although FuseMix provides the largest improvement in performance, further validating our proposed approach.

7. Conclusion and Future Work

In this work, we have proposed a framework for multimodal fusion that is both compute-efficient and data-efficient which can effectively bootstrap from arbitrary pre-trained unimodal encoders. We have introduced FuseMix, a simple yet effective multimodal augmentation scheme on latent space inspired by mixup. However, while our method can benefit from powerful unimodal encoders, we are limited by the semantic information that they have previously learned [40]. It would thus be an interesting future direction to apply efficient fine-tuning methods [20, 34] to the unimodal encoders during fusion, although this would incur additional overhead. We also highlight that since our framework essentially considers unimodal encoders as black box models (i.e. we only use their latent encodings from their penultimate layer), this opens up exciting applications whereby we can perform multimodal fusion with encoders only accessible via an API.
References


[58] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 14


[60] Huaiashuo Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. 2


Appendices

A. Architecture

For our fusion adapters $h_X$ and $h_Y$, we use a simple inverted bottleneck MLP architecture. To illustrate the simplicity of our design, we provide its pseudocode in Algorithm 2. By default, we use an expansion factor of 4, dropout of 0.6, and a shared latent space of dimension 512. We specify the fusion adapter depths we used for each task in Appendix B.

Algorithm 2: PyTorch-style pseudocode of our fusion adapters.

```python
# D_x, D_y: latent dimension of unimodal encoders
# D_s: latent dimension of shared space
# depth_x, depth_y: number of blocks for each adapter
# expansion_factor: expansion factor hyperparameter
# dropout: dropout hyperparameter
from torch import nn
class Block(nn.Module):
    def __init__(self, dim, expansion_factor=4, dropout=0.6):
        super().__init__()
        self.fn = nn.Sequential(
            nn.Linear(dim, int(expansion_factor * dim)),
            nn.GELU(),
            nn.Dropout(dropout),
            nn.Linear(int(expansion_factor * dim), dim),
        )
        self.ln = nn.LayerNorm(dim)
    def forward(self, x):
        return x + self.fn(self.ln(x))

h_X = nn.Sequential(
    *(Block(D_x, expansion_factor, dropout) for _ in range(depth_x)),
    nn.LayerNorm(D_x),
    nn.Linear(D_x, D_s),
)

h_Y = nn.Sequential(
    *(Block(D_y, expansion_factor, dropout) for _ in range(depth_y)),
    nn.LayerNorm(D_y),
    nn.Linear(D_y, D_s),
)
```

B. Implementation Details

For all experiments, we use the AdamW [58] optimizer during training. We perform learning rate warmup by linearly increasing the learning rate from $10^{-6}$ to $l_r$ (which we specify for each task below) during the first epoch. We then decay the learning rate using a cosine schedule [57]. We also set our FuseMix Beta distribution hyperparameter as $\alpha = 1$ so that the interpolation coefficient is sampled as $\lambda \sim B(1, 1)$.\(^7\) We note that when mixup is performed on ambient space, it is common to select small $\alpha$ [31, 90, 109]. This ensures that inputs are only slightly perturbed so that they remain semantically meaningful. Conversely, in FuseMix, we are operating on the latent space of pre-trained unimodal encoders where we find that relatively larger $\alpha$ can improve performance in our experiments, which suggests that larger perturbations on latent space can remain semantically meaningful (see result in Appendix C). We next describe specific details and hyperparameters for each task we consider:

**Image-Text Retrieval.** We use a depth of 4 for both fusion adapters (see ablation in Appendix C) which we train for 500 epochs with a batch size of $B = 20K$. We set the learning rate as $l_r = 10^{-3}$ and use weight decay of 0.1 during optimization.

**Audio-Text Retrieval.** We use a depth of 2 for both fusion adapters, which we train for 50 epochs with a batch size of $B = 2K$. We set the learning rate as $l_r = 10^{-4}$ and use weight decay of 0.5 during optimization.

**Audio-to-Image Generation.** Since we align the latent space of Whisper’s encoder into the latent space of CLIP, we are treating CLIP’s latent space as our shared space. This means that we only require one fusion adapter to map from Whisper space into CLIP space – for which we use a depth of 2. We note that this does not require any changes to our framework since it is equivalent to setting one of our fusion adapters as the identity network in Algorithm 1. For this experiment, we use 50K audio-text pairs from the AudioCaps [38] training set and a 50K subset of AudioSet [26]. Other hyperparameters are identical to those for audio-text retrieval. During inference, we can therefore map audio inputs to CLIP space and treat them as though they were CLIP text latents, which GLIDE can then use for conditioning.

C. Additional Ablations

![Figure 6. Text-to-image results evaluated on the Flickr30k test set.](image)

We provide results for a few additional ablations. First, we observe in Figure 6a that our method can generally benefit from larger $\alpha$ (see Appendix B for a relevant discussion). We also find in Figure 6b that as the fusion adapters deepen, performance gradually increases until a depth of 4 where performance peaks. These results validate our setting of these hyperparameters detailed in Appendix B.
D. Determinantal Point Processes

We begin with a brief summary of determinantal point processes (DPPs) for completeness, and refer readers to [43] for a thorough overview of DPPs in machine learning. Consider the set \( \mathcal{I} \triangleq \{1, 2, \ldots, N\} \), which should be understood as the set of indices of a dataset \( \{z_i\}_{i \in \mathcal{I}} \subset \mathcal{Z} \) with \( N \) distinct elements. Consider also a symmetric positive semi-definite \( N \times N \) matrix \( L \), such that \( L_{ij} \) measures similarity between \( z_i \) and \( z_j \). A common choice for this matrix is to specify a positive semi-definite kernel \( K : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \) and set \( L_{ij} = K(z_i, z_j) \). A DPP is a distribution over subsets of \( \mathcal{I} \), where the probability of obtaining \( S \subset \mathcal{I} \) is given by

\[
p_S(S) = \frac{\det L_S}{\sum_{S' \subset \mathcal{I}} \det L_{S'}},
\]

where \( L_S \) corresponds to the \( |S| \times |S| \) submatrix of \( L \) whose row and column indices are given by \( S \). The idea behind DPPs is that diverse subsets are more likely to be sampled, where diversity is measured through dissimilarity (as specified in \( L \)) of the elements in \( \{z_i\}_{i \in S} \). DPPs can be extended to \( k \)-DPPs [42], where an integer \( k \) is specified and the constraint is added that \( S \) must have exactly \( k \) elements, or more formally

\[
p_S(S \mid |S| = k) = \frac{\det L_S}{\sum_{S' \subset \mathcal{I}, |S'| = k} \det L_{S'}} \mathbb{1}(\{|S| = k\}),
\]

where \( \mathbb{1}(\cdot) \) denotes an indicator function. In the DPP literature, it can be of interest to find a mode of a DPP or \( k \)-DPP (i.e. finding “maximally diverse” subsets, potentially of specified size \( k \)) rather than to sample from these distributions. In our case, we follow the greedy algorithm proposed in [10], whose goal is to obtain a mode \( S^* \) of a \( k \)-DPP:

\[
S^* \in \arg \max_{S \subset \mathcal{I}, |S| = k} \det L_S.
\]

To specify \( L \), we first considered the kernel \( K(z, z') = z \cdot z' \) in an attempt to leverage the prior knowledge that cosine similarity is sensible on the latent space \( \mathcal{Z} \) of pre-trained encoders.\(^8\) However, the resulting matrix \( L \) has low rank – at most the dimension of \( \mathcal{Z} \) – and a requirement for the \( \arg \max \) in Equation 7 to not be the empty set is that \( k \leq \text{rank}(L) \). To be able to use larger \( k \), we thus changed the kernel to \( K(z, z') = (z \cdot z' + 1)^2 \), which is monotonically increasing in \( z \cdot z' \), but results in an \( L \) with much larger rank. We emphasize that in our work we are using \( k \)-DPPs only to evaluate the effect of dataset diversity for various values of \( k \) (i.e. various subset sizes) rather than suggesting its use to curate diverse datasets in practice, which would be too costly.

\(^8\)Recall that \( K \) is a positive semi-definite kernel if, for every \( N \) and every finite subset \( \{z_i\}_{i \in \mathcal{I}} \) of \( \mathcal{Z} \) of size \( N \), the corresponding \( N \times N \) matrix \( L \) is always positive semi-definite.

\(^9\)In our experiments, we subsampled 75K (i.e. \( N = 75K \)) image-text pairs from the COCO dataset to ensure \( L \) was able to fit in memory, and took \( \mathcal{Z} \) as the latent space of the BGE text encoder.