UniPT: Universal Parallel Tuning for Transfer Learning with Efficient Parameter and Memory

Haiwen Diao[1], Bo Wan[2], Ying Zhang[3], Xu Jia[1], Huchuan Lu[*1], Long Chen[4]
[1]Dalian University of Technology [2]KU Leuven [3]Tencent WeChat [4]HKUST
diaohw@mail.dlut.edu.cn; bwan@esat.kuleuven.be; yinggzhang@tencent.com;
xjia@dlut.edu.cn; lhchuan@dlut.edu.cn; longchen@ust.hk

Abstract

*Parameter-efficient transfer learning (PETL), i.e., fine-tuning a small portion of parameters, is an effective strategy for adapting pre-trained models to downstream domains. To further reduce the memory demand, recent PETL works focus on the more valuable memory-efficient characteristic. In this paper, we argue that the scalability, adaptability, and generalizability of state-of-the-art methods are hindered by structural dependency and pertinency on specific pre-trained backbones. To this end, we propose a new memory-efficient PETL strategy, Universal Parallel Tuning (**UniPT**), to mitigate these weaknesses. Specifically, we facilitate the transfer process via a lightweight and learnable parallel network, which consists of: 1) A parallel interaction module that decouples the sequential connections and processes the intermediate activations detachedly from the pre-trained network. 2) A confidence aggregation module that learns optimal strategies adaptively for integrating cross-layer features. We evaluate UniPT with different backbones (e.g., T5 [69], VSE∞ [12], CLIP4Clip [58], Clip-ViL [73], and MDETR [42]) on various vision-and-language and pure NLP tasks. Extensive ablations on **18** datasets have validated that UniPT can not only dramatically reduce memory consumption and outperform the best competitor, but also achieve competitive performance over other plain PETL methods with lower training memory overhead. Our code is publicly available at: https://github.com/Paranioar/UniPT.*

1. Introduction

Large-scale deep neural networks [5, 17, 21, 68] trained on massive data have been successfully investigated in various vision and multimodal learning tasks [2, 18, 19, 26, 75, 87]. The most prevalent and straightforward strategy for transferring the knowledge from pre-trained models to downstream tasks is fully fine-tuning [12, 42, 58, 73]. However, fine-tuning the entire network is prohibitively expensive, especially given a large model with millions of parameters. Meanwhile, it easily suffers from the over-fitting problem with a relatively "small" downstream dataset. To address it, there is an increasing interest in *parameter-efficient transfer learning* (PETL) [3, 34, 49, 65], which facilitates domain adaptation by adjusting or inserting a few modules.

Currently, mainstream state-of-the-art PETL approaches can be coarsely grouped into three categories: (a) **Partially Tuning** [44, 90]: It only updates a few task-specific parameters and freezes most original parameters, such as modifies the bias items [6, 90] or the normalization layers [44]. (b) **Adapter Tuning** [24, 32, 34, 59]: It usually inserts a new bottleneck-shaped module after each backbone layer, which is the only part that needs to be updated. Typically, the new module consists of a linear down-projection, a non-linearity activation, and a linear up-projection. (c) **Prompt Tuning** [27, 39, 49, 95]: It first integrates a fixed number of learnable vectors as additional input tokens (*i.e.*, prompts). Then, it only learns the prompts and freezes all the raw parameters of the pre-trained network during the fine-tuning stage. Although all the above PETL methods can dramatically reduce the trainable parameters and storage constraints, their memory consumption remains costly during the training stage. As shown in Figure 1(a-c), the backward gradients still need to go through (nearly the entire) foundation models. This memory-extensive characteristic severely limits their applications in resource-constrained scenarios.

Consequently, a few recent works [6, 22, 54, 77, 80, 91] emphasize the requirements of both *parameter* and *memory* efficiency during the training process. In particular, the most successful method that achieves a good balance between performance and efficiency is Ladder Side Tuning (**LST**) [77]. As shown in Figure 1(d), it first constructs a lightweight side network by keeping the same structure as the pre-trained network but reducing the dimension of each original layer by a predefined reduction factor. Then, it learns a static gated ladder connection to combine the pair-wise features at each layer between the side and pre-trained network.

*Corresponding author. Work was done when Haiwen visited HKUST.

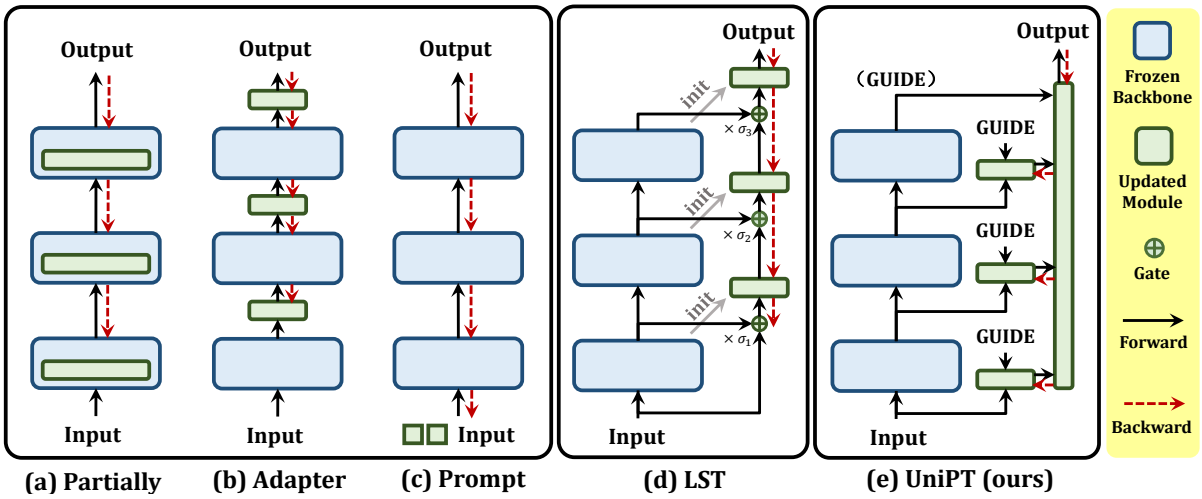

Figure 1. Overview of recent PETL methods including *Partially Tuning*, *Adapter Tuning*, *Prompt Tuning*, and *Ladder Side Tuning (LST)*.

Nevertheless, we argue that these designs have several potential drawbacks: 1) *Scalability*: The model complexity of the side network goes linearly proportional to the original pre-trained network, making its efficiency susceptible to the original architecture, *i.e.*, the larger the pre-trained network, the less efficient the side network. 2) *Adaptability*: The static gate mechanism simply sums up the outputs of each pre-trained and its corresponding side layer. It overlooks the latent semantic misalignments of pair-wise features between them, and also neglects to dynamically equip different samples with the most suitable aggregation strategy. 3) *Generalizability*: Most of the above strategies are primarily suitable for the Transformer-family. For instance, in other prevalent neural networks (*e.g.*, CNNs), the ladder gate connection in LST cannot directly handle the discrepancy of spatial-wise and channel-wise dimensions between cross-layer features. Therefore, how to extend these PETL methods to various architectures is still under-explored.

Based on such considerations, in this paper, we propose a new memory-efficient PETL strategy, dubbed Universal Parallel Tuning (**UniPT**). As shown in Figure 1(e), we facilitate the transfer process via a lightweight learnable parallel network, whose structures are independent of the backbone (*Scalablity*) and could work across various architectures, such as Transformers, CNNs and Encoder-Decoder structures (*Generalizability*). Specifically, our UniPT consists of two modules: 1) A parallel interaction module decouples the inherently sequential connections across layers and handles each layer's intermediate detachedly. It treats the feature outputs across layers equally and highlights more discriminative representations inside each layer. 2) A confidence aggregation module learns adaptively optimal strategies for integrating cross-layer features based on different input embeddings and model structures. It requires no manual tuning while staying effective and efficient (*Adaptability*).

To fully evaluate the generalization capability, we have made comprehensive investigations on both challenging cross-modal and widely-used uni-modal domains by typically involving more diverse network architectures than previous works. Extensive results on vision-and-language tasks (*i.e.*, *image-text retrieval* [51, 88], *video-text retrieval* [9, 87], *visual question answering* [26], *compositional question answering* [37], and *visual grounding* [63, 89]) and one GLUE benchmark have validated the effectiveness of our UniPT, *i.e.*, it achieves the best balance and trade-off between performance and parameter/memory efficiency.

2. Related Work

Parameter-Efficient Transfer Learning (PETL). Recently, large pre-trained models have sprung up in computer vision (CV) [7, 21, 31, 55], natural language processing (NLP) [17, 53, 69], and vision-and-language (VL) fields [42, 47, 48, 68]. To efficiently transfer pre-trained knowledge, PETL methods have drawn much research attention and become a promising direction. Earlier PETL studies [44, 90] attempted to update a few pre-trained network parameters during fine-tuning including the bias parameters [6, 90] and layer normalization layers [44]. However, they are not always applicable, *e.g.*, the pre-trained T5 model [69] does not have any bias items. Hence, some works [28, 76] tried to control which components inside the large network to be updated or fixed during training by learning sparse binary marks against the pre-trained weights. Nevertheless, the resulting performance is extremely sensitive to the sparsity of the binary mask.

In contrast, another line of research introduced an extra lightweight learnable subnetwork (Adapter) that is integrated into the original network layers and arouses a surge of interest in NLP [3, 32, 65, 92], CV [13, 16, 24, 50], and VL areas [10, 35, 56, 66, 78]. They typically injected a tiny bot-

tleneck module in parallel or after each multi-head attention and feed-forward layer, and kept the rest of the pre-trained network frozen. Particularly, some works [29, 34, 52] inserted a trainable vector or low-rank matrix into multi-head attention to influence the query and value projection, while other methods [40, 59, 60] implemented matrix decomposition, low-rank parameterization, and hyper-network generation for the adapter weights respectively to further reduce the number of parameters that need to be trained. Concurrently, another popular technology namely Prompt Tuning [27, 46, 49, 96] in NLP validated that simply prepending several learnable tokens into the input sequence of each Transformer layer can also achieve competitive performance during fine-tuning, which further triggers a series of follow-up explorations in CV [11, 23, 39, 43, 71, 94] and VL fields [36, 41, 57, 79, 83, 93]. The most critical issue is that the optimal prompt module requiring elaborate manual tuning and designs is extremely challenging for specific downstream datasets. Although PETL methods significantly decreased the trainable parameters and produced remarkable results on the downstream task, they suffer from expensive computational memory consumption, making them inapplicable to resource-constrained scenarios during training.

Memory-Efficient Transfer Learning. Current PETL methods investigate the ways to achieve competitive performance with as few trainable parameters as possible. Nevertheless, the training memory is dominated by activations, not parameters, which means that parameter efficiency is not equivalent to memory efficiency. Hence, some approaches [15, 33, 68, 69] fixed the pre-trained backbone and only updated the last layers for domain transfer, which, though memory-efficient, have limited model capacity and lag far behind the results by fully fine-tuning. Hence, Side-Tuning [91] adopted a cheap side network, whose outputs are combined with the backbone outputs with a curriculum schedule for continuous task adaptation. Besides, Y-tuning [54] exhausted all possible labels and fed their dense representations via a side feature integration for the final label selection. To address intractable answer collection and focus more on memory reduction, LST [77] proposed a small and separate network that receives intermediate activations from backbone networks via ladder gated connections and makes predictions. As mentioned before, our UniPT outperforms their solutions in terms of Scalability, Adaptability, and Generalizability, which displays more powerful capability and broad applicability over various model architectures in multiple VL tasks. Actually, there are some popular and generalized strategies [45, 62] to alleviate the training memory footprint. Some of them avoid saving all the intermediate activations and re-compute discarded ones during backward via reconstruction from the backward layers [25] or gradient checkpoint operation [14], while others [82] developed new reduced-precision floating-point formats to decrease the bitwidth of training activations. Note that our UniPT is orthogonal to these techniques and can be combined to further pursue a higher level of memory efficiency.

3. Methodology

We propose a novel Universal Parallel Tuning (UniPT) for memory-efficient transfer learning. In Section 3.1, we explain the detailed architecture and implementation of UniPT including a parallel interaction layer and a confidence aggregation layer. Then, we demonstrate its applications to CNN, Transformer, and Encoder-Decoder network in Section 3.2.

3.1. Universal Parallel Tuning

As illustrated in Figure 2, we construct a tiny and parallel network on top of the pre-trained network that could transform all the intermediate layer features into the final representations for new domains. It does not require costly backward gradients through the large pre-trained backbone and vastly lessens memory overhead during the training process.

Parallel Interaction Layer. We build parallel interaction layers that are lightweight and independent of the pre-trained network. In Figure 2(a), they take each layer's intermediate representation from the pre-trained backbone as input and handle the feature extraction of each layer detachedly.

Existing memory-efficient PETL methods [54, 77, 91] implement self-attention inside scaled-down side blocks and incorporate hierarchical features in a bottom-up manner. However, direct lateral connection might neglect the potential semantic inconsistency between side and pre-trained backbones. More significantly, the weakened interaction would dilute the discriminative representations within each layer necessary for ultimate adaptation. Inspired by this, we conduct a top-down interaction process, leveraging ultimate outputs from the pre-trained network as interaction guidance. This choice stems from the superior representation and transfer capability exhibited by the original last layer outputs in new domains. By utilizing them as uniform guidance for other layers before hierarchical aggregation, it could ensure semantic coherence across different layers and capture the essential features inside each layer independently, thereby in turn enhancing the ultimate representations accordingly.

Concretely, we first map all K hidden features of all N layers to a unified dimension $d = D/r$ via a reduction factor r, where D is the dimension of the original backbone output. We empirically observe a significant disparity among feature norms across different layers within various backbones. Computation in the standard attention layer would suffer numerical issues and unstable gradients during backpropagation. Hence, after obtaining all the layer features $\boldsymbol{F} = \{\boldsymbol{F}_i \in \mathbb{R}^{K\times d} \,|\, i \in \{0, 1, ..., N\}\}$, we first compute the inner product matrices between last layer $\boldsymbol{F}_N$ (Query) and other layers $\boldsymbol{F}_{\setminus N} = \{\boldsymbol{F}_i \in \mathbb{R}^{K\times d} \,|\, i \in$

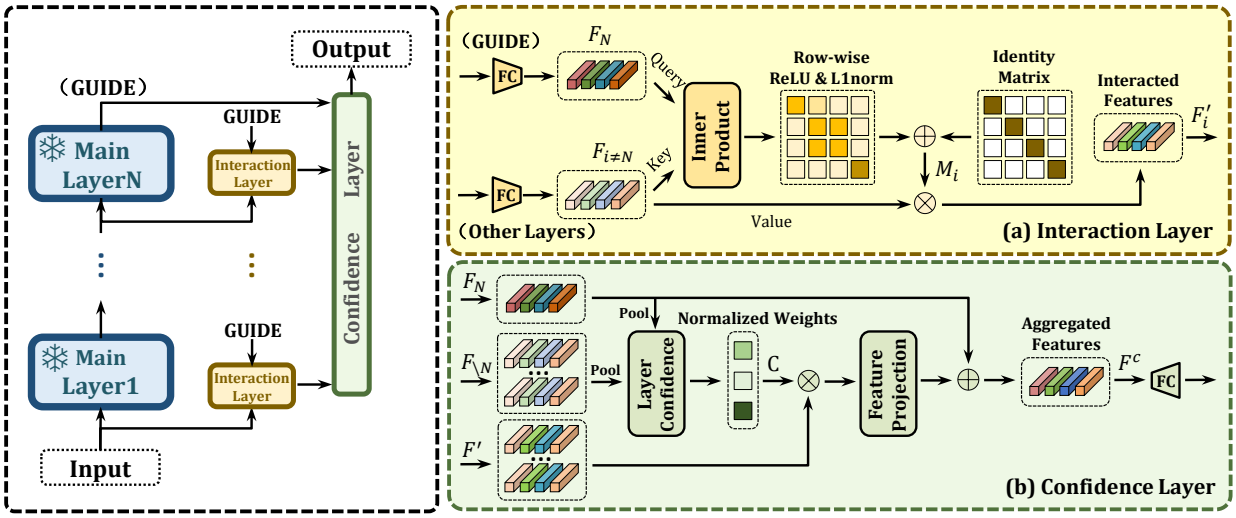

Figure 2. Overview of the Framework with (a) parallel interaction and (b) confidence aggregation layers. The former attempts to extract more discriminative features at each layer independently guided by the relatively most powerful output features, while the latter learns a dynamic and optimal combination strategy over the blended features at each layer for the ultimate domain adaptation.

$\{0, 1, ..., N-1\}\}$ (Key). Then, we adopt ReLU activation (σ) and L1 normalization ($\texttt{L1Norm}$) to eliminate all the negative connections and generate the normalized attention weights between a Query and all Key features. Considering that these weights occasionally are all zeros, we add an extra identity matrix bias (*i.e.* residual connection with original layer features) to obtain the final attention weights $\boldsymbol{M} = \{\boldsymbol{M}_i \in \mathbb{R}^{K\times K} \,|\, i \in \{0, 1, ..., N-1\}\}$. Lastly, the blended features $\boldsymbol{F}' = \{\boldsymbol{F}'_i \in \mathbb{R}^{K\times d} \,|\, i \in \{0, 1, ..., N-1\}\}$ are calculated as follows:

$$\boldsymbol{M}_i = \texttt{L1Norm}_{\boldsymbol{F}_i}\sigma(\boldsymbol{F}_N\boldsymbol{F}_i^\top) + \boldsymbol{I}\,, \quad \boldsymbol{F}'_i = \boldsymbol{M}_i\boldsymbol{F}_i\,, \quad (1)$$

where $\boldsymbol{F}'_i$ denotes K merging features in the i-th layer, each of which corresponds the one in $\boldsymbol{F}_{\setminus N}$.

Confidence Aggregation Layer. The sequential ladder-gated connection [77] as static fusion strategy fails to dynamically adapt to different inputs and suffers from exponential attenuation of earlier layer features. Therefore, we explore an adaptive confidence module that treats the merging features from each layer equally and automatically learns the optimal aggregation strategies for various inputs and model structures. In this way, it can highlight more discriminative features and discard less informative ones from the multi-granularity layers for better domain adaptation. Given the blend features $\boldsymbol{F}'$ and $\boldsymbol{F}$, we first obtain the holistic representations of each layer $\boldsymbol{F}^g = \{\boldsymbol{F}^g_i \in \mathbb{R}^{1\times d} \,|\, i \in \{0, 1, ..., N\}\}$ by averaging all the features $\boldsymbol{F}$. Comparing the relations between last layer $\boldsymbol{F}^g_N$ and other layers $\boldsymbol{F}^g_{\setminus N}$, we obtain the normalized confidence weights $\boldsymbol{C}$ of the blended features $\boldsymbol{F}'$ via a fully-connected (FC) layer and a softmax function. Once all the blended features are merged, they are converted into a shared space as $\boldsymbol{F}_N$ via $\texttt{MLP}(\boldsymbol{X}) = \sigma(\boldsymbol{X}\boldsymbol{W}_1)\boldsymbol{W}_2$, where $\boldsymbol{W}_1 \in \mathbb{R}^{d\times D}, \boldsymbol{W}_2 \in \mathbb{R}^{D\times d}$ as follows:

$$\begin{aligned} \boldsymbol{C} &= \texttt{Softmax}_{\boldsymbol{F}^g_{\setminus N}}((\boldsymbol{F}^g_N \odot \boldsymbol{F}^g_{\setminus N})\boldsymbol{W}_3)\,, \\ \boldsymbol{F}^c &= \texttt{MLP}(\textstyle\sum_i \boldsymbol{C_i} \cdot \boldsymbol{F}'_i) + \boldsymbol{F}_N\,. \end{aligned} \quad (2)$$

Note that $\boldsymbol{W}_3 \in \mathbb{R}^{d\times 1}$ and $\odot$ denotes element-wise multiplication. Eventually, we sum up them and the original output $\boldsymbol{F}_N$ as $\boldsymbol{F}^c \in \mathbb{R}^{K\times d}$, which are upsampled by a new FC layer to produce the final outputs. Note that the dimension of the UniPT output is consistent with that of the raw pre-trained backbone for the new task domains.

3.2. Diverse Backbone Application

As illustrated in Figure 3, our proposed method has broad applications over various frameworks ϕ, including CNN, Transformer, and Encoder-Decoder architectures. The proposed UniPT consists of two main parts: parallel interaction layer φ and confidence aggregation layer θ.

Application on Transformer. In Figure 3(a), we demonstrate that the UniPT can be seamlessly integrated into existing Transformer architectures. Thereinto, for mono-modality encoders like BERT [17] or ViT [21] with text or image as input, we extract the word or patch embeddings, all the hidden states, and the original last state, which are then fed to the respective $\varphi^T_{1:N}$ layers and integrated into the ultimate feature output by the θ^T layer. For cross-modality correspondence like CLIP-ViL [73], the image and text features are first mapped into the same dimensions and concatenated into the follow-up cross-Transformer $\phi_{1:N}$ layers. Analogously maintaining the same numbers and dimensions as the pre-trained Transformer output, the final feature representation is enhanced via the parallel $\varphi^T_{1:N}$ and θ^T layers.

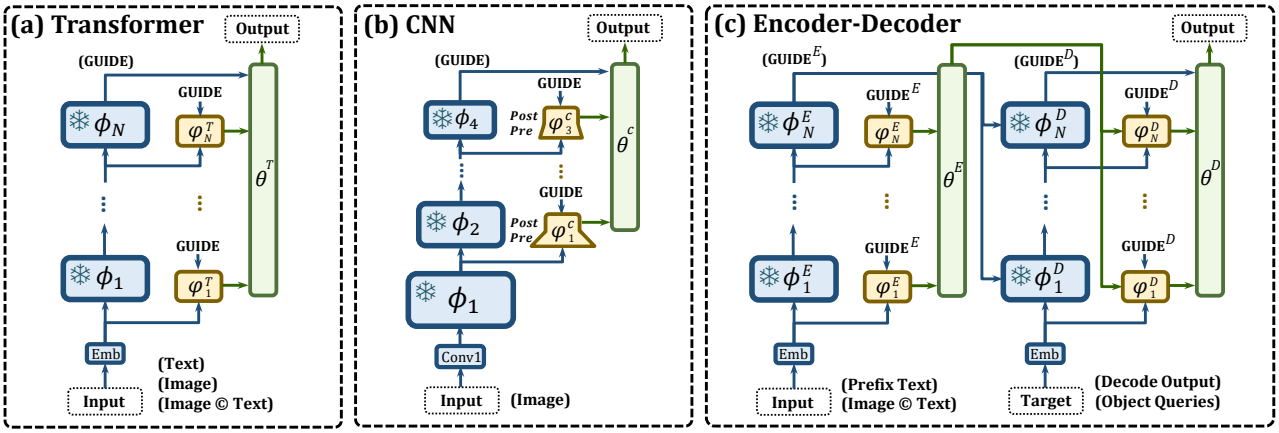

Figure 3. Overview of the Application with UniPT network (φ, θ) over (a) Transformer, (b) CNN, and (c) Encoder-Decoder (ϕ) architectures.

Application on CNN. For CNN (*e.g.* ResNeXt [86]), it processes image inputs with a 2-D convolution kernel where each layer has varied block numbers and structures, as shown in Figure 3(b). More importantly, the spatial sizes and channel dimensions of intermediate feature maps across the layers are different and doubled in value. Therefore for cross-layer feature maps, it is impossible to perform one-to-one gated addition like LST, but it does not interfere with our UniPT owing to the separate processing. Note that directly employing interaction layer $\varphi^C_{1:3}$ on the substantial shallow features would inevitably increase the computational cost. To further improve memory efficiency, we decompose the standard interaction layer as pre-interaction and post-interaction. Specifically, we start by dividing the shallow feature map from each layer into multiple non-overlapping chunks, of which the number aligns with the size of the output feature map. Given that the representations within each chunk are most relevant to the corresponding positional feature within the last feature map due to the relatively limited receptive field, we thereby implement the interaction process inside a confined region between the last feature and its corresponding shallow chunk, termed as pre-interaction. With the feature maps from Conv2-5 $\phi_{1:4}$ layers, we totally obtain four spatial-reduced feature representations with the same spatial resolution, that then pass the same procedure as the usage in Transformer, denoted as post-interaction.

Application on Encoder-Decoder. As an extension of the conventional Transformer, the pre-trained encoder-decoder model in Figure 3(c) serves as the particular backbone for auto-regressive tasks (*e.g.*, MDETR [42] for multi-modal detection and T5 [69] for GLUE benchmark). For the Encoder, we implement the Encoder UniPT (φ^E, θ^E) with both image and text features in the same way that it does in the Transformer, while for the Decoder, the inputs for the Decoder UniPT (φ^D, θ^D) includes an additional output of Encoder UniPT module. Inspired by the original backbone workflow, we also decompose the whole interaction layer φ^D as pre-interaction and post-interaction as well. Concretely, we first perform the pre-interaction between the Decoder intermediate features and the Encoder UniPT output to enhance the feature representations of each Decoder layer, and then accomplish the post-interaction under the guidance of the original Decoder output for the final domain adaptation.

4. Experiments

4.1. Setups

Datasets. We evaluate our proposed UniPT on VL and NLP tasks. Specifically, the VL tasks cover image-text retrieval (**ITR**: MSCOCO [51], Flickr30K [88]), video-text retrieval (**VTR**: MSR-VTT [87], MSVD [9]), question answering (**VQA&GQA**: VQAv2 [26], GQA [37]), and visual grounding (**VG**: RefCOCO, RefCOCO+ [89], RefCOCOg [63]). For ITR and VTR, we report Recall@1 (R@1) on sentence retrieval (I-T, V-T), image retrieval (T-I), and video retrieval (T-V), and Rsum of R@1,5,10 in two directions for comprehensive verification. For VQA and GQA, we compare the results on Test-Dev and Test-Std sets, while for VG, we report the performance on the images containing multiple people (TestA) and multiple instances of all other objects (TestB). The GLUE benchmark [81] consists of linguistic acceptability (CoLA [84]), sentiment analysis (SST2 [74]), similarity and paraphrase (MRPC [20], QQP [38], STS-B [8]), and natural language inference (MNLI [85], QNLI [70], RTE [4]).

Training Details. All experiments are conducted using eight GeForce RTX 3090Ti (24GB) and keep most default configurations of the pre-trained models, *e.g.*, choice of optimizer, warm-up schedule, input image resolution, video sequence length, input text processing, *etc*. Specially, we follow their original batch sizes for most tasks, with the exception of maximum batch size=112 (vs. raw 128) for VSE∞ in ITR, due to the out-of-memory (OOM) issue. The reduction factors r of LST/UniPT are set to 8 and 4/2 for

Method	Params. (M)	Memory (G)	BERT I-T	BERT T-I	BERT Rsum	Params. (M)	Memory (G)	ResNeXt-101 I-T	ResNeXt-101 T-I	ResNeXt-101 Rsum	Params. (M)	Memory (G)	ViT + Text Transf. T-V	ViT + Text Transf. V-T	ViT + Text Transf. Rsum
Fully-FT	109.5	9.9	79.7	62.1	513.5	90.9	21.8 * 8	85.6	70.2	539.0	151.3	12.2 * 4	42.8	42.1	389.2
Adapter [32]	2.6	8.8	79.1	60.5	511.3	3.5	22.1 * 8	66.8	62.9	493.3	5.2	10.3 * 4	38.3	39.6	364.3
LoRA [34]	1.1	8.8	78.8	59.6	508.2	–	–	–	–	–	1.3	10.2 * 4	38.8	39.9	366.8
BitFit [90]	0.9	8.6	77.3	57.8	503.9	2.2	21.0 * 8	83.4	67.4	530.6	0.1	10.5 * 4	38.1	40.6	370.8
Prompt [49]	10.7	9.4	78.7	59.0	508.5	–	–	–	–	–	0.2	10.7 * 4	36.8	37.5	358.8
SSF [50]	0.2	8.4	80.0	60.4	512.8	0.1	20.4 * 8	83.7	66.8	528.5	0.5	9.8 * 4	40.2	41.8	376.6
FacT [40]	0.6	8.7	79.2	59.3	508.8	–	–	–	–	–	0.8	10.2 * 4	38.7	39.8	367.2
AdaLoRA [92]	1.0	8.8	79.8	60.1	510.3	–	–	–	–	–	1.2	10.5 * 4	39.2	39.6	368.5
Partially⇓	0.8	1.0	74.8	53.7	485.5	2.1	14.9	75.2	58.2	505.8	0.7	1.9 * 4	36.4	37.0	353.9
LST [77]	7.5	4.6	77.9	57.3	501.9	2.27	15.0	82.3	66.1	526.7	11.2	8.0 * 4	37.0	37.8	356.7
UniPT	5.9	3.1	**80.2**	**59.8**	**510.5**	6.4	15.0	**84.0**	**67.7**	**532.1**	9.6	3.4 * 4	**38.9**	**39.3**	**361.3**

Table 1. Comparisons with popular parameter/memory-efficient methods on Flickr30K using *VSE*∞ with CNN (ResNeXt-101) or single Transformer (BERT), and MSR-VTT using *CLIP4Clip* with dual Transformer encoders (ViT + Text Transformer). We report Recall@1 (R@1) on sentence retrieval ("I-T", "V-T"), image retrieval ("T-I"), video retrieval ("T-V"), and "Rsum" of R@1,5,10 on bi-directional retrievals.

Method	Update Params. per Task (%)	Memory (G) Train	Memory (G) Test	CoLA	SST-2	MRPC	QQP	MNLI	QNLI	RTE	STS-B	Avg.
Fully-FT	100	17.6	0.86	62.8	93.9	91.9	89.9	86.2	92.5	74.1	90.3	85.2
Adapter [32]	1.63	13.0	0.87	64.4	94.2	88.9	88.9	86.4	93.1	75.1	91.1	85.3
LoRA [34]	1.71	12.6	0.86	63.3	94.3	90.1	89.0	86.3	93.2	75.5	90.9	85.3
BitFit [90]	0.13	10.7	0.86	61.8	94.3	91.0	88.7	85.6	93.1	67.6	90.8	84.1
Prompt [49]	0.03	22.2	0.87	0	90.3	74.6	88.5	82.5	92.5	59.5	90.1	72.2
LST [77]	1.74	5.5	0.88	58.1	94.1	90.4	88.8	**85.6**	**93.3**	**71.9**	**90.7**	84.1
UniPT	1.36	2.9	0.86	**62.2**	**94.2**	**90.8**	**88.9**	85.5	**93.3**	69.8	89.7	**84.3**
LST [77] (T5-large)	1.23	12.2	2.88	65.3	95.7	91.6	**89.7**	**88.6**	94.1	**79.9**	**92.4**	87.1
UniPT (T5-large)	0.92	9.1	2.82	**65.7**	**95.8**	**92.0**	**89.7**	88.2	**94.2**	79.6	92.0	**87.2**

Table 2. Comparisons with parameter/memory-efficient methods on GLUE benchmark using encoder-decoder *T5*. Following LST, we adopt *T5-base* as baseline and larger *T5-large* for further comparison. We report accuracy for SST-2, MNLI, QNLI and RTE. For CoLA and STS-B, we use Matthew's Correlation and Pearson-Spearman Correlation respectively. For MRPC and QQP, we average the F1 score and accuracy.

NLP and VL tasks respectively, unless otherwise noted. We also maintain the original learning rate lr and the number of epochs for fully fine-tuning, and search learning rates over $\{20 \times lr, 10 \times lr, lr\}$ for PETL methods. We implement parameter- and memory-efficient methods through the official GitHub source of LST [77] and AdapterHub library [64].

4.2. Main Results

4.2.1 UniPT vs. Non-memory-efficient PETL

Model Settings. In Tables 1 and 2, we compare our UniPT against Fully Fine-Tuning and several popular PETL methods. In *Adapter* [32], we insert tiny trainable modules into every attention and feed-forward layer of the Transformer and conv2-5 layers of CNN, while in *BitFit* [90], we only update the bias terms for the pre-trained models. Besides, *LoRA* [34] and *Prompt* [49] introduce additional trainable low-rank matrices and learnable token inputs into the attention layer of the Transformer that are not compatible with CNN. We reproduce *LST* [77] by extracting the ladder blocks from the pre-trained networks, but for CNN, we abandon the corresponding conv1 layer in the side network to address the size misalignment across layers. In Fully Fine-Tuning (*Fully-FT*), all network parameters are updated, while in Partially Tuning (*Partially*⇓), we only update the last projection and aggregation modules after the pre-trained network.

Results on VL Tasks. In Table 1, we select various transfer paradigms for diverse and challenging validation. To identify the impact on pre-trained CNN, single Transformer, and dual Transformer encoders, we adopt two structures from *VSE*∞ on ITR and one construction from *CLIP4Clip* on VTR:

- ResNeXt-101(32×8d) [86] + BiGRU [72] (tiny);
- BERT-base [17] + BUTD regions [1] (tiny);
- ViT-base [21] + Text Transformer [68].

We can discover that our UniPT achieves competitive results with recent state-of-the-art PETL works including SSF [50], FacT [40], and AdaLoRA [92], and even outperforms some popular PETL methods in a low-memory regime. Note that *(1) Adapter displays a difficult optimization situation on CNN and achieves sub-optimal performance after careful adjustments.* We assume that this may be due to the CNN's deeper network structure compared with the Transformer,

Method	Params.	Memory (G)		Flickr30K			MSCOCO1K			MSCOCO5K			Params.	Memory (G)		MSR-VTT			MSVD		
	(M)	Train	Test	I-T	T-I	Rsum	I-T	T-I	Rsum	I-T	T-I	Rsum	(M)	Train	Test	T-V	V-T	Rsum	T-V	V-T	Rsum
Fully-FT	201.2	22.1 * 8	20.12	85.6	73.3	546.6	83.1	71.7	542.7	64.2	51.2	468.9	151.3	12.2 * 4	1.12	42.8	42.1	389.2	45.2	57.1	425.5
Partially⇓	2.9	15.0	20.12	75.6	59.8	508.3	74.6	59.5	510.5	51.2	37.6	401.7	0.7	1.9 * 4	1.12	36.4	37.0	353.9	37.4	52.4	406.4
LST [77]	9.7	15.1	20.31	82.1	66.5	529.5	78.2	64.8	525.8	57.8	43.1	434.5	11.2	8.0 * 4	1.15	37.0	37.8	356.7	35.5	55.4	407.2
UniPT	12.4	15.1	20.19	**84.8**	**69.1**	**537.4**	**80.6**	**67.5**	**532.9**	**61.1**	**45.9**	**445.3**	9.6	3.4 * 4	1.13	**38.9**	**39.3**	**361.3**	**40.9**	**59.7**	**432.1**

Method	Params.	Memory (G)		VQAv2		GQA		Params.	Memory (G)		RefCOCO			RefCOCO+			RefCOCOg	
	(M)	Train	Test	Test_{Dev}	Test_{Std}	Test_{Dev}	Test_{Std}	(M)	Train	Test	Val	TestA	TestB	Val	TestA	TestB	Val	Test
Fully-FT	236.8	20.5 * 4	12.64	76.71	76.86	60.25	61.44	185.2	19.8 * 2	3.36	86.51	89.13	81.22	79.54	84.54	70.63	80.92	80.95
Partially⇑	117.6	10.5 * 4	12.64	76.73	76.84	61.13	62.28	18.3	11.3 * 2	3.36	85.39	88.40	79.78	77.66	84.00	69.38	79.92	80.08
LST [77]	13.4	6.4 * 4	12.76	75.29	75.44	59.93	**60.75**	0.9	6.3 * 2	3.41	81.63	85.19	76.03	71.32	78.20	62.06	72.53	73.67
UniPT	10.3	2.9 * 4	12.67	**75.33**	**75.53**	**60.10**	60.72	0.7	3.4 * 2	3.38	**82.71**	**86.25**	**78.16**	**72.94**	**79.18**	**64.49**	**77.04**	**77.33**

Table 3. Comparisons with the best memory-efficient counterpart LST on VL tasks with various architectures. We adopt *VSE*∞ with both CNN and Transformer (ResNeXt-101 + BERT-base) on ITR task, *CLIP4Clip* with dual Transformer encoders (ViT-base + Text Transformer) on VTR task, *CLIP-ViL* with Cross-modal Transformer on VQA and GQA tasks, and *MDETR* with Encoder-Decoder architecture on VG task.

leading to the vanishing gradient problem of the Adapter inside the shallow layers. *(2) For CNN, the two-stage interaction of UniPT leads to a modest increase in trainable parameters, and the marginal memory gains are primarily a result of reaching memory-saving saturation.* That is because the forward process of the large backbone takes up the main memory costs, where the simplest transfer method Partially⇓ still occupies the memory overhead by 14.9G.

Results on NLP Benchmark. Table 2 demonstrates the comparisons on GLUE benchmark. Based on the *T5-base*, UniPT significantly reduces memory overhead from 17.6G to 2.9G with similar trainable parameter usage, and achieves competitive performance as full fine-tuning and other PETL works. To further utilize the memory-efficient advantage of UniPT, we cooperate it with *T5-large* and find that even with lower memory budget in Adapter, LoRA, and BitFit, UniPT with can surpass other PETL methods by a large margin. *Limited by current device, we would accommodate the training of UniPT on the T5-3B model in the future.*

4.2.2 UniPT vs. Memory-efficient PETL

Model Settings. In Table 3, we compare our UniPT against the best memory-efficient competitor LST. For *VSE*∞ [12] on ITR task and *CLIP4Clip* [58] on VTR task, we freeze their dual encoders and only update the last small aggregation modules during training as *Partially*⇓, which serves as the lowest performance bound for LST and UniPT. Besides for *CLIP-ViL* [73] on QA task and *MDETR* [42] on VG task, we freeze the vision backbone except for *Fully-FT*, because the performance and efficiency gains of fine-tuning it over keeping it frozen are both limited [77, 78]. Meanwhile, we unfreeze their respective cross-modal or encoder-decoder Transformer during fine-tuning as *Partially*⇑, which thereby represents the upper performance bound for LST and UniPT.

Results on VL Tasks. Table 3 shows the comparisons on diverse VL tasks with various pre-trained architectures:

- ITR task: *VSE*∞ [12] with the strongest combination of BERT-base [17] model and ResNeXt-101(32×8d) [86] backbone pre-trained on Instagram (WSL) [61];
- VTR task: *CLIP4Clip* [58] with pre-trained CLIP [68] using Text Transformer [67] and ViT-B/32 [21] models;
- QA task: *CLIP-ViL* [73] that utilizes CLIP image backbone [68] and encodes the text into word embedding sequence, followed by a cross-modal Transformer;
- VG task: *MDETR* [42] with pre-trained ResNet-101 [30], RoBERTa-base [53], and encoder-decoder Transformer.

(1) Larger performance gains. Our UniPT outweighs the best LST on various tasks and backbones. In particular, for the larger and more compelling retrieval datasets *i.e.* MSCOCO5K and MSR-VTT, our UniPT achieves absolute R@1 improvements than LST by 3.3/2.8% and 1.9/1.5%, validating the superior applicability in handling more challenging matching patterns. Besides, on the much smaller MSVD dataset, our UniPT obtains much better Rsum than Fully-FT (432.1 vs. 425.5%), indicating the anti-overfitting capability under the limited data-driven scenario. *(2) Lower training memory usage.* UniPT achieves nearly twice the training memory savings of LST in most cases, excluding the *VSE*∞ in ITR and *T5-large* in GLUE. *(3) Negligible inference memory costs.* UniPT requires less extra memory consumption during the inference process by an average of 0.02G vs. 0.06G of LST with various networks on all multi-modal tasks. In summary, our UniPT outperforms the leading counterpart LST by achieving the optimal trade-off in both training efficiency and transfer capability.

4.3. Ablation Studies

Necessity of Interaction Guidance and Dynamic Fusion. ***Settings.*** Figure 4 compares several UniPT variations using *VSE*∞ on Flickr30K. **(1)** For interaction, we test Adapter-like network (Bottleneck) and self-attention between intermediates inside each layer (w/o Guidance) as counterparts. **(2)** For aggregation, we utilize average pooling (Pooling)

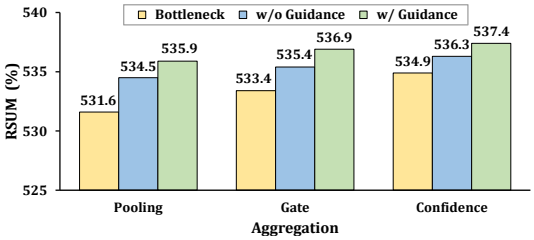

Figure 4. Rsum (%) on Flickr30K by *VSE*∞ w/ or w/o interaction guidance and dynamic aggregation strategies.

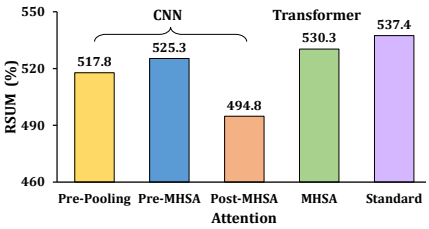

Figure 5. Rsum (%) on Flickr30K by *VSE*∞ w/ our truncated or other alternative attention.

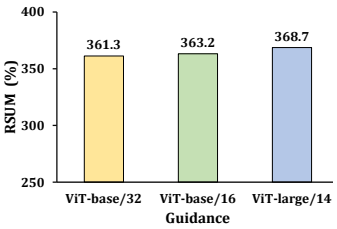

Figure 6. Rsum (%) on MSR-VTT by *CLIP4Clip* w/ stronger guidance.

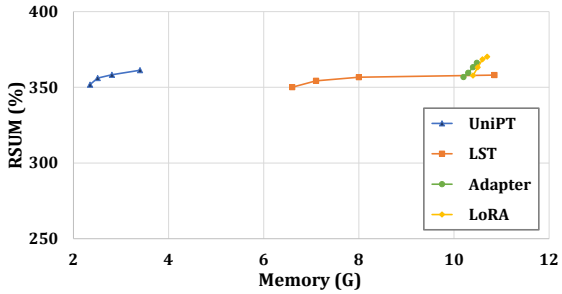

Figure 7. Accuracy-memory trade-off on MSR-VTT by *CLIP4Clip* for several PETL approaches w/ reduction factor $r \in \{8, 4, 2, 1\}$.

and static ladder-gated connection (Gate) for comparison.

Results. In Figure 4, we discover that interaction w/ Guidance obtains more promising performance than the ones w/o Guidance, indicating that guidance from the last layer output could enhance hidden features inside each layer with more powerful adaptation capability. Besides, dynamic confidence also suppresses other strategies by adaptively adjusting the proportion of each layer over various inputs and producing more discriminative representations for domain transfer.

Superiority of UniPT over Multi-head Self-attention.

Settings. Figure 5 reports the comparisons between our truncated attention (Standard) and multi-head self-attention (MHSA) using *VSE*∞ on Flickr30K. We set the head of MHSA as 4 (work best). Here, we replace our standard attention in the pre-interaction layer in CNN by Pooling (Pre-Pooling CNN) and MHSA (Pre-MHSA CNN), and the post-interaction layer by MHSA in CNN (Post-MHSA CNN).

Results. Current UniPT attention outperforms MHSA by a large margin in all alternative locations. Notably, Post-MHSA in CNN shows a sharp decline in accuracy that reflects its incompatibility and instability over various complex interaction patterns for feature maps or token embeddings.

Bonus of Stronger Backbone Output as Guidance.

Settings. In Figure 6, we take larger ViT-B/16 or ViT-L/14 as stronger guidance for our UniPT with ViT-B/32 using *CLIP4Clip* on MSR-VTT. The larger model does not directly participate in the final output, which only serves as queries to compute interaction and confidence weights for ViT-B/32.

Results. We surprisingly find that more powerful guidance brings more beneficial gains, further validating the significance of introducing interaction and aggregation guidance.

Optimal Balance of UniPT over PETL Methods.

Settings. Figure 7 shows accuracy-memory trade-off using *CLIP4Clip* on MSR-VTT with varying reduction factor r.

Results. UniPT stands out by dramatically reducing training memory usage with competitive performance as LoRA, Adapter, and LST. Besides, it displays good stability and robustness across a diverse spectrum of side network sizes.

5. Conclusion

In this paper, we propose Universal Parallel Tuning (UniPT), a brand-new PETL paradigm that hits a sweet spot between performance and memory efficiency on various VL and GLUE downstream tasks. Crucially, our research unveils two intriguing phenomena: 1) Delicately leveraging the ultimate outputs as guidance can facilitate the adaptation capability of feature representations inside each layer of the pre-trained network; 2) Dynamically learning the aggregation weights also demonstrates optimal strategy and greater suitability for various inputs across modalities. By incorporating a lightweight network in parallel, our UniPT capitalizes on various pre-trained architectures across diverse domains, and more importantly, requires no backpropagation through the large backbone network. Extensive experiments have validated that our UniPT can not only significantly reduce memory footprint and surpass existing memory-efficient methods with good flexibility and broad applicability over various network backbones, but also achieve impressive benefits beyond recent PETL methods in a low-memory regime.

Limitations. We believe UniPT is a brand-new attempt for a desired and powerful architecture capable of memory-efficient transfer learning over various pre-trained backbones. A key limitation is that there is still a performance gap between UniPT and fully fine-tuning. Besides, a large input size (*e.g.*, shallow feature maps in CNN) may affect the computational complexity of UniPT, and the current pre-post design can further alleviate the memory consumption but improve the trainable parameters to some extent.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under grant No. 62293540, 62293542. It was also sponsored by CAAI-MindSpore Open Fund, developed on OpenI Community.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018. 6

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018. 1

[3] Ankur Bapna and Orhan Firat. Simple, scalable adaptation for neural machine translation. In *EMNLP*, pages 1538–1548, 2019. 1, 2

[4] Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In *TAC*, 2009. 5

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 1

[6] Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. Tinytl: Reduce memory, not parameters for efficient on-device learning. In *NeurIPS*, 2020. 1, 2

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9630–9640, 2021. 2

[8] Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *arXiv: 1708.00055*, 2017. 5

[9] David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, pages 190–200, 2011. 2, 5

[10] Guangyi Chen, Xiao Liu, Guangrun Wang, Kun Zhang, Philip H. S. Torr, Xiao-Ping Zhang, and Yansong Tang. Tem-adapter: Adapting image-text pretraining for video question answer. In *ICCV*, 2023. 2

[11] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Plot: Prompt learning with optimal transport for vision-language models. In *ICLR*, 2023. 3

[12] Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. In *CVPR*, pages 15789–15798, 2021. 1, 7

[13] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *NeurIPS*, 2022. 2

[14] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv: 1604.06174*, 2016. 3

[15] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 3

[16] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *ICLR*, 2023. 2

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019. 1, 2, 4, 6, 7

[18] Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. Similarity reasoning and filtration for image-text matching. In *AAAI*, pages 1218–1226, 2021. 1

[19] Haiwen Diao, Ying Zhang, Wei Liu, Xiang Ruan, and Huchuan Lu. Plug-and-play regulators for image-text matching. *TIP*, 32:2322–2334, 2023. 1

[20] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *IWP*, 2005. 5

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 4, 6, 7

[22] Yonatan Dukler, Alessandro Achille, Hao Yang, Varsha Vivek, Luca Zancato, Benjamin Bowman, Avinash Ravichandran, Charless C. Fowlkes, Ashwin Swaminathan, and Stefano Soatto. Your representations are in the network: composable and parallel adaptation for large scale models. In *NeurIPS*, 2023. 1

[23] Kaifeng Gao, Long Chen, Hanwang Zhang, Jun Xiao, and Qianru Sun. Compositional prompt tuning with motion cues for open-vocabulary video relation detection. In *ICLR*, 2023. 3

[24] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv: 2110.04544*, 2021. 1, 2

[25] Aidan N. Gomez, Mengye Ren, Raquel Urtasun, and Roger B. Grosse. The reversible residual network: Backpropagation without storing activations. In *NeurIPS*, pages 2214–2224, 2017. 3

[26] 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*, pages 6325–6334, 2017. 1, 2, 5

[27] Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. PPT: pre-trained prompt tuning for few-shot learning. In *ACL*, pages 8410–8423, 2022. 1, 3

[28] Demi Guo, Alexander M. Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In *ACL*, pages 4884–4896, 2021. 2

[29] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *ICLR*, 2022. 3

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7

[31] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735, 2020. 2

[32] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *ICML*, pages 2790–2799, 2019. 1, 2, 6

[33] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *ACL*, pages 328–339, 2018. 3

[34] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 1, 3, 6

[35] Zi-Yuan Hu, Yanyang Li, Michael R. Lyu, and Liwei Wang. VL-PET: vision-and-language parameter-efficient tuning via granularity control. In *ICCV*, 2023. 2

[36] Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. In *CVPR*, pages 6565–6574, 2023. 3

[37] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. 2, 5

[38] Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al. First quora dataset release: Question pairs. *data. quora. com*, 2017. 5

[39] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709–727, 2022. 1, 3

[40] Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *AAAI*, 2023. 3, 6

[41] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, pages 105–124, 2022. 3

[42] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR - modulated detection for end-to-end multi-modal understanding. In *ICCV*, pages 1760–1770, 2021. 1, 2, 5, 7

[43] Muhammad Uzair Khattak, Hanoona Abdul Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. *arXIv: 2210.03117*, 2022. 3

[44] Konwoo Kim, Michael Laskin, Igor Mordatch, and Deepak Pathak. How to adapt your large-scale vision-and-language model. *openreview*, 2021. 1, 2

[45] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020. 3

[46] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, pages 3045–3059, 2021. 3

[47] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, pages 9694–9705, 2021. 2

[48] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICLR*, pages 12888–12900, 2022. 2

[49] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, pages 4582–4597, 2021. 1, 3, 6

[50] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. In *NeurIPS*, 2022. 2, 6

[51] 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*, pages 740–755, 2014. 2, 5

[52] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *arXiv: 2205.05638*, 2022. 3

[53] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv: 1907.11692*, 2019. 2, 7

[54] Yitao Liu, Chenxin An, and Xipeng Qiu. Y-tuning: An efficient tuning paradigm for large-scale pre-trained models via label representation learning. *arXiv: 2202.09817*, 2022. 1, 3

[55] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002. IEEE, 2021. 2

[56] Haoyu Lu, Mingyu Ding, Yuqi Huo, Guoxing Yang, Zhiwu Lu, Masayoshi Tomizuka, and Wei Zhan. Uniadapter: Unified parameter-efficient transfer learning for cross-modal modeling. *arXiv: 2302.06605*, 2023. 2

[57] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022. 3

[58] 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: 2104.08860*, 2021. 1, 7

[59] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *NeurIPS*, pages 1022–1035, 2021. 1, 3

[60] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *ACL*, pages 565–576, 2021. 3

[61] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe,

and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, pages 185–201, 2018. 7

[62] Karttikeya Mangalam, Haoqi Fan, Yanghao Li, Chao-Yuan Wu, Bo Xiong, Christoph Feichtenhofer, and Jitendra Malik. Reversible vision transformers. In *CVPR*, pages 10820–10830, 2022. 3

[63] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016. 2, 5

[64] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. In *EMNLP*, pages 46–54, 2020. 6

[65] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *EACL*, pages 487–503, 2021. 1, 2

[66] Yanyuan Qiao, Zheng Yu, and Qi Wu. VLN-PETL: parameter-efficient transfer learning for vision-and-language navigation. In *ICCV*, 2023. 2

[67] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 7

[68] 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*, pages 8748–8763, 2021. 1, 2, 3, 6, 7

[69] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21:140:1–140:67, 2020. 1, 2, 3, 5

[70] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392, 2016. 5

[71] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, pages 18061–18070, 2022. 3

[72] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *TSP*, 45(11):2673–2681, 1997. 6

[73] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can CLIP benefit vision-and-language tasks? In *ICLR*, 2022. 1, 4, 7

[74] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642, 2013. 5

[75] 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*, pages 6418–6428, 2019. 1

[76] Yi-Lin Sung, Varun Nair, and Colin Raffel. Training neural networks with fixed sparse masks. In *NeurIPS*, pages 24193–24205, 2021. 2

[77] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. LST: ladder side-tuning for parameter and memory efficient transfer learning. In *NeurIPS*, 2022. 1, 3, 4, 6, 7

[78] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. VL-ADAPTER: parameter-efficient transfer learning for vision-and-language tasks. In *CVPR*, pages 5217–5227, 2022. 2, 7

[79] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *NeurIPS*, pages 200–212, 2021. 3

[80] Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. In *CVPR*, pages 7725–7735, 2023. 1

[81] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019. 5

[82] Naigang Wang, Jungwook Choi, Daniel Brand, Chia-Yu Chen, and Kailash Gopalakrishnan. Training deep neural networks with 8-bit floating point numbers. In *NeurIPS*, pages 7686–7695, 2018. 3

[83] Zhen Wang, Jun Xiao, Yueting Zhuang, Fei Gao, Jian Shao, and Long Chen. Learning combinatorial prompts for universal controllable image captioning. *arXiv preprint arXiv:2303.06338*, 2023. 3

[84] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *TACL*, 7:625–641, 2019. 5

[85] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*, pages 1112–1122, 2018. 5

[86] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. 5, 6, 7

[87] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016. 1, 2, 5

[88] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014. 2, 5

[89] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85, 2016. 2, 5

[90] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*, pages 1–9, 2022. 1, 2, 6

[91] Jeffrey O. Zhang, Alexander Sax, Amir Zamir, Leonidas J. Guibas, and Jitendra Malik. Side-tuning: A baseline for network adaptation via additive side networks. In *ECCV*, pages 698–714, 2020. 1, 3

[92] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao.

Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*, 2023. 2, 6

[93] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv: 2302.00923*, 2023. 3

[94] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16795–16804, 2022. 3

[95] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 1

[96] Jiawen Zhu, Simiao Lai, Xin Chen, Dong Wang, and Huchuan Lu. Visual prompt multi-modal tracking. In *CVPR*, pages 9516–9526, 2023. 3