

SLIMLLAVA: AUTOMATIC PRUNING FOR LARGE VISION-LANGUAGE MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

Multimodal large language models achieve satisfying performance in complex reasoning tasks, while still suffers from high model complexity in deployment especially for resource-limited devices. In this paper, we propose an automatic pruning method of large vision-language models for efficient multimodal reasoning. Conventional methods leverage the training data of the original model to select the proper pruning ratio for different network components, while they are infeasible for large vision-language models due to the unbearable search cost caused by web-scale training corpus. On the contrary, we only use a few samples to search the desired pruning policy by maximizing its generalization ability on the unknown training data despite of the model accuracy, so that the optimal accuracy-efficiency trade-off can be obtained for large vision-language models. Specifically, we formulate the generalization gap for the pruning policy based on the structural risk minimization principle. With the task performance and the generalization ability, we iteratively search for the optimal pruning policy in the given search space and optimize the vision projector to evolve the search space with higher upper bound of performance. We conduct extensive experiments on ScienceQA, Vizwiz, MM-vet and LLaVA-Bench datasets for the task of visual question answering. With only 64 samples for pruning policy search, our method achieves 83.05% accuracy on ScienceQA and $\times 1.47$ speedup compared to the dense LLaVA-v1.5-7B model.

1 INTRODUCTION

Multimodal large language models have been widely adopted in complex reasoning tasks such as visual question answering (Shao et al., 2023; Guo et al., 2023), embodied task planning (Huang et al., 2022; Mendez-Mendez et al., 2023) and dialogue systems (Li et al., 2018). Although they achieve excellent performance in the high-level tasks with rich commonsense, the inference stage of large vision-language models (LVLMs) in deployment requires large memory footprint and long latency because of the tremendous neurons. In order to deploy the powerful LVLMs on mobile devices such as cellphones, navigation robots and autonomous vehicles with strict computational resource limit, we are required to reduce the model complexity without obvious degradation on the performance.

Network pruning aims to remove unimportant components including layers, channels, tokens and neurons to reduce the storage and computational complexity without harming the task performance, which have been widely studied for computer vision (Burtsev et al., 2018; Kong et al., 2022), natural language processing (Sanh et al., 2020; Ma et al., 2023) and system control (Tang et al., 2020). The components importance can be defined as the l_1 and l_2 magnitude of weights and activations (Sun et al., 2023), and the Jacobian and Hessian matrix regarding the loss functions (Qiao & Yoo, 1999). Since different components usually have various sensitivity to pruning in the downstream tasks, automatic pruning assigns the optimal pruning ratio to each component in order to achieve higher accuracy-efficiency trade-off. In conventional methods, the training data of the original model is leveraged to evaluate the sampled pruning policy based on the accuracy and the model complexity, which provides feedback to the search algorithm to select better candidates. However, LVLMs require extensive training corpus, and the search cost associated with existing automatic pruning methods is prohibitively high.

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

In this paper, we present SlimLLaVA that automatically prunes LVLMs to achieve low model complexity and unaffected task performance with acceptable search cost. Unlike existing automatic pruning methods that leverage the large-scale training data of the original models for pruning policy search, our method only use a few samples to assign the optimal pruning ratio for weight matrix in each layer. Since we maximize the generalization ability of the pruning policy to the unknown web-scale training corpus, the acquired solution achieves satisfying trade-off between accuracy and efficiency in a wide variety of downstream tasks. More specifically, we first formulate the generalization gap of the pruning policy between the proxy samples for policy search and the unknown training set via structural risk minimization principle, which can be evaluated by the Frobenius norm of the weight matrix in the pre-trained LLMs. We then evolve the pruning policy candidates with the goal of achieving higher task performance and generalization ability, where the weights of MLP and attention are pruned based with the selected pruning ratio. The vision projector is optimized to enhance the upper bound of the accuracy and the generalization ability for all policies in the pruning space, which is estimated via the Euclidean distance between the sampled policy and neighbor candidates. The policy candidate selection and the pruning space evolution are implemented iteratively to acquire satisfying accuracy-efficiency trade-off in the extremely large pruning space with acceptable cost. Figure 1 demonstrates the trade-off between accuracy and inference latency on ScienceQA for uniform pruning methods SparseGPT (Frantar & Alistarh, 2023) and our SlimLLaVA. We conducted extensive experiments in a wide variety of multimodal reasoning tasks including short-answer, option-only for multiple-choice and natural QA, and the results shows that SlimLLaVA can achieve 83.05% accuracy on ScienceQA and $\times 1.47$ speedup compared to the dense LLaVA-v1.5-7B model. Our contributions can be summarized as follows:

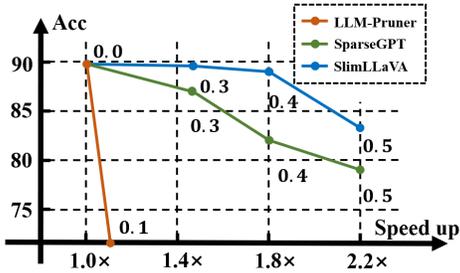


Figure 1: The trade-off between accuracy and inference speed on ScienceQA with SlimLLaVA.

- We introduce SlimLLaVA, a novel pruning method for large vision-language models that effectively reduces model complexity while preserving task performance with minimal data usage.
- Our approach leverages the structural risk minimization principle to maximize the generalization ability of pruning policies, enabling efficient adaptation to unknown training data.
- Extensive experiments demonstrate that SlimLLaVA achieves a significant speed-up in inference while maintaining competitive accuracy across various multimodal reasoning tasks.

2 RELATED WORK

Large Vision-language Models: Large Vision-language Models(LVLMs) fuse visual and linguistic modalities via bridging the gap between different modalities that achieved outstanding performance such as in-context predictions (Liu et al., 2023; Salewski et al., 2024), multi-image and chain-of-thought reasoning (Yang et al., 2023; Driess et al., 2023). To utilize learnable connector module projecting visual information into language modal space, token-level fusion and feature-level fusion are presented for encoding multimodal information to LLMs. For token-level fusion, LLaVA (Liu et al., 2024b) series utilized linear MLPs to align the image tokens with language tokens, finetuning LVLMs with visual instruction samples to enhance zero-shot capabilities. MM1 (McKinzie et al., 2024) compared the significance of varied architectural aspects and found the number of visual tokens and input resolution was more important than the design of projector. For feature-level fusion, Flamingo (Alayrac et al., 2022) inserted extra cross-attention layers to connect pre-trained vision-only and language-only model, significantly boosting learning efficiency. CogVLM (Wang et al., 2024) integrated a visual expert module into every Transformer layer to fuse the vision and language features. Existing methods usually focus on the performance of LVLMs, ignoring the real-world deployment particularly in environments with limited resources. Despite light-weight LVLMs such as MobileVLM (Chu et al., 2023), TinyLLaVA (Zhou et al., 2024) and quantized methods such as GPTQ (Frantar et al., 2022), AWQ (Lin et al., 2023) have reduced the size of LMMs, the cost still exceeds the resource constraint of mobile devices and they fail to consider the intermodal interaction. Differently, our method focus on pruning the pre-trained LVLMs to find the optimal trade-off between accuracy and efficiency with few-shot proxy samples.

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

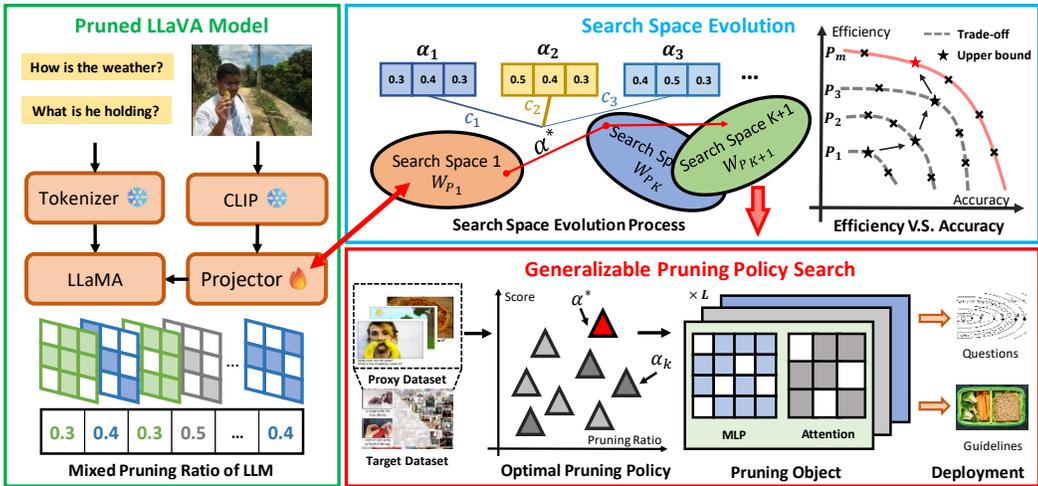


Figure 2: The overall pipeline of SlimLLaVA. In each iteration, we first search for the optimal pruning policy for matrix in each LLaMA layer, where evolutionary algorithms are employed with the fitness function containing model accuracy and generalization ability. Then we evolve the search space by optimizing the projector weight so that the upper bound of accuracy and generalization ability for all policies can be improved.

Automatic Pruning: Automatic pruning(Luo & Wu, 2020; Lin et al., 2020) removes unnecessary connections, neurons, and layers to reduce the model size, while the model accuracy remains high. The early work of automatic pruning AMC (He et al., 2018) utilized reinforcement learning to effectively explore the design space automatically, thereby enhancing the quality of model compression. To achieve fine-grained pruning across samples, dynamic pruning is proposed to predict an instance-wise pruning strategy for each input sample. AutoPrune (Xiao et al., 2019) pruned the network through optimizing a set of trainable parameters instead of original weights. DynamicVit (Rao et al., 2021) progressively and adaptively removed unnecessary tokens based on the input due to the sparse attention. AutoCompress(Liu et al., 2020) introduces an automatic structured pruning framework that improves model compression and inference speed using ADMM-based pruning and heuristic search, achieving high pruning rates. When pruning large pre-trained neural networks, the training set for the original model cannot be acquired. To address this, post-training pruning(Kwon et al., 2022; Lazarevich et al., 2021) derives the sensitivity of each component to pruning by first-order and second-based gradient information. Dynamic Context Pruning (Anagnostidis et al., 2024) utilized a learnable mechanism that selects uninformative tokens to be removed from the context at various stages of the generation process. However, traditional automatic pruning methods are not satisfied for LVLMs due to ignoring the generalization gap between training data and few-shot proxy samples. Different from the previous works, our method evolve the search space to find the optimal pruning policy maximizing the generalization ability.

3 APPROACH

In this section, as illustrated in Figure 2, we first introduce the preliminary of automatic pruning for LVLMs. We then formulate the generalization gap between the few-shot proxy samples for pruning policy search and the overall data distribution, and search the pruning policy with generalizability maximization in given search space. Finally, we evolve the search space to improve the upper bound of accuracy and generalizability ability for all policies. We ensure that the search space remains flexible and evolves dynamically with each iteration, improving the overall pruning performance.

3.1 AUTOMATIC PRUNING FOR LVLMs

Model pruning aims to remove unnecessary or redundant elements to reduce the size of neural network and computational complexity, while minimizing the impact on performance. Automatic pruning assigns optimal pruning ratio to different components to achieve the optimal trade-off

162 between accuracy and efficiency. However, conventional automatic pruning methods (Gao et al.,
 163 2022; Anagnostidis et al., 2024) search the pruning policy on the large-scale training data, which is
 164 impractical for LVLMs due to the prohibitive training cost and the lack of the training corpus. The
 165 goal of automatic model pruning is to minimize loss objective with the sparsity constraint:

$$166 \min_{\alpha} L_{train}(\hat{\mathbf{M}}(\alpha), \alpha)$$

$$167$$

$$168 \text{ s.t. } \|\alpha\|_1 = C_0, \hat{\mathbf{M}}(\alpha) = \arg \min L_{train}(\mathbf{M}, \alpha) \quad (1)$$

169 where L_{train} represents the loss in the large-scale training set for LVLMs. α refers to the pruning
 170 policy, which determines how much of each layer or component to prune. The pruning policy α
 171 denotes sparsity ratio for weight matrix in each layer, and $\|\cdot\|_1$ means the L_1 norm. This ensures that
 172 a certain proportion of the model’s parameters are removed (pruned) while minimizing the impact on
 173 accuracy. C_0 stands for the overall sparsity limit for the entire LVLM. The optimal pruning masks
 174 for weights $\hat{\mathbf{M}}(\alpha)$ are acquired by minimizing the loss on the training data. Due to the limited proxy
 175 samples during pruning policy search of pre-trained LVLMs, the acquired policy usually overfits
 176 the proxy data without generalization to the entire dataset. Therefore, we should also consider the
 177 generalization gap between the provided proxy samples and the real training data distribution for the
 178 pruning policy, and we can obtain pruning policies with satisfying accuracy-efficiency trade-off in
 179 diverse downstream tasks.

181 3.2 GENERALIZABLE PRUNING POLICY SEARCH FOR LVLMs

182 Evaluating the true network objective for searched pruning policies is very challenging because of the
 183 limited samples compared with the entire training data. Therefore, we maximize the generalization
 184 ability of the pruning policy via the structural risk minimization (SRM) principle, which can be
 185 bounded by the empirical risk and the unseen data distribution. Structural Risk Minimization (SRM)
 186 is a principle from statistical learning theory that seeks to balance model complexity with training
 187 data performance, minimizing the risk on both known and unseen data. The optimal trade-off between
 188 complexity and model performance ensures that our approach can be applied effectively in various
 189 real-world applications. For a dataset containing N samples in pruning policy search, the structural
 190 risk is written in the following with the probability at least $1 - \epsilon$:

$$191 \mathbb{E}_r[L(\mathbf{M}, \alpha)] \leq \mathbb{E}_e[L(\mathbf{M}, \alpha)] + 2\mathcal{R}(\mathcal{F}) + \sqrt{\frac{\ln 4/\epsilon}{N}} \quad (2)$$

192 where \mathbb{E}_r and \mathbb{E}_e are the expectation over the unknown real data distribution and the empirical
 193 expectation over limited proxy samples for policy search. $\mathcal{R}(\mathcal{F})$ means the Rademacher complexity
 194 over the objective function class \mathcal{F} . A lower Rademacher complexity indicates better generalization
 195 on unseen data. Since the empirical risk represents the performance of the pruning policy on the
 196 limited provided samples, we minimize the Rademacher complexity to reduce the gap between the
 197 empirical risk and the real structural risk. The Rademacher complexity means the upper bound of the
 198 weighted sum of objectives across samples for all possible objectives. Following (Neyshabur et al.,
 199 2015), we efficiently evaluate the Rademacher complexity via the matrix norm for all layers in the
 200 model:

$$201 \mathcal{R}(\mathcal{F}) = \frac{1}{N} \cdot \mathbb{E}_{\sigma} \left[\sup_{L \in \mathcal{F}} \sum_{i=1}^N \sigma_i L_{x_i}(\mathbf{M}, \alpha) \right] \propto \eta \prod_{i=1}^K \mathcal{A}_i \quad (3)$$

202 where L_{x_i} represents the loss function of the i_{th} sample x_i for policy search, and $\{\sigma_i\}$ are independent
 203 random variables drawn from the Rademacher distribution. \mathcal{A}_i stands for the Frobenius norm of the
 204 parameter matrix in the i_{th} layer, and η can be regarded as a constant related to the layer number
 205 and the sample size. Low weight norm indicates that the model output is weakly correlated with the
 206 input, which leads to high generalization ability for different data distribution. Despite of selecting
 207 components with satisfying accuracy-efficiency trade-offs on the limited proxy samples, we also
 208 encourage the remaining components to have low weight norm. Therefore, the acquired pruning
 209 policy can be adapted to downstream tasks with unseen diverse data distribution.

210 Since the LLaMA model in multimodal LLMs contribute significantly to the overall model complexity,
 211 we prune the weight matrix for each MLP in the LLaMA model. The reason for excluding the visual
 212 encoder and projector from pruning is that pruning these components minimally impacts computation
 213

but can cause significant performance degradation. The k_{th} element of α represents the pruning ratio of the k_{th} MLP layer. The fitness function J for generalizable search can be written as follows:

$$\max_{\alpha} J(\alpha) = \mathbb{E}_e[L(\mathbf{M}, \alpha)] + \eta \prod_{i=1}^K \mathcal{A}_i(\mathbf{M}_i, \alpha_i) \quad (4)$$

where $\mathcal{A}_i(\mathbf{M}_i, \alpha_i)$ means the matrix norm of the i_{th} layer for the pruning ratio α_i and the pruning mask \mathbf{M}_i . Given the pruning ratio for each layer, we acquire the pruning mask by OBS algorithm (Frantar & Alistarh, 2023) removing weights that have lowest influence on the layer output. The OBS (Optimal Brain Surgeon) algorithm optimally removes weights while minimizing the increase in loss, thus preserving model performance. We employ the evolutionary algorithms to search for the optimal pruning policy.

3.3 SEARCH SPACE EVOLUTION OF PRUNING POLICIES FOR LVLMS

LVLMS align the visual tokens with the language tokens via the projector layer, so that the knowledge learned in the pre-trained CLIP visual encoder and the LLaMA model can be fully leveraged for multimodal reasoning. Therefore, we can optimize the weights of the projector, where the possible highest accuracy and generalizability can be acquired for the pruning policies given the resource budget. In this context, the projector plays a crucial role in fusing visual and language modalities, making it a key target for optimization in ensuring high model performance even after pruning. The projector weights are regarded as the search space of the pruning policy search in LVLMS, and optimizing the projector weights is equivalent to search space evolution. The objective for the evolution is as follows:

$$\max_{\mathbf{W}_p} \sup_{\alpha} J(\mathbf{W}_p, \alpha) = \sup_{\alpha} \left[\mathbb{E}_e[L(\mathbf{W}_p, \mathbf{M}, \alpha)] + \eta \prod_{i=1}^L \mathcal{A}_i(\mathbf{M}, \alpha) \right] \quad (5)$$

where $J(\mathbf{W}_p, \alpha)$ means the fitness function with respect to the projector weight \mathbf{W}_p and the pruning policy α , and $L(\mathbf{W}_p, \mathbf{M}_w, \alpha)$ means the loss function of the model regarding the projector weight despite of the pruning mask and the pruning policy. The optimization goal here is to find the best combination of projector weights and pruning ratios to maximize performance. This step ensures that the search space continually adapts and improves, rather than being static. By maximizing the upper bound of the fitness function with search space evolution, pruning policies with higher performance and generalizability appear in the evolved search space. However, estimating the upper bound of the fitness function over all pruning policies is infeasible as numerating the countless selections is computationally prohibited. However, calculating the upper bound for all pruning policies is computationally infeasible. We relax the original problem as the following one with the weighted fitness function \hat{J} :

$$\max_{\mathbf{W}_p} \hat{J}(\mathbf{W}_p, \alpha) = \mathbb{E}_e[c_m L_m(\mathbf{W}_p, \mathbf{M}, \alpha)] \quad (6)$$

where c_m and L_m represent the importance weight and the loss function for the m_{th} candidate pruning policy in the current search space. The candidate that is closer to the upper bound should be considered with higher importance in the search space evolution, and the corresponding importance weight should be larger. Meanwhile, the matrix norm is independent of the projector weights, and it is omitted in the optimization objective of the projector weights. The importance weight is estimated via the following rule:

$$c_m = \frac{\exp(-\sum_{\alpha \in \mathcal{N}_m} \|\alpha_m - \alpha\|^2)}{\sum_k \exp(-\sum_{\alpha \in \mathcal{N}_k} \|\alpha_k - \alpha\|^2)} \quad (7)$$

where \mathcal{N}_m means the neighborhood of the m_{th} candidate pruning policy defined by Euclidean distance. The normalized importance weight c_m assigns higher values to candidates with closer neighbors, implying that candidates in dense regions of the search space (close to other candidates) are considered more important. Different candidates tend to converge towards the optimal pruning policy, and a policy with neighbors at shorter distances suggests a higher likelihood of being optimal, as it is closer to the optimal pruning solution. This weight calculation ensures that pruning policies with stronger potential for optimality are prioritized during the search process, improving the overall quality of the evolved search space.

Algorithm 1: Search Space Evolution of Generalizable Pruning Policy

Input : Initial search space \mathbf{W}_p^0 , Pruning ratio constraint C , max evolution step τ , the number of pruning policy sampling n , the size of evolution group

Output : The optimal pruning ratio α , search space \mathbf{W}_p

```

1  $G_0 :=$  Randomly sample  $n$  pruning policy  $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$  with the constraints  $C$ 
2 while evolution step  $t \leq \tau$  do
3   Prune networks base on the OBS pruning method and pruning policy  $G_t$  in the  $t_{th}$  round
4   Obtain top- $k$  candidates with fitness function (4)
5   Generate  $G_{t+1}$  according to evolutionary algorithm
6   Predict the optimal pruning policy via Euclidean distance in (7)
7   Evolve the search space  $\mathbf{W}_p^t$  base on (6)
8 end

```

The overall pipeline for automatic pruning of LVLMS is demonstrated in Algorithm 1. For a pre-trained multimodal LLM, we first search for the optimal pruning policy with the limited proxy samples given the search space represented by the projector weights. Then we remove the unimportant weights for each layer with the searched pruning ratio via the OBS pruning method. Finally, we optimize the projector weight for search space evolution. The above three stages are iteratively implemented in each round until reaching the search cost constraint.

4 EXPERIMENTS

In this section, we conducted extensive experiment for LLaVA architectures on multi-modal question answering datasets. Firstly, we introduce the implementation details of our SlimLLaVA and the dataset information. Then we conduct ablation study to evaluate the effectiveness of the generalizable pruning policy search and the search space evolution of pruning policies. Finally, we compare the performance regarding accuracy and efficiency with existing pruning methods.

4.1 SETUPS

Dataset: We conduct the experiments on ScienceQA (Lu et al., 2022), Vizwiz (Bigham et al., 2010), MMVet (Yu et al., 2023) and LLaVA-Bench (Liu et al., 2024b). ScienceQA is the first large-scale multimodal dataset that annotates lectures and explanations for the answers containing 21k multiple-choice science questions. Moreover, ScienceQA dataset encompasses vision-language pairs that can be categorized into subjects such as natural science (NAT), social science (SOC), and language science (LAN). Additionally, another classification based on context modality includes text context (TXT), image context (IMG), and no context (NO). ScienceQA dataset stands out for having the largest collection of images, covering all 12 grades, containing the longest questions, and featuring the most diverse input sources. Vizwiz arose from a natural setting for visual question answering with blind individuals, together with 10 crowdsourced answers per visual question, contributing images and spoken questions, which contains 20k image-question pairs. MMVet leverages the amalgamation of various core visual-linguistic competencies to address complex challenges which defines 16 emergent tasks of interest integrated from the six defined core VL capabilities. LLaVA-Bench is a benchmark designed to test how well multimodal AI models can handle real-world visual tasks by engaging in open-ended conversations based on diverse images. These datasets present distinct challenges, from the complexity of real-world queries to specialized reasoning tasks, ensuring that SlimLLaVA’s performance is tested under a variety of realistic conditions.

Implementation Details We evaluate our automatic pruning method for pre-trained LVLMS including LLaVA-SQA-7B (Liu et al., 2024b), which is finetuned on ScienceQA dataset and LLaVA-v1.5-7B (Liu et al., 2024a), which achieves SoTA on a broad range of 11 tasks. The average pruning ratio p_0 for the entire model is set to 0.3, 0.4 and 0.5. To ensure the average pruning ratio of all candidate pruned networks remains consistent, the choices of pruning ratio for each layer is selected from $[p_0 - 0.1 : 0.05 : p_0 + 0.1]$. This ensures the pruning process adapts to the structure of each layer without significantly affecting the model’s integrity. We also set the upper and lower bounds for the generated candidate pruning ratios as $p_0 + 0.01$ and $p_0 - 0.01$. This range is carefully chosen

Table 1: Comparison of average pruning ratio and the accuracy evaluated on ScienceQA across different pruning method.

Criteria	Avg ratio	Accuracy
Uniform	0.50	80.45
Auto w/o Gen	0.49	81.34
SlimLLaVA	0.50	83.05

Table 2: The effects of search space evolution regarding the average pruning ratio and the accuracy.

technique	Avg ratio	Accuracy
No Evo	0.50	81.07
Evo w/o UB	0.51	82.85
SlimLLaVA	0.50	83.05

to allow sufficient flexibility for pruning each layer while maintaining overall consistency with the target pruning ratio. For proxy data, we use 64 segments with 256 tokens for each segment which are randomly sampled from the PTB dataset (Marcus et al., 1993) which is versatile and has been used in various tasks, covers a wide range of English language constructs, making it a valuable resource for understanding and modeling the syntactic and semantic aspects of the language. It embodies generic data extracted from internet sources, which indicates the zero-shot settings of our experiments because no task-specific data is provided. We evolve 16 policy candidates in each round, and the value of the hyperparameter η in LLaVA-SQA-7B pruning is set to 0.5. This number of policy candidates ensures sufficient exploration of the pruning space without incurring excessive computational overhead. Given the searched pruning ratio, we use the same pruning methods as in SparseGPT (Frantar & Alistarh, 2023) for model pruning to fully leverage the potential in network pruning with high degrees of freedom. We also sparsify Transformer layers sequentially and only prunes the matrix of attention weights and MLP weights in the pre-trained LLaMA model. During search space evolution, we select top-5 nearest neighbors to compute the importance weight in (7). The number of rounds containing policy search, weight pruning and search space evolution is 10.

4.2 ABLATION STUDY

Effects of generalizable automatic pruning: Automatic pruning determines optimal pruning ratios for various components to achieve the ideal trade-off between accuracy and efficiency. In order to minimize the generalization gap between the provided samples and training data distribution, we significantly enhance the generalization ability of the pruning policy via the structural risk minimization (SRM) principle. The use of SRM allows us to balance the pruning policy across layers, taking into account both the accuracy on the proxy samples and the unseen data. This leads to the optimal pruning policy, especially on challenging tasks where overfitting to the proxy samples could otherwise degrade performance. We compare our pruning method with uniform pruning policies (Uniform) and that without considering the generalization ability (Auto w/o Gen). Table 1 shows the average pruning ratio of the pruned model and the corresponding accuracy on ScienceQA, where both the automatic pruning and generalization improvement significantly contribute to the accuracy-efficiency trade-off on the acquired pruning policy. Uniform pruning policy ignores the layer-wise importance variance of pre-trained LLMs, and searching the pruning policy on limited proxy samples without considering the generalization gap substantially degrades the performance of the policy on downstream tasks due to the overfitting.

Influence of number of proxy samples and token length: The pruning results and training cost are influenced by the number of proxy samples n and the length of tokens s during the pruning process. To find the optimal trade-off between the accuracy-efficiency trade-off and the search cost, we varied the value of n and s to observe their effects on the pruning outcomes and computational cost. Figure 3 demonstrates the results where the pruning ratio constraint was set to 0.4. Increasing the number of proxy samples and the token lengths can both improve the accuracy given the resource budget, while also brings significant search cost burdens to the automatic pruning framework. Meanwhile, the marginal improvement becomes negligible when the number of proxy numbers and the token length exceed

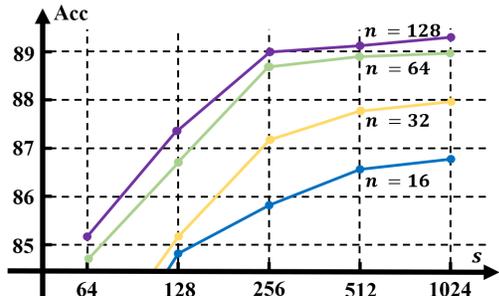


Figure 3: The performance variation with the number of sampled data n and sequence length s in pruning space evolution tested on ScienceQA.

Table 3: The accuracy on ScienceQA for LLaVA-SQA-7B and LLaVA-v1.5-7B, where we evaluate the models across a wide range of domains. Since LLM-Pruner fully loses its question-answering inference capacity when the pruning ratio is above 0.4, comparisons for a pruning ratio of 0.5 are not considered.

	Ratio	Method	Subject			Context Modality			Grade		Average
			NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	
LLaVA-SQA-7B	0.0	-	89.39	96.06	85.64	88.71	87.65	88.50	90.93	87.80	89.91
	0.3	LLM-Pruner	39.34	27.22	47.55	39.69	34.56	43.76	38.22	40.21	38.93
		SparseGPT	88.28	95.16	85.45	87.39	86.71	88.01	89.54	88.00	88.99
		Ours	89.39	93.36	86.45	88.51	86.47	89.34	89.98	88.53	89.46
	0.4	LLM-Pruner	0.18	0.11	0.64	0.20	0.15	0.49	0.18	0.46	0.28
		SparseGPT	80.73	89.65	79.82	80.40	80.17	81.32	83.99	79.43	82.36
		Ours	89.17	91.34	85.37	88.27	85.57	88.57	89.39	87.54	88.73
	0.5	SparseGPT	80.28	89.54	73.45	79.23	78.98	77.84	82.56	76.66	80.45
		Ours	83.61	90.44	75.91	82.80	83.54	79.51	84.73	80.03	83.05
	LLaVA-v1.5-7B	0.0	-	62.57	69.07	65.55	62.81	63.26	63.90	68.65	57.61
0.3		LLM-Pruner	21.67	11.36	12.73	23.17	21.52	13.45	16.30	18.79	17.19
		SparseGPT	58.88	68.05	63.45	59.24	61.87	61.39	65.12	56.36	61.99
		Ours	64.03	67.15	69.36	63.20	62.96	68.9	68.98	60.84	66.07
0.4		LLM-Pruner	0.18	0.11	0.45	0.20	0.15	0.35	0.18	0.33	0.24
		SparseGPT	56.97	65.80	60.55	57.04	59.44	58.40	63.44	63.13	59.75
		Ours	60.35	67.27	66.91	60.26	61.77	64.53	66.74	57.68	63.50
0.5		SparseGPT	53.51	57.71	56.82	54.01	54.64	54.98	56.86	52.34	55.25
		Ours	56.57	56.13	60.55	56.79	54.24	58.54	59.73	53.53	57.51

64 and 256 respectively. In order to acquire the satisfying pruning policy with acceptable search cost, we select the number of proxy numbers and the token length to be 64 and 256 for the rest of the experiments. Moreover, we can also optimize the balance between resource consumption and performance, allowing them to meet their specific requirements.

Effects of search space evolution: Optimizing the projector weight in LVLMs can further significantly enhance the search space for pruning ratio, where the possible highest accuracy and generalization ability can be achieved. To verify the effectiveness of search space evolution, we conduct the ablation study to evaluate different variants: (a) the method without space evolution (No Evo), (b) the method with space evolution while failing to consider the upper bound in evolution (Evo w/o UB), which means all c_m are set to 1, (c) our method. Table 2 illustrates the performance including the average pruning ratio and the model accuracy. Evolving the search space can enhance the accuracy-efficiency trade-off of the acquired pruning policy, because possible highest accuracy and generalization ability are enhanced. Evaluating the upper bound of the fitness function with the average value degrades the accuracy-efficiency trade-off because of the discrepancy of the optimization objective.

4.3 COMPARISON WITH STATE-OF-THE-ART METHODS

In this section, we compare the model pruning techniques in SlimLLaVA with state-of-the-art pruning strategy. As far as we know, we are the first multimodal LVLMs pruning method, and we utilize the strategy designed for LLM pruning including LLM-Pruner (Ma et al., 2023) and SparseGPT (Frantar & Alistarh, 2023) as the baseline method for further comparison. Finally, we provide two visual answers generated by LLaVA-SQA-7B.

Performance Analysis: The task of real-world scientific question answering can be leveraged as an effective way to evaluate the performance of large multimodal models. In order to demonstrate our pruning policy applicable to different models, Table 3 illustrates the accuracy on ScienceQA across LLaVA-SQA-7B and LLaVA-v1.5-7B architectures with variable average pruning ratios. LLaVA-SQA-7B first generates a reasoning process based on the given question, and subsequently derives

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485



 Is Nerium oleander made up of many cells?

 **LLM-Pruner:**
Yes. Some phyla have many different types of organisms. **Other phyla have only one member.** Nerium oleander is a plant. Plants are made up of many cells.

 **SparseGPT:**
How many cells do they have? one | one | one or many | one or many | many | many

 **SlimLLaVA:**
Yes, Nerium oleander is a plant. Plant cells have many more cell types than most animal cells. In a plant, each cell does a different job. Some cells make sugar that the plant can use as food. Other cells make the substances that help the plant grow and survive.

 chocolate syrup	 chocolate milkshake
 tape	 wet glue

 Which property do these four objects have in common?

 **LLM-Pruner:**
The tape and the melted marshmallow are not sticky and stretchy. The property that all four objects have in common is slippery.

 **SparseGPT:**
The tape is stretchy, but the glass bottle is not. The property that all four objects have in common is slippery.

 **SlimLLaVA:**
A stretchy object gets longer when you pull on it. The chocolate milkshake is not stretchy. A sticky object can attach or stick to other things. All four objects are sticky. A slippery object is hard to hold onto or stand on. The tape is not slippery. The property that all four objects have in common is sticky.

Figure 4: Visual examples from LLaVA-SQA-7B. We color the text to show the response of different pruning methods and SlimLLaVA consistently delivers more refined, contextually appropriate responses, showcasing its superior pruning and reasoning capabilities.

the final answer from this reasoning process. On the contrary, LLaVA-v1.5-7B directly generates the final answer without an intermediate reasoning process. Given that LLM-Pruner completely loses its question-answering inference ability at a pruning ratio above 0.4, we do not make comparisons at a pruning ratio of 0.5. SparseGPT (Frantar & Alistarh, 2023) can reach 50% unstructured sparsity while the accuracy degradation is 9.5%. However, it ignores the significance of various layers and generalization gap between the provided proxy samples and the real training data distribution, which leading to the performance degradation. On the contrary, by applying the structural risk minimization principle, SlimLLaVA formulates the generalization gap of the pruning policy and refines pruning policy candidates to enhance task performance and generalization ability. Generally speaking, with an average pruning rate of 0.5, the presented search space evolution for pruning boost the accuracy by 2.6% (80.45% vs. 83.05%) and speed up $\times 2.16$ compared by the none pruned model which get better result.

In order to demonstrate the generalization ability of our pruning policy, we also evaluate LLaVA-v1.5-7B across different dataset including ScienceQA, Vizwiz, MM-Vet and LLaVA-Bench. Compared with SparseGPT, with an average pruning rate of 0.5, SlimLLaVA still achieves 57.51% accuracy on ScienceQA, which proves the versatility of our method, achieving excellent performance. Moreover, in complex multimodal tasks, SlimLLaVA demonstrates superior generalization capabilities

Table 4: The accuracy on VizWiz, MM-Vet and LLaVA-Bench for LLaVA-v1.5-7B, where we evaluate the models across a wide range of domains.

Method	Vizwiz				MM-Vet				LLaVA-Bench			
	0	0.3	0.4	0.5	0	0.3	0.4	0.5	0	0.3	0.4	0.5
LLM-Pruner		20.85	7.30	0.00		14.30	0.10	0.00		37.70	29.30	22.70
SparseGPT	50.05	46.43	41.90	38.61	35.40	21.20	20.30	17.90	55.90	57.90	52.50	48.80
Ours		52.70	52.19	50.28		33.80	33.00	31.70		58.40	54.50	53.00

and adaptability, including Vizwiz(38.61% vs 50.28%), MM-Vet(17.90% vs 31.70%) and LLaVA-Bench(48.80% vs 53.00%) which pruning ratio is 0.5, underscoring its significant potential not only in content generation but also across a wide range of applications that require robust, context-aware processing. For pruned model with low pruning ratio outperforming dense model, we believe that removing ambiguous neurons during the pruning process can lead to better performance on some simple question-answering datasets.

Finally, we perform an investigation into how effectively sparse LVLM can be accelerated in practice using standard tools for CPU. Due to the limitation of experiment equipment, we can not use NVIDIA's official CUTLASS library which is supported by NVIDIA GPUs of generation Ampere and latest theoretically offering $\times 2$ acceleration of matrix multiplications. Therefore, we investigate acceleration of unstructured sparsity for CPU inference based on DeepSparse. Figure 1 shows the detail of acceleration which can speed up $\times 1.47$ for LLaVA-v1.5-7B while the accuracy degradation is only 0.45%. Although LLM-Pruner inference without depending on particular kernels, it is unacceptable of the decrease of performance. Compared with SparseGPT, SlimLLaVA achieve higher performance with the similar speed up. The observed speedups are nearly at the theoretical optimum, which suggests that unstructured sparsity acceleration for LLM inference on CPUs is already a practical approach.

Visualization reasoning examples: We present several qualitative visual reasoning examples of the LLaVA-SQA-7B model using the ScienceQA dataset in Figure 4. Compared to LLM-Pruner, SlimLLaVA obtains more accurate and specific answers to vision-language question pairs, which reveals the higher model capacity even with the similar sparsity. Although LLM-Pruner can produce accurate results in the first case, it also provide irrelevant wrong information. For SparseGPT, the generated table provides incorrect answers. SlimLLaVA can infer correct results and additionally analyzing relevant knowledge. In the second example, LLM-pruner hallucinates that melted marshmallow exists in the image leading to incorrect result. Conversely, SparseGPT accurately identifies the glass but made a critical error by overlooking the chocolate milkshake. SlimLLaVA demonstrates the ability to comprehensively analyze the attributes of all items presented in the task. By leveraging its multimodal reasoning capabilities, it can accurately interpret both visual and textual information to identify the relevant features of each object.

5 CONCLUSION AND FUTURE WORK

In this paper, we present SlimLLaVA, an automatically pruning method for LVLMs with limited sampled data, to achieve low latency and unaffected task performance. We institute the generalization gap of the pruning policy between the calibration samples and the unknown training set for the pruning policy using the structural risk minimization principle. By estimating via the Euclidean distance of the candidate policy set, we enhance the upper bound of the generalization ability for all policies in the pruning space. The pruning policy candidates evolve with the goal of optimizing task performance and generalization by refining the vision projector. SlimLLaVA focuses on minimizing the generalization gap, allowing it to generalize across various scenarios after a single search, without the need for retraining or task-specific fine-tuning. Extensive experiments on question-answer datasets demonstrate that SlimLLaVA achieves competitive performance compared to state-of-the-art pruned LVLMs.

While SlimLLaVA has significantly improved generalization ability and maintained the performance of pruned LVLMs, there are still areas for future work. We aim to develop more efficient search algorithms to reduce the search cost and design a pruning policy that reduces the activation matrix to achieve higher compression ratios, further improving inference time.

REFERENCES

- 540
541
542 Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel
543 Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language
544 model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736,
545 2022.
- 546 Sotiris Anagnostidis, Dario Pavllo, Luca Biggio, Lorenzo Noci, Aurelien Lucchi, and Thomas
547 Hofmann. Dynamic context pruning for efficient and interpretable autoregressive transformers.
548 *Advances in Neural Information Processing Systems*, 36, 2024.
- 549 Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin
550 Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, et al. Vizwiz: nearly real-time answers
551 to visual questions. In *Proceedings of the 23rd annual ACM symposium on User interface software
552 and technology*, pp. 333–342, 2010.
- 553 Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina,
554 Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey
555 Litinsky, Varvara Logacheva, Alexey Lyamar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh,
556 Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. DeepPavlov: Open-
557 source library for dialogue systems. In Fei Liu and Thamar Solorio (eds.), *Proceedings of ACL
558 2018, System Demonstrations*, pp. 122–127, Melbourne, Australia, July 2018. Association for
559 Computational Linguistics. doi: 10.18653/v1/P18-4021. URL [https://aclanthology.
560 org/P18-4021](https://aclanthology.org/P18-4021).
- 561 Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu
562 Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, reproducible and strong vision language
563 assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023.
- 564 Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan
565 Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal
566 language model. *arXiv preprint arXiv:2303.03378*, 2023.
- 567 Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in
568 one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.
- 569 Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training
570 quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.
- 571 Ze-Feng Gao, Peiyu Liu, Wayne Xin Zhao, Zhong-Yi Lu, and Ji-Rong Wen. Parameter-efficient
572 mixture-of-experts architecture for pre-trained language models. *arXiv preprint arXiv:2203.01104*,
573 2022.
- 574 Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and
575 Steven Hoi. From images to textual prompts: Zero-shot visual question answering with frozen
576 large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and
577 Pattern Recognition (CVPR)*, pp. 10867–10877, June 2023.
- 578 Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model
579 compression and acceleration on mobile devices. In *Proceedings of the European Conference on
580 Computer Vision (ECCV)*, September 2018.
- 581 Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan
582 Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda
583 Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning
584 through planning with language models, 2022.
- 585 Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng
586 Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft
587 token pruning. In *European conference on computer vision*, pp. 620–640. Springer, 2022.
- 588 Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir
589 Gholami. A fast post-training pruning framework for transformers. *Advances in Neural Information
590 Processing Systems*, 35:24101–24116, 2022.
- 591
592
593

- 594 Ivan Lazarevich, Alexander Kozlov, and Nikita Malinin. Post-training deep neural network pruning
595 via layer-wise calibration. In *Proceedings of the IEEE/CVF international conference on computer*
596 *vision*, pp. 798–805, 2021.
- 597
- 598 Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. End-to-end task-
599 completion neural dialogue systems, 2018.
- 600
- 601 Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. Awq: Activation-
602 aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*,
603 2023.
- 604 Mingbao Lin, Rongrong Ji, Yuxin Zhang, Baochang Zhang, Yongjian Wu, and Yonghong Tian.
605 Channel pruning via automatic structure search. *arXiv preprint arXiv:2001.08565*, 2020.
- 606
- 607 Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction
608 tuning, 2024a.
- 609
- 610 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in*
611 *neural information processing systems*, 36, 2024b.
- 612
- 613 Ning Liu, Xiaolong Ma, Zhiyuan Xu, Yanzhi Wang, Jian Tang, and Jieping Ye. Autocompress: An
614 automatic dnn structured pruning framework for ultra-high compression rates. In *Proceedings of*
the AAAI Conference on Artificial Intelligence, volume 34, pp. 4876–4883, 2020.
- 615
- 616 Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava,
617 Ce Zhang, Yuandong Tian, Christopher Re, et al. Deja vu: Contextual sparsity for efficient llms
618 at inference time. In *International Conference on Machine Learning*, pp. 22137–22176. PMLR,
619 2023.
- 620
- 621 Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord,
622 Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for
623 science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521,
624 2022.
- 625
- 626 Jian-Hao Luo and Jianxin Wu. Autopruner: An end-to-end trainable filter pruning method for efficient
deep model inference. *Pattern Recognition*, 107:107461, 2020.
- 627
- 628 Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large
629 language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.
- 630
- 631 Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus
of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.
- 632
- 633 Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufer,
634 Dhruvi Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet
635 Singh, Doug Kang, Ankur Jain, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan
636 Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Guoli Yin, Mark Lee, Zirui
637 Wang, Ruoming Pang, Peter Gräsch, Alexander Toshev, and Yinfei Yang. Mm1: Methods, analysis
& insights from multimodal llm pre-training, 2024.
- 638
- 639 Jorge Mendez-Mendez, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Embodied lifelong learning
640 for task and motion planning. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings*
641 *of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning*
642 *Research*, pp. 2134–2150. PMLR, 06–09 Nov 2023. URL [https://proceedings.mlr.](https://proceedings.mlr.press/v229/mendez-mendez23a.html)
643 [press/v229/mendez-mendez23a.html](https://proceedings.mlr.press/v229/mendez-mendez23a.html).
- 644
- 645 Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural
646 networks. In *Conference on learning theory*, pp. 1376–1401. PMLR, 2015.
- 647
- Chunming Qiao and Myungsik Yoo. Optical burst switching (obs)—a new paradigm for an optical
internet¹. *Journal of high speed networks*, 8(1):69–84, 1999.

- 648 Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit:
649 Efficient vision transformers with dynamic token sparsification. *Advances in neural information*
650 *processing systems*, 34:13937–13949, 2021.
- 651 Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. In-context
652 impersonation reveals large language models’ strengths and biases. *Advances in Neural Information*
653 *Processing Systems*, 36, 2024.
- 654 Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning.
655 *Advances in neural information processing systems*, 33:20378–20389, 2020.
- 656 Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. Prompting large language models with answer
657 heuristics for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF*
658 *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14974–14983, June 2023.
- 659 Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for
660 large language models. *arXiv preprint arXiv:2306.11695*, 2023.
- 661 Yehui Tang, Yunhe Wang, Yixing Xu, Dacheng Tao, Chunjing Xu, Chao Xu, and Chang Xu. Scop:
662 Scientific control for reliable neural network pruning. *Advances in Neural Information Processing*
663 *Systems*, 33:10936–10947, 2020.
- 664 Weihang Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang,
665 Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang.
666 Cogvlm: Visual expert for pretrained language models, 2024.
- 667 Xia Xiao, Zigeng Wang, and Sanguthevar Rajasekaran. Autoprune: Automatic network pruning by
668 regularizing auxiliary parameters. *Advances in neural information processing systems*, 32, 2019.
- 669 Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu,
670 Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning
671 and action. *arXiv preprint arXiv:2303.11381*, 2023.
- 672 Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang,
673 and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv*
674 *preprint arXiv:2308.02490*, 2023.
- 675 Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava:
676 A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024.
- 677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701