



AFRL-RI-RS-TR-2023-167

TASKS ALGORITHMICALLY GIVEN LABELS ESTABLISHED VIA TRANSFERRED SYMBOLS (TAGLETS)

BROWN UNIVERSITY

SEPTEMBER 2023

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2023-167 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

PETER JEDRYSIK
Work Unit Manager

/ S /

JULIE BRICHACEK
Chief, Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE		2. REPORT TYPE		3. DATES COVERED	
SEPTEMBER 2023		FINAL TECHNICAL REPORT		START DATE AUGUST 2019	END DATE MAY 2023
4. TITLE AND SUBTITLE Tasks Algorithmically Given Labels Established via Transferred Symbols (TAGLETS)					
5a. CONTRACT NUMBER FA8750-19-2-1006		5b. GRANT NUMBER N/A		5c. PROGRAM ELEMENT NUMBER 61101E	
5d. PROJECT NUMBER		5e. TASK NUMBER		5f. WORK UNIT NUMBER R2VT	
6. AUTHOR(S) Michael Littman, Eli Upfal, Stephen Bach, and James Tompkin					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Brown University 115 Waterman St. Providence RI 02912				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RISB 525 Brooks Road Rome NY 13441-4505			10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI		11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RI-RS-TR-2023-167
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT We conducted research to reduce the amount of labeled data required to train machine learning systems. The pinnacle of this effort is the development of TAGLETS, a machine learning system that seamlessly integrates widely known collections of labeled data with a diverse array of machine learning algorithms, known as weak labelers. The system's evolution has been significantly influenced by comprehensive theoretical explorations into effectively aggregating these weak labelers within the system. The research's scope expands to the application of large pre-trained models in low-resource settings. The result of these efforts is Alfred, a second-generation system tailored for programmatic weak supervision that allows to labeled data via prompts. Furthermore, the research encompassed extensive investigations to elucidate strategies for fine-tuning large models for specific domain tasks with limited labeled data, enhancing adaptability and performance.					
15. SUBJECT TERMS Limited labeled data; semi-supervised learning; unsupervised learning; image classification; weak supervision; theory of weak supervision; weak supervision with large language and vision-language models, adaptation of large models with limited resources; zero-shot compositionality of vision-language models					
16. SECURITY CLASSIFICATION OF:				17. LIMITATION OF ABSTRACT	
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR		18. NUMBER OF PAGES 56
19a. NAME OF RESPONSIBLE PERSON PETER A. JEDRYSIK					19b. PHONE NUMBER (Include area code) N/A

Table of Contents

Section	Page
List of Figures	ii
List of Tables	ii
List of Equations	iii
1. SUMMARY	1
2. INTRODUCTION	4
3. METHODS, ASSUMPTIONS, AND PROCEDURES	8
3.1 TAGLETS (TA1)	8
3.1.1 Structured Collection of Auxiliary Data Sets (SCADS)	11
3.1.2 Modules	12
3.1.3 ZSL-KG (TA1)	13
3.1.4 Theoretical Guarantees for ZSL (TA2)	16
3.1.5 Aggregating weak labelers (TA1 and TA2)	19
3.2 ADAPTATION OF PRE-TRAINED MODELS WITH LOW RESOURCES	28
3.2.1 Enhancing CLIP with limited-label prompt tuning (TA1)	28
3.2.2 Compositionality of CLIP (TA1)	33
3.2.3 Alfred (TA1)	36
4. RESULTS AND DISCUSSIONS	40
5. CONCLUSIONS	42
6. REFERENCES	43
7. LIST OF ACRONYMS & GLOSSARY	49

List of Figures

Figure	Page
Figure 1. A Summary of Brown University Research.....	1
Figure 2. The Usage of Data Before and After TAGLETS	9
Figure 3. TAGLETS’s Architecture	10
Figure 4. A Subset of a SCADS	12
Figure 5. ZSL-KG’s Architecture	14
Figure 6. Comparison of the Zero-Shot Learning Lower Bound with the Empirical Error.	18
Figure 7. Pairwise Misclassification Matrices.....	19
Figure 8. Accuracy of Different Aggregation Methods.....	22
Figure 9. Experiments Results of AMCL on Domain Net.....	23
Figure 10. Algorithm’s Window Sizes and Weak Labelers Accuracies.....	25
Figure 11. Examples of the Expressivity of Partial Labeling Functions	26
Figure 12. Our Design Space to Explore the Effect of Leveraging Pseudolabels in a Unified Way Across Prompt Modalities, Learning Paradigms, and Training Strategies.....	29
Figure 13. Evolution of Pseudolabels Accuracy During Training.....	32
Figure 14. Improvements of FPL and GRIP on CLIP’s Per-Class Accuracies (RESICS45).....	33
Figure 15. An Overview of Compositional Zero-Shot Learning with CSP.....	34
Figure 16. Example Images from Each Dataset.....	36
Figure 17. Examples of a Labeling Function Versus a Prompted Labeling Function.....	37
Figure 18. A Typical Workflow for Programmatic Weak Supervision with Alfred	38

List of Tables

Table	Page
Table 1. Accuracy of TAGLETS and Baselines on OfficeHome Product and for Increasing Number of Labeled, Different Backbones, and Levels of Pruning.....	11
Table 2. Performance of ZSL-KG for Generalized Zero-Shot Fine-Grained Entity Typing on Ontonotes and BBN	15
Table 3. Performance of ZSL-KG for Object Classification on ImageNet.	15
Table 4. Performance of ZSL-KG Generalized Zero-Shot Object Classification on the AWA2 and aPY Datasets.	15
Table 5. Results for Text Classification Using NPLM	27
Table 6. Results for Object Classification	28
Table 7. Accuracy of GRIP Compared to CLIP Zero-Shot (ViT-B/32), CoOp, and VPT.....	30
Table 8. Compared Accuracies of FPL, IFPL, and GRIP.....	31
Table 9. Closed-world (Closed) and Open-world (Open) Results on MIT-States, UT-Zappos, and C-GQA.....	34

Table 10. Generalization to High-Order Composition	35
Table 11. Top-1 Accuracy on YouTube Spam Detection.....	39
Table 12. Top-1 Accuracy on Oxford-IIIT Pet Breed Classification	39

List of Equations

Equation	Page
Equation 1. Lower Bound for Zero-Shot Learning with Attributes.....	17
Equation 2. Lower Bound for Performance-Guaranteed Majority Vote (PGMV)	20
Equation 3. Closed Form Solution to the Linear Problem.....	21
Equation 4. Guarantees on the Estimation of the Correlation Matrix.....	24
Equation 5. Upper Bound for the Expected Difference of the Empirical Correlation Matrices for Identical Distributed Samples	24

1. SUMMARY

We present a concise overview of the research endeavors of the team from Brown University in Figure 1. In the context of the DARPA Learning with Less Labels program (LwLL), our team efforts have encompassed both Technical Area 1 (TA1) and Technical Area 2 (TA2). The goal of TA1 was to develop learning algorithms that: (1) reduce the amount of labeled data required to build a model from scratch by at least a factor of 10^6 ; and (2) adapt to new environments with hundreds of labeled examples. The goal of TA2 was to formally prove limits on the amount of labeled data needed to solve a given machine learning problem. Specifically, our contributions have been twofold: first, we developed widely applicable systems and methods to reduce the need of labeled data for learning, and second, we define robust theoretical analysis and frameworks that underpin the paradigms of programmatic weak supervision and zero-shot learning.

The emergence of pre-trained large language and vision-language models during the program has posed a new challenge, as these models' remarkable generalization capabilities have significantly elevated the baseline performance. While these advancements have lessened the demand for labeled data when refining models for new target tasks, the availability of sufficient target-specific data remains crucial, especially for domains distant from the models' training data. Considering this, our team directed its attention toward the parameter efficient adaptation of these large models in resource-limited scenarios.

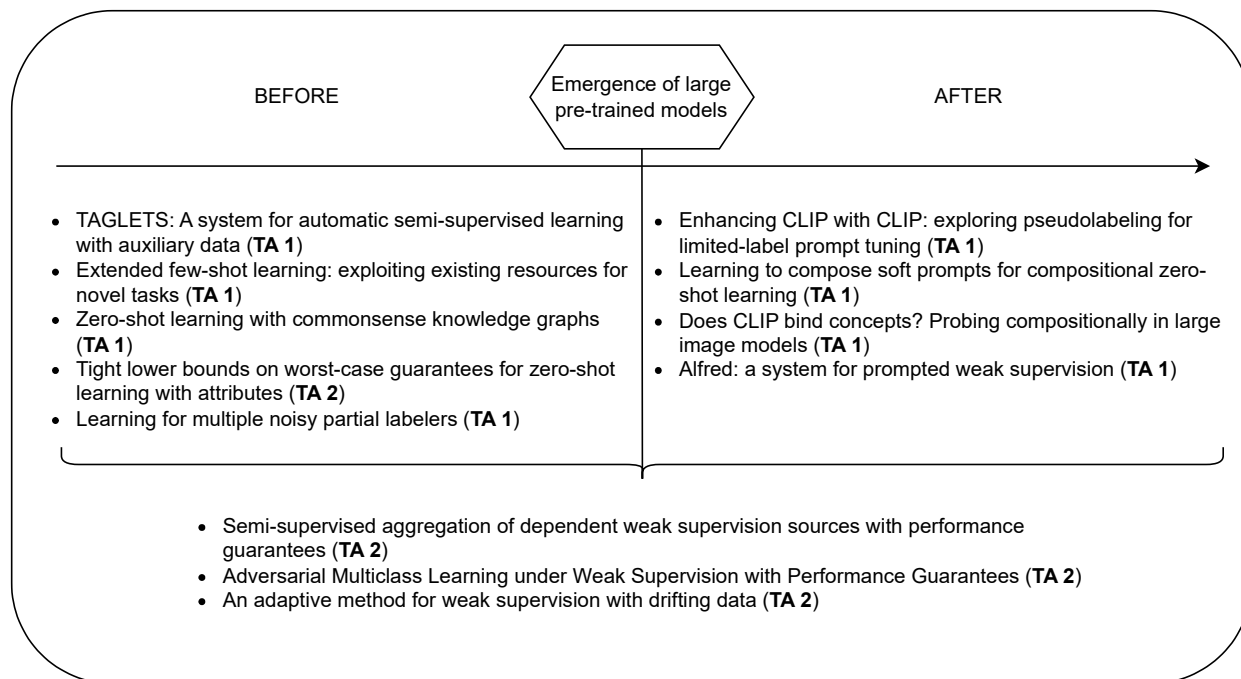


Figure 1. A Summary of Brown University Research

Overall, we summarize our contributions as it follows:

- We delivered TAGLETS [1], the first end-to-end machine learning system that automatically integrates ecosystems of data and models in a unified way. In the low data regime, there is no rule of thumb to pick the best learning strategy a-priori. TAGLETS includes multiple algorithms from various learning paradigms, such as transfer, semi-supervised, and zero-shot learning. Each model is initially fine-tuned on target-related auxiliary data extracted from a collection of annotated labeled data for other tasks [1, 2]. Then, the trained models are used as weak labelers to produce pseudolabels for the unlabeled data. The newly labeled data and the few labeled data available are fed to a final model, being the system's output. TAGLETS was originally built using smaller pre-trained models as backbones for the models. Further experiments that replace such backbones with larger pre-trained models have shown that TAGLETS' architecture and focus of task-related auxiliary data still benefit even larger pre-trained models. **(TA1)**
- We devised a novel zero-shot learning approach that uses common sense knowledge graphs to enrich the information about unseen target classes (ZSL-KG) [3], and we included the method in TAGLETS to diversify the pool of methods in the system. In the context of zero-shot learning with attributes that describe classes, we developed the first non-trivial lower bound on the worst-case error of the best map from attributes to classes, even with perfect attribute detectors [4]. The lower bound characterizes the theoretical intrinsic difficulty of the zero-shot problem based on the available information—the class-attribute matrix—and the bound is practically computable from it. **(TA1 and TA2)**
- We explored how to use pseudolabeling to improve Vision-Language Models (VLMs) capability on new tasks under different low resource learning paradigms [5]. Our main observation is that learning via pseudolabeling consistently improves the performance of Contrastive Language-Image Pre-training (CLIP) by using the same learning strategy in the semi-supervised, unsupervised, and transductive zero-shot learning settings. In addition, tuning CLIP via prompts using pseudolabels mitigates CLIP's biases toward certain classes. **(TA1)**
- We developed a new method to learn how to compose concepts using CLIP (Compositional Soft Prompts) [6]. Our method performs well for compositional zero-shot learning, the task of predicting unseen attribute-object compositions (e.g., old cat and young tiger). In addition to propose this new approach, we studied the ability of CLIP to encode compositional concepts and to bind variables in a structure-sensitive way (e.g., differentiating cube behind sphere from sphere behind cube) [7]. **(TA1)**
- We theoretically studied and provide new methods for combining weak labelers after removing the independence assumption. Particularly, we devised solutions with theoretical guarantees on the errors [8, 9]. We expanded this analysis to drifting data [10]. Moreover, to overcome the assumption that labelers output a single class label, we study the setting where users are enabled to create partial labeler that output subsets of possible class labels [11]. **(TA1 and TA2)**
- After the emergence of large language and vision-language models, we developed Alfred: the first framework that allows to label large collections of unlabeled data via prompts [12]. **(TA1)**

Impact of our methods on the overall scope of the program.

- Brown University evaluated its system TAGLETS on the image and video classification tasks designed by JPL. Moreover, we evaluated GRIP, a CLIP-based prompt tuning method exploiting pseudolabels, to solve JPL's zero-shot learning task.
 - In the image classification tasks, TAGLETS appears among the top-6 performers.
 - For the video classification tasks, TAGLETS is the best performing approach when the quantity of labeled data increases.
 - In the zero-shot learning task, GRIP has been the top-1 performer with a +20% improvement on the baseline.

2. INTRODUCTION

The conventional paradigm of learning with limited labeled data revolves around two key steps. First, it entails learning generalizable features from accessible data, which can be few labeled data, unlabeled data, or annotated data from other context. Second, methods are devised to effectively transfer the knowledge acquired during pre-training to tackle specific target tasks.

In addressing this challenge, our strategy has centered on achieving automated knowledge transfer across datasets and algorithms. This involves the integration of ecosystems of data and methods into a unified system (TAGLETS), ultimately aimed at solving target tasks despite having limited labeled data and a surplus of unlabeled data (TA1). Additionally, we conducted theoretical research on strategies for combining the outcomes of different learning methods employed in our system (TA2).

- **Automatic Knowledge Transfer:** We consider the setting where practitioners have access to a spectrum of data: labeled data for the target task (which is often limited), unlabeled data, and auxiliary data, the many available labeled datasets for other tasks. We developed TAGLETS, a system built to study techniques for automatically exploiting all three types of data and creating high-quality, servable classifiers. The key components of TAGLETS are (1) auxiliary data organized according to a knowledge graph, (2) modules encapsulating different methods for exploiting auxiliary and unlabeled data, and (3) a distillation stage in which the ensembled modules are combined into a servable model. In the context of the grant, we compared TAGLETS with the systems of other teams on image classification and video classification tasks ending up among the top-6 performers. Moreover, we also evaluate TAGLETS on datasets designed for specific applications, not necessarily describing general visual domains. We compared the system with state-of-the-art transfer learning and semi-supervised learning methods covering a range of settings, by varying the amount of labeled data and the semantic relatedness of the auxiliary data to the target task. We find that the intelligent incorporation of auxiliary data and unlabeled data into multiple learning techniques enables TAGLETS to match – and most often significantly surpass – these alternatives. TAGLETS is available as an open-source system at <https://github.com/BatsResearch/taglets>.
 - The training modules integrated in TAGLETS are either variations of known learning algorithms tailored to exploit auxiliary data, or methods that we designed and developed to diversify the pool of automatic weak labelers.
 - **ZSL-KG:** Zero-shot learning relies on semantic class representations such as hand-engineered attributes or learned embeddings to predict classes without any labeled examples. We propose to learn class representations by embedding nodes from common sense knowledge graphs in a vector space. Common sense knowledge graphs are an untapped source of explicit high-level knowledge that requires little human effort to apply to a range of tasks. To capture the knowledge in the graph, we introduce ZSL-KG, a general-purpose framework with a novel transformer graph convolutional network (TrGCN) for generating class representations. Our proposed TrGCN architecture computes non-linear combinations of node neighbourhoods. Our results show that ZSL-KG improves over existing WordNet-based methods on five out of six zero-shot benchmark datasets in language and vision. The code is available at

<https://github.com/BatsResearch/zsl-kg>. ZSL-KG is part of TAGLETS and provides zero-shot learning predictions on the unlabeled data.

- **Extended few-shot learning:** In many practical few-shot learning problems, even though labeled examples are scarce, there are abundant auxiliary datasets that potentially contain useful information. Given a large auxiliary dataset and a notion of semantic similarity among classes, we automatically select pseudo-shots, which are labeled examples from other classes related to the target task. To adjust the features of auxiliary data to be more similar to those of the target classes, we propose a masking module. This approach brought significant improvements over conventional few-shot transfer learning. We release the code of the method at <https://github.com/BatsResearch/efsl>
- **Rigorously combine weak source of supervision:** The third component of TAGLETS consists in combining the outcomes of different modules, i.e., weak labelers, such that we can annotate unlabeled data for the final distillation stage. Programmatic weak supervision created models without hand-labeled training data by combining the outputs of heuristic weak labelers. We identify two main restrictive assumptions in this framework: (1) the errors of the weak labelers are independent or come from a particular family of distributions, and (2) labelers output a single class label.
 - We developed a novel method that provides theoretical guarantees for learning from weak labeled without the (mostly unrealistic) assumption that the errors of the weak labelers are independent or come from a particular family of distributions. We showed a rigorous technique for efficiently selecting small subsets of the labelers so that a majority vote from such subsets had a provably low error rate. We expand such approach to the multiclass classification task when only a very small set of labeled data is available, by developing a rigorous approach for using a set of arbitrarily correlated weak supervision sources. Our learning algorithm provably converges to a model that has minimum empirical risk with respect to an adversarial choice over feasible labelings for a set of unlabeled data, where the feasibility of a labeling is computed through constraints defined by rigorously estimated statistics of the weak supervision sources. We showed theoretical guarantees for this approach that depend on the information provided by the weak supervision sources. Notably, this method does not require the weak supervision sources to have the same labeling space as the multiclass classification task. We studied the effectiveness of this approach with experiments on various image classification tasks and by integrating such approach into TAGLETS.
 - To overcome the assumption that labelers output a single class label, we study the setting where users are enabled to create partial labeler that output subsets of possible class labels. This would greatly expand the expressivity of programmatic weak supervision. We introduced this capability by defining a probabilistic generative model that can estimate the underlying accuracies of multiple noisy partial labelers without ground truth labels. We showed how to scale up learning, and also prove that the defined class of models is generically identifiable up to label swapping under mild conditions. We evaluated the approach on text and object classification tasks – demonstrating its effectiveness compared to standard programmatic supervision approaches.

Considering the emergence of pre-trained large language and vision-language models, renowned for their remarkable generalization capabilities, we decided to align with the significant

advancements they bring to the field. While these models nearly obviate the necessity for curating the pre-training aspect, it is worth noting that despite their robust generalizability, these models still require tailoring for specific domains and tasks.

To keep working on **TA1**, our primary focus has been on achieving efficient parameter adaptation of such models, particularly when dealing with vision-language models in the context of limited labeled data. Additionally, we introduce a fresh framework that enables annotating unlabeled datasets using large language models as weak labelers, which can still benefit from the theory developed for initial effort of the team (**TA2**).

- **Adaptation of pre-trained models with low resources:** We studied vision language models from the perspective of parameter efficient adaptation. In this setting, we studied the ability of such models, like CLIP, of combining known concepts to identify new objects never seen before (zero-shot classification). In addition, we also conduct an extensive study about the usage of pseudolabels generated by large models for limited-label prompt tuning. The latter approach, named GRIP, has been applied to the zero-shot task developed by JPL for the last evaluation. The method turned out to be the top-1 performer bringing +20 points of improvement on the baseline.
 - **Adapting CLIP with pseudolabels.** Fine-tuning vision-language models (VLMs) like CLIP to downstream tasks is often necessary to optimize their performance. However, a major obstacle is the limited availability of labeled data. We study the use of pseudolabels, i.e., heuristic labels for unlabeled data, to enhance CLIP via prompt tuning. Conventional pseudolabeling trains a model on labeled data and then generates labels for unlabeled data. VLMs's zero-shot capabilities enable a "second generation" of pseudolabeling approaches that do not require task-specific training on labeled data. By using zero-shot pseudolabels as a source of supervision, we observe that learning paradigms such as semi-supervised, transductive, zero-shot, and unsupervised learning can all be seen as optimizing the same loss function. This unified view enables the development of versatile training strategies that are applicable across learning paradigms. We investigate them on image classification tasks where CLIP exhibits limitations, by varying prompt modalities, e.g., textual or visual prompts, and learning paradigms. We find that (1) unexplored prompt tuning strategies that iteratively refine pseudolabels consistently improve CLIP accuracy, by 19.5 points in semi-supervised learning, by 28.4 points in transductive zero-shot learning, and by 15.2 points in unsupervised learning, and (2) unlike conventional semi-supervised pseudolabeling, which exacerbated model biases toward classes with higher-quality pseudolabels, prompt tuning leads to a more equitable distribution of per-class accuracy. The code to reproduce the experiments is at <https://github.com/BatsResearch/menghini-enhanceCLIPwithCLIP-code>.
 - **CLIP's compositionality.** We introduce compositional soft prompting (CSP), a parameter-efficient learning technique to improve the zero-shot compositionality of large-scale pretrained VLMs like CLIP. We develop CSP for compositional zero-shot learning, the task of predicting unseen attribute-object compositions (e.g., old cat and young tiger). VLMs have a flexible text encoder that can represent arbitrary classes as natural language prompts but they often underperform task specific architectures on the compositional zero-shot benchmark datasets. CSP treats the attributes and objects that define classes as learnable tokens of vocabulary. During training, the vocabulary is tuned to recognize

classes that compose tokens in multiple ways (e.g., old cat and white cat). At test time, we recompose the learned attribute-object vocabulary in new combinations to recognize novel classes. We show that CSP outperforms the CLIP on benchmark datasets by an average of 10.9 percentage points on Area Under the Curve (AUC). CSP also outperforms CoOp, a soft prompting method that fine-tunes the prefix context tokens, by an average of 5.8 percentage points on AUC. We perform additional experiments to show that CSP improves generalization to higher-order attribute-attribute-object compositions (e.g., old white cat) and combinations of pretrained attributes and fine-tuned objects. The code is available at <https://github.com/BatsResearch/csp>.

- **Alfred: A system for programmatic weak supervision.** Alfred is the first system for programmatic weak supervision (PWS) that creates training data for machine learning by prompting. In contrast to typical PWS systems where weak supervision sources are programs coded by experts, Alfred enables users to encode their subject matter expertise via natural language prompts for language and vision-language models. Alfred provides a simple Python interface for the key steps of this emerging paradigm, with a high-throughput backend for large-scale data labeling. Users can quickly create, evaluate, and refine their prompt-based weak supervision sources; map the results to weak labels; and resolve their disagreements with a label model. Alfred enables a seamless local development experience backed by models served from self-managed computing clusters. It automatically optimizes the execution of prompts with optimized batching mechanisms. Alfred is open source, available at <https://github.com/BatsResearch/alfred>.

In the rest of the document, we will provide more detailed information about each of the main contributions highlighted above.

3. METHODS, ASSUMPTIONS, AND PROCEDURES

We frame our research within two periods: before and after the introduction of large pre-trained models. Before the spread of the latter, we developed TAGLETS - the original core deliverable of the project - a system that for the first time integrated under a unified framework a multitude of datasets and models while reducing the need for labeled data and maximizing the exploitation of unlabeled task-related data.

The self-supervised learning techniques that characterize the development of large pre-trained models have led to models with remarkable generalization abilities which substantially reduce the need for labeled data to effectively adapt models to novel tasks. In this context, we focus on understanding how adaptation can work under various learning settings and for specific applications, while keep focusing on weak supervision systems that exploit multiple models.

3.1 TAGLETS (TA1)

The bottleneck of labeled training data in machine learning has driven research on exploiting alternative, more abundant sources of data such as unlabeled data and auxiliary data, i.e., labeled data for possibly related tasks other than the target task.

Automatically exploiting auxiliary data, unlabeled data, and limited labeled data for the target task simultaneously requires addressing three main challenges:

1. **Relevant auxiliary data is a needle in a haystack.** Often a huge number of auxiliary datasets are available, but likely only a very small subset is significantly related to any given target task. Further, related examples in auxiliary datasets can be useful for many distinct reasons. In the object classification setting, related datasets might contain examples of visually similar objects, functionally similar objects, objects likely to co-occur in similar settings, and more. How can we automatically find the relevant data quickly?
2. **Best practices vary by amount of training data.** The most accurate methods for learning with limited labeled data vary according to several factors, including how much labeled data is available and how related auxiliary data is to the target task. How can we automatically be robust to these variations?
3. **Pipelines are difficult to serve in production.** In low-latency applications with service-level agreements, it is necessary to make predictions in a fixed amount of time. This requirement limits the complexity of prediction pipelines that are practical. How can we distill all the information in labeled training data, unlabeled data, and auxiliary data into a single, servable model?

We address these challenges with TAGLETS, a modular end-to-end machine learning system for automatic semi-supervised learning with auxiliary data.

TAGLETS automatically selects relevant subsets of auxiliary data and makes them available to semi-supervised, transfer, and zero-shot learning modules. The modules create an intermediate pseudolabeling, and then TAGLETS distills them into a final, servable model. The system is designed to work with limited labeled data, instead relying on the auxiliary annotated datasets publicly available and internal to organizations. Additionally, unlabeled data from the task

distribution is used to further boost performance. In Figure 2, we depict the differences between common practices in using data for different learning paradigms and the approach we use with TAGLETS. Side (A) represents the common practice. On the left, three categories of data separately passed as input to methods belonging to diverse learning approaches. Each method returns a distinct output. Side (B) represents the envisioned approach. Methods can be trained on all available data. The predictions of such methods are then combined to produce the intermediate pseudolabels for training an end model.

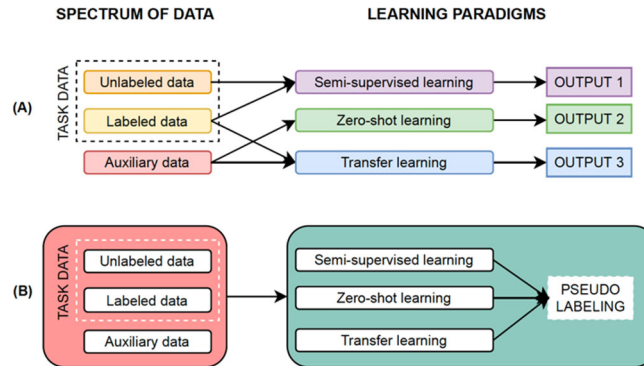


Figure 2. The Usage of Data Before and After TAGLETS

We have distilled principles that guided the design of TAGLETS from our experience working with mixtures of labeled, unlabeled, and auxiliary data.

1. **Select the best auxiliary data available.** In some cases, a target task might consist of classes that overlap substantially with those in auxiliary data. In others, the auxiliary might be only more loosely related. The system should allow models to transfer knowledge from auxiliary data in both cases.
2. **Learn from limited target data.** Different scenarios have access to different amounts of labeled data. The system should be effective when learning from one training example per target class to dozens.
3. **Automatically distill to a servable model.** A user should be able to create a model by providing a minimal semantic description of their target task, and any available labeled and unlabeled data. The system should then automatically produce a high-quality model small enough to be served in production.

We designed TAGLETS to address the above challenges and principles. The key contributions of our work are:

- **Structured collection of datasets.** We introduce the concept of a structured collection of annotated datasets (SCADS). A SCADS gathers all available labeled datasets and connects concepts described in those datasets via a knowledge graph. This repository of linked data is where TAGLETS automatically sources the auxiliary data. SCADS is described in Section 3.1.1.
- **Ensemble of methods exploiting auxiliary data.** We propose ensembling methods that can exploit a SCADS for robust predictive accuracy. We show how to adapt existing transfer and

semi-supervised learning methods to exploit a SCADS in a modular framework. Zero-shot learning based on knowledge graphs further boosts accuracy. We show that ensembling these methods is key to robust accuracy across a range of scenarios covering different amounts of labeled target data and relatedness of the data in the SCADS to the target task. See Section 3.1.2.

- **End-to-end system.** TAGLETS is the first automatic end-to-end machine learning system that simultaneously exploits ecosystems of auxiliary data and models. It achieves state-of-the-art performances on practically useful tasks with limited labeled data. We release it as an open-source package at github.com/BatsResearch/taglets.

The architecture of TAGLETS (Figure 3) consists of (1) SCADS which organizes and finds auxiliary data related to the target classes. (2) Multiple modules that are trained to perform pseudolabeling. (3) In the distillation stage, we pseudo-label the unlabeled data and train the end model on both the pseudolabeled and labeled data.

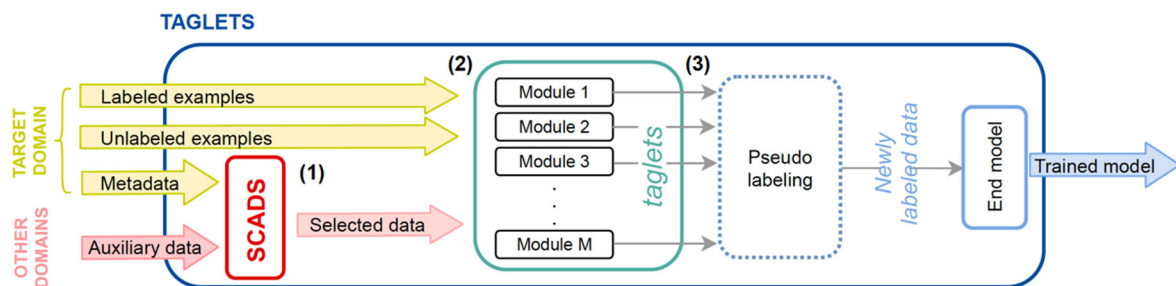


Figure 3. TAGLETS’s Architecture

Beyond JPL’s tasks, we evaluate TAGLETS by comparing with state-of-the-art transfer and semi-supervised learning methods on four practical object classification tasks. Our study includes experiments with a range of labeled data for target tasks (from 1 shot per class to 20), a range of available auxiliary data, and multiple image embedding backbones. There are several key findings. First, when auxiliary data is closely related to the target task, TAGLETS is most beneficial in the few-shot setting. We find here that TAGLETS improves accuracy over BigTransfer [13] by an average 11.9 percentage points in the one-shot case and 2.13 in the five-shot case. It also improves over the best-performing semi-supervised learning method by an average of 11.54 and 1.96 points in the one-shot and five-shot cases, respectively. In Table 1, we report the accuracy (%) of TAGLETS and baselines on OfficeHome Product and Clipart (Split 0) for increasing number of labeled examples (1, 5, and 20 shots), different backbones (ResNet-50 trained on ImageNet-1k and BiT trained on ImageNet-21k) and levels of pruning (no-pruning, 0 and 1). We bold and underline the best performances, and those within their 95% confidence intervals, carried out using BiT and ResNet backbones, respectively. Measurements are repeated for three seeds and we report the 95% confidence intervals.

In an alternative scenario in which only more distantly related auxiliary data is available, TAGLETS significantly outperforms traditional transfer learning and state-of-the-art semi-supervised learning methods in the one-shot and five-shot cases. In addition, we conduct many

ablation experiments to analyze how the multiple components of TAGLETS contribute to its performance. These results demonstrate that automatically selecting auxiliary data, and then exploiting it in combination with unlabeled data and limited labeled data, can lead to large improvements in accuracy across a range of scenarios.

Table 1. Accuracy of TAGLETS and Baselines on OfficeHome Product and for Increasing Number of Labeled, Different Backbones, and Levels of Pruning

Method	Backbone	OfficeHome-Product			OfficeHome-Clipart		
		1-shot	5-shot	20-shot	1-shot	5-shot	20-shot
Fine-tuning	BiT (ImageNet-21k)	57.28 $\pm$ 5.20	84.41 $\pm$ 1.17	90.67 $\pm$ 1.10	30.36 $\pm$ 2.54	63.44 $\pm$ 2.10	77.13 $\pm$ 1.34
Fine-tuning (Distilled)	BiT (ImageNet-21k)	58.21 $\pm$ 4.56	85.03 $\pm$ 0.22	92.00 $\pm$ 0.38	31.74 $\pm$ 1.96	62.92 $\pm$ 1.99	78.26 $\pm$ 0.88
FixMatch	BiT (ImageNet-21k)	49.03 $\pm$ 2.54	83.03 $\pm$ 2.97	92.21 $\pm$ 0.44	30.67 $\pm$ 5.44	64.77 $\pm$ 1.75	79.95 $\pm$ 2.30
Meta Pseudo Label	BiT (ImageNet-21k)	59.95 $\pm$ 0.44	81.53 $\pm$ 1.01	89.90 $\pm$ 0.88	31.74 $\pm$ 2.1	62.92 $\pm$ 2.13	76.05 $\pm$ 1.72
TAGLETS	BiT (ImageNet-21k)	70.92 $\pm$ 1.67	86.31 $\pm$ 1.01	92.36 $\pm$ 3.18	46.97 $\pm$ 2.46	67.64 $\pm$ 3.34	78.62 $\pm$ 2.76
Fine-tuning	ResNet-50 (ImageNet-1k)	32.51 $\pm$ 3.83	63.44 $\pm$ 7.45	81.13 $\pm$ 0.22	16.26 $\pm$ 2.87	43.49 $\pm$ 0.58	68.56 $\pm$ 2.24
Fine-tuning (Distilled)	ResNet-50 (ImageNet-1k)	32.72 $\pm$ 3.67	66.87 $\pm$ 5.60	83.85 $\pm$ 2.68	15.85 $\pm$ 0.66	44.26 $\pm$ 2.34	72.00 $\pm$ 2.32
FixMatch	ResNet-50 (ImageNet-1k)	39.28 $\pm$ 3.34	72.87 $\pm$ 1.45	85.03 $\pm$ 1.81	20.51 $\pm$ 0.96	49.69 $\pm$ 3.31	66.56 $\pm$ 1.23
Meta Pseudo Label	ResNet-50 (ImageNet-1k)	47.38 $\pm$ 2.98	76.67 $\pm$ 1.75	87.38 $\pm$ 1.75	22.77 $\pm$ 3.99	57.25 $\pm$ 4.64	75.17 $\pm$ 1.22
TAGLETS	ResNet-50 (ImageNet-1k)	67.64 $\pm$ 3.61	83.54 $\pm$ 2.89	88.77 $\pm$ 2.89	43.44 $\pm$ 4.46	63.90 $\pm$ 1.17	75.54 $\pm$ 4.34
TAGLETS prune-level 0	ResNet-50 (ImageNet-1k)	<u>65.33 $\pm$ 1.23</u>	<u>81.23 $\pm$ 0.00</u>	<u>88.62 $\pm$ 1.38</u>	<u>41.49 $\pm$ 1.54</u>	62.21 $\pm$ 1.89	<u>75.79 $\pm$ 3.11</u>
TAGLETS prune-level 1	ResNet-50 (ImageNet-1k)	57.28 $\pm$ 1.89	76.21 $\pm$ 1.54	<u>86.67 $\pm$ 0.58</u>	36.87 $\pm$ 2.17	55.33 $\pm$ 4.19	<u>72.72 $\pm$ 0.96</u>

3.1.1 Structured Collection of Auxiliary Data Sets (SCADS)

The goal of SCADS is to help identify a relevant subset of auxiliary data for a target task. In general, categories described in annotated datasets can overlap or relate to each other. For instance, given two datasets, both gathering images of postcards, we need to query them separately to extract all the corresponding examples. Further, when two datasets collect images of related categories, e.g., one contains images of paper, and the other of cards and postcards, we need to automatically identify that these examples are related to associate them. We therefore design SCADS to efficiently retrieve examples from related categories using semantic relations among them. Figure 4 shows an example of a subset of a SCADS: (A) All images of a category belong to the same corresponding concept, and each concept is associated with an embedding, i.e., SCADS embedding, which can be used to determine the relations among concepts. (B) Given a query class plastic, SCADS retrieves the most similar concepts, according to graph-based similarity.

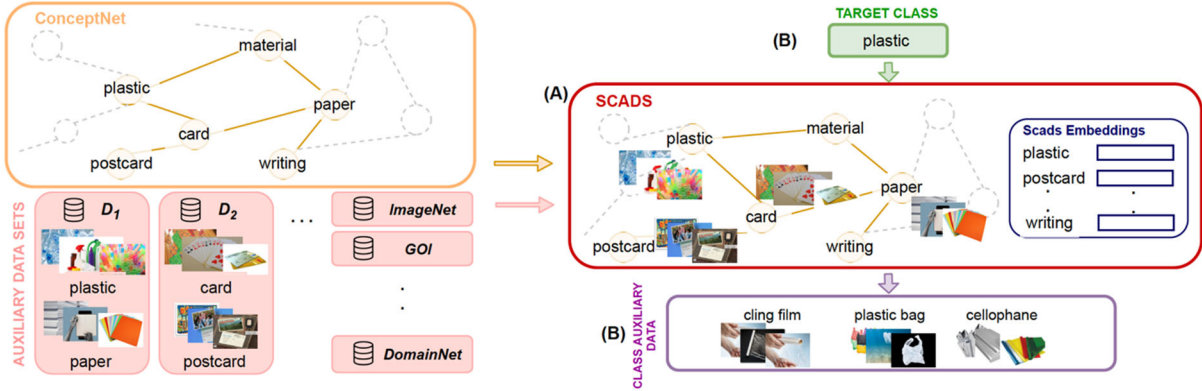


Figure 4. A Subset of a SCADS

Construct SCADS. Users can construct a SCADS for their domain by joining all of their data to a common sense knowledge graph. Common sense knowledge graphs are powerful resources that make explicit many facts about the world by stating relationships among millions of concepts. We propose using them as the underlying structure of SCADS. In particular, we choose ConceptNet [14], a common sense knowledge graph $G = (Q, E)$ whose nodes Q are words and phrases of natural language (i.e., concepts), and edges E express their relations. To build SCADS, we map the classes in the auxiliary data and concepts. In the knowledge graph, such that all the images of the same class are associated to the corresponding node in G (Figure 4(A)). SCADS enables end-users to easily install and remove any available annotated dataset. Its extensibility makes possible to unlock the potential of datasets that otherwise could not as easily be used as standalone sources of information. This joining process requires a one-time labor cost, but we find that it is small and can be automated for large datasets like ImageNet-21k using the many links in ConceptNet to external resources like WordNet and Wikipedia.

Select auxiliary data. Joining auxiliary data and ConceptNet defines a rich set of semantic relations on top of examples, and we want to exploit them to select a relevant subset of auxiliary data for the target task. Typical approaches to find data related to the target classes rely on visual similarity [15, 16]. However, such methods can be computationally expensive, by involving pairwise calculations. In our setting, we look for task-related data among enormous amounts of auxiliary data (e.g., ImageNet-21k), so visual pairwise-comparisons become intractable. Thus, we leverage SCADS underlying structure and use graph-based semantic similarity between concepts in SCADS and a target class c , to find the subset of concepts mostly related to c . Further, we get from the most related concepts all the corresponding images.

3.1.2 Modules

In TAGLETS, we include four methods in the areas of transfer, semi-supervised and zero-shot learning, and adapt them to fully exploit the knowledge collected in SCADS. We call a method tailored to use SCADS a module and its output (i.e., a trained model) a *taglet*. Modules are independently trained, and their predictions are ensembled to take advantage of their diversity. This modular framework is extensible, as other methods can be incorporated on top of the ones we develop here.

Specifically, the modules in the current version of TAGLETS are:

- The *transfer module* which fine-tunes sequentially on task-related auxiliary and labeled data to leverage the knowledge in SCADS.
- The *multi-task module* forms an auxiliary classification task out of selected auxiliary data, most semantically similar to the target data, and jointly learns the target task and the auxiliary task.
- The *FixMatch module* which is the FixMatch [17] algorithm with the backbone previously pre-trained on auxiliary data selected by SCADS.
- The *ZSL-KG module* which works in the zero-shot learning setting and is presented in the next section.

3.1.3 ZSL-KG (TA1)

Finding generalized representations of target classes with minimal data can be hard, especially with only one labeled example per class. To complement the first three modules, we include a zero-shot learning module that makes predictions without any labeled data for the target task. Instead, it uses only the knowledge graph in SCADS.

Zero-shot learning is a training strategy which allows a machine learning model to predict novel classes without the need for any labeled examples for the new classes [18, 19, 20]. These models are trained on a set of labeled examples from seen classes, along with their class representations. During inference, new class representations are provided for the unseen classes. Previous zero-shot learning systems have used hand-engineered attributes [21, 22], pretrained embeddings [23] and learned embeddings (e.g., sentence embeddings) as class representations [24].

Previous approaches for class representations have various limitations. Attribute-based methods provide rich features and have achieved state-of-the-art results on several zero-shot object classification datasets, but the attributes have to be fixed ahead of time for the unseen classes and cannot adapt to new classes beyond the dataset. Furthermore, creating attribute datasets can take up to thousands of hours of labor [25]. Finally, attribute-based methods may not be readily applicable to tasks in language, as they might require greater nuance and flexibility [26].

Recently, there is growing interest in methods using graph neural networks on the ImageNet graph, a noun subset of the WordNet graph, to learn to map nodes to class representations [27]. These graph-based methods have achieved strong performance on zero-shot object classification. They offer the benefits of high-level knowledge from the graph, with the flexibility of pre-trained embeddings. They are general-purpose and applicable to a broader range of tasks beyond object classification, since we show that they can be adapted to language datasets as well. However, the ImageNet graph is specialized to an object-type hierarchy, and may not provide rich features as suitable for a wide range of downstream tasks in multiple domains.

We propose to learn class representations better suited for a wider range of tasks from common sense knowledge graphs. Common sense knowledge graphs [14, 28, 29, 30] are an untapped source of explicit high-level knowledge that requires little human effort to apply to a range of tasks. These graphs have explicit edges between related concept nodes and provide valuable information to

distinguish between different concepts. However, adapting existing zero-shot learning frameworks to learn class representations from common sense knowledge graphs is challenging in several ways. Graph Convolutional Network Zero-shot (GCNZ) [31] learns graph neural networks with a symmetrically normalized graph Laplacian and requires the entire graph structure during training, i.e., GCNZ is not inductive. Common sense knowledge graphs can be large (2 million to 21 million edges) and training a graph neural network on the entire graph can be prohibitively expensive. Dense Graph Propagation (DGP) [32] is an inductive method and aims to generate expressive class representations, but assumes a directed acyclic graph such as WordNet. Common sense knowledge graphs do not have a directed acyclic graph structure.

To address these limitations, we propose ZSL-KG, a framework with a novel transformer graph convolutional network (TrGCN) to learn class representations. Graph neural networks learn to represent the structure of graphs by aggregating information from each node’s neighbourhood.

In Figure 5, we present an overview of the ZSL-KG architecture. On the left, a sample from the 2-hop neighbourhood for the concept lawyer from ConceptNet. On the right: the ZSL-KG architecture. The text with the named entity is passed through the example encoder (biLSTM with attention) to generate the example representation. For the class representation, we take the k-hop neighbourhood and pass the nodes through their hop-specific transformer graph convolutional network (TrGCN). Next, we take the dot products between the example representations and the class representations to get compatibility scores. Finally, we train the example encoder and the class encoder by minimizing the task-specific loss over the true labels.

Aggregation techniques used in GCNZ, DGP, and most other graph neural network approaches are linear, in the sense that they take a (possibly weighted) mean or maximum of the neighbourhood features. To capture the complex information in the common sense knowledge graph, TrGCN learns a transformer-based aggregator to compute a non-linear combination of the node neighbours and increases the expressiveness of the class representation [33]. Our framework is also inductive, i.e., the graph neural network can be executed on graphs that are different from the training graph, which is necessary for generalized zero-shot learning under which the test classes are unknown during training.

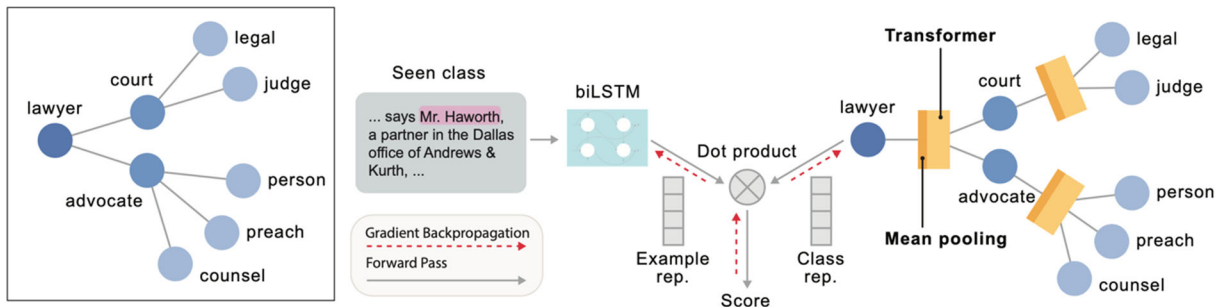


Figure 5. ZSL-KG’s Architecture

ZSL-KG achieves new state-of-the-art accuracies for zero-shot learning on the OntoNotes [34], BBN [35], attribute Pascal Yahoo (aPY) [21] and SNIPS-NLU [36] datasets. Finally, ZSL-KG also outperforms existing WordNet-based zero-shot learning frameworks on the ImageNet 22K dataset (all classes used for testing) [37], while maintaining competitive performance on the Animals with Attributes 2 (AWA2) dataset [38]. In Table 2, the results for generalized zero-shot fine-grained entity typing on OntoNotes and BBN. We report the average strict accuracy, loose micro F1, and loose macro F1 of the models on 5 random seeds and the standard error. In Table 3, the results for object classification on ImageNet dataset. We report the class-balanced top-k accuracy on zero-shot learning (ZSL) and generalized zero-shot learning (GZSL) for ImageNet classes k-hops away from the ILSVRC 2012 classes. The results for GCNZ, SGCN, and DGP are obtained from [32].

In Table 4, the results for generalized zero-shot object classification on the AWA2 and aPY datasets. We report the average class-balanced accuracy of the models for ZSL (T1) and GZSL (U, S, H) on 5 random seeds and the standard error.

Table 2. Performance of ZSL-KG for Generalized Zero-Shot Fine-Grained Entity Typing on OntoNotes and BBN

	<i>OntoNotes</i>			<i>BBN</i>			Overall
	Strict	Loose Mic.	Loose Mac.	Strict	Loose Mic.	Loose Mac.	Avg. Strict
GCNZ	41.46 $\pm$ 0.81	54.61 $\pm$ 0.52	61.65 $\pm$ 0.52	21.47 $\pm$ 1.22	47.17 $\pm$ 1.04	56.81 $\pm$ 0.68	31.47
SGCN	42.64 $\pm$ 0.69	56.25 $\pm$ 0.29	62.93 $\pm$ 0.39	24.91 $\pm$ 0.24	50.02 $\pm$ 0.73	59.68 $\pm$ 0.54	33.77
DGP	41.11 $\pm$ 0.78	54.08 $\pm$ 0.62	61.38 $\pm$ 0.76	23.99 $\pm$ 0.14	47.19 $\pm$ 0.51	57.62 $\pm$ 0.89	32.55
ZSL-KG	45.21 $\pm$ 0.36	55.81 $\pm$ 0.41	63.64 $\pm$ 0.51	26.69 $\pm$ 2.41	50.30 $\pm$ 1.21	57.66 $\pm$ 1.15	35.95

Table 3. Performance of ZSL-KG for Object Classification on ImageNet.

		Hit@k(%)														
		2-Hops					3-Hops					All				
		1	2	5	10	20	1	2	5	10	20	1	2	5	10	20
ZSL	GCNZ	19.8	33.3	53.2	65.4	74.6	4.1	7.5	14.2	20.2	27.7	1.8	3.3	6.3	9.1	12.7
	SGCN	26.2	40.4	60.2	71.9	81.0	6.0	10.4	18.9	27.2	36.9	2.8	4.9	9.1	13.5	19.3
	DGP	26.6	40.7	60.3	72.3	81.3	6.3	10.7	19.3	27.7	37.7	3.0	5.0	9.3	13.9	19.8
	ZSL-KG	26.3	40.6	60.3	71.9	81.2	6.3	11.1	20.1	28.8	38.8	3.0	5.3	9.9	14.8	21.0
GZSL	GCNZ	9.7	20.4	42.6	57.0	68.2	2.2	5.1	11.9	18.0	25.6	1.0	2.3	5.3	8.1	11.7
	SGCN	11.9	27.0	50.8	65.1	75.9	3.2	7.1	16.1	24.6	34.6	1.5	3.4	7.8	12.3	18.2
	DGP	10.3	26.4	50.3	65.2	76.0	2.9	7.1	16.1	24.9	35.1	1.4	3.4	7.9	12.6	18.7
	ZSL-KG	11.1	26.2	50.0	64.3	75.3	3.4	7.5	16.9	26.1	36.5	1.7	3.8	8.5	13.5	19.9

Table 4. Performance of ZSL-KG Generalized Zero-Shot Object Classification on the AWA2 and aPY Datasets.

	<i>AWA2</i>				<i>aPY</i>				Overall
	T1	U	S	H	T1	U	S	H	Avg. H
GCNZ	77.00 $\pm$ 2.05	66.62 $\pm$ 1.28	81.63 $\pm$ 0.18	73.30 $\pm$ 0.75	51.66 $\pm$ 0.74	49.11 $\pm$ 0.39	71.26 $\pm$ 0.20	58.14 $\pm$ 0.29	65.74
SGCN	77.10 $\pm$ 1.49	67.51 $\pm$ 1.29	81.16 $\pm$ 0.11	73.67 $\pm$ 0.74	52.32 $\pm$ 0.37	47.29 $\pm$ 0.63	71.05 $\pm$ 0.18	56.78 $\pm$ 0.47	65.23
DGP	77.10 $\pm$ 1.33	71.26 $\pm$ 1.02	79.39 $\pm$ 0.09	75.09 $\pm$ 0.61	49.73 $\pm$ 0.30	46.16 $\pm$ 0.66	70.16 $\pm$ 0.25	55.68 $\pm$ 0.52	65.38
ZSL-KG	78.08 $\pm$ 0.84	66.80 $\pm$ 0.70	84.42 $\pm$ 0.33	74.58 $\pm$ 0.53	60.54 $\pm$ 0.58	55.16 $\pm$ 0.50	69.66 $\pm$ 0.52	61.57 $\pm$ 0.45	68.07

3.1.4 Theoretical Guarantees for ZSL (TA2)

In ZSL, we want to recognize items of unseen classes, for which labeled data is not available. A ZSL model is trained on a disjoint set of similar classes, called seen classes, for which labeled data is available instead. The model is trained to map examples to auxiliary information describing the seen classes. Then, at test time, predictions can be made using only descriptions of the unseen classes. While ZSL is increasingly common in practice, from a theoretical perspective ZSL is a hard problem that defies analysis, because in the worst case there can be an arbitrary shift between the distributions of the seen and unseen classes. We take a step towards a better theoretical understanding of ZSL. We investigate the question: *Given only auxiliary information in the form of attributes describing unseen classes, what is the smallest worst-case error that any method can guarantee?* We provide the first non-trivial answer to this question by developing a framework based on adversarial optimization. We also show that this framework has practical application as a method for identifying when the predictions of ZSL methods on certain unseen classes are more likely to be incorrect.

We analyze ZSL with attributes by first observing that it is a two-stage process consisting of a training phase and an inference phase. In the training phase, we learn a map from the items to the attribute space using the seen classes, while in the inference phase we use the class-attribute matrix to infer the correct class given the item-attribute representation. Based on this two-stage decomposition, we can identify two kinds of errors. The first kind, related to the training phase, is due to domain shift. The map from items to the attribute space that is trained on the seen classes might not generalize accurately to the unseen classes. This contribution to the error can be arbitrarily large without introducing strong assumptions, as no labeled data is available for the unseen classes. Thus, it is impossible to characterize the domain shift between seen and unseen classes. The second kind, related to the inference phase, is due to the fact that the class-attribute matrix might not fully differentiate among the unseen classes. In particular, there can be an item of the unseen classes with a set of attributes that according to the class-attribute matrix relation conforms with the description of two different classes. The first kind of error, domain shift, has been extensively studied both in the theory and the experimental literature [39, 40, 41, 42, 43]. For the first time, we theoretically characterize the contribution of the second kind of error. It is important to understand and characterize this error for specific ZSL tasks because it corresponds to an inherent information gap in the problem setting that cannot be circumvented with a smarter algorithm.

We provide tight lower and upper bounds on the worst-case error of the best map from attribute representations to classes based on the class-attribute matrix. Our analysis gives a lower bound in the sense that it bounds from below the minimum error that any method can guarantee given only the information of the class-attribute matrix. The class-attribute matrix specifies the fraction of items in each class that exhibit each attribute. There is a range of class-attribute distributions that satisfy the constraints defined by a given matrix. We give a lower bound on the error of the best possible method for the worst-case distribution in that range. This distribution represents a worst-case correlation between attributes that satisfies the class-attribute matrix while maximizing the difficulty to distinguish between attribute-representation of items belonging to distinct classes. Our analysis also gives an upper bound in the sense that we show a randomized classifier that achieves at most the error of the lower bound, assuming perfect item-to-attribute mapping. This also shows

that the lower bound is tight. Interestingly, the value of the lower bound can also be interpreted as the quality of the information provided by the class-attribute matrix.

To summarize,

- Assume to have access to a perfect image-to-attributes mapping (otherwise additional error).
- Let $\mathbf{G}$ be the set of all the functions from binary-attributes to classes.
- Let $\mathbf{A}$ be the (probabilistic) class-attribute matrix for the classification task.

Then, the lower bound is

$$Q = \max_{p \in \mathcal{P}(\mathbf{A})} \min_{g \in \mathcal{G}} \varepsilon(g, p) \quad (1)$$

Equation 1. Lower Bound for Zero-Shot Learning with Attributes

Where the bound is the maximum among the errors of the best map from attributes to classes, for all possible joint distributions over {classes x attributes} that satisfy the constraints imposed by $\mathbf{A}$. The value Q represents a lower bound to the best error rate that an algorithm can guarantee.

We run extensive experiments comparing the theoretical results with the error of popular attribute-based ZSL methods, on benchmark datasets. We show that information given by the bound can be predictive of how standard methods behave, including which classes will likely be confused with others.

In the first set of experiments, we follow the standard way to evaluate ZSL models. We train our model on the seen classes, and then compare our lower bound with the empirical error of the ZSL models on the unseen classes. Since the computation of the lower bound is very expensive for a large number of attributes, we focus on a subset of them. We propose the following greedy strategy to ensure a selection of attributes that are informative with respect to the target classes. Starting with no attributes, we iteratively add the attribute that decreases the most the value of the lower bound, up to 15 attributes.

In the first row of Figure 6, we report results for aPY, Awa2, and CUB. We observe that the value of the lower bound (Q) can be significantly lower than the error rate of the ZSL models. This gap is most probably due to the fact that the learned map from images to attributes does not generalize perfectly to the unseen classes. In fact, in this setting we can identify two main sources of error for the ZSL models: (1) the arbitrary error due to the domain shift, and (2) the error due to how discriminative is the attribute space to differentiate between the different classes. Our lower bound only addresses the latter, as no method can guarantee a smaller error than the lower bound to map from attributes to classes given only the information of the class-attribute matrix. Nonetheless, for CUB we observe that the empirical error of ZSL models roughly follow the trend of the lower bound. This suggests that the lower bound can still be used as a tool to capture how the additional information provided by an attribute leads to improvements of the ZSL models.

In the second set of experiments, we empirically demonstrate our theory by showing that even if we minimize the error due to domain shift, there exists data for which no method can do better than our lower bound. To this end, for each dataset we adversarially generate synthetic data with attribute values satisfying the dataset’s class-attribute matrix. Specifically, we use the same class-

attribute matrix with 15 attributes as in the previous set of experiments in order to compute the adversarial distribution p over attributes and classes. The data is generated by sampling attribute-class pairs from this distribution and using the attribute vector as the feature vector. In order to minimize the error due to domain shift, this distribution is used to generate data for both training and testing of the ZSL methods, and the same class-attribute matrix is used for both seen and unseen classes.

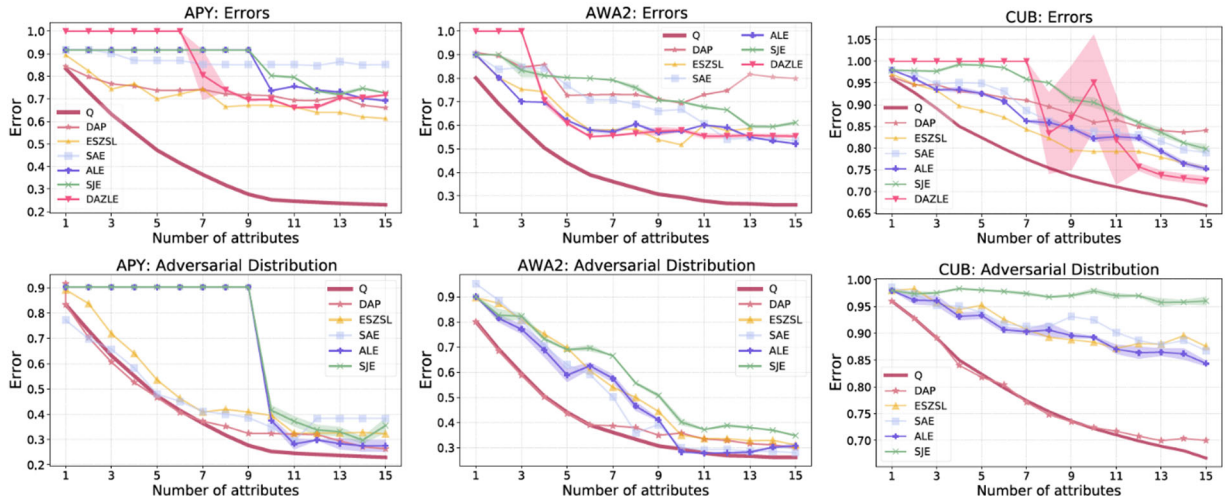


Figure 6. Comparison of the Zero-Shot Learning Lower Bound with the Empirical Error.

We report the results of the experiments in the second row of Figure 6, iterating over the same attributes greedily selected in the first set of experiments for each dataset. For this set of experiments, we do not report results for DAZLE as this method relies on the input items being images, so it does not apply to our synthetic data. In this case, the methods are able to achieve errors that are comparable with the lower bound as we minimized the error due to domain shift. This experiment empirically validates that even in the absence of domain shift, there exists a distribution of the data that satisfy the constraints imposed by the class-attribute matrix for which no method can do better than the lower bound. This adversarial distribution represents an intrinsic error gap due to the quality of the information provided by the class-attribute matrix. This is the first work to quantify such information in ZSL.

In Figure 7, we plot the pairwise lower bound error matrix L (lower bound on the error to distinguish between a pair of classes given the class-attribute matrix) together with the misclassification matrices M of two ZSL methods: Direct Attribute Prediction (DAP) and Embarrassingly Simple Zero-shot Learning (ESZL), computed on the unseen classes of aPY. Darker squares indicate higher values, and light blue on the diagonal is 0. High values of the lower bound indicate classes that are harder (in the worst-case) to distinguish in theory, and high values in M indicate pair of classes that are often confused by the ZSL model.

The pairwise lower bound matrix L has large values within multiple groups of semantically similar classes, e.g., animals and transportation means. This is in line with human intuition, as we expect

visually similar classes to exhibit similar attributes. Correspondingly, the misclassification matrices of DAP and ESZSL highlight the presence of many errors for classes belonging to these groups. We also note that ZSL models could misclassify other pairs of classes due to other source of errors, such as an inaccurate map from image to attributes.

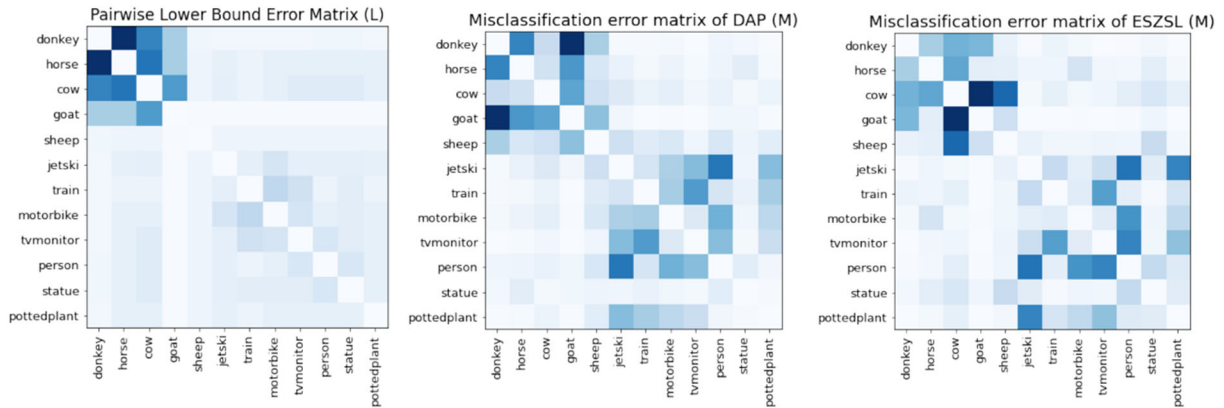


Figure 7. Pairwise Misclassification Matrices.

3.1.5 Aggregating weak labelers (TA1 and TA2)

The combination of the outcome of weak labelers has been studied while holding two restrictive assumptions: (1) the errors of the weak labelers are independent or come from a particular family of distributions, and (2) labelers output a single class label.

We provide theoretical analysis to remove the first assumption, and we deliver a new framework using partial labelers to remove the second.

Semi-Supervised Aggregation of Dependent Weak Supervision Sources With Performance Guarantees (TA2)

Weak supervision [44, 45, 46] exploits the information contained in weak classifiers for mildly related tasks, also referred to as labelers. The labelers generate noisy labels for abundant unlabeled data, which are then aggregated and used to train an end model for the classification task of interest. Empirically, this process yields better generalization than using the labelers directly for prediction.

The problem of aggregating the output of the labelers to obtain accurate labels for training is very challenging, as the estimation of statistical properties of the labelers is hard with few labeled data for the classification task of interest. This issue is typically mitigated by making strong modeling assumptions on the labelers. The assumption most commonly used is that the labelers make errors independently with respect to the classification task of interest [45]. If the independence assumption holds, many properties of the labelers, such as their error rates with respect to the classification task of interest, can be estimated with only unlabeled data. Further, taking a majority vote weighted by a function of each labeler's accuracy is the optimal way to combine their outputs to maximize accuracy [47].

While the independence assumption is reasonable in the crowdsourcing setting, as we are combining the answers of different people who are given the same classification task, it is problematic in the weak supervision setting. There is no reason to expect that the labelers are independent. In fact, labelers are themselves often learned with respect to different classification tasks, and it is likely that there are dependencies introduced based on the similarities of these tasks.

We introduce novel methods to compute analytical bounds on the worst-case error of the majority vote of any set of labelers. Based on this technique, we develop algorithms that select a subset of the labelers which minimizes this worst-case error. Our key insight is that by using few labeled data and abundant unlabeled data, we can reliably estimate properties of the labelers that are sufficient to bound this worst-case error without any prior assumption on the joint distribution of labelers and true labels. The labeled data is only used to estimate the labelers' individual accuracies, while the unlabeled data is used to estimate properties of the distribution of the output of the labelers (the pairwise labeler agreements).

Intuitively, labelers whose difference is small contain similar information, hence we would like to identify labelers that significantly differ from one another. It is important to point out that the difference between labelers can be estimated using only unlabeled data.

We can define a linear program (LP) that gives the worst-case error of majority vote.

$$\begin{aligned}
 \max_{S \in \mathcal{S}(\vec{\epsilon}, D)} \varepsilon(\lambda_{\mathcal{I}} \circ \ell_S) &= \max_{\vec{a} \in \{0,1\}^k: |\vec{a}|_1 < k/2} \sum_{\vec{a}} p_{\vec{a}} & (2) \\
 (a) \quad \sum_{\vec{a} \in \{0,1\}^k: a_j = 0} p_{\vec{a}} &= \epsilon_{i_j} \quad \text{for } j = 1, \dots, k \\
 (b) \quad \sum_{\vec{a} \in \{0,1\}^k: a_h \neq a_j} p_{\vec{a}} &= D_{i_h, i_j} \quad \text{for } h \neq j \\
 (c) \quad \sum_{\vec{a}} p_{\vec{a}} &= 1 \\
 (d) \quad p_{\vec{a}} &\geq 0 \quad \forall \vec{a}
 \end{aligned}$$

Equation 2. Lower Bound for Performance-Guaranteed Majority Vote (PGMV)

Given a set of labelers S , the correctness of the output of each labeler cannot be arbitrary as the labelers need to respect the errors and difference constraints. The strategy of the linear program is to find the values p_a , indicating the probability of the labeler's correctness, that maximize the error of the majority vote of the labelers, while satisfying the set of constraints.

The constraint (a) specifies that each labeler i must have error rate ϵ_i . The constraint (b) specifies that for any two different labelers their pairwise difference must be equal to $D(i, j)$. The constraints (c) and (d) impose that the variables p_a are probabilities.

We point out that the error rates are estimated from a few labeled instances and the pairwise difference is estimated from the unlabeled data.

By applying this technique to all the possible sets of labelers, we can choose the best subset with the minimal error. Computing the LP for each possible combination of labelers might become infeasible. However, we show in the experiments that we achieve high accuracy even when we restrict the search to the majority function on small subsets of combinations. A particularly efficient solution is considering only subsets of 3 labelers. In that case we have a closed form solution to the linear problem.

Proposition 5. *Let $\mathcal{I} = \{i, j, k\}$. Let q be equal to* (3)

$$q = \frac{1}{2} \max\{\epsilon_i + \epsilon_j - D_{i,k} - D_{j,k}, \epsilon_j + \epsilon_k - D_{i,j} - D_{i,k}, \\ \epsilon_i + \epsilon_k - D_{i,j} - D_{j,k}, 0\}$$

Then, we have that

$$\max_{S \in \mathcal{S}(\vec{\epsilon}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\ell}_S) = \epsilon_i + \epsilon_j + \epsilon_k \\ - \frac{D_{i,j} + D_{j,k} + D_{i,k}}{2} - 2q$$

Equation 3. Closed Form Solution to the Linear Problem

This result clearly shows the behavior of the worst-case majority vote depending on the error rates and the pairwise differences, and it reinforces our intuition that the labelers need to significantly differ from one another if we want to improve upon the best-performing labeler in the worst case.

We conduct experiments on 45 image classification tasks to test the effectiveness of our methods. Our experiments show that our algorithms match or outperform the standard semi-supervised model [48] that assumes independence. Our methods achieve as much as 24 percentage points higher accuracy on problems where labelers are inaccurate and there are only few labeled data. Our methods also match or outperform a recent method for labeler aggregation that does not make assumptions about the distribution of labelers' errors [49], which does not come with any theoretical guarantees.

In Figure 8, we plot the accuracy of different methods on the 9 hardest pairs of classes from Animals with Attributes. Dawid-Skene is a standard crowdsourcing/Snorkel model, which we make semi-supervised. Performance-Guaranteed Majority Vote (PGMV) et al. are our methods.

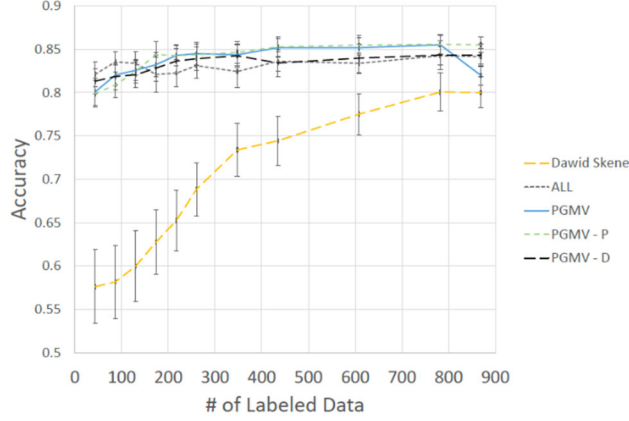


Figure 8. Accuracy of Different Aggregation Methods

Adversarial Multiclass Learning under Weak Supervision with Performance Guarantees (TA2)

The method presented above is inherently limited to hard binary classification, as it exploits the fact that when two labelers disagree, one must be correct. We introduce a novel method to use the information provided by a set of arbitrarily correlated weak supervision sources to learn a classifier for a given target task. Inspired by previous work, we use a small amount of labeled data to compute statistics of the weak supervision sources, and we formulate an optimization problem to find the prediction model that achieves the lowest empirical risk with respect to an adversarial choice of a labeling of an unlabeled dataset that agrees with those statistics. This is the first method with theoretical guarantees for learning multiclass classifiers from weak supervision sources without any prior assumptions on the joint distribution of their outputs and the true label.

Recent work has addressed the problem of combining weak labelers without distributional assumptions by taking an adversarial approach. For binary classification, Balsubramani & Freund [50] formulate the problem as minimax optimization, where the goal is to find the labels of an unlabeled dataset that minimize the error with respect to the worst-case assignment to the unknown ground-truth labels, while satisfying statistical constraints on the individual error of the weak labelers. This minimax problem can be optimally solved for a large family of loss functions [51]. The adversarial label learning (ALL) framework [49] uses a similar minimax optimization to learn a model that minimizes risk using the worst-case assignment to the unknown groundtruth labels, and was later extended to the multiclass setting [52], but it does not optimally solve the minimax optimization problem, and provides no generalization guarantees for the models it learns.

We address the limitations of previous work by providing a framework for multiclass classification with weak supervision, with rigorous computational efficiency and generalization error guarantees. Similar to ALL, we formulate the search for ground truth as a search over the set of feasible labelings that satisfy statistical constraints on the weak supervision sources. However, ALL lacks theoretical guarantees, and we show using techniques from convex optimization that our training algorithm rapidly converges to the optimal solution of the minimax optimization problem. Furthermore, we provide generalization bounds through uniform convergence theory for the learned model, in terms of the information provided by the weak supervision sources (with

respect to the target classification), geometrically represented as the diameter of the set of feasible labelings.

We demonstrate the applicability and performance of our method on image multiclass classification tasks derived from the DomainNet dataset [53].

We refer to our algorithms by using the acronym AMCL-CC to indicate an implementation of our method that uses a Convex Combination of the weak supervision sources as the prediction model. In Figure 9, we compare the accuracy of AMCL with the best weak supervision source (WSS), the majority vote predictions (MV), and a semi-supervised extension of the standard crowdsourcing algorithm (DS). For each method, we use 500 unlabeled data while the labeled data is increased.

We observe that our method matches or outperforms the baselines over all quantities of labeled data. We note that in the quickdraw domain, the weak supervision sources are overall very inaccurate, and it is difficult to recover useful information from them. However, unlike the baseline algorithms DS and MV, AMCL-CC can still recover and improve upon the best weak supervision source.

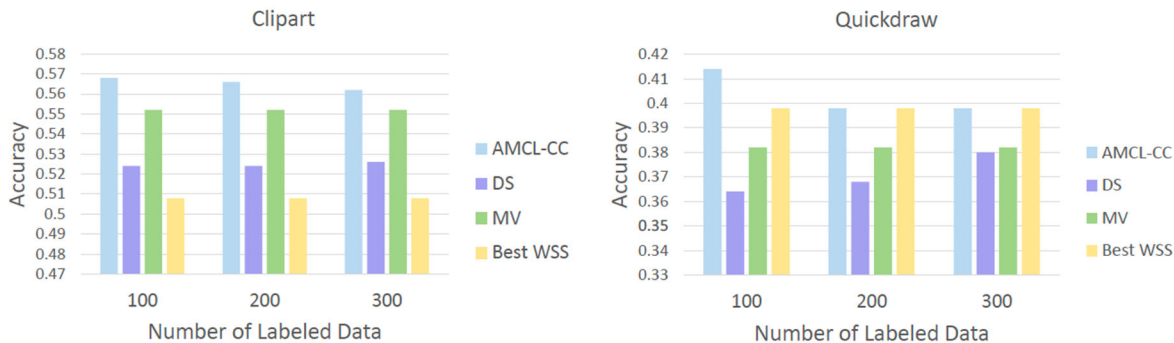


Figure 9. Experiments Results of AMCL on Domain Net

An Adaptive Method for Weak Supervision with Drifting Data (TA2)

Most of the works about weak supervision assume that the examples are independent and identically distributed (i.i.d.). This is a restrictive assumption when data is collected over time, and it is natural to observe a change, or drift, in the distribution of the examples. In our work, we relax the identically distributed assumption, and assume only that the examples are independent. This introduces a trade-off: if we want to obtain a good estimate at the current time, using more past examples provides more data, which might result in a better estimate if that data is similarly distributed, but might harm the estimate if the window includes a significant distribution drift.

We introduce an adaptive method with formal quality guarantees for weak supervision in a non-stationary setting. Our goal is to infer the unknown labels of a sequence of data by using weak supervision sources that provide independent noisy signals of the correct classification for each data point. This setting includes crowdsourcing and programmatic weak supervision. We focus on the non-stationary case, where the accuracy of the weak supervision sources can drift over time, e.g., because of changes in the underlying data distribution. Due to the drift, older data could

provide misleading information to infer the label of the current data point. Previous work relied on a priori assumptions on the magnitude of the drift to decide how much data to use from the past.

Our method selects the amount of data to use based on differences in the rates of agreement among the labelers. We derive a principled decision rule for this selection and provide a rigorous analysis that bounds the resulting error of the estimated accuracies of the labelers. Our novel bound separates the statistical error of estimating the parameters from the error caused by possible drift. This analysis enables the algorithm to select a close-to-optimal trade-off to minimize the worst-case error.

The conceptual difference between our approach and all previous work is that we do not rely on prior information about the drift or try to learn the drift from the data (both unrealistic in many applications). Instead, at each time step, our algorithm compares its estimation obtained using different window sizes and uses this information to detect drift and adjust the window size for the decision at that step. We analytically prove that this information is sufficient to allow the algorithm to efficiently adapt to drift in distribution, without explicitly estimating the magnitude of the drift.

Our estimation method is based on the technique developed by [54] that uses the weak labelers' correlation matrix to estimate the expertise of each weak labeler in a no-drift setting. When there is no distribution drift and under mild assumptions on the bias of the estimates of the weak supervision sources, it is possible to show that a good estimation of the correlation matrix implies a good estimation of the labelers' accuracies.

Given the empirical correlation matrix computed on the latest r samples, the guarantees on the estimation of the correlation matrix $C(t)$ are the following.

Lemma 1. *Let $t \geq 1$ and let $\delta \in (0, 1)$ and let Assumption 2 hold. The following inequality holds with probability at least $1 - \delta$:*

$$\|C(t) - \hat{C}^{[r]}(t)\|_{\infty} \leq \sqrt{\frac{2 \ln(n(n-1)/\delta)}{r}} + 12 \sum_{k=t-r+1}^{t-1} \|p(k) - p(k+1)\|_{\infty} \quad (4)$$

Equation 4. Guarantees on the Estimation of the Correlation Matrix

The goal of the algorithm is to increase the window size ending at time t as long as it does not include significant drift in distribution. As a reference for making this decision we observe that if the samples are identically distributed, we have that for any pair i, j , it holds:

$$D_1 = \dots = D_t \implies \mathbb{E} \left| \hat{C}_{ij}^{[r_{k+1}]} - \hat{C}_{ij}^{[r_k]} \right| \leq \sqrt{\frac{1}{r_k} - \frac{1}{r_{k+1}}} \quad (5)$$

Equation 5. Upper Bound for the Expected Difference of the Empirical Correlation Matrices for Identical Distributed Samples

The strategy of our algorithm is the following. Starting with $k = 1$, we iteratively compare the empirical covariance matrix computed respectively with r_k and r_{k+1} samples. If there is minimal drift, the difference of the correlation matrices should be smaller or comparable to the right-hand side of (5) for any entry i, j . If that is the case, we increase the value of k . If this empirical quantity is larger, then a significant drift must have occurred. In this case, we can stop and show that using r_k samples is provably close to optimal.

We demonstrate the functionality and the advantage of our algorithm over fixed-window-size strategies in two experimental settings. In our experimental evaluation, we show that on synthetic data and a drifting dataset constructed from Animals with Attributes2 [38], our algorithm adapts to the drift as it occurs, dynamically selecting the amount of data to use in an effective way. Unlike fixed-window-size strategies, we find this approach consistently maintains good performance as drift occurs. As the better window sizes change over time, the algorithm adapts by shifting to the better ones.

For synthetic data, in Figure 10, we plot the window size used by the adaptive algorithm and its estimates of the accuracies of each weak labeler in each time t , $1 \leq t \leq 4T$. The reported results are an average over 10 independent generations of this synthetic data. The main observation is that our algorithm correctly identifies a change in distribution and reduces the window size whenever it transitions to the next block. This allows for a very good estimation of the weak labeler accuracies, as the algorithm uses data mostly from the current block.

Clearly, there is a delay before our algorithm can correctly identify the distribution change, since it needs to collect enough data from the new block to assess that a significant change happened. As a result, the estimation of the weak labelers accuracy is worse for the data right after a block change.

The variation in the accuracy estimates in the middle of a block are due to the window size selection strategy of the algorithm: whenever the algorithm increases the window size, the larger window includes in the following few steps a small amount of samples from the previous block, resulting in a small additional error in the estimation.

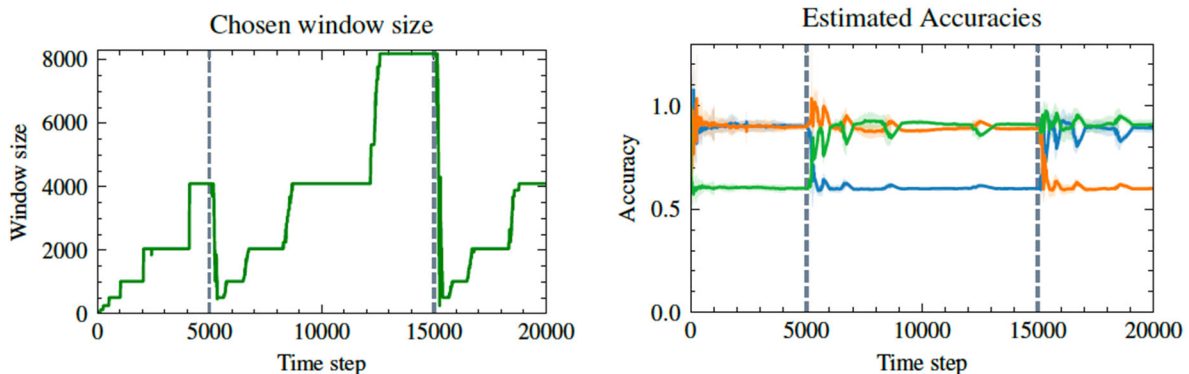


Figure 10. Algorithm's Window Sizes and Weak Labelers Accuracies

Learning from Multiple Noisy Partial Labelers (TA1)

In *programmatically weak supervision* (PWS), labeling functions, such as user-written rules and other heuristics, provide votes on the true labels for unlabeled examples or abstain from voting. Then, a label model, such as a probabilistic generative model or minimax game, is often used to estimate the true labels in a way that accounts for unknown differences in accuracies and other properties of the labeling functions. Finally, these estimated labels are used to train an end model on the unlabeled data to generalize beyond the information contained in the labeling functions.

All of the previous methods assume that labeling functions cast votes for individual classes. We propose to generalize PWS to support labeling functions that cast votes for a subset of classes, called partial labels. We refer to such labeling functions as *partial labeling functions* (PLFs). Our goal is to aggregate information from multiple partial labeling functions that are noisy (i.e., have imperfect accuracy) in order to estimate labels for unlabeled data.

Incorporating partial labels into PWS would enable users to take advantage of a wider range of domain knowledge. In typical PWS frameworks, only heuristics that are specific to one class can be incorporated. As a result, creating labeling functions requires careful task-specific engineering to avoid features that are shared by more than one class. For example, consider the task of classifying images of animals with a label from the set {HORSE, TIGER, LION, ZEBRA}. There are many useful heuristics that can be learned from other labeled data sets, such as detectors for claws or stripes [55]. Such heuristics divide the label space with multiple partitions. A claw detector could produce two partial labels: {TIGER, LION} if a claw is detected and {HORSE, ZEBRA} if not. Likewise, a stripe detector could output {TIGER, ZEBRA} if stripes are detected and {HORSE, LION} if not (Figure 11).

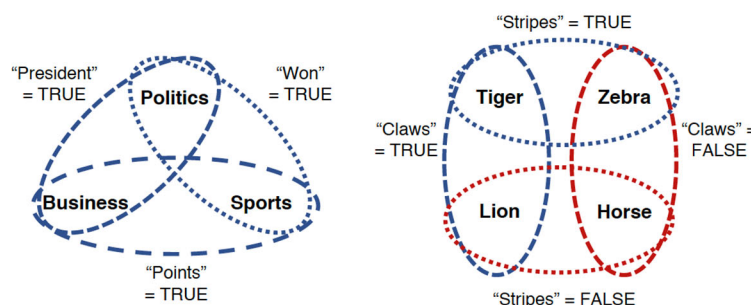


Figure 11. Examples of the Expressivity of Partial Labeling Functions

However, these heuristics cannot be used as labeling functions in current PWS frameworks. More generally, we observe a need for partial labeling functions in many multiclass applications where users want to express heuristics that narrow down the set of possible class labels but are not specific to a single class.

Learning from multiple noisy PLFs is challenging because we must resolve ambiguity arising from three sources: (1) PLF imprecision, i.e., voting for a set of classes instead of a single class, (2) PLF

inaccuracy, i.e., voting for a set of classes that does not contain the true class, and (3) conflict among multiple PLFs. A further requirement is that PWS frameworks should support labeling functions that abstain, meaning they can choose not to label certain examples. This is particularly critical for hand-engineered rules that might be highly specialized.

We propose a generalized PWS framework that supports partial labels and handles the additional ambiguity caused by the imprecise outputs of the PLFs. First, we introduce a probabilistic generative model that estimates the agreement between the outputs of each partial labeling function and the latent, true label. Second, we show how to learn the model parameters efficiently for large datasets. For example, we can learn on 100k examples with 10 PLFs in one minute. Since PWS is inherently human-in-the loop, fast iteration is crucial. Third, we prove that this model’s parameters are generically identifiable up to label swapping under mild conditions on the PLFs. This result means that we can estimate the accuracy of each partial labeling function without access to ground truth labels in a principled way. Using the learned parameters, we can compute the posterior distribution over true labels for each example. These probabilistic training labels can then be used to train an end model in the same manner as other PWS frameworks.

We demonstrate benefits of incorporating partial labels into PWS on applications in text and object classification. In Table 5, we evaluate methods for aggregating the outputs of the PLFs to train an end model. First, as a baseline, we consider using only the PLFs that are equivalent to traditional labeling functions, i.e., they always output one label or abstain. We call this baseline LFs Only. Second, as another baseline we use a heuristic called Nearest Class (NC), which chooses the class with the highest number of compatible partial labels. This baseline is a generalized majority vote heuristic for PLFs. Finally, our method, called Noisy Partial Label Model (NPLM), is our label model. Our proposed approach significantly improves accuracy over both baselines that (1) use only traditional labeling functions and (2) heuristically aggregate partial labels without a probabilistic model.

Table 5. Results for Text Classification Using NPLM

	SciCite		TREC-6		AG-News	
	ACC	F1	ACC	F1	ACC	F1
Supervised (Dev. Set)	78.5±1.1	75.5±1.5	88.3±1.6	76.2±2.0	80.1±1.7	79.9±1.9
LFs Only (w/o End)	65.1	44.7	22.0	23.8	33.0	25.1
NC (w/o End)	73.2	69.2	29.6	32.6	46.1	43.5
NPLM (w/o End)	71.5	69.4	38.2	43.0	51.1	49.4
LFs Only	78.6±0.7	76.7±0.8	67.2±0.9	69.3±1.0	79.8±0.9	79.6±0.8
NC	80.2±0.6	77.7±0.6	67.8±1.5	56.5±0.3	78.0±1.0	77.2±1.0
NPLM	81.4±1.3	79.5±1.3	85.0±0.8	85.7±0.7	85.0±0.5	84.7±0.5
NPLM vs. LFs Only	↑ 2.8	↑ 2.8	↑ 17.8	↑ 16.4	↑ 5.2	↑ 5.1
NPLM vs. NC	↑ 1.2	↑ 1.8	↑ 17.2	↑ 29.2	↑ 7.0	↑ 7.5

For the object classification task, we show how our framework can be used to model discrete visual attribute detectors, and that this approach can achieve results competitive with recent embedding

based ZSL methods. Although they have not been used often in recent ZSL work, discrete attribute detectors have benefits such as modularity and interpretability. These experiments show that modeling them as PLFs with our unsupervised label model can lead to good accuracy. In Table 6 we compare the accuracy of NPLM against NC and other attribute-based transductive zero-shot learning approaches. For Large-scale Attribute Datasets (LAD), we report mean accuracy (ACC) with 95% Confidence Intervals (CIs) across the five standard splits for each of the five subtasks. For AWA2, we report mean class accuracy (MCA) with 95% CIs. We evaluate AWA2 in a generalized setting: S and U denotes MCA on the seen and unseen classes respectively. H is the harmonic mean.

Table 6. Results for Object Classification

	LAD (ACC)						AwA2 (MCA)		
	Animals	Fruit	Vehicles	Electronics	Hairstyles	Avg.	<i>U</i>	<i>S</i>	<i>H</i>
ConSE—Norouzi et al. (2013)	36.9	29.8	37.5	28.3	24.6	31.4	0.5	90.6	1.0
ESZSL—Romera-Paredes and Torr (2015)	50.2	37.2	45.8	32.8	31.8	39.6	77.8	5.9	11.0
SynC—Changpinyo et al. (2016)	61.6	51.4	54.9	43.0	29.1	48.0	90.5	10.0	18.0
VCL—Wan et al. (2019)	75.4± 0.8	35.0± 1.0	62.4± 0.5	36.7± 0.5	33.8± 0.7	48.7± 0.3	21.4	89.6	34.6
QFSL—Song et al. (2018)	-	-	-	-	-	-	66.2	93.1	77.4
WDVSc—Wan et al. (2019)	97.2± 0.8	43.3± 1.3	82.1± 0.6	54.8± 1.1	31.1± 2.6	61.7± 0.6	76.4	88.1	81.8
NC (w/o End)	65.8	31.2	60.3	40.3	39.1	47.3	47.7	-	-
NPLM (w/o End)	86.0	38.7	73.5	51.8	45.9	59.2	68.2	-	-
NC	71.9±1.2	36.2±0.6	65.3±1.2	48.0±0.7	40.9±0.5	52.5±0.3	43.1±1.2	91.8±0.2	58.6± 1.1
NPLM	87.6± 0.2	42.4± 0.8	77.0± 0.2	57.7± 0.7	46.9± 0.9	62.3± 0.2	71.1±0.6	91.9±0.1	80.1±0.3
NPLM vs. NC	↑ 15.7	↑ 6.2	↑ 11.7	↑ 9.7	↑ 6.0	↑ 9.8	↑ 28.0	-	↑ 21.5

3.2 ADAPTATION OF PRE-TRAINED MODELS WITH LOW RESOURCES

3.2.1 Enhancing CLIP with limited-label prompt tuning (TA1)

Large pre-trained vision-language models (VLMs) [56, 57, 58] achieve remarkable accuracy without task-specific training but still require adaptation for optimal performance. Prompt-tuning [59, 60] is an approach to efficiently enhance VLMs performance on downstream tasks by learning inputs to the model. We investigate how the use of out-of-the-box pseudolabels assigned by CLIP can contribute to improving CLIP’s own performance in the semi-supervised, transductive zero-shot, and unsupervised learning settings.

Pseudolabels are heuristic labels assigned by a model to unlabeled data, which are leveraged to further train the model [61]. Successful training with pseudolabels relies on two factors: the quality of the labels and how they are used during training. To assign pseudolabels, we follow the method of Huang et al. proposing to select the most confident samples for each class [62], and we address the second point by exploring how to make the best use of them. We design a broad space of analysis considering three dimensions: prompt modalities, which are the model inputs we learn; learning paradigms, which define the data we have available; and training strategies, which describe the process used to optimize performance. In Figure 12, we summarize the design space where the green (dashed) path has already been explored [62], while the red (solid) lines are the unexplored combinations for prompt tuning.

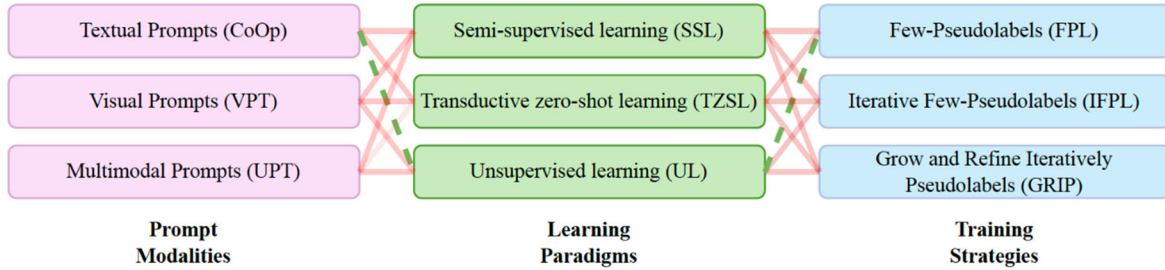


Figure 12. Our Design Space to Explore the Effect of Leveraging Pseudolabels in a Unified Way Across Prompt Modalities, Learning Paradigms, and Training Strategies.

Research on prompt tuning has demonstrated that training strategies used for learning prompts in one modality can be transferred to learning prompts in a different modality. On the contrary, different learning paradigms with limited labeled data typically require distinct approaches specifically tailored to extract information from the available data [63, 64]. However, we observe that this changes by using VLM’s generated pseudolabels. Unlike conventional pseudolabeling approaches that bootstrap off labeled data and are used as semi-supervised learning techniques [65, 17, 66], VLMs can generate pseudolabels in any learning setting. This offers a significant advantage, expanding the scope of pseudolabeling beyond semi-supervised learning, and making it a promising approach for other settings, such as transductive zero-shot and unsupervised learning. By using CLIP-based pseudolabels as a source of supervision, we can view these settings as optimizing the same loss function, which is simply a weighted sum of the errors on labeled data, if available, and pseudolabeled data. Given that we can express different settings as the same problem, we can propose training strategies, i.e., the way of using pseudolabels, that suit them all.

By standardizing the training strategies across various prompt modalities and learning settings, we can conduct experiments on different applications of pseudolabels for various combinations of prompt modalities, learning paradigms, and training strategies, as illustrated in Figure 12.

Rather than relying on a fixed set of pseudolabels, we propose GRIP an iterative training technique that allow for the ongoing refinement and expansion of the pool of pseudolabeled data used during training. We begin by obtaining the top-K pseudolabels for each target class. These pseudolabels are then used to train a new task-specific prompt. After completing the training, we use the acquired prompt to compute the top-K pseudolabels per class again. Subsequently, we reinitialize the prompt and repeat this entire process for a total of I iterations. In each iteration, we progressively increase the value of K . Specifically, during the i th iteration, we use $K = i/I$ -th of the unlabeled data to perform the steps in the iterative process. In this way, with each iteration, we progressively enhance CLIP’s pseudolabeling ability, allowing us to extend the set of pseudolabeled data while maintaining the high quality of the initial pseudolabels, which tend to be reliable.

In Table 7, reports the performance of GRIP using textual and visual prompts under different learning settings: semi-supervised learning (SSL), unsupervised learning (UL), and transductive zero-shot learning (TRZSL). We compare the accuracy of GRIP with CLIP zero-shot (ViT-B/32), CoOp, and Visual Prompt Tuning (VPT). We average the accuracy on 5 seeds and report the

standard deviation. Δ METHOD is the difference between the accuracy of GRIP and METHOD. We note that for UL we cannot apply CoOp and VPT since no labeled data is available.

In Table 8, we report the performance on the same learning paradigms, this time comparing GRIP to other training strategies using pseudolabels. Particularly, FPL learns prompts on K fixed pseudolabels, while IFPL works like GRIP with the difference that K is fixed to 16 shots. For each learning paradigm, we compare FPL, IFPL, and GRIP on Flowers102, RESICS45, and Describable Texture Dataset (DTD). We average across 5 runs and report the standard deviation. Δ METHOD is the difference between the accuracy of FPL and METHOD.

Table 7. Accuracy of GRIP Compared to CLIP Zero-Shot (ViT-B/32), CoOp, and VPT

Textual prompts									
Method	Flowers102			RESICS45			FGVCAircraft		
	SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL
CLIP	63.67 _{0.00}	-	63.40 _{0.00}	54.48 _{0.00}	-	54.46 _{0.00}	17.58 _{0.00}	-	17.86 _{0.00}
CoOp	76.76 _{1.11}	-	63.22 _{0.02}	58.53 _{10.81}	-	63.37 _{0.02}	14.91 _{3.22}	-	21.70 _{0.03}
GRIP	83.6 _{0.68}	69.84 _{1.06}	86.26 _{0.00}	74.11 _{0.68}	70.55 _{0.88}	81.07 _{0.00}	16.98 _{0.82}	15.22 _{0.71}	25.55 _{0.66}
Δ CLIP	$\uparrow 19.93$	$\uparrow 6.17$	$\uparrow 22.86$	$\uparrow 19.63$	$\uparrow 16.07$	$\uparrow 26.61$	$\downarrow 0.6$	$\downarrow 2.36$	$\uparrow 7.69$
Δ CoOp	$\uparrow 6.84$	-	$\uparrow 23.04$	$\uparrow 15.58$	-	$\uparrow 17.70$	$\uparrow 2.07$	-	$\uparrow 3.85$
Method	MNIST			EuroSAT			DTD		
	SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL
CLIP	25.10 _{0.00}	-	20.77 _{0.00}	32.88 _{0.00}	-	30.54 _{0.00}	43.24 _{0.00}	-	43.45 _{0.00}
CoOp	56.42 _{2.66}	-	21.15 _{0.09}	59.51 _{4.55}	-	49.68 _{0.08}	37.10 _{5.45}	-	46.3 _{0.03}
GRIP	72.58 _{2.16}	67.88 _{2.76}	74.06 _{0.00}	58.66 _{2.64}	57.21 _{1.77}	92.33 _{0.00}	56.07 _{0.85}	46.09 _{1.06}	65.30 _{0.01}
Δ CLIP	$\uparrow 47.48$	$\uparrow 42.78$	$\uparrow 53.29$	$\uparrow 25.78$	$\uparrow 24.33$	$\uparrow 61.79$	$\uparrow 12.83$	$\uparrow 2.85$	$\uparrow 21.85$
Δ CoOp	$\uparrow 16.16$	-	$\uparrow 52.91$	$\downarrow 0.85$	-	$\uparrow 42.65$	$\uparrow 18.97$	-	$\uparrow 19.00$
Visual prompts									
Method	Flowers102			RESICS45			FGVCAircraft		
	SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL
CLIP	63.67 _{0.00}	-	63.40 _{0.00}	54.48 _{0.00}	-	54.46 _{0.00}	17.58 _{0.00}	-	17.86 _{0.00}
VPT	63.73 _{1.52}	-	64.71 _{0.00}	60.80 _{1.65}	-	67.06 _{0.00}	17.76 _{0.68}	-	26.69 _{0.00}
GRIP	67.95 _{1.2}	63.09 _{0.55}	77.18 _{0.00}	71.22 _{0.77}	68.43 _{0.61}	82.19 _{0.00}	19.43 _{0.5}	17.51 _{0.61}	26.30 _{0.36}
Δ CLIP	$\uparrow 4.28$	$\downarrow 0.58$	$\uparrow 13.78$	$\uparrow 16.74$	$\uparrow 13.95$	$\uparrow 27.73$	$\uparrow 1.85$	$\downarrow 0.07$	$\uparrow 8.56$
Δ VPT	$\uparrow 4.22$	-	$\uparrow 12.47$	$\uparrow 10.42$	-	$\uparrow 15.13$	$\uparrow 1.67$	-	$\downarrow 0.27$
Method	MNIST			EuroSAT			DTD		
	SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL
CLIP	25.10 _{0.00}	-	20.77 _{0.00}	32.88 _{0.00}	-	30.54 _{0.00}	43.24 _{0.00}	-	43.45 _{0.00}
VPT	42.53 _{14.13}	-	25.51 _{0.05}	47.13 _{1.34}	-	62.24 _{0.02}	36.41 _{2.17}	-	44.16 _{0.01}
GRIP	69.44 _{4.35}	68.04 _{1.11}	69.54 _{0.01}	63.48 _{3.09}	63.68 _{3.42}	96.97 _{0.00}	54.57 _{4.86}	50.51 _{0.99}	62.78 _{0.00}
Δ CLIP	$\uparrow 44.34$	$\uparrow 42.94$	$\uparrow 48.77$	$\uparrow 30.60$	$\uparrow 30.80$	$\uparrow 66.43$	$\uparrow 11.33$	$\uparrow 7.27$	$\uparrow 19.33$
Δ VPT	$\uparrow 26.92$	-	$\uparrow 44.03$	$\uparrow 16.35$	-	$\uparrow 34.73$	$\uparrow 18.16$	-	$\uparrow 18.62$

Table 8. Compared Accuracies of FPL, IFPL, and GRIP

Textual prompts									
Method	Flowers102			RESICS45			DTD		
	SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL
FPL	75.96 _{0.74}	65.67 _{0.23}	80.97 _{0.00}	68.13 _{0.55}	63.07 _{0.38}	72.11 _{0.00}	37.10 _{5.45}	44.96 _{0.55}	46.30 _{0.03}
IFPL	78.68 _{0.75}	69.56 _{1.05}	82.08 _{0.00}	70.52 _{1.12}	64.11 _{0.98}	75.51 _{0.00}	55.24 _{0.97}	47.77 _{1.15}	59.14 _{0.02}
GRIP	83.60 _{0.48}	69.84 _{1.06}	86.26 _{0.00}	74.11 _{0.68}	70.55 _{0.88}	81.07 _{0.00}	56.07 _{0.85}	46.09 _{1.06}	65.30 _{0.01}
Δ IFPL	↑ 2.72	↑ 3.89	↑ 1.11	↑ 2.39	↑ 1.07	↑ 3.4	↑ 18.14	↑ 2.81	↑ 12.84
Δ GRIP	↑ 7.64	↑ 4.17	↑ 5.29	↑ 5.89	↑ 7.48	↑ 8.96	↑ 18.97	↑ 1.13	↑ 19.00
Visual prompts									
FPL	67.03 _{0.65}	65.50 _{0.41}	71.94 _{0.00}	65.14 _{0.25}	62.24 _{0.22}	67.85 _{0.00}	47.60 _{1.09}	47.69 _{0.48}	52.43 _{0.00}
IFPL	68.69 _{0.45}	66.12 _{0.46}	76.91 _{0.00}	67.11 _{1.19}	62.93 _{1.23}	73.53 _{0.00}	51.65 _{0.70}	50.34 _{0.65}	57.86 _{0.01}
GRIP	67.95 _{1.2}	63.09 _{0.56}	77.18 _{0.00}	71.22 _{0.77}	68.43 _{0.61}	82.19 _{0.00}	54.57 _{4.86}	50.51 _{0.99}	62.78 _{0.00}
Δ IFPL	↑ 1.66	↓ 0.38	↑ 4.97	↑ 1.97	↑ 0.69	↑ 5.68	↑ 4.05	↑ 2.65	↑ 5.43
Δ GRIP	↑ 0.92	↓ 3.41	↑ 5.24	↑ 6.08	↑ 6.19	↑ 14.34	↑ 6.97	↑ 2.82	↑ 10.35
Multimodal prompts									
FPL	72.54 _{0.36}	65.26 _{0.38}	77.47 _{0.00}	62.84 _{1.05}	62.32 _{0.65}	71.43 _{0.00}	43.71 _{2.19}	44.85 _{0.31}	54.86 _{0.00}
IFPL	73.14 _{0.87}	65.39 _{1.33}	81.47 _{0.00}	70.60 _{1.04}	63.69 _{0.53}	46.04 _{0.36}	53.21 _{1.24}	47.59 _{1.04}	43.17 _{0.25}
GRIP	74.56 _{2.02}	64.82 _{1.63}	82.01 _{0.00}	73.78 _{0.91}	69.37 _{0.61}	82.17 _{0.00}	54.07 _{2.25}	47.37 _{0.70}	63.42 _{0.00}
Δ IFPL	↑ 1.91	↑ 0.13	↑ 4.00	↑ 7.76	↑ 1.37	↓ 25.39	↑ 9.5	↑ 2.74	↓ 11.69
Δ GRIP	↑ 2.02	↓ 0.44	↑ 4.54	↑ 10.84	↑ 7.05	↑ 10.74	↑ 10.36	↑ 2.52	↑ 8.56

We note that GRIP is the same technique we used to solve JPL’s zero-shot learning task.

Beyond quantitative insights, the exploration of this design space leads to interesting qualitative take aways.

- **There is a trade-off between quality and quantity of pseudolabels.** Figure 13 shows the progression of pseudolabels quality for the iterative learning of textual prompts (the rows refer to SSL, UL, and TRZSL, in order. IFPL refers to the top x-axis, while CLIP and GRIP to the bottom). IFPL maintains a fixed set of 16 pseudolabels, improving their quality with each iteration. On the other hand, GRIP expands pseudolabels by incorporating an additional decile of unlabeled data in each iteration. Initially, GRIP maintains accuracy, but as it nears completion the quality tends to decrease, while a larger dataset with good-quality pseudolabels becomes available. Comparing GRIP and CLIP, GRIP’s expanded pseudolabels exhibit superior quality and performs better (Table 7). Even though IFPL’s pseudolabel accuracy surpasses GRIP in the final iteration, GRIP’s overall performance remains better due to training on a larger number of pseudolabels (Table 8). This suggests that numerous, slightly noisier pseudolabels can yield better results, highlighting a trade-off and offering insights for future approaches.

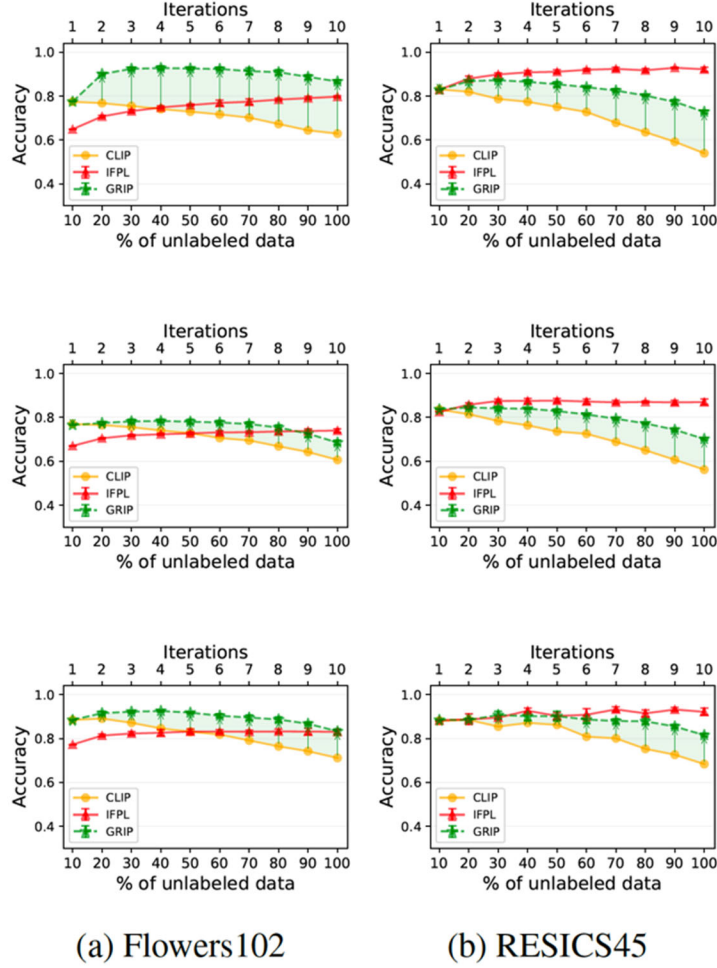


Figure 13. Evolution of Pseudolabels Accuracy During Training

- The Robin Hood effect.** Although training models with pseudolabels can lead to good performance, it can also result in biased predictions and generate disparate impacts on sub-populations, i.e., the “Matthew effect” [67, 68]. Particularly, the use of pseudolabels can lead to improved performance in well-behaved (high accuracy) classes but can cause stagnation or decreased performance in poorly behaved (low accuracy) classes. As we explore the use of pseudolabels, we investigate how the accuracy of the analyzed approaches distributes across classes. Figure 14, where X-axis is the ranked class index, while y-axis is the accuracy, shows an opposite scenario from typical SSL. For all learning paradigms, the iterative training strategies increase the accuracy of classes where CLIP is not proficient, while maintaining or decreasing the accuracy of initially well-behaved classes. This effect, we call the “Robin Hood effect,” is very interesting because it shows how CLIP can mitigate its own bias toward certain classes by learning from itself.

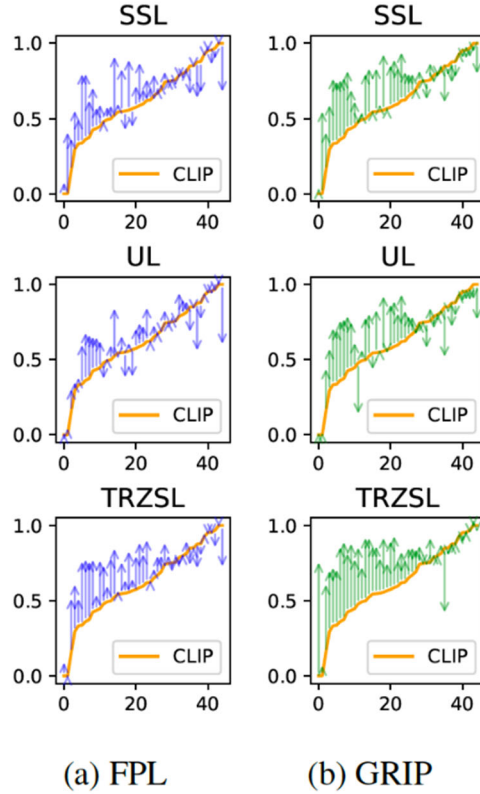


Figure 14. Improvements of FPL and GRIP on CLIP’s Per-Class Accuracies (RESICS45)

3.2.2 Compositionality of CLIP (TA1)

To improve VLMs for compositional zero-shot learning, we introduce compositional soft prompting (CSP), a parameter-efficient learning technique that tunes tokens of vocabulary to represent primitive concepts in a composable way.

Fine-tuning large pre-trained models such as CLIP [69] requires huge amounts of compute and may lead to overfitting [70, 71]. This challenge has motivated several soft prompting techniques in both language and vision [72, 73, 74, 75]. These works tune a single prompt on a downstream supervised task, often in a few-shot setting. For instance, they typically use prompts such as “A photo of [class]” and tune the prefix “A photo of” on the entire dataset.

In contrast, CSP is a novel way of soft prompting. We treat the attributes and objects that are composed to define classes as learnable tokens of vocabulary in a prompt as “A photo of [attribute] [object]”. We tune on multiple [attribute] and [object] prompt compositions, and then we recompose them into new prompts for zero-shot inference (Figure 15).

The goal of CSP is to improve VLMs such as CLIP on compositional generalization where they seem to underperform the current state-of-the-art methods. This is perhaps because CLIP’s pretraining on data crawled from the web does not provide sufficient supervision about attributes and how they combine with different objects. Therefore, we aim to teach VLMs such as CLIP how

to better compose primitive concepts. We approach this as a vocabulary-learning problem because it is parameter efficient and gives us a natural way to compose new classes.

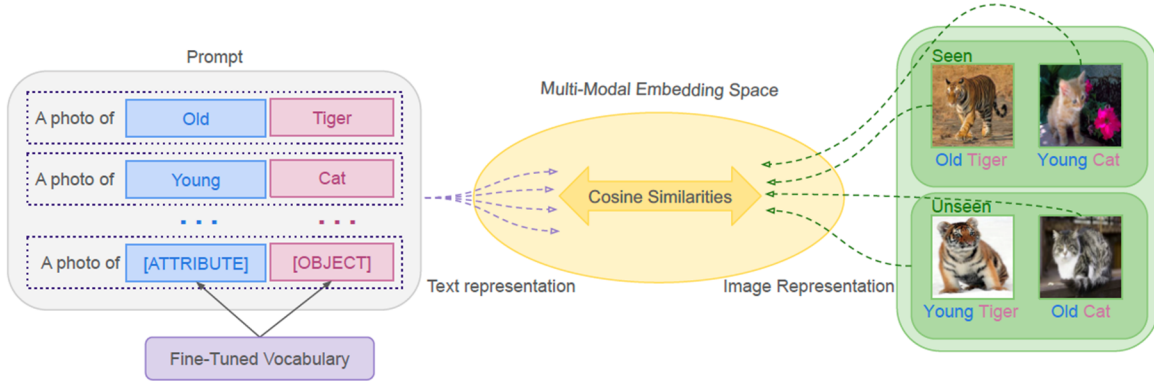


Figure 15. An Overview of Compositional Zero-Shot Learning with CSP

CSP treats the attributes and objects that are composed to define classes as learnable tokens of vocabulary and tunes them on multiple prompt compositions. We represent each primitive concept, either attribute or object, as a new, auxiliary token in the VLM’s vocabulary. To represent a class, we combine a fixed prefix and the corresponding primitive concepts, for example “A photo of young tiger”, where young maps to the auxiliary weight vector for the corresponding attribute and tiger maps to the auxiliary weight vector for the corresponding object. Then, we use these prompts for compositional zero-shot learning in the same way as we would for CLIP. Training and inference with CSP is very simple as we only need to swap the vocabulary of the attributes and objects for any desired composition in the prompt.

Our results show that CSP improves over the zero-shot performance of CLIP. CSP significantly improves over CLIP across three benchmark datasets by an average accuracy of 13.7 percentage points in the closed-world setting and 8.0 percentage points in the open-world setting (using the AUC metric). CSP also outperforms CoOp [76], a soft prompting method that tunes the prefix context, by an average of 7.3 percentage points in the closed-world setting and 4.3 percentage points in the open-world setting on the AUC metric (see Table 9).

Table 9. Closed-world (Closed) and Open-world (Open) Results on MIT-States, UT-Zappos, and C-GQA.

	Method	MIT-States				UT-Zappos				C-GQA			
		S	U	H	AUC	S	U	H	AUC	S	U	H	AUC
Closed	CLIP	30.2	46.0	26.1	11.0	15.8	49.1	15.6	5.0	7.5	25.0	8.6	1.4
	CoOp	34.4 _{0.1}	47.6 _{0.1}	29.8 _{0.1}	13.5 _{0.0}	52.1 _{0.5}	49.3 _{1.8}	34.6 _{1.7}	18.8 _{1.4}	20.5 _{0.2}	26.8 _{0.3}	17.1 _{0.2}	4.4 _{0.1}
	CSP	46.6 _{0.1}	49.9 _{0.1}	36.3 _{0.1}	19.4 _{0.1}	64.2 _{0.7}	66.2 _{1.2}	46.6 _{1.2}	33.0 _{1.3}	28.8 _{0.1}	26.8 _{0.1}	20.5 _{0.1}	6.2 _{0.0}
Open	CLIP	30.1	14.3	12.8	3.0	15.7	20.6	11.2	2.2	7.5	4.6	4.0	0.27
	CoOp	34.6 _{0.1}	9.3 _{0.0}	12.3 _{0.1}	2.8 _{0.0}	52.1 _{0.5}	31.5 _{2.9}	28.9 _{2.3}	13.2 _{1.6}	21.0 _{0.2}	4.6 _{0.1}	5.5 _{0.1}	0.70 _{0.0}
	CSP	46.3 _{0.3}	15.7 _{0.1}	17.4 _{0.1}	5.7 _{0.0}	64.1 _{0.7}	44.1 _{0.3}	38.9 _{0.5}	22.7 _{0.4}	28.7 _{0.2}	5.2 _{0.1}	6.9 _{0.1}	1.20 _{0.0}

In addition to improved benchmark accuracy, CSP has several other advantages when tested on other kinds of zero-shot inferences without any changes to training. We show that the learned

attribute vocabulary can be decomposed to better classify attributes in isolation, using prompts of the form “A photo of [attribute] object”. We also show that training CSP with attribute-object compositions improves CLIP’s performance on attribute-attribute-object compositions.

To test if training CSP with attribute-object compositions can generalize to higher-order compositions, such as attribute-attribute object compositions, we annotate a novel challenge dataset: AAO-MIT-States, a subset derived from the MIT-State dataset. In this dataset, we annotate the images in the test split of the MIT-States dataset with an additional attribute, to get an attribute-attribute-object pair as the class label. We compare CLIP, CoOp, and CSP to classify images with attribute-attribute-object classes.

Table 10 shows that CSP improves over CLIP by an average 9.9 percentage points on unseen accuracy and generalizes to attribute-attribute-object compositions without any modifications or training. The results demonstrate that CSP improves the compositionality of CLIP’s vocabulary, even in ways that were not explicitly supervised.

Table 10. Generalization to High-Order Composition

Method	Accuracy
CLIP	62.7
CoOp	65.2 _{0.3}
CSP	72.6_{0.4}

Does CLIP Bind Concepts? Probing Compositionality in Large Image Models (TA1)

Good semantic representations are generally assumed to require, at a minimum, compositionality and groundedness. That is, meanings of sentences should be functions of the words they contain and the syntax via which those words are combined [77] (compositionality), and such meanings should be at least in part responsible for reference to the real world, e.g., via truth conditions (groundedness).

We focus on vision-and-language models (specifically CLIP [69] and fine-tuned variants of CLIP, like CSP), and seek to answer, in a controlled setting, whether such models meet basic tests of grounded compositionality. Specifically, we consider three basic types of linguistic compositions: combining a single adjective and noun (*red cube*), combining two adjectives with respective nouns (*red cube and blue sphere*), and relating two nouns (*cube behind sphere*). All three of these settings (see Figure 16) require some degree of compositionality and groundedness, with the latter two exemplifying a more abstract type of compositionality (pervasive in language) which depends not only on recognizing a conjunction of constituents but an ability to bind meaning representations to abstract syntactic roles.

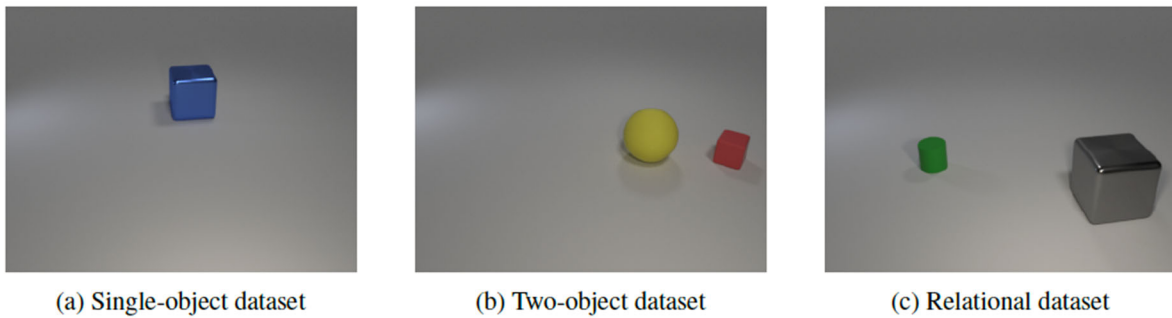


Figure 16. Example Images from Each Dataset

Overall, we see that, on single adjective-noun compositions (red cube), CLIP performs better than any of the more explicitly *compositional distributional semantics models* (CDSMs) [78, 79, 80, 81, 82]. In the other settings, which rely on the ability to bind variables, we see that using CDSMs for the text encoder sometimes improves performance, but not always, and that, across all models, performance is essentially at chance in the best case. These results suggest that CLIP’s representation of the visual world is poorly suited for compositional semantics, and suggest that future work on improving these representations is a necessary next step in advancing work on grounded compositional distributional semantics.

3.2.3 Alfred (TA1)

Programmatic weak supervision (PWS) provides a more efficient method of data annotation by using noisy heuristics to label data. In a typical PWS setup, domain experts design labeling functions (LFs) as programs that vote for a label or abstain [45, 44]. Recently, there has been a growing interest in creating LFs from large, pre-trained models through prompting [83, 84, 85]. In the shift to this new setting, executing LFs goes from the least to the most computationally expensive part of the process, highlighting the importance of providing a software infrastructure that facilitates efficient development. Prompted models offer a unique opportunity to enhance existing PWS systems, as they allow the easy expression of complex rules that were previously difficult to specify using code.

We look at a couple of examples in Figure 17. For the first example, each expresses supervision relating mentions of President Biden to the category of politics. Instead of specifying an intricate regular expression, a prompted labeling function uses the prompt “Text: [[instance]] Does this text mention President Biden?” where [[instance]] is replaced by the news article to be labeled. The response is mapped to a vote on the true label.

The second example demonstrates how heuristics about positive sentiment that were previously hard to define can be flexibly expressed as a natural language question. Instead of defining a set of keywords for fuzzy sentiment matching, we can simply ask large pretrained models for answers about the sentiment. For the third example, we consider an animal labeling task where we use the visual attributes “stripes” to vote for the classes TIGER and ZEBRA. Previously, we would need to first collect supervised training data for attributes like stripes and then train classifiers to make

the decisions. With modern vision-language models (e.g. CLIP), we can simply express the attribute detection task as a set of candidate prompts

The ability to use prompts to define labeling functions simplifies and streamlines the weak supervision process, as well as potentially elevating the quality of the annotations. This benefit is particularly helpful for tasks involving computer vision, where previously PWS has been limited to tasks for which models can identify key features (such as objects) over which to program rules. Whether the domain is language or multi-modal, prompts let users experiment with different heuristics (and phrasings of those heuristics). Therefore, enabling an iterative development experience is essential for the success of a prompt based PWS system.

Labeling Function	Prompted Labeling Function
<pre>def biden_mention(instance): if re.match(r"(President\s)?(Jo(seph) e)\s)?(Biden)/i", instance): return POLITICS else: return ABSTENTION</pre>	"Text: [[instance]] Does this text mention President Biden?" "Yes" → POLITICS "No" → ABSTENTION
<pre>def positive_sentiment(instance): for token in tokenize(instance): if token in positive_words: return POSITIVE return ABSTENTION</pre>	"Text: [[instance]] Does this comment express positive sentiment?" "Yes" → POSITIVE "No" → ABSTENTION
<pre>def stripe_detect(instance): if stripe_detector.infer(instance) >= 0.5: return [TIGER, ZEBRA] else: return ABSTENTION</pre>	instance "A photo of an animal with stripes", "a photo of an animal" "A photo of an animal with stripes" → [TIGER, ZEBRA] "A photo of an animal" → ABSTENTION

Figure 17. Examples of a Labeling Function Versus a Prompted Labeling Function

We developed Alfred, a versatile PWS system that leverages large pre-trained models for image and text annotations. Alfred aims to provide an environment for the rapid development of prompt-based supervision, while maintaining a consistent development experience like established PWS systems. We designed Alfred with usability and efficiency in mind, aiming to provide a rapid and smooth experience for developing prompt-based supervision.

Alfred supports popular large language models from Hugging Face’s transformer package [86], including the Generative Pre-Trained Transformer (GPT) family [87], the T5 family [88], etc., and vision-language models like CLIP [69], etc. Alfred also supports local Open Neural Network Exchange (ONNX) models, or API-based models from OpenAI, AI21, and Cohere. Moreover, Alfred provides easy templating tools to help users quickly create, evaluate, and refine prompted LFs. Alfred offers easy ways to deploy inference servers remotely, in addition to local model hosting. Alfred also optimizes model inference throughput with improved batching techniques and provides utilities for efficient Large-Language Models (LLM) deployment and interaction. Finally, Alfred contains a library of label models to distill the outputs of prompted labeling functions into the final training datasets for downstream end models.

Alfred is a prototype for a second generation of PWS systems with prompting at their core. To this end, we highlight three key features of Alfred:

- **Prompt-based weak supervision for images and text.** Alfred provides the necessary tools for users to create, evaluate, and refine prompt templates for both image and text weak supervision tasks. The inclusion of a query caching system that automatically stores and updates model responses facilitates development.
- **Optimized inference throughput.** Alfred implements a dynamic batching mechanism that optimizes large sets of prompts. This feature allows models hosted by Alfred to achieve 2-3× greater throughput than naive implementations.
- **Seamless local development experience.** Alfred can host models remotely and make them accessible to developers via gRPC, a high-throughput protocol for remote procedure calls. It enables sending datasets to servers in large chunks to maintain higher throughput. Alfred also implements an SSH-based port-forwarding utility for the gRPC connection, easing deployment on shared clusters such as those at universities.

We illustrate a typical workflow for using Alfred to create a training dataset in Figure 18. First, developers use Alfred to iteratively design, evaluate, and refine their prompted labeling functions (LFs). They use prompt Templates to generate Queries to Models based on data. The responses of Models are mapped to votes on the true labels for examples by Voters, which can be calibrated. Then the included Label Models combine the noisy votes to produce probabilistic training labels for an end model. With Alfred users can quickly create, evaluate, and refine their prompt-based weak supervision sources; map the results to weak labels; and resolve their disagreements with a label model.

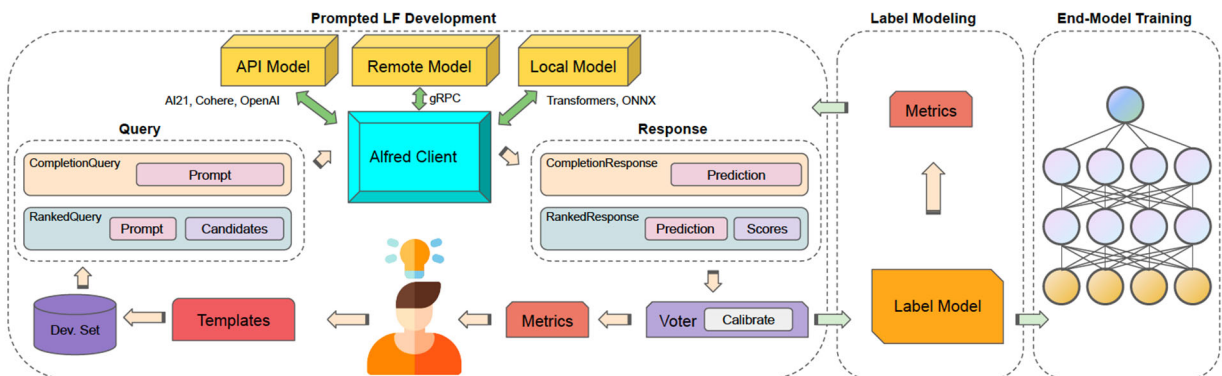


Figure 18. A Typical Workflow for Programmatic Weak Supervision with Alfred

Alfred is available at <https://github.com/BatsResearch/alfred>. We refer to the README file for a quick overview of-how to setup and use the framework.

We run two example use cases for how Alfred can be used to create training data for specific machine learning tasks using natural language prompts in both text (YouTube Comment Spam Detection) and image (Pet breed classification – OxfordPet) domains. We measure the labeling quality by taking the top-1 accuracy of the estimated probabilistic labels given by the label model. Table 11 and 12 show results.

Table 11. Top-1 Accuracy on YouTube Spam Detection

Zero Shot	Prompted LFs	Prompted LFs+C
46.8	57.8	65.3

Table 12. Top-1 Accuracy on Oxford-IIIT Pet Breed Classification

Zero Shot	Prompted LFs
86.0	92.4

4. RESULTS AND DISCUSSIONS

The development and evaluation of TAGLETS demonstrated that when task-related labeled data is limited

- Knowledge graphs are an efficient way to manage and retrieve a subset of data relevant to a target task.
- In low-shot settings, task-specific selected auxiliary data serve their purpose of improving performances even when all the auxiliary data is seen in the pre-training. As the training data for large pre-trained models increases, and we lose control over its composition, this observation is particularly important to keep assessing the validity of our research.
- Distilling models trained with different learning approaches to a single end model improves accuracy and makes the final model servable.

TAGLETS has emerged as a standout performer, ranking within the top 6 contenders in the program for image classification tasks. Its achievement demonstrates its prowess in leveraging both established labeled datasets and a multitude of weak labelers to achieve remarkable classification accuracy. The system's evolution has been significantly influenced by comprehensive theoretical explorations into effectively aggregating these interdependent weak labelers within the system.

- The behavior of the worst-case majority vote depends on the error rate and the pairwise differences among weak labelers. We show that to improve upon the best-performing labeler in the worst case, the labelers need to significantly differ from one another.
- We show ways of removing the (mostly unrealistic) independent assumption and relying on statistics computed from few labeled instances (errors) and unlabeled data (pairwise differences) to compute the bounds on binary and multiclass learning.
- We proposed an adaptive algorithm for programmatic weak supervision in the presence of drift and demonstrated that, based on the differences in the rates of agreements among the labelers during time, we can derive a principled decision rule to select the data to consider for bounding the error of the estimated labelers' accuracies.
- Extend the paradigm of weak supervision to partial labelers, which return more than a label, and demonstrate that it can improve over standard labeling functions in text classification and attribute-based object classification.

We substantially contributed to the area of zero-shot learning

- We demonstrated the possibility of describing classes starting from common sense knowledge graphs, and that employing them for zero-shot learning for image tasks is effective. Additionally, we demonstrate that such approach works also for language tasks.
- We closed the gap in the theory of zero-shot learning by providing the first non-trivial lower bound for zero-shot learning with attributes. Particularly, given the class-attribute matrices, describing the probability an attribute appears in a class, of seen and unseen classes, we can formulate a lower-bound on the error.

By studying how to enhance CLIP in low-resource settings we observed that

- General purpose zero-shot learners used as general purpose pseudolabelers open the opportunity to develop training strategies that leverage pseudolabeled data beyond semi-supervised learning. We point out that different learning paradigms, such as semi-supervised, transductive zero-shot, and unsupervised learning, can be all considered as special cases of a single objective function, by using pseudolabels as a source of supervision.
- Simple iterative training strategies for refining pseudolabels are highly effective approaches for limited-label prompt tuning. In fact, regardless of the prompt modality and learning setting, these strategies significantly improve CLIP.
- Prompts learned with iterative strategies help mitigate the "rich get richer, poor get poorer" effect observed in semi-supervised approaches leveraging pseudolabels. By redistributing the quality of pseudolabels across different classes, we observe a "Robin Hood effect" where the extremely rich classes' accuracy stays the same or decreases, while poorer classes get richer, leading to a more equitable distribution of per-class accuracy.

We contributed to understand and mitigate the limitations of CLIP related to composing concepts.

- We can improve the compositionality of large-scale vision-language models via compositional soft prompting. Particularly, we learn tokens of the prompts like objects and attribute from multiple combinations and then recombine them in new combinations for inference.
- This kind of training is not only beneficial for compositional tasks, but also improves CLIP's ability of recognizing attributes in an image.
- CLIP's ability to encode compositional concepts and to bind variables in a structure-sensitive way (e.g., differentiating cube behind sphere from sphere behind cube) is limited.

Finally, introducing Alfred we launch a second generation of programmatic weak supervision systems that drastically empower the experience of annotating data by relying on the definition of flexible rules using natural language, and a variety of large language models.

5. CONCLUSIONS

The landscape of machine learning research has been significantly reshaped by the emergence of large pre-trained models, particularly in the realm of learning from limited labeled data. Shifting from the initial inquiry of "How can we effectively train models for tasks devoid of labeled data, leveraging external knowledge sources?" we have swiftly transitioned to an effective paradigm where a handful of labeled examples suffice to fine-tune extensive pre-trained models for downstream tasks. While this assertion holds true, the complexity of this fine-tuning process amplifies when the target domain substantially diverges from the domains covered by the training data.

The challenge of Learning with Less Labels (LwLL) has been partially alleviated by capitalizing on more knowledgeable models, consequently reducing the demand for copious amounts of data. Nonetheless, stumbling blocks persist in scenarios where labeled data remains scarce or altogether absent. This is precisely where the research undertaken during the LwLL initiative remains profoundly relevant to the machine learning landscape. The insights gleaned during this endeavor continue to wield applicability in the current era of expansive models, especially in resource-constrained settings.

6. REFERENCES

- [1] W. Piriyaakulkij, C. Menghini, R. Briden, N. V. Nayak, J. Zhu, E. Raisi and S. H. Bach, "TAGLETS: A System for Automatic Semi-Supervised Learning with Auxiliary Data," in *Machine Learning Systems*, 2022.
- [2] R. Esfandiarpour, A. Pu, M. Hajabdollahi and S. H. Bach, "Extended few-shot learning: Exploiting existing resources for novel tasks," in *ArXiv*, 2020.
- [3] N. V. Nayak and S. H. Bach, "Zero-Shot Learning with Common Sense Knowledge Graphs," in *Transactions on Machine Learning Research*, 2022.
- [4] A. Mazzetto, C. Menghini, A. Yuan, E. Upfal and S. H. Bach, "Tight Lower Bounds on Worst-Case Guarantees for Zero-Shot Learning with Attributes," in *Advances in Neural Information Processing Systems*, 2022.
- [5] C. Menghini, A. Delworth and S. H. Bach, "Enhancing CLIP with CLIP: Exploring Pseudolabeling for Limited-Label Prompt Tuning," in *ArXiv*, 2023.
- [6] N. V. Nayak, P. Yu and S. H. Bach, "Learning to compose soft prompts for compositional zero-shot learning," in *ICLR*, 2023.
- [7] M. Lewis, N. V. Nayak, P. Yu, Q. Yu, J. Merullo, S. H. Bach and E. Pavlick, "Does CLIP Bind Concepts? Probing Compositionality in Large Image Models," in *ArXiv*, 2023.
- [8] A. Mazzetto, D. Sam, A. Park, E. Upfal and S. H. Bach, "Semi-supervised aggregation of dependent weak supervision sources with performance guarantees," in *International Conference on Artificial Intelligence and Statistics*, 2021.
- [9] A. Mazzetto, C. Cousins, D. Sam, S. H. Bach and E. Upfal, "Adversarial multi class learning under weak supervision with performance guarantees," in *International Conference on Machine Learning*, 2021.
- [10] A. Mazzetto, R. Esfandiarpour, E. Upfal and S. H. Bach, "An Adaptive Method for Weak Supervision with Drifting Data," in *ArXiv*, 2023.
- [11] P. Yu, T. Ding and S. H. Bach, "Learning from multiple noisy partial labelers," in *International Conference on Artificial Intelligence and Statistics*, 2022.
- [12] P. Yu and S. H. Bach, "Alfred: A System for Prompted Weak Supervision," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.
- [13] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly and N. and Houlsby, "Big transfer (bit): General visual representation learning.," in *Computer Vision - ECCV*, 2020.
- [14] R. Speer, J. Chin and C. and Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge.," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [15] Y. Cui, Y. Song, C. Sun, A. Howard and S. and Belongie, "Large scale fine-grained categorization and domainspecific transfer learning.," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [16] X. Yan, D. Acuna and S. Fidler, "Neural data server: A large-scale search engine for transfer learning data," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

- [17] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin and C.-L. Li, "Fixmatch: Simplifying semisupervised learning with consistency and confidence," in *Advances in Neural Information Processing Systems*, 2020.
- [18] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning (ICML)*, 2015.
- [19] R. Socher, M. Ganjoo, C. D. Manning and A. Ng, "Zero-shot learning through cross-modal transfer," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.
- [20] W. Wang, V. W. Zheng, H. Yu and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," in *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2019.
- [21] A. Farhadi, I. Endres, D. Hoiem and D. A. Forsyth, "Describing objects by their attributes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [22] C. H. Lampert, H. Nickisch and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.
- [23] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.
- [24] S. Reed, Z. Akata, H. Lee and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [25] B. Zhao, Y. Fu, R. Liang, J. Wu, Y. Wang and Y. Wang, "A large-scale attribute dataset for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.
- [26] N. Gupta, K. Lin, D. Roth, S. Singh and M. Gardner, "Neural module networks for reasoning over text," in *International Conference on Learning Representations (ICLR)*, 2020.
- [27] C. Wang, S. Min, X. Chen, X. Sun and H. Li, "Dual progressive prototype network for generalized zero-shot learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [28] P. Singh and H. Liu, "Conceptnet—a practical commonsense reasoning tool-kit," *BT technology journal*, vol. 22, no. 4, pp. 211--226, 2004.
- [29] H. Zhang, D. Khashabi, Y. Song and D. Roth, "Transomcs: From linguistic graphs to commonsense knowledge," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.
- [30] F. Ilievski, P. Szekely and B. Zhang, "Cskg: The commonsense knowledge graph," in *Extended Semantic Web Conference (ESWC)*, 2020.
- [31] W. Xiaolong, Y. Yufei and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [32] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang and E. P. Xing, "Rethinking knowledge graph propagation for zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

- [33] W. Hamilton, Z. Ying and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [34] D. Gillick, N. Lazic, K. Ganchev, J. Kirchner and D. Huynh, "Context-dependent fine-grained entity type tagging," in *arXiv*, 2014.
- [35] W. Ralph and B. Ada, "Bbn pronoun coreference and entity type corpus," in *Technical report*, 2005..
- [36] D. Jia, D. Wei, S. Richard, L. Li-Jia, L. Kai and F.-F. and Li, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [37] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril and e. al, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," in *ICML workshop on Privacy in Machine Learning and Artificial Intelligence*, 2018.
- [38] Y. Xian, C. H. Lampert, B. Schiele and Z. Akata, "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2018.
- [39] Y. Mansour, M. Mohri and R. A., "Domain adaptation: Learning bounds and algorithms," in *arXiv*, 2009.
- [40] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira and J. W. Vaughan, "A theory of learning from different domains," in *Machine learning*, 2010.
- [41] O. Sener, H. O. Song, A. Saxena and S. Savarese, "Learning transferrable representations for unsupervised domain adaptation," in *NIPS*, 2016.
- [42] P. H. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.
- [43] Y. Luo, L. Zheng, T. Guan, J. Yu and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [44] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu and C. Ré, "Snorkel: Rapid training data creation with weak supervision," in *Proceedings of the VLDB Endowment*, 2017.
- [45] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam and C. and Ré, "Data programming: Creating large training sets, quickly," in *Neural Information Processing Systems (NeurIPS)*, 2016.
- [46] S. H. Bach, B. He, A. Ratner and C. and Ré, "Learning the structure of generative models with- out labeled data," in *International Conference on Machine Learning (ICML)*, 2017.
- [47] S. Nitzan and J. Paroush, "Optimal decision rules in uncertain dichotomous choice situations," in *International Economic Review*, 1982.
- [48] D. A. P. and S. A. M., "Maximum likelihood estimation of observer error-rates using the EM algorithm," in *Journal of the Royal Statistical Society*, 1979.
- [49] C. Arachie and B. Huang, "Adversarial label learning," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.
- [50] A. Balsubramani and Y. Freund, "Optimally combining classifiers using unlabeled data," in *Conference on Learning Theory (COLT)*, 2015.

- [51] A. Balsubramani and Y. Freund, "Optimal binary classifier aggregation for general losses," in *Neural Information Processing Systems*, 2016.
- [52] C. Arachie and B. Huang, "A general framework for adversarial label learning," in *The Journal of Machine Learning Research*, 2021.
- [53] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.
- [54] T. Bonal and R. Com, "A minimax optimal algorithm for crowdsourcing," in *Advances in Neural Information Processing Systems*, 2017.
- [55] C. H. Lampert, H. Nickisch and S. Harmeling, "Learning to detect unseen object classes by betweenclass attribute transfer," in *CVPR*, 2009.
- [56] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y.-H. Sung, Z. Li and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International Conference on Machine Learning*, 2021.
- [57] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language models are unsupervised multitask learners," 2019.
- [58] L. Yuan, D. Chen, Y.-L. Chen, N. C. F. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, C. Liu, M. Liu, Z. Liu, Y. Lu, Y. Shi, L. Wang, J. Wang, B. Xiao and Z. Xiao, "Florence: A new foundation model for computer vision," in *ArXiv*, 2021.
- [59] C. Ge, R. Huang, M. Xie, Z. Lai, S. Song, S. Li and G. Huang, "Domain adaptation via prompt learning," in *ArXiv*, 2022.
- [60] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan and S.-N. Lim, "Visual prompt tuning," in *17th European Conference on Computer Vision*, 2022.
- [61] D.-H. Lee, "Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks," in *ICML 2013 Workshop: Challenges in Representation Learning (WREPL)*, 2013.
- [62] H. Huang, J. Chu and F. Wei, "Unsupervised prompt learning for vision-language models," in *ArXiv*, 2022.
- [63] S. Manli, N. Weili, H. De-An, Y. Zhiding, G. Tom, A. Anima and X. Chaowei, "Test-time prompt tuning for zero-shot generalization in vision-language models," in *NeurIPS*, 2022.
- [64] Y. Gao, X. Shi, Y. Zhu, H. Wang, Z. Tang, X. Zhou, M. Li and D. N. Metaxas, "Visual prompt tuning for test-time domain adaptation," in *ArXiv*, 2022.
- [65] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang and C. Raffel, "Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring," in *International Conference on Learning Representations*, 2020.
- [66] Z. D. Qizhe Xie, E. Hovy, T. Luong and Q. Le, "Unsupervised data augmentation for consistency training," in *Advances in Neural Information Processing Systems*, 2020.
- [67] B. Chen, J. Jiang, X. Wang, P. Wan, J. Wang and M. Long, "Debiased self-training for semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2022.
- [68] Z. Zhu, T. Luo and Y. Liu, "The rich get richer: Disparate impact of semisupervised learning," *International Conference on Learning Representations*, 2022.

- [69] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.
- [70] Y.-L. Sung, V. Nair and C. Raffel, "Training neural networks with fixed sparse masks," in *NeurIPS*, 2021.
- [71] E. Mitchell, C. Lin, A. Bosselut, C. Finn and C. D. Manning, "Fast model editing at scale," in *International Conference on Learning Representations*, 2022.
- [72] B. Lester, R. Al-Rfou and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.
- [73] G. Qin and J. Eisner, "Learning how to ask: Querying LMs with mixtures of soft prompts," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.
- [74] T. Vu, B. Lester, N. Constant, R. Al-Rfou and D. M. Cer, "Spot: Better frozen model adaptation through soft prompt transfer," in *ArXiv*, 2021.
- [75] K. Zhou, J. Yang, C. C. Loy and Z. Liu, "Learning to prompt for visionlanguage models," in *ArXiv*, 2021.
- [76] K. Zhou, J. Yang, C. C. Loy and Z. Liu, "Learning to prompt for visionlanguage models," in *ArXiv*, 2021.
- [77] B. Partee, "Lexical semantics and compositionality," in *An invitation to cognitive science: Language*, 1995.
- [78] K. Erk and S. Padó, "A structured vector space model for word meaning in context," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008.
- [79] J. Mitchell and M. Lapata, "Composition in Distributional Models of Semantics," in *Cognitive Science*, 2010.
- [80] M. Baroni and R. Zamparelli, "Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010.
- [81] B. Coecke, M. Sadrzadeh and S. Clark, "Mathematical Foundations for a Compositional Distributional Model of Meaning," in *ArXiv*, 2010.
- [82] G. Boleda, "Distributional Semantics and Linguistic Theory," in *ArXiv*, 2020.
- [83] R. Smith, J. A. Fries, B. Hancock and S. H. Bach, "Language models in the loop: Incorporating prompting into weak supervision," in *ArXiv*, 2022.
- [84] S. Arora, A. Narayan, M. F. Chen, L. J. Orr, N. Guha, K. Bhatia, I. Chami, F. Sala and C. Ré, "Ask me anything: A simple strategy for prompting language models," in *ArXiv*, 2022.
- [85] R. Zhang, Y. Yu, P. Shetty, L. Song and C. Zhang, "Prboost: Promptbased rule discovery and boosting for interactive weakly-supervised learning," in *ArXiv*, 2022.
- [86] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. v. Platen, C. Ma, Y. Jernite, J. Plu, C. Xu and e. al, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020.

- [87] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever and e. al, "Language models are unsupervised multitask learners," in *OpenAI blog*, 2019.
- [88] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," in *The Journal of Machine Learning Research*, 2020.

7. LIST OF ACRONYMS & GLOSSARY

ALL – Adversarial Label Learning
AMCL-CC – Adversarial Multi-Class Learning - Convex Combination
aPY – attribute Pascal Yahoo
AWA2 – Animals with Attributes 2
CDSM – Compositional Distributional Semantics Models
CI – Confidence Interval
CLIP – Contrastive Language-Image Pre-training
CSP – Compositional Soft-Prompts
CUB – Caltech University of San Diego Birds
DAP – Direct Attribute Prediction
DARPA – Defense Advanced Research Projects Agency
DAZLE – Dense Attention Zero-shot LEarning
DGP – Dense Graph Propagation
DTD – Describable Texture Dataset
ESZSL –Embarrassingly Simple Zero-Shot Learning
GCNZ –Graph Convolutional Network Zero-shot
GPT – Generative Pre-trained Transformer
GRIP – Grow and Refine Iteratively Pseudolabels
gRPC – Remote Procedure Call
GZSL – Generalized Zero-Shot Learning
IFPL – Iterative Fixed Pseudolabels
JPL – Jet Propulsion Laboratory
LAD – Large-scale Attribute Datasets
LF – Labeling Functions
LLM – Large Language Models
LP – Linear Program
LwLL – Learning with Less Labels
NC – Nearest Class
NLU – Natural Language Understanding
NPLM – Noisy Partial Label Model
ONNX – Open Neural Network Exchange
PGMV – Performance-Guaranteed Majority Vote
PLF – Partial Labeling Function
PWS – Programmatic Weak Supervision
SCADS – Structured Collection of Auxiliary Data Sets
SGCN – Sparsified Graph Convolutional Network
SSH – Secure Shell
SSL – Semi-Supervised Learning
TAGLETS – Tasks Algorithmically Given Labels Established via Transferred Symbols
TRZSL – Transductive Zero-Shot Learning
TrGCN – Transformer Graph Convolutional Network
UL – Unsupervised Learning
VLM – Vision-Language Models
VPT – Visual Prompt Tuning

WSS – Weak Supervision Source

ZSL-KG – Zero-Shot Learning with Knowledge Graphs