LEARNING MULTIMODAL DATA AUGMENTATION IN FEATURE SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

The ability to jointly learn from multiple modalities, such as text, audio, and visual data, is a defining feature of intelligent systems. While there have been promising advances in designing neural networks to harness multimodal data, the enormous success of data augmentation currently remains limited to single-modality tasks like image classification. Indeed, it is particularly difficult to augment each modality while preserving the overall semantic structure of the data; for example, a caption may no longer be a good description of an image after standard augmentations have been applied, such as translation. Moreover, it is challenging to specify reasonable transformations that are not tailored to a particular modality. In this paper, we introduce LeMDA, *Learning Multimodal Data Augmentation*, an easy-to-use method that automatically learns to jointly augment multimodal data in feature space, with no constraints on the identities of the modalities or the relationship between modalities. We show that LeMDA can (1) profoundly improve the performance of multimodal deep learning architectures, (2) apply to combinations of modalities that have not been previously considered, and (3) achieve state-of-the-art results on a wide range of applications comprised of image, text, and tabular data.

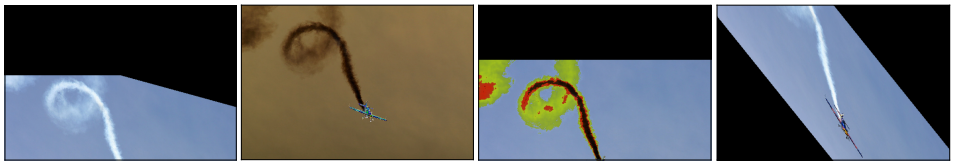
1 INTRODUCTION

Imagine watching a film with no sound, or subtitles. Our ability to learn is greatly enhanced through jointly processing multiple data modalities, such as visual stimuli, language, and audio. These information sources are often so entangled that it would be near impossible to learn from only one modality in isolation — a significant constraint on traditional machine learning approaches. Accordingly, there have been substantial research efforts in recent years on developing multimodal deep learning to jointly process and interpret information from different modalities at once (Baltrušaitis et al., 2017). Researchers studied multimodal deep learning from various perspectives such as model architectures (Kim et al., 2021b; Pérez-Rúa et al., 2019; Nagrani et al., 2021; Choi & Lee, 2019), training techniques (Li et al., 2021; Chen et al., 2019a), and theoretical analysis (Huang et al., 2021; Sun et al., 2020b). However, data augmentation for multimodal learning remains relatively unexplored (Kim et al., 2021a), despite its enormous practical impact in single modality settings.

Indeed, data augmentation has particularly proven its value for data efficiency, regularization, and improved performance in computer vision (Ho et al., 2019; Cubuk et al., 2020; Müller & Hutter, 2021; Zhang et al., 2017; Yun et al., 2019) and natural language processing (Wei & Zou, 2019; Karimi et al., 2021; Fadaee et al., 2017; Sennrich et al., 2015; Wang & Yang, 2015; Andreas, 2020; Kobayashi, 2018). These augmentation methods are largely tailored to a particular modality in isolation. For example, for object classification in vision, we know certain transformations such as translations or rotations should leave the class label unchanged. Similarly, in language, certain sentence manipulations like synonym replacement will leave the meaning unchanged.

The most immediate way of leveraging data augmentation in multimodal deep learning is to separately apply well-developed unimodal augmentation strategies to each corresponding modality. However, this approach can be problematic because transforming one modality in isolation may lead to disharmony with the others. Consider Figure 1, which provides four training examples from SNLI-VE (Xie et al., 2019a), a vision-language benchmark dataset. Each description is paired with the image on the top left, and the label refers to the relationship between the image and description.

	Label	Description
(a)	Entailment	The plane is doing tricks while flying down.
(b)	Entailment	There is a plane in the air.
(c)	Entailment	The pilot is aware that the plane is doing a loop.
(d)	Neutral	The plane is falling down.



AutoAugment-Cifar10 AutoAugment-ImageNet AutoAugment-SVHN TrivialAugment

Figure 1: The top row shows four training samples drawn from SNLI-VE (Xie et al., 2019a), a visual entailment dataset. Each text description is paired with the image on the top left. The task is to predict the relationship between the image and the text description, which can be "Entailment", "Neutral", or "Contradiction". The bottom row shows four augmented images generated by different image-only augmentation methods. If we pair the text description with the augmented images, we observe mislabeled data. For example, the smoke loop is cropped out in the image augmented via TrivialAugment. The new image does not match the description: "The pilot is aware that the plane is doing a loop", as in data (c). However, the label of the augmented pair will still be "Entailment".

The bottom row provides four augmented images generated by state-of-the-art image augmentation methods (Cubuk et al., 2019; Müller & Hutter, 2021). In the image generated by AutoAugment-Cifar10 and AutoAugment-SVHN, the plane is entirely cropped out, which leads to mislabeling for data (a), (b), (c), and (d). In the image generated by AutoAugment-ImageNet, due to the change in smoke color, this plane could be on fire and falling down, which leads to mislabeling for data (a) and (d). In the image generated by TrivialAugment (Müller & Hutter, 2021), a recent image augmentation method that randomly chooses one transformation with a random magnitude, the loop is cropped out, which leads to mislabeling for data (a) and (c). Mislabeling can be especially problematic for over-parameterized neural networks, which tend to confidently fit mislabeled data, leading to poor performance (Pleiss et al., 2020).

There are two key challenges in designing a general approach to multimodal data augmentation. First, multimodal deep learning takes input from a diverse set of modalities. Augmentation transformations can be obvious for some modalities such as vision and language, but not others, such as sensory data which are often numeric or categorical. Second, multimodal deep learning includes a diverse set of tasks with different cross-modal relationships. Some datasets have redundant or totally correlated modalities while others have complementary modalities. There is no reasonable assumption that would generally preserve labels when augmenting modalities in isolation.

In this work, we propose LeMDA (*Learning Multimodal Data Augmentation*) as a general multimodal data augmentation method. LeMDA augments the latent representation and thus can be applied to any modalities. We design the augmentation transformation as a learnable module such that it is adaptive to various multimodal tasks and cross-modal relationships. Our augmentation module is learned together with multimodal networks to produce informative data through adversarial training, while preserving semantic structure through consistency regularization. With no constraints over the modalities and tasks, one can simply plug-and-play LeMDA with different multimodal architectures. We summarize our contributions as follows.

- In Section 3, we introduce LeMDA, a novel approach to multimodal data augmentation. Section 3.1 shows how to use LeMDA with multimodal networks, and Section 3.2 describes how to train the augmentation module to produce informative and label preserving data. The method is notable for several reasons: (1) it can be applied to any modality combinations; (2) it is attractively simple and easy-to-use; (3) it is the first augmentation method to be applied to the joint of text, image, and tabular data, which is essentially uncharted territory.

- In Section 4, we show that LeMDA consistently boosts accuracy for multimodal deep learning architectures compared to a variety of baselines, including state-of-the-art input augmentation and feature augmentation methods.
- In Section 4.4, we provide an ablation study validating the design choices behind LeMDA. In particular, we study the architecture of the augmentation module, and the effects of consistency regularizer. We demonstrate that the consistency regularizer clearly outperforms L_2 regularization (Tang et al., 2020b).

2 Background and Related Work

Multimodal network architectures. Multimodal deep learning architectures are categorized as performing early or late fusion, depending on the stage of combining information from each modality. In *early fusion*, the network combines the raw input or token embedding from all the modalities. Early fusion architectures can be designed to exploit the interaction between low-level features, making it a good choice for multimodal tasks with strong cross-modal correlations (Barnum et al., 2020; Gao et al., 2020). For example, there exist low-level correspondence in image captioning task because different words in the caption may relate to different objects in the image. We note that feature-space augmentation procedures are typically computationally intractable on early-fusion architectures, because early fusion would require combining a large number of latent features, such as a long sequence of token embeddings. On the other hand, in *late fusion*, the focus of our work, input from each modality is independently processed by different backbones. The representations provided by different backbones are fused together in later layers, often just before the classifier layer (Shi et al., 2021a; Wang et al., 2017; Schönfeld et al., 2019; Mahajan et al., 2020). This design is straightforward to apply to any new modality and any multimodal task. Late fusion often uses pre-trained networks as backbones in each modality, making it more computationally tractable. In both early and late fusion, there are a variety of methods to fuse information. Standard approaches include (1) feed all modalities as token embedding into the network, (2) perform cross-attention between modalities, (3) concatenate representations from all modalities, and (4) combine the predictions from each modality in an ensemble (Baltrušaitis et al., 2017). Researchers usually design the multimodal network by considering the task objective, the amount of data available, and the computation budget (Shi et al., 2021b; Chen et al., 2019b; Li et al., 2022; Tsai et al., 2019; Mahajan & Roth, 2020). Baltrušaitis et al. (2017) provides further readings.

Data augmentation for single modality tasks. Data augmentation is widely adopted in vision and natural language tasks. In vision, we can manually intervene on a per-task basis to apply transformations that should leave our label invariant — e.g., translations, rotations, flipping, cropping, and color adjustments. A transformation on one task may not be suitable for another: for example, flipping may be reasonable on CIFAR-10, but would lose semantic information on MNIST, because a flipped '6' becomes a '9'. Accordingly, there are a variety of works for automatic augmentations in vision, including neural architecture search (Ho et al., 2019; Cubuk et al., 2020), reinforcement learning (Cubuk et al., 2019), generative modelling (Ratner et al., 2017), mixing aspects of the existing data (Zhang et al., 2017; Yun et al., 2019), and adversarial training for informative examples (Fawzi et al., 2016; Goodfellow et al., 2015; Zhang et al., 2019; Suzuki, 2022; Tang et al., 2020b; Tsai et al., 2017; **?**) . Similarly, in natural language processing there are a variety of standard interventions (replacement, deletion, swapping) (Wei & Zou, 2019; Karimi et al., 2021; Fadaee et al., 2017), and more automatic approaches such as back-translation (Sennrich et al., 2015), context augmentation (Wang & Yang, 2015; Andreas, 2020; Kobayashi, 2018), and linear interpolation of training data (Sun et al., 2020a). Data augmentation is less explored for tabular data, but techniques in vision, such as mixup (Zhang et al., 2017) and adversarial training (Goodfellow et al., 2015) have recently been adapted to the tabular setting with promising results (Kadra et al., 2021). Latent space augmentation is much less explored than input augmentation, as it is less obvious what transformations to apply. To augment latent vectors produced by passing data inputs through a neural network (*feature space* augmentation), researchers have considered interpolation, extrapolation, noise addition, and generative models (Verma et al., 2019; Liu et al., 2018; Kumar et al., 2019).

Multimodal data augmentation. There are a small number of works considering multimodal data augmentation, primarily focusing on vision-text tasks. In visual question answering, Tang et al. (2020a) proposes to generate semantic similar data by applying back-translation on the text

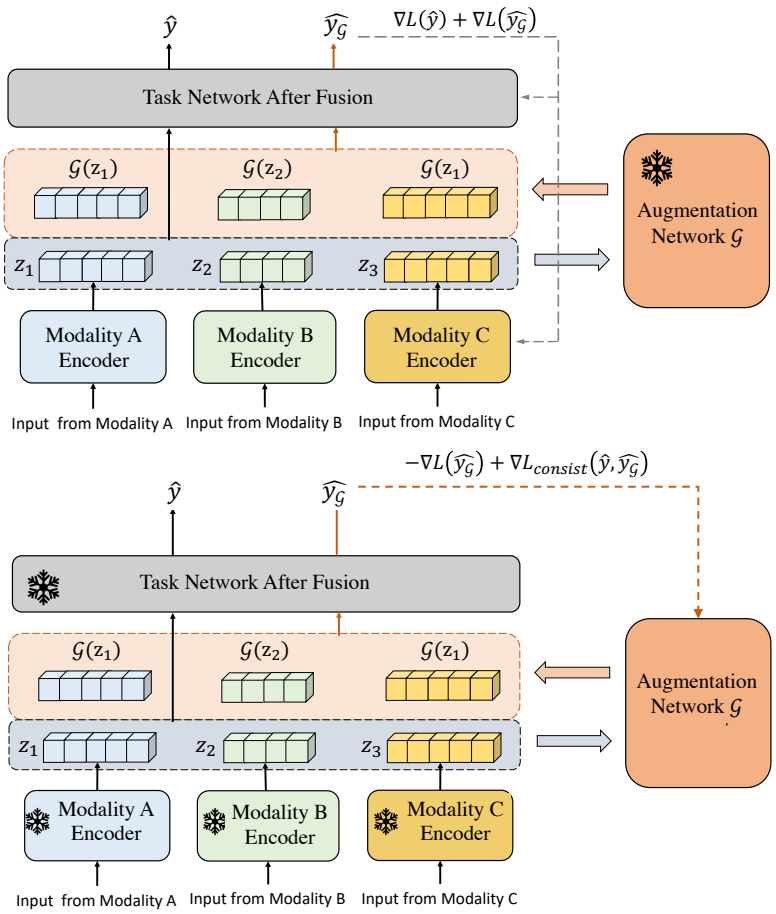


Figure 2: LeMDA training as described in Algorithm 1. **Top:** the training process for the task network. Latent representations for each modality z_i are passed into the augmentation network, which generates a new latent vector for each modality. Both original features and augmented features are passed into the rest of the task network. **Bottom:** the training process for the augmentation network. The augmentation network is trained to maximize task loss while minimizing consistency loss. We describe our standard choices for fusion in Section 2, and the design of our augmentation network in Section 3.3.

and adversarial noise on the image. Wang et al. (2021) generates text based on images using a variational autoencoder. In cross-modal retrieval, Gur et al. (2021) query similar data from external knowledge sources for cross-modal retrieval tasks. The state-of-the-art augmentation procedure for visual-language representation learning generates new image-text pairs by interpolating between images and concatenating texts in a method called *MixGen* (Hao et al., 2022).

All prior work on multimodal data augmentation relies on tailored modality-specific transformations. By contrast, our proposed approach is fully automatic and can be applied to any arbitrary modality. Indeed, for the first time, we consider augmentation jointly over the tabular, image, and language modalities. Moreover, even for image-text specific problems, we show that our approach outperforms MixGen, the state-of-the-art tailored approach.

3 LeMDA: Learning Multimodal Data Augmentation

We now introduce LeMDA, a simple and automatic approach to multi-modal data augmentation. LeMDA learns an *augmentation network* $\mathcal{G}$, along with the multimodal *task network* $\mathcal{F}$ to generate informative data that preserves semantic structure. In Sections 3.1 and 3.2 we describe how we learn the parameters the augmentation and task networks, respectively. We summarize the training algorithm for LeMDA in Figure 2 and Algorithm 1. In Section 3.4 we provide intuition for the consistency loss. Finally, in Section 3.3 we describe how we design the augmentation network.

Algorithm 1 LeMDA Training

Input: Task network before fusion $\mathcal{F}_{\text{before}}$; Task network after fusion $\mathcal{F}_{\text{after}}$; Augmentation network $\mathcal{G}$; Training set $\mathcal{X}$; Task loss function L; Consistency loss L_{consist};
while $\mathcal{F}$ not converged **do**
  Sample a mini-batch from $\mathcal{X}$
  Compute $z \leftarrow \mathcal{F}_{\text{before}}(x)$
  Generate augment feature $\mathcal{G}(z)$
  $\hat{y} \leftarrow \mathcal{F}_{\text{after}}(z)$, $\hat{y}_{\mathcal{G}} \leftarrow \mathcal{F}_{\text{after}}(\mathcal{G}(z))$
  Update the augmentation network $\mathcal{G}$ by stochastic gradient $-\nabla L(\hat{y}_{\mathcal{G}}) + \nabla L_{\text{consist}}(\hat{y}, \hat{y}_{\mathcal{G}})$
  Update the task network $\mathcal{F}$ by stochastic gradient $\nabla L(\hat{y}) + \nabla L(\hat{y}_{\mathcal{G}})$
end while

3.1 Training the Task Network

The task network can be divided into two parts at the fusion layer $\mathcal{F}(x) = \mathcal{F}_{\text{after}}(\mathcal{F}_{\text{before}}(x))$ where $\mathcal{F}_{\text{before}}$ denotes the layers before fusion, $\mathcal{F}_{\text{after}}$ denotes the layers after the fusion. Given a training sample x, we pass x until the fusion layer and obtain the latent features for each modality $\{z_i\}_{i=1}^N = \mathcal{F}_{\text{before}}(x)$ where N is the number of modalities. Taken $\{z_i\}_{i=1}^N$ as inputs, the augmentation network $\mathcal{G}$ generates additional latent vectors $\mathcal{G}(\{z_i\}_{i=1}^N)$. Both $\{z_i\}_{i=1}^N$ and $\mathcal{G}(\{z_i\}_{i=1}^N)$ are fed through the rest of target network $\mathcal{F}_{\text{after}}$ as distinct training data. Then, the task network is trained in the standard way, taking the task loss function on both original data and augmented data, to find $\min \mathbb{E}_{x\sim\mathcal{X}}(L(\hat{y}) + L(\hat{y}_{\mathcal{G}}))$ where $\hat{y} = \mathcal{F}_{\text{after}}(\mathcal{F}_{\text{before}}(x))$ and $\hat{y}_{\mathcal{G}} = \mathcal{F}_{\text{after}}(\mathcal{G}(\mathcal{F}_{\text{before}}(x)))$.

3.2 Training the Augmentation Network

Inspired by adversarial data augmentation, we optimize parameters for the augmentation network to maximize the task loss such that the task network's representation is encouraged to be updated by the augmented data. At the same time, we introduce a consistency regularizer that encourages a similar output distribution given the original data and the augmented data to preserve the semantic structure. Formally, we find $\max \mathbb{E}_{x\sim\mathcal{X}}(L(\hat{y}_{\mathcal{G}})) + \min \mathbb{E}_{x\sim\mathcal{X}}(L_{\text{consist}}(\hat{y}, \hat{y}_{\mathcal{G}}))$ where $L_{\text{consist}}(\hat{y}, \hat{y}_{\mathcal{G}})$ denotes a divergence metric between the logit outputs on original data $\hat{y}$ and on augmented data $\hat{y}_{\mathcal{G}}$ such as the Kullback-Leibler divergence.

Confidence masking. For classification problems, we apply the consistency term only to the samples whose highest probability is greater than a threshold α. If the task network can't make a confident prediction, it is unlikely the prediction provides a good reference to the ground truth label.

Design decisions. The simplicity and generality of this approach, combined with its strong empirical performance in Section 4, are LeMDA's most appealing features. The few design decisions for training involve how the consistency regularizer should be defined and to what extent it should be applied. For example, as an alternative to a KL-based consistency regularizer, we could minimize the L_2 distance of the augmented feature vector to the original feature vector as a proxy for preserving the label of the augmentation. We provide ablations of these factors in Section 4.4.

3.3 The Design of Augmentation Network

The augmentation network can take various forms depending on the multimodal learning tasks and the fusion strategies. In our experiments, we use a variational autoencoder (VAE) as the augmentation network, since VAEs have generally been found effective for augmentation purposes (Tang et al., 2020b). We consider two architectural choices:

MLP-VAE: The encoder and decoder of VAE are MLPs. $\{z_i\}_{i=1}^N$ are concatenated as the input.

Attention-VAE: The encoder and decoder are made of self-attention and feedforward networks. $\{z_i\}_{i=1}^N$ are treated as N tokens where each token has an embedding z_i.

There are two loss terms in the standard VAE, the reconstruction loss, and the KL divergence regularizer. We only adopt the KL regularizer on the encoder distribution. The updating step for augmentation networks is $-\nabla L(\hat{y}_{\mathcal{G}}) + \nabla L_{\text{consist}}(\hat{y}, \hat{y}_{\mathcal{G}}) + +\nabla L_{\text{VAE}}$, where L_{VAE} refers to the KL divergence regularizer on the latent encoding distribution.

The major deciding factor between MLP-VAE and Attention-VAE is the multimodal task network architectures. With late fusion architectures, which is the primary focus of this paper, z_i refers to the representation from a single modality backbone (e.g. CLS embedding from a BERT model), and N is the number of modalities or the number of backbone models. We can concatenate $\{z_i\}_{i=1}^{N}$ as one vector input to MLP-VAE, or we can treat $\{z_i\}_{i=1}^{N}$ as a sequence of N tokens to Attention-VAE. Attention-VAE may be less intuitive here because N is usually a small number in late fusion architectures(2 or 3 in our experiment). We provide a performance comparison between these two architectures in Section 4.4. On the other hand, for early fusion architectures, z_i could be a sequence of token embedding for a text or a sequence of patch embedding for an image. Concatenation will result in a really high dimension input, which makes MLP-VAE less favorable.

3.4 Intuition on Why Consistency Regularizer Discourages Mislabeled Data

In Figure 3 we provide intuition for the consistency regularizer using a simple illustrative binary classification. Darker background corresponds to higher task training loss, the solid green line is the actual decision boundary, and the dashed green line is the model's decision boundary. Starting from a point in feature space, moving to D1 and D2 would provide a similar increase in task loss and thus are equally favored by the adversarial loss term. However, D2 crosses the model's decision boundary, and thus would be heavily penalized by the consistency regularizer — as we would hope, since such a point is likely to have a different class label. On the other hand, an L2 regularizer between the original and augmented points in feature space would have no preference between D1 and D2, as they are an equal distance away from the starting point. Empirically, in Section 4, we see the consistency loss confers accuracy improvements over both pure adversarial training and L2 regularizer.

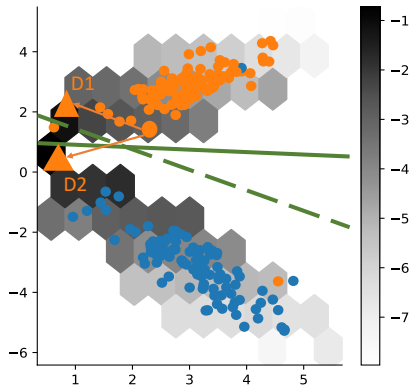


Figure 3: Motivation for the consistency regularizer. The solid and dashed green lines are the ground truth and model decision boundaries, respectively. Darker background corresponds to a higher loss for the task network. We intuitively prefer D1, because the augmented point should be informative but preserve the same label. The consistency loss will prefer D1 over D2, because D2 crosses the model's decision boundary, even though both points incur the same training loss.

Similar intuition is in Suzuki (2022), which uses the logits distribution from the teacher model (an exponential moving average over the model's weights) as the soft target such that the augmented data is still recognizable by the teacher, and Xie et al. (2019b), which designs the unsupervised training objective to encourage similar logits for augmented data.

4 Experiments

We evaluate LeMDA over a diverse set of real-world multimodal datasets. We curate a list of public datasets covering image, text, numerical, and categorical inputs. Table 1 provides a summary of the source, statistic, and modality identity. We introduce baselines in Section 4.1, and describe

Dataset	# Train	#Test	Metric	Image	Text	Tabular
Hateful Memes	7134	1784	Accuracy	✓	✓	
Food101	67972	22716	Accuracy	✓	✓	
SNLI-VE	529527	17901	Accuracy	✓	✓	
Petfinder	11994	2999	Quadratic Kappa	✓	✓	✓
Melbourne Airbnb	18316	4579	Accuracy		✓	✓
News Channel	20284	5071	Accuracy		✓	✓
Wine Reviews	84123	21031	Accuracy		✓	✓
Kick Starter Funding	86502	21626	ROC-AUC		✓	✓

Table 1: This table provides a summary of the source, statistic, and modality identity.

experimental settings in Section 4.2 We provide the main evaluation result in Section 4.3. Finally, we investigate the effects of the consistency regularizer and the choices of augmentation model architecture in Section 4.4.

4.1 BASELINES

To the best of our knowledge, there exist no general-purpose multimodal augmentation methods. We compare against a diverse set of state-of-the-art data augmentation methods from vision, language, and vision-text tasks. We additionally consider baselines for feature augmentation, since LeMDA augments in the feature space. Finally, we compare with state-of-the-art multimodal augmentation methods from the vision-text tasks, although we note, unlike LeMDA, these methods are not general purpose and cannot be directly applied to our datasets that have tabular inputs.

- **Input Augmentation.** We apply state-of-the-art input augmentation independently on the data from each modality. For images, we use TrivialAugment (Müller & Hutter, 2021), a simple and effective method for image classification tasks. For text, we apply EDA (Wei & Zou, 2019) and AEDA (Karimi et al., 2021). We randomly sample one transformation from all transformations proposed in EDA and AEDA with a randomly generated magnitude.
- **Mixup.** Mixup was originally proposed to perform interpolation between two images in the training data for image classification. We adopt the original Mixup for images and numerical features and extend it for text and categorical features. Specifically, given a pair of data, we construct the mixed data as follows. We generate a random number j uniformly between 0.0 to 1.0. If $j < \alpha$, we use the first data, else we use the second data.
- **Manifold Mixup.** Manifold Mixup (Verma et al., 2019) performs interpolation between hidden representations and thus can be applied to all modalities. We applied Manifold Mixup to the exact feature in the multimodal network as LeMDA.
- **MixGen.** MixGen (Hao et al., 2022) is a state-of-the-art data augmentation designed specifically for vision-text tasks. MixGen generates new data by interpolating images and concatenating text. We apply MixGen to datasets only consisting of images and text.

4.2 EXPERIMENT SETUP

We use Multimodal-Net (Shi et al., 2021a) for all the datasets except SNLI-VE. Multimodal-Net passes input from each modality through separate backbones, concatenates the representation(e.g. the CLS embedding) from all backbones, and passes them through fusion MLP. We use the default hyper-parameters provided by Multimodal-Net and plug LeMDA before the fusion layer. We use ConvNet as the image backbone and ELECTRA as the text backbone.

To further demonstrate LeMDA's generalizability, we evaluate LeMDA with early fusion architectures ALBEF (Li et al., 2021) on SNLI-VE. ALBEF performs cross-attention between image patch embeddings and text token embeddings. We keep all original configurations except the batch size due to limitations in computation memory. We set the batch size to be half of the default. We load the 4M pre-trained checkpoints. In this setting, we apply LeMDA before the cross-attention layer. The augmentation network augments every image patch embedding and every text token embedding.

For LeMDA, we set the confidence threshold for consistency regularizer α as 0.5, and we study this choice in Section 4.4. For our baselines, we follow the recommended hyperparameters. For Mixup and Manifold Mixup, we set α as 0.8, and for MixGen, we set λ as 0.5.

4.3 MAIN RESULTS

We summarize the performance comparison in Table 2. Plugging LeMDA in both Multimodal-Net and ALBEF leads to consistent accuracy improvements. There are also some particularly notable improvements, such as a 6% increase in accuracy for both Hateful Memes and Petfinder. Table 2 illustrates how LeMDA performs comparing to the baselines. We see that single modality input augmentation methods can hurt accuracy, for example, on News Channel, in accordance with the intuition from our introductory example in Figure 1. Mixup also can hurt accuracy, for example, on Wine Reviews. Similarly, in the latent space, Manifold Mixup fails to improve accuracy across datasets. On Melbourne Airbnb and Wine Reviews, Manifold Mixup results in accuracy drops. On

	Multimodal Network	Input Augmentation	Mixup	Manifold MixUp	MixGen	LeMDA
Hateful Memes	0.6939	0.7057	0.6939	0.6878	0.7510	**0.7562**
Food101	0.9387	0.9432	0.9400	0.9390	0.9432	**0.9452**
Petfinder	0.2911	0.3236	0.3244	0.3492	-	**0.3537**
Melbourne Airbnb	0.3946	0.3978	0.3966	0.3840	-	**0.4047**
News Channel	0.4754	0.4745	0.4723	0.4757	-	**0.4798**
Wine Reviews	0.8192	0.8212	0.8143	0.8126	-	**0.8262**
Kick Starter Funding	0.7571	0.7572	0.7597	0.7578	-	**0.7614**
SNLI-VE	0.7916	0.7931	0.7957	0.7929	0.7950	**0.7981**

Table 2: LeMDA not only significantly increases accuracy over the original architectures but also outperforms all baselines.

the contrary, LeMDA consistently improves upon original architectures and provides clearly better performance than a wide range of baselines.

4.4 Ablation Study

We now perform three ablation studies to support the design choice of LeMDA.

Dataset	No Regularizer	Consistency	L2	Consistency + L2
Hateful Memes	0.7433	**0.7562**	0.7472	0.7545
Food101	0.9433	**0.9452**	0.9415	0.9438
Petfinder	0.3369	**0.3537**	0.3420	0.3461
Melbourne Airbnb	0.3935	0.4047	0.3987	**0.4051**
News Channel	0.4851	0.4869	0.4869	**0.4894**
Wine Reviews	0.8228	**0.8263**	0.8255	0.8262
Kick Starter Funding	0.7609	**0.7614**	0.7604	**0.7614**

Table 3: The effects of regularizer choice. Regularization over the augmentation network generally lead to better performance. Consistency regularizer consistently outperforms a standard L2 regularizer in feature space. Moreover, combining the consistency regularizer with an L2 regularizer improves over only using an L2 regularizer.

Dataset	No Augmentation	MLP-VAE	Attention-VAE
Hateful Memes	0.6939	**0.7562**	0.7483
Food101	0.9387	**0.9452**	0.9443
Petfinder	0.2911	**0.3537**	0.3456
Melbourne Airbnb	0.3946	**0.4047**	0.4031
News Channel	0.4754	**0.4798**	0.4733
Wine Reviews	0.8192	**0.8262**	0.8250
Kick Starter Funding	0.7571	**0.7614**	0.7586

Table 4: Both MLP-VAE and Attention-VAE augmentation networks provide significant gains over no augmentation. MLP-VAE outperforms Attention-VAE in the late fusion setting because the input of augmentation networks is only 2 or 3 latent representations.

Augmentation Network Regularizer. We argue in Section 3.4 that consistency regularizer helps preserve the semantic structure of augmentations. In Table 3, we see that this consistency regularizer significantly improves performance, and also outperforms L2 regularization in feature space. While L2 regularization attempts to keep augmented features close in distance to the original as a proxy for semantic similarity, the consistency regularization has access to the softmax outputs of the target and augmentation networks, providing direct information about labels.

Architecture Difference. We consider the two augmentation architectures introduced in Section 3.3, MLP-VAE and Attention-VAE. In Table 4 we see both architectures increase performance over no augmentation. We also see that MLP-VAE generally outperforms Attention-VAE. We suspect the reason is that Multimodal-Net passes the concatenation of N latent vector into fusion layers,

Dataset	$\alpha = 0$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.8$
Hateful Memes	0.7410	0.7443	**0.7556**	0.7371
Food101	0.9431	0.9438	**0.9447**	0.9438
Petfinder	0.3243	0.3462	**0.3676**	0.3497
Melbourne Airbnb	0.3988	0.3964	**0.3988**	0.3964
News Channel	**0.4869**	0.4851	**0.4869**	0.4695
Wine Reviews	0.8228	0.8274	**0.8275**	0.8274
Kick Starter Funding	0.7614	0.7617	**0.7620**	0.7618

Table 5: The influence of confidence-based masking. $\alpha = 0$ indicates no masking such that consistency loss is calculated with all data. We see that filtering out low-confidence data leads to better end-to-end accuracy.

where N is the number of modalities (2 or 3 in our experiments). For Attention-VAE, this means that the input is only 2 or 3 tokens. However, we note that MLP-VAE is not reasonable for ALBEF, since it would require concatenating thousands of tokens.

Confidence Masking. Here, we investigate the effect of confidence masking, as well as the choice of α in Table 5. $\alpha = 0$ means no masking, and all training data are used to calculate consistency loss. We see that confidence masking generally leads to higher accuracy, and that the performance is not particularly sensitive to the precise value of α.

4.5 The Relationship between Modalities

We can categorize the relationship between available modalities by looking at $P(y|x)$ where $y \sim \mathcal{Y}$ and $\mathcal{Y}$ is the target domain. Let $x = \{x_1, x_2, \ldots, x_N\}$ consist of N modalities.

Perfect Correlation $P(y|x) = P(y|x_n)$. Essentially, one modality alone provides enough information to make the right prediction. Nevertheless, data still comprises multiple modalities for reasons such as easier training (Huang et al., 2021). One example could be Food101, where the task is to predict the food from the text of a recipe and the photo of the food.

Complementary $P(y|x) = P(y|\{x_1, x_2, \ldots, x_N\})$. This category suggests that information aggregated from all modalities is necessary to make the right prediction. Each modality is complementary to each other, and missing one modality would lead to information loss. One example could be Hateful Memes, where only the combined meaning of text and image indicates harmful content.

The design for LeMDA does not exploit any assumption over the cross-modal relationship. We observe from Table 2 that LeMDA consistently improves performance regardless of the relationship.

5 Conclusion

Jointly learning from multiple different modalities will be crucial in our quest to build autonomous intelligent agents. We introduce the first method, LemDA, for jointly learning data augmentation across arbitrary modalities. LeMDA is simple, automatic, and achieves promising results over a wide range of experiments. Moreover, our results provide several significant conceptual findings about multimodal data augmentation in general: (1) separately augmenting each modality performs much worse than joint augmentation; (2) although feature augmentation is less popular than input augmentation for single-modality tasks because it is less interpretable, feature augmentation is particularly promising for modality-agnostic settings; (3) a learning-based multimodal augmentation policy can outperform even tailored augmentations, and significantly improve accuracy when augmentation transformations are not obvious such as for categorical data.

Our investigation has primarily focused on late-fusion architectures, showing strong results over a wide range of settings. In general, applying feature augmentation strategies to early-fusion architectures is an open question. Early fusion combines a large number of latent features (e.g., a long sequence of token embeddings), resulting in typically intractable computational costs for augmenting every latent feature. Our experiment with an early-fusion architecture shows however that developing more efficient augmentation networks, or selectively generating only a few important latent vectors, is a promising direction for future work.

ETHICS STATEMENT

We anticipate no direct harmful social consequences from this paper. We believe that data augmentations are beneficial in terms of data efficiency and training efficiency, helping to save costs in data collection. We believe that investigating data augmentation for multimodal deep learning settings is helpful to the community in building less data hungry models.

REPRODUCIBILITY STATEMENT

We use AWS for experiments. We carefully describe out experimental setup, and provide ablations for our design choices. Moreover, we will make code available upon publication.

REFERENCES

Jacob Andreas. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7556–7566, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.676. URL https://aclanthology.org/2020.acl-main.676.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy, 2017. URL https://arxiv.org/abs/1705.09406.

George Barnum, Sabera Talukder, and Yisong Yue. On the benefits of early fusion in multimodal representation learning, 2020. URL https://arxiv.org/abs/2011.07191.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: learning universal image-text representations. *CoRR*, abs/1909.11740, 2019a. URL http://arxiv.org/abs/1909.11740.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning, 2019b. URL https://arxiv.org/abs/1909.11740.

Jun-Ho Choi and Jong-Seok Lee. Embracenet: A robust deep learning architecture for multimodal classification. *Information Fusion*, 51:259–270, 2019.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18613–18624. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d85b63ef0ccb114d0a3bb7b7d808028f-Paper.pdf.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 567–573, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2090. URL https://aclanthology.org/P17-2090.

Alhussein Fawzi, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. Adaptive data augmentation for image classification. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3688–3692, 2016. doi: 10.1109/ICIP.2016.7533048.

Jing Gao, Peng Li, Zhikui Chen, and Jianing Zhang. A Survey on Deep Learning for Multimodal Data Fusion. *Neural Computation*, 32(5):829–864, 05 2020. ISSN 0899-7667. doi: 10.1162/neco_a_01273. URL https://doi.org/10.1162/neco_a_01273.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.

Shir Gur, Natalia Neverova, Chris Stauffer, Ser-Nam Lim, Douwe Kiela, and Austin Reiter. Cross-modal retrieval augmentation for multi-modal classification, 2021. URL https://arxiv.org/abs/2104.08108.

Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. Mixgen: A new multi-modal data augmentation, 2022. URL https://arxiv.org/abs/2206.08358.

Daniel Ho, Eric Liang, Ion Stoica, P. Abbeel, and Xi Chen. Population based augmentation: Efficient learning of augmentation policy schedules. In *ICML*, 2019.

Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. What makes multi-modal learning better than single (provably), 2021. URL https://arxiv.org/abs/2106.04538.

Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned simple nets excel on tabular datasets. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=d3k38LTDCyO.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. AEDA: an easier data augmentation technique for text classification. *CoRR*, abs/2108.13230, 2021. URL https://arxiv.org/abs/2108.13230.

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021a.

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision, 2021b. URL https://arxiv.org/abs/2102.03334.

Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 452–457, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2072. URL https://aclanthology.org/N18-2072.

Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. A closer look at feature space data augmentation for few-shot intent classification. *CoRR*, abs/1910.04176, 2019. URL http://arxiv.org/abs/1910.04176.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven C. H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *CoRR*, abs/2107.07651, 2021. URL https://arxiv.org/abs/2107.07651.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

Xiaofeng Liu, Yang Zou, Lingsheng Kong, Zhihui Diao, Junliang Yan, Jun Wang, Site Li, Ping Jia, and Jane You. Data augmentation via latent space interpolation for image classification. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 728–733, 2018. doi: 10.1109/ICPR.2018.8545506.

Shweta Mahajan and Stefan Roth. Diverse image captioning with context-object split latent spaces. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3613–3624. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/24bea84d52e6a1f8025e313c2ffff50a-Paper.pdf.

Shweta Mahajan, Iryna Gurevych, and Stefan Roth. Latent normalizing flows for many-to-many cross-domain mappings, 2020. URL https://arxiv.org/abs/2002.06661.

Samuel G. Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation, 2021. URL https://arxiv.org/abs/2103.10158.

Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. *CoRR*, abs/2107.00135, 2021. URL https://arxiv.org/abs/2107.00135.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.

Juan-Manuel Pérez-Rúa, Valentin Vielzeuf, Stéphane Pateux, Moez Baccouche, and Frédéric Jurie. Mfas: Multimodal fusion architecture search, 2019. URL https://arxiv.org/abs/1903.06496.

Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. *Advances in neural information processing systems*, 30, 2017.

Edgar Schönfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Cross-linked variational autoencoders for generalized zero-shot learning, 2019. URL https://openreview.net/forum?id=BkghJoRNO4.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data, 2015. URL https://arxiv.org/abs/1511.06709.

Xingjian Shi, Jonas Mueller, Nick Erickson, Mu Li, and Alex Smola. Multimodal automl on structured tables with text fields. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021a.

Xingjian Shi, Jonas Mueller, Nick Erickson, Mu Li, and Alexander J Smola. Benchmarking multimodal automl for tabular data with text fields. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021b.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S. Yu, and Lifang He. Mixup-transformer: Dynamic data augmentation for nlp tasks, 2020a. URL https://arxiv.org/abs/2010.02394.

Xinwei Sun, Yilun Xu, Peng Cao, Yuqing Kong, Lingjing Hu, Shanghang Zhang, and Yizhou Wang. TCGM: an information-theoretic framework for semi-supervised multi-modality learning. *CoRR*, abs/2007.06793, 2020b. URL https://arxiv.org/abs/2007.06793.

Teppei Suzuki. Teachaugment: Data augmentation optimization using teacher knowledge, 2022. URL https://arxiv.org/abs/2202.12513.

Ruixue Tang, Chao Ma, W. Zhang, Qi Wu, and Xiaokang Yang. Semantic equivalent adversarial data augmentation for visual question answering. *ArXiv*, abs/2007.09592, 2020a.

Zhiqiang Tang, Yunhe Gao, Leonid Karlinsky, Prasanna Sattigeri, Rogerio Feris, and Dimitris Metaxas. Onlineaugment: Online data augmentation with less domain knowledge, 2020b. URL https://arxiv.org/abs/2007.09271.

Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. Learning robust visual-semantic embeddings, 2017. URL https://arxiv.org/abs/1703.05908.

Yao-Hung Hubert Tsai, Paul Pu Liang, Amir Zadeh, Louis-Philippe Morency, and Ruslan Salakhutdinov. Learning factorized multimodal representations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rygqqsA9KX.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6438–6447. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/verma19a.html.

Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks, 2017. URL https://arxiv.org/abs/1704.03470.

William Yang Wang and Diyi Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2557–2563, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1306. URL https://aclanthology.org/D15-1306.

Zixu Wang, Yishu Miao, and Lucia Specia. Cross-modal generative augmentation for visual question answering, 2021. URL https://arxiv.org/abs/2105.04780.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019. URL https://arxiv.org/abs/1901.11196.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019a.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training, 2019b. URL https://arxiv.org/abs/1904.12848.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. *arXiv preprint arXiv:1912.11188*, 2019.

A MORE DETAILS ON THE DESIGN

A.1 DETAILED ARCHITECTURES OF THE AUGMENTATION NETWORK

We consider two VAE architectures in LeMDA, depending on the architecture of the task network. The latent dimension in VAE is set as 8. We adopted the KL divergence regularizer on the encoder distribution. Note that we do not use the reconstruction loss between the input and the output. In MLP-VAE, the encoder and decoder are standard fully connected layers with ReLU as the activation function. Dropout is used with $p = 0.5$. In Attention-VAE, the encoder is implemented as torch.nn.TransformerEncoder. We set numlayers as 4 and nhead as 8. One fully connected layer is used to mapThe decoder is symmetric as the encoder. Features from all modalities are treated as token embedding with no cross-attention.

We use VAE for its simplicity. The main focus of this paper is to demonstrate the effectiveness of a learnable augmentation network for multimodal learning. Other generative models, such as diffusion models and GANs, are also valid architectures. The main concern may lie in efficiency, and we leave this direction as future work.

A.2 IMPLEMENTATION DETAILS OVER THE TRAINING PROCEDURE

In practice, we iterative train the task and augmentation networks using the same batch of training data. Specifically, we perform two separate forward passes using $\mathcal{F}_{\mathsf{after}}$ for easy implementation with pyTorch Autograd. We use two optimizers, one for the task network and one for the augmentation network.

B EXPERIMENT DETAILS

B.1 ADDITIONAL STUDIES ON THE TRAINING COST

One limitation of a learning-based approach is the extra training cost. LeMDA optimizes the augmentation network along with the task network and does incur extra training costs. Here, we investigate the training throughput to provide a more complete understanding of the method. We summarize the training throughput(it/second) in Table 6. As expected, we observe lower throughput for LeMDA compared to other baselines.

	Multimodal Network	Input Augmentation	Mixup	Manifold MixUp	MixGen	LeMDA
Hateful Memes	2.39	2.17	2.35	1.63	2.35	1.41
Food101	4.27	4.46	4.31	4.48	4.47	2.21
Petfinder	2.36	2.29	2.95	2.36	-	1.87
Melbourne Airbnb	5.66	5.94	5.59	5.69	-	4.13
News Channel	8.14	7.18	7.31	7.12	-	5.12
Wine Reviews	12.54	11.60	11.89	11.46	-	6.28
Kick Starter Funding	12.37	12.57	12.62	12.21	-	6.69

Table 6: This table summarizes the training throughput, measured as it/second. Experiments were conducted on a server with 8 V100 GPU. As expected, learning-based approach incur higher training cost.

However, efficiency can be improved. The most straightforward direction is to reduce the frequency of updating the augmentation network. Currently, the augmentation network is updated every iteration. However, the parameters for our task network change slowly, especially in the later stage of training. We leave this part as future direction.

B.2 ADDITIONAL STUDIES ON THE HYPER-PARAMETERS

The optimization for the augmentation network is a min-max game, which leads to hyperparameters to balance the contradicting loss. Specifically, $-w_1 \nabla L(\hat{y}_{\mathcal{G}}) + w_2 \nabla L_{\mathsf{consist}}(\hat{y}, \hat{y}_{\mathcal{G}}) + w_3 \nabla L_{\mathsf{VAE}}$, where L_{VAE} refers to the KL divergence regularizer on the latent encoding distribution.

In our main experiment, we use $w_1 = 0.0001, w_2 = 0.1, w_3 = 0.1$ on all datasets except Melbourne Airbnb and SNLI-VE. On Melbourne Airbnb and SNLI-VE, we use $w_1 = 0.001, w_2 = 0.1, w_3 = 0.1$. Note that the hyperparameters are relative consistent across datasets.

Further, we investigate the influence of the different combinations of w_1, w_2, and w_3. We summarize the result on Petfinder Table 7. We observe consistent improvements over the original multimodal network across various combinations.

w_1	w_2	w_3	Accuracy
0.0001	0.1	0.1	0.3539
0.0001	0.01	0.01	0.3400
0.005	0.1	0.1	0.3482
0.005	0.01	0.01	0.3464
0.001	0.1	0.1	0.3371
0.001	0.01	0.01	0.3467
Multimodal Network			0.2911

Table 7: LeMDA improves over Multimodal Network in accuracy with different sets of hyperparameters on Petfinder. Specifically, $-w_1 \nabla L(\hat{y}_{\mathcal{G}}) + w_2 \nabla L_{\mathsf{consist}}(\hat{y}, \hat{y}_{\mathcal{G}}) + w_3 \nabla L_{\mathsf{VAE}}$

C MOTIVATIONAL EXAMPLES WHEN ONLY AUGMENTING ONE MODALITY

We have performed an additional set of experiments to investigate the effect of augmenting a single modality on Hateful Memes using state-of-the-art augmentation techniques. In Hateful Memes, both text and image are required to decide if the content is hateful. Two modalities provide complementary information to each other. We run baseline augmentation to only one modality or independently on two modalities. We observe no consistent improvements. Essentially, performing augmentation naively to one modality or jointly without considering cross-modality relationships won't lead to effective augmentation.

Multimodal Network	Method	Image	Text	Image + Text
0.6939	Trivial Augment	0.7040	0.6860	0.7057
	MixUp	0.6855	0.6777	0.6939
	Manifold Mixup	0.6323	0.7444	0.6878
	MixGen	0.7427	0.6872	0.7510