Language Models are General-Purpose Interfaces

Yaru Hao[*] **Haoyu Song**[*] **Li Dong**[*]
Shaohan Huang, Zewen Chi, Wenhui Wang, Shuming Ma, Furu Wei[†]
Microsoft Research
https://github.com/microsoft/unilm

Abstract

Foundation models have received much attention due to their effectiveness across a broad range of downstream applications. Though there is a big convergence in terms of architecture, most pretrained models are typically still developed for specific tasks or modalities. In this work, we propose to use language models as a general-purpose interface to various foundation models. A collection of pretrained encoders perceive diverse modalities (such as vision, and language), and they dock with a language model that plays the role of a universal task layer. We propose a semi-causal language modeling objective to jointly pretrain the interface and the modular encoders. We subsume the advantages and capabilities from both causal and non-causal modeling, thereby combining the best of two worlds. Specifically, the proposed method not only inherits the capabilities of in-context learning and open-ended generation from causal language modeling, but also is conducive to finetuning because of the bidirectional encoders. More importantly, our approach seamlessly unlocks the combinations of the above capabilities, e.g., enabling in-context learning or instruction following with finetuned encoders. Experimental results across various language-only and vision-language benchmarks show that our model outperforms or is competitive with specialized models on finetuning, zero-shot generalization, and few-shot learning.

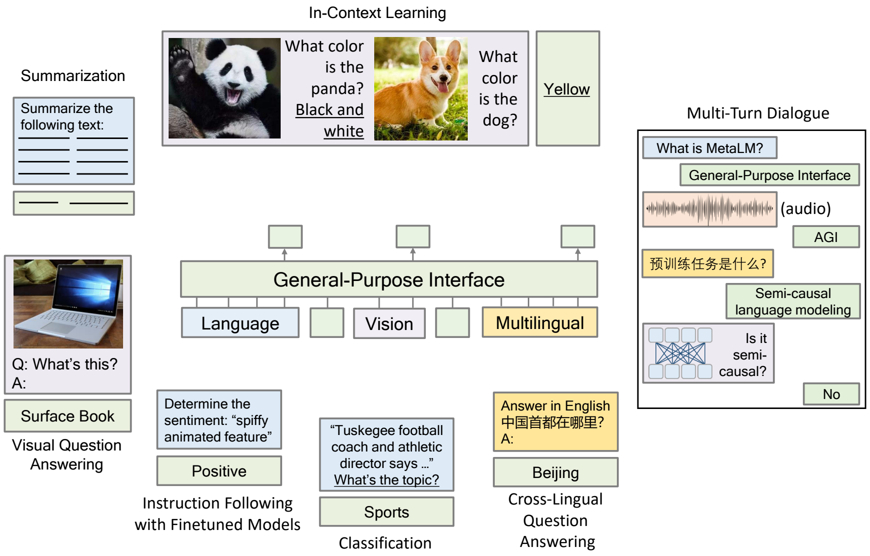

In-Context Learning
What color is the panda? Black and white
What color is the dog?
Yellow
Summarization
Summarize the following text:
Multi-Turn Dialogue
What is MetaLM?
General-Purpose Interface
(audio)
AGI
预训练任务是什么?
Semi-causal language modeling
Is it semi-causal?
No
General-Purpose Interface
Language
Vision
Multilingual
Q: What's this?
A:
Surface Book
Visual Question Answering
Determine the sentiment: "spiffy animated feature"
Positive
Instruction Following with Finetuned Models
"Tuskegee football coach and athletic director says …"
What's the topic?
Sports
Classification
Answer in English
中国首都在哪里?
A:
Beijing
Cross-Lingual Question Answering

Figure 1: Language models as a general-purpose interface to various foundation models.

[*] Equal contribution. [†] Corresponding author.

Contents

1 Introduction: Design Principles

Language models as a universal task layer. The large-scale language model serves as a general-purpose interface not only for language tasks, but also for vision, and multimodal tasks. Language models have open-ended output space, which generalizes to a wide range of tasks. As long as we can describe the predictions via natural language, the downstream task can fit in with language-model-based task layer. It is natural that transforming various predictions to free-text sequences (Raffel et al., 2020). For example, we can transform the target labels, and answers to texts for classification, and question answering, respectively. In addition, with the help of the universal task layer, the prediction process can go beyond single turn, i.e., a multi-turn dialogue interface can be built upon language models by conditioning on history context. Such unification of various tasks is important to general-purposed AI, which unifies representations, transformations, and expressions into a shared module.

Causal language modeling (i.e., unidirectional decoder) is conducive to zero-shot generalization and in-context learning. GPT-3 (Brown et al., 2020) has shown that the intriguing properties emerge from causal language model pretraining. Because of the favorable sample efficiency and inductive bias (Wang et al., 2022b) of causal language modeling (i.e., all tokens make predictions and produce supervision signals) compared with other counterparts (such as masked language modeling), it is effective to give models the desired properties via causal language modeling. The capabilities of zero- and few-shot learning are critical to be a general-purpose task layer. Zero-shot generalization indicates that language models have learned an enormous amount of world knowledge (Dai et al., 2021) and patterns by reading large-scale text corpora. The memorized information can serve as reusable background knowledge and basic skills for a wide range of end tasks. Moreover, in-context learning enables us to easily adapt either pretrained or finetuned models to new scenarios. For example, we can use task instructions (Ouyang et al., 2022) to repurpose the model, and use demonstrations of some examples to conduct few-shot learning.

Non-causal modeling (i.e., bidirectional encoder) is conducive to transfer across tasks, languages, and modalities. Although causal language models are good at zero- and few-shot generalization, BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) show that having bidirectional encoders pretrained by masked language modeling achieves much better finetuning performance. Once the whole input is given, non-causal modeling is quite rational for encoding data. Because all the context can access each other, while causal modeling can only make use of history tokens one by one. The advantage of finetuning is helpful for the data-rich setting where there are many annotated data available. In addition, non-causal encoder pretrained by the masked language modeling objective achieves competitive performance on cross-lingual transfer (Conneau et al., 2020), which makes it effective to adapt models to the multilingual setting.

Semi-causal language modeling as a meta-pretraining task. Semi-causal language modeling plays the role of linking together non-causal encoders and the causal language model. It is a meta task in the sense of universal interface pretraining of pretrained encoders. Specifically, non-causal encoders learn to represent various input data, and a causal language model serves as a universal task layer. Non-causal encoders dock with a causal language model, so that we can benefit from both modeling methods described as above. In comparison with previous encoder-decoder pretraining (such as prefix language modeling, and T5; Raffel et al. 2020), our task non-causally encodes random spans of the whole sequence, while generating the rest via causal language modeling. Moreover, in terms of architecture, we directly feed the outputs of bidirectional encoders into the causal decoder, rather than relying on cross attention (Vaswani et al., 2017). Besides, multiple bidirectional encoders can be mounted to the causal language model, but the encoder-decoder architecture usually has only one encoder.

Non-causal encoders as System 1, and causal language models as System 2. Cognition is usually categorized into two levels (Kahneman, 2011; Bengio, 2019): System 1 (i.e., intuitive, and unconscious) and System 2 (i.e., sequential, conscious, planning, and reasoning). In the proposed framework, the modules can be regarded as an implementation of these two levels, respectively. To be specific, non-causal encoders pretrained by masked data modeling, such as BERT (Devlin et al., 2019) and BEiT (Bao et al., 2022), are used as a perception layer to encode various input modalities. The encoding modules can be viewed as System 1. After we obtain the input representations, we

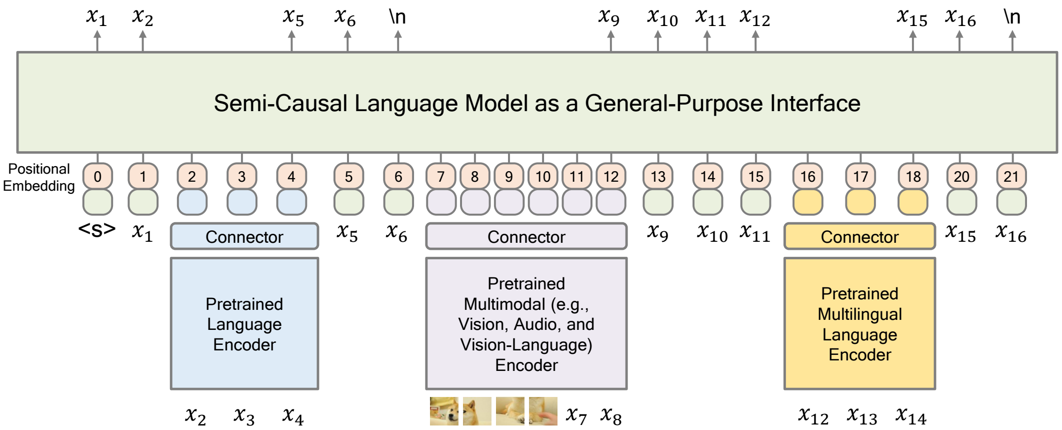

x_1 x_2 x_5 x_6 \n x_9 x_{10} x_{11} x_{12} x_{15} x_{16} \n
Semi-Causal Language Model as a General-Purpose Interface
Positional Embedding
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 21
<s> x_1 Connector x_5 x_6 Connector x_9 x_{10} x_{11} Connector x_{15} x_{16}
Pretrained Language Encoder
Pretrained Multimodal (e.g., Vision, Audio, and Vision-Language) Encoder
Pretrained Multilingual Language Encoder
x_2 x_3 x_4
x_7 x_8
x_{12} x_{13} x_{14}

Figure 2: Overview of METALM. The semi-causal language model serves as a general-purpose interface and supports interactions with various foundation models.

feed them to the causal language model, which has shown promising performance on commonsense reasoning (Chowdhery et al., 2022) and planning (Huang et al., 2022). The universal task layer is designed to play a role of System 2 in our method.

Natural language interface between users and pretrained models. The universal task layer based on causal language modeling enables users to interact with pretrained non-causal encoders using natural language. First, language can be used as a programming language for the underlying pretrained or finetuned models, which is compiled by the universal interface. For example, we can write text-based instructions (Ouyang et al., 2022) and explanations (Wei et al., 2022) to repurpose and guide the model behaviors. Second, the universal interface enables the models to present the results using free texts, making predictions directly understandable and explainable. Third, the proposed framework natively supports multi-turn conversational interactions. In each turn, we can feed the encoded input to the interface layer and then generate response results in a semi-causal manner.

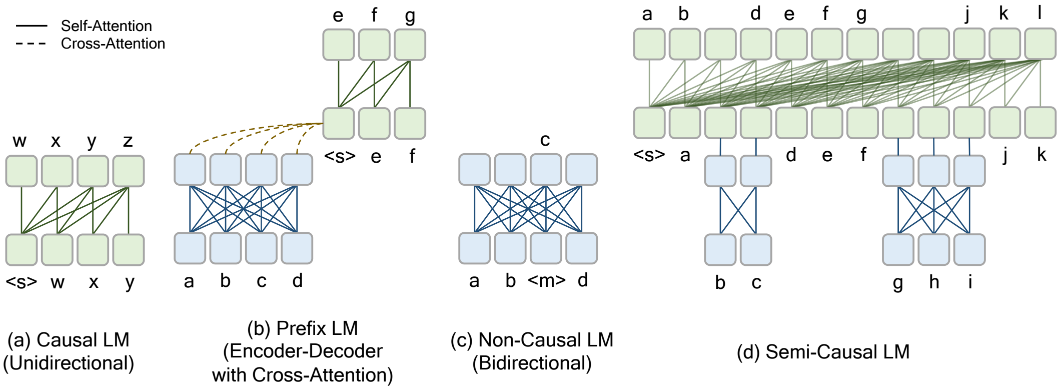
2 METALM: Meta Language Model

Guided by the design principles in Section 1, we present **Meta L**anguage **M**odel (METALM), a semi-causal language model that plays the role of a general-purpose interface and supports interactions with various foundation models. An overview of our framework is shown in Figure 2. Specifically, a collection of pretrained encoders, that perceive diverse modalities, dock with a language model. The language model is regarded as a universal task layer (i.e., general-purpose interface), which unifies various tasks as free-text generation.

In order to pretrain METALM, we propose a semi-causal language modeling task to jointly learn the modules. METALM subsumes the advantages and capabilities from both worlds. From the language model, METALM inherits the capabilities of in-context learning, multi-turn interaction, and open-ended generation. Moreover, the underlying foundation models are conducive to finetuning because of bidirectional modeling (Wang et al., 2022b).

2.1 Input Representation

Input representations of METALM are grouped into two categories. The first type is contextualized representations obtained by the underlying encoders and then projected by a connector layer. For example, as shown in in Figure 2, the image patches and x_7, x_8 are encoded by the bidirectional vision-language encoder. The second category is token embeddings of texts, such as x_5, and x_6 in Figure 2. The representations of these two categories are summed with positional embeddings before feeding into the general-purpose interface.



(a) Causal LM (Unidirectional)

(b) Prefix LM (Encoder-Decoder with Cross-Attention)

(c) Non-Causal LM (Bidirectional)

(d) Semi-Causal LM

Figure 3: Comparisons between different language model (LM) variants: (a) causal LM with unidirectional decoder (Brown et al., 2020); (b) prefix LM with encoder-decoder architecture (Raffel et al., 2020); (c) non-causal LM with bidirectional encoder (Devlin et al., 2019); (d) semi-causal LM proposed in this work.

2.2 Model Architecture

As shown in Figure 3, we summarize the model architectures of three language model variants and the proposed semi-causal language model. First, causal language model (such as GPT; Brown et al. 2020) is a left-to-right Transformer decoder. Second, prefix language model uses the encoder-decoder architecture with cross-attention connections to complete the sequence. Third, non-causal language model is a bidirectional encoder, which is usually pretrained by masked language modeling (Devlin et al., 2019). Forth, the proposed semi-causal language model has a unidirectional Transformer decoder, and multiple bidirectional encoders that dock with the decoder. In other words, our model processes the whole session from left to right, while having some spans pre-encoded by non-causal encoders.

Backbone Network We use Transformer (Vaswani et al., 2017) to build the models. Given an input sequence, we first pack their vector representations together. Then we feed the vectors into a multi-layer Transformer, which encodes the input to contextualized representations. In each Transformer block, there is a multi-head self-attention layer and a feed-forward network layer that are used to aggregate the hidden states of the previous layer. Moreover, attention masks are used to control the context access. We use a triangular matrix as the attention mask for the universal task layer, so that it processes the input from left to right. For the bidirectional encoder, we allow all the tokens to access each other. After obtaining the output vectors of the universal task layer, we use a softmax classifier to predict over the vocabulary. The weight matrix is shared with the input token embeddings.

Connector As shown in Figure 2, there is a connector layer between the universal task layer and various bidirectional encoders. The connectors project vector representations of bidirectional encoders before feeding them into the general-purpose interface. Moreover, the connectors are used to match the output dimensions of foundation models with the universal task layer. We empirically find that both linear projection and feed-forward network work well in our experiments.

2.3 Proposed Objective: Semi-Causal Language Modeling

In order to pretrain METALM, we introduce the semi-causal language modeling objective. As shown in Figure 2, our pretraining task autoregressively generates the tokens of a sequence, while some spans are represented by bidirectional encoders.

Given an input sequence $\boldsymbol{x} = x_1, x_2, ..., x_n$, we assume there are k non-causal spans denoted as $\{\boldsymbol{x}_{s_1}^{e_1}, ..., \boldsymbol{x}_{s_k}^{e_k}\}$, where $\boldsymbol{x}_{s_i}^{e_i} = x_{s_i}, ..., x_{e_i-1}$. For each non-causal span $\boldsymbol{x}_{s_i}^{e_i}$, we use a bidirectional encoder to obtain its vector representations $\boldsymbol{h}(\boldsymbol{x}_{s_i}^{e_i})$. The choose of bidirectional encoders is dependent on the modality of the non-causal span.

Then the semi-causal language modeling objective is formulated as:

$$\max \sum_{i=0}^{k} \sum_{t=e_i}^{s_{(i+1)}} \log P(x_t | \boldsymbol{x}_{<t}, \{\boldsymbol{h}(\boldsymbol{x}_{s_j}^{e_j})\}_{j<i}) \tag{1}$$

where $e_0 = 1$, $s_{(k+1)} = n$, and $\{\boldsymbol{h}(\boldsymbol{x}_{s_j}^{e_j})\}_{j<i} = \{\boldsymbol{h}(\boldsymbol{x}_{s_1}^{e_1}), \cdots, \boldsymbol{h}(\boldsymbol{x}_{s_{(i-1)}}^{e_{(i-1)}})\}$. Notice that the next token of each non-causal span is generated at the last position of the span. Typically the number of non-causal spans and their positions are randomly sampled. The spans do not have overlaps with each other.

By leveraging the proposed objective, we jointly pretrain the general-purpose interface and the underlying foundational models, and seamlessly connect them together. We pretrain MetaLM for both the language-only (Section 3) and vision-language (Section 4) settings.

2.4 Capabilities on Downstream Tasks

In-Context Learning MetaLM can adapt to a new task by conditioning on natural language instructions or several input-output pairs (i.e., demonstrations), without updating any parameter. We first describe the usage of k-shot learning. For each demonstration input, we conduct bidirectional encoding. Then we feed the encoded vectors and the label into the general-purpose interface. By conditioning on the given demonstrations, MetaLM predicts the target output of unseen examples. For zero-shot generalization, there is only the test input, typically with prompts used to describe the task. We feed the example with the task instruction into bidirectional encoders. The target output is generated by the universal task layer.

Finetuning Finetuning is especially helpful when many annotated examples of the downstream task are available. We unify various tasks to the open-ended generation format, i.e., targets are transformed to free texts. During finetuning, MetaLM learns to generate the target output, conditioning on the bidirectionally encoded input. Compared with causal language models, MetaLM inherits the excellent finetuning capability of bidirectional encoders.

In-Context Customization A typical usage is that we first finetune the model on a large amount of data, and then use in-context learning to customize the finetuned model. So we can easily transfer the knowledge of labeled data to new tasks. As we subsume the advantages of both causal and non-causal modeling, MetaLM unlocks the combinations of the capabilities, i.e., good finetuning performance of non-causal modeling, and in-context learning of causal modeling.

Multimodal Multi-Turn Interaction MetaLM supports multi-turn interactions between users and pretrained models. For each turn, non-causal modules encode user inputs, which accepts multimodal contents by using the corresponding pretrained encoders. The output responses are generated by the general-purpose interface. By conditioning on the history conversations, MetaLM naturally works as a conversational interface. Moreover, the conversation can include multiple modalities instead of plain texts.

3 Experiments on Language-Only Tasks

We first conduct experiments on language-only datasets to demonstrate the versatility and effectiveness of MetaLM. Here the non-causal encoder is a pretrained language foundation model that docks with the universal task layer. The intriguing capabilities emerge through pretraining, which enables the general-purpose interface to transfer across tasks and scenarios.

3.1 Evaluation Settings

We elaborate on language-only evaluation settings in Table 1. We demonstrate the capabilities of MetaLM, including multitask finetuning (Section 3.3), single-task finetuning (Section 3.4), instruction tuning (Section 3.5), and in-context learning (Section 3.6). The capabilities are task-agnostic and broadly applicable to understanding, generation, and interaction, which facilitates skill adaptation and communication with users. Moreover, the evaluation settings of multitask finetuning

Evaluation Setting	Capability
Multitask Finetuning	Perform a wide range of tasks competitively.
Single-Task Finetuning	Tackle individual tasks with remarkable performance.
Instruction Tuning	Zero-shot generalization after finetuning with instructions.
Zero-/Few-Shot Learning	Adapt to a new task given zero/few labeled examples.

Table 1: Summary of evaluation settings for language-only METALM. Each setting highlights an essential capability of METALM.

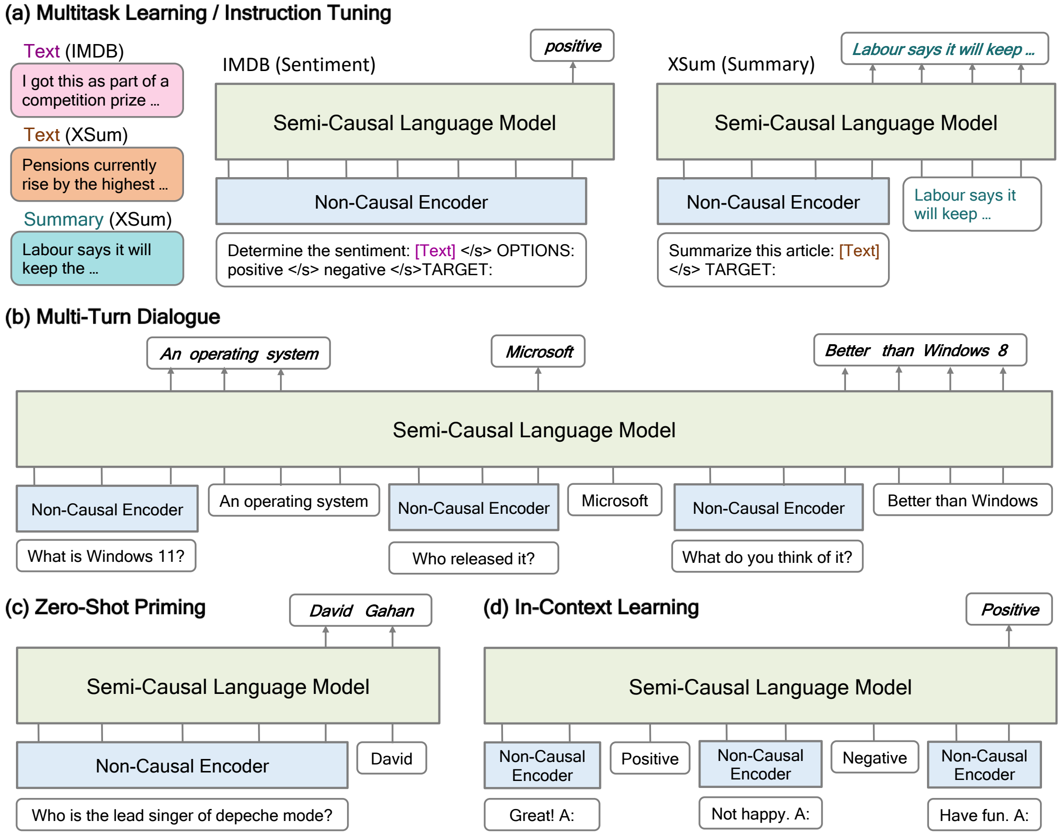


Figure 4: METALM can be applied in different language-only scenarios: (a) multitask finetuning and instruction tuning, i.e., perform various tasks simultaneously in an open-ended manner. (b) multi-turn dialogue, i.e., generate multi-turn responses according to the encoded input of users. (c) zero-shot priming, e.g., natural question answering. (d) few-shot learning, e.g., sentiment analysis.

and instruction tuning are seamlessly built upon the capability combination of finetuning and in-context learning. In addition, because the tasks are unified in the free-text format, we can handle diverse downstream tasks using the same interface.

Figure 4 illustrates how to apply our model to different scenarios. Generally, the input examples and instructions are fed to the non-causal language encoder, and the target outputs are produced from the universal task layer. Moreover, the predictions are generated in a generative manner, which is open-ended.

3.2 Pretraining Setup

We use sinusoidal position embeddings (Vaswani et al., 2017) for the language model. The number of layers is $L = 24$, each layer consists of $A = 32$ attention heads and the hidden dimension is

$H = 2048$. The number of parameters is about 1.3B. For the non-causal part, we use encoder-only Transformers, where $A = 16$, $H = 1024$, $L = 24$. We utilize the learnable position embedding and relative position bias (Raffel et al., 2020) for the non-causal model. The number of parameters is about 366M. We use DeepNorm (Wang et al., 2022a) for Transformers. The connector module is a linear projection layer in our implementation.

The maximum input lengths for non-causal and semi-causal models are 512 and 2048, respectively. We randomly sample random spans whose lengths are between 64 and 128, and feed them to the non-causal part. The total length of non-causal spans is 25% of the original sequence length. The spans do not cross document boundaries. We pretrain the semi-causal language model from scratch. The non-causal module is initialized from a pretrained bidirectional encoder, using the replaced token detection task (Clark et al., 2020). During pretraining, we freeze all parameters of the non-causal encoder except the last two layers. We pretrain METALM for 300k steps with a batch size of 1024 and use Adam (Kingma and Ba, 2015) for optimization. We disable dropout of the semi-causal model and set the dropout rate of the non-causal model to 0.1. We use a learning rate of 6e-4 with warm-up. Please refer to Appendix A.1 for more pretraining details.

We pretrain the model on Pile (Gao et al., 2021), which is a massive English text dataset constructed from diverse data sources and targeted at training large-scale language models. We exclude data splits of GitHub, arXiv, and PubMed Central. Please refer to Appendix B.1 for detailed descriptions about Pile. The pretraining data is tokenized by SentencePiece (Kudo and Richardson, 2018). We construct the input in the "full-sentence" format (Liu et al., 2019b), i.e., each input sequence is packed with full sentences sampled contiguously from one or more documents. We additionally introduce three special tokens for input construction: `<s>` indicates the start of a sequence, `</s>` indicates the end of a paragraph and `</d>` indicates the end of a document.

3.3 Multitask Finetuning

We first evaluate METALM under the multitask finetuning setting. To be specific, we unify a wide range of tasks in an open-ended generation manner, so that they can be processed by the universal task layer without any task-specific architecture. Figure 4(a) shows an example of how METALM handles multitask finetuning. During finetuning, we randomly sample training examples and feed the inputs into the bidirectional language encoder. The finetuning objective is to maximize the likelihood of the correct labels generated from the interface.

We conduct experiments on a mixture of 34 NLP datasets (refer to Appendix B.2 for more details) grouped into ten task clusters, including both language understanding tasks and generation tasks:

- **Natural Language Inference**: ANLI (R1-R3), CB, MNLI, QNLI, RTE, SNLI, WNLI
- **Sentiment Classification**: IMDB, SST-2, Sentiment140, Yelp
- **Paraphrase Detection**: QQP, MRPC, Paws Wiki
- **Coreference Resolution**: DPR, Winogrande, WSC
- **Commonsense Reasoning**: HellaSwag, PiQA, COPA
- **Reading Comprehension**: DROP, SQuADv1, SQuADv2, OBQA, BoolQ
- **Miscellaneous**: CoLA, WiC, TREC
- **Closed-Book QA**: ARC-easy, NQ
- **Struct to Text**: CommonGen, E2ENLG
- **Summarization**: AESLC, SamSum, XSum

3.3.1 Evaluation Setup

METALM is finetuned on a mixture of all the mentioned datasets. We limit the maximum number of training examples in each dataset to 30k. We follow the prompts used in (Wei et al., 2021). If the dataset is a multi-choice task, all possible options are provided in the template. For instance, the input format of an example from a sentiment classification dataset is "`<s>` *Would the following phrase be considered positive or negative?* `</s>` [text] `</s>` *OPTIONS:* `</s>` *Positive* `</s>` *Negative* `</s>` *TARGET:*". The model determines the sentiment by generating *Positive* or *Negative*.

	Task Cluster	**GPT**	**METALM**
NLU	Natural Language Inference	65.0	**79.1**
	Sentiment	92.9	**94.6**
	Paraphrase	83.9	**89.6**
	Coreference	67.1	**84.3**
	Commonsense Reasoning	63.3	**84.2**
	Reading Comprehension	64.5	**73.1**
	Miscellaneous	80.3	**84.3**
NLG	Closed-Book QA	38.2	**44.3**
	Struct to Text	**44.2**	44.1
	Summarization	29.8	**31.0**

Table 2: Performance comparisons of multitask finetuning between METALM and GPT. We limit the number of training examples in each dataset to 30k during finetuning. For each task cluster, we present the average result over all sub-datasets within it. All results are reported on validation sets.

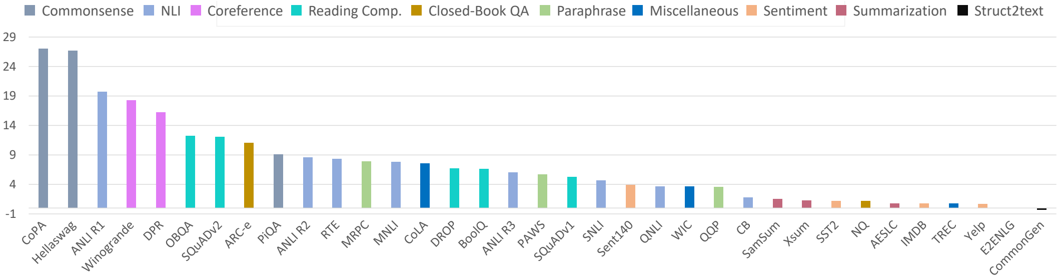


Figure 5: Score difference of multitask finetuning results between METALM and GPT. We observe that METALM achieves consistent improvements over all tasks except the cluster of struct to text.

We finetune METALM for 20k steps with a batch size of 256. The total length of input and answer tokens is restricted to 2048. Following (Raffel et al., 2020), we pack multiple training examples into one sequence to make computation batch-friendly. The learning rate is set to 1e-4. For more details, please refer to Appendix A.2.

For multi-choice tasks, we report the exact match score without decoding constraints. For SQuAD, DROP, and closed-book QA datasets, we report the F1 score with greedy decoding. When evaluating on the struct2text and summarization clusters, we use beam search (Sutskever et al., 2014) with a beam size of 4 and a length penalty of $\alpha = 0.6$. We report ROUGE scores for the above two clusters.

3.3.2 Results

Table 2 compares the multitask finetuning results of METALM and GPT. The GPT baseline follows the same configuration and training corpus for a fair comparison. Each result represents the average score of all datasets of one task cluster. The full results of all task clusters are reported in Appendix C. We also illustrate the score differences between METALM and GPT for all datasets in Figure 5.

We observe that METALM consistently surpasses GPT by a large margin on almost all the task clusters. The results indicate that our method inherits the performant finetuning ability from the non-causal encoder. Particularly, METALM performs much better than GPT on NLU tasks. It partially confirms that non-causal modeling is conducive to finetuning (Wang et al., 2022b; Tay et al., 2022; Artetxe et al., 2022). For more challenging tasks, such as natural language inference, and reading comprehension, the improvement of METALM is very prominent (14.1% and 9.6%). Furthermore, we find that finetuning of GPT brings relatively small gains on commonsense reasoning tasks, whose results are comparable to zero-shot generalization. By contrast, finetuning of METALM obtains decent gains over zero-shot numbers. With regard to language generation, METALM consistently outperforms GPT except on struct-to-text datasets. For closed-book question answering and text summarization, METALM achieves better performance than GPT too, benefiting from the non-causal modeling of input text.

Model	**MNLI** (acc)	
	-m	-mm
GPT	87.7	87.6
BERT (Devlin et al., 2019)	86.6	-
RoBERTa (Liu et al., 2019b)	90.2	90.2
ELECTRA (Clark et al., 2020)	90.9	-
METALM	**91.1**	**91.0**

Table 3: Single-task finetuning results on matched (-m) and mismatched (-mm) validation sets of MNLI. Each score is the average of multiple runs with different random seeds.

3.4 Single-Task Finetuning

We explore the finetuning capability of METALM under data-rich settings. We design a new finetuning paradigm for METALM. For each downstream task, we only update the parameters of the non-causal encoder while keeping the language model frozen. We demonstrate that the proposed strategy achieves excellent performance, and preserves the general-purpose interface's capabilities of in-context learning and open-endedness.

3.4.1 Finetuning Setup

We conduct single-task finetuning on the natural language inference dataset MNLI (Williams et al., 2018). We use the template "`<s>` *Premise:*[*] `</s>` *Hypothesis:*[*] `</s>` *Label:*". The task is to determine whether a hypothesis is true, false or undetermined given a premise. The corresponding labels are "*entailment*", "*contradiction*" and "*neutral*", respectively. During finetuning, we freeze the general-purpose interface and only update the non-causal encoder and the connector. In contrast, all parameters are updated for the GPT baseline. We finetune both METALM and GPT for three epochs with a learning rate of 5e-5 and a batch size of 32.

3.4.2 Results

Table 3 reports single-task finetuning accuracy. MNLI-m and -mm represent the matched and the mismatched validation sets respectively. Each score is the average of three runs with different random seeds. Compared with GPT, METALM improves the accuracy of MNLI by 3.4 absolute points, despite updating much fewer parameters. In addition to Section 3.3, the results show that bidirectional encoders benefit finetuning performance (Wang et al., 2022b; Tay et al., 2022; Artetxe et al., 2022). Furthermore, we also present three strong baselines derived from finetuning bidirectional language encoders, including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b) and ELECTRA (Clark et al., 2020). All these three models are in large size. Results show that METALM achieves comparable or better performance than the bidirectional encoders.

3.5 Instruction-Tuned Zero-Shot Generalization

We investigate instruction tuning for METALM, which finetunes the model on a variety of tasks with instructions. After finetuning, we evaluate the performance of instruction following and zero-shot generalization for the models. Because our goal is to investigate the zero-shot generalization on held-out tasks. Therefore, when evaluating on a specific dataset, all datasets in the same category (i.e., task cluster) are not seen during the training stage. For example, if we evaluate on the classification dataset SST-2, the entire cluster of sentiment analysis is excluded during instruction tuning.

3.5.1 Instruction-Tuning Setup

We follow the evaluation pipeline proposed in FLAN (Wei et al., 2021). We conduct instruction tuning with METALM and GPT on the same dataset mixture described in Section 3.3 except for the summarization cluster. For each dataset, we use ten different templates manually composed by FLAN (Wei et al., 2021) and randomly apply one of them for every example. As mentioned in (Wei et al., 2021), there are some templates that "turned the task around" to increase learning diversity,

	Avg template		**Best template**	
	GPT	METALM	GPT	METALM
Natural Language Inference				
ANLI R1	31.6$_{0.6}$	**36.2**$_{2.5}$	32.5	**40.5**
ANLI R2	33.4$_{0.5}$	**36.3**$_{1.2}$	34.0	**38.2**
ANLI R3	35.9$_{1.3}$	**38.9**$_{0.9}$	37.8	**39.8**
CB	60.3$_{4.3}$	**75.0**$_{7.9}$	66.1	**83.9**
MNLI-m	45.8$_{2.6}$	**51.0**$_{1.7}$	48.5	**52.3**
QNLI	59.3$_{0.7}$	**66.1**$_{1.3}$	60.6	**68.0**
RTE	61.0$_{2.0}$	**70.2**$_{3.3}$	64.3	**75.5**
SNLI	41.6$_{4.8}$	**52.1**$_{4.3}$	49.8	**58.1**
WNLI	53.2$_{2.5}$	**65.1**$_{3.9}$	56.3	**71.8**
Average	46.9	**54.5**	50.0	**58.7**
Sentiment				
IMDB	84.6$_{2.6}$	**85.8**$_{2.9}$	87.2	**89.6**
SST-2	77.8$_{6.1}$	**81.4**$_{6.4}$	83.9	**89.9**
Sent140	85.4$_{1.1}$	**86.4**$_{1.7}$	87.2	**88.3**
Yelp	84.1$_{10.8}$	**91.0**$_{1.7}$	**93.2**	92.9
Average	83.0	**86.2**	87.9	**90.2**
Paraphrase				
QQP	**60.7**$_{0.7}$	59.7$_{2.1}$	61.6	**62.1**
MRPC	62.6$_{1.6}$	**68.4**$_{0.5}$	65.2	**69.1**
Average	61.7	**64.1**	63.4	**65.6**
Reading Comprehension				
DROP	**18.1**$_{0.4}$	13.7$_{0.5}$	**18.7**	14.5
SQuADv1	51.6$_{3.0}$	**60.4**$_{1.5}$	55.6	**62.7**
SQuADv2	24.9$_{1.3}$	**28.7**$_{1.8}$	27.1	**30.2**
OBQA	28.4$_{1.3}$	**36.2**$_{1.4}$	30.0	**38.8**
BoolQ	51.7$_{3.8}$	**53.5**$_{2.6}$	**57.8**	56.7
Average	34.9	**38.5**	37.8	**40.6**

Table 4: Full results of instruction tuning. We report the accuracy for all datasets except using F1 score for DROP, SQuADv1, and SQuADv2. The average score of each dataset is computed across five different templates.

e.g., for sentiment classification, the model is prompted to generate a movie review based on the given sentiment label "*Positive*".

Most finetuning configurations are the same as in Section 3.3.1. We experiment on four task clusters, including natural language inference, sentiment classification, paraphrase detection, and reading comprehension. Following the evaluation protocol of (Wei et al., 2021), the paraphrase cluster is dropped when evaluating on inference cluster and vice-versa. We finetune METALM and GPT for 30k steps with a batch size of 512. The learning rate is set to 1e-4. The sequence length for each example is limited to 1024. We also use the data packing strategy as in Section 3.3 to improve efficiency. The detailed hyper-parameters is provided in Appendix A.2.

3.5.2 Results

Table 4 reports the full results of instruction tuning on four task clusters. For each dataset, we use five different templates for evaluation, and present both the average and the best score. We observe that METALM achieves large improvements over the GPT baseline, which indicates the effectiveness of semi-causal language modeling. Considering the natural language inference cluster, GPT fails to obtain reasonable zero-shot results on difficult datasets (such as ANLI and WNLI), while METALM consistently performs well on various datasets. We notice similar trends on the other task clusters, i.e., sentiment, paraphrase, and reading comprehension. In addition to the average results, METALM outperforms the GPT baseline in terms of the best performance.

Task	**k=0**		**k=1**		**k=4**	
	GPT	MetaLM	GPT	MetaLM	GPT	MetaLM
StoryCloze	72.4	**73.1**	72.5	**74.2**	72.5	**73.6**
HellaSwag	52.9	**53.5**	51.8	**52.7**	51.8	**52.7**
Winograd	71.9	**75.8**	73.0	**75.8**	71.9	**76.8**
Winogrande	**57.2**	56.1	55.2	**56.8**	**56.4**	**56.4**
ARC-e	50.6	**52.6**	**53.1**	51.1	54.3	**56.1**
ARC-c	28.8	**31.2**	**28.5**	**28.5**	**29.5**	**29.5**
PIQA	**73.1**	72.3	**73.6**	72.2	**73.1**	71.9
BoolQ	62.1	**62.2**	57.6	**57.9**	**61.5**	61.3
Copa	**70.0**	67.0	**69.0**	**69.0**	**71.0**	70.0
Average	59.9	**60.4**	59.4	**59.8**	60.2	**60.9**

Table 5: Performance comparisons of in-context learning between MetaLM and GPT. k represents the number of shots.

The setting of instruction tuning requires the capabilities of both finetuning and zero-shot generalization. Experimental results indicate that our method combines the best of causal and non-causal language models. MetaLM not only achieves favorable finetuning performance because of bidirectional encoders, but also retains the causal language model's intriguing capability of zero-shot generalization.

3.6 In-Context Learning

We compare the performance of in-context learning (Brown et al., 2020) between MetaLM and GPT. Conditioned on the task instruction and several input-label pairs, language models are repurposed towards the desired downstream task, following the input pattern while without updating parameters. As illustrated in Figure 4(d), the demonstrations consist of two parts, the example input is passed through the non-causal encoder and the label token uses original embeddings. Then the target label of the test input is generated by the universal task layer.

3.6.1 Evaluation Setup

We conduct experiments under zero-shot, one-shot, and four-shot settings. We follow the evaluation protocol of GPT-3 (Brown et al., 2020). We evaluate each test example by randomly sampling examples from the training set as demonstrations. The Winograd only has the test set, so we sample demonstrations directly from it. Under few-shot settings, all examples are delimited by the separator token `</s>`.

We evaluate MetaLM and the GPT baseline on nine tasks, including cloze and completion tasks (i.e, StoryCloze, HellaSwag), Winograd-style tasks (i.e, Winograd, Winogrande), commonsense reasoning (i.e, ARC-easy, ARC-challenge, PIQA), and two datasets BoolQ and Copa from the SuperGLUE benchmark (Wang et al., 2019). The detailed descriptions of these datasets are provided in Appendix B.3.

3.6.2 Results

Table 5 reports accuracy results of in-context learning. Compared with GPT, MetaLM achieves better or comparable results. For Winograd and completion tasks (i.e, StoryCloze, and HellaSwag), the performance of MetaLM has consistent improvements over GPT. Considering the average result over these datasets, MetaLM is better in both zero-shot ($k = 0$) and few-shot ($k = 1, 4$) settings. The findings indicate that MetaLM inherits the excellent in-context learning ability, and the contextualized representations of non-causal encoders tend to help the model to generalize better.

Dataset	**Task description**	**Metric**	**Zero-shot**	**In-context**	**Finetuning**
VQAv2	Visual question answering	VQA acc.	✓	✓	✓
OK-VQA	Knowledge-based VQA	VQA acc.	✓	✓	✓
VQA Karpathy	Visual question answering	VQA acc.			✓
COCO Caption	Image captioning	CIDEr, etc.	✓		✓
Flickr30k Caption	Image captioning	CIDEr, etc.	✓		
NoCaps	Image captioning	CIDEr, etc.	✓		
NLVR^2	Visual reasoning	acc.			✓
E-SNLI-VE label	Visual reasoning	acc.			✓
E-SNLI-VE explanation	Explanation generation	CIDEr, etc.			✓

Table 6: Evaluation summary of the vision-language datasets. We evaluate the capabilities of zero-shot, in-context learning, and finetuning.

4 Experiments on Vision-Language Tasks

We conduct experiments under the vision-language setting. The underlying non-causal encoder is a pretrained vision-language foundation model, which docks with the general-purpose interface. The pretraining task is similar to the language-only setting, despite the use of image-text pairs. Specifically, given an image-text pair, the image tokens are prepended to the text tokens. As shown in Figure 2, the non-causal encoder produces bidirectional fused representations of the image and a text prefix of random length. The causal decoder is pretrained to autoregressively predict the remaining tokens conditioning on the bidirectional fused representations. Text-only data is also leveraged and follows the same preparation protocol. We jointly pretrain on both image-text data and text-only data during the vision-language METALM pretraining.

4.1 Evaluation Settings

Table 6 summarizes what capabilities we would like to evaluate and the corresponding vision-language datasets. We conduct experiments on zero-shot generalization in Section 4.3, in-context learning in Section 4.4, and finetuning in Section 4.5. The tasks can be grouped into several categories, i.e., visual question answering, visual reasoning, image captioning, and explanation generation. The evaluation across nine datasets covers both understanding and generation.

Figure 6 illustrates how we evaluate METALM in different settings. The input image and prompts are fed to a vision-language encoder, while the target output is generated by the language model. All the tasks are formulated in an open-ended generative manner.

4.2 Pretraining Setup

We use a 12-layer non-causal vision-language encoder and a 24-layer language model. The universal task layer follows the same network architectures and configurations of GPT-2 (Radford et al., 2019). The hidden size is 1024, and there are 16 attention heads. We employ sinusoidal position embeddings (Vaswani et al., 2017). The number of parameters is 353M. For the non-causal encoder, we use a vision-language model pretrained as in VLMo (Wang et al., 2021). The number of parameters is 192M. We use 224x224 resolution during pretraining for images. The connector is a three-layer feed-forward network. More details about hyper-parameters can be found in Appendix D.1.

We pretrain METALM for 350k steps with 256 batch size. We use AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The learning rate is 1e-4 and weight decay is 0.01. We use linear decay and apply warm-up at the first 2,500 steps. The dropout rate is set to 0.1.

We pretrain METALM using image-text pairs and text documents. For image-text pairs, our pretraining data consists of Conceptual Captions (Sharma et al., 2018), Visual Genome (Krishna et al., 2017), COCO Caption (Chen et al., 2015), and SBU Caption (Ordonez et al., 2011) datasets. Together, there are about 4M images and 10M image-text pairs. For text documents, following (Liu et al., 2019b) and (Radford et al., 2019), we use the OpenWebText (Gokaslan and Cohen, 2019) corpus, which is an open-source recreation of the Reddit web text, as the pretraining data.

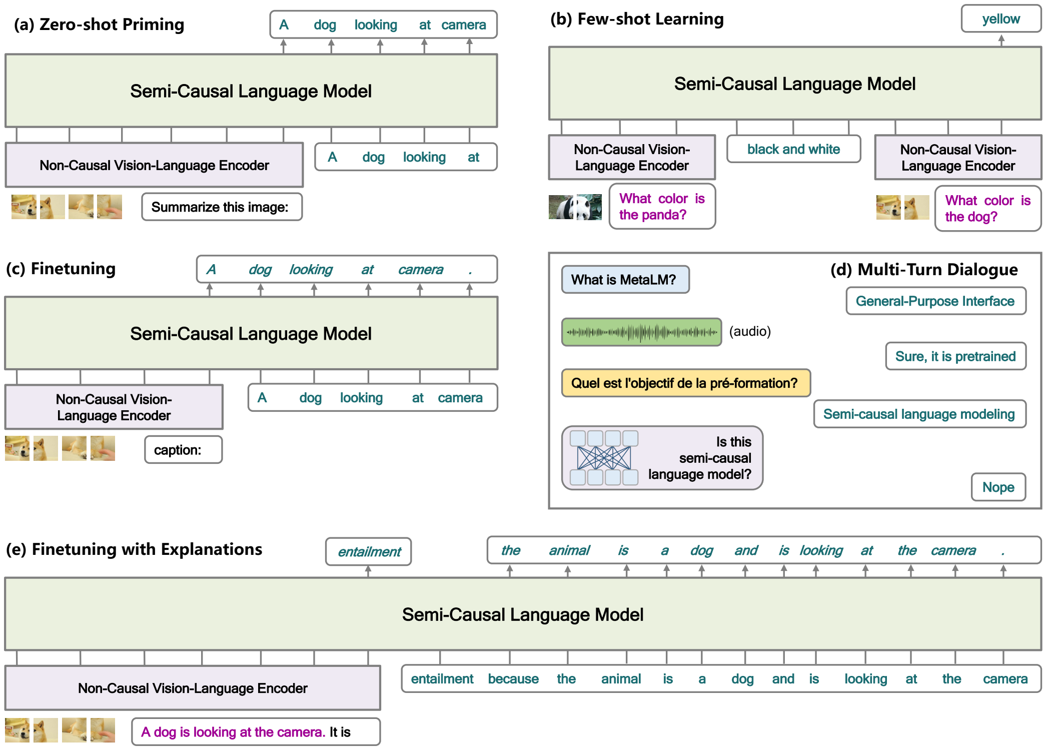

(a) Zero-shot Priming
A dog looking at camera
Semi-Causal Language Model
Non-Causal Vision-Language Encoder
A dog looking at
Summarize this image:
(b) Few-shot Learning
yellow
Semi-Causal Language Model
Non-Causal Vision-Language Encoder
black and white
Non-Causal Vision-Language Encoder
What color is the panda?
What color is the dog?
(c) Finetuning
A dog looking at camera .
Semi-Causal Language Model
Non-Causal Vision-Language Encoder
A dog looking at camera
caption:
What is MetaLM?
(d) Multi-Turn Dialogue
General-Purpose Interface
(audio)
Sure, it is pretrained
Quel est l'objectif de la pré-formation?
Semi-causal language modeling
Is this semi-causal language model?
Nope
(e) Finetuning with Explanations
entailment
the animal is a dog and is looking at the camera .
Semi-Causal Language Model
Non-Causal Vision-Language Encoder
entailment because the animal is a dog and is looking at the camera
A dog is looking at the camera. It is

Figure 6: The METALM's capabilities include: (a) zero-shot priming, e.g., zero-shot image captioning with language prompts. (b) few-shot learning, e.g., visual question answering with in-context learning. (c) finetuning on different downstream tasks, e.g., image captioning, visual reasoning, etc. (d) multi-turn conversational interactions. (e) finetuning with explanations, i.e., using natural language explanations to guide the task learning.

4.3 Zero-Shot Generalization

We evaluate the zero-shot generalization capability of METALM under vision-language settings. Specifically, we conduct experiments on two tasks, including image captioning, and visual question answering. For image captioning, only an input image is given, and the goal is to generate its description. For visual question answering, a question is asked for the given image, and the model needs to predict the correct answers.

4.3.1 Evaluation Setup

We apply greedy decoding during inference. The input images are resized to 224x224. We describe the datasets and specific setups of two tasks as follows:

Image Captioning We evaluate zero-shot caption generation on MS COCO Caption (Chen et al., 2015), NoCaps (Agrawal et al., 2019), and Flickr30k (Young et al., 2014). We evaluate on the test set of COCO *Karpathy split* (Karpathy and Fei-Fei, 2017), which re-partitions the train2014 and val2014 images (Lin et al., 2014) into 113,287, 5,000, and 5,000 for train, validation, and test. For NoCaps and Flickr30k, following (Jin et al., 2022), we evaluate on their validation set and test set, respectively. We use BLEU (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), METEOR (Banerjee and Lavie, 2005), and SPICE (Anderson et al., 2016) as caption generation metrics. We utilize COCOEvalCap[2] to compute scores. We prompt METALM with "*Summarize this image:*" for all zero-shot caption generation experiments.

Visual Question Answering Following (Tsimpoukelli et al., 2021), we evaluate the zero-shot performance on VQAv2 (Goyal et al., 2017) validation set and OK-VQA (Marino et al., 2019) test

[2] https://github.com/tylin/coco-caption

Model	COCO Caption Karpathy Test			
	BLEU-4	CIDEr	METEOR	SPICE
ZeroCap (Tewel et al., 2021)	2.6	14.6	11.5	5.5
VLKD$_{\text{ViT-B/16}}$ (Dai et al., 2022)	16.7	58.3	19.7	13.4
MetaLM	**24.5**	**82.2**	**22.5**	**15.7**

Table 7: Zero-shot generalization on COCO image captioning.

Model	NoCaps		Flickr30k	
	CIDEr	SPICE	CIDEr	SPICE
VL-T5 (Cho et al., 2021)	4.4	5.3	2.6	2.0
FewVLM (Jin et al., 2022)	42.2	8.5	31.0	10.0
MetaLM	**58.7**	**8.6**	**43.3**	**11.7**

Table 8: Zero-shot image captioning results on NoCaps validation and Flickr30k test. All the results are from their *base* size models, and the numbers are taken from (Jin et al., 2022).

set. VQA score is calculated using normalization rules of the VQAv2 evaluation code.[3] Different from classification over a predefined set of candidate answers, MetaLM predicts answers in an open-ended generation manner. We prompt MetaLM with the template"question: *question text* answer:" for all visual question answering experiments.

4.3.2 Results

Table 7 and Table 8 show the zero-shot captioning results on COCO Karpathy test split, NoCaps validation set, and Flickr30k test set. MetaLM outperforms recent strong methods on three image captioning datasets. To be specific, the compared model FewVLM (Jin et al., 2022) leverages different prompts for image captioning, and we report its best results. By contrast, we use the same prompt "*Summarize this image:*" for comparisons in all the experiments. Our model robustly follows the instruction to produce readable captions in the zero-shot manner.

Table 9 reports the results of zero-shot visual question answering on VQAv2 and OK-VQA. On both datasets, MetaLM achieves better zero-shot results than Frozen (Tsimpoukelli et al., 2021) and VLKD (Dai et al., 2022), even though Frozen has significantly more parameters. In addition, the OK-VQA dataset is designed for visual question answering that is supposed to require external knowledge. For example, the input image is a train, and the asked question is "*When is it invented?*". The reasonable performance on OK-VQA indicates that the language model of MetaLM tends to serve as a knowledge source. Once object information is perceived by the vision encoder, the universal task layer generates the answer as language modeling.

The experimental results across five datasets show that MetaLM has the capabilities of zero-shot generalization and open-ended generation. We can use prompts to re-purpose the pretrained vision-language model to image captioning and visual question answering.

4.4 In-Context Learning

We evaluate the capability of in-context learning (Brown et al., 2020) on visual question answering. We conduct k-shot learning, where k demonstrations are used to guide the prediction of new examples without finetuning the parameters.

4.4.1 Evaluation Setup

Following (Tsimpoukelli et al., 2021), we carry out few-shot experiments on the VQAv2 (Goyal et al., 2017) validation set and OK-VQA (Marino et al., 2019) test set. We randomly sample up to four full

[3] https://github.com/GT-Vision-Lab/VQA

Model	VQAv2	OK-VQA
Frozen (Tsimpoukelli et al., 2021)	29.5	5.9
VLKD$_{\text{ViT-B/16}}$ (Dai et al., 2022)	38.6	10.5
MetaLM	**41.1**	**11.4**

Table 9: Zero-shot generalization on visual question answering. All models predict in a generative manner without additional information, such as captions and object tags.

Model	VQAv2		OK-VQA	
	k=1	k=4	k=1	k=4
Frozen (Tsimpoukelli et al., 2021)	35.7	38.2	9.7	12.6
MetaLM	**42.4**	**45.3**	**13.2**	**16.0**

Table 10: In-context learning on visual question answering. All models predict in a generative manner without additional information, such as captions and object tags. k is the number of in-context examples (Brown et al., 2020) that the model can learn from.

examples from the training set for each test instance. The predicted answers are evaluated against the ground-truth answers following the normalization rules from the VQAv2 evaluation code. We use an image resolution of 224x224 during inference.

As shown in Figure 6(b), we put several examples before the test input and directly obtain the prediction from the universal task layer. Specifically, a full example is denoted as $e = [i, q, a]$, where *i*, *q*, *a* denote image, question, and answer, respectively. Similarly, a test input *t* is denoted as $t = [i, q]$. For k-shot in-context learning, the whole input sequence is $e_1, ..., e_k, t$. Moreover, we use "*Question: [question text] Answer:*" as the prompt to instruct MetaLM. Then MetaLM uses greedy decoding to generate answers.

4.4.2 Results

Table 10 reports the in-context learning results on the visual question answering datasets VQAv2 and OK-VQA. The results show that adding in-context demonstrations improves the performance over zero-shot generalization as shown in Table 9. Besides, adding more examples brings larger improvements to both datasets. Compared with Frozen (Tsimpoukelli et al., 2021), MetaLM obtains better performance despite the use of relatively small model size. We find that MetaLM can conduct in-context learning on visual question answering without modifying the underlying vision-language model. Although the non-causal encoder only sees one example each time, the language model successfully adapts the model according to the k demonstrations. In addition, with the help of the universal task layer, we can augment the existing foundation models with the general capability of in-context learning.

4.5 Finetuning on Downstream Tasks

We finetune the pretrained MetaLM on a wide range of vision-language tasks, including image captioning (Karpathy and Fei-Fei, 2017), visual question answering (Goyal et al., 2017; Marino et al., 2019), visual reasoning (Suhr et al., 2019), and explainable visual reasoning (Kayser et al., 2021). We compare the finetuned MetaLM with both the strong discriminative models and recent generative models.

4.5.1 Finetuning Setup

For all tasks, we use the resolution of 384x384 during finetuning. We also apply RandAugment (Cubuk et al., 2020) for image augmentation. We keep the learning rate 1e-5 fixed for all datasets. More detailed hyper-parameters can be found at Appendix D.2. We describe the setups of various tasks as follows.

Model	**VQAv2**		**VQA Karpathy-test**		**NLVR^2**
	test-dev	test-std	In-domain	Out-domain	test-P
Discriminative Prediction					
ViLBERT (Lu et al., 2019)	70.6	70.9	-	-	-
Oscar (Li et al., 2020)	73.2	73.4	-	-	78.4
UNITER (Chen et al., 2020)	72.3	72.9	74.4	10.0	77.9
Generative Prediction					
VL-T5 (Cho et al., 2021)	-	70.3	71.4	13.1	73.6
VL-BART (Cho et al., 2021)	-	71.3	72.1	13.2	70.3
VLKD$_{\text{ViT-B/16}}$ (Dai et al., 2022)	69.8	-	69.2	18.6	-
METALM	**74.4**	**74.5**	**77.9**	**21.1**	**80.9**

Table 11: Comparison of finetuning results on different vision-language tasks. The discriminative manner predicts a distribution over a pre-defined set of labels, e.g., 3129 most common answers for VQAv2. In contrast, the open-ended generative manner handles all tasks with free-text generation. Notice that all the reported results are from their *base* size models.

Visual Question Answering We evaluate on VQAv2 (Goyal et al., 2017), VQA Karpathy split (Cho et al., 2021), and OK-VQA (Marino et al., 2019). For VQAv2, models are finetuned on the training and validation sets. We report the VQA score on the *test-dev* and *test-std* sets. For VQA Karpathy split, models are finetuned on the training and validation sets. We report the VQA score on the *in-domain* and *out-domain* test set. We finetune METALM for 140k steps for both the above two datasets. For OK-VQA, models are finetuned on the training set. We report the normalized VQA score on the test set. We finetune METALM with 10k steps. We apply a "Question: [*question text*] Answer: [*answer text*]" prompt for generative finetuning.

Visual Reasoning We evaluate on the NLVR^2 dataset (Suhr et al., 2019). The example in NLVR^2 consists of two images and one sentence, where the sentence describes the relations between the images. Following previous work (Tan and Bansal, 2019; Li et al., 2020), we re-split the data into two individual image-text pairs and get their representations respectively. Then we leverage the concatenation of representations to generate the *yes* or *no* predictions. We apply "it is [*label*]" for generative finetuning. We finetune METALM for 5 epochs.

Image Captioning We evaluate on the COCO caption dataset with *Karpathy split* (Karpathy and Fei-Fei, 2017). Following (Cho et al., 2021), we report BLEU-4, CIDEr, METEOR, and SPICE as the evaluation metrics. All reported results are from cross-entropy finetuning without reinforced CIDEr optimization (Rennie et al., 2017). Object tags are not used during finetuning. We apply a "caption: [*caption text*]" prompt for generative finetuning and finetune METALM for 100k steps on the training split.

Explainable Visual Reasoning We evaluate on the E-SNLI-VE dataset (Kayser et al., 2021), which requires the models to predict the entailment labels between an image-text pair and simultaneously generate explanations for the prediction. We finetune METALM for 7 epochs. This task is naturally compatible with the language generation manner. We apply a "it is [*entailment label*] because [*explanation*]." prompt for generative finetuning.

4.5.2 Results: Visual Question Answering and Visual Reasoning

Table 11 reports the finetuning results on VQAv2, VQA Karpathy, and NLVR^2. The finetuning performance is strong across the datasets. More importantly, METALM not only outperforms previous models with generative prediction, but also achieves competitive or better results compared with discriminative vision-language models. The property is favorable as the nature of some tasks is generative. For example, visual question answering needs open-ended predictions, rather than restricting the output space. The advantages of open-endedness are shown on the out-domain set of the VQA Karpathy-test. The top answers of the out-domain set are not in the most common 3,129 VQA answers. As the discriminative models can only make predictions that appear in the training set,

Model	**OK-VQA**
Discriminative Prediction	
ViLBERT (Lu et al., 2019)	35.2
KRISP (Marino et al., 2021)	38.9
MAVEx (Wu et al., 2022)	40.3
Generative Prediction	
$\text{VLKD}_{\text{ViT-B/16}}$ (Dai et al., 2022)	36.3
METALM	**46.5**

Table 12: Finetuning results on the knowledge-intensive OK-VQA dataset. Different from the VQAv2, this dataset requires not only understanding images and questions but also leveraging world knowledge. For example, for an image of a plane, the question is "*who invented this?*". All the reported results are taken from their *base* size models.

Model	**Accuracy**
(Park et al., 2018)	69.2
(Wu and Mooney, 2019)	73.7
(Marasović et al., 2020)	72.0
(Kayser et al., 2021)	79.5
(Sammani et al., 2022)	73.9
METALM	**79.9**
w/o appending explanations after labels	79.6

Table 13: Comparison of finetuning results on E-SNLI-VE (Kayser et al., 2021). Without explanation METALM still predicts the entailment label in an open-ended generative manner. The compared results are taken from (Kayser et al., 2021) and (Sammani et al., 2022).

it is difficult to generalize to out-domain examples. Among all the models, METALM achieves the best out-domain results. In comparison, although previous generative models get better results on the out-domain set, they usually underperform on other datasets. By contrast, METALM consistently achieves competitive results.

As shown in Table 12, we report the finetuning results on OK-VQA (Marino et al., 2019). Different from VQAv2, the dataset requires models to draw upon external knowledge to answer questions. Previous methods (Marino et al., 2021; Wu et al., 2022) typically leverage a knowledge base to filter candidate answers. In contrast, language models have acquired rich world knowledge during pretraining. METALM grants the flexibility of leveraging such knowledge from the causal language model. As a result, METALM obtains significant improvements on this task without relying on additional knowledge bases.

Table 13 reports the finetuning results on E-SNLI-VE entailment label prediction. METALM is trained to jointly generate the entailment label and explanation with the "it is [*entailment label*] because [*explanation*]" prompt. METALM achieves the best accuracy compared with previous methods. Moreover, an important advantage of the generative model is that METALM can leverage explanations to improve the performance of entailment label prediction. It indicates that the explanation is of help to entailment classification. The results demonstrate that METALM can be used to facilitate the interactions between users and foundation models. In other words, we can use natural language to guide model finetuning via the general-purpose interface.

The competitive results across the above datasets demonstrate that the bidirectional modeling benefits finetuning in METALM. So we can have good performance of finetuning and open-ended prediction at the same time.

Model	COCO Caption Karpathy Test			
	BLEU-4	CIDER	METEOR	SPICE
Oscar (Li et al., 2020)	34.5	115.6	29.1	21.9
Unified VLP (Zhou et al., 2020)	36.5	117.7	28.4	21.3
VL-T5 (Cho et al., 2021)	34.6	116.1	28.8	21.9
VL-BART (Cho et al., 2021)	34.2	114.1	28.4	21.3
METALM	**37.6**	**126.6**	**30.0**	**22.9**

Table 14: Finetuning results on the COCO caption Karparthy test split. All models are directly finetuned without using CIDEr optimization (Rennie et al., 2017) and object tags. The results of base-size models are taken from (Cho et al., 2021).

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	CIDER
(Park et al., 2018)	29.4	18.0	11.3	7.3	28.6	14.7	72.5
(Wu and Mooney, 2019)	30.6	19.2	12.4	8.2	29.9	15.6	83.6
(Marasović et al., 2020)	29.9	19.8	13.6	9.6	27.3	18.8	81.7
(Kayser et al., 2021)	30.1	19.9	13.7	9.6	27.8	**19.6**	85.9
(Sammani et al., 2022)	37.0	25.3	17.9	12.9	34.2	18.8	117.4
METALM	**40.6**	**26.7**	**18.7**	**13.5**	**37.6**	19.4	**119.3**

Table 15: Finetuning results of E-SNLI-VE explanation generation. METALM jointly generates entailment labels and explanations. The compared results are taken from (Sammani et al., 2022).

4.5.3 Results: Visually Grounded Language Generation

Table 14 reports the finetuning results of caption generation on COCO Karpathy test split. We directly compare with the results without CIDEr optimization (Rennie et al., 2017) for fair comparisons. The results show that METALM obtains substantial improvements over other models.

Table 15 shows the explanation generation results on E-SNLI-VE. We jointly generate entailment labels and explanations. METALM outperforms previous strong models on most metrics. Together with the label accuracy results on the same dataset in Table 13, our model achieves good performance for both understanding and explanation generation. In contrast, the method of (Sammani et al., 2022) obtains competitive performance for explanation generation, while getting inferior accuracy for entailment classification.

The results of visually grounded language generation show that our architecture is general enough to be applied to various sequence-to-sequence learning problems. METALM can achieve good performance via finetuning for vision-language generation tasks.

5 Related Work

5.1 Language Model Pretraining

Large-scale language model pretraining has achieved strong performance across various downstream tasks and aroused extensive research interest. The difference between the models mainly lies in the pretraining objective and model architecture. GPT (Radford et al., 2018; 2019; Brown et al., 2020) pretrains causal language models with decoder-only Transformers, demonstrating intriguing properties of few-shot and in-context learning. Recent efforts (Rae et al., 2021; Du et al., 2021; Smith et al., 2022; Hoffmann et al., 2022; Thoppilan et al., 2022; Chowdhery et al., 2022) focus on scaling up in terms of data and model size. In order to implement bidirectional encoding, Devlin et al. (2019) propose the masked language modeling objective. Clark et al. (2020) introduce the replaced token detection task to improve pretraining efficiency. Furthermore, some efforts investigate frameworks that can handle both natural language understanding and generation tasks. T5 (Raffel et al., 2020) introduces an encoder-decoder framework that converts all tasks into a text-to-text format. BART (Lewis et al., 2020) is a sequence-to-sequence model pretrained by reconstructing the original text from corrupted documents. UniLM (Dong et al., 2019; Bao et al., 2020) presents to jointly optimize unidirectional, bidirectional and sequence-to-sequence language modeling objectives controlled by different self-attention masks. Wang et al. (2022b), Tay et al. (2022), and Artetxe et al. (2022) study the effects of different pretraining objectives and architectures on downstream generalization. Specifically, causal language models are good at zero-shot or in-context learning, while non-causal models perform better for finetuning. In our work, we combine the best of both worlds by introducing semi-causal language modeling. So we can obtain decent finetuning performance and benefit from the capability of in-context learning. Moreover, the unification enables us to build a general-purpose interface to various foundation models.

5.2 General-Purpose Modeling

Some efforts investigate the general-purpose model that supports multiple tasks, transformations, and modalities in a shared module. MT-DNN (Liu et al., 2019a) trains on many tasks through multitask learning. Specific to language-only general-purpose, UniLM (Dong et al., 2019) and T5 (Raffel et al., 2020) unify understanding and generation ability in a single model. Moreover, language models are finetuned to follow instructions (Ouyang et al., 2022; Wei et al., 2021; Sanh et al., 2022), i.e., aligning language models with user intentions to implement the general-purpose capability. There are some work that support not only multitask but also multimodality. Jaegle et al. (2022) introduce Perceiver IO, a general architecture across multiple domains including language/visual understanding, multimodal and symbolic representations for games. Baevski et al. (2022) propose a unified learning framework for different modalities but still use modality specific encoders. Tsimpoukelli et al. (2021) demonstrate that the in-context learning ability of frozen language models can be transferred to a vision-language setting. Alayrac et al. (2022) also implement general-purpose understanding of image, video, and text by a large frozen language model. Reed et al. (2022) build a generalist agent that works as a multi-modal, multi-task, multi-embodiment generalist policy.

6 Conclusion

We present METALM, a general-purpose interface to foundation models across tasks and modalities. METALM consists of a causal decoder as the universal task layer, and multiple pretrained non-causal encoders mounted to it. We pretrain METALM with a new objective called semi-causal language modeling. Experimental results show that METALM exhibits strong finetuning and in-context learning performance across a wide range of language-only and vision-language tasks.

In the future, we would like to scale up (Wang et al., 2022a; Chi et al., 2022) the model size. Moreover, we are interested in extending METALM to multilingual settings, and handling more modalities (including language, vision, audio, and multimodality) simultaneously. Another strand of work is to extend the universal task layer to vision tasks, such as object detection, and semantic segmentation. We will also investigate parameter-efficient finetuning with METALM.

References

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *ICCV*, pages 8948–8957, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, pages 382–398, 2016.

Mikel Artetxe, Jingfei Du, Naman Goyal, Luke Zettlemoyer, and Ves Stoyanov. On the role of bidirectionality in language model pre-training. *ArXiv*, abs/2205.11726, 2022.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. *ArXiv*, abs/2202.03555, 2022.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. UniLMv2: Pseudo-masked language models for unified language model pre-training. In *ICML 2020*, volume 119, pages 642–652, 2020.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. The second PASCAL recognising textual entailment challenge. In *Proceedings of the PASCAL Workshop on Textual Entailment and Paraphrasing*, 01 2006.

Yoshua Bengio. From system 1 deep learning to system 2 deep learning. In *NeurIPS 2019 – Posner Lecture*, 2019.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. The fifth pascal recognizing textual entailment challenge. In *In Proc Text Analysis Conference*, 2009.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *AAAI*, pages 7432–7439, 2020.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, pages 632–642, 2015.

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*, 2020.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, pages 104–120, 2020.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, and Furu Wei. On the representation collapse of sparse mixture of experts. *ArXiv*, abs/2204.09179, 2022.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *ICML*, pages 1931–1942, 2021.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and Noah Fiedel. PaLM: Scaling language modeling with Pathways. *ArXiv*, abs/2204.02311, 2022.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *NAACL-HLT*, pages 2924–2936, 2019.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *ACL 2020*, pages 8440–8451, 2020.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR*, pages 702–703, 2020.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *MLCW*, pages 177–190, 2006. ISBN 3-540-33427-0, 978-3-540-33427-9.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. Enabling multimodal generation on CLIP via vision-language knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2383–2395, May 2022.

Marie-Catherine De Marneff, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. *proceedings of Sinn und Bedeutung 23*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019*, pages 4171–4186, 2019.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, 2005.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *NeurIPS 2019*, pages 13042–13054, 2019.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathy Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts. *CoRR*, abs/2112.06905, 2021.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL-HLT*, pages 2368–2378, 2019.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. 59:123–156, January 2020. doi: 10.1016/j.csl.2019.06.009.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, 2021.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, June 2007.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, November 2019.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *Stanford Project Report*, pages 1–6, 2009.

Aaron Gokaslan and Vanya Cohen. OpenWebText corpus, 2019.

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.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. Training compute-optimal large language models. *ArXiv*, abs/2203.15556, 2022.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001.

Wenlong Huang, P. Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *ArXiv*, abs/2201.07207, 2022.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. 2022.

Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. In *ACL*, pages 2763–2775, May 2022.

Daniel Kahneman. *Thinking, fast and slow*. 2011. ISBN 9780374275631 0374275637.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):664–676, 2017.

Maxime Kayser, Oana-Maria Camburu, Leonard Salewski, Cornelius Emde, Virginie Do, Zeynep Akata, and Thomas Lukasiewicz. e-ViL: A dataset and benchmark for natural language explanations in vision-language tasks. In *ICCV*, pages 1244–1254, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *ECML*, volume 3201, pages 217–226, 2004.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers, MTSummit 2005, Phuket, Thailand, September 13-15, 2005*, pages 79–86, 2005.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*, pages 66–71, 2018.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *TACL*, 7:453–466, 2019.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *ACL*, July 2019.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Principles of Knowledge Representation and Reasoning*, 2012.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880, 2020.

Xin Li and Dan Roth. Learning question classifiers. In *COLING*, 2002.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, pages 121–137, 2020.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, November 2020.

Tsung-Yi Lin, Michael Maire, Serge 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.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *ACL*, pages 4487–4496, 2019a.

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. *CoRR*, abs/1907.11692, 2019b.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, pages 13–23, 2019.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150, June 2011.

Ana Marasović, Chandra Bhagavatula, Jae sung Park, Ronan Le Bras, Noah A. Smith, and Yejin Choi. Natural language rationales with full-stack visual reasoning: From pixels to semantic frames to commonsense graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2810–2829, November 2020.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3195–3204, 2019.

Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based VQA. In *CVPR*, pages 14111–14121, 2021.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *EMNLP*, pages 2381–2391, 2018.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51, 2017.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745, 2018.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *ACL*, pages 4885–4901, 2020.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *NeurIPS*, 24, 2011.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *CVPR*, pages 8779–8788, 2018.

Mohammad Taher Pilehvar and José Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *NAACL-HLT*, pages 1267–1273, 2019.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *ICLR*, 2020.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/2112.11446, 2021.

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):1–67, 2020.

Altaf Rahman and Vincent Ng. Resolving complex cases of definite pronouns: The winograd schema challenge. In *EMNLP-CoNLL*, pages 777–789, 2012.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *ACL*, pages 784–789, 2018.

Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *ArXiv*, abs/2205.06175, 2022.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, pages 7008–7024, 2017.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium*, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *AAAI*, pages 8732–8740, 2020.

Fawaz Sammani, Tanmoy Mukherjee, and Nikos Deligiannis. NLX-GPT: A model for natural language explanations in vision and vision-language tasks. In *CVPR*, 2022.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *ICLR*, 2022.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *ICLR*, 2019.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pages 2556–2565, 2018.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model. *CoRR*, abs/2201.11990, 2022.

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.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NeurIPS*, pages 3104–3112, 2014.

Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP-IJCNLP*, pages 5100–5111, 2019.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. Unifying language learning paradigms. *ArXiv*, abs/2205.05131, 2022.

Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zero-shot image-to-text generation for visual-semantic arithmetic. *arXiv preprint arXiv:2111.14447*, 2021.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. Lamda: Language models for dialog applications. *CoRR*, abs/2201.08239, 2022.

Jörg Tiedemann. Finding alternative translations in a large corpus of movie subtitle. In *LREC*, 2016.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *NeurIPS 2021*, 34, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS 2017*, pages 5998–6008, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015.

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 *BlackboxNLP*, pages 353–355, 2018.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2019.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling Transformers to 1,000 layers. *ArXiv*, abs/2203.00555, 2022a.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective work best for zero-shot generalization? *ArXiv*, abs/2204.05832, 2022b.

Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *ArXiv*, abs/2111.02358, 2021.

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

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *CoRR*, abs/2109.01652, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.

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

Jialin Wu and Raymond Mooney. Faithful multimodal explanation for visual question answering. In *BlackboxNLP*, pages 103–112, August 2019.

Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. Multi-modal answer validation for knowledge-based VQA. In *AAAI*, 2022.

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.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *ACL*, pages 4791–4800, 2019.

Rui Zhang and Joel Tetreault. This email could save your life: Introducing the task of email subject line generation, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, page 649–657, 2015.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: paraphrase adversaries from word scrambling. In *NAACL-HLT*, pages 1298–1308, 2019.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. In *AAAI*, volume 34, pages 13041–13049, 2020.

A Hyperparameters of Language-Only Experiments

A.1 Pretraining

We provide the detailed pretraining hyperparameter settings of language-only METALM. Model hyperparameters are shown in Table 16 and optimization hyperparamters are shown in Table 17.

Hyperparameters	Non-causal	Semi-causal
Number of layers	24	24
Hidden size	1024	2048
FFN inner hidden size	4096	8192
Attention heads	16	32
Attention head size	64	64
Dropout	0.1	0.0
Attention Dropout	0.1	0.0
Initialization	DeepNorm	DeepNorm
Max length	512	2048
Position Embedding	Learnable	Sinusoidal

Table 16: Hyperparameters of non-causal and semi-causal models for language-only pretraining.

Hyperparameters	Value
Training steps	300,000
Warm up steps	375
Batch size	512
Optimizer	Adam
Learning rate	6e-4
Learning Rate Decay	Linear
Adam ϵ	1e-6
Adam β	(0.9, 0.98)
Weight decay	0.01
Non-causal percent	0.25

Table 17: Optimization hyperparameters for language-only pretraining.

A.2 Multitask Finetuning and Instruction Tuning

We provide the detailed settings of language-only multitask finetuning and instruction tuning with METALM in Table 18.

Hyperparameters	Multitask Finetuning	Instruction Tuning
Training steps	20,000	30,000
Warm up steps	2,000	3,000
Batch size	256	512
Optimizer	Adam	Adam
Learning rate	1e-4	1e-4
Adam ϵ	1e-6	1e-6
Adam β	(0.9, 0.98)	(0.9, 0.98)
Weight decay	0.01	0.01
Max length	2048	1024
Dropout of non-causal model	0.1	0.1
Dropout of causal model	0.0	0.0

Table 18: Hyperparameters used for language-only multitask finetuning and instruction tuning.

B Datasets Used for Language-Only Experiments

B.1 Pretraining

Language-only METALM is pretrained on Pile (Gao et al., 2021), which is an 800 GB English text corpus combining 22 diverse sources. We exclude data sources of GitHub, arXiv and PubMed Central from the original Pile. Thus the pretraining corpus we used is composed of 19 sources, divided into the following five categories:

- **Academic**: FreeLaw, USPTO Backgrounds, PhilPapers, NIH Exporter, PubMed Abstracts
- **Internet**: Pile-CC, OpenWebText2, StackExchange, Wikipedia (English)
- **Prose**: BookCorpus2, Books3, Gutenberg (Rae et al., 2020, PG-19)
- **Dialogue**: OpenSubtitles (Tiedemann, 2016), Youtube Subtitles, EuroParl (Koehn, 2005), Hacker News, Ubuntu IRC
- **Miscellaneous**: Enron Emails (Klimt and Yang, 2004), DM Mathematics (Saxton et al., 2019)

B.2 Multitask Finetuning and Instruction Tuning

We list the datasets we used for language-only multitask finetuning and instruction tuning.

- **Natural Language Inference** is to determine whether a hypothesis is true (entailment), false (contradiction) or undetermined (neutral) given a premise. We use the following datasets: ANLI (Nie et al., 2020), CB (De Marneff et al., 2019), MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), SNLI (Bowman et al., 2015) and WNLI (Levesque et al., 2012).
- **Sentiment Classification** is to determine the emotional tone of a piece of text, whether it is positive or negative: IMDB (Maas et al., 2011), SST-2 (Socher et al., 2013), Sentiment140 (Go et al., 2009), Yelp (Zhang et al., 2015).
- **Paraphrase Detection** is to detect the semantic similarity of two sentences: QQP (Wang et al., 2018), MRPC (Dolan and Brockett, 2005), Paws Wiki (Zhang et al., 2019).
- **Coreference Resolution** is to determine if two expressions refer to the same entity in a text: DPR (Rahman and Ng, 2012), Winogrande (Sakaguchi et al., 2020), WSC (Levesque et al., 2012).
- **Commonsense Reasoning** evaluates the ability to perform physical or social commonsense: HellaSwag (Zellers et al., 2019), PiQA (Bisk et al., 2020), COPA (Roemmele et al., 2011).
- **Reading Comprehension** is to answer some questions conditioned on a given passage: DROP (Dua et al., 2019), SQuADv1 (Rajpurkar et al., 2016), SQuADv2 (Rajpurkar et al., 2018), OBQA (Mihaylov et al., 2018), BoolQ (Clark et al., 2019).
- **Miscellaneous** consists of some additional datasets: CoLA (Warstadt et al., 2019), WiC (Pilehvar and Camacho-Collados, 2019), TREC (Li and Roth, 2002; Hovy et al., 2001).
- **Closed-Book QA** is to answer a question without external knowledge: ARC-easy (Clark et al., 2018), NQ (Kwiatkowski et al., 2019; Lee et al., 2019).
- **Struct to Text** is to construct a natural language description for some structured data: CommonGen (Lin et al., 2020), E2ENLG (Dušek et al., 2020).
- **Summarization** is to generate a summary of a given passage: AESLC (Zhang and Tetreault, 2019), SamSum (Gliwa et al., 2019), XSum (Narayan et al., 2018).

Furthermore, we utilize the hand-crafted templates from FLAN (Wei et al., 2021), which composes ten templates for each dataset. For multitask finetuning, we apply only the first template for each dataset. For instruction tuning, we apply all the ten templates.

B.3 In-Context Learning

We conduct experiments of in-context learning on four categories:

- Cloze and completion tasks: StoryCloze (Mostafazadeh et al., 2017), HellaSwag (Zellers et al., 2019)
- Winograd-style tasks: Winograd (Levesque et al., 2012), Winogrande (Sakaguchi et al., 2020)
- Commonsense reasoning: ARC-easy/ARC-challenge (Clark et al., 2018), PIQA (Bisk et al., 2020)
- Two datasets from SuperGLUE benchmark (Wang et al., 2019): BoolQ (Clark et al., 2019), Copa (Roemmele et al., 2011)

C Detailed Results of Multitask Finetuning in Section 3.3

We list the full results of language-only multitask finetuning for all task clusters in our experiments. Results of natural language inference are shown in Table 19. Results of sentiment classification are shown in Table 20. Results of paraphrase detection are shown in Table 21. Results of reading comprehension are shown in Table 22. Results of coreference resolution are shown in Table 23. Results of miscellaneous cluster are shown in Table 24. Results of commonsense reasoning are shown in Table 25. Results of struct to text are shown in Table 26. Results of closed-book QA are shown in Table 27. Results of text summarization are shown in Table 28.

	ANLI R1	ANLI R2	ANLI R3	CB	MNLI-m	QNLI	RTE	SNLI	WNLI
GPT	52.9	45.5	43.5	89.3	81.1	90.1	79.4	85.3	18.3
METALM	**72.6**	**54.1**	**49.5**	**91.1**	**88.9**	**93.8**	**87.7**	**90.0**	**84.5**

Table 19: Multitask finetuning results of natural language inference.

	IMDB	SST-2	Sent140	Yelp
GPT	95.7	93.5	85.2	97.2
METALM	**96.5**	**94.7**	**89.1**	**97.9**

Table 20: Multitask finetuning results of sentiment classification.

	QQP	MRPC	PAWS Wiki
GPT	84.4	78.9	88.4
METALM	**88.0**	**86.8**	**94.1**

Table 21: Multitask finetuning results of paraphrase detection.

	DROP	SQuADv1	SQuADv2	OBQA	BoolQ
Metric	f1	f1	f1	acc	acc
GPT	38.5	84.0	72.0	49.6	78.4
METALM	**45.2**	**89.3**	**84.1**	**61.8**	**85.0**

Table 22: Multitask finetuning results of reading comprehension.

	DPR	Winogrande
GPT	71.6	62.5
METALM	**87.8**	**80.8**

Table 23: Multitask finetuning results of coreference resolution.

	CoLA	WIC	TREC
GPT	79.7	63.9	97.2
METALM	**87.3**	**67.6**	**98.0**

Table 24: Multitask finetuning results of miscellaneous cluster.

	HellaSwag	PiQA	CoPA
GPT	56.5	67.3	66.0
METALM	**83.2**	**76.4**	**93.0**

Table 25: Multitask finetuning results of commonsense reasoning.

	NQ	ARC-e
GPT	15.3	61.1
METALM	**16.5**	**72.1**

Table 26: Multitask finetuning results of closed-book question answering.

	CommonGen			E2ENLG		
	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
GPT	**47.3**	18.7	**41.0**	64.2	**37.0**	**47.5**
METALM	46.8	**18.8**	40.7	**64.4**	36.4	**47.5**

Table 27: Multitask finetuning results of struct to text.

	AESLC			SamSum			XSum		
	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
GPT	29.1	**15.3**	28.7	45.3	20.2	36.7	29.8	10.5	24.0
METALM	**30.3**	15.0	**29.5**	**46.5**	**21.7**	**38.2**	**31.3**	**11.6**	**25.3**

Table 28: Multitask finetuning results of text summarization.

D Hyperparameters of Vision-Language Experiments

D.1 Hyperparameters of Vision-Language Pretraining

We report the detailed pretraining hyperparameter settings of the vision-language METALM in Table 29 and report the optimization hyperparameters in Table 30.

Hyperparameters	Non-causal	Semi-causal
Number of layers	12	24
Hidden size	768	1,024
FFN inner hidden size	3,072	4,096
Attention heads	12	16
Dropout	0.1	0.2
Attention Dropout	0.1	0.2
Vocabulary Size	50,259	50,259
Pretraining length	236	1024
Freeze in pretraining	No	No
Position Embedding	Learnable	Sinusoidal

Table 29: Hyperparameters of non-causal and semi-causal models for vision-language pretraining.

Hyperparameters	Value
Training steps	350,000
Warm up steps	2,500
Batch size	256
Optimizer	AdamW
Learning rate	1e-4
Learning Rate Decay	Linear
Adam ϵ	1e-8
Adam β	(0.9, 0.98)
Weight decay	0.01
Image size	224x224
Patch size	16

Table 30: Optimization hyperparameters for vision-language pretraining.

D.2 Hyperparameters in Vision-Language Finetuning

We report the finetuning settings along with the prompts in Table 31. The vision-language METALM applies a 384x384 image size and greedy decoding for all finetuning tasks.

Task	Learning rate	Batch size	Steps	Language prompt
VQAv2	1e-5	64	140k	question: [question] answer: [answer]
VQA Karpathy	1e-5	64	140k	question: [question] answer: [answer]
OK-VQA	1e-5	8	10k	question: [question] answer: [answer]
COCO Caption	1e-5	64	100k	caption: [caption]
NLVR^2	1e-5	32	54k	it is [label]
E-SNLI-VE label	1e-5	64	54k	it is [label] because [explanation]
E-SNLI-VE explanation	1e-5	64	54k	it is [label] because [explanation]

Table 31: Summary of hyperparameters for vision-language finetuning.