


Multimodality Self-distillation for Fast Inference of Vision and Language Pretrained Models

Jun Kong , Jin Wang , Liang-Chih Yu , and Xuejie Zhang 

Abstract—The computational cost of the vision and language pretrained models (VL-PTMs) limits their deployment in resource-constrained devices that require low latency. One existing solution is to apply the early exiting (EE) strategy to accelerate the inference. This technique can force model prediction using only a few former transformer layers. However, these former layers behave differently with the final classifier, inevitably resulting in performance decline. To counter such limitation, self-distillation has been commonly introduced to enhance the representation abilities of the EE classifiers. This results in a semantic gap since EE classifiers are directly trained to mimic the outputs of the final classifier without access to the modality-specific behaviors. This study proposes a multimodality self-distillation method for the fast inference of VL-PTMs. To fill the semantic gap between modalities, we split the multimodalities into separate modalities and added them as extra inputs to encourage the effective distillation of each modality. Furthermore, the mean squared error (MSE) is introduced to minimize the distance of feature maps and further enhance the representation ability of the EE classifiers. Experiments show that the proposed method outperforms the previous EE strategies with the same inference time, and performs competitively even if the model exited very early.

Index Terms—Accelerating inference, early exiting, multimodality self-distillation, vision and language pretrained models.

I. INTRODUCTION

TRANSFORMER architecture applications are increasingly being used for various multimodal tasks, including visual question answering (VQA) [1], visual entailment (VE) [2] and natural language for visual reasoning (NLVR2) [3]. This success is attributed to the shared underlying textual and visual properties associated with texts with visual concepts. Vision and language pretrained models (VL-PTMs), such as ViLBERT [4], VL-BERT [5], Unicoder-VL [6] and UNITER [7], can be fine-tuned to improve the performance of downstream multimodal tasks. However, the resulting exponential growth of the

parameters may severely limit the deployment and flexibility of these models for real-time applications on resource-constrained platforms, such as drones, self-driving cars, and wearable devices.

Recent studies have suggested compressing the parameters in pretrained models (PTMs) [8], [9], [10] to reduce the computational cost and accelerate the inference. Existing methods include knowledge distillation (KD) [11], [12], [13], pruning [14], [15], and quantization [16]. Knowledge distillation (KD) [17], [18] refers to the use of the predictive distributions of a powerful teacher model as soft targets to guide the training of a smaller student model, such that the student model becomes an equally effective model with a tolerable performance sacrifice. Similarly, pruning [19] removes unnecessary parts of PTMs after training, whereas quantization [20] truncates floating point numbers such that only a few bits are used, thus accelerating the computation. These techniques permanently discard parts of PTMs, leading to an inevitable decline in performance. Moreover, once the models are redesigned and retrained, their parameters and computations are fixed, making it impossible to migrate to other platforms.

An alternative approach to accelerate model inference for PTMs is the early exiting (EE) strategy [21], [22], used in applications such as DeeBERT [23] and PABEE [24]. Specifically, extra classifiers (that is, *off-ramps* for EE) are inserted between each two transformer layers of the PTMs. After an input goes through a transformer layer, the EE classifier determines whether the prediction is sufficiently robust to achieve adequate performance as the final classifier. Once the *off-ramp* is sufficiently confident, the result is returned; otherwise, the sample is passed to the next layer, and the calculation is repeated.

Although EE classifiers in the former layers are already sufficiently confident, previous studies have shown that different levels of features can be learned in different transformer layers [25]. For example, surface features are typically learned in former layers, syntactic features in middle layers, and semantic features in deeper layers. This leads to the contradiction where the earlier the model exits, the fewer semantic features can be learned. In addition, EE classifiers that share the same parameters are not applicable for input features with different semantic levels. Since the EE classifiers tend to capture fewer features, and primarily those at the surface level, they may behave differently from the final classifier that can learn a greater number of and more semantic features. As shown in Fig. 1(a), the logits of the EE classifiers in the former layers are quite different from

Manuscript received 24 November 2022; revised 9 October 2023 and 18 March 2024; accepted 25 March 2024. Date of publication 2 April 2024; date of current version 21 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61966038 and Grant 62266051, and in part by the Ministry of Science and Technology, Taiwan, under Grant MOST111-2628-E-155-001-MY2. The Associate Editor coordinating the review of this manuscript and approving it for publication was Mrs. Si Liu. (Corresponding authors: Jin Wang; Liang-Chih Yu.)

Jun Kong, Jin Wang, and Xuejie Zhang are with the School of Information Science and Engineering, Yunnan University, Kunming 650000, China (e-mail: kongjun@mail.ynu.edu.cn; wangjin@ynu.edu.cn; xjzhang@ynu.edu.cn).

Liang-Chih Yu is with the Department of Information Management, Yuan Ze University, Taoyuan 32003, Taiwan (e-mail: lcyu@saturn.yzu.edu.tw).

The code for this paper is available at: <https://github.com/JunKong5/UNITER-MSD>.

Digital Object Identifier 10.1109/TMM.2024.3384060

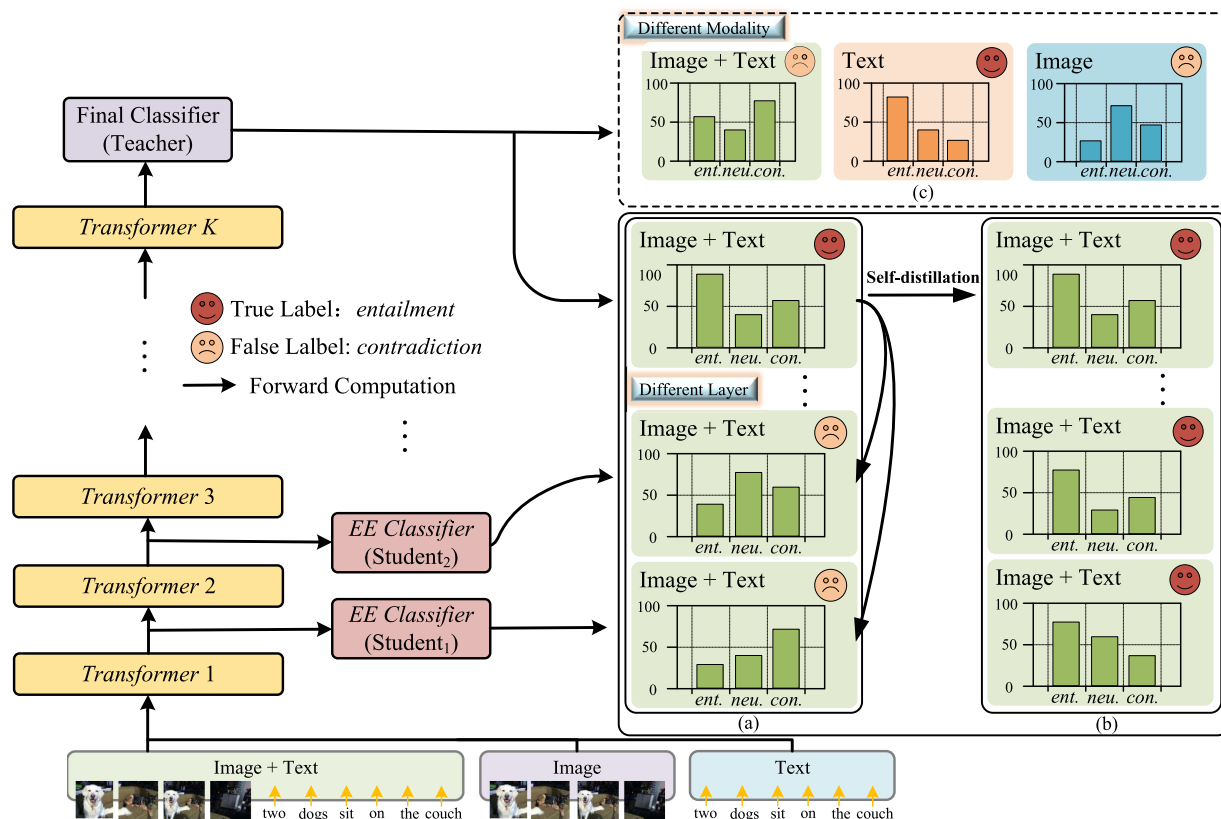


Fig. 1. Conceptual diagram of early exiting strategy for vision and language pretrained model.

those of the final classifier. In this circumstance, applying the EE strategy may lead to performance decline.

To ensure that the former layers can approximate the higher layers in terms of representation ability, FastBERT [26] adopts a self-distillation strategy to improve the stability of the EE classifier, where the final classifier was applied as a teacher to transfer the representation abilities to the EE classifiers as students. As shown in Fig. 1(b), once the self-distillation strategy is applied, the logits of the EE classifiers are improved, and are similar to those of the final classifier. However, multimodal inputs contain different modalities, such as text and images, presenting different types of knowledge. The amount of information in each modality is likely to differ [27]. As a result, different modalities exert different degrees of influence on different tasks. For example, the language modality dominates on VQA, while the visual modality dominates on SNLI-VE.

While self-distillation can be applied to multimodal tasks, EE classifiers are trained with the joint modality without considering individual modalities, which can also provide useful knowledge for prediction. As the example shown in Fig. 1(c), the individual modality (Text) provides more practical knowledge than the joint modality (Image+Text). In this example, the EE classifiers trained with self-distillation can only mimic the outputs of the final classifier with the joint modality, thus producing incorrect predictions.

To address this issue, this study proposes a multimodality self-distillation (MSD) method for fast inference of VL-PTMs. The proposed method improves existing EE strategies by

splitting the multimodalities into separate modalities and adding them as extra inputs to encourage effective distillation from each modality. Additionally, instead of exiting from the former layers, which lack the necessary semantic information, the proposed method transfers knowledge from the last transformer layer to guide the training of the former layers and ensure they remain consistent with the behavior of the final classifier. Even if the VL-PTMs exit very early, they can still achieve competitive performance with the original VL-PTMs model. Inspired by Sun et al. [18], the mean squared error (MSE) is used to minimize the distance of the feature maps between the teacher and student model to guide the training, and further enhance the representation ability of the EE classifiers.

Comparative experiments were conducted on various vision and language multimodal tasks. The results showed that the proposed method significantly outperformed the previous EE and KD methods. Multimodality distillation is superior to conventional distillation because the former layers mimic the final classifier on each modality for better knowledge transfer. Even if the PTMs exit very early, performance and inference time are better balanced with little performance loss.

The remainder of this paper is structured as follows. Section II reviews related work on compression methods for VL-PTMs. Section III describes the proposed multimodality self-distillation method. Section IV presents extensive experiments for comparison with several existing methods. Finally, conclusions are presented in Section V.

II. RELATED WORK

A. Vision and Language Pretrained Models

In relevant fields of both natural language processing tasks [28], [29] and computer vision tasks [30], [31], transformer-based models [32], [33], [34], [35] have achieved remarkable success in multimodality tasks, such as the embedded AI vision-language navigation task. Since only minimal information can be obtained from language instructions, Gao et al. [36] proposed a cross-modality knowledge reasoning (CKR) model utilizing common-sense knowledge to address the remote embodied visual referring expression in real indoor environments (REVERIE) task [37]. Furthermore, Qiao et al. [38] proposed a history-enhanced and order-aware pretraining with the complementing fine-tuning paradigm (HOP+) for vision-and-language navigation. The regions of interest (ROI) are extracted from an image modality as image features using the Fast R-CNN algorithm [39]. At the same time, texts are mapped as a token representation, followed by applying a dual- or single-encoder architecture.

Dual-encoder Architecture: The dual-encoder architectures assign a separate transformer encoder to each modality and learn the output embeddings separately. Then, several projection layers are added to both the vision and text encoders to project the output embeddings to a shared latent space. Based on this, LXMERT [40] and ViLBERT [4] introduce extra pre-training tasks to enhance the performance of these encoders, including masked multimodality modeling and multimodality alignment prediction. 12-in-1 [41] is a vision-language multitask learning model based on ViLBERT as the backbone. VL-BERT [5] is pretrained on visual-linguistic and raw text datasets. Considering the confounding effect, CATT [42] uses causal attention to eliminate confounding effects in existing attention-based visual language methods. The results demonstrate that joint pretraining can improve the generalization of complex sentences and enhance the performance of visual representations.

Single-encoder Architecture: Different from the dual-encoder architecture, the single-encoder architecture feeds text and images into a joint transformer encoder. The main challenge of these models lies in the alignment of latent visual and textual spaces. To accomplish this goal, UNITER [7] introduces four extra pretraining tasks, including masked language modeling, image-text matching, word-region alignment, and masked region modeling. The results indicate that joint image-text pretraining is more effective than separate pretraining on either vision or language. Similarly, Pixel-BERT [43] applies the pixel-level image feature to complement the language information, bridging the gap between ROI features and language understanding. Oscar [44] aligns image and text modalities in the same shared semantic space using the detected object labels as anchor points. InterBERT [45] proposes a broader range of masking operations and modality feature fusion, and retains modality independence. VILLA [46] improves generalization capabilities using adversarial and adversarial fine-tuning.

B. Model Compression

Model compression seeks to minimize model size while retaining model performance, thereby reducing the neural network footprint, increasing its inference speed, and reducing energy consumption. Existing model compression methods include pruning, quantization, knowledge distillation and conditional computation.

Pruning applies a binary criterion to identify weights for pruning, with weights that match the pruning criteria assigned a value of zero [47], [48], [49]. Pruned elements are trimmed from the model, i.e., their values are zeroed and are excluded from back-propagation. Based on this, several studies [50], [51] have investigated the importance of model parameters and neurons during the training process. The less important parameters and neurons are then zeroed, which may negatively impact network accuracy.

Model Quantization refers to reducing the number of bits representing a number. It converts high numerical precision integers, e.g., usually 32-bit float, into low-precision integers, e.g., 8-bit integers, thus effectively reducing computational cost and parameter size to accelerate model inference. For the transformer, Q-BERT [16] implements a hybrid precision quantization for the BERT model. For visual transformer, Liu et al. [52] proposed quantizing similarity perception and ranking-aware quantization for feed-forward networks and multi-head self-attention in encoders. In addition, Gao et al. [53] proposed simultaneously quantizing the activation function and weight parameter to reduce quantization errors.

Knowledge Distillation is a model compression method in which a trained larger model is used as a teacher model to supervise a smaller untrained model as the student. The knowledge contained in the teacher model can then be transferred to the student model through a special distillation operation. Hinton et al. [11] used the category probability distributions of the final classification layer as soft labels and minimized the KL-divergence between the teacher and student models for information transfer. Unlike using the final classification layer for information transfer, BERT-PKD [18] further learns from the intermediate layer. TinyBERT [54] performs knowledge distillation during pretraining and task-specific fine-tuning phases and uses data augmentation to improve the student model accuracy.

Conditional Computation refers to a class of algorithms in which each input sample uses a different part of the model, thereby reducing the average computational resource requirements, latency, or power consumption. The most widely used method is adaptive inference, which usually inserts additional early exiting classifiers between each of the two transformer layers of the PTMs. After the input samples pass through the encoder layers, the early exiting classifier determines whether prediction confidence is sufficiently strong. The result is returned once the early exiting classifier is sufficiently confident; otherwise, the samples are passed to the next layer and the computation is repeated. DeeBERT [23] inserts additional classifiers for each pair of transformer layers and adopts a two-stage training approach. First, the backbone of BERT is fine-tuned for

TABLE I
CHARACTERISTICS OR TECHNIQUES OF PREVIOUS METHODS AND THE PROPOSED MSD METHOD FOR ADAPTIVE INFERENCE

	Early Exiting Classifier	Two-stage Training	Joint Training	Self-distillation	Multimodality Self-distillation
DeeBERT	✓	✓			
RightTool	✓	✓			
FastBERT	✓	✓		✓	
PABEE	✓		✓		
Proposed MSD	✓		✓	✓	✓

downstream tasks. Then, it is frozen to fine-tune the early exiting classifiers. Such a two-stage strategy brings extra costs for computation and training time. RightTool [21] applies the early exiting approach to BERT on the document ranking task. To ensure that the former layers can obtain powerful representation ability, FastBERT [26] adopted the self-distillation strategy to improve the stability of the EE classifier. PABEE [24] proposes a patience-based exit strategy considering the consistency of early exiting classifier predictions. The model will stop inference and exit if the exit prediction remains constant for a preset time. The performance of former layers is reduced due to the lack of high-level semantic information in former layers.

C. Discussion

Table I summarizes the characteristics or techniques of existing approaches to adaptive inference. The EE strategy with adaptive inference dynamically activates only some of the parts in a model according to the properties of the input samples. The immediate effect of activating fewer units accelerates information propagation through the network in training and testing.

Based on this, DeeBERT [23] inserts additional EE classifiers for each pair of transformer layers and adopts a two-stage training approach. RightTool [21] also applies the same EE structure in document ranking and sets different thresholds to alleviate class distribution imbalance. Unfortunately, the information learned by the former EE classifiers differs from that of the final classifier, often resulting in divergent predictions. As a result, the earlier the model exits, the fewer semantic features required for the task are learned, thus degrading performance.

To address this shortcoming, PABEE [24] introduced a patience strategy to preserve the consistency of EE classifier predictions. Meanwhile, FastBERT [26] applied a self-distillation approach to transfer knowledge from the final classifier to the EE classifiers, improving the representation ability of the former layers. By directly introducing these methods into multimodal tasks, the EE classifiers are trained to mimic the outputs of the final classifier without access to the modality-specific behaviors. As a result, the semantic gap occurred between the modality-specific behaviors of the teacher and the student, and self-distillation was inefficient since the EE classifiers did not carefully mimic the modality-specific prediction. Furthermore, the two-stage training strategy of these methods increases computational and training time costs.

This paper proposes splitting the multimodalities into separate modalities to fill the semantic gap between modalities. It adds them as extra inputs to promote the effective distillation of each modality and obtain modality-specific information. Furthermore, MSE is used to minimize the distance of feature maps

between the teacher and student model to guide the training and further enhance the representation ability of the EE classifiers.

III. MULTIMODALITY SELF-DISTILLATION

Fig. 2 shows an overview of the proposed multimodality self-distillation, unifying knowledge distillation and EE with dynamic inference to accelerate the VL-PTMs. The last transformer layer is used as the teacher to guide the training of the former layers, such that these layers can mimic the teacher's behavior. The texts and images are used as separate inputs for self-distillation to learn the individual features of language and vision, respectively. The confidence in each EE classifier is measured to determine whether the model should be returned to this layer. The details of each module are presented as follows.

A. Early Exiting Classifier

Typically, a VL-PTM contains K layers of transformers [55]. The multimodal input contains both image regions and text words, which are then encoded as a representation sequence of both image and text, that is, $V = \{v_1, v_2, \dots, v_o\}$ and $W = \{w_1, w_2, \dots, w_q\}$, where o and q respectively denote the number of visual regions and textual tokens. The corresponding ground-truth label is y . A special token [CLS] is added to the head of the sequence such that the corresponding hidden state $h_{[\text{CLS}]^{(k)}} \in \mathbb{R}^{d_h}$ in each layer is a joint representation of both images and texts. For the k -th layer, the encoding process of the transformer is defined as follows:

$$\begin{aligned} & \left[h_{[\text{CLS}]^{(k)}}, h_{v_1}^{(k)}, \dots, h_{v_o}^{(k)}, h_{w_1}^{(k)}, \dots, h_{w_q}^{(k)} \right] \\ &= f^{(k)} \left(\left[h_{[\text{CLS}]^{(k-1)}}, h_{v_1}^{(k-1)}, \dots, h_{v_o}^{(k-1)}, \right. \right. \\ & \quad \left. \left. h_{w_1}^{(k-1)}, \dots, h_{w_q}^{(k-1)} \right] \right) \end{aligned} \quad (1)$$

where $f^{(k)}$ denotes the k -th layer of the transformer encoder. The standard approach of EE is to add EE classifiers in the intermediate layers, similar to the final classifier in the last layer. Each EE classifier is a fully connected layer with *softmax* activation. The EE classifiers take as input the joint embeddings corresponding to the [CLS] token, that is, $h_{[\text{CLS}]^{(k)}}$ in the k -th layer of the VL-PTMs, which is formulated as follows:

$$z^{(k)} = W_z^{(k)} h_{[\text{CLS}]^{(k)}} + b_z^{(k)} \quad (2)$$

$$\hat{y}_z^{(k)} = \text{softmax} \left(z^{(k)} \right) \quad (3)$$

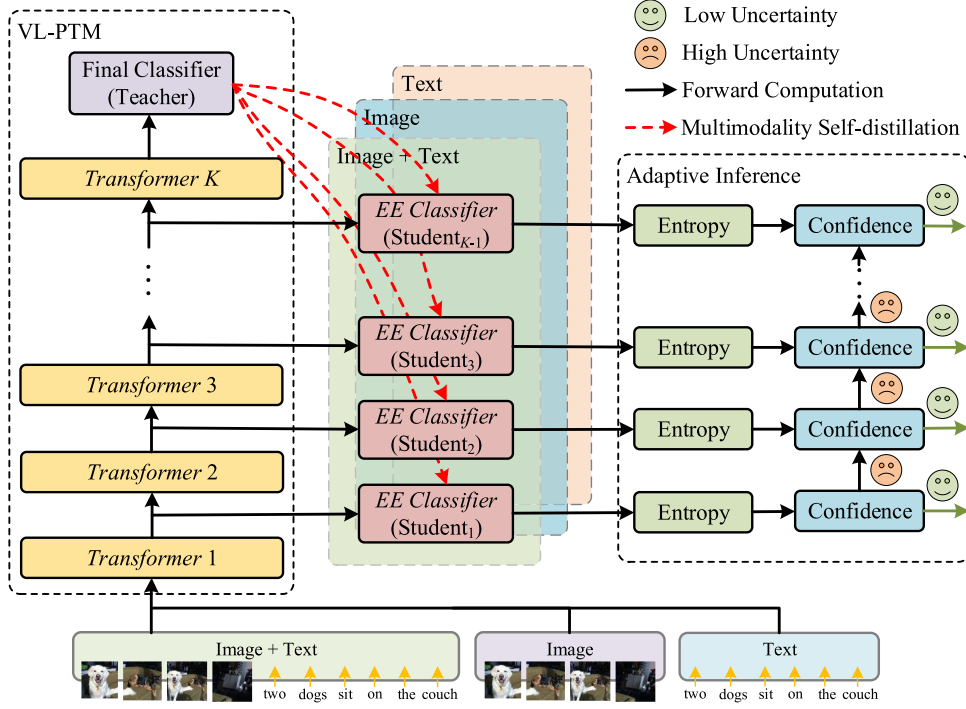


Fig. 2. Overall architecture of the proposed multimodality self-distillation for fast inference of VL-PTMs.

where $W_z^{(k)} \in \mathbb{R}^{C \times d_h}$ and $b_z^{(k)} \in \mathbb{R}^C$ respectively represent the weights and bias of the k -th EE classifier, and C denotes the number of classes. The training objective of these classifiers is categorical cross-entropy (CE) \mathcal{L}_{CE} , defined as follows:

$$\mathcal{L}_{CE} = - \sum_{k=1}^K \mathbb{I}(y) \circ \log(\hat{y}_z^{(k)}) \quad (4)$$

where y and $\hat{y}_z^{(k)}$ denote the corresponding ground-truth label and probability distribution of the k -th layer, respectively. $\mathbb{I}(y)$ denotes a one-hot label and \circ represents an element-wise multiplication operation.

B. Multimodality Self-Distillation

Using only the ground-truth label to train EE classifiers will result in their representation abilities diverging from the final classifier because of their different inputs. Thus, self-distillation was applied to encourage EE classifiers in the former layers to mimic the behavior of the final classifier and obtain rich semantic information in the hidden representation $h_{[\text{CLS}]}^{(k)}$. As shown in Fig. 3, the classifier in the K -th layer was regarded as the teacher model, whereas the other EE classifiers in the former layers were regarded as the student model. In knowledge distillation [11], student models can learn from the distribution of teacher models to improve their classification performance. The feature maps $z_s^{(k)} \in \mathbb{R}^{d_p}$ and $z_t^{(K)} \in \mathbb{R}^{d_p}$ respectively of the EE classifier (student) and the final classifier (teacher), are denoted as follows,

$$z_s^{(k)} = W_s^{(k)} h_{[\text{CLS}]}^{(k)} + b_s^{(k)} \quad (5)$$

$$z_t^{(K)} = W_t^{(K)} h_{[\text{CLS}]}^{(K)} + b_t^{(K)} \quad (6)$$

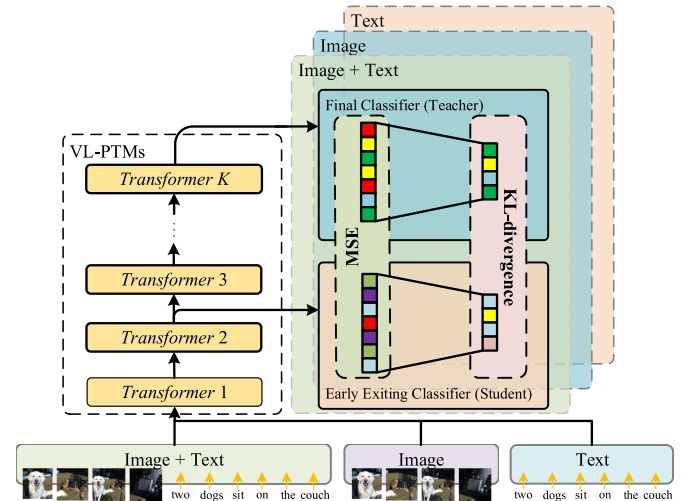


Fig. 3. Overall architecture of the proposed multimodality self-distillation for fast inference of VL-PTMs.

where $h_{[\text{CLS}]}^{(k)}$ and $h_{[\text{CLS}]}^{(K)}$ are hidden representations in the k -th and the final transformer layers, and $W_s^{(k)}$, $b_s^{(k)}$, $W_t^{(K)}$ and $b_t^{(K)}$ are weights and biases associated with the EE classifiers and the final classifier. Each EE classifier is required to mimic the behavior of the final classifier. The self-distillation loss function $\mathcal{L}_{\text{KL}}^{(k)}$ measures the Kullback-Leibler (KL) divergence between the k -th EE classifier and the final classifier, denoted as

$$p_s^{(k)} = \text{softmax}(z_s^{(k)} / \tau) \quad (7)$$

$$p_t^{(K)} = \text{softmax}(z_t^{(K)} / \tau) \quad (8)$$

$$\mathcal{L}_{\text{KL}}^{(k)} = \tau^2 \text{KL} \left(p_s^{(k)} \parallel p_t^{(K)} \right) \quad (9)$$

where $\text{KL}(\bullet \parallel \bullet)$ denotes the KL-divergence, τ denotes the temperature, which is used to control the softness of the distribution, and τ^2 compensates for the size of the gradient scaled by the soft target, ensuring that there is no negative impact on the gradient size. $p_s^{(k)} \in \mathbb{R}^{d_p}$ and $p_t^{(K)} \in \mathbb{R}^{d_p}$ respectively represent the soft probability distributions of the k -th EE classifier (student) and the final classifier (teacher). Notably, $p_s^{(k)}$ are the soft probability distributions using both textual $w_i \in \mathbb{R}^{d_w}$ and visual $v_i \in \mathbb{R}^{d_v}$ modalities as input.

Considering that the input sample contains both image and text modalities, we add each modality as an extra individual input to the VL-PTMs separately, such that the students can separately learn the effective information of the teacher model toward a specific modality to fill the semantic gap between different modalities. By successively masking the inputs of visual modality $v_i \in \mathbb{R}^{d_v}$ and textual modality $w_i \in \mathbb{R}^{d_w}$, we respectively obtain the output soft probability distribution of textual modality $p_w^{(k)}$ and visual modality $p_v^{(k)}$. Thus, two extra losses of KL-divergence were introduced to each modality, respectively denoted as,

$$\mathcal{L}_{\text{T-KL}}^{(k)} = \tau^2 \text{KL} \left(p_w^{(k)} \parallel p_w^{(K)} \right) \quad (10)$$

$$\mathcal{L}_{\text{I-KL}}^{(k)} = \tau^2 \text{KL} \left(p_v^{(k)} \parallel p_v^{(K)} \right) \quad (11)$$

where $\mathcal{L}_{\text{T-KL}}^{(k)}$ and $\mathcal{L}_{\text{I-KL}}^{(k)}$ respectively represent the self-distillation loss functions of the textual and visual modalities. The multimodality self-distillation (MS) loss \mathcal{L}_{MS} is defined as follows:

$$\mathcal{L}_{\text{MS}} = \lambda \sum_{k=1}^{K-1} \mathcal{L}_{\text{KL}}^{(k)} + \lambda^w \sum_{k=1}^{K-1} \mathcal{L}_{\text{T-KL}}^{(k)} + \lambda^v \sum_{k=1}^{K-1} \mathcal{L}_{\text{I-KL}}^{(k)} \quad (12)$$

where λ s denote hyper-parameters used to balance the different modality loss functions.

Using only the logits of the teacher for knowledge distillation is insufficient to make the students imitate the teacher's behavior entirely. Inspired by Sun et al. [18], we also minimize the mean squared error (MSE) between the input features of the student and the teacher for each input modality.

$$\mathcal{L}_{\text{MSE}}^{(k)} = \text{MSE} \left(z_s^{(k)}, z_t^{(K)} \right) \quad (13)$$

$$\mathcal{L}_{\text{T-MSE}}^{(k)} = \text{MSE} \left(z_w^{(k)}, z_w^{(K)} \right) \quad (14)$$

$$\mathcal{L}_{\text{I-MSE}}^{(k)} = \text{MSE} \left(z_v^{(k)}, z_v^{(K)} \right) \quad (15)$$

where $z_w^{(k)}$ and $z_v^{(k)}$ respectively denote the feature maps of the EE classifier (student) of the textual and visual modalities. Correspondingly, the mean squared error between the multimodal inputs is defined as follows:

$$\mathcal{L}_{\text{ED}} = \lambda \sum_{k=1}^{K-1} \mathcal{L}_{\text{MSE}}^{(k)} + \lambda^w \sum_{k=1}^{K-1} \mathcal{L}_{\text{T-MSE}}^{(k)}$$

$$+ \lambda^v \sum_{k=1}^{K-1} \mathcal{L}_{\text{I-MSE}}^{(k)} \quad (16)$$

where \mathcal{L}_{ED} denotes the multimodality feature self-distillation loss. The total training objective \mathcal{L} of multimodality self-distillation is formulated as follows,

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{MS}} + \mathcal{L}_{\text{ED}} \quad (17)$$

Instead of using two-stage training in previous EE models, all components for self-distillation in the proposed method can be trained using a standard back-propagation algorithm in an end-to-end manner.

C. Adaptive Inference

Inspired by Teerapittayanon et al. [56], the features learned in the former layers are sufficiently robust to provide *easy* examples to obtain a performance similar to that of the final layer, such that the EE classifier can accelerate the inference. Conversely, *hard* examples must be propagated through the classifier in the latter layers. In practice, most samples are relatively easy; therefore, adaptive inference can be applied to force the model to exit early to reduce the inference time and computational cost.

To determine whether inference can be terminated at a specific layer, the output distribution of the k -th EE classifier is determined by an entropy value $\mathbb{E}^{(k)}$, calculated as follows:

$$\begin{aligned} \mathbb{E}^{(k)} &= - \sum_{d_p} z_s^{(k)} \log z_s^{(k)} \\ &= \ln \left(\sum_{d_p} \exp \left(z_s^{(k)} \right) \right) - \frac{\sum_{d_p} z_s^{(k)} \exp \left(z_s^{(k)} \right)}{\sum_{d_p} \exp \left(z_s^{(k)} \right)} \end{aligned} \quad (18)$$

where d_p denotes the dimensionality of the hidden representation of the EE classifier. Here, $\mathbb{E}^{(k)}$ measures the uncertainty in the output of the EE classifier, i.e., $z_s^{(k)}$. The higher the entropy value, the higher the EE classifier's uncertainty. If the uncertainty $\mathbb{E}^{(k)}$ is lower than a preset threshold F , the EE classifier is sufficiently confident to achieve performance competitive with the final classifier. The model then takes the prediction of the EE classifier as the result, and immediately suspends the inference of the latter layers. Otherwise, it continues to execute on the next layer until it falls below the required threshold.

The right part in Fig. 2 shows the adaptive inference of VL-PTMs. Intuitively, the former classifiers predict the *easy* samples, whereas the latter classifiers predict only the *hard* examples. Based on this, adaptive inference can drastically improve efficiency by reducing computation requirements on the portion of easy examples in the dataset.

IV. EXPERIMENTS

A. Experimental Setup

1) *Dataset*: We conducted experiments on different multimodal datasets to evaluate the effectiveness of the proposed multimodality self-distillation model inference. Descriptive statistics of the datasets are shown in Table II.

TABLE II
DESCRIPTIVE STATISTICS OF THE DATASETS

Task	Dataset	Image Src.	#Images	#Text	Metric	#Train	#Dev	#Test
NLVR	NLVR2	ILSVRC-2014 ImageNet	107k	29k	Accuracy	86k	6k	14k
VE	SNLI-VE	Flickr30k	31k	528k	Accuracy	529k	17k	17k
VQA	VQA 2.0	MS COCO	204k	1105k	Accuracy	443k	214k	447k

- Visual entailment (**SNLI-VE**) is used to predict whether a given image semantically contains an input sentence. The model’s performance was measured using the classification accuracy over three categories: *entailment*, *neutral*, and *contradiction*.
- Visual question answering (**VQA**) refers to answering content-related questions for an image. The dataset was divided into three subsets: train, val, and test. The test subset was further divided into test-dev and test-std for online evaluation.
- Natural language for visual reasoning (**NLVR2**) determines whether the correspondence between a pair of images and natural language captions is consistent.

2) *Inference Time Measurement*: The runtime of the VL-PTMs is highly dependent on the hardware environment and is thus unstable in most cases. Following Xin et al. [23] and Zhou et al. [24], we gradually adjusted the threshold F to measure the time reduction ratio ρ by comparing the adaptive inference with the original execution with complete layers, that is, the ratio of the EE layers to the originally required total layers for all samples. For a K -layers model, the time reduction ratio ρ of the inference is measured by

$$\rho = \frac{\sum_{k=1}^K k \times m^{(k)}}{K \times M} \quad (19)$$

where $m^{(k)}$ denotes the number of samples exited at the k -th layer, and K and M respectively denote the number of layers and samples.

3) *Implementation Details*: Both UNITER [7] and Oscar [44] were used as the backbone model of VL-PTMs. They all included 12 layers and 768 hidden dimensions. For the VQA and SNLI-VE datasets, we fed image and text pairs into UNITER. Then we extracted the joint embeddings corresponding to the [CLS] token with a fully connected layer as the final representation of the input image and text pairs. For the NLVR2 dataset, each input sample contains a description text with two images. The two outputs of the text and image were integrated using a bi-attention layer for classification. The AdamW optimizer [57] was used for the training. The learning rate and weight decay were $8e-5$ and 0.01 for VQA, $7e-6$ and 0.01 for SNLI-VE, and $3e-5$ and 0.01 for NLVR2, respectively. The maximum length of the input text was 128. Using the grid search strategy, the optimal settings of λ , λ^w and λ^v were respectively 0.5 , 0.25 and 0.25 for VQA and SNLI-VE, and 0.7 , 0.15 , and 0.15 for NLVR2.

B. Baselines

To comprehensively evaluate the proposed multimodality self-distillation, comparative experiments were conducted

against various knowledge distillation, and early exiting methods as well as several complete VL-PTMs, including VisualBERT [58], UNITER [7], LXMERT [40], UNITER-KD, UNITER-EE. The UNITER-KD denotes the original UNITER model, which was compressed using conventional knowledge distillation. The UNITER-EE denotes that an EE method was introduced into the UNITER model. An EE classifier was added to each intermediate layer of UNITER.

C. Hyperparameters Fine-Tuning

Several hyperparameters may affect the performance of the proposed MSD method in downstream tasks. Fig. 4 shows the optimal settings for selecting different hyperparameters depending on the final performance of the development set. We fine-tuned each parameter to obtain the optimal value, which is then fixed so that the other parameters can be fine-tuned in turn.

For the temperature τ in (5) and (6), we try to select the superior values in the set of temperature candidates in 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, as shown in Fig. 4(b). VQA, SNLI-VE, and NLVR2 perform best at τ of 3, 2, 5, respectively.

The balance coefficients λ , λ^w , and λ^v in (10) are used to balance the information of different modalities. We also use different combinations for the grid search while ensuring that the different balance coefficients sum to 1. As shown in Fig. 4(a), the proposed method performs best on the VQA dataset when λ is set to 0.5 and both λ^w , and λ^v are set to 0.25. The proposed method performs best on the NLVR2 and SNLI-VE datasets when λ is set to 0.7 and both λ^w , and λ^v are set to 0.15. This illustrates that the information of different modalities has different effects on the model in different tasks. Once these parameters are exceeded, the performance of the optimal settings will degrade. The results indicate that appropriate parameters can improve the performance of early exiting for accelerating inference.

D. Comparative Results

To fairly compare the proposed method with the baseline, we tuned the time reduction ratio ρ of the proposed method to be consistent with that of the corresponding baselines. Different expected ρ values were obtained by adjusting the confidence threshold F . For original the VL-PTMs (e.g., UNITER, LXMERT, VL-BERT (Large), and Visual-BERT), ρ is 100%. Tables III and IV compare the accuracy under a certain time reduction ratio of the proposed UNITER-MSD and Oscar-MSD against the baselines. Experiments with different VL-PTMs as the backbone demonstrate the effectiveness and generalizability of the method. Both UNITER and Oscar were used as the backbone for these baselines.

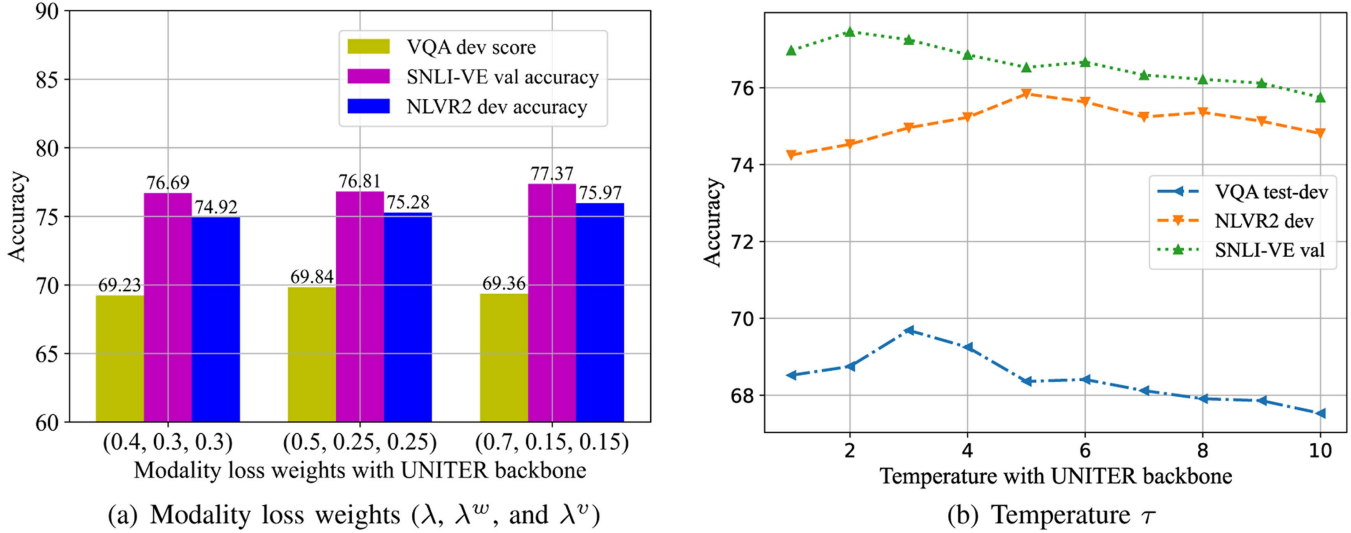


Fig. 4. Hyperparameters fine-tuning on different datasets.

TABLE III
EXPERIMENTAL RESULTS FOR COMPARING BASELINE METHODS ON MULTIMODAL DATASETS WITH UNITER BACKBONE

ρ	Methods	SNLI-VE			NLVR2			VQA				
		Val	Time%	Test	Time%	Dev	Time%	Test-P	Time%	Test-dev	Test-std	Time%
$\sim 100\%$	UNITER	78.59	100	78.28	100	77.18	100	77.85	100	72.70	72.9	100
	LXMERT	-	-	-	-	74.90	100	74.50	100	72.42	72.54	100
	VL-BERT(Large)	-	-	-	-	-	-	-	-	71.79	72.22	100
	VisualBERT	77.57	100	77.32	100	67.40	100	67.00	100	70.80	71.00	100
$\sim 50\%$	UNITER-KD-6L	75.14	50	75.41	50	57.05	50	57.93	50	69.20	69.33	50
	UNITER-EE	76.51	49.09	76.44	48.97	75.18	50.49	75.60	50.29	69.26	69.49	50.51
	UNITER-MSD	77.37	48.26	76.96	48.55	75.84	49.34	75.94	50.23	69.69	69.84	49.76
$\sim 42\%$	UNITER-KD-5L	74.07	41.67	74.29	41.67	56.30	41.67	55.92	41.67	69.08	69.27	41.67
	UNITER-EE	76.45	41.48	76.39	41.36	74.18	41.50	74.42	41.32	67.40	67.73	43.67
	UNITER-MSD	77.29	40.34	76.94	40.44	75.12	40.41	75.20	41.00	69.11	69.32	41.12
$\sim 33\%$	UNITER-KD-4L	74.02	33.33	74.23	33.33	54.57	33.33	53.41	33.33	67.11	67.43	33.33
	UNITER-EE	76.32	32.45	76.25	32.73	72.99	34.27	72.92	33.32	65.55	65.83	33.76
	UNITER-MSD	77.21	31.98	76.78	32.03	73.59	32.78	73.69	32.66	67.26	67.51	33.27

Compared to the original complete VL-PTMs, the proposed UNITER-MSD and Oscar-MSD significantly improved efficiency and reduced inference times with little performance loss. Compared to the original complete UNITER, the performance of the UNITER-MSD decreased by 1.22% and 1.34% respectively in SNLI-VE and NLVR2, but the efficiency improved by 50% ($\rho \sim 50\%$) in both SNLI-VE and NLVR2. Furthermore, UNITER-MSD achieved extremely competitive results but faster inference than the original VisualBERT in NLVR2. Oscar-MSD outperformed LXMERT and VisualBERT on the time reduction ratios ($\rho \sim 50\%$).

With different time reduction ratios ρ , the proposed UNITER-MSD and Oscar-MSD outperformed the KD and EE methods. For $\rho \sim 50\%$, UNITER-MSD outperformed UNITER-KD by 2.23% and UNITER-EE by 0.86% on the SNLI-VE dataset. Similar improvements were observed in the other datasets. This was because multimodality self-distillation allowed the performance of EE classifiers to approximate that of the final classifier closely and to learn high-level information. If the time reduction ratio requirement ρ changes, a new student model must be trained from scratch using KD. Thus, it does not apply to

different platforms with different requirements of ρ . Conversely, the proposed UNITER-MSD can perform adaptive inference by adjusting the threshold F to provide a different ρ with robust performance.

E. Ablation Experiments

Table V presents the results of the ablation experiments to evaluate each component's effectiveness in the proposed UNITER-MSD. We successively removed one or more loss functions, that is, \mathcal{L}_{MS} in (12) and \mathcal{L}_{ED} in (16). UNITER-MSD w/o MS indicates the removal of \mathcal{L}_{MS} . UNITER-MSD w/o ED indicates the removal of \mathcal{L}_{ED} . UNITER-MSD w/o MS & ED indicates the removal of both \mathcal{L}_{MS} and \mathcal{L}_{ED} , that is, UNITER-EE. UNITER-MSD w/o SD denotes EE strategy and separate modalities as inputs, and without KD. As indicated, the removal of each component of the proposed method will degrade the performance, indicating that the \mathcal{L}_{MS} and \mathcal{L}_{ED} loss functions are indispensable to performance improvement. Namely, the main advantage of UNITER-MSD comes from the proposed multimodality self-distillation and self-distillation with MSE.

TABLE IV
EXPERIMENTAL RESULTS FOR COMPARING BASELINE METHODS ON MULTIMODAL DATASETS WITH OSCAR BACKBONE

ρ	Methods	NLVR2				VQA		
		Dev	Time%	Test-P	Time%	Test-dev	Test-std	Time%
$\sim 100\%$	Oscar	78.07	100	78.36	100	73.16	73.44	100
	LXMERT	74.90	100	74.50	100	72.42	72.54	100
	VL-BERT(Large)	-	-	-	-	71.79	72.22	100
	VisualBERT	67.40	100	67.00	100	70.80	71.00	100
$\sim 50\%$	OSCAR-FT-6L	74.51	50	74.67	50	69.95	70.64	50
	Oscar-KD-6L	76.05	50	76.33	50	71.12	71.33	50
	Oscar-EE	75.23	52.06	75.36	52.66	70.36	70.51	52.37
	Oscar-MSD	76.53	50.36	76.64	50.23	71.58	71.82	51.23
$\sim 42\%$	OSCAR-FT-5L	73.08	41.67	73.16	41.67	68.26	68.39	41.67
	Oscar-KD-5L	73.66	41.67	73.87	41.67	69.52	69.65	41.67
	Oscar-EE	73.38	42.67	73.56	42.13	68.66	68.89	42.56
	Oscar-MSD	75.06	43.10	75.17	42.86	70.17	70.53	42.15
$\sim 33\%$	OSCAR-FT-4L	72.14	33.33	72.25	33.33	67.85	67.96	33.33
	Oscar-KD-4L	73.31	33.33	73.46	33.33	68.97	69.10	33.33
	Oscar-EE	72.87	34.56	72.95	34.23	68.05	68.37	34.78
	Oscar-MSD	73.70	32.87	73.85	33.18	69.36	69.72	34.55

TABLE V
RESULTS OF THE ABLATION STUDY OF THE PROPOSED UNITER-MSD MODEL

ρ	Methods	NLVR2					SNLI-VE				
		entropy	Dev	Time%	Test-P	Time%	entropy	Val	Time%	Test	Time%
$\sim 60\%$	UNITER-MSD w/o ED	0.03	76.07	58.21	76.43	59.76	0.04	76.70	59.46	76.63	59.56
	UNITER-MSD w/o MS	0.02	76.17	58.14	76.51	60.07	0.09	76.84	59.40	76.86	59.33
	UNITER-MSD w/o MS & ED	0.03	76.04	57.94	76.29	59.08	0.04	76.61	60.26	76.54	60.22
	UNITER-MSD w/o SD	0.04	76.06	58.05	76.40	59.02	0.06	76.68	60.18	76.60	60.15
	UNITER-MSD w/o ED-T&I	0.05	76.15	58.36	76.48	59.86	0.06	76.92	59.66	76.73	60.12
	UNITER-MSD w/o MS-T&I	0.04	76.19	57.89	76.53	60.10	0.08	76.98	60.15	76.88	59.68
	UNITER-MSD w/o MS-T&I & ED-T&I	0.05	76.10	58.28	76.42	60.05	0.05	76.86	59.86	76.62	60.35
	UNITER-MSD	0.06	76.30	57.25	76.59	58.73	0.07	77.43	58.59	76.97	58.62
$\sim 40\%$	UNITER-MSD w/o ED	0.20	74.32	40.59	74.56	41.07	0.20	76.67	39.81	76.61	39.88
	UNITER-MSD w/o MS	0.12	74.36	40.26	74.84	41.05	0.28	76.81	40.28	76.83	40.51
	UNITER-MSD w/o MS & ED	0.20	74.18	41.50	74.42	41.32	0.17	76.45	41.48	76.39	41.36
	UNITER-MSD w/o SD	0.22	74.29	41.46	74.53	41.30	0.21	76.63	40.23	76.58	40.42
	UNITER-MSD w/o ED-T&I	0.17	74.39	40.46	74.86	40.89	0.19	76.83	41.25	76.72	41.25
	UNITER-MSD w/o MS-T&I	0.18	74.42	41.38	74.91	41.08	0.26	76.89	40.65	76.75	41.36
	UNITER-MSD w/o MS-T&I & ED-T&I	0.28	74.27	40.88	74.49	41.23	0.23	76.65	41.36	76.61	40.93
	UNITER-MSD	0.24	74.85	39.76	75.14	40.29	0.24	77.27	39.61	76.91	39.69

Specifically, UNITER-MSD without MS and ED performed 0.82% lower than UNITER-MSD on the SNLI-VE dataset, and 0.26% lower on NLVR2 for $\rho \sim 60\%$. Without the MS and ED loss function, the former layers only learned low-level features that are not competent for the final classification. Furthermore, UNITER-MSD w/o SD performs better than UNITER-MSD w/o MS & ED, illustrating the effectiveness of different modalities as additional inputs for multimodal learning and for filling the semantic gap between modalities.

A similar observation can be obtained that the accuracies of UNITER-MSD without MS and without ED were lower than that of UNITER-MSD by 0.53% and 0.49%, respectively, for NLVR2, and 0.44% and 0.37% for SNLI-VE for $\rho \sim 40\%$. There was no MS loss function or reduction in model performance. This indicates that multimodality self-distillation encourages the former layers to learn useful information about specific modalities. Furthermore, UNITER-MSD without MS simultaneously distills texts and images. Unfortunately, if one modality is dominant, the distillation of the other will fail, leading to a performance decline. Without the ED loss function, the EE classifiers are too shallow to learn enough information for classification.

To further examine the effect of adding individual modalities in multimodality self-distillation, we removed the loss functions of text (T) and image (I) modalities in \mathcal{L}_{MS} and \mathcal{L}_{ED} . UNITER-MSD w/o MS-T&I indicates the removal of \mathcal{L}_{T-KL} and \mathcal{L}_{I-KL} in \mathcal{L}_{MS} , UNITER-MSD w/o ED-T&I indicates the removal of \mathcal{L}_{T-MSE} and \mathcal{L}_{I-MSE} in \mathcal{L}_{ED} , and UNITER-MSD w/o MS-T&I & ED-T&I indicates the removal of the \mathcal{L}_{T-KL} and \mathcal{L}_{I-KL} terms in \mathcal{L}_{MS} and \mathcal{L}_{T-MSE} and \mathcal{L}_{I-MSE} in \mathcal{L}_{ED} . Note that removing the loss functions of individual modalities (T&I) still retains the loss function of the joint modality \mathcal{L}_{KL} in \mathcal{L}_{MS} and \mathcal{L}_{MSE} in \mathcal{L}_{ED} . That is, comparing UNITER-MSD and w/o MS-T&I (or w/o ED-T&I) can show the effect of individual modalities, and comparing w/o MS-T&I (or w/o ED-T&I) and w/o MS (or w/o ED) can show the effect of the joint modality. The results show that the performance degradation from UNITER-MSD to w/o MS-T&I is greater than that from w/o MS-T&I to w/o MS. Comparing UNITER-MSD, w/o ED-T&I and w/o ED also shows similar results. These findings indicate that adding individual modalities in self-distillation contributes more than using the joint modality. To further verify the effectiveness of individual modalities, we measured the KL divergence of the final layer output and the other former layers'

TABLE VI
NUMBER OF PARAMETERS AND TRAINING TIME FOR DIFFERENT MODELS SNLI-VE.

Model	# Param (Transformers)	# Param (EE classifiers)	Total Params	Train Time (Step)	ρ	Acc
UNITER12	111.1 M	N/A	111.1 M	4000 (1 \times)	100 (1 \times)	78.59
UNITER-MSD	111.1 M	25.7 M	136.7 M	9500 (2.37 \times)	32.55 (3.07 \times)	77.32

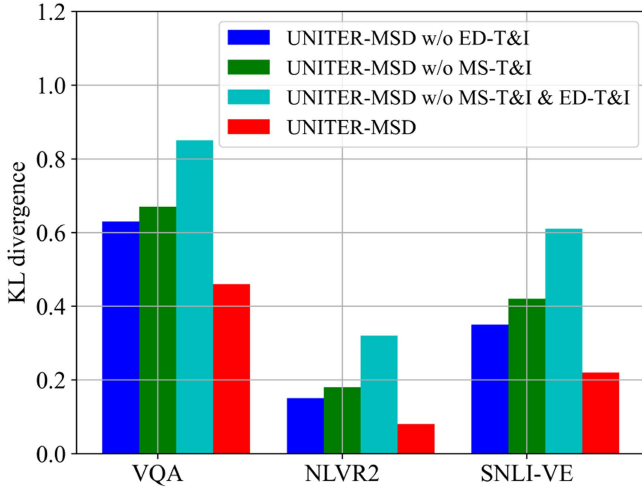


Fig. 5. KL divergence between the final layer output and the outputs of the other former layers before and after removing individual modalities.

outputs to see if their outputs are similar, as shown in Fig. 5. Compared with UNITER-MSD, the KL divergences increased after removing individual modalities (T&I), indicating that the outputs of the final layer and the other former layers diverge without the help of individual modalities. This also shows that using individual modalities as an additional knowledge source improved self-distillation effectiveness.

F. Performance-Time Tradeoff

To explore whether the proposed UNITER-MSD and UNITER-EE performance varies with ρ , Fig. 6 presents the inference time-performance trade-off curves for SNLI-VE and NLVR2.

For the same ρ , the proposed method obtained a better accuracy than UNITER-EE. The performance of UNITER-EE decreased sharply as ρ gradually decreased. As shown in Fig. 6, the proposed method has a high-performance improvement for the low-level early exiting classifier, demonstrating that relatively high performance is guaranteed even when the inference time is reduced. It also shows that the performance of existing EE methods decreases significantly with the inference time. This limits the usefulness of the existing EE methods in meeting higher inference requirements. Conversely, the proposed method can guarantee good performance while reducing the inference time, making it more robust and efficient than the existing EE methods.

G. Analysis of Model Efficiency

To better analyze the model’s efficiency, Table VI shows the model’s training time, size, and reduced inference time.

As indicated, the proposed EE strategy with multimodality self-distillation aims to provide more efficient inference with limited performance loss. Due to the introduction of multimodality self-distillation in the training of the proposed EE strategy, the overall training time has been increased by 2.37 \times . However, the inference speed has also been accelerated by 3.07 \times . Consistent with KD and other EE strategies, the proposed method trades off increased training costs with accelerated inference time. The difference is that the proposed method can dynamically adjust the acceleration ratio according to different running platforms, while KD needs to retrain the model for the new platform.

H. The Effect of Multimodality Self-distillation

Fig. 7 illustrates a detailed analysis of the effects of multimodality self-distillation. The test samples were first divided into several groups according to the layers at which the samples exit. Performance was calculated for each group. For both UNITER-EE and UNITER-MSD, relatively poor performances were obtained for the former layers. In contrast, the latter layers (layers 9–11) all achieved a performance similar to that of the final classifier in the 12th layer. That is, the earlier the model exits, the lower the performance of the model is achieved. On both SNLI-VE and NLVR2 dev datasets, the proposed UNITER-MSD outperformed UNITER-EE by 0.75% and 3.0% on average in all layers, respectively. In the latter layers (layers 9–12), the performances of UNITER-EE and UNITER-MSD were similar. In the former layers (layers 1–6), UNITER-MSD outperformed UNITER-EE by 0.87% and 4.78%, particularly in the first two layers. The rationale is that the former layers of the proposed UNITER-MSD can mimic the final classifier’s behavior to improve the EE classifiers’ performance.

Fig. 8 describes the number of samples that exit early at different layers with different time reduction ratios ρ . The UNITER-KD permanently discards the former six transformer layers in the UNITER of the model and thus can only obtain a fixed ρ in one distillation; the proposed UNITER-MSD can dynamically change the confidence threshold F to satisfy the requirement and obtain different ρ . If the requirement of ρ is strict, 79.92% and 83.50% of the samples tend to exit as soon as possible (layers 1–6) on SNLI-VE and NLVR2, respectively. If the requirement of ρ is slightly lenient, the *hard* samples (51.40% and 44.85%) will choose to exit at the latter layer (layers 9–12), thereby improving model performance. Conversely, both *easy* and *hard* samples will be treated equally and exited at the final classifier using KD.

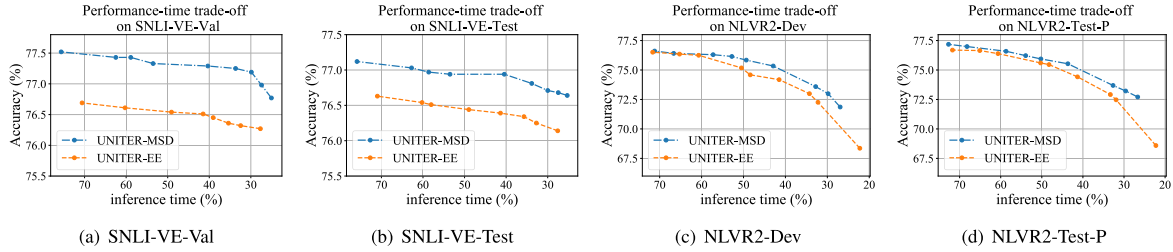


Fig. 6. Performance-time trade-off curve on SNLI-VE and NLVR2 datasets of UNITER-MSD and UNITER-EE.

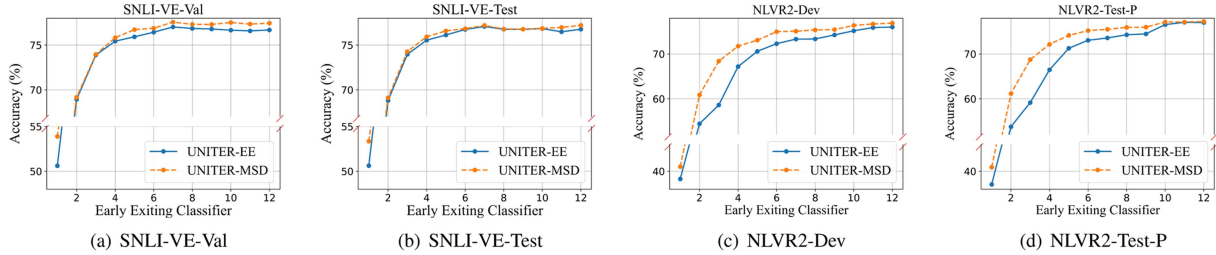


Fig. 7. Performance of early exiting classifiers of UNITER-EE and UNITER-MSD on SNLI-VE and NLVR2 datasets.

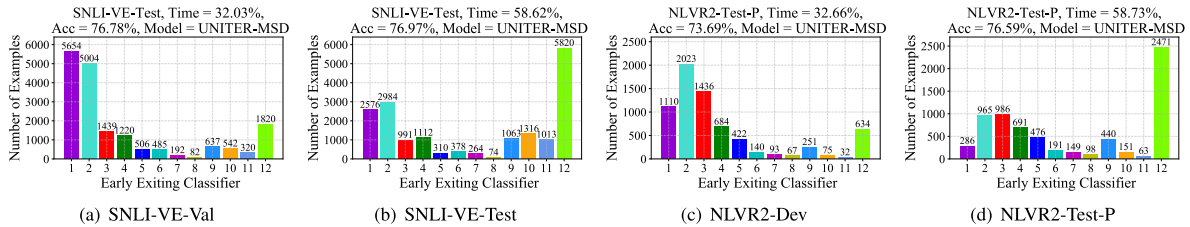


Fig. 8. Statistics of the number of samples that exit early at different layers on SNLI-VE and NLVR2 datasets.

V. CONCLUSION

This paper proposes a multimodality self-distillation method for the fast inference of VL-PTMs to improve existing EE strategies. The classifier in the final layer is used to distill all EE classifiers in the former layers so that the EE classifiers can mimic the behavior of the final classifier to improve performance. To fill the semantic gap between modalities, the multimodalities are split into separate modalities as an extra individual input to encourage the effective distillation of each modality. Furthermore, the MSE was introduced to minimize the distance of feature maps between the teacher and student models and further enhance the representation ability of the EE classifiers. Experiments showed that the proposed method outperformed the KD and EE strategies for the same time reduction requirement, and performed competitively even if the model exited very early.

Future work will explore the problem of mutual interference that can exist in self-distillation between different layers.

REFERENCES

- [1] S. Antol et al., "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2425–2433.
- [2] N. Xie, F. Lai, D. Doran, and A. Kadav, "Visual entailment: A novel task for fine-grained image understanding," 2019, *arXiv:1901.06706*.
- [3] A. Suhr et al., "A corpus for reasoning about natural language grounded in photographs," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 6418–6428.
- [4] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Proc. 33th Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 13–23.
- [5] W. Su et al., "VL-BERT: Pre-training of generic visual-linguistic representations," 2020, *arXiv:1908.08530*.
- [6] G. Li, N. Duan, Y. Fang, M. Gong, and D. Jiang, "Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 11336–11344.
- [7] Y. C. Chen et al., "UNITER: Universal image-Text representation learning," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 104–120.
- [8] Z. Lan et al., "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.
- [9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.
- [10] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
- [11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
- [12] R. Tang et al., "Distilling task-specific knowledge from BERT into simple neural networks," 2019, *arXiv:1903.12136*.
- [13] Z. Sun et al., "MobileBERT: A compact task-agnostic BERT for resource-limited devices," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 2158–2170.
- [14] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?," in *Proc. 33th Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 14014–14024.

- [15] S. Bao, H. He, F. Wang, H. Wu, and H. Wang, "PLATO: Pre-trained dialogue generation model with discrete latent variable," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 85–96.
- [16] S. Shen et al., "Q-BERT: Hessian based ultra low precision quantization of BERT," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 8815–8821.
- [17] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.
- [18] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for BERT model compression," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4323–4332.
- [19] A. Fan, E. Grave, and A. Joulin, "Reducing transformer depth on demand with structured dropout," 2019, *arXiv:1909.11556*.
- [20] O. Zafrir, G. Boudoukh, P. Izsak, and M. Wasserblat, "Q8BERT: Quantized 8Bit BERT," in *Proc. 5th Workshop Energy Efficient Mach. Learn. Cogn. Comput.-NeurIPS Edition (EMC2-NIPS)*, 2019, pp. 36–39.
- [21] R. Schwartz, G. Stanovsky, S. Swayamdipta, J. Dodge, and N. A. Smith, "The right tool for the job: Matching model and instance complexities," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6640–6651.
- [22] J. Xin, R. Nogueira, Y. Yu, and J. Lin, "Early exiting BERT for efficient document ranking," in *Proc. SustainNLP: Workshop Simple Efficient Natural Lang. Process.*, 2020, pp. 83–88.
- [23] J. Xin, R. Tang, J. Lee, Y. Yu, and J. Lin, "DeeBERT: Dynamic early exiting for accelerating BERT inference," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 2246–2251.
- [24] W. Zhou et al., "BERT loses patience: Fast and robust inference with early exit," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 18330–18341.
- [25] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3651–3657.
- [26] W. Liu et al., "FastBERT: A self-distilling BERT with adaptive inference time," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6035–6044.
- [27] W. Wang, D. Tran, and M. Feiszli, "What makes training multi-modal classification networks hard?," in *Proc. IEEE/CVF Comput. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12692–12702.
- [28] R. Socher et al., "Recursive deep models for semantic compositionality Over a sentiment treebank richard," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.
- [29] W. B. Dolan and C. Brockett, "Automatically Constructing a Corpus of Sentential Paraphrases," in *Proc. 3rd Int. Workshop Paraphrasing*, 2005, pp. 9–16.
- [30] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [31] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Comput. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
- [32] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
- [33] W. Li et al., "UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process. Conf.*, 2021, pp. 2592–2607.
- [34] W. Kim, B. Son, and I. Kim, "ViLT: Vision-and-language transformer without convolution or region supervision," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 5583–5594.
- [35] F. Yu et al., "ERNIE-ViL: Knowledge enhanced vision-language representations through scene graph," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 1–9.
- [36] C. Gao et al., "Room-and-object aware knowledge reasoning for remote embodied referring expression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3063–3072.
- [37] Y. Qi et al., "REVERIE: Remote embodied visual referring expression in real indoor environments," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9979–9988.
- [38] Y. Qiao et al., "HOP+: History-enhanced and order-aware pre-training for vision-and-language navigation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 8524–8537, Jul. 2023.
- [39] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
- [40] H. Tan and M. Bansal, "LXMert: Learning cross-modality encoder representations from transformers," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2020, pp. 5100–5111.
- [41] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee, "12-in-1: Multi-task vision and language representation learning," in *Proc. IEEE/CVF Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10434–10443.
- [42] X. Yang, H. Zhang, G. Qi, and J. Cai, "Causal attention for vision-language tasks," in *Proc. IEEE/CVF Comput. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9842–9852.
- [43] Z. Huang, Z. Zeng, B. Liu, D. Fu, and J. Fu, "Pixel-BERT: Aligning image pixels with text by deep multi-modal transformers," 2020, *arXiv:2004.00849*.
- [44] X. Li et al., "OSCAR: Object-semantics aligned pre-training for vision-language tasks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 121–137.
- [45] J. Lin et al., "InterBERT: Vision-and-Language Interaction for Multimodal Pretraining," 2020, *arXiv:2003.13198*.
- [46] Z. Gan et al., "Large-scale adversarial training for vision-and-language representation learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 6616–6628.
- [47] M. A. Gordon, K. Duh, and N. Andrews, "Compressing BERT: Studying the effects of weight pruning on transfer learning," 2020, *arXiv:2002.08307v2*.
- [48] J. S. McCarley, R. Chakravarti, and A. Sil, "Structured pruning of BERT-based question answering models," 2019, *arXiv:1910.06360v3*.
- [49] Z. Lin, J. Liu, Z. Yang, N. Hua, and D. Roth, "Pruning redundant mappings in transformer models via spectral-normalized identity prior," in *Proc. Findings Assoc. Comput. Linguistics*, 2020, pp. 719–730.
- [50] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Proc. 29th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.
- [51] H. Li, H. Samet, A. Kadav, I. Durdanovic, and H. P. Graf, "Pruning filters for efficient ConvNets," in *Proc. 5th Int. Conf. Learn. Representations*, 2017, pp. 1–13.
- [52] Z. Liu, Y. Wang, K. Han, S. Ma, and W. Gao, "Post-training quantization for vision transformer," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 1–12.
- [53] Z. Gao et al., "Extremely low footprint end-to-end ASR system for smart device," in *Proc. Interspeech 22th Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 4548–4552.
- [54] X. Jiao et al., "TinyBERT: Distilling BERT for natural language understanding," 2019, *arXiv:1909.10351*.
- [55] A. Vaswani et al., "Attention is all you need," in *Proc. 31th Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [56] S. Teerapittayanon, B. McDanel, and H. T. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," in *Proc. IEEE 23th Int. Conf. Pattern Recognit.*, 2016, pp. 2464–2469.
- [57] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.
- [58] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visual-BERT: A simple and performant baseline for vision and language," 2019, *arXiv:1908.03557*.



Jun Kong received the bachelor's degree in information engineering from Southwest Forestry University, Kunming, China. He is currently working toward the Ph.D. degree with the School of Information Science and Engineering, Yunnan University, Kunming. His research interests include natural language processing, text mining, and machine learning.



Jin Wang received the Ph.D. degree in computer science and engineering from Yuan Ze University, Taoyuan, Taiwan, and second Ph.D. degree in communication and information systems from Yunnan University, Kunming, China. He is currently a Professor with the School of Information Science and Engineering, Yunnan University. His research interests include natural language processing, text mining, and machine learning.



Xuejie Zhang received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 1998. He is currently a Professor with the School of Information Science and Engineering, and the Director of High-Performance Computing Center, Yunnan University, Kunming, China. His research interests include high-performance computing, cloud computing, and Big Data analytics.



Liang-Chih Yu received the Ph.D. degree in computer science and information engineering from National Cheng Kung University, Tainan City, Taiwan. From 2007 to 2008, he was a Visiting Scholar with Natural Language Group, Information Sciences Institute, the University of Southern California, Los Angeles, CA, USA, and with DOCOMO Innovations for three months in 2018. He is currently a Professor with the Department of Information Management, Yuan Ze University, Taoyuan, Taiwan. His research interests include natural language processing, sentiment

analysis, and computer-assisted language learning. He is a Board Member and Convener of SIGCALL of the Association for Computational Linguistics and Chinese Language Processing. He is also an Editorial Board Member of *International Journal of Computational Linguistics and Chinese Language Processing*. His team has developed systems that ranked first in IJCNLP 2017 Task 4: Customer Feedback Analysis, and second in SemEval and BEA shared task competitions.