MixGen: A New Multi-Modal Data Augmentation

Xiaoshuai Hao*
Institute of Information Engineering, CAS

Yi Zhu
Amazon Web Services

Srikar Appalaraju
Amazon Web Services

Aston Zhang
Amazon Web Services

Wanqian Zhang, Bo Li
Institute of Information Engineering, CAS

Mu Li
Amazon Web Services

Abstract

Data augmentation is a necessity to enhance data efficiency in deep learning. For vision-language pre-training, data is only augmented either for images or for text in previous works. In this paper, we present MixGen: a joint data augmentation for vision-language representation learning to further improve data efficiency. It generates new image-text pairs with semantic relationships preserved by interpolating images and concatenating text. It's simple, and can be plug-and-played into existing pipelines. We evaluate MixGen on four architectures, including CLIP, ViLT, ALBEF and TCL, across five downstream vision-language tasks to show its versatility and effectiveness. For example, adding MixGen in ALBEF pre-training leads to absolute performance improvements on downstream tasks: image-text retrieval (+6.2% on COCO fine-tuned and +5.3% on Flicker30K zero-shot), visual grounding (+0.9% on RefCOCO+), visual reasoning (+0.9% on NLVR^2), visual question answering (+0.3% on VQA2.0), and visual entailment (+0.4% on SNLI-VE).

1. Introduction

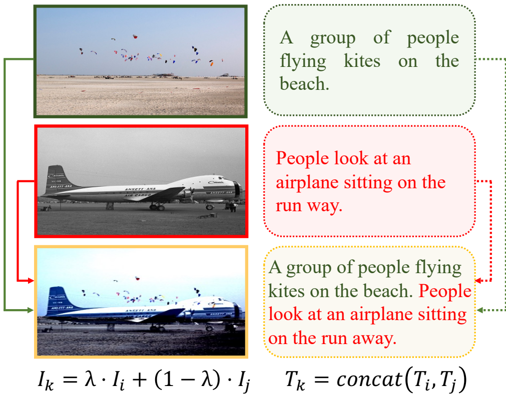
Recent years have witnessed an explosion in vision-language research [31, 42, 3, 21, 37, 14, 48, 1]. In joint modality learning, models extract rich information across modalities to learn better latent representations. These models, however, are often trained on a massive number of image-text pairs using thousands of GPUs. For example, CLIP [38] matches the accuracy of ResNet-50 [13] on ImageNet by only using zero-shot, but it is trained with 400M image-text pairs for 12 days on 256 V100 GPUs. Furthermore, most of these large-scale datasets [38, 15, 63] are not publicly accessible. Even if they are available, reproduction and further improvement on existing methods are challenging for researchers with limited computing resources.

Data augmentation is widely used in deep learning to improve data efficiency and provide explicit regularization during model training in both computer vision (CV) [9, 65, 6, 64, 7, 68, 46] and natural language processing (NLP) [66, 49, 16, 54, 51, 5, 40, 59]. Applying existing data augmentation techniques to vision-language learning, however, is not straightforward. In an image-text pair, both the image and the text contain rich information that matches each other. Intuitively, we hope that their semantics still match after data augmentation. For example, consider an image with its paired sentence "a *white* dog playing in the *right* corner of the *green* lawn". Applying data augmentation methods such as cropping, color changing, and flipping to this image may require changing the color and positional words in its paired sentence at the same time.

In order to preserve the semantic relationship, previous works perform mild data augmentation on *either* vision *or* text modality. ViLT [19] and following works [48, 23] adopt RandAugment [7] for image augmentations without color inversion. CLIP [38] and ALIGN [15] only use random resized crop without other image augmentations. In terms of language, most literature just leaves text data augmentation to be handled by masked language modeling [8]. There are works using co-augmentation [45, 58], but only designed for specific downstream tasks, rather than generic vision-language pre-training.

In this work, we introduce a multi-modal joint data augmentation method for pre-training: **Mix Gen**eration (MixGen). As shown in Figure 1, MixGen generates a new training sample by linearly interpolating images and concatenating text sequences from two existing image-text pairs. We can see that most objects and scene layout remain in the blended image, while the text information is fully preserved. The semantic relationship within the newly generated image-text pair is expected to match in most cases.

*Work done during an internship at AWS. First three authors contribute equally.



Algorithm 1 Pseudocode of MixGen data augmentation

```
# image, text:  a batch of B randomly sampled {image, text}
# M: number of new image-text pairs generated by MixGen
# λ:  image mixup ratio

def mixgen(image, text, M, λ):
    for i in range(M):
        # image mixup
        image[i,:]  = (λ * image[i,:]  + (1-λ) * image[i+M,:])
        # text concatenation
        text[i] = text[i] + " " + text[i+M]
    return image, text
```

Figure 1. Our proposed multi-modal data augmentation technique: **MixGen**. Given any two image-text pairs (I_i, T_i) and (I_j, T_j), we interpolate the two images and concatenate two text sequences, to generate a new image-text pair (I_k, T_k). Figure best viewed in color.

Thus, we can use the augmented data to improve vision-language model training.

Despite its simplicity, using MixGen on top of strong baselines (e.g., ALBEF [23]) consistently improves state-of-the-art performance across five downstream vision-language tasks: image-text retrieval (+6.2% on COCO fine-tuned and +5.3% on Flicker30K zero-shot), visual grounding (+0.9% on RefCOCO+), visual reasoning (+0.9% on NLVR^2), visual question answering (+0.3% on VQA2.0), and visual entailment (+0.4% on SNLI-VE). MixGen also leads to enhanced data efficiency, e.g., the performance of ALBEF with MixGen when pre-trained on 1M/2M/3M samples match baseline ALBEF pre-trained on 2M/3M/4M samples, respectively. In addition, we perform extensive ablation studies to understand the effects of various design choices in MixGen. Finally, MixGen can be incorporated into most methods with only a few lines of code. In terms of fine-tuned image-text retrieval on COCO, MixGen brings absolute improvements on four popular and varied architectures: ViLT (+17.2%), CLIP (+4.1%), ALBEF (+7.0%) and TCL (+3.2%).

2. MixGen

In this section, we introduce our multi-modal joint data augmentation technique: **Mix Gen**eration (MixGen). Suppose that we have a dataset of N image-text pairs, where images and text are denoted as I and T with subscripts, respectively. Given two image-text pairs (I_i, T_i) and (I_j, T_j) for any $i, j \in \{1, \dots, N\}$ and $i \neq j$, a new training sample (I_k, T_k) is generated via

$$\begin{aligned} I_k &= \lambda \cdot I_i + (1-\lambda) \cdot I_j \\ T_k &= \text{concat}(T_i, T_j), \end{aligned} \tag{1}$$

where λ is a hyper-parameter between 0 and 1, indicating linear interpolation between raw pixels of two images I_i and I_j; and the concat operator directly concatenates two text sequences T_i and T_j to best preserve original information. In this way, the semantic relationship within the newly generated image-text pair (I_k, T_k) still holds in most scenarios, such as in Figure 1 and Figure 2. This random combination of image-text samples also increase diversity to model training, which elicits making available rare concepts.

The pseudocode of MixGen data augmentation is presented in Algorithm 1. Given a minibatch of B randomly sampled image-text pairs, MixGen replaces the first M training samples with the newly generated pairs. Hence, the batch size, total training iterations and overall training pipeline remain the same. By default, we set $\lambda = 0.5$ and $M = B/4$ in Algorithm 1. Such a plug-and-play technique can be easily incorporated into most vision-language representation learning approaches and tasks: all it takes is just a few lines of code with minimal computational overhead. Take ALBEF [23] as an example, adding MixGen on top of it during pre-training only increases its overall wall clock training time by 0.4%.

2.1. MixGen variants

MixGen is in a very simple form (Algorithm 1). However, there could be multiple variants depending on how image and text augmentations are performed. Theoretically, we could also use other augmentations for images beyond mixup, and other text augmentations beyond concatenation, but the design space will be intractable. Thus, we focus on using mixup for image and concatenation for text, and select 5 most straightforward variants of MixGen to support our final design choice.

Since default MixGen takes a fixed λ, we introduce variant (a) with $\lambda \sim$ Beta(0.1, 0.1), following original mixup [65] that samples λ from a Beta distribution. In order to show the benefit of performing joint image-text augmentation, we propose variant (b) and (c). To be specific,

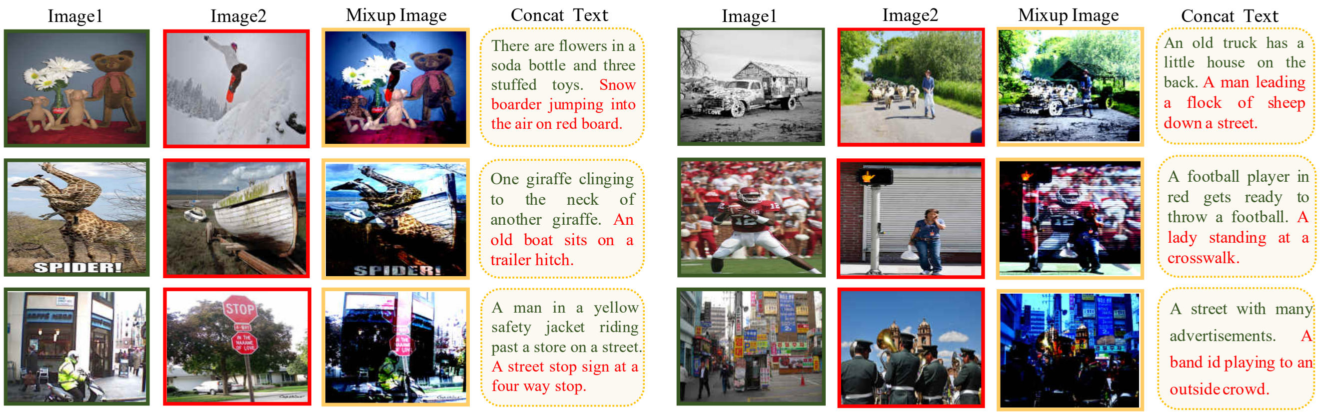


Figure 2. **More image-text pairs generated by MixGen**. The new pairs not only preserve original semantic relationships, but also increase diversity to model training. Figure best viewed in color.

variant (b) mixup two images and uniformly pick one text sequence, while variant (c) concatenates two text sequences and uniformly pick one image. In the end, we study whether we should use a subset of tokens, instead of concatenating all tokens from two text sequences. Variant (d) takes tokens from two text sequences proportionally based on λ similar as image mixup and then concatenate. The other variant (e) first concatenates all the tokens, but randomly keep half of them to generate a new text sequence.

More detailed definitions of these 5 variants can be seen in Table 6. We also perform extensive ablation studies on them. As we will see in Table 7, our default MixGen achieves the overall best performance, and consistently outperforms other variants across four different vision-language downstream tasks.

2.2. Input-level and embedding-level MixGen

Another design perspective is where to apply the data augmentation. The formulation in Equation (1) is directly performed on the raw input, e.g., images and text sequences. Alternatively, the idea of MixGen can be applied on the embedding level. To be specific, instead of interpolating raw image pixels, we can interpolate image features that are extracted from an image encoder. Similarly, instead of concatenating two text sequences, we can concatenate two sequence features that are extracted from a text encoder. Denoting the training pairs with respect to their embedding as (f_{I_i}, f_{T_i}) and (f_{I_j}, f_{T_j}), the newly generated training pair in its embedding form is

$$\begin{aligned} f_{I_k} &= \lambda \cdot f_{I_i} + (1-\lambda) \cdot f_{I_j} \\ f_{T_k} &= \text{concat}(f_{T_i}, f_{T_j}). \end{aligned} \tag{2}$$

We call MixGen that is performed on the raw input as *input-level MixGen*, and that performed on the embedding level as *embedding-level MixGen*. As we will show in Figure 6 left, input-level MixGen consistently performs better than embedding-level MixGen. In addition, input-level MixGen has an advantage of implementational simplicity, since we don't need to modify network architecture nor touch model forward. Hence, we refer to *input-level MixGen* as our proposed MixGen for the rest of the paper unless otherwise stated.

3. Experiments

In this section, we first describe model pre-training in Sec. 3.1. Then we introduce five downstream tasks (image-text retrieval, visual question answering, visual entailment, visual reasoning and visual grounding) and present experimental results of using MixGen in Sec. 3.2 and Sec. 3.3. In the end, we show visualizations in Sec. 3.4 to demonstrate the benefits of using MixGen.

3.1. Pre-training

Pre-training methods Data augmentation techniques are usually applicable to various models, because they are agnostic to network architectures, training objectives, optimizers, or learning schedules. In this study, we plug-and-play MixGen on four recent popular approaches with varied architectures, CLIP [38] (dual-encoder), ViLT [19] (single fusion encoder), ALBEF [23] (dual-encoder followed by fusion encoder) and TCL [60] (dual-encoder followed by fusion encoder). Also note, the pre-training objectives for these approaches are varied. We emphasize that MixGen can be easily incorporated to other vision-language pre-training methods with only a few lines of code. To be specific, once we get a minibatch out of the dataloader, we apply MixGen to generate new image-text pairs and update the minibatch. We then forward the updated batch to the network without modifying any of their original training settings.

Pre-training datasets There are four widely adopted datasets for vision-language model pre-training, includ-

Table 1. **Fine-tuned image-text retrieval** on Flickr30K and MSCOCO datasets.

Method	#Images	MSCOCO(5K test set)							Flickr30K(1K test set)						
		Text Retrieval			Image Retrieval				Text Retrieval			Image Retrieval			
		R@1	R@5	R@10	R@1	R@5	R@10	RSUM	R@1	R@5	R@10	R@1	R@5	R@10	RSUM
UNITER-base [3]	4M	64.4	87.4	93.1	50.3	78.5	87.2	460.9	85.9	97.1	98.8	72.5	92.4	96.1	542.8
VILLA-base [10]	4M	-	-	-	-	-	-	-	86.6	97.9	99.2	74.7	92.9	95.8	547.1
OSCAR-base [27]	4M	70.0	91.1	95.5	54.0	80.8	88.5	479.9	-	-	-	-	-	-	-
UNIMO-base [26]	4M	-	-	-	-	-	-	-	89.7	98.4	99.1	74.7	93.4	96.1	551.4
ViLT-base [19]	4M	61.5	86.3	92.7	42.7	72.9	83.1	439.2	83.5	96.7	98.6	64.4	88.7	93.8	525.7
ALBEF-base [23]	4M	73.1	91.4	96.0	56.8	81.5	89.2	488.0	94.3	99.4	99.8	82.8	96.7	98.4	571.4
ALBEF-base [23]	3M	72.5	91.7	95.9	55.8	81.3	88.4	485.6	**95.1**	99.1	99.7	81.4	96.0	**98.2**	569.5
ALBEF-base+**MixGen**	3M	**74.2**	**92.8**	**96.4**	**57.3**	**82.1**	**89.0**	**491.8**$_{+6.2}$	94.8	**99.4**	**100.0**	**82.4**	**96.3**	98.0	**570.9**$_{+1.4}$

Table 2. **Zero-shot image-text retrieval** on Flickr30K dataset.

Method	#Images	Flickr30K(1K test set)						
		Text Retrieval			Image Retrieval			
		R@1	R@5	R@10	R@1	R@5	R@10	RSUM
UNITER-base [3]	4M	80.7	95.7	98.0	66.2	88.4	92.9	521.9
ViLT-base [19]	4M	73.2	93.6	96.5	55.0	82.5	89.8	490.3
ALBEF-base [23]	4M	90.5	98.8	99.7	76.8	93.7	96.7	556.2
ALBEF-base [23]	3M	91.1	98.2	99.3	75.7	92.5	96.0	552.8
ALBEF-base+**MixGen**	3M	**91.6**	**99.2**	**99.9**	**77.2**	**93.6**	**96.6**	**558.1**$_{+5.3}$

ing COCO [28], Visual Genome (VG) [20], SBU Captions [35] and Conceptual Captions (CC) [2]. Most literature [3, 48, 19, 23] used a combination of the four datasets as the standard pre-training setting, which leads to a total of 4M unique images and 5.1M image-text pairs. However, for SBU and CC datasets, image-text pairs are provided in url format, and a portion of the urls are unaccessible. For example, we can only download 2.2M out of 3M images of CC dataset. Hence, our final training set only has 3.3M unique images and 4.4M image-text pairs. Compared with previous literature that has access to the original 5.1M image-text pairs, we have about 700K less training samples. In this paper, we term the original setting as 4M, and ours as 3M.

Pre-training implementation details Majority of our experiments are performed based on ALBEF [23] given its superior performance on a range of downstream tasks. We adopt the official codebase[1] kindly provided by the authors. Following [23], the model is trained for 30 epochs with a total batch size of 512 on 8 V100 GPUs. When using MixGen, the default value of M is set to a quarter of the batch size B, i.e., a minibatch will contain 384 existing samples and 128 new image-text pairs. AdamW [30] is adopted as the optimizer with a weight decay of 0.02. Learning rate is warmed-up to $1e^{-4}$ in the first 1000 iterations and then decayed to $1e^{-5}$ following a cosine schedule. For ViLT and TCL, we use their official codebase[2] [3]. For CLIP, we adopt a reproducible implementation from open-clip[4]. To save memory and speed up experiments, we use mix precision

Table 3. **Compatibility to other vision-language pre-training methods**. Adding MixGen leads to consistent performance boost in terms of image-text retrieval. Note the models here are pre-trained on three datasets (COCO, VG and SBU) with 1M images. Ft: fine-tuned setting. Zs: zero-shot setting.

Methods	Venue	Flickr30K-Ft (1K test set)	MSCOCO-Ft (5K test set)	Flickr30K-Zs (1K test set)
ViLT [19]	ICML 21	232.2	172.5	135.5
ViLT + **MixGen**	-	**241.4**$_{+9.20}$	**189.7**$_{+17.2}$	**150.4**$_{+14.9}$
CLIP [38]	ICML 21	538.2	450.5	485.3
CLIP + **MixGen**	-	**541.0**$_{+2.80}$	**454.6**$_{+4.10}$	**492.1**$_{+6.80}$
ALBEF [23]	NeurIPS 21	555.7	470.7	518.0
ALBEF + **MixGen**	-	**561.0**$_{+5.30}$	**477.7**$_{+7.00}$	**524.3**$_{+6.30}$
TCL [60]	CVPR 22	561.5	487.0	537.3
TCL + **MixGen**	-	**563.6**$_{+2.10}$	**490.2**$_{+3.20}$	**539.5**$_{+2.20}$

training [34].

3.2. Image-text retrieval

Image-text retrieval includes two subtasks: (1) retrieve images with given text (Image Retrieval) and (2) retrieve text with given images (Text Retrieval). We conduct experiments on MSCOCO [28] and Flickr30K [36] datasets. In terms of evaluation, we use recall at K ($R@K$) metric, where $K = \{1, 5, 10\}$. For the purpose of easy comparison, we also use RSUM as the metric to reveal the overall performance of models following [52], which is defined as the sum of recall metrics at $K = \{1, 5, 10\}$ of both image and text retrieval tasks.

Fine-tuned results In Table 1, we can see that MixGen consistently improves over our ALBEF baseline on both datasets. Under the 3M setting, simply adding MixGen without any modifications leads to an improvement of 6.2% RSUM score on COCO and 1.4% RSUM score on Flicker30K. Note that, due to the missing data problem, our reproduced ALBEF (trained on 3.3M pairs) achieves marginally lower performance than the numbers reported in the original paper (trained on 4M pairs, gray row in Table 1). However, after adding MixGen, our performance is even better than the original paper on COCO and competitive on Flicker30K, despite our model is trained with 700K less image-text pairs. This clearly shows that MixGen improves the data efficiency for model training.

Zero-shot results In Table 2, similar conclusions can be ob-

[1] https://github.com/salesforce/ALBEF
[2] https://github.com/dandelin/ViLT
[3] https://github.com/uta-smile/TCL
[4] https://github.com/mlfoundations/open_clip

Table 4. **Comparison with state-of-the-art methods on downstream vision-language tasks**. MixGen consistently improve across VQA, VR and VE.

Method	#Images	VQA		NLVR^2		SNLI-VE	
		test-dev	test-std	dev	test-P	val	test
VisualBERT-base [24]	4M	70.80	71.00	67.40	67.00	-	-
OSCAR-base [27]	4M	73.16	73.44	78.07	78.36	-	-
UNITER-base [3]	4M	72.70	72.91	77.18	77.85	78.59	78.28
ViLT-base [19]	4M	71.26	-	75.70	76.13	-	-
UNIMO-base [26]	4M	73.79	74.02	-	-	80.00	79.10
VILLA-base [10]	4M	73.59	73.67	78.39	79.30	79.47	79.03
ALBEF-base [23]	4M	74.54	74.70	80.24	80.50	80.14	80.30
ALBEF-base [23]	3M	74.38	74.51	79.47	80.05	79.49	79.69
ALBEF-base+**MixGen**	3M	**74.51**$_{+0.13}$	**74.79**$_{+0.28}$	**80.23**$_{+0.76}$	**80.94**$_{+0.89}$	**80.05**$_{+0.56}$	**80.05**$_{+0.36}$

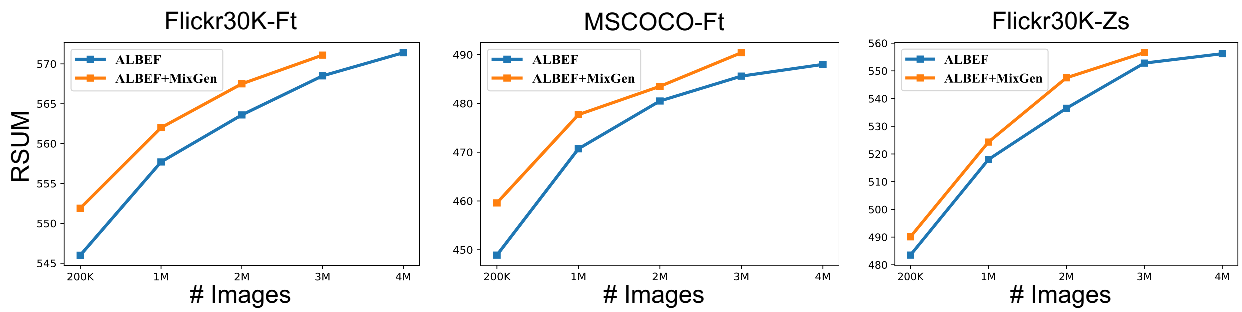


Figure 3. **Image-text retrieval performance given various number of training images**. Note that ALBEF with MixGen trained on 3M images achieves competitive performance or outperform baseline ALBEF trained on 4M images. This indicates the data efficiency of MixGen as an effective data augmentation method.

served. Under our 3M setting, MixGen leads to an improvement of 5.3% RSUM score on Flicker30K. Since zero-shot setting treats pre-trained models as feature extractors, such significant performance gains suggest that the multi-modal features learned with MixGen during pre-training generalize well.

Compatibility with other VL models We show MixGen's compatibility with other vision-language pre-training methods, i.e., CLIP [38], ViLT [19] and TCL [60]. Besides adding MixGen, we do not modify their original training settings. Given ViLT training is very costly (e.g., 3 days with 64 V100 GPUs), we only use three datasets (COCO, VG, and SBU) instead of four during pre-training for this experiment. The dataset consists of 1M unique images and 2.2M image-text pairs, which we term it the 1M setting. As shown in Table 3, simply adding MixGen on top of these strong baselines consistently improve state-of-the-art performance. In terms of fine-tuned image-text retrieval on COCO, MixGen demonstrates significant accuracy boost (absolute): ViLT (+17.2%), CLIP (+4.1%), ALBEF (+7.0%) and TCL (+3.2%). This shows the versatility of MixGen as image-text data augmentation in pre-training.

Data efficiency We investigate how much data efficiency MixGen can achieve. We reduce the number of unique images used for pre-training, from 3M to 2M, 1M and 200K. We have described 1M and 3M settings. For 2M, we use three datasets plus a random subset from CC dataset. For 200K, we simply use two datasets (COCO and VG). The performance of image-text retrieval can be seen in Figure 3. We first notice that adding MixGen is always better than without it. In particular, the improvement in low data regime is more significant. Secondly, the performance of ALBEF with MixGen when trained on 1M, 2M and 3M samples match baseline ALBEF when trained on 2M, 3M and 4M samples, respectively. This again indicates the data efficiency of MixGen.

Table 5. **Weakly-supervised visual grounding** on RefCOCO+

Method	#Images	Val	TestA	TestB
ARN [29]	14M	32.78	34.35	32.13
CCL [67]	14M	34.29	36.91	33.56
ALBEF$_{itc}$ [23]	14M	51.58	60.09	40.19
ALBEF$_{itm}$ [23]	14M	58.46	65.89	46.25
ALBEF$_{itm}$ [23]	3M	57.76	65.08	45.57
ALBEF$_{itm}$+**MixGen**	3M	56.72	**65.35**$_{+0.27}$	**46.47**$_{+0.9}$

3.3. Evaluation on VQA, VR, VE and VG

We first describe the other four downstream tasks. Visual Question Answering (VQA) aims to predict an answer given an image and a corresponding question. We conduct experiments on the VQA 2.0 dataset [11] and consider it as

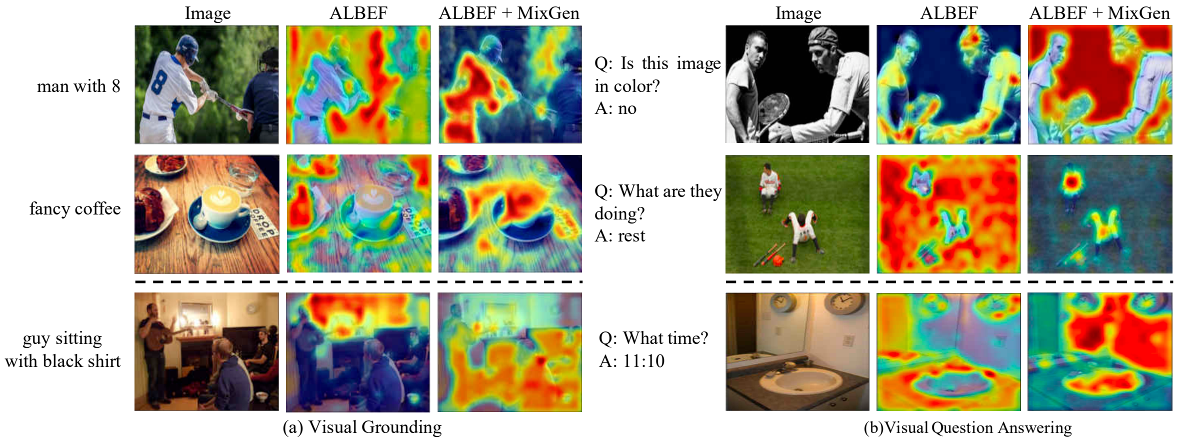


Figure 4. **Grad-CAM visualization for visual grounding and VQA**. First two rows are successful cases where MixGen helps, and the third row shows failure cases. Figure best viewed in color.

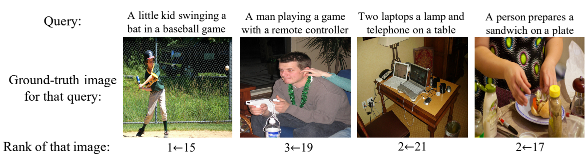


Figure 5. **Text-to-image retrieval on MSCOCO**. Given a text query, $A \leftarrow B$ indicates the rank of the retrieved ground-truth image improves from B (wo MixGen) to A (w MixGen).

an answer generation task following [23, 4]. We constrain the fine-tuned answer decoder to generate answers from the set of $3,192$ candidates for fair comparison to other literature. We use the official evaluation server[5] to report accuracy. Visual Entailment (VE) is a visual reasoning task to predict whether the relationship between an image and text is entailment, neutral or contradictory. Following [3, 23], we treat this task as a three-way classification problem and report classification accuracy on the SNLI-VE [53] dataset. Visual Reasoning (VR) requires the model to determine whether a textual statement describes a pair of images. We conduct experiments on the NLVR^2 [43] dataset following [19, 23, 27] and report standard per-example prediction accuracy. Visual Grounding (VG) localizes an image region that corresponds to a text description. Here, we follow [23, 33] to evaluate on the RefCOCO+ dataset [62] in a weakly-supervised setting.

Table 4 reports the performance comparisons of different vision-language pre-training baselines on downstream VQA, VR and VE tasks. Similar to image-text retrieval task, MixGen consistently boosts performance on the three tasks. Under the 3M setting, ALBEF with MixGen outperforms its corresponding baseline by absolute 0.28% on VQA test-std, 0.89% on NLVR^2 test-P, and 0.36% on SNLI-VE test. Note that ALBEF with MixGen under 3M setting even outperforms original ALBEF trained with 4M images on VQA and NLVR^2.

Table 5 reports the performance of visual grounding on RefCOCO+ dataset. ALBEF$_{itc}$ and ALBEF$_{itm}$ are two variants that compute Grad-CAM visualization for grounding. Since ALBEF$_{itm}$ performs better, we adopt ALBEF$_{itm}$ as our baseline and add MixGen on top of it. We can see that our model with MixGen trained under 3M setting not only surpasses its corresponding baseline on both test sets, but also outperforms ALBEF$_{itm}$ trained with 14M dataset on TestB. All empirical results above suggest the superior data efficiency brought by MixGen.

3.4. Visualizations of MixGen

Image-text retrieval In Figure 5, we show visualization of text-to-image retrieval on MSCOCO. To be specific, given a text query, we want to compare the rank of the retrieved ground-truth image out of all the retrieved images between ALBEF with and without MixGen. We can see that MixGen is usually able to locate the matching image in top-3 retrievals, performing significantly better than the baseline ALBEF.

Visual grounding and VQA In Figure 4, we show Grad-CAM visualizations to help us understand why MixGen is benficial. For the visual grounding task on the RefCOCO+ dataset, we can see that model trained with MixGen can locate image regions more precisely according to the text query. Even for the failure case, model trained on MixGen is able to attend better on “sitting men”, instead of focusing on the wall (wo MixGen model). For VQA task on VQA2.0 dataset, MixGen can attend to important cues that lead to the correct answer (e.g., black background for predicting if this is a black-white image). For the failure case, telling time might be too challenging, where both ALBEF and ALBEF + MixGen fail.

[5] https://visualqa.org/evaluation.html

Table 6. **MixGen variants.** For (d) and (e), $|T|_\lambda$ indicates randomly keeping $\lambda \cdot |T|$ tokens in text sequence T, where $|T|$ is the number of tokens in this sequence. See text for more details.

Variants	Image	Text	λ
MixGen	$I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$	$T_k = \text{concat}(T_i, T_j)$	$\lambda = 0.5$
(a)	$I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$	$T_k = \text{concat}(T_i, T_j)$	$\lambda \sim$ Beta(0.1, 0.1)
(b)	$I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$	$T_k = T_i$ or T_j	$\lambda = 0.5$
(c)	$I_k = I_i$ or I_j	$T_k = \text{concat}(T_i, T_j)$	-
(d)	$I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$	$T_k = \text{concat}(\|T_i\|_\lambda, \|T_j\|_{1-\lambda})$	$\lambda \sim$ Beta(0.1, 0.1)
(e)	$I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$	$T_k = \|\text{concat}(T_i, T_j)\|_{0.5}$	$\lambda \sim$ Beta(0.1, 0.1)

Table 7. **Ablation of MixGen design**. For image-text retrieval task, we report the RSUM metric. Ft: fine-tuned setting. Zs: zero-shot setting.

MixGen variants	Flickr30K-Ft (1K test set)	MSCOCO-Ft (5K test set)	Flickr30K-Zs (1K test set)	SNLI-VE (test)	NLVR^2 (test-P)	VQA (test-dev)
ALBEF-base	555.7	470.7	518.0	78.91	78.09	73.62
MixGen	**561.0**	**477.7**	**524.3**	**79.65**	**79.42**	73.84
(a)	553.2	467.4	512.9	78.68	77.22	73.15
(b)	557.2	470.0	516.2	78.78	78.23	73.76
(c)	555.4	472.3	518.2	78.87	78.60	73.34
(d)	555.2	463.7	506.8	79.07	78.30	73.29
(e)	559.2	477.1	523.7	79.36	78.33	**73.95**

4. Ablation Studies

In this section, we perform various ablation studies to support the design choice of MixGen. Unless otherwise stated, we adopt ALBEF as baseline and use the 1M setting which consists of three datasets (COCO, VG, and SBU) in model pre-training.

Design variants In Sec. 2.1, we introduce 5 variants to support the design choice of MixGen. Detailed comparisons among variants can be found in Table 6. Here we use the same training setting for all 5 variants and evaluate them across five downstream datasets. As we can see in Table 7, our default MixGen achieves the overall best performance, and consistently outperform other variants on different tasks except VQA.

Input-level vs embedding-level MixGen Recall in Sec. 2.2 that MixGen can be either performed on the raw input as input-level MixGen, or performed on the embedding level as embedding-level MixGen. Here, we report the overall RSUM scores on COCO and Flicker30K datasets. As we can see in Figure 6 left, both input-level and embedding-level MixGen perform better than the baseline. In addition, input-level MixGen performs consistently better than embedding-level MixGen.

Optimal ratio of MixGen samples In our pseudocode, we replace the first M training samples with the newly generated pairs by MixGen, so that the total batch size and training iterations stay the same. Motivated by mixup [65], we could randomly shuffle existing pairs and replace them all by new samples, i.e., $M = B$. Here, we perform an ablation study to investigate how many new pairs is optimal. As we can see in Figure 6 right, given a fixed batch size $B = 512$ while varying M, $M = 128$ (i.e., $M = B/4$) leads to the best performance.

5. Discussion

While we have shown MixGen's effectiveness across multiple settings, there are many more factors that we do not consider in this paper. First, there are many popular unimodal data augmentation techniques we could compare to [18, 47]. However, due to its unlimited combination, we can only choose the most related configurations for evaluation, and leave the rest comparisons for future work. Second, due to the computational limits, we only perform pre-training on the most widely adopted 4M set and compare to other work on this setting for fair comaprison. However, there are larger-scale datasets becoming public available recently that we could explore, such as LAION-400M [41]. Third, despite we have evaluated MixGen on four models and five vision-language tasks, there are more models and tasks in vision-language domain. For example, several recent work [22, 61, 50] have adopted encoder-decoder architecture and reported performance on generation tasks like image captioning. There are also work in video-text retrieval [32], text-to-image generation [39], language guided object detection [25] or segmentation [57]. It would be interesting to investigate the generalization capability of MixGen for these methods and tasks.

Methods	Flickr30K-Ft (1K test set)	MSCOCO-Ft (5K test set)
Baseline	555.7	470.7
+ Embedding-level	558.9$_{+3.2}$	471.0$_{+0.3}$
+ Input-level	**561.0**$_{+5.3}$	**477.7**$_{+7.0}$

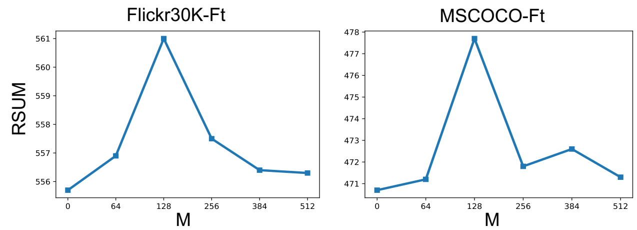


Figure 6. **Left:** Input-level vs embedding-level MixGen. Input-level performs consistently better than embedding-level MixGen. **Right:** Optimal ratio of MixGen samples. Given batch size B=512, we find M=128 (i.e., M=$B/4$) leads to best performance.

6. Related Work

Data augmentation in CV and NLP Data augmentation serves an integral part to the success of training most deep networks in CV and NLP, especially when the training set is small. For CV, the techniques evolve with the model development in the past decade, starting from basic ones (random translation, horizontal flipping, random rotation, color jittering, random resized crop, etc.), to more advanced ones (Cutout [9], mixup [65], CutMix [64], random erasing [68], etc.). With the rise of AutoML, researchers start to search data augmentation policies automatically from the data, such as AutoAugment [6] and RandAugment [7]. For NLP, besides paraphrasing-based methods like back-translation [54], tokens in text can be replaced by applying thesaurus derived from WordNet [66], word and framework embeddings [49], and masked language models [16] without changing the meaning of the original text. Swapping, removal, insertion, substitution are common random operators to add noise to text at both the word level [51, 5, 40] and the sentence level [59]. All in all, data augmentation leads to improved performance, higher data efficiency, and robustness to domain shift and adversarial attack, which is worthy of investigating in the multi-modal domain.

Vision-language pre-training Despite the rapid progress of data augmentation in each uni-modal domain (CV, NLP, speech, etc.), joint multi-modal data augmentation is rarely studied. Most of the focus from the community has been on modeling improvements [42, 38, 19, 4, 44, 17, 56], multi-modal fusion techniques [31, 48, 1, 23, 10] and multi-modal loss functions [38, 19, 60, 55, 69]. Existing works often perform mild data augmentation on either vision or text modality, but not jointly. There are a few related work on joint augmentation. [45] has proposed to generate semantically equivalent adversarial examples of both visual and textual data as augmented training samples for VQA. [58] learns a cross-modality matching network to select image–text pairs from existing uni-modal datasets as the multi-modal synthetic dataset, and uses this dataset to enhance the performance of classifiers for sentiment analysis. Both approaches are effective, but only designed for specific vision-language downstream tasks. On the contrary, our proposed MixGen is simple, effective, and compatible with generic vision-language pre-training methods.

Relation to mixup Mixup was originally introduced as a data-agnostic augmentation technique for CV [65]. It is a widely adopted (even by-default) technique for training CNNs or vision transformers. Specifically, mixup generates new training samples via

$$\begin{aligned} x_k &= \lambda \cdot x_i + (1-\lambda) \cdot x_j \\ y_k &= \lambda \cdot y_i + (1-\lambda) \cdot y_j, \end{aligned} \tag{3}$$

where x_i and x_j are raw input vectors, y_i and y_j are one-hot label encodings, and $\lambda \in [0, 1]$ is sampled from a Beta distribution. Recently, [12] adapted mixup to the NLP domain by replacing raw image pixels x with text embeddings.

MixGen differs from mixup in several aspects. First, MixGen is proposed for multi-modal data augmentation. The input to MixGen is data from different modalities, and there is no one-hot label encodings y involved in Equation (1). Second, MixGen does not perform mixup on the text modality. Instead, MixGen simply concatenates text sequences to perform image-text co-augmentation and best preserves information.

7. Conclusion

In this work, we present a new vision-language joint data augmentation method termed MixGen. Adding MixGen on four recent state-of-the-art models achieves consistent improvement across five different downstream tasks. Strong empirical results suggest that MixGen not only makes these models learn better multi-modal latent representations, but also improves their data efficiency. We hope that this work will provide useful data points for future research on joint multi-modal data augmentation.

References

[1] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. DocFormer: End-to-End Transformer for Document Understanding. In *ICCV*, 2021.

[2] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *CVPR*, 2021.

[3] 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. In *ECCV*, 2020.

[4] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying Vision-and-Language Tasks via Text Generation. In *ICML*, 2021.

[5] Claude Coulombe. Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs. *arXiv preprint arXiv:1812.04718*, 2018.

[6] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Strategies From Data. In *CVPR*, 2019.

[7] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In *NeurIPS*, 2020.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.

[9] Terrance Devries and Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[10] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-Scale Adversarial Training for Vision-and-Language Representation Learning. In *NeurIPS*, 2020.

[11] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *CVPR*, 2017.

[12] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting Data with Mixup for Sentence Classification: An Empirical Study. *arXiv preprint arXiv:1905.08941*, 2019.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[14] Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing Out of the Box: End-to-End Pre-Training for Vision-Language Representation Learning. In *CVPR*, 2021.

[15] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *ICML*, 2021.

[16] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for Natural Language Understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[17] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. MDETR - Modulated Detection for End-to-End Multi-Modal Understanding. In *ICCV*, 2021.

[18] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle Mix: Exploiting Saliency and Local Statistics for Optimal Mixup. In *ICML*, 2020.

[19] Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. In *ICML*, 2021.

[20] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV*, 2017.

[21] Chenge Li, István Fehérvári, Xiaonan Zhao, Ives Macedo, and Srikar Appalaraju. SeeTek: Very Large-Scale Open-Set Logo Recognition With Text-Aware Metric Learning. In *WACV*, 2022.

[22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML*, 2022.

[23] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. In *NeurIPS*, 2021.

[24] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv preprint arXiv:1908.03557*, 2019.

[25] Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded Language-Image Pre-training. In *CVPR*, 2022.

[26] Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. UNIMO: Towards Unified-Modal Understanding and Generation via Cross-Modal Contrastive Learning. In *ACL*, 2021.

[27] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *ECCV*, 2020.

[28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[29] Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Dechao Meng, and Qingming Huang. Adaptive Reconstruction Network for Weakly Supervised Referring Expression Grounding. In *ICCV*, 2019.

[30] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019.

[31] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*, 2019.

[32] Huaishao 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. *arXiv preprint arXiv:2104.08860*, 2021.

[33] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and Comprehension of Unambiguous Object Descriptions. In *CVPR*, 2016.

[34] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed Precision Training. In *ICLR*, 2018.

[35] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *NeurIPS*, 2011.

[36] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In *ICCV*, 2015.

[37] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. ImageBERT: Cross-modal Pre-training with Large-scale Weak-supervised Image-Text Data. *arXiv preprint arXiv:2001.07966*, 2020.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021.

[39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv preprint arXiv:2204.06125*, 2022.

[40] Mehdi Regina, Maxime Meyer, and Sébastien Goutal. Text Data Augmentation: Towards Better Detection of Spear-phishing Emails. *arXiv preprint arXiv:2007.02033*, 2020.

[41] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[42] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *ICLR*, 2020.

[43] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A Corpus for Reasoning about Natural Language Grounded in Photographs. In *ACL*, 2019.

[44] Hao Tan and Mohit Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP*, 2019.

[45] Ruixue Tang, Chao Ma, Wei Emma Zhang, Qi Wu, and Xiaokang Yang. Semantic Equivalent Adversarial Data Augmentation for Visual Question Answering. In *ECCV*, 2020.

[46] Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris Metaxas. SelfNorm and CrossNorm for Out-of-Distribution Robustness. In *ICCV*, 2021.

[47] A. F. M. Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization. In *ICLR*, 2021.

[48] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. VLMo: Unified Vision-Language Pre-Training with Mixture-of-Modality-Experts. *arXiv preprint arXiv:2111.02358*, 2021.

[49] 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 *EMNLP*, 2015.

[50] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple Visual Language Model Pretraining with Weak Supervision. In *ICLR*, 2022.

[51] Jason Wei and Kai Zou. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[52] Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified Visual-Semantic Embeddings: Bridging Vision and Language With Structured Meaning Representations. In *CVPR*, 2019.

[53] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual Entailment: A Novel Task for Fine-Grained Image Understanding. *arXiv preprint arXiv:1901.06706*, 2019.

[54] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised Data Augmentation for Consistency Training. In *NeurIPS*, 2020.

[55] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A Fast Proximal Point Method for Computing Exact Wasserstein Distance. In *UAI*, 2020.

[56] Haiyang Xu, Ming Yan, Chenliang Li, Bin Bi, Songfang Huang, Wenming Xiao, and Fei Huang. E2E-VLP: End-to-End Vision-Language Pre-training Enhanced by Visual Learning. In *ACL*, 2021.

[57] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. GroupViT: Semantic Segmentation Emerges from Text Supervision. In *CVPR*, 2022.

[58] Nan Xu, Wenji Mao, Penghui Wei, and Daniel Zeng. MDA: Multimodal Data Augmentation Framework for Boosting Performance on Sentiment/Emotion Classification Tasks. *IEEE Intelligent Systems*, 2021.

[59] Ge Yan, Yu Li, Shu Zhang, and Zhenyu Chen. Data Augmentation for Deep Learning of Judgment Documents. In *IScIDE*, 2019.

[60] Jinyu Yang, Jiali Duan, S. Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul M. Chilimbi, and Junzhou Huang. Vision-Language Pre-Training with Triple Contrastive Learning. In *CVPR*, 2022.

[61] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive Captioners are Image-Text Foundation Models. *arXiv preprint arXiv:2205.01917*, 2022.

[62] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling Context in Referring Expressions. In *ECCV*, 2016.

[63] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A New Foundation Model for Computer Vision. *arXiv preprint arXiv:2111.11432*, 2021.

[64] 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 *ICCV*, 2019.

[65] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *ICLR*, 2018.

[66] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. In *NeurIPS*, 2015.

[67] Zhu Zhang, Zhou Zhao, Zhijie Lin, Jieming Zhu, and Xiuqiang He. Counterfactual Contrastive Learning for Weakly-Supervised Vision-Language Grounding. In *NeurIPS*, 2020.

[68] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random Erasing Data Augmentation. In *AAAI*, 2020.

[69] Mohammadreza Zolfaghari, Yi Zhu, Peter Gehler, and Thomas Brox. CrossCLR: Cross-modal Contrastive Learning For Multi-modal Video Representations. In *ICCV*, 2021.