A. Appendix

There are three main sections in this supplementary material. In Section A.1 we list the training details for downstream tasks. Section A.2 shows the specific image-text statistics of our pre-training corpus. As we use different data augmentations for images and texts during pre-training, we detail them in Section A.3. We give more results comparison on video-text matching in Section A.4. In Section A.5 we give more examples of the proposed knowledge sharing and illustrations of the effectiveness of weight sharing transformer encoder.

A.1. Training Details for Downstream Tasks

The details for downstream multi-modal matching datasets are listed in Table 1. For ITM and VTM tasks, the number of images doesn't equal that of texts, thus I:T refers to the number of captions describing a picture. For TM, the evaluation is an unsupervised process, thus we directly use the test sets of STS12-16 and STSB. For IR, models are trained with the train sets and use the test sets as the queries to retrieve images in the retrieval sets.

Image-Text Matching We finetune COOKIE on MSCOCO and Flickr30K for 20 and 16 epochs, respectively. The initial learning rate is 2e-5 or 1e-5 and decays by 10 times after half of the total epochs. We use AdamW optimizer with a weight decay factor 1e-4 and a 0.1 warm-up proportion. For ResNeXt-101-based models, the batch size for MSCOCO is set to 384 and is 288 for Flickr30K. The batch size is set to 320 or 240 when the visual backbone is substituted with ResNet101. The definition of the hinged hard triplet loss is defined below.

$$\begin{aligned} L_{imt} =&[\alpha + S(\vec{I}^{'}, \vec{T}) - S(\vec{I}, \vec{T})]_{+} + \\ &[\alpha + S(\vec{I}, \vec{T}^{'}) - S(\vec{I}, \vec{T})]_{+}, \end{aligned} \tag{1}$$

where $S(\cdot)$ refers to the similarity function which is cosine similarity in our model. Here $[x]_+ \equiv max(x, 0)$ and α is the margin, which is set to 0.2. We use $MAX\text{-}Pooling$ for image-text matching tasks.

Video-Text Matching We finetune our pre-trained model on MSRVTT benchmark for 10 epochs. Similarly, the learning rate is 2e-5 and decays by 10 times after 5 epochs. We use a batch size of 320. We use the standard split with 6573 videos for training, 2990 for testing, and 497 for validation. We use $MEAN\text{-}Pooling$ or $G\text{-}Pooling$ for video-text matching tasks.

Text Matching We directly use the sentence embeddings for evaluating on semantic text similarity task, thus no training process is required. We use $MAX\text{-}Pooling$ for text matching tasks. Pearson and Spearman coefficients are two widely used metrics to evaluate the correlation between the predicted similarity scores and the labels. Thus we use the mean value of them for evaluation.

Task	Dataset	Train	Test	Retrieval	I:T	Class
ITM	MSCOCO	113,287	5,000	-	1:5	-
ITM	Flickr30K	29,000	1,000	-	1:5	-
VTM	MSRVTT	6,753	2,990	-	1:20	-
TM	STSB	5,749	1,379	-	-	-
IR	MSCOCO	10,000	5,000	112,217	-	80
IR	NUSWIDE	10,000	2,040	149,685	-	21

(a)

Dataset	STS12	STS13	STS14	STS15	STS16
Test	3,108	1,500	3,750	3000	1186

(b) For TM, STS12-16 only contain test set.

Table 1: Experimental settings for all downstream datasets. ITM: image-text matching, VTM: video-text matching, TM: text matching, IR: image retrieval. For ITM and IR, we list num of images. For VTM, we list num of videos. For TM, we list num of text pairs.

Dataset	Image	Text
CC(train)	2.8M	2.8M
SBU(all)	0.8M	0.8M
MSCOCO(train)	113k	566k
Flickr30K(train)	29k	145k
VQA2.0(train)	83k	444k
GQA(bal-train)	79k	1.0M
Total	3.9M	5.9M

Table 2: Statistics of the pre-training corpus.

Image Retrieval For image retrieval, the number of output bits is either 16, 32, or 64. The learning rate is set to 1e-4 initially and decays with a ratio of 0.7 every 10 epoch. We finetune COOKIE for a total of 100 epochs with a batch size of 320. We use $MEAN\text{-}Pooling$ for image retrieval tasks. As IR is to judge whether the retrieval is correct according to the category, we record MAP@5000, which is a common metric for image retrieval.

A.2. Pre-training Corpus

We use a total of 5.9M image-text pairs to pre-train our COOKIE. The details of these datasets are shown in Table 2. It is noticed that due to broken URLs, for conceptual captions, we only collected 2.8M of the 3M image-text pairs. And for SBU captions, we only collected 0.8M of the 1M pairs. For the top-4 datasets, images are paired with captions describing them, which are collected from social networks. While for VQA2.0 and GQA, the images are paired

Methods	Video-to-Text			Text-to-Video			
	R@1	R@5	R@10	R@1	R@5	R@10	Rsum
MEE^{*}	13.4	32.0	44.0	6.8	20.7	31.1	148.0
CE^{**}	15.6	40.9	55.2	10.0	29.0	41.2	191.9
$W2VV^{*}_{++}$	17.5	40.2	52.5	11.1	29.6	40.5	191.4
$DualEncoding^{*}$	**22.5**	**47.1**	**58.9**	**11.6**	**30.3**	**41.3**	**211.7**
COOKIE(gpo)	20.0	42.0	54.9	9.8	28.3	39.6	194.6

Table 3: Results on video-text matching task with MSRVTT dataset. Methods with $*$ use stronger ResNeXt-ResNet visual features and methods with $**$ use seven-modal features.

with questions. These questions are based on the image itself, but they may not accurately describe the image.

A.3. Data Augmentations for Pre-training

We use data augmentations for images in visual contrastive learning and texts in textual contrastive learning. details are followed.

Visual Augmentations For visual contrastive learning, we design five augmentations before resizing. All operations are carried out step by step.

- Cropping. Every image is cropped into the size of $(\sigma_1 * H, \sigma_2 * W)$, where H, W are the height and width of original image and σ_1 and σ_2 are two random numbers ranged 0.6-1.
- Flipping. Images flip horizontally 50% of the time.
- Gaussian Blur. With a probability value of 0.5, we blur an image by a Gaussian function.
- Color jitter. Color jitter is performed with a probability of 0.8. It includes jitter of brightness, contrast, saturation, and hue.
- Color dropping. We convert RGB images to grayscale images 20% of the time.

Textual Augmentations Visual augmentation plays a key role in visual representation learning, however, for texts, classic textual augmentations will lead to information losing thus changing the meaning of the whole sentence. Wu et al. proved the effectiveness of textual augmentations in textual contrastive learning. Specifically in our method, each token in the sentence has a 20% probability of being processed. If a token is to be processed, the procedure can be illustrated by

- 50% of the time: Replace the word with the $[MASK]$ token.
- 10% of the time: Replace the word with a random word chosen from the vocabulary.
- 40% of the time: Delete the word directly.

A.4. More Comparison for Video-Text Matching

As seen in Table 3, more comparison for video-text matching on MSRVTT are given. It is noticed that three methods with $*$ use stronger ResNeXt-ResNet visual features and $CE**$ uses seven-modal features comparing to our official ResNet152 features. For fair comparison, we didn't list them in original Table 3 in the main text. Even without those stronger features, our COOKIE still outperforms most of them.

A.5. More Illustrations

In the main body of the paper, we illustrate the concept of sharing textual knowledge with images as well as the effectiveness of the weight-sharing transformer encoder (Figure 1 and Figure 3 in the main body). Here, we firstly give out the illustration of sharing visual knowledge with texts. Secondly, another enlarged figure explaining how the WS-TE works is shown.

Illustration of Knowledge Sharing As seen in Fig. 1, although the two sentences have many words in common like "black" and "ocean", they possess different semantic meanings. As a result, the single-modal methods Sentence-BERT predicts a similarity value of 3.95 while the label is only 0.6. It's hard to judge with mere texts. With the information given by images, we can figure out that (a) places emphasis on "dog" and "rock" while (b) pays more attention to "wave" and "ocean". With the help of cross-modal contrastive learning, the similarity score predicted by COOKIE descends to 2.42, which is less than half of the total score 5.0.

Illustration of Weight-Sharing Transformer We give out another visualization of the effectiveness of the weight-sharing transformer encoder(WS-TE) in Fig. 2. Without weight sharing, the transformer encoders on top of each path make the image and text focus on different semantics. With weight sharing, tokens with similar semantic meanings are given similar attention values, thus the two modalities pay similar attention to the cook and two people in the background, which are easy to be ignored.

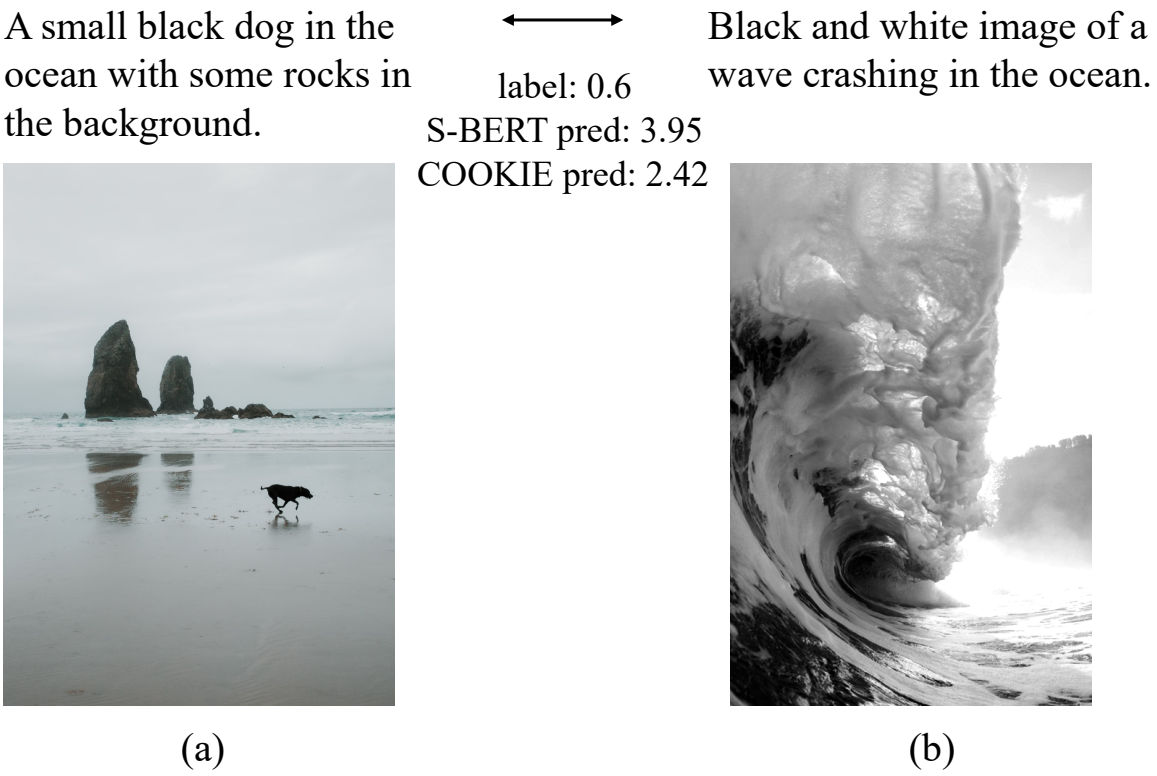

Figure 1: Example of knowledge sharing. Though the two sentences have some words in common like “black” and “ocean”, they possess different semantic meanings. It’s hard measure this dissimilarity with mere texts. However, if you judge with the pictures, it’s easy to see that they are describing different scenes. The similarity score ranges from 0 to 5.

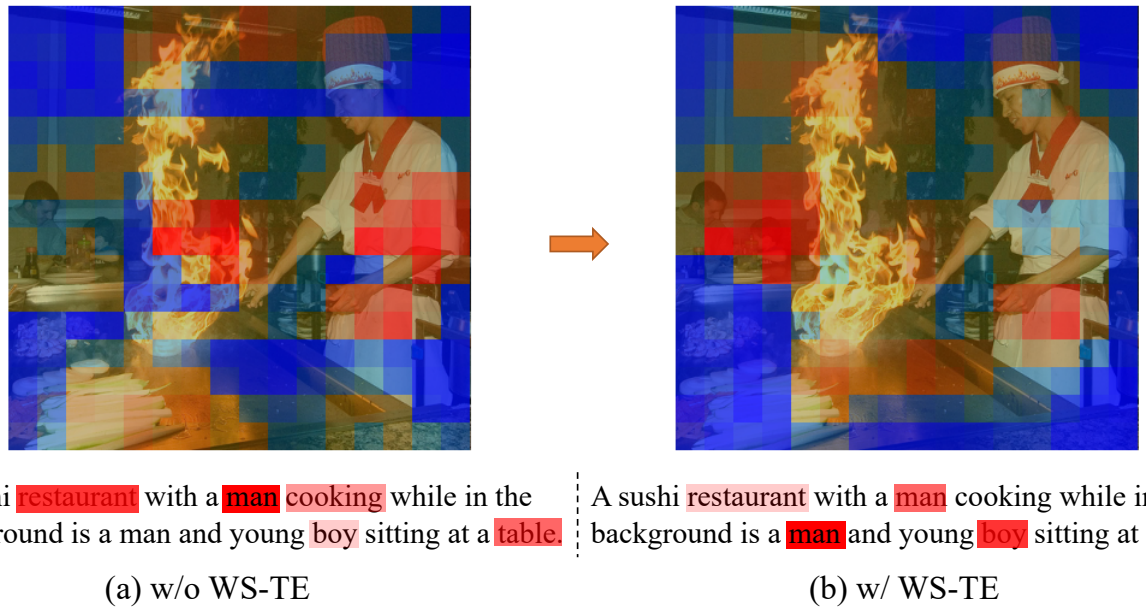

Figure 2: With the WS-TE, images and texts concentrate on the same semantics. To align the image and text, the cook and two people behind him should be paid more attention rather than the fire or the table.