# Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

To build an artificial neural network like the biological intelligence system, recent works have unified numerous tasks into a generalist model, which can process various tasks with shared parameters and do not have any task-specific modules. While generalist models achieve promising results on various benchmarks, they have performance degradation on some tasks compared with task-specialized models. In this work, we find that interference among different tasks and modalities is the main factor to this phenomenon. To mitigate such interference, we introduce the Conditional Mixture-of-Experts (Conditional MoEs) to generalist models. Routing strategies under different levels of conditions are proposed to take both the training/inference cost and generalization ability into account. By incorporating the proposed Conditional MoEs, the recently proposed generalist model Uni-Perceiver can effectively mitigate the interference across tasks and modalities, and achieves state-of-the-art results on a series of downstream tasks via prompt tuning on 1% of downstream data. Moreover, the introduction of Conditional MoEs still holds the generalization ability of generalist models to conduct zero-shot inference on new tasks, *e.g.,* video-text retrieval and video caption. Code and pre-trained generalist models shall be released.

## 1 Introduction

Generalist models that handle multiple modalities and numerous tasks have been long pursued by the machine learning community. However, previous researches [65, 90, 71] focus on developing specialized models with task-specific modules. When these models are applied to new tasks, the specifically-designed components need to be redesigned on demand and fine-tuned on sufficient downstream data. As a result, their model size increases with the number of diverse downstream tasks, conflicting with the goal of generalist models.

Recently, some pioneers [94, 80, 3, 85, 87, 62] have made preliminary attempts to build generalist models by modeling various tasks into a unified formulation. With the unified modeling, large-scale pre-training on various datasets enables the generalist models to process different downstream tasks using shared parameters. These generalist models not only achieve competitive performance on pre-training tasks [80, 3, 85, 87], but also can perform zero-shot inference on novel tasks without introducing additional parameters [94, 62].

However, compared to specialized models with specific parameters for each task, generalist models with shared parameters would suffer from the task-interference issue — different tasks with shared parameters may conflict with each other [89]. The same issue is also observed in multilingual NLP models [4, 82, 84]. We argue that the task-interference issue is mainly caused by the inconsistent optimization in multi-task learning. As shown in Tab. 1, during the training phase of generalist models, the gradient directions of different tasks would be inconsistent or even opposite. Thus,

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

if multiple tasks share parameters, the optimal update direction of the shared parameters will be uncertain, resulting in sub-optimal performance.

Allowing conflicting modalities and tasks to use separate parameters should effectively mitigate the interference issue in generalist models. Mixture of Experts (MoEs) [43, 23] provides a potential solution, which learns to activate sub-networks dynamically without introducing any task-specific modules. Nevertheless, vanilla MoEs [67] select the experts according to token representations, which suffers from high training/inference cost and neglects the information of different tasks and modalities. In this work, we argue that routing strategies of MoEs require special design when applied to generalist models for mitigating the task-interference issue.

To address the task-interference issue in generalist models, we propose Conditional Mixture-of-Experts (Conditional MoEs), which improve vanilla MoEs by introducing information under different levels of conditions, including token-level, context-level, modality-level, task-level, and predefined token attributes. In this case, vanilla MoEs is a token-level variant of our Conditional MoEs, which can be replaced by other-level variants to implement stronger generalist models. We carefully discussed the training/inference cost and generalization ability of different variants, and ablated their performances in mitigating the interference issue of generalist models. Notably, Conditional MoEs with predefined token attributes introduces 8-bit attribute embedding to describe the information of currently processed task and modalities, which demonstrate excellent computational and memory efficiency and good generalization ability.

To verify the effectiveness of Conditional MoEs, we incorporated it with the recently proposed generic perception model Uni-Perceiver [94] by replacing the linear projection in self-attention and FFN blocks with conditional MoE layers. Experiments demonstrate that, by mitigating task interference with our proposed Conditional MoEs, Uni-Perceiver can be pre-trained on various tasks jointly without performance degradation, while its generalization to other tasks can be maintained simultaneously. Our main contributions are as follows:

- We carefully analyze the task-interference issue in generalist models, and provide an explanation from the gradient direction perspective as well as a metric to quantify the issue.
- We propose Conditional MoEs to address the task-interference issue in generalist models. By introducing the information of currently processed task and modalities, Conditional MoEs effectively mitigate the interference issue, while keeping low computational and memory cost.
- Compared with previous SOTAs, our generalist model with 1% downstream data prompt tuning achieves competitive performance, while only <5% training data and <10% training cost are used. We hope this work can serve as a solid baseline for generalist models and motivate further research.

## 2 Related Works

**Specialized Models.** Previous research focuses on building specialized models for specific tasks. CNNs [47, 26, 70] and ViTs [20, 53, 77, 81] are developed for image classification. Subsequent works re-design them to adapt to diverse downstream visual tasks, *e.g.,* object detection [63] and segmentation [15, 48]. In NLP, different architectures are specifically designed for neural machine translation [78], natural language understanding [19], and natural language generation [51]. As for vision-language tasks, previous works usually combined modality-specific encoders and representation fusion modules together [13, 54]. Recently, [90, 65, 71] integrate several specialized models into a single one to handle diverse tasks. Such integrated specialized models are equipped with multiple task-specific modules to adapt to as many downstream tasks as possible. However, these methods still follow the task-specific paradigm, which conflicts with the objective of generalist models.

**Vanilla Generalist Models.** Vanilla generalist models handle different tasks and modalities with shared parameters. Uni-Perceiver [94] formulates various perception tasks as finding the maximum likelihood target for each input through the similarity of their representations. OFA [80], Flamingo [3] and SimVLM [85] attempt to unify different tasks into sequence-to-sequence generation. UniCORN [87] and Gato [62] further incorporate bounding box and reinforcement learning tasks into the unified formulation, respectively. These generalist models not only achieve competitive performance on pre-training tasks with shared parameters, but also can perform zero-shot inference on new tasks [62, 94]. However, these methods rarely investigate the potential interference among different modalities and tasks, which could result in the performance degradation of generalist models.

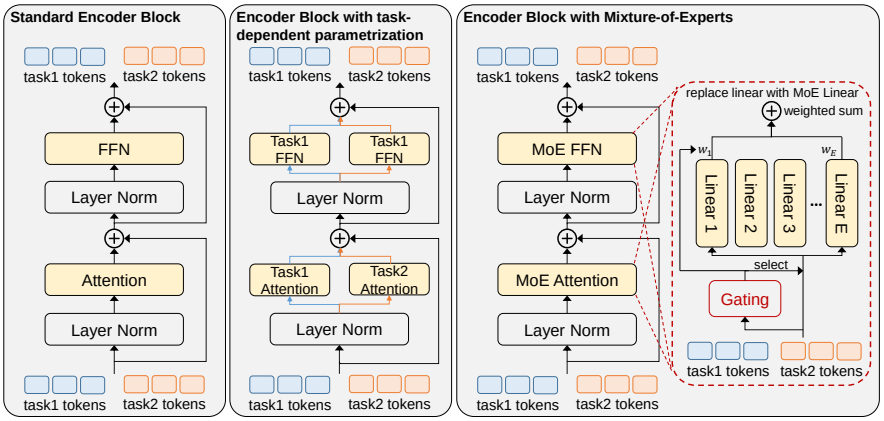


Figure 1: Comparisons of fully-shared standard encoder block, task-specific encoder block with task-dedicated parameters, and encoder block with efficient MoE parameterization.

**Multi-Task Learning.** Multi-task learning [8, 17] has been widely studied in the community of vision [27, 74, 72], language [25, 16, 50] and vision-language learning [10, 55, 29]. While multi-task training enables collaboration between tasks, it may also introduce the task interference problem [82, 84, 28, 36, 72]. To mitigate the task-interference issue, some works[14, 24, 38] propose to dynamically adjust the loss weight for each task, while others [91, 49, 36] instead use task-dedicated parameters. However, methods with task-specific parameters are difficult to generalize to new tasks and do not meet the requirements of generalist models.

**Mixture of Experts (MoEs).** MoEs has shown its remarkable ability to scale neural networks [67, 43, 23, 64, 21]. [67] first proves the effectiveness of MoEs by stacking MoE layers in the LSTM models. [68, 43] further introduce this approach to Transformer architectures. [23, 40] train language models with trillion parameters successfully by utilizing simplified MoE routing strategy and efficient training techniques. There are also some works applying MoEs to CNNs for computer vision tasks [1, 86, 83]. Recently, V-MoE [64] successfully employs MoEs to ViTs, showing promising performance on many visual tasks. Task-MoE [42] focuses on applying MoEs for multilingual translation to mitigate the interference among different languages. In this work, we aim to explore MoEs under different conditions for general models.

# 3 Methodology

In this section, we first analyze the task-interference problem from the gradient direction perspective. Based on the analysis, we propose Conditional Mixture-of-Experts (Conditional MoEs) for generalist models, which introduces parameters conditioned by information of different levels to mitigate the task-interference issue with negligible overhead.

## 3.1 Task Interference

To quantify the interference of the $j$-th task on the $i$-th task, we estimate the change in loss $L_i$ of the $i$-th task, when optimizing the shared parameters $\theta$ according to the $j$-th task $L_j$ as:

$$\Delta_j L_i(x_i) \doteq \mathbb{E}_{x_j}\left(L_i(x_i;\theta) - L_i(x_i;\theta - \lambda \frac{\nabla_\theta L_j(x_j)}{\|\nabla_\theta L_j(x_j)\|})\right) \approx \lambda \mathbb{E}_{x_j}\left(\frac{\nabla_\theta L_j(x_j)}{\|\nabla_\theta L_j(x_j)\|}^T \nabla_\theta L_i(x_i)\right), \quad (1)$$

where $x_i$ and $x_j$ are the sampled training batches of the $i$-th and $j$-th tasks, respectively. Without loss of generality, we only consider the update direction ignoring the update norm. Then, the interference of the $j$-th task on the $i$-th task can be quantified as:

$$\mathcal{I}_{i,j} = \mathbb{E}_{x_i}\left(\frac{\Delta_j L_i(x_i)}{\Delta_i L_i(x_i)}\right), \quad (2)$$

where the denominator is used to normalize the loss change scale. As reported in Tab. 1, we sample 100 batches for each tasks, and record the gradients to calculate the average interference metric $\mathcal{I}_{i,j}$ of the $j$-th task on the $i$-th task at the 4-th/12-nd FFN blocks. We see that, at shallow layers, the image

Table 1: The average interference metric $\mathcal{I}_{i,j}$ of the task $j$ on the task $i$ at the 4-th/12-nd FFN blocks. To calculate the interference metric, we sample 100 batches for each tasks, and record the gradients based on the pre-trained Uni-Perceriver-Ti. The red value indicates that the task $j$ has a negative impact on the task $i$, and the green value indicates a positive impact.

(a) The 4-th FFN Block

| Task $i$ \ Task $j$ | ImgCLS (Img) | MLM (Text) | Caption (Img-Text) |
|---|---|---|---|
| ImgCLS (Img) | 1.00 | -0.57 | 1.29 |
| MLM (Text) | 0.07 | 1.00 | 0.68 |
| Caption (Img-Text) | 0.01 | 0.01 | 1.00 |

(b) The 12-nd FFN Block

| Task $i$ \ Task $j$ | ImgCLS (Img) | MLM (Text) | Caption (Img-Text) |
|---|---|---|---|
| ImgCLS (Img) | 1.00 | -2.91 | -2.45 |
| MLM (Text) | -1.65 | 1.00 | -1.05 |
| Caption (Img-Text) | -0.11 | 0.19 | 1.00 |

caption task has positive impacts on image classification and masked language modeling, suggesting that cooperation between different tasks exists. While at deep layers, tasks with different optimization objectives hardly enhance each other, and the gradient directions may even opposite.

Fig. 1 summarizes three mainstream architectures for multi-task models. The first is the standard architecture [94, 31, 32] with parameters fully shared by different tasks, which suffers from task interference problem as analyzed above. The second is task-specific parameterized architecture [90, 65, 29, 71] equipped with dedicated parameters for each task. Although this architecture address the interference problem by task-specific parameters, it is difficult to generalize to new tasks that did not emerge in the training phase. Unlike the above two architectures, the Mixture-of-Experts (MoE) architecture [67, 43, 23, 40, 64] activates models sparsely according to different given inputs by selectively utilizing different subset of the model parameters. The sparse routing mechanism makes it possible to train very large generalist models, which maximizes the collaboration and meanwhile mitigates the interference problem. In this work, we focus on exploring Conditional MoEs for general models, whose experts are gated by conditions from different levels.

### 3.2 Conditional Mixture-of-Experts (Conditional MoEs)

We first describe the prototype of Conditional MoEs, and then provide its specific instantiations under different conditions, as well as the application to generalist models.

**Prototype.** Given any token $x_i$ in the input sequence $X = \{x_i\}_{i=1}^{L}$, conditional MoEs with $E$ experts firstly introduces a gate decision vector $\mathcal{G} \in \mathbb{R}^E$ that dispatches different input tokens to different experts, which is calculated as:

$$\mathcal{G} = \mathrm{top}_k\left(\mathrm{softmax}\left(\mathbf{W}_g \cdot R(x_i) + \epsilon\right)\right). \tag{3}$$

where $R(\cdot)$ defines a general routing strategy for gate decision, which is alternative under different conditions. $\mathbf{W}_g$ is the trainable weights in gate decision and $\epsilon$ is the noise term. The $\mathrm{top}_k(\cdot)$ operator sets all values to be zero except the largest $k$ values. Since $\mathcal{G}$ only has $k \ll E$ non-zero values, the token $x_i$ is routed to only a small number of experts. After getting the gate decision vector $\mathcal{G}$, the corresponding output $y_i$ is the weighted combination of each expert's computation on $x_i$ as:

$$y_i = \sum_{e=1}^{E} \mathcal{G}_e \cdot \mathbf{W}_e \cdot x_i, \tag{4}$$

where $\mathbf{W_e}$ is the linear projection weights of the $e$-th expert and gate decision $\mathcal{G}_e$ determines how much the $e$-th expert contributes to the output $y_i$. Note that, experts with $\mathcal{G}_e = 0$ does not need to be computed for saving computation.

In Conditional MoEs, the routing strategy $R(\cdot)$ plays an important role in the multi-modality and multi-task training of generalist models. By sparsely activating experts according to different conditions, Conditional MoEs can mitigate the interference issue while maintaining the generality of the pretrained model. Next, we introduce variants with specific routing strategies under different conditions, as shown in Fig. 2.

**Token-Level Routing.** Similar to vanilla MoEs [67, 43, 23, 40, 64], the token-Level MoEs directly use the token representation for the routing strategy, which can be written as:

$$R_{\text{token}}(x_i) = x_i. \tag{5}$$

The routing strategy of token-level MoEs is an identical function, where the gate decision only depends on each token's own representation.

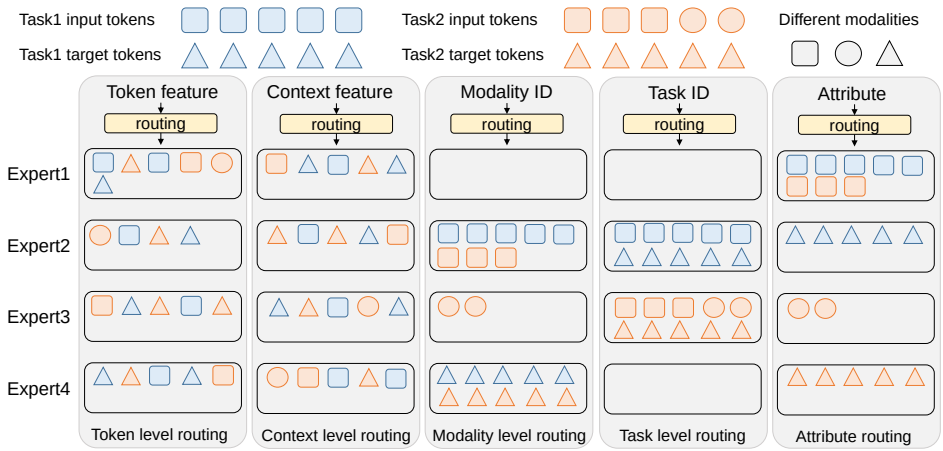


Figure 2: Comparisons of routing strategies with the top-1 gate decisions under 2-task training.

Table 2: The 8-dimensional binary embedding used for attribute-level routing strategy. The attribute embedding is assigned to a token by checking whether the statements of the eight descriptions match the current token. For example, the attribute embedding for any token from the input sequences of image classification task should be $[1, 0, 0, 1, 1, 0, 0, 1]$. Please refer to the Appendix for detailed look-up table of attribute embeddings for all tasks in our work.

| Index | Descriptions | Yes | No |
|---|---|---|---|
| 0 | Visual modality exists in the inputs of the current task. | 1 | 0 |
| 1 | Text modality exists in the inputs of the current task. | 1 | 0 |
| 2 | Visual modality exists in the targets of the current task. | 1 | 0 |
| 3 | Text modality exists in the targets of the current task. | 1 | 0 |
| 4 | The modality of current token is visual. | 1 | 0 |
| 5 | The modality of current token is text. | 1 | 0 |
| 6 | The attention mask of the current token is causal. | 1 | 0 |
| 7 | The current token comes from the inputs, not the targets. | 1 | 0 |

**Context-Level Routing.** Tokens with similar representations may appear in conflicting tasks, whose optimal expert decisions should be different to mitigate the task interference. Therefore, to help gate function making more reliable decisions, we explore the combination of global context and local token representation. The routing strategy utilizing global context can be expressed as:

$$R_{\text{context}}(x_i) = \text{concat}(x_i, \text{attnpool}(X)), \tag{6}$$

where $\text{concat}(\cdot)$ indicates the concatenation operation, $X = \{x_i\}_{i=1}^{L}$ is sequence of all tokens in the current sample, and $\text{attnpool}(\cdot)$ indicates the attention pooling operator [61].

**Modality-Level Routing.** Most current practice uses modality-specific encoders with independent parameters for different modality inputs. Inspired by this, we also explore to leverage the modality of the current token as a routing strategy:

$$R_{\text{modal}}(x_i) = \text{embed}(\text{id}_{\text{modal}}(x_i)) \tag{7}$$

Here, $\text{embed}(\cdot)$ represent the embedding layer, and $\text{id}_{\text{modal}}(\cdot)$ indicates the modality index of current token $x_i$. The routing function will assign this token to experts according to its modality embedding.

**Task-Level Routing.** In addition to modality information, task information can also be used to guide gate functions to make reliable decisions for mitigating the task interference. Similar to Eqn. (7), the routing strategy can be formulated as:

$$R_{\text{task}}(x_i) = \text{embed}(\text{id}_{\text{task}}(x_i)), \tag{8}$$

where $\text{id}_{\text{task}}(\cdot)$ is the task index of current token $x_i$. The task embedding for this task will be used to compute gate decision. Since all tokens from one task have the same task embedding, all tokens corresponding to this task will be routed to the same set of experts.

**Attribute Routing.** Among the aforementioned variants, token-level, context-level, and modality-level routing strategies only focus on input tokens but omit information about the currently processed

task. While task-level routing strategy relies on task-specific ids, it limits the generalization ability to new downstream tasks. To introduce the information of currently processed task and modalities without losing the generalization ability of generalist models, we propose to introduce token attributes to assist the gate decision.

As described in Tab. 2, the attributes of the current token are represented as an 8-dimensional binary embedding, whose attributes include the modalities of current task and token (index 0$\sim$5), the causation type of the model (index 6), and the token source (index 7). As a result, the designed token attributes provide comprehensive information of currently processed task meanwhile keeping the task generalization ability. Based on the attribute embedding, the routing strategy is expressed as:

$$R_{\text{attr}}(x_i) = \text{layernorm}\left(\mathbf{W}_{\text{attr}} \cdot \text{attr}(x_i)\right). \tag{9}$$

Here, $\text{attr}(x_i)$ is the 8-dimensional binary attribute embedding of the current token $x_i$ as described in Tab. 2. $\mathbf{W}_{\text{attr}}$ is the learnable weights to transform the attribute embedding to latent representation, and $\text{layernorm}(\cdot)$ denotes the layer normalization [6] for training stabilization.

**Application to Generalist Models.** Without loss of generality, we explore the application of Conditional MoEs to the generalist model Uni-Perceiver [94], which uses Transformers to handle various modalities and tasks with shared parameters. We replace linear projection layers in both self-attention and FFN blocks with Conditional-MoE layers (see Fig. 1).

### 3.3 Comparison of Conditional-MoE Variants

As illustrated in Fig. 2, among the variants of Conditional MoEs, token-level and context-level MoEs are data-dependent, while modality-level, task-level, and attribute MoEs are data-independent.

**Training and Inference Cost.** Compared to dense models with the same number of parameters, all Conditional MoE variants can significantly reduce the computational cost benefiting from the sparse routing mechanism. Due to the dependence of input data, the memory consumption of token-level and context-level MoEs is relatively high during model training, and model parallelism is required to relieve memory cost by partitioning experts across multiple devices, leading to heavy inter-device communication overhead. This problem persists when using pre-trained models for task-specific inference, where all experts need to be loaded into memory and might be activated by any token.

Different from data-dependent Conditional MoEs, data-independent variants such as modality-level, task-level, and attribute MoEs have excellent memory efficiency, since only top-$k$ experts need to be activated for all tokens with the same modality/task/attributes. Moreover, in both training and inference phase, the experts in a data-independent MoE layer can be merged into a single linear projection using reparameterization techniques. In this case, the computation cost of the network with data-independent Conditional MoEs will be equivalent to a dense model without MoEs.

**Generalization Ability.** We hope to mitigate the task-interference issue in generalist models, while keeping their generalization ability to new downstream tasks. While token-level, context-level, and modality-level MoEs without task-specific designs do not harm the generalization ability, they ignore the task-level information which is essential to resolve the task interference. Conversely, task-level routing strategy is tied to a specific task id, which is difficult to generalize to new downstream tasks. Attribute MoEs introduce predefined token attributes to comprehensively describe the information of currently processed task and modalities, which can be transferred to new downstream tasks without any task-specific modifications. This gives attribute MoEs the potential to mitigate task interference without losing generalization ability.

## 4 Experiments

In this section, we first describe our experimental setup. Then, we confirm the task-interference issue in the generalist model Uni-Perceiver [94] and ablate the the ability of different Conditional MoEs to mitigate task interference. Finally, large-scale training is conducted to verify the effectiveness of our proposed Conditional-MoEs and its generalization ability to novel tasks.

Table 3: The performance of different routing strategies for Conditional MoEs. The base model is Uni-Perceiver with $\text{BERT}_{\text{tiny}}$. We also illustrate the task-specific variant where each task has its own specialized parameters. The training and validation performance reported on three tasks: image classification on ImageNet-1K [18], image caption on COCO Caption [12], and Masked Language Modeling(MLM) on Books&Wiki. The best results within a tolerance of 1% are in **bold**.

| model | task-specific parameterization | training time | inference time | ImageNet-1k | | COCO Caption | | MLM | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | $\uparrow\text{acc}_{\text{train}}$ | $\uparrow\text{acc}_{\text{val}}$ | $\uparrow\text{acc}_{\text{train}}$ | $\uparrow\text{B@4}_{\text{val}}$ | $\uparrow\text{acc}_{\text{train}}$ | $\downarrow\text{ppl}_{\text{val}}$ |
| Uni-Perceiver-Ti [94] | | 1.0× | 1.0× | 47.3 | 68.3 | 49.2 | 18.2 | 54.5 | 5.86 |
| | ✓ | 1.1× | 1.0× | **53.3** | **73.5** | **52.6** | 20.4 | **60.5** | **4.48** |
| + Conditional MoEs $_{\text{token}}$ | | 1.8× | 2.2× | **53.1** | 72.7 | **52.9** | 20.9 | 58.3 | 4.96 |
| + Conditional MoEs $_{\text{context}}$ | | 2.2× | 2.6× | 52.5 | **73.1** | **52.8** | 21.5 | 58.6 | 4.86 |
| + Conditional MoEs $_{\text{modality}}$ | | 1.4× | 1.0× | 51.7 | 72.6 | 52.1 | 21.8 | 57.5 | 5.06 |
| + Conditional MoEs $_{\text{task}}$ | | 1.4× | 1.0× | **52.9** | **73.2** | **52.7** | 21.2 | **59.9** | **4.56** |
| **+ Conditional MoEs $_{\text{attribute}}$** | | 1.4× | 1.0× | **52.8** | **73.3** | **53.1** | **23.0** | **60.0** | **4.56** |

### 4.1 Datasets

We use the same datasets in Uni-Perceiver [94] to pre-train our models[1]. Specifically, ImageNet-21k [18] is used for image classification pre-training. Kinetics-700 [37] and Moments in Time [57] are used for video classification pre-training. Language modeling task is trained on BookCorpus [95] & English Wikipedia (Books&Wiki). For language modeling with image clues and image-text retrieval, we use a combination of image-text-pair datasets: SBU Captions (SBU) [58], Visual Genome [41], COCO Caption [12], CC3M [66], CC12M [9] and YFCC [35]. Following Uni-Perceiver, Imagenet1K [18], Kinetics-400 [37], COCO Caption [12], and Flickr30k [59] are utilized to evaluate the performance of generalist models on downstream tasks. We also use two datasets that evaluate the generalization ability to novel tasks: MSVD [11] and GLUE [79]. Additionally, all dataset licenses are included in Appendix.

### 4.2 Implementation Details

We incorporate the vanilla generalist model Uni-Perceiver with Conditional MoEs for experiments with three different variants: Uni-Perceiver-Ti (Tiny), Uni-Perceiver-B (Base), and Uni-Perceiver-L (Large). Please refer to Appendix for architecture hyperparameters. If not specified, the input image resolution is set to 224×224. In each training iteration, each GPU independently samples a single task and dataset. The gradients of different GPUs are synchronized after the gradient back-propagation. We use the AdamW optimizer with a base learning rate of 0.0005 and a weight decay of 0.05. Similar to [52, 61], we find setting $\beta_2 = 0.98$ and $\epsilon = 10^{-6}$ helps improve stability when large-scale training. Besides, gradient clipping with 0.5 is used to stabilize training.

Uni-Perceiver-B and Uni-Perceiver-L are equipped with Conditional-MoEs layer for every other layers while Uni-Perception-Ti use Conditional MoEs in all layers. A normal noise is also used on the gate logits following [64] for a better exploration for new potential experts. If not specialized, top-2 gate function is used. For other hyper-parameters of MoE layers, please refer to Appendix.

### 4.3 Ablation Studies

This part explores whether Conditional MoEs can effectively mitigate task interference in generalist models and compares different routing strategies. Tab. 3 summarizes the performance of Uni-Perceiver and its variants on three typical tasks. Compared with task-specific parameterization, the performance degradation of Uni-Perceiver confirms the existence of task interference. Incorporating Conditional MoEs with any routing strategy can mitigate the task-interference issue and significantly improve the performance. Among these five routing strategies, token-level, context-level, and modality-level MoEs deliver slightly worse performance. We argue the missed task information is critical for resolving the task interference. Besides, the data-dependent MoEs, *i.e.,* token-level and context-level, have relatively higher training and inference cost, while the other three MoEs have excellent efficiency by using reparameterization techniques. Although both task-level and attribute MoEs achieve good performance, the specialized task-id design in task-level MoEs makes it difficult to generalize to new tasks. Therefore, the Conditional MoEs with attribute routing strategy will be used.

[1] As far as we know, all datasets do not contain any personally identifiable information or offensive content.

Table 4: The performance of incorporating Conditional MoEs with Uni-Perceiver on image classification, video classification and image-text retrieval. "#param" is the parameters required during model deployment. "#data" is the amount of visual training samples involved. "WT", "PT", and "FT" indicate w/o tuning, prompt tuning, and fine-tuning, respectively. "1%" and "100%" indicate the proportion of downstream data used for tuning. "$\text{FT}_{100\%}\uparrow$" means fine-tuning with larger image size. The subscript number next to score indicates that a different image resolution than 224 is used. † These methods use $> 20\times$ training data size and $> 10\times$ training cost than ours.

(a) Image Classification accuracy on ImageNet-1k.

| Method | #param | #data | WT | $\text{PT}_{1\%}$ | $\text{FT}_{100\%}$ | $\text{FT}_{100\%}\uparrow$ |
|---|---|---|---|---|---|---|
| DeiT-B [76] | 86M | 1.28M | - | - | 81.8 | $83.1_{384}$ |
| ViT-B [73] | 86M | 15.5M | - | - | 84.0 | $85.5_{384}$ |
| ViT-L [73] | 307M | 15.5M | - | - | 84.0 | $85.6_{384}$ |
| OFA [80] | 472M | 60.6M | - | - | - | $84.9_{480}$ |
| CLIP [61] | 307M | 400M | $76.2_{336}$ | - | - | - |
| †ALIGN [33] | 480M | 1.8B | $76.4_{289}$ | - | - | $88.6_{289}$ |
| †Florence [90] | 637M | 900M | $83.7_{384}$ | - | - | $90.0_{\geq 384}$ |
| †CoCa-B [88] | 86M | 4.8B | $82.6_{576}$ | - | - | $88.3_{576}$ |
| †CoCa-L [88] | 303M | 4.8B | $84.8_{576}$ | - | - | $90.2_{576}$ |
| †Flamingo-3B [3] | 3.2B | 2.3B | - | $71.0_{320}$ | - | - |
| Uni-Perceiver-B | 86M | 44.1M | 79.2 | 80.9 | 84.0 | $85.2_{384}$ |
| + Conditional MoEs | 86M | 44.1M | 80.3 | 82.0 | 84.5 | $85.8_{384}$ |
| Uni-Perceiver-L | 354M | 303M | 82.7 | 84.2 | 86.2 | $87.0_{384}$ |
| + Conditional MoEs | 303M | 44.1M | 83.4 | 84.9 | 86.4 | $87.0_{384}$ |

(b) Video classification accuracy on Kinetics-400.

| Method | #param | #data | WT | $\text{PT}_{1\%}$ | $\text{FT}_{100\%}$ |
|---|---|---|---|---|---|
| TimeSformer-B [7] | 121.4M | 14.2M | - | - | 80.7 |
| VATT-B [2] | 87.9M | 238M | - | - | $79.6_{320}$ |
| VATT-L [2] | 306.1M | 238M | - | - | $82.1_{320}$ |
| ViViT-L [5] | >307M | 14.2M | - | - | 81.7 |
| ViViT-L [5] | >307M | 300M | - | - | 84.9 |
| †Florence [90] | 647M | 900M | - | - | $86.5_{384}$ |
| †CoCa [88] | 2.1B | 4.8B | - | - | $88.9_{576}$ |
| Uni-Perceiver-B | 86M | 44.1M | 74.5 | 74.8 | 77.7 |
| + Conditional MoEs | 86M | 44.1M | 76.8 | 77.2 | 79.3 |
| Uni-Perceiver-L | 303M | 44.1M | 79.5 | 80.0 | 81.9 |
| + Conditional MoEs | 303M | 44.1M | 82.1 | 83.0 | 84.2 |

(c) Image-text retrieval R@1 performance.

| Method | | | Flickr30K | | | | | | MSCOCO Caption | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Image → Text | | | Text → Image | | | Image → Text | | | Text → Image | | |
| | #param | #data | WT | $\text{PT}_{1\%}$ | $\text{FT}_{100\%}$ | WT | $\text{PT}_{1\%}$ | $\text{FT}_{100\%}$ | WT | $\text{PT}_{1\%}$ | $\text{FT}_{100\%}$ | WT | $\text{PT}_{1\%}$ | $\text{FT}_{100\%}$ |
| ImageBERT [60] | 170M | 10M | 70.7 | - | 87.0 | 54.3 | - | 73.1 | 44.0 | - | 66.4 | 32.3 | - | 50.5 |
| UNITER-B [13] | 146M | 9.6M | 80.7 | - | 85.9 | 66.2 | | 72.5 | - | - | 64.4 | - | - | 50.3 |
| UNITER-L [13] | 363M | 9.6M | 83.6 | - | 87.3 | 68.7 | - | 75.6 | - | - | 65.7 | - | - | 52.9 |
| ViLT [39] | 87M | 9.7M | 73.2 | - | 74.8 | 56.5 | - | 61.5 | 55.0 | - | 64.4 | 40.4 | - | 42.7 |
| FLAVA [71] | 215M | 70M | 67.7 | - | - | 65.2 | - | - | 42.7 | - | - | 38.4 | - | - |
| CLIP [61] | 417M | 400M | $88.0_{336}$ | - | - | $68.7_{336}$ | - | - | $58.4_{336}$ | - | - | $37.8_{336}$ | - | - |
| †ALIGN | 820M | 1.8B | $88.6_{289}$ | - | $95.3_{289}$ | $75.7_{289}$ | - | $84.9_{289}$ | $58.6_{289}$ | - | $77.0_{289}$ | $45.6_{289}$ | - | $59.9_{289}$ |
| †Florence [90] | 893M | 900M | $90.9_{384}$ | - | $97.2_{384}$ | $76.7_{384}$ | - | $87.9_{384}$ | $64.7_{384}$ | - | - | $47.2_{384}$ | - | - |
| †CoCa-B [88] | 383M | 4.8B | $89.8_{576}$ | - | - | $76.8_{576}$ | - | - | $63.8_{576}$ | - | - | $47.5_{576}$ | - | - |
| †CoCa-L [88] | 787M | 4.8B | $92.5_{576}$ | - | - | $80.4_{576}$ | - | - | $66.3_{576}$ | - | - | $51.2_{576}$ | - | - |
| †Flamingo-3B [3] | 3.2B | 2.3B | $89.3_{320}$ | - | - | $79.5_{320}$ | - | - | $65.9_{320}$ | - | - | $48.0_{320}$ | - | - |
| Uni-Perceiver-B | 124M | 44.1M | 82.3 | 91.0 | 92.7 | 71.1 | 76.0 | 77.5 | 64.9 | 68.4 | 69.8 | 50.7 | 51.9 | 53.9 |
| + Conditional MoEs | 167M | 44.1M | 82.1 | 91.3 | 93.6 | 72.4 | 78.5 | 79.8 | 64.6 | 68.9 | 70.5 | 51.6 | 52.6 | 54.1 |
| Uni-Perceiver-L | 354M | 44.1M | 83.7 | 92.1 | 94.7 | 74.2 | 80.0 | 82.1 | 67.8 | 73.3 | 74.4 | 54.1 | 56.2 | 57.9 |
| + Conditional MoEs | 505M | 44.1M | 83.6 | 92.4 | 94.1 | 75.9 | 80.6 | 83.7 | 67.9 | 73.3 | 74.7 | 55.3 | 57.1 | 58.3 |

### 4.4 Evaluation on Pre-training tasks

Large-scale training is conducted to verify the effectiveness of our method, we first evaluate it on tasks involved in pre-training. Specifically, we use widely-used Imagenet-1k [18] and Kinetics-400 [37] to evaluate image and video classification respectively, and use popular Flickr30k [59] and COCO Caption [12] to evaluate image caption and image-text retrieval.

Tab. 12 and Tab. 13a show the results on the four pre-training tasks. We see that Uni-perceiver with our Conditional MoEs consistently outperforms vanilla Uni-perceiver by a large margin. Without any tuning, our models achieve comparable performance with task-specific SOTAs trained with similar model size and training data size. Note that, our approach is a generalist model pretrained on a unified task formulation, while task-specific approaches are trained specifically for the target task.

When prompt tuned on only 1% downstream data, the performance of our models are boosted to a level close to counterparts that use $>50\times$ training data sizes and $>10\times$ training cost. For the prompt tuning of our models, only a small amount of parameters are tuned, and the encoder is still fixed and shared among different tasks, indicating that generalist models with Conditional MoEs can handle different tasks with significant low cost than counterparts.

We further fine-tune our models with 100% of the downstream data. In this case, our model achieves performance on-par with or better than the SOTAs trained with similar data size on all these tasks, which proves generalist models with Conditional MoEs has learned high-quality representations.

Table 5: The performance of incorporating Conditional MoEs with Uni-Perceiver on image caption, natural language understanding, video-text retrieval and video caption, where the last three tasks are not involved in pre-training.

(a) Image caption BLEU@4 performance. * means methods use region features as network inputs. ‡ indicates that Cider optimization is used.

| Method | #param | data | MSCOCO Caption | | | Flickr30k | | |
|---|---|---|---|---|---|---|---|---|
| | | | WT | $PT_{1\%}$ | $FT_{100\%}$ | WT | $PT_{1\%}$ | $FT_{100\%}$ |
| *Unified VLP [93] | 86M | 3.0M | - | - | 36.5 | - | - | 30.1 |
| *OSCAR-B [46] | 154M | 6.5M | - | - | 36.5 | - | - | - |
| *OSCAR-L [46] | 384M | 6.5M | - | - | 37.4 | - | - | - |
| UNICORN [87] | 198M | 200k | - | - | 35.8 | - | - | - |
| BLIP-B [44] | 252M | 129M | - | - | $39.7_{384}$ | - | - | - |
| BLIP-L [44] | 473M | 129M | - | - | $40.4_{384}$ | - | - | - |
| CLIP-VIL [69] | > 459M | 400M | - | - | 40.2 | - | - | - |
| SimVLM [85] | 632M | 1.8B | $11.2_{480}$ | - | $40.6_{480}$ | - | - | - |
| OFA [80] | 472M | 60.6M | - | - | $42.4_{480}$ | - | - | - |
| *‡OSCAR-L [46] | 384M | 6.5M | - | - | 41.7 | - | - | - |
| †CoCa [88] | 2.1B | 4.8B | | | $40.9_{576}$ | - | - | - |
| Uni-Perceiver-B | 124M | 44.1M | 32.0 | 35.5 | 36.4 | 14.7 | 30.2 | 31.2 |
| + Conditional MoEs | 167M | 44.1M | 33.2 | 36.8 | 37.3 | 15.9 | 30.7 | 32.4 |
| Uni-Perceiver-L | 354M | 44.1M | 35.3 | 38.6 | 39.2 | 15.1 | 32.9 | 35.5 |
| + Conditional MoEs | 505M | 44.1M | 35.5 | 39.3 | 40.5 | 15.8 | 33.7 | 36.2 |

(b) Natural language understanding (novel task) fine-tuned on GLUE. $BERT_{BASE}$ records from [34]. VisualBERT and LXMERT record from [30]. *RoBERTa uses $10\times$ training text tokens than ours.

| Method | MNLI (Acc) | QNLI (Acc) | QQP (F1) | RTE (Acc) | SST-2 (Acc) | MRPC (F1) | CoLA (Mcc) |
|---|---|---|---|---|---|---|---|
| LXMERT [75] | 80.4 | 84.2 | 75.3 | 57.2 | 90.2 | 80.4 | 39.0 |
| VisualBERT [45] | 81.6 | 87.0 | 86.0 | 56.6 | 89.4 | 82.1 | 38.6 |
| SimVLM-B [85] | 83.4 | 88.6 | 87.2 | 63.9 | 90.9 | 84.4 | 46.7 |
| BERT-B [79] | 84.5 | 88.4 | 88.3 | 63.5 | 92.9 | 89.0 | 54.7 |
| BERT-L [79] | 86.6 | 92.3 | 91.3 | 70.4 | 93.2 | 88.0 | 60.6 |
| OFA-B [80] | 84.3 | 91.1 | 88.4 | 70.8 | 92.7 | 90.6 | 52.3 |
| OFA-L [80] | 86.6 | 92.8 | 88.9 | 73.6 | 94.7 | 91.4 | 53.1 |
| *RoBERTa-B [52] | 87.6 | 92.8 | 91.9 | 78.7 | 94.8 | 90.2 | 63.6 |
| *RoBERTa-L [52] | 90.2 | 94.7 | 92.2 | 86.6 | 96.4 | 90.9 | 68.0 |
| Uni-Perceiver-B | 79.7 | 87.3 | 86.7 | 71.1 | 89.3 | 86.0 | 43.1 |
| + Conditional MoEs | 81.5 | 88.2 | 87.8 | 75.8 | 90.9 | 87.1 | 52.2 |
| Uni-Perceiver-L | 82.5 | 89.2 | 87.7 | 73.7 | 91.2 | 90.2 | 52.0 |
| + Conditional MoEs | 85.7 | 91.9 | 89.5 | 78.4 | 93.4 | 91.2 | 57.4 |

(c) Video-text retrieval (novel task) Recall@1 and video caption (novel task) BLEU@4 performance on MSVD.

| Method | #param | #data | Video → Text | | | Text → Video | | | Video Caption | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | WT | $PT_{1\%}$ | $FT_{100\%}$ | WT | $PT_{1\%}$ | $FT_{100\%}$ | WT | $PT_{1\%}$ | $FT_{100\%}$ |
| CLIP2video [22] | 132M | 400M | - | - | 58.7 | - | - | 47.0 | - | - | - |
| HunYuan_tvr [56] | 364M | 400M | - | - | 68.0 | - | - | 52.7 | - | - | - |
| ORG-TRL [92] | 86M | 2.0M | - | - | - | - | - | - | - | - | 54.3 |
| Uni-Perceiver-B | 124M | 44.1M | 50.3 | 62.7 | 62.8 | 38.7 | 43.8 | 45.8 | 22.6 | 59.5 | 63.3 |
| + Conditional MoEs | 167M | 44.1M | 52.8 | 65.6 | 65.0 | 40.0 | 45.3 | 47.8 | 23.4 | 60.0 | 65.4 |
| Uni-Perceiver-L | 354M | 44.1M | 45.4 | 65.5 | 65.2 | 34.2 | 48.6 | 50.8 | 24.7 | 67.2 | 68.3 |
| + Conditional MoEs | 505M | 44.1M | 45.7 | 66.4 | 67.6 | 41.9 | 50.3 | 52.3 | 24.6 | 67.6 | 68.9 |

### 4.5 Generalization to Novel Tasks

The generalization ability is the most attractive aspect of generalist models, while the dynamic sub-networks activation of Conditional MoEs should maintain this ability while mitigating task interference. To verify this, we conduct experiments on video caption, video-text retrieval, and natural language understanding tasks, which did not appear in pre-training. As shown in Tab. 13c, our Uni-Perceiver equipped with Conditional MoEs could generalize to video-related tasks very well. They can obtain reasonable zero-shot performance on those tasks and also perform better than vanilla Uni-Perceiver with a great margin. Moreover, Uni-Perceiver-MoEs can achieve comparable results to SOTA methods with similar training cost by further conducting prompt tuning with only 1% data. Beyond that, Conditional MoEs can significantly boost the performance of Uni-Perceiver on GLUE benchmarks (Tab. 13b), owing to its excellent ability to resolve task interference in generalist models.

## 5 Conclusion

In this paper, we propose Conditional MoEs to address the task-interference issue in generalist models. By sparsely activate sub-networks without introducing any task-specific designs, generalist models can be pre-trained on multiple tasks jointly without performance degradation, while keeping the generalization ablity to novel tasks. We incorporate Conditional MoEs with the recently proposed generalist model Uni-Perceiver. With prompt tuning on 1% downstream data, the proposed sparse generalist model achieves competitive performance with previous SOTAs using only $<5\%$ training data and $<10\%$ training cost. We hope this work can motivate further research in generalist models.

**Limitations.** Our method is currently verified on generalist models with millions of parameters. For generalist models with billions of parameters, whether the task-interference issue exists and whether our method is still effective are questionable, which we leave them to future work.

**Potential Negative Societal Impact.** This work shares the common negative impacts of large-scale training, which may consume lots of electricity and result in increased carbon emissions. This method also learns from a large number of datasets that may contain data biases.

# References


[1] A. Abbas and Y. Andreopoulos. Biased mixtures of experts: Enabling computer vision inference under data transfer limitations. *TIP*, 2020.

[2] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *NIPS*, 2021.

[3] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[4] N. Arivazhagan, A. Bapna, O. Firat, D. Lepikhin, M. Johnson, M. Krikun, M. X. Chen, Y. Cao, G. Foster, C. Cherry, et al. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019.

[5] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021.

[6] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[7] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

[8] R. Caruana. Multitask learning. *Machine learning*, 1997.

[9] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

[10] D. S. Chaplot, L. Lee, R. Salakhutdinov, D. Parikh, and D. Batra. Embodied multimodal multitask learning. *arXiv preprint arXiv:1902.01385*, 2019.

[11] D. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011.

[12] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[13] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. 2020.

[14] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018.

[15] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. 2022.

[16] K. Clark, M.-T. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le. Bam! born-again multi-task networks for natural language understanding. *arXiv preprint arXiv:1907.04829*, 2019.

[17] M. Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[21] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. *arXiv preprint arXiv:2112.06905*, 2021.

[22] H. Fang, P. Xiong, L. Xu, and Y. Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021.

[23] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021.

[24] M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei. Dynamic task prioritization for multitask learning. In *ECCV*, 2018.

[25] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*, 2016.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[27] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[28] C. Hokamp, J. Glover, and D. Gholipour. Evaluating the supervised and zero-shot performance of multi-lingual translation models. *arXiv preprint arXiv:1906.09675*, 2019.

[29] R. Hu and A. Singh. Unit: Multimodal multitask learning with a unified transformer. *arXiv preprint arXiv:2102.10772*, 2021.

[30] T. Iki and A. Aizawa. Effect of visual extensions on natural language understanding in vision-and-language models. *arXiv preprint arXiv:2104.08066*, 2021.

[31] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, A. Brock, E. Shelhamer, O. Hénaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver io: A general architecture for structured inputs & outputs, 2021.

[32] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021.

[33] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[34] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*, 2019.

[35] S. Kalkowski, C. Schulze, A. Dengel, and D. Borth. Real-time analysis and visualization of the yfcc100m dataset. In *Proceedings of the 2015 workshop on community-organized multimodal mining: opportunities for novel solutions*, pages 25–30, 2015.

[36] M. Kanakis, D. Bruggemann, S. Saha, S. Georgoulis, A. Obukhov, and L. V. Gool. Reparameterizing convolutions for incremental multi-task learning without task interference. 2020.

[37] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[38] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

[39] W. Kim, B. Son, and I. Kim. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv preprint arXiv:2102.03334*, 2021.

[40] Y. J. Kim, A. A. Awan, A. Muzio, A. F. C. Salinas, L. Lu, A. Hendy, S. Rajbhandari, Y. He, and H. H. Awadalla. Scalable and efficient moe training for multitask multilingual models. *arXiv preprint arXiv:2109.10465*, 2021.

[41] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.

[42] S. Kudugunta, Y. Huang, A. Bapna, M. Krikun, D. Lepikhin, M.-T. Luong, and O. Firat. Beyond distillation: Task-level mixture-of-experts for efficient inference. *arXiv preprint arXiv:2110.03742*, 2021.

[43] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[44] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

[45] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[46] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020.

[47] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2021. doi: 10.1109/TNNLS.2021.3084827.

[48] Z. Li, W. Wang, E. Xie, Z. Yu, A. Anandkumar, J. Alvarez, T. Lu, and P. Luo. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. 2022.

[49] Z. Lin, L. Wu, M. Wang, and L. Li. Learning language specific sub-network for multilingual machine translation. *arXiv preprint arXiv:2105.09259*, 2021.

[50] P. Liu, X. Qiu, and X. Huang. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*, 2017.

[51] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang. Gpt understands, too. *arXiv preprint arXiv:2103.10385*, 2021.

[52] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[53] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021.

[54] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

[55] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee. 12-in-1: Multi-task vision and language representation learning. In *CVPR*, 2020.

[56] S. Min, W. Kong, R.-C. Tu, D. Gong, C. Cai, W. Zhao, C. Liu, S. Zheng, H. Wang, Z. Li, et al. Hunyuan_tvr for text-video retrivial. *arXiv preprint arXiv:2204.03382*, 2022.

[57] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding. *TPAMI*, 2019.

[58] V. Ordonez, G. Kulkarni, and T. Berg. Im2text: Describing images using 1 million captioned photographs. *NeurIPS*, 2011.

[59] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

[60] D. Qi, L. Su, J. Song, E. Cui, T. Bharti, and A. Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *arXiv preprint arXiv:2001.07966*, 2020.

[61] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[62] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas. A generalist agent, 2022.

[63] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015.

[64] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. Susano Pinto, D. Keysers, and N. Houlsby. Scaling vision with sparse mixture of experts. *NIPS*, 2021.

[65] J. Shao, S. Chen, Y. Li, K. Wang, Z. Yin, Y. He, J. Teng, Q. Sun, M. Gao, J. Liu, et al. Intern: A new learning paradigm towards general vision. *arXiv preprint arXiv:2111.08687*, 2021.

[66] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[67] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[68] N. Shazeer, Y. Cheng, N. Parmar, D. Tran, A. Vaswani, P. Koanantakool, P. Hawkins, H. Lee, M. Hong, C. Young, et al. Mesh-tensorflow: Deep learning for supercomputers. *NIPS*, 31, 2018.

[69] S. Shen, L. H. Li, H. Tan, M. Bansal, A. Rohrbach, K.-W. Chang, Z. Yao, and K. Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

[70] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[71] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela. Flava: A foundational language and vision alignment model. *arXiv preprint arXiv:2112.04482*, 2021.

[72] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese. Which tasks should be learned together in multi-task learning? In *ICML*, 2020.

[73] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

[74] G. Strezoski, N. v. Noord, and M. Worring. Many task learning with task routing. In *ICCV*, 2019.

[75] H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[76] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[77] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[78] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[79] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[80] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*, 2022.

[81] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021.

[82] X. Wang, Y. Tsvetkov, and G. Neubig. Balancing training for multilingual neural machine translation. *arXiv preprint arXiv:2004.06748*, 2020.

[83] X. Wang, F. Yu, L. Dunlap, Y.-A. Ma, R. Wang, A. Mirhoseini, T. Darrell, and J. E. Gonzalez. Deep mixture of experts via shallow embedding. In *Uncertainty in artificial intelligence*. PMLR, 2020.

[84] Z. Wang, Z. C. Lipton, and Y. Tsvetkov. On negative interference in multilingual models: Findings and a meta-learning treatment. *arXiv preprint arXiv:2010.03017*, 2020.

[85] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

[86] B. Yang, G. Bender, Q. V. Le, and J. Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *NIPS*, 32, 2019.

[87] Z. Yang, Z. Gan, J. Wang, X. Hu, F. Ahmed, Z. Liu, Y. Lu, and L. Wang. Crossing the format boundary of text and boxes: Towards unified vision-language modeling. *arXiv preprint arXiv:2111.12085*, 2021.

[88] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[89] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. *NIPS*, 33:5824–5836, 2020.

[90] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[91] B. Zhang, A. Bapna, R. Sennrich, and O. Firat. Share or not? learning to schedule language-specific capacity for multilingual translation. In *ICLR*, 2020.

[92] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, and Z.-J. Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, pages 13278–13288, 2020.

[93] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, 2020.

[94] X. Zhu, J. Zhu, H. Li, X. Wu, X. Wang, H. Li, X. Wang, and J. Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. *arXiv preprint arXiv:2112.01522*, 2021.

[95] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, pages 19–27, 2015.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Our code will be released once the paper is accepted, and the used data is publicly available.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Most experiments have stable results with little variance.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite the open datasets that we use.
   (b) Did you mention the license of the assets? [Yes] See Section 4.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] Our code will be released once the paper is accepted.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Section 4.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Section 4.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]