基于推理多模态大模型的权重剪枝探究

褚子源

2025 年 8 月 6 日

目录

1 摘要

非结构化剪枝是常用的的大模型压缩和加速技术，通过剪掉冗余的权重，稀疏化权重矩阵来实现大模型的压缩和加速。常见的非结构化剪枝技术比如直接根据 l_1 范数稀疏化和使用标定集计算权重的激活值来指导剪枝都实现了出色的压缩效果也很好地维持了模型性能，但是对于多模态大模型，由于其不同模态 tokens 的分布不同，直接基于 tokens 计算激活值的方式效果不尽人意。在本次实验中，我们对于多模态推理大模型的剪枝做出研究，尝试将基于 LLaVA 的自适应剪枝方法推广到基于 Qwen2.5VL 的具有推理能力的模型上，评测并分析其效果。通过对比自适应剪枝技术和 l_1 范数稀疏化的剪枝技术，我们论证了自适应剪枝技术的优势并分析了两种剪枝技术的差异。最后我们进一步提出了我们对于非结构化剪枝技术未来研究的展望。

2 介绍

近年来，随着大语言模型参数规模的不断增长，模型部署和推理的成本问题日益凸显。模型剪枝，作为模型压缩的重要手段之一，能够在保证模型总体性能的同时，有效地减少了模型的部署规模和计算开销。

2.1 结构化剪枝

结构化剪枝侧重于剪枝模型中的某一完整模块，比如去掉某一个 Transformer layer [1] 或者去掉某一个 Attention head [4]，从而实现减小模型体量，加速生成的效果。结构化剪枝一般具有较高的可部署性，不需要任何硬件友好机制就能真实地提升其运行速度。但是随着大模型技术的不断发展，大模型各个模块的冗余程度不断降低，每一个模块都具有其独特的功能，因此直接去掉模型内部的某些模块或者结构可能会对模型带来较大程度的性能下降。

2.2 非结构化剪枝

区别于结构化剪枝，非结构化剪枝主要聚焦于剪掉模型内部的一些特定权重，将模型内部一些冗余的、影响力较低的权重剪掉，从而维持模型性能的同时减小模型的体量。比如 sparseGPT [2] 提出了一种基于 Hessian 矩阵的模型权重更新方法，在剪掉一些权重的同时更新其他权重从而维持模型性能。除此之外 wanda [6] 提出了更简洁的权重剪枝办法，只需要根据权重的激活值便能够在不更新权重的情况下实现对模型的高效剪枝。在 wanda 的基础上，TAMP [3] 采用了逐层自适应稀疏率和自适应 tokens 选择的方式实现了在多模态大模型上的高效剪枝。在本次实验中，我们主要聚焦于 TAMP 剪枝方式，尝试将其推广到 Qwen-VL 推理大模型上来测试并分析其剪枝效果。

3 实验方案与过程

3.1 TAMP 剪枝

TAMP（Token-Aware Multi-modal Pruning）是一种针对多模态大模型的非结构化剪枝方法。其核心思想是结合模态间差异性与注意力机制，在 token-level 构造剪枝激活，再计算权重的重要性，实现更精细的参数裁剪。该方法的整体流程如下：

1. **计算每层的多样性评分**

   对每层输出 Token，分别计算以下三项多样性指标：

- 视觉模态内部的多样性 s_v
- 语言模态内部的多样性 s_l
- 跨模态之间的多样性 s_{vl}

$$s_v = E_{i,j \in C_v}[d_{i,j}], s_{v,l} = E_{i \in C_v, j \in C_l}[d_{i,j}]$$

$$d_{i,j} = 1 - < Z_i, Z_j >$$

三者的平均作为该层的综合重要性评分：

$$s = \frac{s_v + s_l + s_{vl}}{3}$$

设定该层的稀疏度与 s 成反比，即多样性越高的层保留更多参数。

2. **选择用于激活计算的关键 Token**

   (a) **获取 Token 的注意力贡献分数**从每层 Attention 输出中，取最后一列作为每个输入 Token 的贡献评分。

   (b) **加入邻居信息平滑分数**结合 Token 之间的余弦相似度，引入邻居 Token 的得分实现平滑。

$$a_i \leftarrow a_i + \Sigma_{j \in N_i} exp(-\gamma \times d_{i,j}) \times a_j$$

   (c) **贪婪地选择代表性 Token** 重复以下过程，直到选中的 Token 集合 C' 满足与原始 Token 集合的分布相似性（由 MMD 判定）：

$$MMD = A(C, C) + A(C', C') - 2A(C, C') < 0.1 \times \sqrt{s}$$

$$A(C, C') = \frac{1}{|c||c'|} \times \Sigma_{i \in C, j \in C'} e_{i,j}$$

   - 选择当前贡献分数最高的 Token；
   - 对其邻居施加惩罚，防止冗余；

$$a_j \leftarrow a_j - e_{i,j} \times a_i, \forall j \in N_i$$

   - 添加该 Token 至代表集合 C'。

3. **计算输入激活值**对于上一步选出的代表性 Token 集合 C'，计算其在每个通道上的激活强度，使用 L2 范数：

$$\text{activation} = \|X_{C'}\|_2$$

4. **计算权重的重要性分数**对于每层线性权重 W，使用通道激活值作为加权系数，计算参数重要性：

$$\text{importance} = \text{activation} \times \|W\|_1$$

5. **剪除低得分参数**根据步骤 1 中获得的层稀疏度，剪除当前层中得分最低的参数，形成稀疏网络。

3.2 实验设置

3.2.1 模型选择

在实验中我们选择的是使用 VLM-R1 [5] 框架训练后的 Qwen2.5VL-3B 模型，我们一共尝试了在两种模型上进行剪枝并评测其效果，分别是经过后训练后的开放词目标检测模型和数学推理模型。这两个模型都是基于 Qwen2.5VL-3B 的架构并都分别采用 VLM-R1 框架在特定的领域进行了强化后训练，并达到了较高的评测效果。

3.2.2 baseline 设置

我们选择了较为简单的剪枝方法作为 baseline 与 TAMP 剪枝进行对比，从而评测 TAMP 方式在推理大模型上面的效果。baseline 方法在每一个 linear 中都直接按照权重 l_1 范数大小进行剪枝，将较小的权重剪掉。我们设置的剪枝部分为每一个 Transformer layer 中 Attention 层的线性权重，为了达到控制变量的效果，我们每次控制其平均稀疏率相等。对于 TAMP 的逐层自适应稀疏率而言，每一层稀疏率和总体平均稀疏率的关系如下：

$$\frac{\Sigma_{i=1}^{n} k \times \frac{1}{s_i}}{n} = sparse_{avg}$$

$$sparse_i = k \times \frac{1}{s_i}$$

$sparse_i$ 就是每一层的稀疏率，$sparse_{avg}$ 是平均稀疏率。除此之外我们还将未剪枝的原始权重模型进行了相同的评测，一同对比其效果。

3.2.3 benchmark 选择

对于开放词目标检测模型，我们采用了 OVDEval [9] benchmark 中的数据集进行评测，采用的推理框架是 vllm，评测标准是依据其在图片中预测框和真实框之间的交并比 (IoU)，通过设置交并比的阈值计算其准确率来评测效果。IoU：

$$IoU = \frac{S_{\cap}}{S_{\cup}}$$

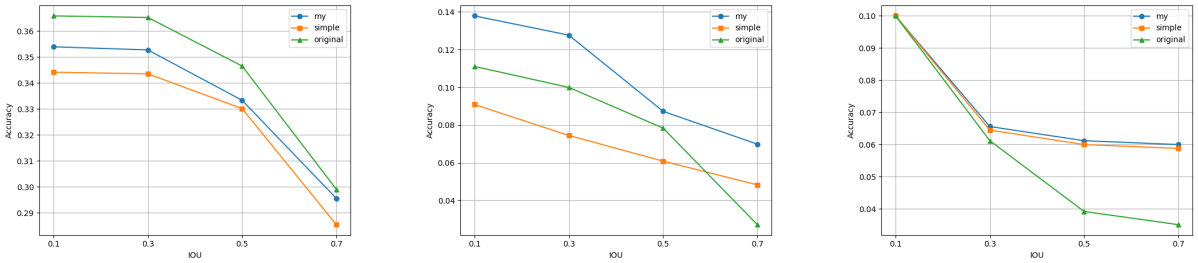
$S_{\cap}$ 是预测框和真实框之间的交集面积，$S_{\cup}$ 是预测框和真实框并集的面积。

对于数学模型，我们采用 MATH-Vision [7] [8] benchmark 中的数据集进行评测，检测模型在不同的平均稀疏率下的准确率。

4 实验结果分析

4.1 开放词目标检测

在 OVDEval 的 benchmark 上，我们尝试在三种数据集上面评测剪枝的效果。这三种数据集分别为 negation，relationship 和 logo。negation 数据集的任务是让模型根据否定命题从图片中寻找目标框，比如：不穿黑色衬衫的人。relationship 数据集的任务是让模型能够区分主语和宾语的检测目标，比如：人拿着的雨伞和拿着雨伞的人。logo 数据集的任务是让模型识别不同的商标，所有这些任务都是 out-of-domain 的。最终评测效果如图 1所示。在这里我们设定的平均稀疏率为 0.3，可以看到 TAMP 剪枝的效果在三个数据集上面的表现均超过了 baseline。



(a) negation (b) relationship (c) logo

图 1: 在三个数据集上面评测效果对比，蓝色为 TAMP 剪枝后的模型效果，黄色为 baseline 剪枝，绿色为剪枝前效果。横坐标为 IoU 阈值，纵坐标为在这一阈值下的准确率。

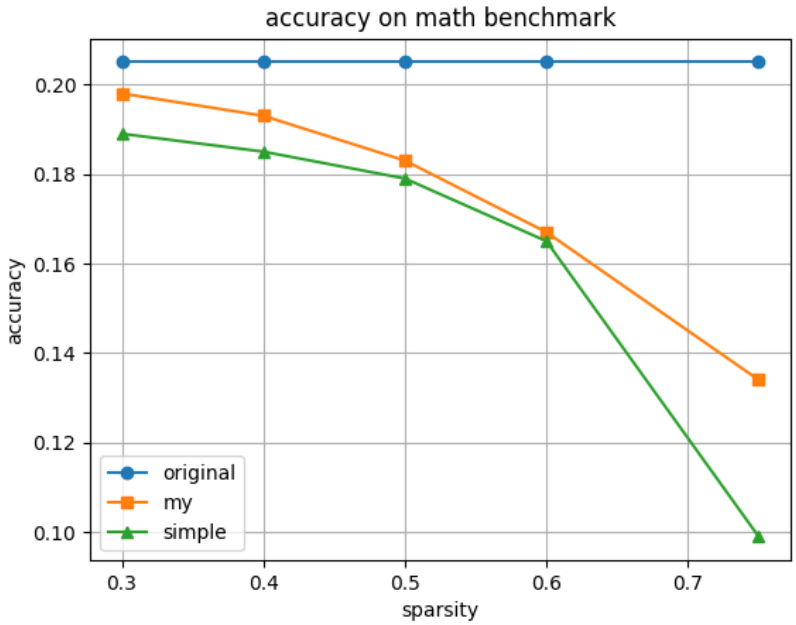


图 2: 在 MATH 数据集上面评测效果对比，蓝色为剪枝前的原始模型，黄色是 TAMP，绿色为 baseline。横坐标为每一个剪枝模块整体的平均稀疏率，纵坐标是在数据集上面评测的准确率。

4.2 数学能力评测

MATH-Vision benchmark 中一共有 3040 道告质量多模态数学题，题目均为选择题或者填空题的形式，因此我在每一个道题的输入 prompt 中强调让模型把最终的答案按照特定的形式给出，从而方便从模型的回复中解析出其答案进行评测。我们在多种不同的平均稀疏率下对模型进行剪枝并评测其效果，对比他们的准确率，同时与剪枝前的模型进行对比，最终的结果如图 2所示。

在稀疏率为 0.75 时，我们详细输出了模型的回答，发现 TAMP 还能够正常给出回复，而 baseline 方法出现了大量的乱码，导致无法解析出正确回复，但是两种方式此时所能达到的准确率都很低，在图中表现出来准确率相差很大主要原因是 baseline 方法输出的大量乱码导致我们解析输出的方式失效，因此，在很高的稀疏率下单纯从准确率上来看不能体现出双方效果的差异。进一步，我们又把稀疏率设置到了 0.90，来观察 TAMP 剪枝方法对于数学问题给出的输出，我们发现此时模型已经无法给出任何解答，但是即便如此，模型仍然能够生成自然语言，不会出现乱码的情况，他给出的回复大部分是类似”I’m sorry, I will not be able to help you ” 的句子。虽然模型无法解答数学

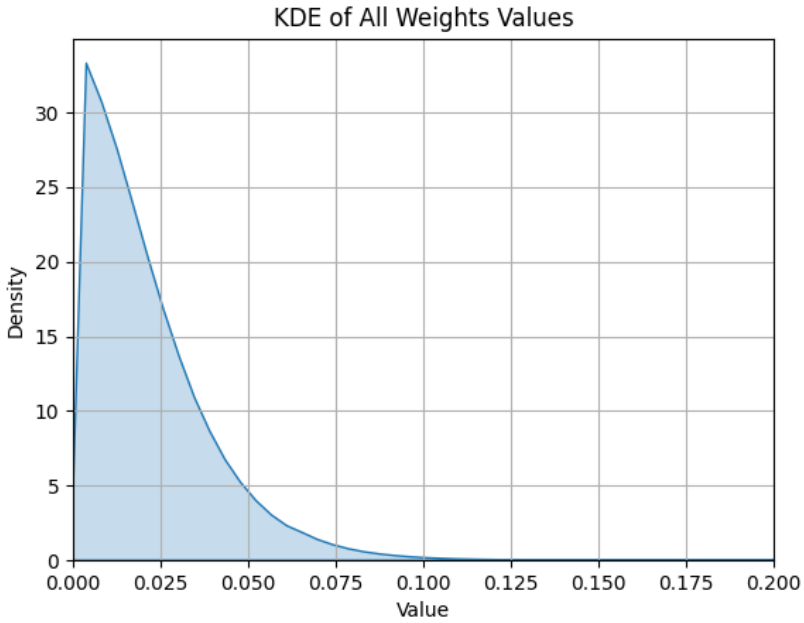


图 3: 模型权重 l_1 范数的分布图，可以看到模型权重的分布十分集中，大部分权重值在 0 到 0.05 之间，这之后权重的分布密度随着权重值的增加呈指数级减小

问题，但是相比于 baseline 他在更高的稀疏率下仍然能生成不带乱码的自然语言，这也能一定程度上体现出 TAMP 方法的优势。

5 思考与讨论

5.1 进一步分析

对于在 MATH-Vision benchmark 上面测评的结果，我们发现在当稀疏率过高导致准确率无法反应其真实表现之前，TAMP 剪枝方法和 baseline 剪枝方法的准确率会随着稀疏率的增加而趋同，也就是说，两种剪枝方法的效果随着稀疏率的增加会越来越相似，我们尝试解释这一现象。

我们随机选择了某一个 Transformer layer 中的某一个 linear 层，并从其中随机抽取了 2000 个权重，我们把他们按照 l_1 范数由小到大排序。同时我们按照 TAMP 的方式求出这些权重的激活值，并将他们按照激活值由小到大排序，我们将这些权重按照他们的 l_1 范数排名和激活值排名画在一个二维坐标系中，从而可视化每一个权重在两种剪枝视角下的重要性，分布图如图 4上图所示。

我们定义权重交并比 IoU_s 为这 2000 个权重在稀疏率 s 下的权重个数交并比。比如：稀疏率为 0.5，那么在这 2000 个权重中，我们要剪掉其中排名前 1000 的权重，对于 baseline 剪枝方法，要剪掉的权重是横坐标小于 1000 的所有权重，而对于 TAMP 方法，要剪掉的权重是纵坐标小于 1000 的所有权重，那么他们的交集便是两种方法共同剪掉的权重，也就是在原点和 (1000,1000) 所围成的正方形所包含的权重，而并集便是这两种方法所剪掉的全部权重，因此其交并比便是交际中权重的个数与并集中权重个数的比值。一般来说，我们认为其交并比越高，便代表两种剪枝方法所剪掉的公共权重占比越大，这也就意味着两种方法的表现差距也就越小。如图 4下图所示，我们计算出了在不同稀疏率下的交并比，可以看到，随着稀疏率的增加，其交并比也在增大，这也就解释了为什么图 2中两种剪枝方法的准确率差距会随着稀疏率增加而变小。

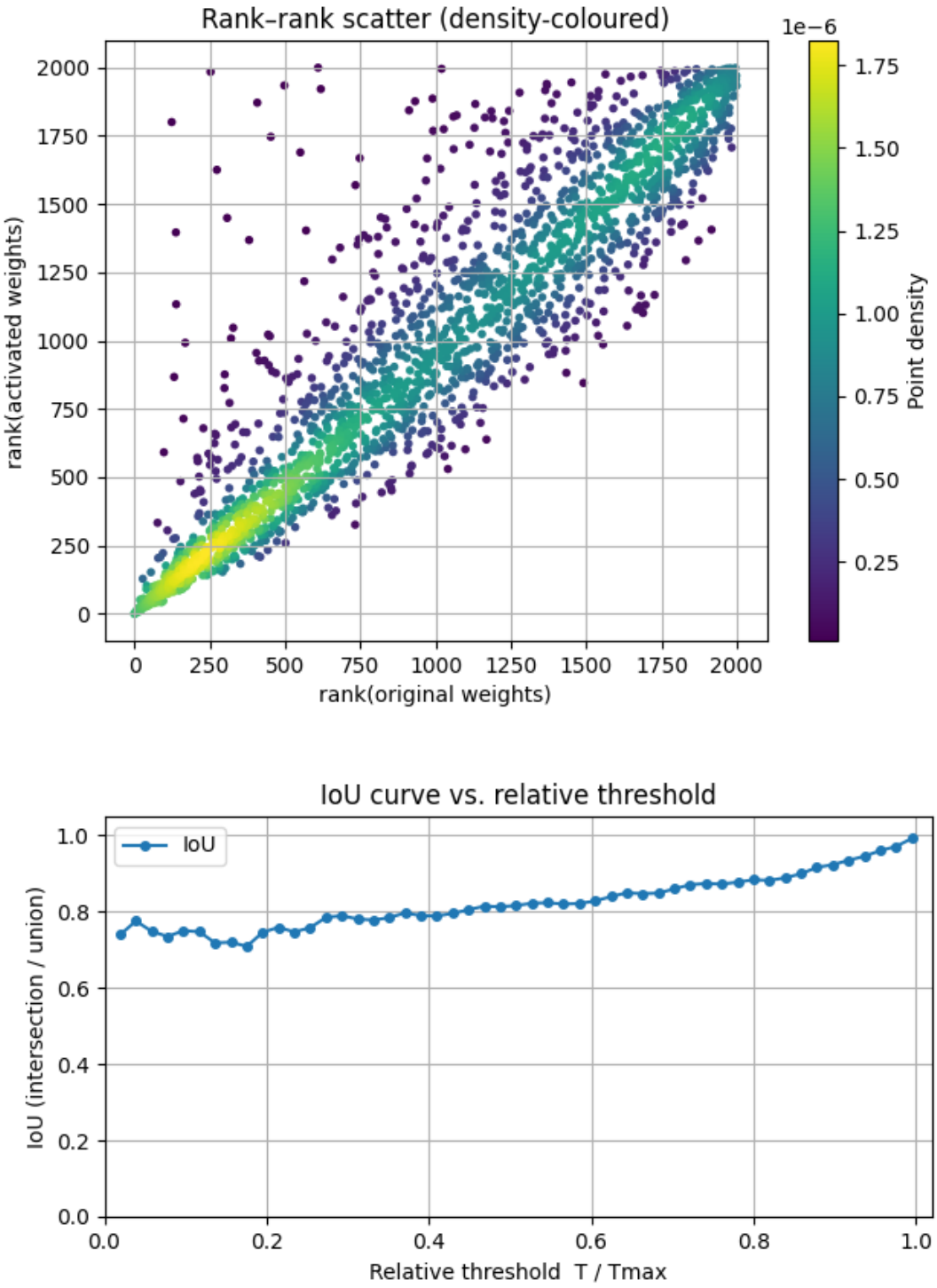

Rank–rank scatter (density-coloured)
rank(activated weights)
rank(original weights)
0
250
500
750
1000
1250
1500
1750
2000
1e−6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
Point density
IoU curve vs. relative threshold
IoU
IoU (intersection / union)
Relative threshold T / Tmax
0.0
0.2
0.4
0.6
0.8
1.0

图 4: 权重分布分析

5.2 展望

大模型权重剪枝的主要目的在于去除冗余，加速大模型的推理。对于非结构化剪枝而言，由于需要采用一些硬件友好机制而将其加速效果体现在真实应用场景下，比如在 NVIDIA Ampere 及以上架构所支持的 N:M 稀疏模式，按照要求进行稀疏化便能够实现矩阵乘法的加速，但是这些需要对非结构化剪枝方法做出适配性的修改，同时一般而言会要求较高的稀疏率 (大于等于 50%)，而实际上通过我们上面的分析，在较高稀疏率下两种剪枝方法的差距并不显著，因此对于非结构化剪枝技术，还需要寻找到效果更好的剪枝算法或者更加普适的硬件友好机制能够实现对较低稀疏率下的加速。

6 总结

在本次实验中，我们对于多模态推理大模型的剪枝做出研究，首先我们仔细分析了基于 LLaVA 的自适应剪枝方法 TAMP 的工作流程，进一步我们又将这一技术推广到了 Qwen2.5VL 推理大模型上，并将基于 l_1 范数的剪枝技术与其进行对比，分别评测了两种方法在多个 benchmark 上面的效果。更进一步，我们尝试分析两种算法之间的异同并尝试解释了他们在评测中的表现，同时也论证了自适应剪枝技术的优势。最后针对我们实验的结论，我们对于非结构化剪枝技术提出了未来研究的展望。

参考文献

[1] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019.

[2] Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR, 2023.

[3] Jaewoo Lee, Keyang Xuan, Chanakya Ekbote, Sandeep Polisetty, Yi R Fung, and Paul Pu Liang. Tamp: Token-adaptive layerwise pruning in multimodal large language models. *arXiv preprint arXiv:2504.09897*, 2025.

[4] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[5] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

[6] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

[7] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[8] Ke Wang, Junting Pan, Linda Wei, Aojun Zhou, Weikang Shi, Zimu Lu, Han Xiao, Yunqiao Yang, Houxing Ren, Mingjie Zhan, and Hongsheng Li. Mathcoder-VL: Bridging vision and code for enhanced multimodal mathematical reasoning. In *The 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

[9] Yiyang Yao, Peng Liu, Tiancheng Zhao, Qianqian Zhang, Jiajia Liao, Chunxin Fang, Kyusong Lee, and Qing Wang. How to evaluate the generalization of detection? a benchmark for comprehensive open-vocabulary detection. *arXiv preprint arXiv:2308.13177*, 2023.