Shaping Visual Representations with Language for Few-Shot Classification

Jesse Mu[1], **Percy Liang**[1], **Noah D. Goodman**[1,2]
Departments of [1]Computer Science and [2]Psychology
Stanford University
{muj,ngoodman}@stanford.edu, pliang@cs.stanford.edu

Abstract

By describing the features and abstractions of our world, language is a crucial tool for human learning and a promising source of supervision for machine learning models. We use language to improve few-shot visual classification in the underexplored scenario where natural language task descriptions are available during training, but unavailable for novel tasks at test time. Existing models for this setting sample new descriptions at test time and use those to classify images. Instead, we propose *language-shaped learning* (LSL), an end-to-end model that regularizes visual representations to predict language. LSL is conceptually simpler, more data efficient, and outperforms baselines in two challenging few-shot domains.

1 Introduction

Humans are powerful and efficient learners partially due to the ability to *learn from language* (Chopra et al., 2019; Tomasello, 1999). For instance, we can learn about *robins* not by seeing thousands of examples, but by being told that *a robin is a bird with a red belly and brown feathers*. This language further shapes the way we view the world, constraining our hypotheses for new concepts: given a new bird (e.g. *seagulls*), even without language we know that features like belly and feather color are relevant (Goodman, 1955).

In this paper, we guide visual representation learning with language, studying the setting where *no language is available at test time*, since rich linguistic supervision is often unavailable for new concepts encountered in the wild. How can one best use language in this setting? One option is to just regularize, training representations to predict language descriptions. Another is to exploit the compositional nature of language directly by using it as a bottleneck in a discrete latent variable model.

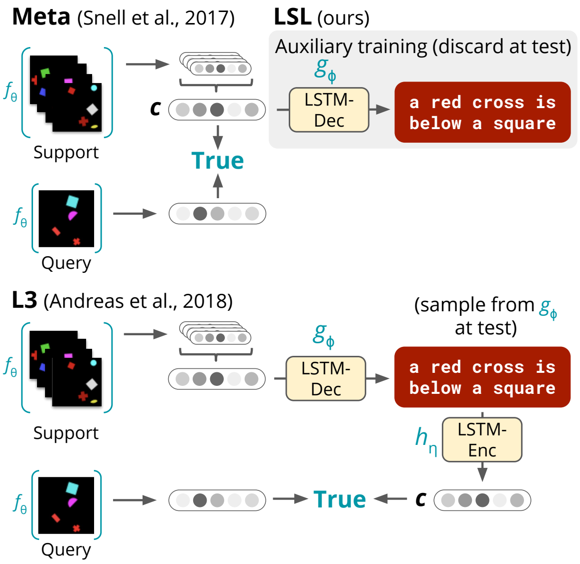

Meta (Snell et al., 2017)
LSL (ours)
Auxiliary training (discard at test)
g_ϕ
LSTM-Dec
a red cross is below a square
f_θ
c
Support
True
Query
L3 (Andreas et al., 2018)
(sample from g_ϕ at test)
g_ϕ
LSTM-Dec
a red cross is below a square
h_η
LSTM-Enc
f_θ
Support
f_θ
True
c
Query

Figure 1: We propose few-shot classification models whose learned representations are constrained to predict natural language task descriptions during training, in contrast to models which explicitly use language as a bottleneck for classification (Andreas et al., 2018).

For example, the recent *Learning with Latent Language* (L3; Andreas et al., 2018) model does both: during training, language is used to classify images; at test time, with no language, descriptions are sampled from a decoder conditioned on the language-shaped image embeddings.

Whether the bottleneck or regularization most benefits models like L3 is unclear. We disentangle these effects and propose *language-shaped learning* (LSL), an end-to-end model that uses visual representations shaped by language (Figure 1), thus avoiding the bottleneck. We find that discrete bottlenecks can hurt performance, especially with limited language data; in contrast, LSL is architecturally simpler, faster, uses language more efficiently, and outperforms L3 and baselines across two few-shot transfer tasks.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4823–4830
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

2 Related Work

Language has been shown to assist visual classification in various settings, including traditional visual classification with no transfer (He and Peng, 2017) and with language available at test time in the form of class labels or descriptions for zero- (Frome et al., 2013; Socher et al., 2013) or few-shot (Xing et al., 2019) learning. Unlike past work, we have no language at test time and test tasks differ from training tasks, so language from training cannot be used as additional class information (cf. He and Peng, 2017) or weak supervision for labeling additional in-domain data (cf. Hancock et al., 2018). Our setting can be viewed as an instance of *learning using privileged information* (LUPI; Vapnik and Vashist, 2009), where richer supervision augments a model only during training.

In this framework, learning with attributes and other domain-specific rationales has been tackled extensively (Zaidan et al., 2007; Donahue and Grauman, 2011; Tokmakov et al., 2019); language less so. Gordo and Larlus (2017) use METEOR scores between captions as a similarity measure for specializing embeddings for image retrieval, but do not directly ground language explanations. Srivastava et al. (2017) explore a supervision setting similar to ours, except in simple text and symbolic domains where descriptions can be easily converted to executable logical forms via semantic parsing.

Another line of work studies the generation of natural language explanations for interpretability across language (e.g. entailment; Camburu et al., 2018) and vision (Hendricks et al., 2016, 2018) tasks, but here we examine whether predicting language can actually improve task performance; similar ideas have been explored in text (Rajani et al., 2019) and reinforcement learning (Bahdanau et al., 2019; Goyal et al., 2019) domains.

3 Language-shaped learning

We are interested in settings where language explanations can help learn representations that generalize more efficiently across tasks, especially when training data for each task is scarce and there are many spurious hypotheses consistent with the input. Thus, we study the few-shot (meta-)learning setting, where a model must learn from a set of train tasks, each with limited data, and then generalize to unseen tasks in the same domain.

Specifically, in N-way, K-shot learning, a task t consists of N *support* classes $\{\mathcal{S}_1^{(t)}, \dots, \mathcal{S}_N^{(t)}\}$ with K examples each: $\mathcal{S}_n^{(t)} = \{\mathbf{x}_{n,1}^{(t)}, \dots, \mathbf{x}_{n,K}^{(t)}\}$. Each task has M *query* examples $\mathcal{Q}^{(t)} = \{(\mathbf{x}_1^{(t)}, y_1^{(t)}), \dots, (\mathbf{x}_M^{(t)}, y_M^{(t)})\}$. Given the m-th query example $\mathbf{x}_m^{(t)}$ as input, the goal is to predict its class $y_m^{(t)} \in \{1, \dots, N\}$. After learning from a set of tasks $\mathcal{T}_{\text{train}}$, a model is evaluated on unseen tasks $\mathcal{T}_{\text{test}}$.

While the language approach we propose is applicable to nearly any meta-learning framework, we use prototype networks (Snell et al., 2017), which have a simple but powerful inductive bias for few-shot learning. Prototype networks learn an embedding function f_θ for examples; the embeddings of the support examples of a class n are averaged to form a class *prototype* (omitting task $^{(t)}$ for clarity):

$$\mathbf{c}_n = \frac{1}{K} \sum_{k=1}^{K} f_\theta(\mathbf{x}_{n,k}). \tag{1}$$

Given a query example $(\mathbf{x}_m, y_m)$, we predict class n with probability proportional to some similarity function s between $\mathbf{c}_n$ and $f_\theta(\mathbf{x}_m)$:

$$p_\theta(\hat{y}_m = n \mid \mathbf{x}_m) \propto \exp\left(s\left(\mathbf{c}_n, f_\theta\left(\mathbf{x}_m\right)\right)\right). \tag{2}$$

f_θ is then trained to minimize the *classification loss*

$$\mathcal{L}_{\text{CLS}}(\theta) = -\sum_{m=1}^{M} \log p_\theta\left(\hat{y}_m = y_m \mid \mathbf{x}_m\right). \tag{3}$$

3.1 Shaping with language

Now assume that during training we have for each class $\mathcal{S}_n$ a set of J_n associated natural language descriptions $\mathcal{W}_n = \{\mathbf{w}_1, \dots, \mathbf{w}_{J_n}\}$. Each $\mathbf{w}_j$ should explain the relevant features of $\mathcal{S}_n$ and need not be associated with individual examples.[1] In Figure 1, we have one description $\mathbf{w}_1 = (\texttt{A}, \texttt{red}, \dots, \texttt{square})$.

Our approach is simple: we encourage f_θ to learn prototypes that can also decode the class language descriptions. Let $\tilde{\mathbf{c}}_n$ be the prototype formed by averaging the support *and* query examples of class n. Then define a language model g_ϕ (e.g., a recurrent neural network), which conditioned on

[1] If we have language associated with individual examples, we can regularize at the instance-level, essentially learning an image captioner. We did not observe major gains with instance-level supervision (vs class-level) in the tasks explored here, in which case class-level language is preferable, since it is much easier to obtain. There are likely tasks where instance-level supervision is superior, which we leave for future work.

$\tilde{\mathbf{c}}_n$ provides a probability distribution over descriptions $g_\phi(\hat{\mathbf{w}}_j \mid \tilde{\mathbf{c}}_n)$ with a corresponding *natural language loss*:

$$\mathcal{L}_{\text{NL}}(\theta, \phi) = -\sum_{n=1}^{N} \sum_{j=1}^{J_n} \log g_\phi(\mathbf{w}_j \mid \tilde{\mathbf{c}}_n), \quad (4)$$

i.e. the total negative log-likelihood of the class descriptions across all classes in the task. Since $\mathcal{L}_{\text{NL}}$ depends on parameters θ through the prototype $\tilde{\mathbf{c}}_n$, this objective should encourage our model to better represent the features expressed in language.

Now we jointly minimize both losses:

$$\underset{\theta,\phi}{\arg\min} \left[\mathcal{L}_{\text{CLS}}(\theta) + \lambda_{\text{NL}} \mathcal{L}_{\text{NL}}(\theta, \phi)\right], \quad (5)$$

where the hyperparameter λ_{NL} controls the weight of the natural language loss. At test time, we simply discard g_ϕ and use f_θ to classify. We call our approach *language-shaped learning* (LSL; Figure 1).

3.2 Relation to L3

L3 (Andreas et al., 2018) has the same basic components of LSL, but instead defines the concepts $\mathbf{c}_n$ to be embeddings of the language descriptions themselves, generated by an additional recurrent neural network (RNN) encoder h_η: $\mathbf{c}_n = h_\eta(\mathbf{w}_n)$. During training, the ground-truth description is used for classification, while g_ϕ is trained to produce the description; at test time, L3 *samples* candidate descriptions $\hat{\mathbf{w}}_n$ from g_ϕ, keeping the description most similar to the images in the support set according to the similarity function s (Figure 1).

Compared to L3, LSL is simpler since it (1) does not require the additional embedding module h_η and (2) does not need the test-time language sampling procedure.[2] This also makes LSL much faster to run than L3 in practice: without the language machinery, LSL is up to 50x faster during inference in our experiments.

4 Experiments

Here we describe our two tasks and models. For each task, we evaluate LSL, L3, and a prototype network baseline trained without language (Meta; Figure 1). For full details, see Appendix A.

ShapeWorld. First, we use the ShapeWorld (Kuhnle and Copestake, 2017) dataset used by Andreas et al. (2018), which consists of 9000 training, 1000 validation, and 4000 test tasks (Figure 2).[3] Each task contains a single support set of $K = 4$ images representing a visual concept with an associated (artificial) English language description, generated with a minimal recursion semantics representation of the concept (Copestake et al., 2016). Each concept is a spatial relation between two objects, each object optionally qualified by color and/or shape, with 2-3 distractor shapes present. The task is to predict whether a query image $\mathbf{x}$ belongs to the concept.

For ease of comparison, we report results with models identical to Andreas et al. (2018), where f_θ is the final convolutional layer of a fixed ImageNet-pretrained VGG-16 (Simonyan and Zisserman, 2015) fed through two fully-connected layers:

$$f_\theta(\mathbf{x}) = \text{FC}(\text{ReLU}(\text{FC}(\text{VGG-16}(\mathbf{x})))). \quad (6)$$

However, because fixed ImageNet representations may not be the most appropriate choice for artificial data, we also run experiments with convolutional networks trained from scratch: either the 4-layer convolutional backbone used in much of the few-shot literature (Chen et al., 2019), as used in the Birds experiments we describe next, or a deeper ResNet-18 (He et al., 2016).

This is a special binary case of the few-shot learning framework, with a single positive support class $\mathcal{S}$ and prototype $\mathbf{c}$. Thus, we define the similarity function to be the sigmoid function $s(a, b) = \sigma(a \cdot b)$ and the positive prediction $P(\hat{y} = 1 \mid \mathbf{x}) = s(f_\theta(\mathbf{x}), \mathbf{c})$. g_ϕ is a 512-dimensional gated recurrent unit (GRU) RNN (Cho et al., 2014) trained with teacher forcing. Through a grid search on the validation set, we set $\lambda_{\text{NL}} = 20$.

Birds. To see if LSL can scale to more realistic scenarios, we use the Caltech-UCSD Birds dataset (Wah et al., 2011), which contains 200 bird species, each with 40–60 images, split into 100 train, 50 validation, and 50 test classes. During training, tasks are sampled dynamically by selecting N classes from the 100 train classes. K support and 16 query examples are then sampled from each class (similarly for val and test). For language, we use the descriptions collected by Reed et al. (2016), where

[2] LSL is similar to the "Meta+Joint" model of Andreas et al. (2018), which did not improve over baseline. However, they used separate encoders for the support and query examples, with only the support encoder trained to predict language, resulting in overfitting of the query encoder.

[3] This is a larger version with 4x as many test tasks for more stable confidence intervals (see Appendix A).

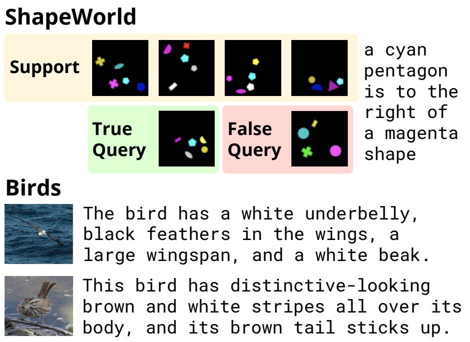

ShapeWorld
Support
a cyan pentagon is to the right of a magenta shape
True Query
False Query
Birds
The bird has a white underbelly, black feathers in the wings, a large wingspan, and a white beak.
This bird has distinctive-looking brown and white stripes all over its body, and its brown tail sticks up.

Figure 2: Example language and query examples for ShapeWorld and Birds.

AMT crowdworkers were asked to describe individual images of birds in detail, without reference to the species (Figure 2).

While 10 English descriptions per image are available, we assume a more realistic scenario where we have *much less* language available only at the class level: removing associations between images and their descriptions, we aggregate D descriptions for each class, and for each K-shot training task we sample K descriptions from each class n to use as descriptions $\mathcal{W}_n$. This makes learning especially challenging for LSL due to noise from captions that describe features only applicable to individual images. Despite this, we found improvements with as few as $D = 20$ descriptions per class, which we report as our main results, but also vary D to see how efficiently the models use language.

We evaluate on the $N = 5$-way, $K = 1$-shot setting, and as f_θ use the 4-layer convolutional backbone proposed by Chen et al. (2019). Here we use a learned bilinear similarity function, $s(a, b) = a^\top \mathbf{W} b$, where $\mathbf{W}$ is learned jointly with the model. g_ϕ is a 200-dimensional GRU, and with another grid search we set $\lambda_{\text{NL}} = 5$.

5 Results

Results are in Table 1. For ShapeWorld, LSL outperforms the meta-learning baseline (Meta) by 6.7%, and does at least as well as L3; Table 2 shows similar trends when f_θ is trained from scratch. For Birds, LSL has a smaller but still significant 3.3% increase over Meta, while L3 drops below baseline. Furthermore, LSL uses language more efficiently: Figure 3 shows Birds performance as the captions per class D increases from 1 (100 total) to 60 (6000 total). LSL benefits from a remarkably small number of captions, with limited gains past 20; in contrast, L3 requires much more language to even approach baseline performance.

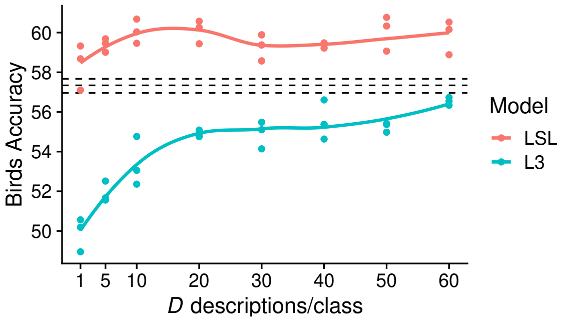


Figure 3: Varying the descriptions per class, D, for Birds. Each dot is a separate independently trained model. The dashed lines represent independently trained baselines (Meta).

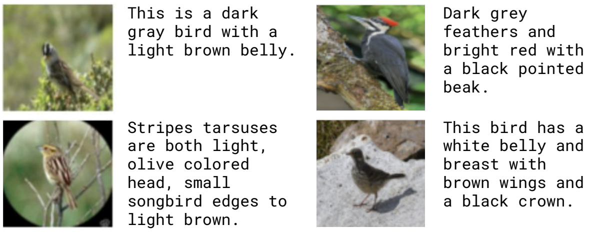

This is a dark gray bird with a light brown belly.
Dark grey feathers and bright red with a black pointed beak.
Stripes tarsuses are both light, olive colored head, small songbird edges to light brown.
This bird has a white belly and breast with brown wings and a black crown.

Figure 4: Examples of language generated by the L3 decoder g_ϕ for Birds validation images. Since the LSL decoder is identically parameterized, it generates similar language.

In the low-data regime, L3's lower performance is unsurprising, since it must generate language at test time, which is difficult with so little data. Example output from the L3 decoder in Figure 4 highlights this fact: the language looks reasonable in some cases, but in others has factual errors (*dark gray bird; black pointed beak*) and fluency issues.

These results suggest that any benefit of L3 is likely due to the regularizing effect that language has on its embedding model f_θ, which has been trained to predict language for test-time inference; in fact, the discrete bottleneck actually hurts in some settings. By using only the regularized visual representations and not relying exclusively on the generated language, LSL is the simpler, more efficient, and overall superior model.

Table 1: Test accuracies (± 95% CI) across 1000 (ShapeWorld) and 600 (Birds) tasks.

	ShapeWorld	Birds ($D = 20$)
Meta	60.59 ± 1.07	57.97 ± 0.96
L3	66.60 ± 1.18	53.96 ± 1.06
LSL	**67.29 ± 1.03**	**61.24 ± 0.96**

Table 2: ShapeWorld performance with different f_θ architectures trained from scratch.

f_θ	Conv4	ResNet-18
Meta	50.91 ± 1.10	58.73 ± 1.08
L3	62.28 ± 1.09	67.90 ± 1.07
LSL	**63.25** ± **1.06**	**68.76** ± **1.02**

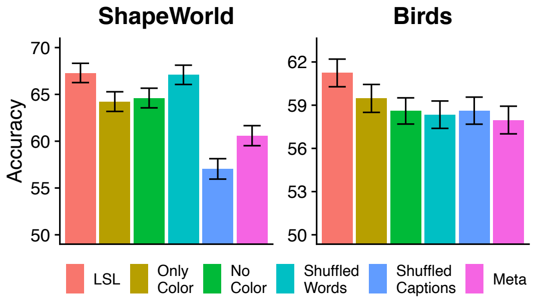


Figure 5: Language ablations. Error bars are 95% CIs.

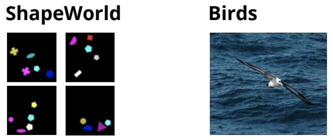
5.1 Language ablation

To identify which aspects of language are most helpful, in Figure 5 we examine LSL performance under ablated language supervision: (1) keeping only a list of common color words, (2) filtering out color words, (3) shuffling the words in each caption, and (4) shuffling the captions across tasks (see Figure 6 for examples).

We find that while the benefits of color/no-color language varies across tasks, neither component provides the benefit of complete language, demonstrating that LSL leverages both colors and other attributes (e.g. size, shape) described in language. Word order is important for Birds but surprisingly unimportant for ShapeWorld, suggesting that even with decoupled colors and shapes, the model can often infer the correct relation from the shapes that consistently appear in the examples. Finally, when captions are shuffled across tasks, LSL for Birds does no worse than Meta, while ShapeWorld suffers, suggesting that language is more important for ShapeWorld than for the fine-grained, attribute-based Birds task.

6 Discussion

We presented LSL, a few-shot visual recognition model that is regularized with language descriptions during training. LSL outperforms baselines across two tasks and uses language supervision more efficiently than L3. We find that if a model is trained to expose the features and abstractions in language, a linguistic bottleneck on top of these language-shaped representations is unnecessary, at least for the kinds of visual tasks explored here.



	ShapeWorld	Birds
Original	a cyan pentagon is to the right of a magenta shape	The bird has a white underbelly, black feathers in the wings, a large wingspan, and a white beak.
Only Color	cyan magenta	white black white
No Color	a pentagon is to the right of a shape	The bird has a underbelly feathers in the wings, a large wingspan, and a beak.
Shuffled Words	shape right the is a pentagon a of cyan to magenta	The , a and a . , beak bird in wingspan feathers large the black white underbelly has , white a wings
Shuffled Captions	a green square is below a triangle	This magnificent fellow is almost all black with a red crest, and white cheek patch.

Figure 6: Examples of ablated language supervision for the Birds and ShapeWorld tasks.

The line between language and sufficiently rich attributes and rationales is blurry, and recent work (Tokmakov et al., 2019) suggests that similar performance gains can likely be observed by regularizing with attributes. However, unlike attributes, language is (1) a more natural medium for annotators, (2) does not require preconceived restrictions on the kinds of features relevant to the task, and (3) is abundant in unsupervised forms. This makes shaping representations with language a promising and easily accessible way to improve the generalization of vision models in low-data settings.

Acknowledgments

We thank Pang Wei Koh, Sebastian Schuster, and Dan Iter for helpful discussions and feedback, Mike Wu and Jacob Andreas for discussions and code, and our anonymous reviewers for insightful comments. This work was supported by an NSF Graduate Research Fellowship for JM, a SAIL-Toyota Research Award, and the Office of Naval Research grant ONR MURI N00014-16-1-2007. Toyota Research Institute (TRI) provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

Reproducibility

Code, data, and experiments are available at `https://github.com/jayelm/lsl` and on CodaLab at `https://bit.ly/lsl_acl20`.

References

Jacob Andreas, Dan Klein, and Sergey Levine. 2018. Learning with latent language. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2166–2179.

Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. 2019. Learning to understand goal specifications by modelling reward. In *International Conference on Learning Representations (ICLR)*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9539–9549.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. In *International Conference on Learning Representations (ICLR)*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Sahil Chopra, Michael Henry Tessler, and Noah D Goodman. 2019. The first crank of the cultural ratchet: Learning and transmitting concepts through language. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society*, pages 226–232.

Ann A Copestake, Guy Emerson, Michael Wayne Goodman, Matic Horvat, Alexander Kuhnle, and Ewa Muszynska. 2016. Resources for building applications with dependency minimal recursion semantics. In *International Conference on Language Resources and Evaluation (LREC)*.

Jeff Donahue and Kristen Grauman. 2011. Annotator rationales for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1402.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Ranzato Marc'Aurelio, and Tomas Mikolov. 2013. DeViSE: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2121–2129.

Nelson Goodman. 1955. *Fact, fiction, and forecast*. Harvard University Press, Cambridge, MA.

Albert Gordo and Diane Larlus. 2017. Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6589–6598.

Prasoon Goyal, Scott Niekum, and Raymond J. Mooney. 2019. Using natural language for reward shaping in reinforcement learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2385–2391.

Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. Training classifiers with natural language explanations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1884–1895.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Xiangteng He and Yuxin Peng. 2017. Fine-grained image classification via combining vision and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5994–6002.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. 2016. Generating visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19.

Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. 2018. Grounding visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 264–279.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Alexander Kuhnle and Ann Copestake. 2017. Shapeworld-a new test methodology for multimodal language understanding. *arXiv preprint arXiv:1704.04517*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! Leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4932–4942, Florence, Italy.

Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 49–58.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4077–4087.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 935–943.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2017. Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1527–1536.

Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. 2019. Learning compositional representations for few-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6372–6381.

Michael Tomasello. 1999. *The Cultural Origins of Human Cognition*. Harvard University Press, Cambridge, MA.

Vladimir Vapnik and Akshay Vashist. 2009. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The Caltech-UCSD Birds-200-2011 dataset.

Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro O Pinheiro. 2019. Adaptive cross-modal few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4848–4858.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using annotator rationales to improve machine learning for text categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference (NAACL-HLT)*, pages 260–267.

A Model and training details

A.1 ShapeWorld

f_θ. Like Andreas et al. (2018), f_θ starts with features extracted from the last convolutional layer of a fixed ImageNet-pretrained VGG-19 network (Simonyan and Zisserman, 2015). These 4608-d embeddings are then fed into two fully connected layers $\in \mathbb{R}^{4608\times512}, \mathbb{R}^{512\times512}$ with one ReLU nonlinearity in between.

LSL. For LSL, the 512-d embedding from f_θ directly initializes the 512-d hidden state of the GRU g_ϕ. We use 300-d word embeddings initialized randomly. Initializing with GloVe (Pennington et al., 2014) made no significant difference.

L3. f_θ and g_ϕ are the same as in LSL and Meta. h_η is a unidirectional 1-layer GRU with hidden size 512 sharing the same word embeddings as g_ϕ. The output of the last hidden state is taken as the embedding of the description $\mathbf{w}^{(t)}$. Like Andreas et al. (2018), a total of 10 descriptions per task are sampled at test time.

Training. We train for 50 epochs, each epoch consisting of 100 batches with 100 tasks in each batch, with the Adam optimizer (Kingma and Ba, 2015) and a learning rate of 0.001. We select the model with highest epoch validation accuracy during training. This differs slightly from Andreas et al. (2018), who use different numbers of epochs per model and did not specify how they were chosen; otherwise, the training and evaluation process is the same.

Data. We recreated the ShapeWorld dataset using the same code as Andreas et al. (2018), except generating 4x as many test tasks (4000 vs 1000) for more stable confidence intervals.

Note that results for both L3 *and the baseline model* (Meta) are 3–4 points lower than the scores reported in Andreas et al. (2018) (because performance is lower for all models, we are not being unfair to L3). This is likely due to differences in model initialization due to our PyTorch reimplementation and/or recreation of the dataset with more test tasks.

A.2 Birds

f_θ. The 4-layer convolutional backbone f_θ is the same as the one used in much of the few-shot literature (Chen et al., 2019; Snell et al., 2017). The model has 4 convolutional blocks, each consisting of a 64-filter 3x3 convolution, batch normalization, ReLU nonlinearity, and 2x2 max-pooling layer. With an input image size of 84×84 this results in 1600-d image embeddings. Finally, the bilinear matrix $\mathbf{W}$ used in the similarity function has dimension 1600×1600.

LSL. The resulting 1600-d image embeddings are fed into a single linear layer $\in \mathbb{R}^{1600\times200}$ which initializes the 200-d hidden state of the GRU. We initialize embeddings with GloVe. We did not observe significant gains from increasing the size of the decoder g_ϕ.

L3. f_θ and g_ϕ are the same. h_η is a unidirectional GRU with hidden size 200 sharing the same embeddings as g_ϕ. The last hidden state is taken as the concept $\mathbf{c}_n$. 10 descriptions per class are sampled at test time. We did not observe significant gains from increasing the size of the decoder g_ϕ or encoder h_η, nor increasing the number of descriptions sampled per class at test.

Training. For ease of comparison to the few-shot literature we use the same training and evaluation process as Chen et al. (2019). Models are trained for 60000 episodes, each episode consisting of one randomly sampled task with 16 query images per class. Like Chen et al. (2019), they are evaluated on 600 episodes. We use Adam with a learning rate of 0.001 and select the model with the highest validation accuracy after training.

Data. Like Chen et al. (2019), we use standard data preprocessing and training augmentation: ImageNet mean pixel normalization, random cropping, horizontal flipping, and color jittering.