UNIVERSITY OF CALIFORNIA SAN DIEGO

Towards Enhanced Language Model Reasoning and Efficient Knowledge Transfer

A Thesis submitted in partial satisfaction of the requirements
for the degree Master of Science

in

Computer Science

by

Yunhao Fang

Committee in charge:

Professor Hao Su, Chair
Professor Jingbo Shang
Professor Xiaolong Wang

2024

The Thesis of Yunhao Fang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Hao Su for their support as the chair of my committee, as well as so many brilliant minds in SULab. This one and a half year is a happy journey, and I learned a lot in the friendly and competitive environment.

ABSTRACT OF THE THESIS

Towards Efficient Knowledge Transfer and Enhanced Reasoning for Foundation Models

by

Yunhao Fang

Master of Science in Computer Science and Engineering

University of California San Diego, 2024

Professor Hao Su, Chair

ABSTRACTION

Large language models (LLMs) and vision language models (VLMs) are changing the world and gradually presenting human-level intelligence in various real-world scenarios, including knowledge-based question answering, mathematics, and programming. During the master period, my research focuses on understanding and improving current large language models' reasoning capacity towards general problems solving, and efficient methods to enable the knowledge transfer for vision-language models: distill knowledge from large vision-language models.

INTRODUCTION

The central problem of this thesis is how we can empower intelligent computer agents complex reasoning capacity and develop efficient approaches to edit their generalized features spaces with limited computation and data. Though current language models have shown strong potential in reasoning, mathematical examinations and code generation tasks, they are still uncapable to solve tasks requires complex reasoning, for example mathematical proof and designing creative methods for open-ended problems. As an overview of the thesis, we take an artificial intelligence perspective to boost language models' reasoning ability with deductive verification, a logistic method which helps reasoners recover from knowledge hallucinations and unstrict reasoning process and pick out the final answers[1]. At the same time, we are among the pioneers who argue the importance of language models' unbiased exploration for reasoning, even though they are trained with biased data distribution[2].

Alongside pursuing the limitation of large language models' reasoning capacity, we take another step towards designing efficient algorithms for knowledge transfer through low-cost adaptation. This includes distilling vision-language model's generalizable feature space into smaller models for the purpose of facilitating downstream tasks[3], and instilling human common sense like 3D spatial awareness which is seldom annotated into large vision-language models.

Artificial general intelligence is no longer an unbelievable word and deserve more attention for everyone. I hope my thoughts and effort in pushing the edge of understanding and improving foundation models during my master period will pave the road of my further research career, and finally contribute to this pure will towards the super-human machine intelligence.

Chapter 1 Deductive Verification of Chain-of-Thought Reasoning

Large Language Models (LLMs) significantly benefit from Chain-of-Thought (CoT) prompting in performing various reasoning tasks. While CoT allows models to produce more comprehensive reasoning processes, its emphasis on intermediate reasoning steps can inadvertently introduce hallucinations and accumulated errors, thereby limiting models' ability to solve complex reasoning tasks. Inspired by how humans engage in careful and meticulous deductive logical reasoning processes to solve tasks, we seek to enable language models to perform explicit and rigorous deductive reasoning, and also ensure the trustworthiness of their reasoning process through self-verification. However, directly verifying the validity of an entire deductive reasoning process is challenging, even with advanced models like ChatGPT. In light of this, we propose to decompose a reasoning verification process into a series of step-by-step subprocesses, each only receiving their necessary context and premises. To facilitate this procedure, we propose Natural Program, a natural language-based deductive reasoning format. Our approach enables models to generate precise reasoning steps where subsequent steps are more rigorously grounded on prior steps. It also empowers language models to carry out reasoning self-verification in a step-by-step manner. By integrating this verification process into each deductive reasoning stage, we significantly enhance the rigor and trustfulness of generated reasoning steps. Along this process, we also improve the answer correctness on complex reasoning tasks.

The transformative power of large language models, enhanced by Chain-of-Thought (CoT) prompting[4][5][6]**Error! Reference source not found.**, has significantly reshaped the landscape of information processing[7][9][10][11][12], fostering enhanced abilities across a myriad of disciplines and sectors. While CoT allows models to produce more comprehensive

reasoning processes, its emphasis on intermediate reasoning steps can inadvertently introduce hallucinations[13][14][15][16] and accumulated errors[17][18], thereby limiting models' ability to produce cogent reasoning processes.

In fact, the pursuit of reliable reasoning is not a contemporary novelty; indeed, it is an intellectual endeavor that traces its roots back to the time of Aristotle's ancient Greece. Motivated by the desire to establish a rigorous reasoning process, in his "Organon," Aristotle introduced principles of logic, in particular, syllogism, a form of logical argument that applies deductive reasoning to arrive at a conclusion based on two or more propositions assumed to be true. In disciplines that rigorous reasoning is critical, such as judicial reasoning and mathematical problem solving, documents must be written in a formal language with a logical structure to ensure the validity of the reasoning process.

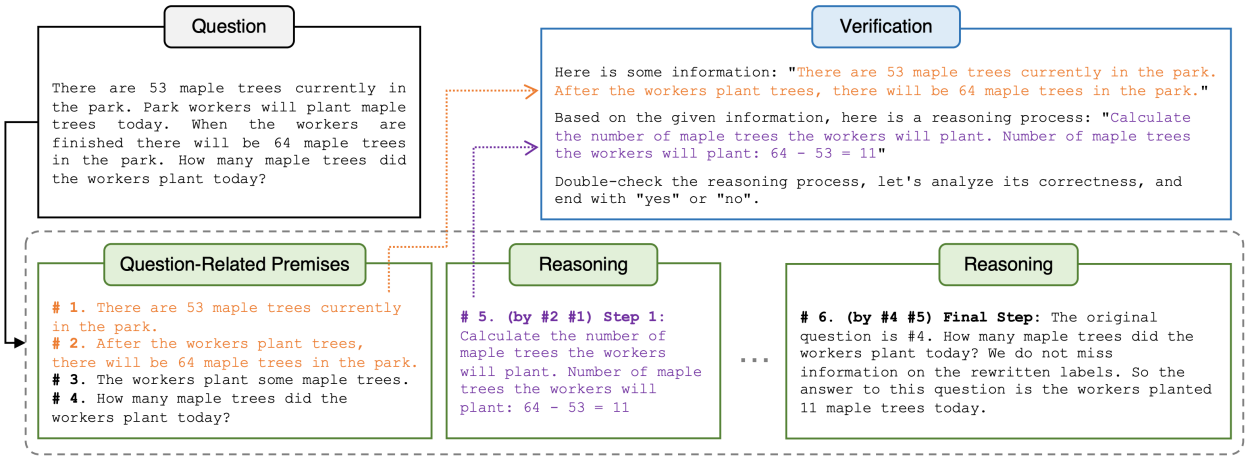

Figure 1: An overview of our proposed deductive reasoning and verification process. In response to an input question, LLMs generate deductive reasoning chains using the Natural Program format (bottom 3 boxes), a natural language-based deductive reasoning approach.

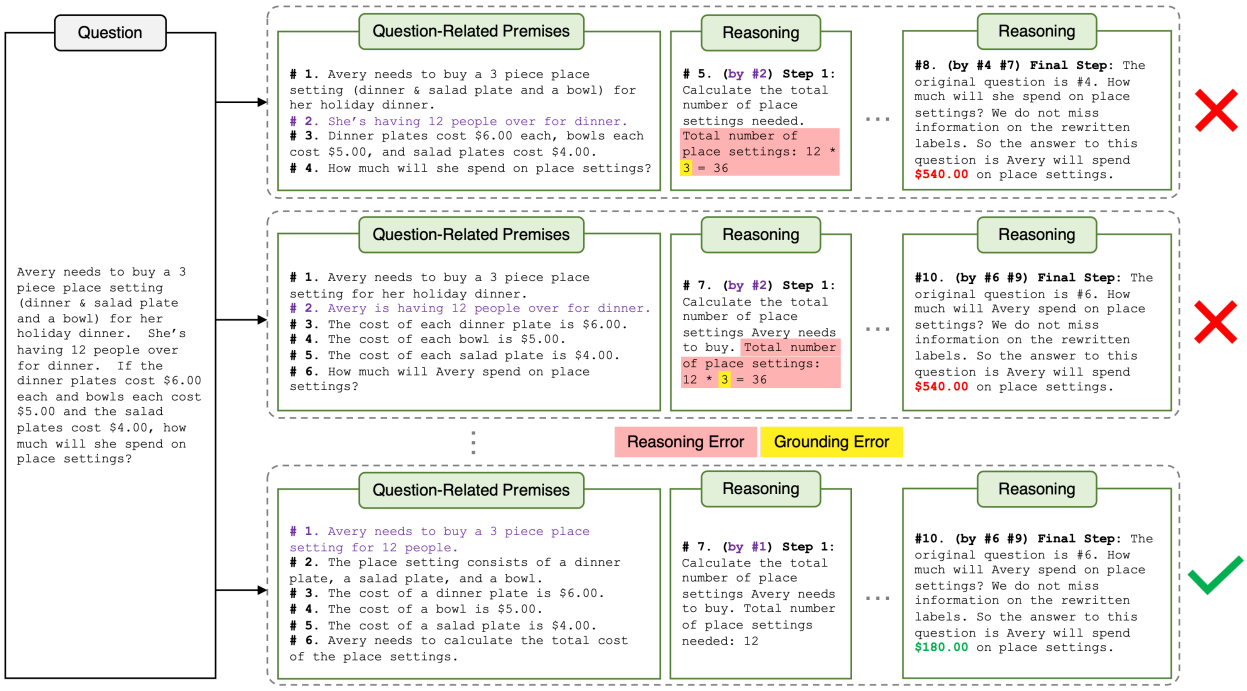

Figure 2: Through our Natural Program-based deductive reasoning verification approach, we identify and eliminate reasoning chains that contain errors in reasoning and grounding (we define grounding error as utilizing information that is not present in cited premises).

We yearn for this sequence of reliable knowledge when answering questions. Our goal is to develop language models that can propose potential solutions through reasoning in logical structures. Simultaneously, we aim to establish a verifier capable of accurately assessing the validity of these reasoning processes. Despite recent significant explorations in the field, such as [19]'s emphasis on self-consistency and [20][21]'s innovative use of codes to represent the reasoning process, these approaches still exhibit considerable limitations. For example, consistency and reliability are not inherently correlated; as for program codes, they are not powerful enough to represent many kinds of reasoning process, e.g., in the presence of quantifiers ("for all", "if there exists") or nuances of natural language (moral reasoning, "likely", . . .).

We propose leveraging the power of natural language to achieve the deductive reasoning emphasized in ancient Greek logic, introducing a “natural program”. This involves retaining natural language for its inherent power and avoiding the need for extensive retraining with large data sets. A natural program represents a rigorous reasoning sequence, akin to a computer program. We expect implementations of the idea to have two properties: 1) that natural programs are generated with minimal effort from an existing language model capable of CoT reasoning, preferably through in-context learning; 2) that the natural program can be easily verified for reliability in the reasoning process.

Through a step-by-step investigation, we discovered that large language models have the potential to meet our expectation. Naïve CoT prompts like "Let us think step by step." has many flaws, and entrusting the entire verification process to a large model like ChatGPT can still lead to significant error rates. However, we found that, if the reasoning process is very short, and only based on necessary premises and contexts, the verification of existing large language models is already quite reliable. Therefore, our approach is to design prompts that induce CoT processes comprised of rigorous premises/conditions and conclusions with statement labels, and verification can be done by gradually isolating very few statements within the long thought chain. Experimentally, we found that most reasoning that passed the verification was rigorous, and many that did not pass had elements of imprecision in the reasoning process, even if they occasionally arrived at correct answers. It is worth emphasizing that, we are not looking for a method to just maximize the correctness rate of final answers; instead, we aspire to generate a cogent reasoning process, which is more aligned with the spirit of judicial reasoning. When combined with sampling-based methods, our method can identify low-probability but rigorous reasoning processes. When repeated sampling fails to yield a rigorous reasoning process, we can

output "unknown" to prevent hallucinations that mislead users. We demonstrate the efficacy of our natural program-based verification approach across a range of arithmetic and common sense datasets on publicly-available models like OpenAI's GPT-3.5-turbo. Our key contributions are as follows:

1. We propose a novel framework for rigorous deductive reasoning by introducing a "Natural Program" format, which is suitable for verification and can be generated by just in-context learning.

2. We show that reliable self-verification of long deductive reasoning processes written in our Natural Program format can be achieved through step-by-step subprocesses that only cover necessary context and premises.

3. Experimentally, we demonstrate the superiority of our framework in improving the rigor, trustworthiness, and interpretability of LLM-generated reasoning steps and answers.

Given a reasoning chain $S = (s_1, s_2, \dots, s_n)$, a straightforward idea to verify its deductive validity is to ask LLMs to examine the entire reasoning chain at once. To assess the effectiveness of this approach, we conduct a preliminary experiment: for a dataset problem and its reasoning chain S generatedby ChatGPT, we prompt ChatGPT with "Do you think the above reasoning process is correct? Let's think step by step" such that its outputs whether there exists any mistake among any reasoning step in S. However, as demonstrated in Table 1, the verification accuracy is 50% for most datasets, and ChatGPT struggles at finding out mistaken reasonings. Notably, it persistently outputs "Correct" for most reasoning chain queries, regardless of their actual validity. We conjecture that such phenomenon is caused by the abundance of irrelevant premises for each reasoning step. Recall that the premises pi for a reasoning step s_i consist of question Q, the question context C, along with the prior reasoning steps $s_{\leq j} = \{s_j : j < i\}$. For Q

and C, we can further extract and decompose $Q \cup C$ into a set of "question-related premises" $QC = \{qc_1, qc_2, \ldots, qc_m\}$, where qc_i is a premise or condition inferred from $Q \cup C$. Then, it is often the case that most elements of $p_i = QC \cup s_{\leq j}$ are irrelevant to the validity of s_i, leading to erroneous verifications from language models. A very recent work [41] also observes a similar phenomenon where LLMs are easily distracted by irrelevant context.

Hence, we propose a decomposition of the reasoning chain verification process into a series of step-by-step processes, where each step only considers the premises that are necessary. The overall validity of the reasoning chain, denoted as $V(s) = \wedge_{i=1}^{M} V(s_i)$, can be naturally decomposed into individual step validity $V(s_i)$. However, achieving such decomposition is highly challenging without imposing constraints on the format of reasoning chains. Additionally, for each $s_i \in S$, we aim to ensure that it explicitly lists the minimal subset of premises $\overline{p_i} \subseteq p_i$ required for deductive reasoning to avoid potential ambiguities during verification. This motivates us to introduce a natural-language-based deductive reasoning format.

As previously mentioned, we desire LLMs to output deductive reasoning processes that can be easily verified by themselves, specifically by listing out the minimal set of necessary premises p_i at each reasoning step s_i. To accomplish its goal, we propose to leverage the power of natural language, which is capable of rigorously representing a large variety of reasoning processes and can be generated with minimal effort. In particular, we introduce Natural Program, a novel deductive reasoning format for LLMs. More formally, Natural Program consists of the following components:

• An instruction for models to extract question-related premises QC. We use the following instruction: "First, let's write down all the statements and relationships in the question with labels".

• A numbered-list of question-related premises, each prefixed with "#{premise_number}".

• An instruction for models to generate the reasoning chain S based on the question-related premises QC. We use the following instruction: "Next, let's answer the question step by step with reference to the question and reasoning process".

• A list of prefixed reasoning steps S_i . The prefix has the following format: #{number} (by {list_of_premises_used}). Here "number" equals $|QC| + i$, and "list_of_premises_used" consists of numbers from the smallest subset of premises among $QC \cup s_{\leq j}$ that are used for the deductive reasoning of s_i. In addition, for the last reasoning step sm, we ensure that it (1) includes a special tag Final Step; (2) refers to the premise number of the target question to be answered; (3) explicitly gives the final answer to a question.

Given that LLM's reasoning outputs follow the Natural Program format, we can then verify the deductive validity of a single reasoning step s_i through an instruction that consists of (1) the full descriptions of premises used for the reasoning of s_i; (2) the full description of s_i; (3) an instruction for validity verification, such as "Double-check the reasoning process, let's analyze its correctness, and end with "yes" or "no"." Note that throughout this verification process, we only retain the minimal necessary premise and context for s_i, thereby avoiding irrelevant context distraction and significantly improving the effectiveness of validation.

Given that we can effectively verify a deductive reasoning process, we can naturally integrate verification with LLM's sequence generation strategies to enhance the trustworthiness of both the intermediate reasoning steps and the final answers. In this work, we propose Unanimity-Plurality Voting, a 2-phase sequence generation strategy described as follows. Firstly, similar to prior work like [19], we sample k reasoning chain candidates along with their final

answers. In the unanimity phase, we perform deductive validation on each reasoning chain. Recall that a chain S is valid (i.e., $V\ (S) = \ 1$) if and only if all of its intermediate reasoning steps are valid (i.e., $\forall i, V(s_i) = \ 1$). For each intermediate reasoning step s_i, we perform majority voting over k' sampled single-step validity predictions to determine its final validity $V(s_i)$. We then only retain the verified chain candidates $\{S : V(S) = \ 1\}$. In the plurality voting stage, we conduct a majority-based voting among the verified chain candidates to determine the final answer. This voting process ensures that the final answer is selected based on a consensus among the trustworthy reasoning chains.

Table 1: Zero-shot and two-shot reasoning chain verification accuracy for GPT-3.5-turbo (ChatGPT), where an entire reasoning chain is verified at once.

Prompting	Reasoning Correctness	GSM8K	AQuA	MATH	AddSub	Date	Last Letters
Zero-shot	Correct	0.98	0.96	1.00	0.98	0.98	1.00
	Incorrect	0.04	0.06	0.04	0.02	0.04	0.04
	(Average)	0.51	0.51	0.52	0.50	0.51	0.52
Two-shot	Correct	0.98	0.96	1.00	0.92	1.00	0.96
	Incorrect	0.02	0.04	0.00	0.06	0.26	0.06
	(Average)	0.50	0.50	0.50	0.49	0.63	0.51

Benchmarks. We evaluate the deductive verification accuracy and the answer correctness of reasoning chains over a diverse set of reasoning tasks: arithmetic reasoning, symbol manipulation, and date understanding. For arithmetic reasoning, we utilize the following benchmarks: 1) AddSub [22]; 2) GSM8K [23]; 3) MATH [24]; 4) AQuA [25]. Among these benchmarks, the AddSub and GSM8K datasets involve middle school-level multi-step calculations to arrive at a single number as the final answer. The MATH dataset presents more

challenging problems that require expressing the answer as a mathematical expression in LaTeX format. These problems involve concepts from linear algebra, algebra, geometry, calculus, statistics, and number theory. AQuA also features similarly challenging problems, except that questions are in a multiple-choice format. For symbol manipulation, we use Last Letter Concatenation, where the model is tasked with concatenate the last letters of all the words provided in the question.

Deductive verification evaluation setup. For each of the above benchmarks, we select 100 reasoning chains, where 50 of them are deductively valid and 50 of them exhibit reasoning mistakes. The ground-truth deductive validity of each reasoning chain is determined by human annotators.

Answer extraction. To extract answers from reasoning solutions, we first perform text splitting based on answer prefix patterns such as “answer is” or “option is”. Then, using problem type-specific regular expressions, we extract the final answer. To extract the validity results from deductive verification processes, we only keep the last sentence of model response. We then extract the validity answer with regular expressions to obtain attitude words, e.g., “yes” or “no”, to determine the validity answer. Sometimes, language models may not provide a direct answer and instead output phrases like “not applicable” at the end of the response. In such cases, we consider the answer from the model as "yes".

Model and Hyperparameters. We conduct our main experiments with GPT-3.5-turbo (ChatGPT) [27]. For ChatGPT, we use a generation temperature of $T = 0.7$. For Unanimity-Plurality Voting, we set $k = 10$ and $k' = 3$ by default. We use 1-shot prompting for both reasoning chain generation and deductive verification (except reasoning chain generation for the date understanding task where we use 2-shot).

Table 2: Comparison of deductive verification accuracy of reasoning chains for GPT-3.5-turbo (ChatGPT). We compare two approaches: (1) verifying entire reasoning chains generated by Chain-of-Thought prompting; (2) verifying reasoning chains generated in the Natural Program format with step-by-step decomposition.

Verification Method	Reasoning Correctness	GSM8K	AQuA	MATH	AddSub	Date	Last Letters	Overall
CoT Two-shot	Correct	98%	96%	100%	92%	100%	96%	97%
	Incorrect	2%	4%	0%	6%	26%	6%	7%
	(Average)	50%	50%	50%	49%	63%	51%	52%
Natural Program One-shot	Correct	84%	72%	70%	95%	90%	96%	85%
	Incorrect	84%	62%	76%	40%	56%	6%	54%
	(Average)	84%	67%	73%	68%	73%	51%	69%

We compare the verification accuracy of reasoning chains using two methods: (1) verifying the entire reasoning chain at once without utilizing the Natural Program, and (2) our Natural Program-based verification approach with step-by-step decomposition. The results, presented in Table 2, indicate that our approach achieves significantly higher reasoning verification accuracy across most datasets. It effectively identifies erroneous reasoning in faulty chains while maintaining a low rate of false positives for valid chains. However, we observe that our approach's effectiveness is limited on the "Last Letters" task. We hypothesize that this is due to the task's nature, where each subsequent reasoning step is conditioned on all previous steps, presenting greater challenges for reasoning verification due to the increased dependency among premises.

In this paper, we aim to enable Large Language Models (LLMs) to perform explicit and rigorous deductive reasoning while ensuring the trustworthiness of their reasoning processes through self-verification. To this end, we have proposed a novel framework based on "Natural

Program", a natural language-based deductive reasoning format that facilitates reasoning verification and can be easily generated through in-context learning. Within this framework, we decompose the verification process of complex reasoning chains into step-by-step subprocesses that focus solely on necessary context and premises, allowing us to significantly enhance the accuracy of verification. Additionally, we introduce an Unanimity-Plurality Voting strategy to further improve verification accuracy. Experimentally, we demonstrate the superiority of our framework in improving the rigor, trustworthiness, and interpretability of reasoning steps and answers.

Acknowledgement

Chapter 1, in full, is a modified reprint of the material as it appears in the conference of Neural Information Processing 2023. Zhan Ling and Yunhao Fang and Xuanlin Li and Zhiao Huang and Mingu Lee and Roland Memisevic and Hao Su. The dissertation author was the primary investigator and author of this paper.

Chapter 2 Distilling Large Vision-Language Model with Out-of-distribution Generalizability

Large vision-language models have achieved outstanding performance, but their size and computational requirements make their deployment on resource-constrained devices and time-sensitive tasks impractical. Model distillation, the process of creating smaller, faster models that maintain the performance of larger models, is a promising direction towards the solution. This paper investigates the distillation of visual representations in large teacher vision-language models into lightweight student models using a small- or mid-scale dataset. Notably, this study focuses on open-vocabulary out-of-distribution (OOD) generalization, a challenging problem that has been overlooked in previous model distillation literature. We propose two principles from vision and language modality perspectives to enhance student's OOD generalization: (1) by better imitating teacher's visual representation space, and carefully promoting better coherence in vision-language alignment with the teacher; (2) by enriching the teacher's language representations with informative and fine-grained semantic attributes to effectively distinguish between different labels.

We distill a large vision-language teacher model T (e.g., CLIP ViT-L/14 [28]) to a small student image model S (e.g., ResNet18 [29]) by focusing on out-of-distribution (OOD) generalization for open-vocabulary object classification. We choose small- or mid-scale datasets to achieve the distillation so that the distillation process is flexible for fast research cycle and has less resource dependency. The teacher consists of an image encoder $T_{img}(\cdot)$ and a text encoder $T_{txt}(\cdot)$. During distillation, we keep the flexibility of the existing teacher text encoder for the open-set setting, and we let the student model S be vision-only, i.e., $S = S_{img}$. Through this process, we hope that S not only achieves high prediction accuracy on seen labels Y_{id}, but also attains strong generalization ability on out-of-distribution labels Y_{ood}. In addition, we train

students from scratch to avoid label contamination, allowing us to more carefully assess and understand their OOD generalization ability.

Table 3: Comparison between student models trained without teacher-student visual representation space alignment (L_{cls} only), with direct teacher visual feature fitting $+L_{mse}$), with improved teacher-student visual space alignment ($+L_{im-cst}$), and with improved preservation of teacher's vision-language alignment structure $+L_{vlprox}$).

	CaltechBirds	StanfordCars	Flower102	Food101	SUN397	tiered-ImageNet
CLIP ViT-L/14	70.0 / 70.5	79.3 / 78.3	74.4 / 84.1	90.5 / 91.2	72.8 / 74.4	71.1 / 76.3
CLIP RN50	57.1 / 56.4	53.2 / 56.3	59.3 / 65.3	76.5 / 78.3	65.0 / 66.3	55.7 / 62.0
Closed-Set Classification	48.1 / NA / 18.1	27.9 / NA / 10.1	77.1 / NA / 45.0	71.7 / NA / 30.3	57.8 / NA / 31.1	63.4 / NA / 31.2
L_{cls}	61.0 / 14.2 / 34.9	56.3 / 14.7 / 20.0	81.2 / 4.5 / 46.2	72.2 / 16.1 / 24.5	57.5 / 13.6 / 28.6	64.4 / 13.9 / 27.5
L_{mse}	27.0 / 12.0 / 14.3	5.5 / 3.8 / 4.0	48.1 / 7.3 / 15.0	45.0 / 17.0 / 19.3	24.3 / 11.0 / 14.5	49.3 / 14.8 / 23.2
$L_{cls} + L_{mse}$	63.7 / 17.4 / 36.2	62.2 / 18.8 / 35.1	82.6 / 6.3 / 46.0	72.3 / 19.0 / 35.5	57.1 / 15.3 / 29.4	66.2 / 14.9 / 28.5
L_{im-cst}	42.1 / 21.3 / 29.1	33.2 / 13.7 / 20.0	54.8 / 13.3 / 27.3	70.0 / 34.9 / 36.8	45.2 / 22.8 / 27.2	46.3 / 22.8 / 30.8
$L_{cls} + L_{im-cst}$	60.9 / 20.4 / 37.6	59.6 / 18.3 / 31.2	82.4 / 12.7 / 52.5	74.0 / 30.5 / 42.0	62.5 / 18.8 / 35.2	64.4 / 18.0 / 33.5
$L_{cls} + L_{im-cst} + L_{mse}$	62.5 / 20.8 / 39.0	59.6 / 19.0 / 33.1	82.6 / 12.0 / 48.7	75.0 / 31.2 / 42.0	60.0 / 19.8 / 35.2	67.0 / 19.4 / 34.6
$L_{cls} + L_{im-cst} + L_{vlprox}$	62.3 / 21.6 / 39.0	63.9 / 19.8 / 38.5	82.7 / 14.6 / 52.0	74.3 / 32.0 / 43.2	61.7 / 21.5 / 34.7	67.5 / 20.5 / 35.3
$L_{im-cst} + L_{vlprox}$	45.3 / 21.9 / 30.4	46.5 / 17.8 / 26.9	66.9 / 13.5 / 35.4	71.4 / 35.2 / 40.0	52.0 / 23.1 / 28.8	57.5 / 23.0 / 33.2

We adopt a diverse collection of recognition tasks using small to medium-scale datasets, including CaltechBirds [30], StanfordCars [31], Flower102 [32], Food101 [33], SUN397 [34],

and tiered-ImageNet [35]**Error! Reference source not found.**. We split the dataset labels such that $|Y_{id}| = |Y_{ood}|$, except tiered-ImageNet, which comes with an existing split.

We note that teacher's visual representation space is well-aligned with language across diverse datasets, and such alignment demonstrates strong generalization across many domains. By imitating teacher's visual space structure, we hope to enhance the ability for student's visual space to generalize and extrapolate towards unseen concepts, thereby implicitly enhancing the generalizability of student's vision-language alignment and improving its OOD generalization.

A direct approach to achieve this is to align the teacher and student visual representations through the Mean Squared Error (MSE) loss:

$$L_{mse} = \left|\left|S(x) - T_{img}(x)\right|\right|_2^2$$

In Table 3, we show that adding L_{mse} on top of L_{cls} improves student OOD generalization. However, upon further examination, we find that students face significant challenges in precisely reproducing teacher's visual representations. Such errors persist even when the student and teacher networks possess the same representation power (e.g., both ResNet50 networks). This phenomenon highlights that achieving precise matching between teacher and student's high-dimensional visual feature spaces is inherently challenging, which can be attributed to differences in weight initialization, training data, and the presence of local minima in the loss landscape. Moreover, we later find that when students struggle to precisely match teacher's visual features, they also struggle to preserve teacher's local visual space structure and relative visual feature relationship between different images, hindering their OOD generalization ability.

Since precisely matching teacher's visual features is inherently challenging, we propose to augment the training objective with the following contrastive loss, which "softly" matches teacher's visual features:

$$L_{im-cst}(x) = \frac{\exp\left(-\left|\left|S(x) - T_{img}(x)\right|\right|_2^2 / \tau\right)}{\sum_{x'} \exp\left(-\left|\left|S(x) - T_{img}(x')\right|\right|_2^2 / \tau\right)}$$

By combining L_{im-cst} with L_{cls}, we observe in Table 3 that the student exhibits significantly better zero-shot and few-shot OOD generalization ability across different datasets. In the following paragraphs, we will develop several metrics to better assess the teacher-student visual space consistency. These metrics provide us with valuable insights into how L_{im-cst} facilitates students to achieve closer visual space proximity to the teacher while yielding a deeper understanding of the teacher's visual representation space.

Previously, we focused on improving the student's OOD generalization ability by better aligning student-teacher visual spaces. Since teacher's visual space is well-aligned with language across diverse concepts and domains, a better student coherence with teacher's visual space implicitly leads to better vision-language (V-L) alignments. Naturally, an alternative perspective to improve student's OOD generalization becomes to enhance its explicit V-L alignments and improve their coherence with the teacher's, where we previously only used a simple contrastive V-L matching loss L_{cls}. Another motivation to focus on explicit V-L alignments arises from our finding that they play an essential role to ensure precise and accurate V-L alignments, especially when training on seen concepts or performing few-shot learning on novel concepts. Relying solely on implicit V-L alignments is inadequate in these scenarios. This is evident in Table 3,where solely utilizing the visual space alignment loss L_{im-cst} yields better performance on 0-shot X_{ood} (where classes are unseen) but worse performance on X_{id} and 5-shot X_{ood} (where

classes are seen). On the other hand, by combining both implicit and explicit V-L alignment losses ($L_{im-cst} + L_{cls}$), students excel in all of X_{id}, 0-shot X_{ood}, and 5-shot X_{ood} scenarios. Therefore, by improving explicit V-L alignments, we not only hope to further enhance student's 0-shot OOD generalization ability, but also improve their performance on familiar concepts and their ability to few-shot adapt to novel concepts.

We note that while L_{cls} performs explicit V-L alignments, it has the limitation of indiscriminately pushing an image away from all non ground-truth language features, therefore disregarding teacher's relative alignment relationship between the same image and different language features. Furthermore, we find that even though preserving teacher's relative V-L alignment structure is desirable, it may not always be perfect due to potential misalignments between teacher image features and their corresponding language labels. These misalignments can introduce inconsistent noise during distillation, ultimately harming student performance.

Motivated by these observations, we propose to augment our training objective with L_{vlprox}, which effectively and carefully preserves the teacher's vision-language alignment structure while accounting for potential misalignments:

$$L_{vlprox}(x,k) = I(x) \cdot D_{KL}(P_{T,topk}(\cdot\,|x)||P_{S,topk}(\cdot\,|x))$$

$$I(x) = 1[argmax_y P_T(y|x) = label(x)]$$

$$P_{\cdot,topk}(y|x) = \frac{1_{y\in Y_{topk}} P_\cdot(y|x)}{\sum_{y\in Y_{topk}} P_\cdot(y|x)}; Y_{topk} = argtopk_y P_T(y|x)$$

Here P_T and P_S denote teacher and student label probabilities; $I(\cdot)$ filters out images misaligned with language labels; and k controls the number of most-similar language features for each image. In our implementations, we find a larger k beneficial for OOD generalization, and we choose $k = 256$.

We demonstrate the effectiveness of L_{vlprox} in Table 3.We find that by combining L_{vlprox} with L_{cls} and L_{im-cst}, we further improve student's ability to generalize towards OOD concepts. Interestingly, we also observe that while L_{cls} and L_{vlprox} both explicitly perform V-L alignments, adding them together yields significantly better student performance on X_{id} and 5-shot X_{ood} than solely keeping L_{vlprox}. This observation is distinct from those in the traditional model distillation literature [37], where distilling teacher logits alone from vision-only models typically produces good student performance.

In the previous part, we focused on improving the imitation of teacher's visual space and promoting better coherence with teacher's vision-language alignment. Throughout this process, we kept the language representations fixed. However, the quality of language representations also plays a pivotal role in student learning and inference. Ideally, language representations should be capture precise, fine-grained, and meaningful semantic attributes, such that the student can effectively distinguish between different labels. We therefore ask the following question: can we leverage better and richer teacher language representations to further enhance student's OOD generalization ability? We propose the following candidate strategies:

Enriching semantic details of label descriptions by prompting LLMs. Previously, when we generate language representations $l(y)$ = prompt + description(y) for student training, we adopted a simple strategy. In particular, for the description of a label y, we merely used its label name, e.g., "lotus". However, these simplistic descriptions overlook many fine-grained properties of semantic categories, such as the shape, color, and texture of flowers, along with the description of their petals, leaves, and stems. In addition, we hope to automatically and efficiently generate enriched language descriptions for a wide range of labels, ensuring scalability for an arbitrary number of labels. Motivated by the recent progress on instruction-

finetuned large language models (LLMs) [38][39][40], which have demonstrated impressive sequence generation abilities given user prompts, we find these models well-suited for our goal. Therefore, we propose to use ChatGPT to generate category descriptions. We prompt ChatGPT with the following instruction: "Use a single sentence to describe the appearance and shape of {cls}. Only describe the shape and appearance". This allows ChatGPT to generate informative, fine-grained, and meaningful descriptions for target classes (e.g., "large, round, flat leaves; tall, slender stems; delicate petals in shades of pink, white, or yellow"), while keeping sequence lengths within CLIP text encoder's limit. We then set description(y) by concatenating "a photo of {cls}" with ChatGPT-generated class descriptions. We still keep the same vision-language alignment losses (L_{cls} and L_{vlprox}) as before.

Augmenting text through auxiliary captions. Currently, during student training, there is only one language description per category, i.e., $|l(y) : (x, y) \in X_{train}| = |Y_{id}|$. On the other hand, the number of training images significantly exceeds the number of labels, i.e., $|X_{train}| \gg |Y_{id}|$. We therefore wish to generate language descriptions for each individual image, such that we can substantially enrich the number of language features during student training, which potentially benefits student performance. To achieve this, we propose using OFA [41] to generate captions for each image, resulting in a new dataset $\{(x, cap(x), y) : (x, y) \in X_{train}\}$ augmented with captions. During student training, besides using the same vision-language alignment losses L_{cls} and L_{vlprox} as before, we also adopt the following auxiliary loss:

$$L_{cap}(x) = \frac{\exp\left(\cos\left(S(x), T_{txt}(cap(x))/\tau\right)\right.}{\sum_{x',y':y' \neq y} \exp\left(\cos\left(S(x), T_{txt}(cap(x'))/\tau\right)\right.}$$

The loss pushes x and its corresponding caption cap(x) together while pulling away from captions belonging to different categories. Our preliminary experiments show that distinguishing captions belonging to the same category could degrade student performance as they are usually

similar. Note that we only incorporate captions for the auxiliary loss during student training. For student inference and label predictions, we continue to use the same $l(y)$ as before.

Table 4: Comparison between different language representation enrichment strategies. The three numbers $x1/x2/x3$ in each entry denote the evaluation performance on X_{id}, zero-shot performance on X_{ood}, and 5-shot performance on X_{ood}, respectively.

	CaltechBirds	StanfordCars	Flower102	Food101	SUN397	Tiered-ImageNet
Tab. 1 best	62.3 / 21.6 / 39.0	63.9 / 19.8 / 38.5	82.7 / 14.6 / 52.0	74.3 / 32.0 / 43.2	61.7 / 20.5 / 34.7	67.5 / 20.5 / 35.3
Semantic Details	62.0 / 23.2 / 40.4	63.6 / 20.0 / 37.5	82.4 / 17.6 / 52.7	74.8 / 33.9 / 43.7	60.8 / 23.3 / 36.8	69.8 / 23.5 / 36.2
Auxiliary Captions	62.5 / 21.4 / 41.0	65.5 / 19.0 / 38.1	81.9 / 14.3 / 52.5	75.4 / 33.3 / 44.0	61.6 / 22.1 / 36.9	68.7 / 21.0 / 34.4
Semantics + Caption	62.0 / 22.7 / 39.8	64.9 / 20.4 / 39.7	83.7 / 18.2 / 53.4	75.6 / 35.7 / 42.9	61.0 / 24.0 / 37.5	68.9 / 23.6 / 35.8

We adopt the aforementioned language-enriching strategies for student learning, and we present the results in Table 4. We find that combining LLM-enriched label descriptions with auxiliary captions yields the best OOD generalization. However, upon analyzing their individual effectiveness, we find that LLM-enriched label descriptions provide significantly better zero-shot OOD benefit than auxiliary captions, and solely relying on auxiliary captions only marginally improves zero-shot OOD generalizability. Upon further analysis, we find that many generated captions only broadly describe objects and are much less informative than ChatGPT generated descriptions for distinguishing fine-grained categories. For instance, in the StanfordCars dataset, a generated caption for an "Acura Integra Type R 2001" image is "a white car is parked in a field", and solely relying on the white color provides little information to distinguish different car categories. Consequently, captions have limited impact on enhancing the generalizability of student's vision-language alignment structures. Given the significant benefits of LLM-enriched label descriptions, we are particularly interested in exploring different prompts to control how ChatGPT generates semantic details and their influence on OOD generalizability. We design the

following prompts: More Succinct: “Use a single sentence to broadly describe the appearance and shape of {cls}. Don’t give too many details. Only describe the shape and appearance.” More Detailed: “Use a single sentence and short, simple, descriptive phrases to describe the detailed appearance and detailed shape of {cls}.” More Distinct: “Use a single sentence to describe the unique, distinctive appearance and shape of {cls}. Only describe the unique, distinctive shape and appearance.”

Table 5: Results on leveraging different prompts to control semantic details of label descriptions generated by ChatGPT.

	StanfordCards	tiered-ImageNet
No language enrichment	63.9 / 19.8 / 38.5	67.5 / 20.5 / 35.3
Prompt in Table 4	63.3 / 20.0 / 37.5	69.8 / 23.5 / 36.2
More Succinct	63.0 / 18.9 / 37.6	68.8 / 23.1 / 36.8
More Detailed	62.9 / 19.0 / 35.1	69.3 / 24.2 / 37.7
More Distinct	63.9 / 19.7 / 37.1	69.2 / 23.3 / 37.0

We compare these prompts in Table 5. Interestingly, we observe that generating more detailed semantic descriptions on labels does not always perform better. We conjecture that this is because (1) LLM-generated details are not grounded in specific images, causing some attributes to be invisible and confusing the students; (2) the teacher CLIP is trained on LAION [42], where most language descriptions do not contain many fine-grained appearance details, so CLIP’s text embeddings are not very sensitive to some of these details. Additionally, we find that explicitly prompting ChatGPT to generate more concise text descriptions could be still helpful. Upon further analysis, we find that the resulting generations remain highly descriptive, albeit with slightly fewer details (e.g., when describing a “trumbone”, the more concise description becomes “a brass instrument with a long cylindrical tube curved into an elongated S shape with a

flared bell at the end", whereas under our original prompt, additional details like "a sliding U-shaped section called the slide" are included.

In this work, we studied distillation of large teacher vision-language models into lightweight student models by focusing on open-vocabulary out-of-distribution (OOD) generalization for object classification using small to medium-scale datasets. We investigated strengthening students' OOD generalization ability from two key perspectives: first, by better imitating teacher's visual representation space and carefully promoting better teacher-student

vision-language alignment coherence; and second, by enhancing the teacher's language representations with informative and meaningful semantic attributes to effectively differentiate between different labels. We analyzed the efficacy and impact of our techniques by introducing metrics and conducting a comprehensive experimental analysis. Along this process, we significantly improve student's zero-shot and few-shot generalization performance on openvocabulary OOD classification tasks.

Acknowledgement

Chapter 2, in full, is a modified reprint of the material as it appears in the International Conference of Computer Vision, 2023. Li, Xuanlin, Yunhao Fang, Minghua Liu, Zhan Ling, Zhuowen Tu, and Hao Su. The dissertation author was the primary investigator and author of this paper.

REFERENCES

[1] Ling, Zhan, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. "Deductive verification of chain-of-thought reasoning." Advances in Neural Information Processing Systems 36 (2024).

[2] Ling, Zhan, Yunhao Fang, Xuanlin Li, Tongzhou Mu, Mingu Lee, Reza Pourreza, Roland Memisevic, and Hao Su. "Unleashing the Creative Mind: Language Model As Hierarchical Policy For Improved Exploration on Challenging Problem Solving." arXiv preprint arXiv:2311.00694 (2023).

[3] Li, Xuanlin, Yunhao Fang, Minghua Liu, Zhan Ling, Zhuowen Tu, and Hao Su. "Distilling large vision-language model with out-of-distribution generalizability." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2492-2503. 2023.

[4] Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. "Chain-of-thought prompting elicits reasoning in large language models." Advances in neural information processing systems 35 (2022): 24824-24837.

[5] Kojima, Takeshi, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. "Large language models are zero-shot reasoners." Advances in neural information processing systems 35 (2022): 22199-22213.

[6] Zhou, Denny, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, Ed Chi. "Least-to-most prompting enables complex reasoning in large language models." arXiv preprint arXiv:2205.10625 (2022).

[7] Shi, Freda, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tai, Sebastian Ruder, Denny Zhou, Dipanjan Das, Jason Wei. "Language models are multilingual chain-of-thought reasoners." arXiv preprint arXiv:2210.03057 (2022).

[8] Driess, Danny, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, Pete Florence. "Palm-e: An embodied multimodal language model." arXiv preprint arXiv:2303.03378 (2023).

[9] Drozdov, Andrew, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. "Compositional semantic parsing with large language models." In The Eleventh International Conference on Learning Representations. 2022.

[10] Lampinen, Andrew K., Ishita Dasgupta, Stephanie CY Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L. McClelland, Jane X. Wang, and Felix Hill. "Can

language models learn from explanations in context?." arXiv preprint arXiv:2204.02329 (2022).

[11] Lu, Pan, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. "Learn to explain: Multimodal reasoning via thought chains for science question answering." Advances in Neural Information Processing Systems 35 (2022): 2507-2521.

[12] Marasović, Ana, Iz Beltagy, Doug Downey, and Matthew E. Peters. "Few-shot self-rationalization with natural language prompts." arXiv preprint arXiv:2111.08284 (2021).

[13] Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, Yi Zhang. "Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv." arXiv preprint arXiv:2303.12712 (2023).

[14] Guerreiro, Nuno M., Duarte M. Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André FT Martins. "Hallucinations in large multilingual translation models." Transactions of the Association for Computational Linguistics 11 (2023): 1500-1517.

[15] Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. "Survey of hallucination in natural language generation." ACM Computing Surveys 55, no. 12 (2023): 1-38.

[16] Maynez, Joshua, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. "On faithfulness and factuality in abstractive summarization." arXiv preprint arXiv:2005.00661 (2020).

[17] Arora, Kushal, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. "Why exposure bias matters: An imitation learning perspective of error accumulation in language generation." arXiv preprint arXiv:2204.01171 (2022).

[18] Welleck, Sean, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. "Neural text generation with unlikelihood training." arXiv preprint arXiv:1908.04319 (2019).

[19] Wang, Xuezhi, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. "Self-consistency improves chain of thought reasoning in language models." arXiv preprint arXiv:2203.11171 (2022).

[20] Chen, Wenhu, Xueguang Ma, Xinyi Wang, and William W. Cohen. "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks." arXiv preprint arXiv:2211.12588 (2022).

[21] Lyu, Qing, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. "Faithful chain-of-thought reasoning." arXiv preprint arXiv:2301.13379 (2023).

[22] Hosseini, Mohammad Javad, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. "Learning to solve arithmetic word problems with verb categorization." In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 523-533. 2014.

[23] Cobbe, Karl, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, John Schulman. "Training verifiers to solve math word problems." arXiv preprint arXiv:2110.14168 (2021).

[24] Hendrycks, Dan, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. "Measuring mathematical problem solving with the math dataset." arXiv preprint arXiv:2103.03874 (2021).

[25] Ling, Wang, Dani Yogatama, Chris Dyer, and Phil Blunsom. "Program induction by rationale generation: Learning to solve and explain algebraic word problems." arXiv preprint arXiv:1705.04146 (2017).

[26] Srivastava, Aarohi, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown. "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models." arXiv preprint arXiv:2206.04615 (2022).

[27] Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, Ryan Lowe. "Training language models to follow instructions with human feedback." Advances in neural information processing systems 35 (2022): 27730-27744.

[28] Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever. "Learning transferable visual models from natural language supervision." In International conference on machine learning, pp. 8748-8763. PMLR, 2021.

[29] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[30] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltechucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010.

[31] Krause, Jonathan, Michael Stark, Jia Deng, and Li Fei-Fei. "3d object representations for fine-grained categorization." In Proceedings of the IEEE international conference on computer vision workshops, pp. 554-561. 2013.

[32] Parkhi, Omkar M., Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. "Cats and dogs." In 2012 IEEE conference on computer vision and pattern recognition, pp. 3498-3505. IEEE, 2012.

[33] Bossard, Lukas, Matthieu Guillaumin, and Luc Van Gool. "Food-101–mining discriminative components with random forests." In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13, pp. 446-461. Springer International Publishing, 2014.

[34] Xiao, Jianxiong, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. "Sun database: Large-scale scene recognition from abbey to zoo." In 2010 IEEE computer society conference on computer vision and pattern recognition, pp. 3485-3492. IEEE, 2010.

[35] Ren, Mengye, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. "Meta-learning for semi-supervised few-shot classification." arXiv preprint arXiv:1803.00676 (2018).

[36] Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database." In 2009 IEEE conference on computer vision and pattern recognition, pp. 248-255. Ieee, 2009.

[37] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

[38] Chung, Hyung Won, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, Jason Wei. "Scaling instruction-finetuned language models." arXiv preprint arXiv:2210.11416 (2022).

[39] Sanh, Victor, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, Alexander M. Rush. "Multitask prompted training enables zero-shot task generalization." arXiv preprint arXiv:2110.08207 (2021).

[40] Tian, Changyao, Wenhai Wang, Xizhou Zhu, Jifeng Dai, and Yu Qiao. "Vl-ltr: Learning class-wise visual-linguistic representation for long-tailed visual recognition." In European Conference on Computer Vision, pp. 73-91. Cham: Springer Nature Switzerland, 2022.

[41] Wang, Peng, 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, pp. 23318-23340. PMLR, 2022.

[42] Schuhmann, Christoph, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. "Laion-400m: Open dataset of clip-filtered 400 million image-text pairs." arXiv preprint arXiv:2111.02114 (2021).