# Masked AutoDecoder is Effective Multi-Task Vision Generalist

Han Qiu[1], Jiaxing Huang[1], Peng Gao[2], Lewei Lu[3], Xiaoqin Zhang[4], Shijian Lu[1]*

[1]S-Lab, Nanyang Technological University
[2]Shanghai Artificial Intelligence Laboratory, [3]Sensetime Research
[4]College of Computer Science and Technology, Zhejiang University of Technology
han023@e.ntu.edu.sg, {Jiaxing.Huang, Shijian.Lu}@ntu.edu.sg, gaopeng@pjlab.org.cn

## Abstract

*Inspired by the success of general-purpose models in NLP, recent studies attempt to unify different vision tasks in the same sequence format and employ autoregressive Transformers for sequence prediction. They apply uni-directional attention to capture sequential dependencies and generate task sequences recursively. However, such autoregressive Transformers may not fit vision tasks well, as vision task sequences usually lack the sequential dependencies typically observed in natural languages. In this work, we design Masked AutoDecoder (MAD), an effective multi-task vision generalist. MAD consists of two core designs. First, we develop a parallel decoding framework that introduces bi-directional attention to capture contextual dependencies comprehensively and decode vision task sequences in parallel. Second, we design a masked sequence modeling approach that learns rich task contexts by masking and reconstructing task sequences. In this way, MAD handles all the tasks by a single network branch and a simple cross-entropy loss with minimal task-specific designs. Extensive experiments demonstrate the great potential of MAD as a new paradigm for unifying various vision tasks. MAD achieves superior performance and inference efficiency compared to autoregressive counterparts while obtaining competitive accuracy with task-specific models. Code will be released at https://github.com/hanqiu-hq/MAD.*

## 1. Introduction

Computer vision covers various concepts, such as localization, classification, and description, leading to a wide variety of highly structured outputs in different vision tasks, i.e., object detection, instance segmentation, keypoint detection, image captioning, etc. Following natural language processing (NLP), recent methods [7, 22, 32, 44, 45] attempt to

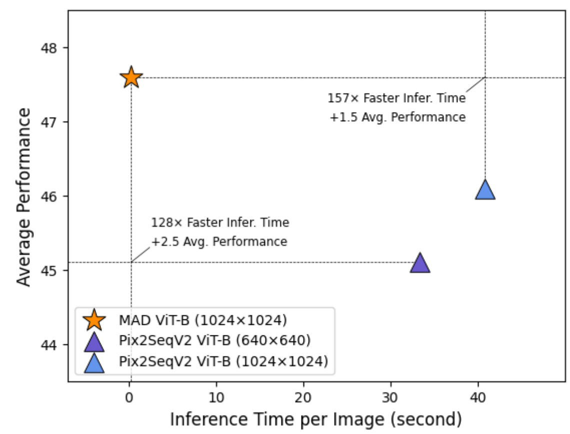



Figure 1. The proposed MAD outperforms the state-of-the-art Pix2SeqV2 [7] significantly in inference time, meanwhile achieves competitive accuracy across four representative vision tasks. The *Average Performance* is averaged over four tasks including object detection (mAP), instance segmentation (mAP), keypoint detection (mAP), and image captioning (B@4). MAD achieves approximately 100× acceleration in inference time.

unify different vision tasks in an autoregressive sequence-to-sequence framework as illustrated in the upper part of Fig. 2. They first model different vision tasks in the same sequence format, such as a sequence of coordinate and class label tokens for object detection, a sequence of contour coordinate tokens for image segmentation, or a sequence of descriptive sentences for image captioning. Additionally, the autoregressive Transformers [4, 34, 35], with its specially designed uni-directional attention to capture sequential dependencies, are employed to recursively predict these vision task sequences.

Despite the success, the autoregressive approach often struggles on vision tasks due to two major factors: (1) *The discrepancy between vision and language*. Language task sequences [4, 43] heavily follow sequential dependencies while vision task sequences may not, e.g., the next

*Corresponding author.

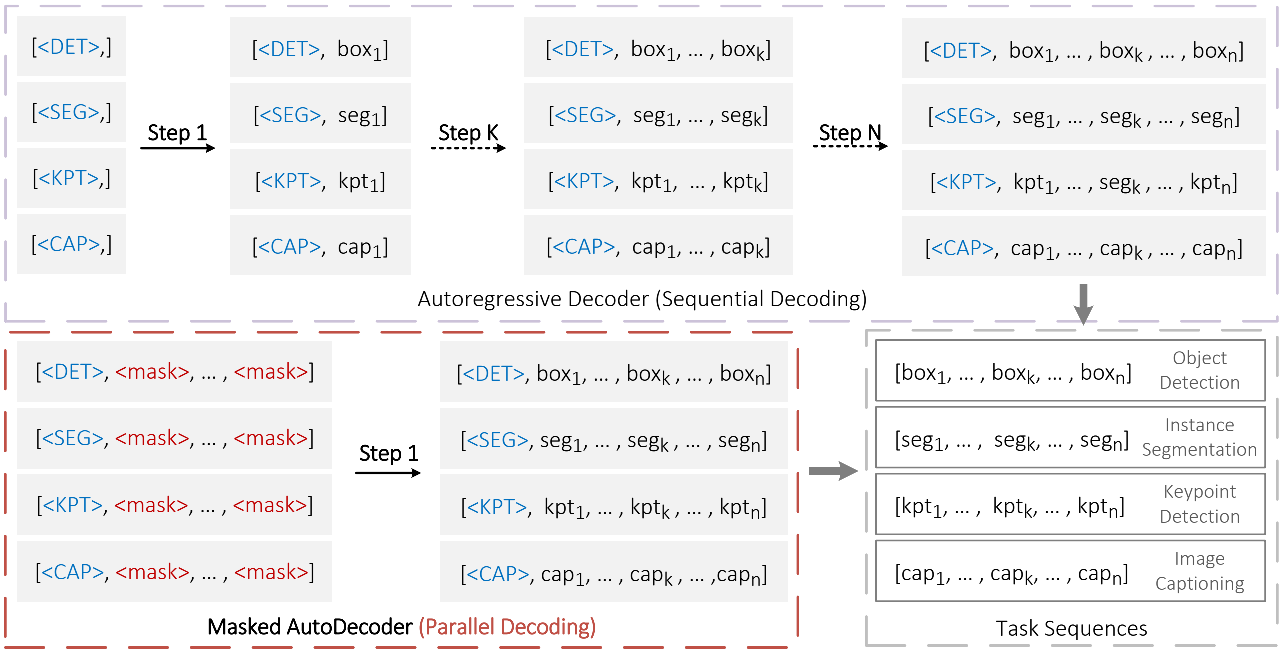



Figure 2. Unified Sequence-to-sequence Modeling of Vision Tasks. The traditional *Autoregressive Decoder* adopts sequential decoding for prediction, and utilizes unidirectional attention where each token can only attend to its previous ones. It generates task sequences token by token, resulting in a slow generation process with $N$ steps up to the length of sequences. The proposed *Masked AutoDecoder* (MAD), equipped with parallel decoding with bidirectional attention and masked sequence modeling, allows decoding task sequences with only one step. Additionally, via masking and reconstructing task sequences, MAD can capture rich task contexts for different tasks, resulting in an effective and efficient vision generalist. Tokens in blue denote task prompts. $< mask >$ denotes mask token. *Task Sequences* are simplified with details to be described in the ensuing Sections.

word prediction in a sentence highly depends on its preceding texts, while the pixel prediction in segmentation tasks largely depends on its neighboring content instead of merely previous ones. The autoregressive approach, with uni-directional attention, can well capture sequential dependencies for language tasks but may not fit well with vision tasks. (2) *Computation Efficiency*. The autoregressive approach predicts tokens in a sequence recursively, which is computation-intensive. The two factors might limit the model performance and efficiency, hindering the application of the autoregressive approach to vision tasks.

One possible solution for mitigating the above two issues is to explore bi-directional attention and parallel prediction for sequence modeling. This design leads to a customized Transformer that is capable of capturing more comprehensive dependencies and decodes the task sequence from scratch in parallel. However, such decoding process from scratch may struggle while modeling task contexts as sequences from different tasks highly vary in patterns, lengths, token vocabularies, etc., which will impede network convergence and result in inferior performance for multi-task learning.

Driven by the above analysis, we present Masked Decoder (MAD), an effective sequence-based generalist for vision tasks. As illustrated in the bottom-left part of Fig. 2, MAD masks tokens randomly from the task sequences and reconstructs the masked tokens based on the unmasked ones and image features, which provides rich task contexts for modeling disparate task sequences. In addition, it adopts an encoder-decoder Transformer architecture with bi-directional attention that leverages comprehensive dependencies in vision tasks to effectively decode task sequences in parallel. These designs enable a more efficient and effective multi-task learning framework that performs multiple vision tasks in a single architecture. Our experiments with four tasks (object detection, instance segmentation. key-point detection, and image captioning) on the COCO dataset demonstrate that a simple MAD can achieve competitive accuracy and efficiency compared to both task-customized approaches and existing generalist models.

## 2. Related Works

**Vision Generalist Models** Learning a vision generalist model capable of handling multiple vision tasks using a shared architecture has long been a goal in computer vision. Inspired by the success of unified sequence-to-sequence based transformer framework [11, 34, 35] in natural language processing (NLP), recent works [2, 8, 38, 44] extend this framework to the field of computer vision and model various vision tasks in a unified sequence-to-sequence autoregressive paradigm. Pioneering works [9, 27, 50] mainly focus on high-level semantic tasks, such as image captioning, visual question answering, image-text matching, etc., considering their intrinsic correlation with language. In pursuit of unifying more vision tasks, especially for those involving dense predictions, Pix2seq [6, 7] and UniTab [49] discrete object positions as a series of coordinate tokens to enable localization capability for generalist models. Unified-IO [32] and UViM [22] encode the per-pixel targets into semantic tokens for vision tasks that require outputs as images, such as depth estimation or panoptic segmentation. Uni-PercieverV2 [26] equips an additional region proposal network to generate sequence predictions for object detection and instance segmentation. VisionLLM [45] leverages LLM to enable flexible task output formats. Different from these methods which focus on customizing and extending more vision tasks in a sequence-based autoregressive framework, we demonstrate that such a framework may not fit well for vision tasks. Our masked Auto-decoding pursues a conceptually different direction and learns diverse task contexts in parallel via masked sequence modeling, leading to a more efficient and effective vision generalist.

**Masked Signal Modeling** The paradigm of learning rich representations via masking and reconstructing has been widely explored in both fields of NLP and computer vision. In NLP, through masking and recovering language sentences, models like BERT [11] and its variants [25, 30] successfully pre-train models capable of generalizing to a broad range of NLP tasks. In computer vision, such a paradigm also leads to multiple masked image modeling (MIM) [12, 16] and masked video modeling (MVM) techniques. For example, BEIT [3] explores MIM by recovering the masked image into visual tokens from discrete VAE [37]. SimMIM [48], MaskFeat [46], and MAE [21] incorporate low-level visual signals, such as RGB pixel value or the feature descriptor HOG [10], as the reconstruction targets. VideoMAE [15] encodes the corrupted video and learns to recover both spatial and temporal signals. The above methods employ masked signal modeling as a self-supervised task, aiming to learn to auto-encode rich representations for downstream tasks. Different from them, we propose masked auto-decoder (MAD), exploring masked sequence modeling for decoding task sequences from its masked variants. Our approach is close to non-autoregressive translation [17, 18] in NLP, but it has very different intrinsic objectives - non-autoregressive translation exploits parallel decoding to improve translation efficiency, while MAD aims to model diverse task contexts for learning multi-task vision generalists.

# 3. Methods

Our proposed unified generalist framework consists of three key components: (1) Unified tokenization of diverse input and output sequences for vision tasks; (2) Masked autodecoding framework for modeling task contexts; (3) An architecture that decodes desired task sequences based on image features. We introduce these components in the following sections.

## 3.1. Task Tokenization

In this work, we consider four vision-related tasks, including object detection, instance segmentation, keypoint detection, and image captioning. These tasks require model abilities from classification to localization, from vision to language, and from image-level to pixel-level recognition. A comprehensive vocabulary is essential to deal with such complex problems. We build a universal vocabulary for all involved tasks and construct task sequences as follows:

For object detection, we convert bounding boxes into a sequence of tokens consisting of discrete coordinates and categories by the order of $[x_{min}, y_{min}, x_{max}, y_{max}, class]$. For implementation, we first construct a sequence of $N$ noise objects, then randomly replace and inject ground truth objects in the sequence. The $< Detection >$ prompt tokens are added before the sequence to identify the task. We set $N$ at 100 by default.

For instance segmentation, we directly predict bit masks following Mask R-CNN [20]. The bit masks of the size $M \times M$ are flattened and transformed into sequences consisting of $< Foreground >$ tokens and $< Background >$ tokens. We concatenate the prompt sequence consisting of task token $< Segmentation >$, bounding box coordinate tokens, and a class token to identify different instances.

For keypoint detection, we predict the coordinates and visibility for each keypoint of the person instance. It can thus be represented as a sequence of $[x, y, visibility, x, y, visibility, ...]$. We adopt two tokens $< Visible >$ and $< Invisible >$ to depict the visibility. The keypoints are arranged by the default order as in COCO dataset [29]. For the occluded keypoints, we replace their coordinate tokens with random coordinates within the bounding box. We utilize the sequence $[< Keypoint >, x_{min}, y_{min}, x_{max}, y_{max}, person]$ to prompt keypoint detection task, where the coordinates in the prompt indicate the bounding box of the corresponding person.

For captioning, we adopt a pre-trained sentence-piece model (SPM) [23] to convert a caption into a sequence of discrete tokens. We randomly replace one of the tokens in the transferred sequence with a random word token for sequence augmentation. All the sequences are padded or truncated to a length of 20 tokens. The $< Caption >$ token is adopted as the prompt.

The final vocabulary thus comprises five parts, including prompt tokens for distinguishing tasks, coordinate tokens for localization, category tokens for classification, task-related special tokens, and word tokens for captioning. Compared to previous studies [7, 32, 45], we modify the format of task sequences accordingly in MAD for embracing parallel decoding.

### 3.2. Masked AutoDecoding

**Masked Training** We propose Masked AutoDecoding for multi-task sequence modeling. We randomly sample a subset of target tokens and mask the remaining ones. The sampling follows a uniform distribution. The masked tokens are replaced by special $< Mask >$ tokens, which are shared among all tasks. During training, we adopt two kinds of sequences for each task, a fully masked sequence and a partly masked sequence as shown in Fig. 3.

The reconstruction of fully masked sequences establishes a basis to train a unified decoder, which is, learning to decode multi-task sequences with only the prompts. Therefore, all the tokens, except those in the prompt sequences, are replaced with $< Mask >$ before being fed into the decoder. The training objective is to reconstruct the desired task sequences based on task prompts. However, different from the autoregressive approach where each task sequence is specified by its corresponding input sequence, a fully masked sequence in auto-decoding might match multiple similar task sequences, such as differently arranged objects for object detection or similar captioning sentences per image for image captioning. Randomly choosing the reconstruction target each time might hinder convergence. Hence, we adopt *Hungarian Matching* [24] to construct the task sequences for object detection and image captioning. For instance segmentation and keypoint detection, the original unmasked sequences are adopted as targets since their prompt sequences with object locations and categories are able to specify the unambiguous task sequences.

However, when all the tokens are masked, it is difficult for the model to distinguish different task sequences based on only a few prompt tokens. We thus leverage partly masked sequences to alleviate this issue. The unmasked tokens provide rich cues for the pattern of different task sequences, which help the decoder capture diverse task contexts. During training, both fully and partly masked sequences are concatenated together and decoded in parallel.

The MAD task is greatly inspired by the self-supervised masked auto-encoding approach in both language and vision domains, which learns and encodes informative representation by reconstructing masked content. We expand this idea of masked modeling to decode multi-task sequences in computer vision. This simple method, by modeling corrected sequences and predicting missing tokens, enables MAD to learn distinct task contexts and inter-sequence dependencies for vision tasks.

**Masked Inference** During inference, we conduct multi-stage masked decoding to refine the prediction. With the initial prediction recovered from the fully masked sequence, we randomly sample tokens from the predicted sequences and replace them with mask tokens. The corrupted sequences are then fed to the decoder for reconstruction. We directly ensemble predictions from masked tokens to their original tokens to obtain more accurate results.

### 3.3. Architecture

Our goal is to build a single model that is capable of handling different vision-related tasks within a unified sequence paradigm with little task-customized designs. Hence, we adopt a simple encoder-decoder transformer architecture, which has been proven successful in handling sequences with variable lengths in both natural language processing and computer vision tasks. As shown in Fig. 3, the overall architecture of MAD consists of two parts, a backbone network with the encoder to extract image features and a decoder to reconstruct the masked sequences.

**Backbone and Transformer Encoder.** Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, a backbone network is adopted to generate a low-resolution image feature with a stride of 32. The encoder then takes the image feature, adds 2D positional encodings, and processes the feature via a series of encoder layers consisting of a self-attention module and feed-forward network (FFN). The image feature is then injected into the decoder as a condition to decode task sequences.

**Transformer Decoder.** The decoder follows standard architecture, reconstructing *Masked Sequences* through self-attention, cross-attention, and FFN layers. To address the sequence order, we introduce learned sequence positional encodings and add them to the input sequences before each attention layer in the decoder. The sequence positional encodings are shared among all tasks and are truncated according to the length of different task sequences. Unlike existing autoregressive methods [7, 32] that adopt uni-directional masks in self-attention layers, our model decodes all sequences in parallel with bi-directional attention, leading to more efficient and effective predictions and multi-task training.

### 3.4. Multi-task Training

**Loss Function.** We adopt a softmax cross-entropy loss to maximize the likelihood of masked sequence conditioned on the image feature:

$$L = \sum_t W_t \frac{1}{N_m} \sum_{i\in M} logP(\hat{y}_i|x,y) \tag{1}$$

where $y$ and $\hat{y}$ are masked and decoded sequences, $W_t$ are loss weights for different tasks, $M$ means the set of masked tokens, $N_m$ denotes the number of masked tokens. Only

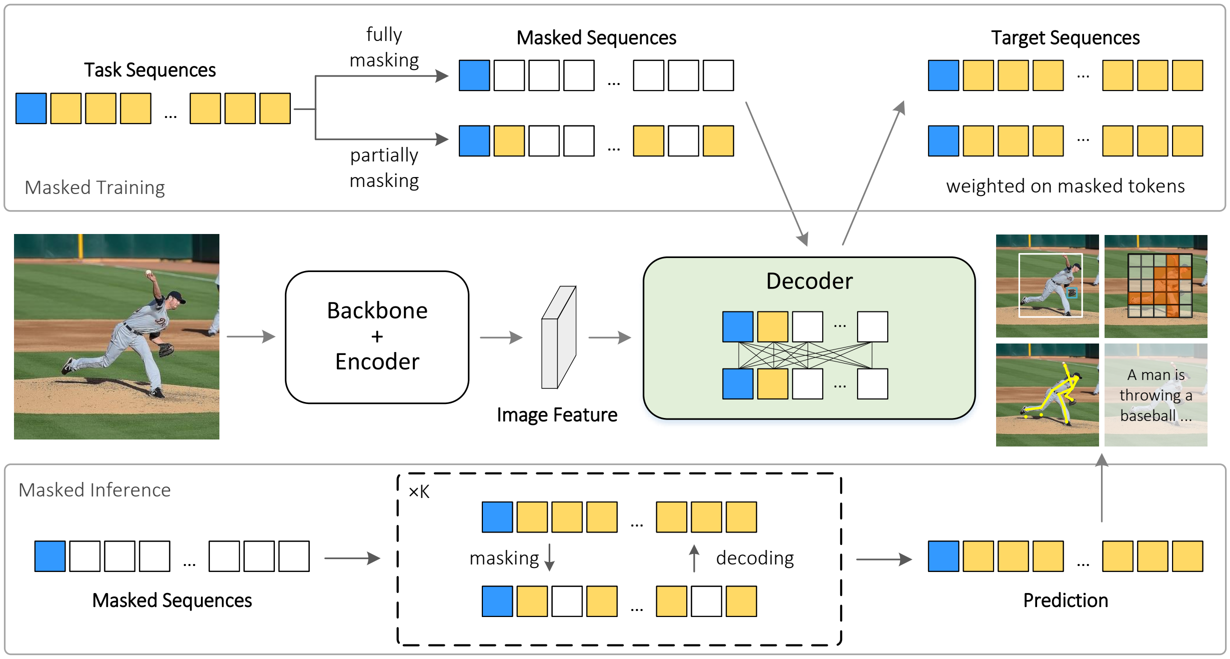



Figure 3. Illustration of our proposed MAD with masked training and masked inference. MAD consists of two major parts, a *Backbone + Encoder* to extract the representation of the input images and a *Decoder* that processes *Masked Sequences* for prediction. During training, MAD randomly masks *Task Sequences* (blue for prompt tokens and yellow for task tokens) to generate *Masked Sequences* (white for masked tokens) and learns to reconstruct the *Task Sequences*. During inference, MAD takes fully *Masked Sequences* as input and repeats the decoding and masking process to refine predictions. When $K$ in masked inference is set to 0, MAD can skip refinement and directly generate prediction in one step.

losses of masked tokens are counted. Following previous works [1, 5], we adopt auxiliary losses for predictions after each decoder layer. For each task, we filter the target token vocabulary so that losses are only calculated on the involved vocabulary to improve training efficiency. Considering tokens of whole vocabulary leads to intensive computation and memory usage since image captioning introduces plenteous text tokens that are not involved in other tasks.

**Task Mixed Sampling.** For learning a single model for multiple tasks, we employ a task-mixed sampling strategy where each image in the dataset is sampled with its annotations mixed from all tasks. The sampled images are processed by the backbone and encoder only once for encoding image features shared by all tasks. Only the decoding process is repeated for different tasks, considering that they hold different sequence lengths and are hard to process in parallel. Such a strategy is conceptually simple and effective compared with the batch mixing strategy from existing work [7, 26] where each batch only samples image-sequence pairs for a single task. Considering that each image might involve multiple vision tasks, batch mixing requires encoding the same image multiple times for different tasks. As a comparison, task mixing provides a more flexible framework to add more data from more tasks, while also sharing most model components among tasks, resulting in better efficiency.

### 3.5. Comparison with Autoregressive Decoding

The central idea of MAD is conceptually simple: parallel decoding with bidirectional attention for effective and efficient task-sequence decoding, and masked autodecoding for task-context modeling. Although the second design shares a similar concept as previous autoregressive counterparts by removing part of the data and learning to recover them, our approach is intrinsically different. Take an autoregressive vision generalist Pix2SeqV2 [7] as an example. The differences can be summarized in four major aspects as listed.

**Sequence Data Corruption Strategies.** Pix2SeqV2 implicitly corrupts sequence data, by applying uni-directional masks on attention layers to restrict tokens from perceiving their following ones. On the contrary, MAD achieves corruption by directly masking tokens of input sequences.

**Content Recovering Strategies.** Pix2SeqV2 adopts next-token prediction, where each token of sequences is trained to predict its subsequent token. On the contrary, MAD predicts masked tokens only with original tokens as targets.

**Self-Attention Mechanisms.** The triangular causal mask is applied to the self-attention layer of the decoder in Pix2SeqV2, resulting in uni-directional attention. On the contrary, MAD adopts bi-directional self-attention for the decoder, which facilitate in capturing comprehensive dependencies for vision task sequences.

**Inference Mechanisms.** Pix2SeqV2 leverages sequential decoding that generates sequences token by token, resulting in slow inference time that increases with sequence length. On the contrary, MAD adopts parallel decoding that could generate task sequences with only one step. In addition, MAD can further refine predictions by repeating the making and decoding process, leading to a more efficient and flexible inference mechanism.

# 4. Experiments

## 4.1. Experimental Settings

**Dataset and Tasks.** Following previous practice [7, 45], we evaluate MAD on MS-COCO dataset [29] which contains 118k training images and 5k validation images with annotations for all four tasks we considered. For object detection, we take $N = 100$ instances per image for training, resulting in a sequence of length 500. The coordinates of bounding boxes are discretized into 500 bins. For instance segmentation, we randomly sample ten instances and transform their segmentation masks into bit masks with a size of $16 \times 16$. For keypoint detection, we train MAD on ten person instances per image and only predict keypoints for detected humans (based on object detection results) during inference. We pad blank instances for the above three tasks if there are not enough instances in the image. For image captioning, we adopt sentence piece model (SPM) from T5 [36] for tokenization, and abbreviate its vocabulary based on COCO dataset, resulting in 11421 remaining text tokens. We use loss weights of [1.5, 2.7, 0.5, 0.3] for object detection, instance segmentation, keypoint detection, and image captioning respectively.

At inference time, different tasks in MAD can be conducted in any order. During implementation, we first predict task sequences for object detection as they will serve as prompts for subsequent tasks. For instance segmentation, we directly convert predicted sequences into bit masks based on probabilities of $< Foreground >$ tokens. For keypoint detection, output sequences are dequantized into tuples of keypoint coordinates with probabilities of $< Visible >$ showing their visibility. For image captioning, sequences are truncated by the first predicted padding token and directly mapped back to texts by SPM. We conduct masked inference on keypoint detection and image captioning tasks with mask ratios of 0.7, and 0.8, 0.6, 0.4.

**Implementation Details.** We implement MAD with three different backbones, Swin-Base [31], ViT [13] and Resnet-50 [19] with detrex [40] toolbox. For ViT, we adopt the adjusted architecture of ViTDet [28] (for single feature map output). Following DETR [5], both the encoder and decoder in MAD consist of 6 layers with a main dimension of 256 and 8 attention heads, and the width of FFN is set to 2048. For sequence modeling, we adopt learned positional encodings with a length of 506 to cover all task sequences.

For comparisons with state-of-the-art methods, we train the model with Swin-Base [31] and ViT-Base [13, 14] for 300 and 100 epochs with a learning rate drop after 200 and 80 epochs. Our ablation experiments with the ResNet-50 backbone are trained with a shorter schedule of 50 epochs. More training details can be found in the supplementary material. The inference time (speeds) of all experiments are the total time for inferring on four tasks, tested on a single A100 with a batch size of one image.

## 4.2. Comparison with State-of-the-art Methods

Tab. 1 shows comparisons with the state-of-the-art (SOTA). We compare MAD with two types of SOTA models: (1) typical task-specific models which leverage task-specific designs and are trained on a single task; (2) generalist models which employ a shared single architecture to handle multiple vision tasks without task-specific designs such as region proposal network (RPN) or ROI Pooling. Compared with the task-specific models, we can see that MAD can achieve competitive and even better accuracy without customized architecture for a single task. On top of that, the sequence-based framework in MAD provides significant scalability and flexibility to new tasks or data formats than these models. We adopt a MIM pre-trained ViT from [14] for comparison with Pix2SeqV2 [7] which is pre-trained on Objects365 [41]. Tab. 1 shows that MAD outperforms Pix2SeqV2 by a large margin, especially on vision-centric tasks, demonstrating the effectiveness of our designs on bidirectional attention and masked auto-decoding strategy. Additionally, with the same backbone, MAD is around 157x faster than Pix2SeqV2 (with sequential decoding and complicated post-processing) in inference time. For image captioning, the autoregressive paradigm in existing methods excels us in modeling language sequential context. We will investigate how to combine the advantages of both in the future to enable a more versatile generalist model.

## 4.3. Ablation Studies

**Main Components Ablation.** We first gradually ablate our main designs as shown in Tab. 2. We convert MAD into an autoregressive variant with the same architecture for comparison (Details can be found in the supplementary material). It can be seen that *Autoregressive Decoding* performs worst in terms of both inference time and accuracy on vision tasks except image captioning. This result is consistent with our analysis that the autoregressive approach might not fit well for vision-centric tasks and struggles with extremely slow predictions. By employing bi-directional attention and parallel decoding (i.e., *Parallel Decoding*), the convergence and inference speed of vision tasks are greatly improved.

Table 1. Comparisons for object detection (mAP), instance segmentation (mAP), keypoint detection (mAP), and image captioning (BLEU@4 [33]) on COCO validation set.

| | Backbone | Parameter(M) | Infer. Time(s) | Object Det. | Instance Seg. | Keypoint Det. | Captioning |
|---|---|---|---|---|---|---|---|
| ***Task-specific Models*** | | | | | | | |
| Faster R-CNN [39] | R101-FPN | 42M | - | 42.0 | - | - | - |
| DETR [5] | R101-DC5 | 60M | - | 44.9 | - | - | - |
| Pix2Seq [6] | R101-DC5 | 57M | - | 45.0 | - | - | - |
| Mask R-CNN [20] | X101-FPN | 107M | - | 42.9 | 38.6 | - | - |
| Keypoint R-CNN [47] | R50-FPN | 59M | - | - | - | 65.5 | - |
| Transformer [42] | Encoder | - | - | - | - | - | 34.0 |
| ***Generalist Models*** | | | | | | | |
| VisionLLM [45] | R50+Alpaca-7B | 40M + 7B | - | 44.6 | 25.1 | - | 31.0 |
| Pix2SeqV2 [7] | ViT-B | 132M | 40.86 | 46.5 | 38.2 | 64.8 | 34.9 |
| **MAD** | Swin-B | 107M | 0.19 | 49.7 | 40.6 | 64.6 | 32.2 |
| **MAD** | ViT-B | 107M | 0.26 | 50.1 | 41.2 | 66.5 | 32.6 |

Table 2. Ablation studies on main components of MAD. "AR Decoding" indicates autoregressive decoding.

| Methods | Time (ms) | Det. | Seg. | Kpt. | Cap. |
|---|---|---|---|---|---|
| AR. Decoding | 3953 | 27.9 | 12.3 | 33.4 | 34.1 |
| + Parallel Decoding | 137 | 35.9 | 29.8 | 51.5 | 18.2 |
| + Masked Training | 137 | 38.9 | 32.3 | 54.6 | 18.6 |
| + Masked Inference | 173 | 38.9 | 32.3 | 54.7 | 29.6 |

Table 3. Ablations on Masking ratio strategies. For "random ratio", we used a random mask ratio lying between 0.6 and 0.8. For "multiple ratios", the task sequences are trained with two masking ratios (i.e., 0.6 and 0.8). The "single ratio" indicates that a single mask ratio is adopted.

| Methods | Det. | Seg. | Kpt. | Cap. |
|---|---|---|---|---|
| random ratio | 38.6 | 31.9 | 54.3 | 29.4 |
| multiple ratios | 38.6 | 32.0 | 54.2 | 29.7 |
| **single ratio** | 38.9 | 32.3 | 54.7 | 29.6 |

We then introduce masked autodecoding. As shown in the fourth row, +*Masked training* performs especially better for object detection, instance segmentation, and keypoint detection, thanks to the task context modeled through masking and reconstruction. Moreover, by further introducing masked inference (i.e., +*Masked Inference*), the accuracy is constantly improved with competitive image caption accuracy to the autoregressive counterpart. In addition, we observe that MAD has different effects on vision-centric tasks and language tasks in training and inference. We speculate that MAD in training could model rich task contexts, such as the relationship among task prompts, vocabulary, and sequence patterns, which are crucial for modeling multi-task sequences. On the contrary, during inference, MAD mainly focuses on dependencies among sequence tokens, which are generally rich in language but lacking in visual sequences.

**Convergence Curves for Vision Tasks.** Fig. 4 compares the detailed training curves between methods in Tab. 2 for different tasks. With bi-directional attention, both MAD and parallel decoding converges much faster than autoregressive decoding which adopts uni-directional attention. In addition, the masked sequence modeling strategy in MAD can further capture rich task contexts and largely improve performances, especially for image captioning. These results further demonstrate the non-trivial design of MAD.

Table 4. Ablations on different mask ratios in training.

| Mask Ratio | Det. | Seg. | Kpt. | Cap. |
|---|---|---|---|---|
| 0.4 | 38.4 | 31.8 | 54.4 | 29.2 |
| 0.6 | 38.5 | 31.9 | 54.2 | 29.7 |
| **0.7** | 38.9 | 32.3 | 54.7 | 29.6 |
| 0.8 | 38.7 | 32.1 | 54.9 | 28.4 |

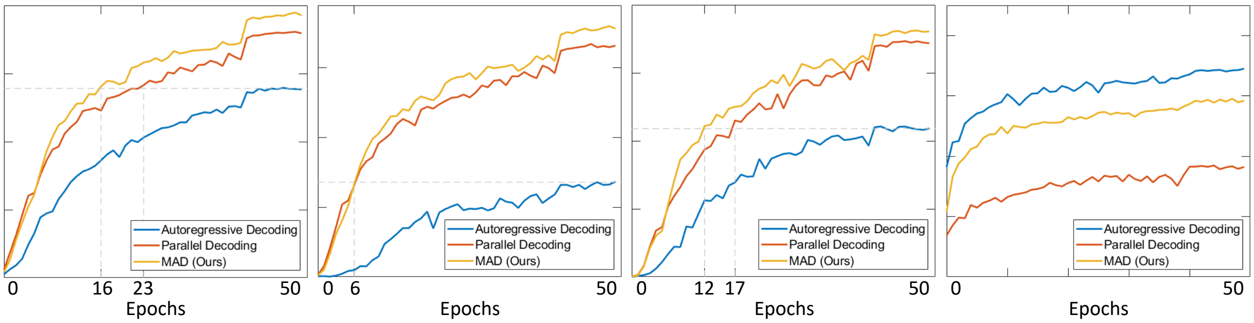
Table 5. Ablations on number of quantization bins for coordinates.

| Number of Bins. | Object Detection | Keypoint Detection |
|---|---|---|
| 300 | 38.5 | 54.2 |
| **500** | 38.9 | 54.7 |
| 800 | 38.6 | 54.5 |
| 1000 | 38.5 | 54.4 |

**Masked Training.** We examine how varying masking strategies and masking ratios affect the training of MAD. As Tab. 3 shows, the simplest strategy with a single masking ratio could achieve the highest performance. As for specific masking ratios in training (under the *single ratio* strategy), Tab. 4 shows that MAD performs the best with a moderate value of 0.7, while a smaller masking ratio results in an over-simplified task, and a larger masking ratio leaves insufficient tokens for modeling task contexts.

**Coordinate Quantization.** We evaluate the effect of the number of the coordinate bins. As Tab. 5 shows, MAD performs robustly under different numbers of bins. We thus adopt 500 as default, while each bin corresponds to approximately 2 pixels for an image with a size between 800 to 1333 pixels, resulting in negligible quantization error.

**Mask Size.** In Tab. 6, we study bit mask sizes. It can be




(a) Object Detection (AP) (b) Instance Segmentation (AP) (c) Keypoint Detection (AP) (d) Image Captioning (B@4)

Figure 4. Convergence curves for *Autoregressive Decoding*, *Parallel Decoding*, and the proposed **MAD** in Tab. 2. MAD achieves much faster convergence for vision-centric tasks and greatly narrows the gap with *Autoregressive Decoding* compared with *Parallel Decoding* for image captioning.

Table 6. Ablations on size of bit mask for image segmentation.

| Mask Size | Object Detection | Instance Segmentation |
|---|---|---|
| 12 | 38.4 | 31.5 |
| 14 | 38.8 | 32.0 |
| **16** | 38.9 | 32.3 |
| 20 | 38.8 | 32.4 |

Table 7. Ablations on inference mask ratios for image captioning.

| Mask Ratio | Image Captioning |
|---|---|
| w/o masked inference | 18.6 |
| {0.7} | 25.8 |
| {0.7, 0.3} | 27.0 |
| **{0.8, 0.6, 0.4}** | 29.6 |

seen that MAD does not benefit much from larger mask sizes, since we do not adopt task-specific operations like ROIAlign [20] or interpolation to align mask pixels and image pixels. Considering that larger mask sizes lead to longer task sequences, we set the mask size at 16 for efficiency.

**Inference Mask Ratio for Captioning.** We examine different inference mask ratios for image captioning. Results in Tab. 7 demonstrate that a combination of gradually decreasing masking ratios ({0.8, 0.6, 0.4}) performs the best.

### 4.4. Joint Training

Tab. 8 shows the performance of MAD under separate training (single-task) and joint training (multi-task), both being equipped with our proposed masked autodecoding. It can be observed that joint training obtains more gains with masked autodecoding. We conjecture that under separate training, the sequence encodings as well as other learnt weights in the decoder are capable of modeling task context of a single task. However, as more vision and language tasks are introduced simultaneously, the jointly trained model might fail to model diverse task contexts and therefore benefit more from our proposed masked autodecoding.

Table 8. Comparing joint training and separated training in MAD. *MM* denotes the proposed masked autodecoding including both **M**asked training and **M**asked inference. For *separate training*, the model is trained on a single task. The models *w/o MM* are conducted with only parallel decoding.

| Methods | Det. | Seg. | Kpt. | Cap. |
|---|---|---|---|---|
| separate training (w/o MM) | 38.4 | 31.2 | 55.1 | 20.6 |
| separate training (w/ MM) | 38.5 | 32.1 | 55.7 | 30.4 |
| joint training (w/o MM) | 35.9 | 29.8 | 51.5 | 18.2 |
| joint training (w/ MM) | 38.9 | 32.3 | 54.7 | 29.6 |

## 5. Conclusion

In this work, we propose Masked AutoDecoder (MAD), a sequence-to-sequence multi-task vision generalist that employs masked sequence modeling and parallel decoding. MAD performs multiple vision tasks with a unified task sequence format, and learns to reconstruct masked task sequences for modeling diverse task contexts. In addition, we employ bidirectional attention and parallel decoding in Transformer, achieving significant speedup in both convergence and inference compared to autoregressive counterparts for vision tasks. Experiments on COCO demonstrate the effectiveness and superiority of MAD as compared with both well-established task-specific models and existing vision generalist models.

**Limitations and Future Work.** In this work, we only explore MAD on four typical vision tasks following [7]. We plan to expand MAD towards more data and tasks in the future work to form a more comprehensive vision generalist and further explore its characteristics.

## Acknowledgement

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

# References

[1] Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3159–3166, 2019. 5

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. 3

[3] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image Transformers. In *ICLR*, 2021. 3

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with Transformers. In *ECCV*, 2020. 5, 6, 7

[6] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021. 3, 7

[7] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey E Hinton. A unified sequence interface for vision tasks. *Advances in Neural Information Processing Systems*, 35:31333–31346, 2022. 1, 3, 4, 5, 6, 7, 8

[8] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 3

[9] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 1931–1942. PMLR, 2021. 3

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, pages 886–893. Ieee, 2005. 3

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[12] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Bootstrapped masked autoencoders for vision bert pretraining. *arXiv preprint arXiv:2207.07116*, 2022. 3

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 6

[14] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 6

[15] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022. 3

[16] Peng Gao, Teli Ma, Hongsheng Li, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders. *arXiv preprint arXiv:2205.03892*, 2022. 3

[17] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019. 3

[18] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017. 3

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3, 7, 8

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 3

[22] Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. In *Advances in Neural Information Processing Systems*, pages 26295–26308. Curran Associates, Inc., 2022. 1, 3

[23] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018. 3

[24] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 4

[25] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. 3

[26] Hao Li, Jinguo Zhu, Xiaohu Jiang, Xizhou Zhu, Hongsheng Li, Chun Yuan, Xiaohua Wang, Yu Qiao, Xiaogang Wang, Wenhai Wang, et al. Uni-perceiver v2: A generalist model for large-scale vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2691–2700, 2023. 3, 5

[27] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 3

[28] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 280–296. Springer, 2022. 6

[29] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3, 6

[30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 3

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision Transformer using shifted windows. In *ICCV*, 2021. 6

[32] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022. 1, 3, 4

[33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 7

[34] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1, 2

[35] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 2

[36] 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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 6

[37] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3

[38] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 3

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 7

[40] Tianhe Ren, Shilong Liu, Feng Li, Hao Zhang, Ailing Zeng, Jie Yang, Xingyu Liao, Ding Jia, Hongyang Li, He Cao, Jianan Wang, Zhaoyang Zeng, Xianbiao Qi, Yuhui Yuan, Jianwei Yang, and Lei Zhang. detrex: Benchmarking detection transformers, 2023. 6

[41] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 6

[42] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 7

[43] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[44] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 1, 3

[45] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023. 1, 3, 4, 6, 7

[46] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 2022. 3

[47] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 7

[48] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A simple framework for masked image modeling. In *CVPR*, 2022. 3

[49] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *European Conference on Computer Vision*, pages 521–539. Springer, 2022. 3

[50] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16804–16815, 2022. 3