CHiLS: Zero-shot Image Classification with Hierarchical Label Sets

Zachary Novack
UC San Diego
znovack@ucsd.edu

Julian McAuley
UC San Diego
jmcauley@ucsd.edu

Zachary Lipton
Carnegie Mellon University
zlipton@andrew.cmu.edu

Saurabh Garg
Carnegie Mellon University
sgarg2@andrew.cmu.edu

Abstract

Open vocabulary models (e.g. CLIP) have shown strong performance on zero-shot classification through their ability generate embeddings for each class based on their (natural language) names. Prior works focused on improving the accuracy of these models through prompt engineering or by finetuning with a small amount of labeled downstream data. However, there has been little focus on improving the richness of the class names themselves, which can pose issues when class labels are coarsely-defined and uninformative. We propose **C**lassification with **Hi**erarchical **L**abel **S**ets (or CHiLS), an alternative strategy for zero-shot classification specially designed for datasets with implicit semantic hierarchies. CHiLS proceeds in three steps: (i) for each class, produce a set of subclasses, using either existing hierarchies or by querying GPT-3; (ii) perform the standard zero-shot CLIP procedure as though these subclasses were the labels of interest; (iii) map the predicted subclass back to its parent to produce the final prediction. Across numerous datasets with underlying hierarchical structure, CHiLS improves accuracy in situations both with and without ground-truth hierarchical information.

1 Introduction

There has been a recent growth of interest in the capabilities of pretrained *open vocabulary models* (Radford et al., 2021; Wortsman et al., 2021; Jia et al., 2021; Gao et al., 2021; Pham et al., 2021; Cho et al., 2022; Pratt et al., 2022). These models, e.g., CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), learn to map images and captions into shared embedding spaces such that images are close in embedding space to their corresponding captions but far from randomly sampled captions. The resulting models can then be used to assess the relative compatibility of a given image with an arbitrary set of textual "prompts". Radford et al. (2021) observed that by inserting each class name directly within a natural language prompt, one can then use CLIP embeddings to perform zero-shot image classification with high success rates (Radford et al., 2021; Zhang et al., 2021b).

Despite the documented successes, the current interest in open vocabulary models poses a new question: **How should we represent our classes for a given problem in natural language?** As class names are part of the predictive pipeline (as opposed to mostly an afterthought in standard settings) for open vocabulary models, CLIP's performance is now directly tied to the descriptiveness of the class "prompts" (Santurkar et al., 2022). While there is a growing body of work on improving the quality of the prompts into which class names are embedded (Radford et al., 2021; Pratt et al., 2022; Zhou et al., 2022b;a; Huang et al., 2022), surprisingly little attention has been paid to improving the *richness of the class names themselves*. This can be particularly crucial in cases where datasets may contain a rich underlying structure but have uninformative class labels. Consider, for an example, the class "large man-made outdoor things" in the CIFAR20 dataset (Krizhevsky, 2009), which includes "bridges" and "roads" but also "castles" and "skyscrapers" (see App. 3 for more information).

In this paper, we introduce a new method to tackle zero-shot classification with CLIP models for classification tasks with coarsely-defined class labels. We refer to our method as **C**lassification with

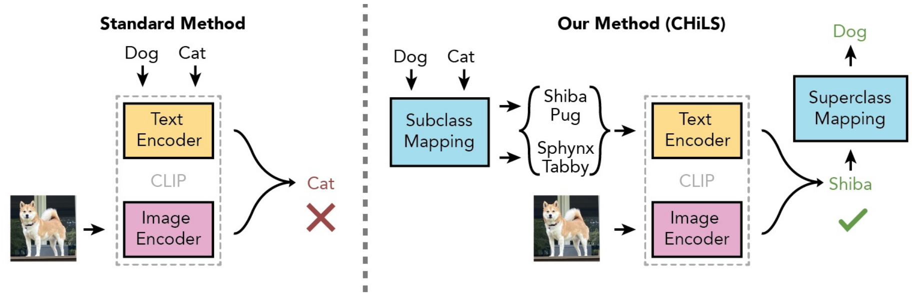


Figure 1: **(Left)** *Standard CLIP Pipeline for Zero-Shot Classification*. For inference, a standard CLIP takes in input a set of classes and an input image, and makes a prediction from that set of classes. **(Right)** *Our proposed method CHiLS for using hierarchical class information in the zero-shot pipeline*. We map each individual class to a set of subclasses, perform inference in the subclass space (i.e., union set of all subclasses), and map the predicted subclass back to its original superclass.

Hierarchical **L**abel **S**ets (CHiLS). Our method utilizes a hierarchical map to convert each class into a list of subclasses, performs normal CLIP zero-shot prediction across the union set of all *subclasses*, and finally uses the inverse mapping to convert the subclass prediction to the requisite superclass. We additionally include a reweighting step wherein we leverage the raw superclass probabilities in order to make our method robust to less-confident predictions at the superclass and subclass level.

We evaluate CHiLS on a wide array of image classification benchmarks with *and* without available hierarchical information. These datasets share the property of having an underlying semantic substructure that is not captured in the set of class labels. In the former case, leveraging preexisting hierarchies leads to strong accuracy gains across all datasets. In the latter, we show that we can use GPT-3 to query a list of *possible* subclasses for each class (whether or not they are actually present in the dataset), which still leads to consistent improved accuracy over raw superclass prediction.

2 Proposed Method

In this paper, we are primarily concerned with the problem of zero-shot image classification in CLIP models For CLIP models, zero-shot classification involves using both a pretrained image encoder and a pretrained text encoder (see the left part of Figure 1). To perform zero-shot classification, we need a predefined set of classes written in natural language. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ be such a set. Given an image and set of classes, each class is embedded within a natural language prompt (through some function $\mathrm{T}(\cdot)$) to produce a "caption" for each class (e.g. one standard prompt mentioned in Radford et al. (2021) is *"A photo of a* $\{\}$*."*). These prompts are then fed into the text encoder and after passing the image through the image encoder, we calculate the cosine similarity between the image embedding and each class-prompt embedding. These similarity scores form the output "logits" of the CLIP model, which can be passed through a softmax to generate the class probabilities.

As noted in Appendix A, previous work has focused on improving the $\mathrm{T}(\cdot)$ for each class label c_i. With CHiLS, we instead focus on the complementary task of directly modifying the set of classes $\mathcal{C}$ when $\mathcal{C}$ is ill-formed or overly general, while keeping $\mathrm{T}(\cdot)$ fixed. Our method involves two main steps: (1) performing zero-shot prediction over label *subclasses* and (2) aligning subclass probabilities with the raw superclass outputs to reconcile both inference methods.

Zero-Shot Prediction with Hierarchical Label Sets Our method CHiLS slightly modifies the standard approach for zero-shot CLIP prediction. As each class label c_i represents some concept in natural language (e.g. the label "dog"), we acquire a **subclass set** $\mathcal{S}_{c_i} = \{s_{c_i,1}, s_{c_i,2}, \ldots, s_{c_i,m_i}\}$ through some mapping function G, where each $s_{c_i,j}$ is a linguistic *hyponym*, or subclass, of c_i (e.g. corgi for dogs) and m_i is the size of the set S_{c_i}. Given a label set S_{c_i} for each class, we proceed with the standard process for zero-shot prediction, but now using the *union* of all label sets as the set of classes. Through this, CHiLS will output a distribution over all subclasses $\widehat{\boldsymbol{y}}_{\text{sub}}$. We then leverage the inverse mapping function G^{-1} to map the argmax subclass probabilitiy back into the corresponding superclass $G^{-1}(\arg\max \widehat{\boldsymbol{y}}_{\text{sub}})$. Our method is detailed more formally in Algorithm 1. In our work, we experiment with two scenarios: (i) when hierarchy information is available and can be readily queried; and (ii) when hierarchy information is *not* available and the label set for each class must be generated, which we do so by prompting GPT-3.

Algorithm 1 Classification with **Hi**erarchical **L**abel **S**ets (CHiLS)

input : data point $\boldsymbol{x}$, class labels $\mathcal{C}$, prompt function T, label set mapping G, CLIP model f

Set $\mathcal{C}_{\text{sub}} \leftarrow \cup_{c_i \in \mathcal{C}} G(c_i)$ ▷ Union of subclasses for subclass prediction
$\widehat{\boldsymbol{y}}_{\text{sub}} = \sigma(f(\boldsymbol{x}, \mathrm{T}(\mathcal{C}_{\text{sub}})))$ ▷ Subclass probabilities
$\widehat{\boldsymbol{y}}_{\text{sup}} = \sigma(f(\boldsymbol{x}, \mathrm{T}(\mathcal{C})))$ ▷ Superclass probabilities
for $i = 1$ to $|\mathcal{C}|$ **do**
  $S_{c_i} = G(c_i)$
  for $s_{c_i,j} \in S_{c_i}$ **do**
    $\widehat{\boldsymbol{y}}_{\text{sub}}[s_{c_i,j}] = \widehat{\boldsymbol{y}}_{\text{sub}}[s_{c_i,j}] * \widehat{\boldsymbol{y}}_{\text{sup}}[c_i]$ ▷ Combines subclass & superclass probability
  end for
end for

output : $G^{-1}(\arg\max \widehat{\boldsymbol{y}}_{\text{sub}})$

Table 1: Zero-Shot performance at different levels of ImageNet hierarchy, where CHiLS has access to true ImageNet leaf node classes. CHiLS shows clear performance gains over the baseline at coarse-to-intermediate granularities.

ImageNet Depth	Standard	CHiLS	% Leaf Classes
1	67.43	97.08	0.0
2	69.22	90.47	0.0
3	63.97	86.20	0.0
4	49.48	80.31	32.03
5	63.80	74.08	77.90
6	62.96	65.07	96.28

Reweighting probabilities with Superclass Confidence While the above method is able to effectively utilize CLIP's ability to identify relatively fine-grained concepts, by predicting on only subclass labels we lose any positive benefits of the superclass label, and performance may vary widely based on the quality of the subclass labels. Given recent evidence (Minderer et al., 2021; Kadavath et al., 2022) that large language models are well-calibrated and generally assign higher probability to correct predictions, we modify our initial algorithm to leverage this behavior and use *both* superclass and subclass information. We provide empirical evidence of this property in Appendix C.1.

Specifically, we include an additional reweighting step within our main algorithm. Here, we reweight each set of subclass probabilities by its superclass probability. Heuristically, as the prediction is now taken as the argmax over *products* of probabilities, large disagreements between subclass and superclass probabilities will be down-weighted (especially if one particular superclass is confident) and subclass probabilities will be more important in cases where the superclass probabilities are roughly uniform. We show ablations on the choice of the reweighting algorithm in Appendix B.1.

3 A Motivating Example

Before validating the effectiveness of CHiLS across standard benchmarks, we provide a more nuanced investigation on the ImageNet dataset at different hierarchy levels. Given that ImageNet is arranged in a rich taxonomical structure, we perform zero-shot classification at progressively finer levels of the hierarchy, where CHiLS is given access to all the leaf nodes in each class at the current level (unless the classes are themselves leaf nodes).

In Table 1, we see that at lower depths (e.g. depth 1 or 2), CHiLS significantly improves on top of standard zero-shot performance. As the depth in the hierarchy increase, the gap between CHiLS's performance and the standard zero shot decreases while the number of leaf nodes increases. This behavior highlights a key fact about CHiLS's potential use cases: *CHiLS can help for tasks where class labels resemble intermediate nodes of the ImageNet hierarchy.*

4 Experiments

Datasets As we are primarily concerned with improving zero-shot CLIP performance in situations with uninformative and / or semantically coarse class labels (as described in Appendix 3), we test our method on 15 different datasets (see Table 2). We use the validation sets for each dataset (if

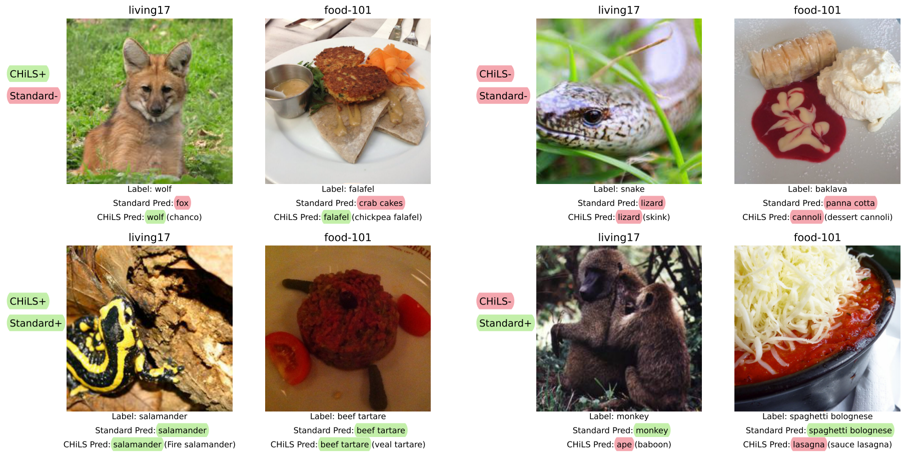


Figure 2: Selected examples of behavior differences between the standard and CHiLS performance across two different datasets. (Upper left): CHiLS is correct, standard prediction is not. (Lower left): Both correct. (Upper right): Both wrong. (Lower Right): standard prediction is correct, CHiLS is not.

Table 2: Zero-shot accuracy performance across image benchmarks with superclass labels (baseline), CHiLS with existing hierarchy (whenever available), and CHiLS with GPT-3 generated hierarchy. CHiLS improves classification accuracy in all situations with given label sets and all but 2 datasets with GPT-3 generated label sets.

Dataset	Superclass	CHiLS (True Map)	CHiLS (GPT-3 Map)
Nonliving26 (Santurkar et al., 2021)	79.8	90.7 (+10.9)	81.7 (+1.9)
Living17 (Santurkar et al., 2021)	91.1	93.8 (+2.7)	91.6 (+0.5)
Entity13 (Santurkar et al., 2021)	77.5	92.6 (+15.1)	78.1 (+0.6)
Entity30 (Santurkar et al., 2021)	70.3	88.9 (+18.6)	71.7 (+1.4)
CIFAR20 (Krizhevsky, 2009)	59.6	85.3 (+25.7)	65.0 (+5.4)
Food-101 (Bossard et al., 2014)	93.9	N/A	93.8 (−0.1)
Fruits360 (Mureşan & Oltean, 2018)	58.7	59.2 (+0.5)	60.1 (+1.4)
Fashion1M (Xiao et al., 2015)	45.8	N/A	47.4 (+1.6)
Fashion-MNIST (Xiao et al., 2017)	68.5	N/A	70.8 (+2.3)
LSUN-scene (Yu et al., 2015)	88.1	N/A	88.8 (+0.7)
Office31 (Saenko et al., 2010)	89.1	N/A	90.5 (+1.4)
OfficeHome (Venkateswara et al., 2017)	88.8	N/A	88.9 (−0.1)
ObjectNet (Barbu et al., 2019)	53.1	85.3 (+32.2)	53.5 (+0.4)
EuroSAT (Helber et al., 2019; 2018)	62.1	N/A	62.4 (+0.3)
RESISC45 (Cheng et al., 2017)	72.6	N/A	72.7(+0.1)

present). These datasets constitute a breadth of different image domains and include datasets with and without available hierarchy information. Additionally, the chosen datasets vary widely in the semantic granularity of their classes, from overly general cases (CIFAR20) to settings with a mixture of general and specific classes (Food-101, OfficeHome). We also examine CHiLS's robustness to distribution shift within a dataset by averaging all results for the BREEDS datasets, Office31, and OfficeHome across different shifts. We additionally modify the Fruits-360 and ObjectNet datasets to create existing taxonomies. More details for dataset preparation are detailed in Appendix C.7. Additionally, see Apppendix C.3 for details regarding the dataset-dependent choice of the prompt template function $\mathrm{T}(\cdot)$.

Model Architecture Unless otherwise noted, we use the ViTL/14@336px backbone (Radford et al., 2021) for our CLIP model, and DaVinci-002 (temperature $= 0.7$) for all ablations with GPT-3.

Choice of Mapping Function G In our experiments, we primarily look at how the choice of the mapping function G influences the performance of CHiLS. In Section 4.1, we first focus on the datasets with available hierarchy information. Here, G and G^{-1} are simply table lookups to find the list of subclasses and corresponding superclass respectively. Later, we explore situations in which the true set of subclasses is unknown. In these scenarios, we use GPT-3 to generate our mapping function G. Specifically, given some label set size m, superclass name `class-name`, and optional **context** (see Appendix C.3), we query GPT-3 with the prompt: `"Generate a list of` **`m`** `types of the following` **`[context]: class-name."`** The resulting output list from GPT-3 thus defines our mapping G from superclass to subclass. Unless otherwise specified, we fix $m = 10$ for all datasets. Additionally, in Appendix B.1 we explore situations in which hierarchical information is present but noisy, i.e. the label set for each superclass contains the true subclasses *and* erroneous subclasses that are not present in the dataset.

4.1 RESULTS

Leveraging Available Hierarchy Information We start in the scenario in which there is hierarchy information already available (or readily accessible). In this situation, the set of subclasses for each superclass is exactly specified and correct (i.e. every image within each superclass falls into one of the subclasses). For example, each class in Nonliving26 is made up of 4 ImageNet subclasses at finer granularity (e.g. "roof" includes "dome" and "thatch"). In Table 2, we can see that our method performs better than using the baseline superclass labels alone across all 7 of the datasets with available hierarchy information, in some cases leading to +15% improvement in predictive accuracy.

CHiLS in Unknown Hierarchy Settings Though we have seen considerable success in situations with access to the true hierarchical structure, in some real-world settings our dataset may not include any available information about the subclasses within each class. In this scenario, we turn to using GPT-3 to approximate the hierarchical map G (as specified in Section 4). It is important to note that GPT-3 may sometimes output suboptimal label sets, most notably in situations where GPT-3 chooses the wrong wordsense or when GPT-3 only lists modifiers on the original superclass (e.g. producing the list `[red, yellow, green]` for types of apples). In order to account for these issues in an out-of-the-box fashion, we automatically append the superclass name (if not already present) to each generated subclass label, and also include the superclass itself within the label set. For a controlled analysis about the effect of including the superclass itself in the label set, see Appendix C.4. In this setting, our method is still able to beat the baseline performance in most datasets, albeit with lower accuracy gains (see Table 2). Thus, while knowing the true subclass hierarchy can lead to large accuracy gains, it is enough to simply enumerate a list of possible subclasses for each class with no prior information about the dataset in order to improve the predictive accuracy. In Figure 2, we show selected examples to highlight CHiLS's behavior across two datasets.

5 CONCLUSION

In this work, we demonstrated that the zero-shot image classification capabilities of CLIP models can be improved by leveraging hierarchical information for a given set of classes. When hierarchical structure is available in a given dataset, our method shows large improvements in zero-shot accuracy, and even when subclass information *isn't* explicitly present, we showed that we can leverage GPT-3 to generate subclasses for each class and still improve upon the baseline (superclass) accuracy. We remark that CHiLS may be quite beneficial to practitioners using CLIP as an out-of-the-box image classifier. Namely, we show that in scenarios where the class labels may be ill-formed or overly coarse, even without existing hierarchical data accuracy can be improved with a *fully automated* pipeline (via querying GPT-3), yet CHiLS is flexible enough that any degree of hand-crafting label sets can be worked into the zero-shot pipeline. Our method has the added benefit of being both *completely zero-shot* (i.e. no training or fine-tuning necessary) and is resource efficient.

Limitations and Future Work We recognize that CHiLS is suited for scenarios in which a semantic hierarchy likely exists, and thus may not be useful for problems where the classes are already fine-grained. We believe that this limitation will not hinder the applicability of our method, as practitioners can assess if their task contains any latent semantic hierarchy (see Appendix 3) and thus choose to use our method or not *a priori*. Given CHiLS's empirical successes, we hope to perform more investigation to develop an understanding of *why* CHiLS is able to improve zero-shot accuracy.

ACKNOWLEDGMENTS

SG acknowledges Amazon Graduate Fellowship and JP Morgan AI Ph.D. Fellowship for their support. ZL acknowledges Amazon AI, Salesforce Research, Facebook, UPMC, Abridge, the PwC Center, the Block Center, the Center for Machine Learning and Health, and the CMU Software Engineering Institute (SEI) via Department of Defense contract FA8702-15-D-0002. ZN thanks Honey the Shiba Inu for denoting their likeness to the opening figure.

REFERENCES

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014.

Sebastian Bujwid and Josephine Sullivan. Large-scale zero-shot image classification from rich and diverse textual descriptions. In *Proceedings of the Third Workshop on Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, 2021.

Zhong Cao, Jiang Lu, Sen Cui, and Changshui Zhang. Zero-shot handwritten chinese character recognition with hierarchical decomposition embedding. *Pattern Recognition*, 2020.

Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. An empirical study on large-scale multi-label text classification including few and zero-shot labels. *arXiv preprint arXiv:2010.01653*, 2020.

Shiming Chen, GuoSen Xie, Yang Liu, Qinmu Peng, Baigui Sun, Hao Li, Xinge You, and Ling Shao. Hsva: Hierarchical semantic-visual adaptation for zero-shot learning. 2021.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017.

Jaemin Cho, Seunghyun Yoon, Ajinkya Kale, Franck Dernoncourt, Trung Bui, and Mohit Bansal. Fine-grained image captioning with clip reward. In *Findings of NAACL*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Hierarchical annotation of medical images. *Pattern Recognition*, 2011.

Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*, 2022.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. 2018.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning (ICML)*, 2019.

Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models, 2022.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *arXiv preprint arXiv:2208.05592*, 2022.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Chia-Chih Kuo and Kuan-Yu Chen. Toward zero-shot and zero-resource multilingual question answering. *IEEE Access*, 2022.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2008.

Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. Improving pretrained models for zero-shot multi-label text classification through reinforced label hierarchy reasoning. *arXiv preprint arXiv:2104.01666*, 2021.

Thomas Mensink, Efstratios Gavves, and Cees GM Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Horea Mureşan and Mihai Oltean. Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica*, 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, et al. Combined scaling for open-vocabulary image classification. *arXiv preprint arXiv: 2111.10050*, 2021.

Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. In *International Conference on Learning Representations (ICLR)*, 2021.

Shibani Santurkar, Yann Dubois, Rohan Taori, Percy Liang, and Tatsunori Hashimoto. Is a caption worth a thousand images? a controlled study for representation learning, 2022.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

Sheng Shen, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Anna Rohrbach, Zhe Gan, Lijuan Wang, Lu Yuan, Ce Liu, Kurt Keutzer, Trevor Darrell, and Jianfeng Gao. K-lite: Learning transferable visual models with external knowledge, 2022.

Carlos Nascimento Silla and Alex Alves Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 2010.

Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic, 2021.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Kai Yi, Xiaoqian Shen, Yunhao Gou, and Mohamed Elhoseiny. Exploring hierarchical graph representation for large-scale zero-shot image classification. *arXiv preprint arXiv:2203.01386*, 2022.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2015.

Youngjae Yu, Jiwan Chung, Heeseung Yun, Jack Hessel, JaeSung Park, Ximing Lu, Prithviraj Ammanabrolu, Rowan Zellers, Ronan Le Bras, Gunhee Kim, and Yejin Choi. Multimodal knowledge alignment with reinforcement learning, 2022.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv*, 2022.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021a.

Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. *arXiv preprint arXiv:2112.02413*, 2021b.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022b.

REPRODUCIBLITY STATEMENT

The source code for reproducing the work presented here is available at `https://github.com/acmi-lab/CHILS`. We implement our method in PyTorch (Paszke et al., 2017) and provide an infrastructure to run all the experiments to generate corresponding results. We have stored all models and logged all hyperparameters and seeds to facilitate reproducibility. Additionally, all necessary data preprocessing details are present in Appendix C.7.

APPENDIX

A RELATED WORK

Transfer Learning While the focus of this paper is to improve CLIP models in the zero-shot regime, there is a large body of work exploring improvements to CLIP's few-shot capabilities. In the standard fine-tuning paradigm for CLIP models, practitioners discard the text encoder and only use the image embeddings as inputs for some additional training layers.

One particular line of work on improving the fine-tuned capabilities of CLIP models leverages model weight interpolation between the pre-trained and fine-tuned models (Wortsman et al., 2021; 2022; Ilharco et al., 2022). There is another line of work that seeks to improve CLIP models by injecting a small amount of learnable parameters into the frozen CLIP backbone, through building on the adapter framework from parameter-efficient learning (Houlsby et al., 2019; Gao et al., 2021; Zhang et al., 2021a) or by adding learnable "prompt" vectors inside the model (Jia et al., 2022). Additionally, some have looked at circumventing the entire process of prompt engineering, whether by directly optimizing the prompt embeddings (Zhou et al., 2022a;b; Huang et al., 2022) or pre-training the language model with a frozen image model (Zhai et al., 2022). In all the above situations, *some* amount of data, whether labeled or not, is used in order to improve CLIP's accuracy.

Zero-Shot Prediction The field of Zero-Shot Learning (ZSL) has existed well before the emergence of open vocabularly models, with its inception traced to Larochelle et al. (2008). Outside of CLIP related methods, the ZSL paradigm has shown success in improving multilingual question answering (Kuo & Chen, 2022) and image classification (Bujwid & Sullivan, 2021; Shen et al., 2022). With CLIP models, ZSL success has been found in a variety of tasks, including 3D recognition (Zhang et al., 2021b), image-to-text generation (Tewel et al., 2021), VQA (Shen et al., 2021), audio-captioning (Yu et al., 2022), object navigation (Gadre et al., 2022), and open-ended reasoning (Zeng et al., 2022). Unlike our work here, these prior directions mostly focus on generative problems or, in the case of Bujwid & Sullivan (2021) and Shen et al. (2022), require rich external databases to employ their methods. In the realm of improving CLIP's zero-shot capabilities for image classification, we particularly note the work of Pratt et al. (2022). Here, authors explore using GPT-3 to generate rich textual prompts for each class rather than using preexisting prompt templates. In contrast, our work explores a complementary direction of leveraging hierarchy in class names to improve zero-shot performance of CLIP with a fixed set of preexisting prompt templates.

Hierarchical Classification Methodologies from Hierarchical Classification (HC) (Silla & Freitas, 2010), where there is a DAG-like structure to the class labels, have been extensively used for multi-label classification (Dimitrovski et al., 2011; Liu et al., 2021; Chalkidis et al., 2020), and recent works have shown that this paradigm can aid in zero-shot learning during the pretraining step (Chen et al., 2021; Mensink et al., 2014; Yi et al., 2022; Cao et al., 2020). While our work is similar in spirit to prior work on HC, we note that there are two crucial distinctions: (i) we are concerned only with the zero-shot *training-free* regime (as we only require class names) while most previous work assumes some amount of training, and (ii) CHiLS only leverages the class hierarchy for the flat task of *superclass* prediction without requiring any supervision at the subclass level.

B MAIN ABLATIONS

B.1 ABLATIONS

Is Reweighting Necessary? Though the reweighting step in CHiLS is motivated by the evidence that CLIP generally assigns higher probability to *correct* predictions rather than incorrect ones (see

Table 3: Average accuracy across datasets for superclass prediction, CHiLS (ours), and CHiLS *without* the reweighting step. While when given the true hierarchy omitting the reweighting step can slightly boost performance beyond CHiLS, in situations without the true hierarchy the reweighting step is crucial to improving on the baseline accuracy.

Experiment	Average Accuracy
Standard	73.28
CHiLS (True Map, No RW)	86.40
CHiLS (True Map, RW)	85.11
CHiLS (GPT Map, No RW)	71.61
CHiLS (GPT Map, RW)	74.49

Table 4: Average accuracy across datasets with GPT-generated label sets for different reweighting algorithms. Using aggregate subclass probabilities for reweighting performs noticeably worse than our initial method and reweighting in superclass space. CHiLS too only performs slightly worse than the contrived best possible union of subclass and superclass predictions.

Experiment	Average Accuracy
Best Possible	78.69
Standard	73.28
CHiLS	74.49
CHiLS (RW subclass w/mean subclass)	72.79
CHiLS (RW mean subclass w/superclass)	74.55

Appendix C.1 for empirical verification), it is not immediately clear whether the reweighting step is truly necessary. Averaged across all documented datasets, in Table 3 we show that in the true hierarchy setting, not reweighting the subclass probabilities can actually slightly *boost* performance (as the label sets are adequately tuned to the distribution of images). However, in situations where the true hierarchy is not present, omitting the reweighting step puts accuracy below the baseline performance. We attribute this difference in behavior to the fact that reweighting multiplicatively combines the superclass and subclass predictions, and thus if subclass performance is sufficient on its own (as is the case when the true hierarchy is available) then combining it with superclass predictions can cause the model to more closely follow the behavior of the underperforming superclass predictor. Thus, as the presence of a ground-truth hierarchy is not guaranteed in the wild, the reweighting step is necessary for CHiLS to improve zero-shot performance.

Different Reweighting Strategies We also experimented on whether the initial reweighting algorithm is the optimal method for combining superclass and subclass predictions. Namely, we investigated whether superclass probabilities could be replaced by the sum over the matching subclass probabilities, *and* whether we can aggregate subclass probabilities and reweight them with the matching superclass probabilities (i.e. performing the normal reweighting step but in the space of superclasses). In Table 4 we show that replacing the superclass probabilities in the reweighting step with aggregate subclass probabilities removes any accuracy gains from CHiLS, but that doing the reweighting step in superclass space *does* maintain CHiLS accuracy performance. This suggests that the beneficial behavior of CHiLS may be due to successfully combining two different sets of class labels. We also display the upper bound for combining superclass and subclass prediction (i.e. the accuracy when a datum is correctly labeled if the superclass *or* subclass predictions are correct) in purple, which we note is impossible in practice, and observe that even the best possible performance is not much higher than the performance of CHiLS.

Noisy Available Hierarchies While lacking access to any existing hierarchical information is the most probable in practice, we additionally investigate the situation in which the hierarchical information is present but *overestimates* the set of subclasses. For example, the scenario in which a dataset with the class “dog” includes huskies and corgis, but CHiLS is provided with huskies, corgis, *and Labradors* as possible subclasses, with the last being out-of-distribution. To do this, we return to the BREEDS datasets presented in Santurkar et al. (2021). As the BREEDS datasets were created so that

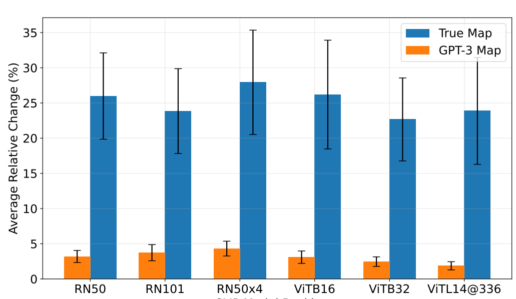
Table 5: CHiLS zero-shot accuracy when G includes *all* subclasses in the ImageNet hierarchy descended from the respective root node. Even in the presence of noise added to the true label sets, CHiLS is able to make large accuracy gains.

Dataset	Standard	CHiLS (True Map)	CHiLS (True Map+)
nonliving26	79.8	90.7 (+10.9)	89.8 (+10.0)
living17	91.1	93.8 (+2.7)	93.2 (+2.1)
entity13	77.5	92.6 (+15.1)	90.7 (+13.2)
entity30	70.3	88.9 (+18.6)	86.7 (+16.4)

each class contains the same number of subclasses (which are ImageNet classes), we modify G such that the label set for each superclass corresponds to *all* the ImageNet classes descended from that node in the hierarchy (see Appendix C.6 for more information). As we can see in Table 5, CHiLS is able to improve upon the baseline performance even in the presence of added noise in each label set.

Label Set Size In previous works investigating importance of prompts in CLIP's performance, it has been documented that the number of prompts used can have a decent effect on the overall performance (Pratt et al., 2022; Santurkar et al., 2022). Along this line, we investigate how the size of the *subclass set* generated for each class effects the overall accuracy by re-running our main experiments with varying values of m (namely, 1, 5, 10, 15, and 50). In Figure 3 (right), there is little variation across label set sizes that is consistent over all datasets, though $m = 1$ has a few very low performing outliers due to the extremely small label set size. We observe that the optimal label set size is context-specific, and depends upon the total number of classes present and the semantic granularity of the classes themselves. Individual dataset results are available in Appendix C.5.

Model Size In order to examine whether the performance of CHiLS only exists within the best performing CLIP backbone (e.g. ViT-L/14@336), we measure the average relative change in accuracy performance between CHiLS and the baseline superclass predictions across all datasets for an array of different CLIP models. Namely, we investigate the RN50, RN101, RN50x4, ViT-B/16, ViT-B/32, and ViT-L/14@336 CLIP backbones (see Radford et al. (2021) for more information on the model specifications). In Figure 3 (left), we show that across the 6 specified CLIP backbones, CHiLS performance leads to relatively consistent relative accuracy gains, with a slight (but not confidently significant) trend showing improved performance for the ResNet backbones over the ViT backbones, which is to be expected given their worse base capabilities. This shows that CHiLS's benefits are not an artifact of large model size.



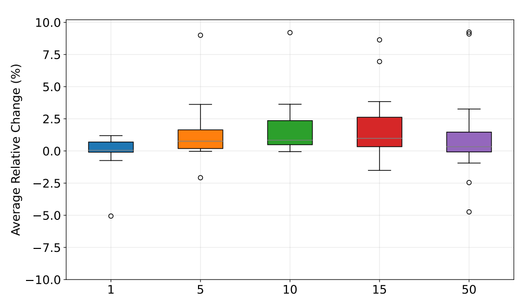


Figure 3: (Left) Average relative change between CHiLS and baseline for true mapping and GPT-3 generated mapping. Across changes in CLIP backbone size and structure, the effectiveness of CHiLS at improving performance only varies slightly. (Right) Average relative accuracy change from the baseline to CHiLS (across all datasets), for varying label set sizes. In all, there is not much difference in performance across label set sizes.

Alternative Aggregating Methods While CHiLS is based on a *set-based* mapping approach for subclasses and a linear averaging for prompt templates (based on Radford et al. (2021)'s procedure),

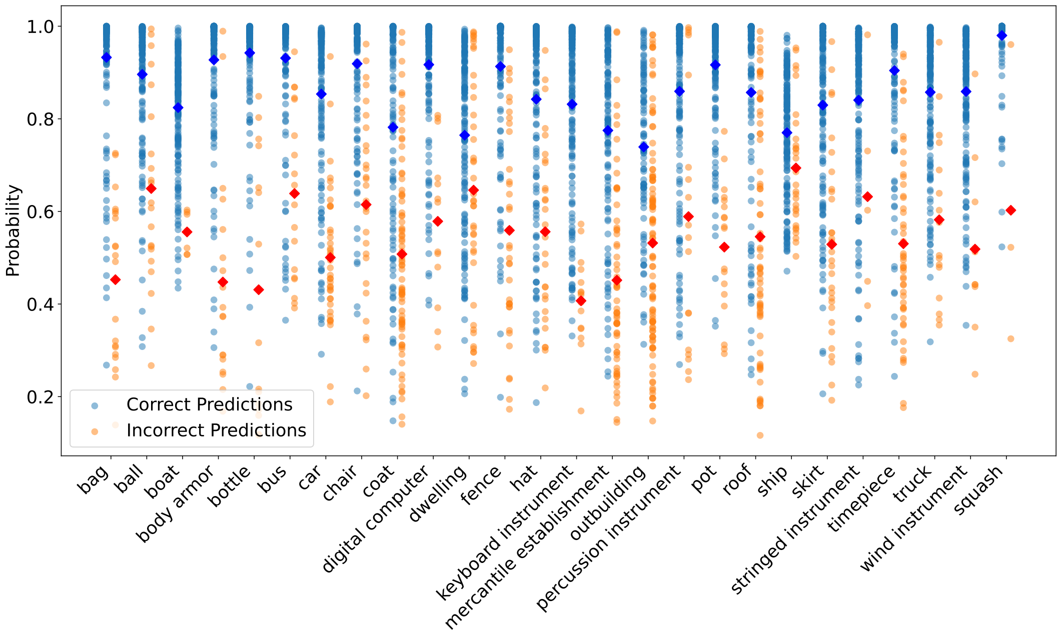
Table 6: Average accuracy across datasets for varying aggregative methods on both the prompt and subclass steps of the zero-shot pipeline. In general, linear averaging for subclasses performs worse than our proposed set-based method, while linear averaging for prompts (for raw superclass prediction) performs better thant using a set-based mapping.

Experiment	Accuracy
Superclass (linear average)	73.28
Superclass (set-based prompt mapping)	72.25
CHiLS (True Map, set-based mapping)	85.11
CHiLS (True Map, linear average)	81.61
CHiLS (GPT Map, set-based mapping)	74.43
CHiLS (GPT Map, linear average)	72.25

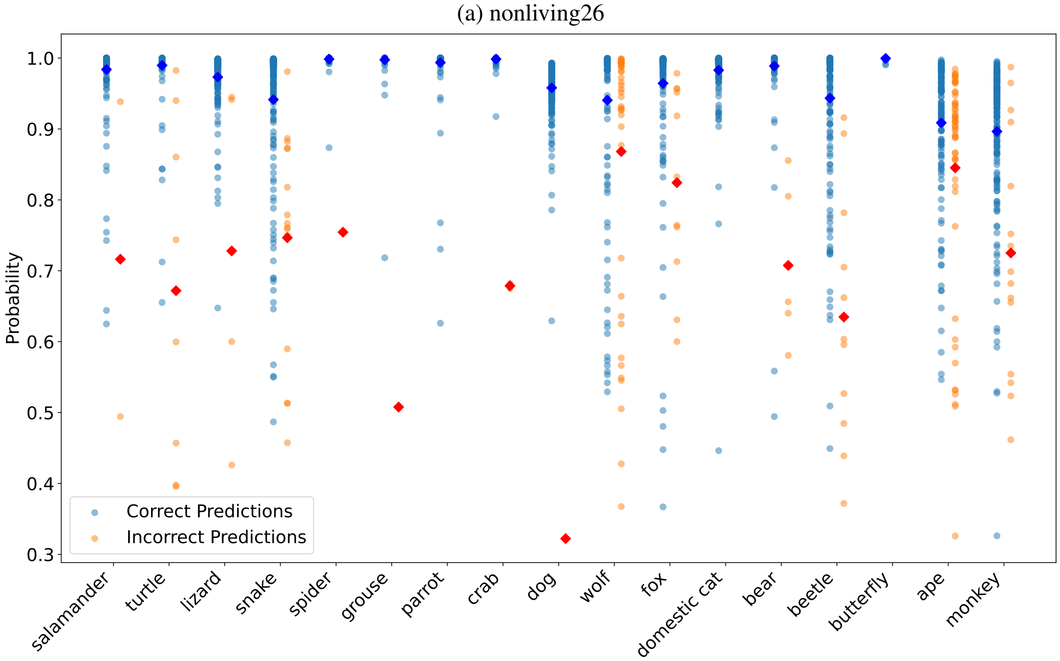
we experimented with two alternative ensembling methods for different parts of the CHiLS pipeline: (1) Using a *linear average* of subclass embeddings rather than the set-based mapping (that is, every superclass's text embedding is the average across all subclass embeddings, each themselves averaged across every prompt template) and (2) Using a *set-based* mapping for prompt templates rather than a linear average (i.e. instead of averaging across prompt templates, predict across each prompt template separately at inference time and then use embedded class to map back to the set of superclasses). Note in the latter case we only experiment with how this effects *superclass* prediction (where each class maps to a set of the dataset's chosen prompt embeddings), as using set-based ensembling for *both* prompts and subclasses within CHiLS quickly becomes computationally expensive. In Table 6, we see that using our initial aggregation methods (i.e. linear averaging for prompts and set mappings for subclasses) achieves greater accuracy.

C ADDITIONAL APPENDICES

C.1 EMPIRICAL EVIDENCE OF CLIP CONFIDENCE



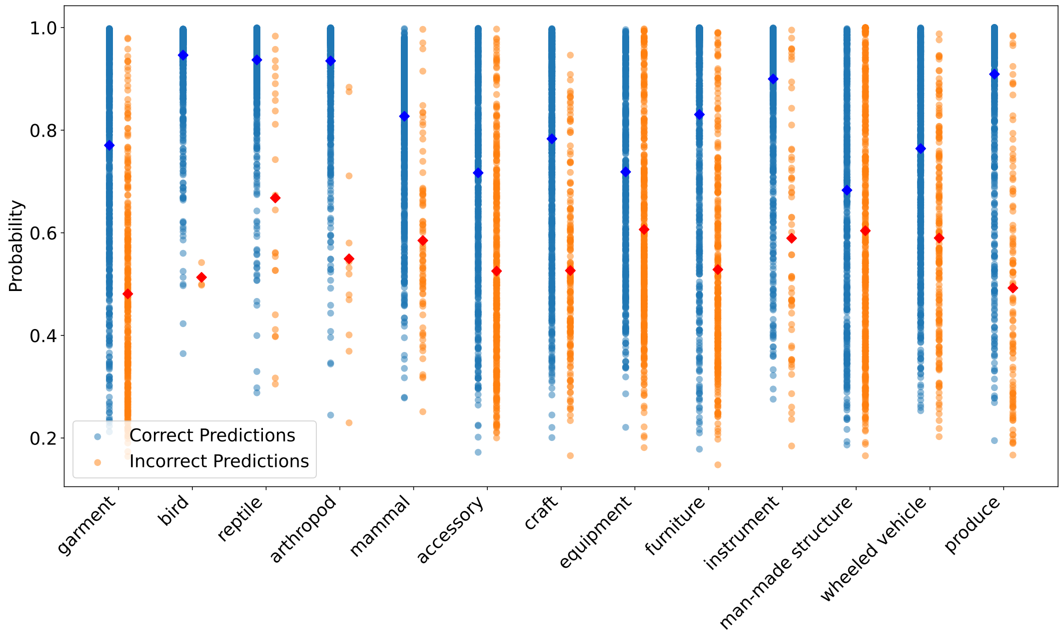
(a) nonliving26



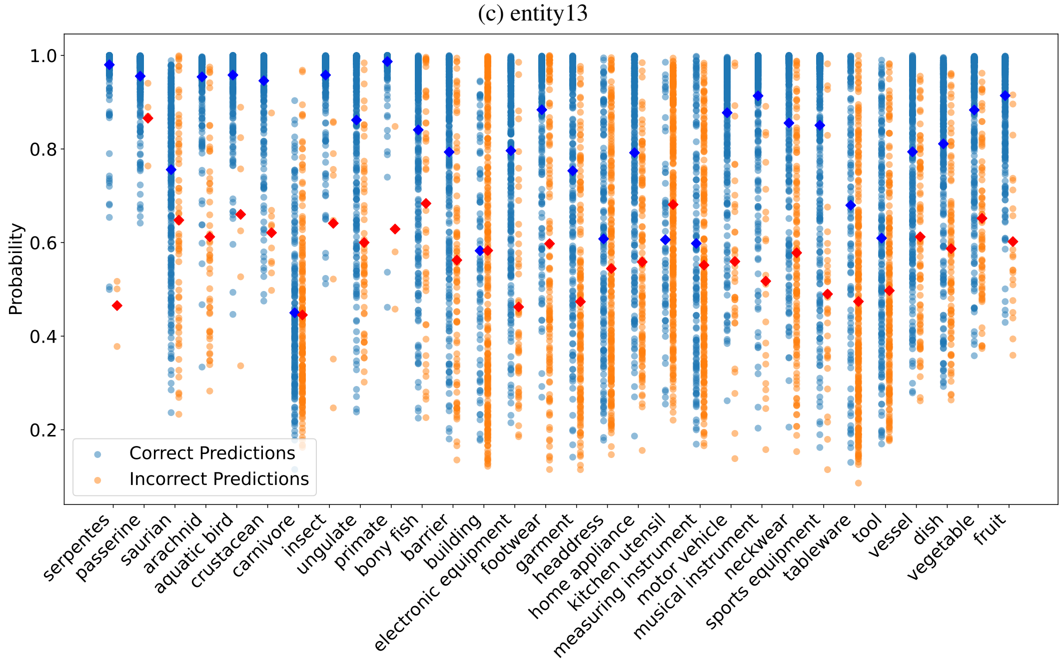
(b) living17

Figure 4: Distribution of argmax probabilities across ImageNet BREEDS datasets for correctly and incorrectly classified data points, with the diamonds representing average probability for each class. Correctly classified probabilities are on average higher than the misclassified probabilities.

The motivation behind the reweighting step of CHiLS primarily comes from the heuristic that LLMs make correct predictions with high estimated probabilities assigned to them (Kadavath et al., 2022), and that CLIP models themselves are well-calibrated (Minderer et al., 2021). However, we also



(c) entity13



(d) entity30

verify whether there is some evidence of this behavior in CLIP models. Given that the output of a CLIP model is a probability distribution over the provided classes, we care specifically about the probability of the *argmax* class (i.e. the predicted class) when the model is correct and when it is incorrect. Across the BREEDS datasets for the standard ImageNet domain, in Figure 4 we show the distribution of the correct and incorrect argmax probabilities for each class (i.e. for each class c_i, we show the output probabilties for c_i when it was correctly classified and the output probabilities of the predicted classes when the true class is c_i). Whenever CLIP is correct, the associated probability is on average much higher than the probabilities associated with misclassification.

C.2 CLIP PRIMER

Open Vocabulary models (as termed in Pham et al. (2021)) refer to models that are able to classify images by associating them with natural language descriptions of each class. These models are "open" in the sense that they are to predict on an *arbitrary* vocabulary of descriptions (as opposed to a fix set), thus allowing for arbitrary-way image classification. Popular open vocabulary models include the model of focus CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) as examples.

Contrastive **L**anguage **I**mage **P**retraining (CLIP) is a family of open vocabulary models, and the focus of the present work. CLIP, which is comprised of a text encoder and an image encoder that project into the same latent space, is trained in the following way: Given a set of image-caption pairs (e.g. a photo of a dog with the caption "a photo of a dog."), CLIP is trained to predict which caption goes with which image as a contrastive learning objective by comparing the similarity between each image embedding and each caption embedding.

At inference time (in the zero-shot setting), a naïve method for image classification (which is the initial baseline tried in Radford et al. (2021)) involves simply passing in the list of class names for a given dataset, and calculating the similarity between a particular image embedding and each one of these class embeddings. However, Radford et al. (2021) found that by taking a cue from the recent literature on prompt engineering for large language models (Gao et al., 2020), CLIP can perform significantly better as a zero-shot predictor if each class name is included in a natural language **prompt** that resembles some sort of image caption (as that is what CLIP was trained on). As an example, the standard baseline prompt mentioned is *"A photo of a* {}*."*. In our work, we define a prompt (or prompt template, which we use interchangeably) as any caption-like phrase in natural language that a class name can be injected into.

C.3 ADDING CONTEXT TO PROMPTS AND GPT-3 QUERIES

Table 7: Context tokens and prompt sets used for each dataset.

Dataset	**[context]**	Prompt Set Used
Nonliving26	N/A	ImageNet
Living17	N/A	ImageNet
Entity13	N/A	ImageNet
Entity30	N/A	ImageNet
CIFAR20	N/A	ImageNet
Food-101	"food"	Dataset-Specific
Fruits-360	"fruit"	Dataset-Specific
Fashion1M	"article of clothing"	Dataset-Specific
Fashion-MNIST	"article of clothing"	ImageNet
LSUN-Scene	N/A	ImageNet
Office31	"office supply"	Dataset-Specific
OfficeHome	"office supply"	ImageNet
ObjectNet	N/A	ImageNet
EuroSAT	N/A	Dataset-Specific
RESISC45	N/A	Dataset-Specific

For the choice of the prompt embedding function $\mathrm{T}(\cdot)$, for each dataset we experiment (where applicable) with two different functions: (1) Using the average text embeddings of the 75 different prompts for each label used for ImageNet in Radford et al. (2021), where the prompts cover a wide array of captions and (2) Following the procedure that Radford et al. (2021) puts forth for more specialized datasets, we modify the standard prompt to be of the form *"A photo of a* {}*, a type of* ***[context]****."*, where **[context]** is dataset-dependent (e.g. "food" in the case of food-101). In the case that a custom prompt set exists for a dataset, as is the case with multiple datasets that the present work shares with Radford et al. (2021), we use the given prompt set for the latter option rather than building it from scratch. For each dataset, we use the prompt set that gives us the best *baseline* (i.e. superclass) zero-shot performance. We follow the procedure laid out for ImageNet in Radford et al. (2021) by averaging the text embeddings of each prompt set for each class label.

Table 8: Zero-Shot Accuracy Performance across benchmarks, controlling for the presence of the superclass label within each respective label set. In the existing map case, adding the superclass labels removes some of the performance gains of the raw existing map. In the GPT-3 Map case, adding the superclass is crucial to maintaining performance in most datasets

Dataset	CHiLS Accuracy (Existing Map)	CHiLS Accuracy (Existing Map+)	CHiLS Accuracy (GPT-3 Map)	CHiLS Accuracy (GPT-3 Map+)
Nonliving26	90.68 (+10.85)	89.80 (+9.97)	81.46 (+1.63)	81.52 (+1.69)
Living17	93.81 (+2.72)	93.62 (+2.53)	91.30 (+0.21)	91.43 (+0.33)
Entity13	92.59 (+15.13)	92.06 (+14.60)	76.97 (-0.48)	78.10 (+0.65)
Entity30	88.87 (+18.55)	87.29 (+16.96)	71.79 (+1.47)	71.74 (+1.42)
CIFAR20	85.28 (+25.71)	81.45 (+21.88)	65.67 (+6.10)	65.91 (+6.34)
Food-101	N/A	N/A	93.66 (-0.21)	93.80 (-0.07)
Fruits-360	59.22 (+0.48)	58.88 (+0.15)	60.53 (+1.79)	60.12 (+1.38)
Fashion1M	N/A	N/A	47.51 (+1.73)	47.44 (+1.66)
Fashion-MNIST	N/A	N/A	70.79 (+2.27)	70.76 (+2.24)
LSUN-Scene	N/A	N/A	88.80 (+0.60)	88.97 (+0.77)
Office31	N/A	N/A	86.58 (−2.71)	89.37 (+0.24)
OfficeHome	N/A	N/A	87.88 (−0.97)	88.76 (−0.09)
ObjectNet	85.34 (+32.24)	81.30 (+28.20)	51.23 (−2.07)	53.52 (+0.42)
EuroSAT	N/A	N/A	62.21 (+0.11)	62.40 (+0.30)
RESISC45	N/A	N/A	71.84 (-0.75)	72.75 (+0.16)

In order to disentangle the effect that well-formed prompt templates have on the success of CHiLS, for each dataset (besides the BREEDS datasets and ObjectNet as they are already semantically similar to ImageNet) we compare the ImageNet 75 classes against a dataset-specific set of prompt templates. In the case of EuroSAT, RESISC45, CIFAR20 and Food-101, we directly use the prompt template set from Radford et al. (2021). For LSUN-Scene, we use the prompt template set for SUN397 (Xiao et al., 2010), as the two datasets are semantically similar. For the rest of the datasets not yet mentioned (namely Fruits360, Fashion1M, Fashion-MNIST, Office31, and OfficeHome) we add the **[context]** marker into the standard prompt template as mentioned in Section 4. The prompt sets themselves can be directly found in the code implementation for this project.

For the GPT-3 Query with additional context, we add the respective **[context]** token to the query *if* the dataset-specific prompt template is used. Note that we did not create **[context]** tokens for EuroSAT, LSUN-Scene, or RESISC45 despite testing dataset-specific prompt templates, as there did not seem to be a concise semantic label to describe the classes in these datasets. In Table 7, we list the dataset, the **[context]** token (if applicable), and the final prompt set used for all the experiments. Here, we found that while dataset-specific prompts often improved baseline performance, they were not *gauranteed* to improve performance, as in both Fasion-MNIST and OfficeHome the general ImageNet prompt set performed better.

C.4 Including Superclass Labels in Label Sets

With CHiLS when the existing map is not available, we append the superclass name to each label set to account for possible noise in the GPT-generated label set. In Table 8, we show the effect that this inclusion has in both the existing map and GPT-map cases. Note that in the main paper, columns 1 and 4 correspond to the main results (i.e. no superclass labels in existing maps and superclass labels in GPT-3 maps). In both cases, the presence of the superclass label more effectively strikes a balance between subclass and superclass predictions. In the existing map case, this actually *hurts* performance, as the subclass labels are optimal in the given dataset. In the GPT-3 map case, while there are some datasets where removing the superclass label improves performance (namely Fruits360 and Entity30), in ever other case removing the superclass label hurts performance, sometimes by multiple percentage points.

C.5 Label Set Ablation Accuracy

Table 9 displays the raw accuracy scores for CHiLS across different label set sizes.

Table 9: Accuracy across different label set sizes generated by GPT-3, with best performing label set size in each row bolded. In general, there is no consistent trend related to label set size and zero-shot performance across datasets.

Dataset	CHiLS ($m = 1$)	CHiLS ($m = 5$)	CHiLS ($m = 10$)	CHiLS ($m = 15$)	CHiLS ($m = 50$)
Nonliving26	79.71 (−0.12)	81.12 (+1.29)	81.68 (+1.85)	**81.98** (+2.15)	80.03 (+0.20)
Living17	91.14 (+0.04)	**92.68** (+1.58)	91.56 (+0.46)	91.73 (+0.63)	91.41 (+0.31)
Entity13	77.43 (−0.02)	78.14 (+0.69)	78.10 (+0.65)	**78.37** (+0.92)	78.28 (+0.83)
Entity30	71.06 (+0.73)	71.48 (+1.15)	71.72 (+1.39)	**73.03** (+2.70)	72.62 (+2.29)
CIFAR20	60.15 (+0.58)	64.93 (+5.36)	**65.05** (+5.48)	63.71 (+4.14)	64.99 (+5.42)
Food-101	93.84 (−0.03)	**93.90** (+0.03)	93.82 (−0.05)	93.81 (−0.06)	93.73 (−0.14)
Fruits360	58.70 (−0.04)	59.70 (+0.96)	**60.14** (+1.40)	59.75 (+1.01)	59.66 (+0.92)
Fashion1M	43.46 (−2.32)	45.77 (−0.01)	**47.44** (+1.66)	46.95 (+1.17)	43.61 (−2.17)
Fashion-MNIST	68.01 (−0.51)	**71.00** (+2.48)	70.81 (+2.29)	69.07 (+0.55)	69.45 (+0.93)
LSUN-scene	88.43 (+0.30)	86.30 (−1.83)	**88.83** (+0.70)	86.80 (−1.33)	85.97 (−2.16)
Office31	89.51 (+0.38)	88.15 (−0.98)	**90.55** (+1.42)	89.43 (+0.30)	89.42 (+0.29)
OfficeHome	88.75 (−0.12)	89.11 (+0.24)	88.76 (−0.09)	**89.16** (+0.29)	88.87 (+0.00)
ObjectNet	53.75 (+0.63)	53.27 (+0.15)	53.53 (+0.41)	57.70 (+4.58)	**58.03** (+4.91)
EuroSAT	62.32 (+0.21)	62.21 (+0.10)	62.40 (+0.29)	**62.72** (+0.61)	62.11 (0.00)
RESISC45	**73.29** (+0.70)	73.05 (+0.46)	72.71 (+0.12)	72.67 (+0.08)	71.90 (−0.69)

C.6 Noisy Available Hierarchy Details

The ImageNet (Deng et al., 2009) dataset itself includes a rich hierarchical taxonomy, where every class is a leaf node of the hierarchy. In the original BREEDS (Santurkar et al., 2021) work, the authors modify the structure slightly in order to place concepts at semantically-similar levels of granularity at the same depth, and additional restrict the number of subclasses within each of the BREEDS datasets in order to balance the data. Thus, it is possible for each BREEDS dataset to use the dataset with its superclasses and restricted set of subclasses but provide CHiLS with *all* the subclass labels present in the ImageNet hierarchy for each superclass (i.e. all leaf nodes descended from each superclass node). In Table 11, we display a subset of the living17 BREEDS dataset class structure with the original subclasses and the ImageNet subclasses. Observe that in some cases, there are many subclass labels provided to CHiLS than is present in the data.

C.7 Dataset Details

Table 10: Domains used for BREEDS, Office31, and OfficeHome.

Dataset	Domains
BREEDS	ImageNet, ImageNet-Sketch, ImageNetv2, ImageNet-c {Fog-1, Contrast-2, Snow-3, Gaussian Blur-4, Saturate-5}
Office31	Amazon, DSLR, webcam
OfficeHome	Clipart, Art, Real World, Product

CHiLS Across Domain Shifts For each of the BREEDS datasets (Santurkar et al., 2021), Office31 (Saenko et al., 2010), and OfficeHome (Venkateswara et al., 2017), all results presented are the average over different domains. The specific domains used are show in Table 10.

Fruits-360 For zero-shot classification with CLIP models, Fruits-360 (Mureşan & Oltean, 2018) in its raw form is somewhat ill-formed from a class name perspective, as there are classes only differentiated by a numeric index (e.g. “Apple Golden 1” and “Apple Golden 2”) and classes at mixed granularity (e.g. “forest nut” and “hazelnut” are separate classes even though hazelnuts are a type of forest nut). We thus manually rename classes using the structure laid out in Table 13, which results in a 59-way superclass classification problem, with 102 ground-truth subclasses.

Table 11: Subset of living17 class hierarchy, showing the difference between the original BREEDS subclasses and the ImageNet subclasses used for the ablation in Section B.1: Noisy Available Hierarchies.

Superclass	Original BREEDS subclasses	All ImageNet subclasses
salamander	European fire salamander, common newt, eft, spotted salamander	European fire salamander, common newt, eft, spotted salamander, axolotl
turtle	loggerhead, leatherback turtle, mud turtle, terrapin	loggerhead, leatherback turtle, mud turtle, terrapin, box turtle
lizard	common iguana, American chameleon, agama, frilled lizard	banded gecko, common iguana, American chameleon, whiptail, agama, frilled lizard, alligator lizard, Gila monster, green lizard, African chameleon, Komodo dragon
snake	thunder snake, ringneck snake, diamondback, sidewinder	thunder snake, ringneck snake, hognose snake, green snake, king snake, garter snake, water snake, vine snake, night snake, boa constrictor, rock python, Indian cobra, green mamba, sea snake, horned viper, diamondback, sidewinder
spider	black and gold garden spider, barn spider, garden spider, black widow	black and gold garden spider, barn spider, garden spider, black widow, tarantula, wolf spider
grouse	black grouse, ptarmigan, ruffed grouse, prairie chicken	black grouse, ptarmigan, ruffed grouse, prairie chicken
parrot	African grey, macaw, sulphur-crested cockatoo, lorikeet	African grey, macaw, sulphur-crested cockatoo, lorikeet
crab	Dungeness crab, rock crab, fiddler crab, king crab	Dungeness crab, rock crab, fiddler crab, king crab

ObjectNet The ObjectNet dataset (Barbu et al., 2019) has partial overlap (113 classes) with the ImageNet (Deng et al., 2009) hierarchical class structure. From this subset of ObjectNet, we use the BREEDS hierarchy (Santurkar et al., 2021) to generate a coarse-grained version of ObjectNet that is shown in Table 12. In this 11-way classification task, the true subclasses are the original ObjectNet classes.

Table 12: Class Structure for ObjectNet experiments.

Superclass	Subclasses (Original ObjectNet)
garment	{Dress, Jeans, Skirt, Suit jacket, Sweater, Swimming trunks, T-shirt}
soft furnishings	{Bath towel, Desk lamp, Dishrag or hand towel, Doormat, Lampshade, Paper towel, Pillow}
accessory	{Backpack, Dress shoe (men), Helmet, Necklace, Plastic bag, Running shoe, Sandal, Sock, Sunglasses, Tie, Umbrella, Winter glove}
appliance	{Coffee/French press, Fan, Hair dryer, Iron (for clothes), Microwave, Portable heater, Toaster, Vacuum cleaner}
equipment	{Cellphone, Computer mouse, Keyboard, Laptop (open), Monitor, Printer, Remote control, Speaker, Still Camera, TV, Tennis racket, Weight (exercise)}
furniture	{Bench, Chair}
toiletry	{Band Aid, Lipstick}
wheeled vehicle	{Basket, Bicycle}
cooked food	{Bread loaf}
produce	{Banana, Lemon, Orange}
beverage	{Drinking Cup}

Table 13: Mapping from original class names to new subclass and superclasses for Fruits-360.

Original Class	Cleaned Subclass	Cleaned Superclass
Apple Braeburn	braeburn apple	apple
Apple Crimson Snow	crimson snow apple	apple
Apple Golden 1	golden apple	apple
Apple Golden 2	golden apple	apple
Apple Golden 3	golden apple	apple
Apple Granny Smith	granny smith apple	apple
Apple Pink Lady	pink lady apple	apple
Apple Red 1	red apple	apple
Apple Red 2	red apple	apple
Apple Red 3	red apple	apple
Apple Red Delicious	red delicious apple	apple
Apple Red Yellow 1	red yellow apple	apple
Apple Red Yellow 2	red yellow apple	apple
Apricot	apricot	apricot
Avocado	avocado	avocado
Avocado ripe	avocado	avocado
Banana	banana	banana
Banana Lady Finger	lady finger banana	banana
Banana Red	red banana	banana
Beetroot	beetroot	beetroot
Blueberry	blueberry	blueberry
Cactus fruit	cactus fruit	cactus fruit
Cantaloupe 1	melon	melon
Cantaloupe 2	melon	melon
Carambula	star fruit	star fruit
Cauliflower	cauliflower	cauliflower
Cherry 1	cherry	cherry
Cherry 2	cherry	cherry
Cherry Rainier	rainier cherry	cherry
Cherry Wax Black	black cherry	cherry
Cherry Wax Red	red cherry	cherry
Cherry Wax Yellow	yellow cherry	cherry
Chestnut	nut	nut
Clementine	orange	orange
Cocos	cocos	cocos
Corn	corn	corn
Corn Husk	corn husk	corn husk
Cucumber Ripe	cucumber	cucumber
Cucumber Ripe 2	cucumber	cucumber
Dates	date	date
Eggplant	eggplant	eggplant
Fig	fig	fig
Ginger Root	ginger root	ginger root
Granadilla	granadilla	passion fruit
Grape Blue	blue grape	grape
Grape Pink	pink grape	grape
Grape White	white grape	grape
Grape White 2	white grape	grape
Grape White 3	white grape	grape
Grape White 4	white grape	grape
Grapefruit Pink	pink grapefruit	grapefruit
Grapefruit White	white grapefruit	grapefruit
Guava	gauva	gauva
Hazelnut	nut	nut
Huckleberry	huckleberry	huckleberry

Kaki	kaki	persimmon
Kiwi	kiwi	kiwi
Kohlrabi	kohlrabi	kohlrabi
Kumquats	kumquat	kumquat
Lemon	lemon	lemon
Lemon Meyer	meyer lemon	lemon
Limes	lime	lime
Lychee	lychee	lychee
Mandarine	orange	orange
Mango	mango	mango
Mango Red	red mango	mango
Mangostan	mangostan	mangostan
Maracuja	maracuja	passion fruit
Melon Piel de Sapo	melon	melon
Mulberry	mulberry	mulberry
Nectarine	nectarine	nectarine
Nectarine Flat	flat nectarine	nectarine
Nut Forest	forest nut	nut
Nut Pecan	pecan nut	nut
Onion Red	red onion	onion
Onion Red Peeled	red onion	onion
Onion White	white onion	onion
Orange	orange	orange
Papaya	papaya	papaya
Passion Fruit	passion fruit	passion fruit
Peach	peach	peach
Peach 2	peach	peach
Peach Flat	flat peach	peach
Pear	pear	pear
Pear 2	pear	pear
Pear Abate	abate pear	pear
Pear Forelle	forelle pear	pear
Pear Kaiser	kaiser pear	pear
Pear Monster	monster pear	pear
Pear Red	red pear	pear
Pear Stone	stone pear	pear
Pear Williams	williams pear	pear
Pepino	pepino	pepino
Pepper Green	green pepper	pepper
Pepper Orange	orange pepper	pepper
Pepper Red	red pepper	pepper
Pepper Yellow	yellow pepper	pepper
Physalis	groundcherry	groundcherry
Physalis with Husk	groundcherry	groundcherry
Pineapple	pineapple	pineapple
Pineapple Mini	mini pineapple	pineapple
Pitahaya Red	dragon fruit	dragon fruit
Plum	plum	plum
Plum 2	plum	plum
Plum 3	plum	plum
Pomegranate	pomegranate	pomegranate
Pomelo Sweetie	pomelo	pomelo
Potato Red	red potato	potato
Potato Red Washed	red potato	potato
Potato Sweet	sweet potato	potato
Potato White	white potato	potato
Quince	quince	quince
Rambutan	rambutan	rambutan
Raspberry	raspberry	raspberry

Redcurrant	redcurrant	redcurrant
Salak	salak	snake fruit
Strawberry	strawberry	strawberry
Strawberry Wedge	strawberry	strawberry
Tamarillo	tamarillo	tamarillo
Tangelo	tangelo	tangelo
Tomato 1	tomato	tomato
Tomato 2	tomato	tomato
Tomato 3	tomato	tomato
Tomato 4	tomato	tomato
Tomato Cherry Red	cherry tomato	tomato
Tomato Heart	heart tomato	tomato
Tomato Maroon	maroon tomato	tomato
Tomato Yellow	yellow tomato	tomato
Tomato not Ripened	unripe tomato	tomato
Walnut	nut	nut
Watermelon	melon	melon