On the Limits of Multi-modal Meta-Learning with Auxiliary Task Modulation Using Conditional Batch Normalization

Jordi Armengol-Estapé[*†1]   **Vincent Michalski**[*2,3]   **Ramnath Kumar**[4]
Pierre-Luc St-Charles[2]   **Doina Precup**[2,5,7]   **Samira Ebrahimi Kahou**[2,6,7]
[1]University of Edinburgh   [2]Mila   [3]Université de Montréal   [4]Google Research
[5]McGill University   [6]University of Calgary   [7]CIFAR
jordi.armengol.estape@ed.ac.uk

Abstract

Few-shot learning aims to learn representations that can tackle novel tasks given a small number of examples. Recent studies show that cross-modal learning can improve representations for few-shot classification. More specifically, language is a rich modality that can be used to guide visual learning. In this work, we experiment with a multi-modal architecture for few-shot learning that consists of three components: a classifier, an auxiliary network, and a bridge network. While the classifier performs the main classification task, the auxiliary network learns to predict language representations from the same input, and the bridge network transforms high-level features of the auxiliary network into modulation parameters for layers of the few-shot classifier using conditional batch normalization. The bridge should encourage a form of lightweight semantic alignment between language and vision which could be useful for the classifier. However, after evaluating the proposed approach on two popular few-shot classification benchmarks we find that a) the improvements do not reproduce across benchmarks, and b) when they do, the improvements are due to the additional compute and parameters introduced by the bridge network. We contribute insights and recommendations for future work in multi-modal meta-learning, especially when using language representations.
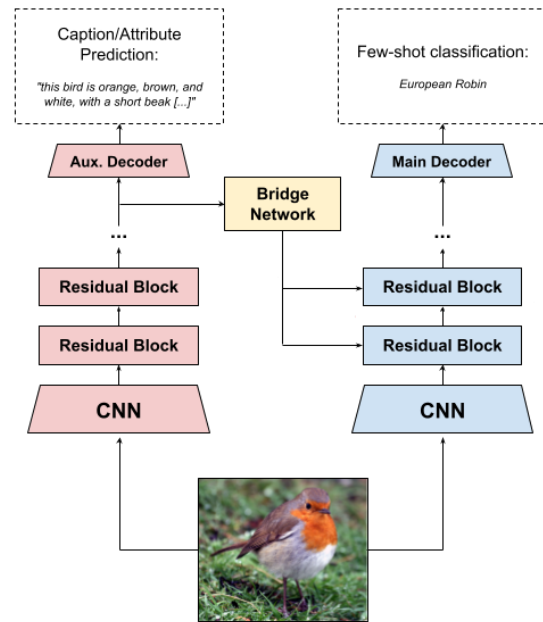
Figure 1: Architectural overview of the method we experimented with. It consists of three components: a classifier, an auxiliary network, and a bridge network. The few-shot classifier and auxiliary network receive the same input example. The bridge network transforms high-level features of the auxiliary network into modulation parameters for layers of the few-shot classifier through conditional batch normalization.

1 Introduction

It is widely recognized that humans can learn new concepts based on very little supervision, i.e. with few examples (or "shots"), and generalize these concepts to unseen data (Lake et al., 2011). Recent advances in deep learning on the other hand have mostly relied on datasets with large amounts of labeled examples, primarily due to overfitting

concerns in low data regimes. Although the development of better data augmentation and regularization techniques can alleviate these concerns, many researchers now assume that future breakthroughs in low data regimes will emerge from either transferring generic models pretrained on very large datasets with unsupervised objectives (Devlin et al., 2019; Brown et al., 2020), or from meta-learning, i.e. "learning-to-learn". Here, we study the problem of learning-to-learn in few shots by using an embedding space in which we perform classification using a similarity metric. In this meta-

---

learning setting, a model is trained on a handful of labeled examples at a time under the assumption that it will learn how to correctly project examples of different classes and generalize this knowledge to unseen labels at test time.

Although this setting is often used to illustrate the remaining gap between human capabilities and machine learning, we could argue that the lack of context poses a serious disadvantage to machine learning models. Indeed, these models typically work based on a single-pass analysis while humans can first look at and understand contextual information before trying to interpret new classes (Swingley, 2010). It has been observed many times in the past that training models with contextual information such as auxiliary modalities can help build a more robust task-independent feature space (Ruder, 2017; Elliott et al., 2016; Radford et al., 2021). Auxiliary tasks however often require large support datasets with good label distributions and a delicate adjustment of network capacity to really help improve performance on the main task (Alonso and Plank, 2016). Multi-modal information can be difficult to process using a simple backbone architecture due to the varied structure and high-level nature of some typically used modalities, although recent Transformer-based works have shown it's possible, albeit costly (Jaegle et al., 2021). We refer to the Appendix A for a more comprehensive study of the related work.

We propose studying whether multitask learning with multi-modal objectives could be beneficial for few-shot learning even with commonly-used low-capacity feature extraction backbones, and without weight sharing between the main and auxiliary tasks. We study a way to condition multiple layers of our main feature extractor using an embedding produced by an entirely separate auxiliary network working on the same input data. The conditioning is applied to normalization layer parameters using a bridge network and it helps specialize the representations produced by the main feature extractor without affecting its architecture. Our idea here is to mimic the way humans can leverage context to help solve the recognition problem by combining low-level and high-level cues. In other words, we allow the main feature extractor to decide ahead of time what it should focus on based on task-level contextual knowledge. The proposed model architecture is illustrated in Figure 1. In contrast with previous works that also studied feature extraction conditioning and multi-modal learning, our

approach is simple and can be applied to any feature extractor with batch normalization layers. The bridged-parallel-network design we propose also simplifies the feature alignment process since both branches process the same input data. Finally, the need for only a single input modality at test time leads to a more practical design for downstream applications.

However, after evaluating the proposed approach on two popular few-shot classification benchmarks we find that a) the improvements do not reproduce across benchmarks, and b) when they do, the improvements are due to the additional compute and parameters introduced by the bridge network. We contribute insights and recommendations for future work in multi-modal meta-learning, especially when using language representations.

2 Proposed method

In this section, we formulate conditional batch normalization in the context of few-shot learning. We propose a model, SimpAux, with two feature extractors that predict high-level (language-based) attributes of images as well as their semantic class. The embeddings of the attribute prediction pipeline (or "auxiliary" pipeline) are used to condition the batch normalization layers of the main visual feature extractor, which is based on a ProtoNet architecture (Snell et al., 2017). More specifically, we use ProtoNet++ improvement introduced in Oreshkin et al. (2018), with a Resnet-12 (He et al., 2016), which is a common choice in few-shot learning settings (e.g. Oreshkin et al., 2018; Jiang et al., 2019). The conditioning happens through a bridge connection composed of dense layers that translates the auxiliary embedding into batch normalization statistics. These three components are shown in Figure 1 and are described in the following subsections. Note that we use the same input modality (imagery) for the auxiliary and main feature extractors. However, our method is not limited to this modality: it was primarily chosen for compatibility with existing datasets. We refer the reader to Appendix B for a review of the fundamental ideas required to better understand our proposed few-shot learning solution from a technical standpoint.

2.1 Auxiliary visual processing

The auxiliary network in our proposed approach is agnostic of the main network's architecture and task. To simplify comparisons with a wider num-

ber of few-shot learning methods and to improve practicality, we formulate this network as a second visual processing pipeline that converts the same images fed to the main network into different embeddings. The multi-modal nature of our overall design comes from the supervised task used to learn the auxiliary network's embeddings: its goal is to predict language-based information from the images. More specifically, we experimented with predicting a) Attributes, available in datasets such as CUB-200-2011 (Wah et al., 2011), with cross-entropy, soft F1, or multi-label soft margin loss functions, and b) caption embeddings, with cosine similarity loss on the sentence embeddings emitted by SentenceBERT (Reimers and Gurevych, 2019). We ended up using multi-label soft margin loss as it was simpler and the other approaches did not provide significant improvements. However, for the datasets for which attributes were not available, we resorted to the sentence embeddings approach. As for the auxiliary model architecture itself, we also use a ResNet-12 as we do for the classifier.

2.2 Conditioning bridge

The role of the conditioning bridge is to transform the embeddings generated by the auxiliary network into an array of γ and β parameters that can be used in the various batch normalization layers of the main network. In contrast with late representation fusion strategies, e.g. the one of De Vries et al. (2017), this strategy allows for the early modulation of the main feature extraction pipeline with the high-level semantic information extracted from the auxiliary pipeline. Our hypothesis is that this information provides adequate context to dynamically adapt the main feature extractor while keeping its original architecture intact (and thus simple).

Since the distribution of the input representation varies at each layer of that network, the normalization parameters also need to be unique for each layer. We define our bridge as a multilayer perceptron (MLP) with a fixed intermediate representation size and an output size that corresponds to twice the total size of batch normalization layers in the main network (to account for both γ and β).

3 Experimental results

We evaluate SimpAux against the baseline, ProtoNet++, (the improved version of ProtoNets suggested in Oreshkin et al. (2018)) on two popular few-shot learning benchmarks,

CUB-200-2011 (Wah et al., 2011) and mini-ImageNet (Vinyals et al., 2016) in 5-shot learning settings, using attributes for CUB and embeddings on synthetic captions for Mini-Imagenet for the auxiliary visual processing network. We refer to Appendix C for additional implementation details.

Table 1 shows the results of ProtoNet++ and SimpAux on CUB 5-shot. Our model clearly outperforms the baseline by a margin of around 1.5 points in accuracy.

Model	Accuracy (%)
ProtoNet++	88.5 ± 0.5
SimpAux	90.0 ± 0.7

Table 1: Accuracy on CUB. Each model was trained with five random seeds. Reported is the mean accuracy with 95% confidence intervals on 600 randomly generated test episodes.

These positive results on CUB showed the promise of the proposed approach. However, in the case of Mini-Imagenet 5-shot, in Table 2 we can see the results of ProtoNet++ and SimpAux on Mini-Imagenet. In this case, the baseline slightly outperforms the proposed method, but recall that here we are using synthetic captions.

Model	Accuracy (%)
ProtoNet++	75.4 ± 0.4
SimpAux	74.9 ± 0.1

Table 2: Accuracy on mini-ImageNet. Each model was trained with five random seeds. Reported is the mean accuracy with 95% confidence intervals on 600 randomly generated test episodes.

Finally, to test the hypothesis that the reason why our approach outperforms the baseline in CUB but not in ImageNet is the quality of the captions, we design an ablation study. We introduce a variation of SimpAux in which we use the exact same bridge network, but without input from the auxiliary network, to see whether the improvements are actually coming from the captions information or the additional compute and parameters from the bridge network. We find that there is no significant improvement over this variant when using the captions, suggesting that the improvement comes from the additional compute and the parameters provided by the bridge network.

4 Discussion and recommendations

From the experimental results, we conclude that a) the improvements provided by SimpAux do not reproduce across benchmarks, and b) when these improvements do indeed take place, they seem to be due to the additional compute and parameters provided by the bridge network. We hypothesize three non-mutually exclusive reasons why image captioning as auxiliary task modulation via conditional batch normalization did not consistently improve the results: 1) a lack of quality of the image captions, attributes, or caption embeddings, 2) the limited impact of the conditional batch normalization approach, and 3) the difficulty of learning the auxiliary task. While improving the quality of captions, attributes and caption embeddings with better annotations or more powerful models could alleviate 1), the following recommendations and observations look at other aspects involved in this work.

Caution when evaluating systems with auxiliary multi-modal information. Training models with contextual information such as auxiliary modalities have been shown to build a more robust task-independent feature space (Ruder, 2017; Elliott et al., 2016; Radford et al., 2021). However, spurious improvements with multi-modal data are not new. For instance, Elliott (2018) empirically raises doubts about whether existing multi-modal translation systems, combining visual and textual data, actually make use of the visual information. Similarly, we have seen the other way around: it is perfectly possible to outperform a unimodal baseline with a multi-modal one without actually making use of the textual information; SimpAux's improvements in CUB were due to the additional parameters introduced by the bridge network. Thus, we recommend extra care when concluding that multi-modal information helps in a certain task, which is definitely possible but could be due to other factors.

Importance of implementation details. We experimented with different activation functions, including ReLU (Agarap, 2018), SELU (Klambauer et al., 2017), and SiLU (Hendrycks and Gimpel, 2016; Ramachandran et al., 2017). We found that SiLU consistently yielded slightly better results across benchmarks and settings. Ensuring that weight decay was not applied to bias parameters, which is not the default behavior in PyTorch (Paszke et al., 2019), also turned out to be key to re-producing few-shot works originally implemented in Tensorflow (Abadi et al., 2015).

Hyperparameter search. In the hyperparameter search, we generally observed consistent results. However, we also observed a few outliers, which can be particularly extreme under certain settings in few-shot learning, and if used as empirical evidence, could totally change the conclusions. Thus, we reiterate the need for reporting averages and variances instead of the results of a single run, and also recommend caution at extracting certain conclusions when performing large-scale hyperparameter searches, as noted by Picard (2021).

Advantages of the proposed architecture. Our network architecture decouples task-specific branches: its bridge acts as a gate that selects relevant hints from the auxiliary network to influence the classification network. It is simpler than previous works that also studied feature extraction conditioning and multi-modal learning, and by design it requires a single input modality at test time, which simplifies practical deployments. SimpAux's architectural considerations are orthogonal to other few-shot learning research lines, and could be combined with them. Thus, we believe that, despite the limited success in the meta-learning setting, these architectural advantages could be a source of inspiration for future work.

Language-informed representations and few-shot learning. Without episodic learning, Radford et al. (2021) showed that language-informed visual representations can be successfully learned with large-scale supervised contrastive pretraining. Their approach, CLIP, obtains high-performance at zero-shot classification. Leveraging their pre-trained encoders could be interesting in the context of bootstrapping episodic learning with auxiliary tasks. It would however be difficult to guarantee that the classes used in few-shot settings have not been observed by CLIP during pretraining.

5 Conclusion

In this work, we have studied a new multi-modal architecture for few-shot learning consisting of an image classifier, an auxiliary network trained with image captions, and a modulating network based on conditional batch normalization to connect the two. While initially promising, we have observed the limits of this approach and how these limits could inform future research.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375.

Héctor Martínez Alonso and Barbara Plank. 2016. When is multitask learning effective? semantic sequence prediction under varying data conditions. *arXiv preprint arXiv:1612.02251*.

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. 2018. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*.

Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. 2016. Learning feed-forward one-shot learners. In *Advances in neural information processing systems*, pages 523–531.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. 2022. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. 2017. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604.

Tristan Deleu, Tobias Würfl, Mandana Samiei, Joseph Paul Cohen, and Yoshua Bengio. 2019. Torchmeta: A Meta-Learning library for PyTorch. Available at: https://github.com/tristandeleu/pytorch-meta.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. 2019. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*.

Desmond Elliott. 2018. Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2978, Brussels, Belgium. Association for Computational Linguistics.

Desmond Elliott, Douwe Kiela, and Angeliki Lazaridou. 2016. Multimodal learning and reasoning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*.

Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. 2017. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT press Cambridge.

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

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. 2021. Perceiver IO: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*.

Xiang Jiang, Mohammad Havaei, Farshid Varno, Gabriel Chartrand, Nicolas Chapados, and Stan Matwin. 2019. Learning to learn with conditional class dependencies. In *International Conference on Learning Representations*.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. 2011. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33.

Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. 2018. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pages 3664–3673. PMLR.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 721–731.

Frederik Pahde, Oleksiy Ostapenko, Patrick Jä Hnichen, Tassilo Klein, and Moin Nabi. 2019. Self-paced adversarial training for multimodal few-shot learning. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 218–226. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2017. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv:1709.07871*.

David Picard. 2021. Torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *CoRR*, abs/2109.08203.

Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. 2019. Transductive episodic-wise adaptive metric for few-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3603–3612.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2017. Searching for activation functions. *CoRR*, abs/1710.05941.

Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning.

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*, pages 49–58.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. 2018. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2483–2493.

Eli Schwartz, Leonid Karlinsky, Rogerio Feris, Raja Giryes, and Alex M Bronstein. 2019. Baby steps towards few-shot learning with multiple semantics. *arXiv preprint arXiv:1906.01905*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087.

Daniel Swingley. 2010. Fast mapping and slow mapping in children's word learning. *Language learning and Development*, 6(3):179–183.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. 2020. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*.

Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. 2020. Cross-domain few-shot classification via learned feature-wise transformation. *arXiv preprint arXiv:2001.08735*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*.

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. 2019. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, pages 1–12.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34.

Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro OO Pinheiro. 2019. Adaptive cross-modal few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4847–4857.

Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. 2020. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817.

Fang Zhao, Jian Zhao, Shuicheng Yan, and Jiashi Feng. 2018. Dynamic conditional networks for few-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–35.

Imtiaz Masud Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. 2020. Laplacian regularized few-shot learning. *arXiv preprint arXiv:2006.15486*.

A   Related Work

Network conditioning. Normalization layers have been used many times in the past as a means to influence the behavior of deep feature extractors. For example, early works in arbitrary style transfer studied how modulating instance normalization parameters could align representations across styles that are not already known at run time (Huang and Belongie, 2017; Ghiasi et al., 2017). The flexibility gained by this modulation strategy has been adopted to tackle many other problems where feature extractors must dynamically change their behavior at run-time. For example, De Vries et al. (2017) and Perez et al. (2017) use conditional normalization layers to manipulate feature extractors in a selective manner for visual question answering and reasoning tasks. In the few-shot learning literature, Oreshkin et al. (2018) apply a form of normalization conditioning for task-dynamic feature extraction. In their case, instances are first encoded with an "unconditioned" feature extractor, and the resulting embeddings are used to condition the same feature extractor in a subsequent pass. In contrast, we base our conditioning on auxiliary labels and formulate a single-pass inference process. We also do not impose any constraints on the architecture of the main or auxiliary networks, meaning one can be much smaller than the other if required by the limited size of the dataset.

Note that there are also alternative conditioning strategies for few-shot learning paradigms that do not involve normalization layers. For example, embeddings can be directly modulated by a second network stage that analyzes the contextual information from the task (Ye et al., 2020; Qiao et al., 2019). Popular feature extractor architectures can also be slightly modified by adding conditionally shifted neurons to adapt representations using context at prediction time (Munkhdalai et al., 2018). Alternatively, the entire parameter set of various convolutional layers inside the feature extractor can be inferred at prediction time using a parallel network (Bertinetto et al., 2016, 2018; Zhao et al., 2018). A recent approach has also been proposed by Chen et al. (2022) to adapt large-scale multi-modal transformer-based backbones. The downside to these solutions is the dependency on large networks that must learn complex modulation operations from the task context, or the use of a memory bank on which an attention mechanism can operate. In contrast, normalization conditioning is a more lightweight approach that is easier to learn in small data regimes due to the reduced complexity of the modulation factors (i.e. the normalization statistics).

Recent trends in few-shot learning. There have been far too many strategies proposed to tackle few-shot learning for us to inventory them here. For a survey and a modern taxonomy, we refer the reader to the work of Wang et al. (2020). Instead, we note that many researchers over the years have highlighted the lack of a universal evaluation methodology for these methods. Recent independent efforts have shown that many "state-of-the-art" solutions are actually quite fragile and can be outperformed by simple baselines when evaluated and compared properly (Chen et al., 2019; Dhillon et al., 2019;

Tian et al., 2020). All of these works found that simple CNN backbones trained using a cross entropy loss and then optionally fine-tuned on test time queries can deliver competitive performance with respect to recent models. Transductive learning using test time queries in particular has been recently re-explored as an effective solution for few-shot learning (Dhillon et al., 2019; Ziko et al., 2020). Such findings highlight that more research effort should be spent on model-agnostic robustness improvements and less on the introduction or tuning of new model architectures as well as their training regimes. Our work falls in line with this idea while also promoting the use of multi-modal labels for improved few-shot learning.

As for multi-modal few-shot learning itself: it is not a new approach to the problem, but it is also not a popular one, as typical benchmarks only focus on using only imagery as input. Nonetheless, multiple strategies have been proposed to help deal with data scarcity in few-shot learning. For example, Pahde et al. (2019) feed image captions to a generative model during training to obtain additional images of the target classes. Their method however relies on several pre-trained and notably hard-to-train model components. Xing et al. (2019) and Schwartz et al. (2019) also leverage caption data but instead combine visual and semantic representations to improve class discrimination in metric space. In contrast to our work, they rely on parallel feature extraction pipelines that are combined in a "late fusion" fashion, whereas we propose a way to modulate the entirety of any visual pipeline architecture with semantic information. Vuorio et al. (2019) applies a similar modulation idea to the model-agnostic, meta-learning (MAML) framework of Finn et al. (2017). In their case, they rely on the modulation layers proposed by Perez et al. (2017) to condition their main task network. Tseng et al. (2020) follow the same strategy to deal with domain generalization issues in few-shot learning. In comparison, our proposed auxiliary network is trained in a supervised cross-modal setting where its embeddings are used to modulate our main network. Also, since we apply modulation through batch normalization, our approach can handle data samples that do not possess auxiliary labels or captions.

B Background

Here, we review some of the fundamental ideas required to understand our proposed few-shot learning solution.

B.1 Episodic few-shot learning and ProtoNets

In episodic few-shot learning, an "episode" is represented as an N-way, K-shot classification problem where N is the number of examples per class and K the number of unique class labels. During training, the data in each episode is provided as a support set $S = \{(\boldsymbol{x}_{1,1}, \boldsymbol{y}_1), ..., (\boldsymbol{x}_{N,K}, \boldsymbol{y}_N)\}$ where $\boldsymbol{x}_{i,j} \in \mathbb{R}^D$ is the i-th instance of the j-th class, and $\boldsymbol{y}_j \in \{0, 1\}^K$ is its corresponding one-hot labeling vector. The goal in each episode is to optimize a function f that classifies new instances provided through a "query" set Q which contains instances of the same classes as S. This task is difficult because N is typically very small (e.g. 1 to 10), the classes change every episode, and the actual test set used to evaluate a model does not contain classes that were seen in support sets during training.

We build our solution on top of Prototypical Networks (ProtoNets; Snell et al., 2017), as it is now accepted as a good yet simple baseline. According to (Chen et al., 2019), it is more robust than other recent few-shot learning approaches and it generalizes well across various dataset domains. ProtoNets tackle few-shot learning by learning an embedding space where each class is represented by a cluster, or *prototype*. A prototype $\boldsymbol{c_k} \in \mathbb{R}^M$ for a class k is simply defined as the mean of the instance embeddings that belong to k, that is:

$$\boldsymbol{c_k} = \frac{1}{S_k} \sum_{(\boldsymbol{x}_{i,j}, \boldsymbol{y}_i) \in S_k} f(\boldsymbol{x}_{i,j}), \qquad (1)$$

where S_k is the support subset of all instances that belong to class k, and f is a learned function. Next, the probability of assigning a new instance x to a class k is computed via the softmax of the distance to all class prototypes:

$$p(y = k|\boldsymbol{x}) = \frac{\exp -d(f(\boldsymbol{x}), \boldsymbol{c_k})}{\sum_{k'} \exp -d(f(\boldsymbol{x}), \boldsymbol{c_{k'}})}, \qquad (2)$$

for any given distance function $d : \mathbb{R}^M \times \mathbb{R}^M \mapsto [0, +\infty)$.

B.2 Batch normalization and conditioning

Batch normalization was proposed by (Ioffe and Szegedy, 2015) as a solution to speed up training

by reducing the problem of coordinating weight updates across the different layers of a model. In short, batch normalization performs a reparameterization on the intermediate representations of a model so that assumptions regarding their spread and distribution in subsequent layers will be less affected by stochastic updates. More specifically, given a batch of n feature maps $B = \{z_1, ..., z_n\}$ with C channels each, batch normalization performs channel-wise reparameterization using

$$\text{BN}(z_{l,c}|B, \gamma, \beta) = \gamma_c \cdot \frac{z_{l,c} - \mu_c}{\sigma_c^2 + \epsilon} + \beta_c, \quad (3)$$

where γ and β are vectors of learned channel-wise parameters, ϵ is a constant used for numerical stability, and μ_c and σ_c^2 are the mean and variation values computed across batch and spatial dimensions of B.

Many researchers now recognize that batch normalization has beneficial side-effects on the landscape of the optimization problem (Goodfellow et al., 2016; Santurkar et al., 2018). These benefits have lead to the rapid adoption of this technique across the majority of new and popular model architectures. Consequently, the important role and ubiquitous nature of batch normalization make it an interesting target for the conditioning of models using auxiliary data. This idea was first introduced by De Vries et al. (2017): they inject visual concepts from natural language in a visual processing pipeline for VQA by manipulating batch normalization parameters. These parameters are influenced by the embeddings produced with a recurrent network. One advantage of this approach is that it can help learn how to dynamically specialize a model at test time without drastically increasing its overall number of learnable parameters. This advantage is very interesting in the context of few-shot learning where only small datasets prone to overfitting are considered.

C Other implementation details

Our ProtoNet backbone is the improved version of the original method (coined ProtoNet++) suggested by Oreshkin et al. (2018) that includes residual connections between convolution layers (Resnet-12). We implement the models and data loaders with PyTorch (Paszke et al., 2019) and Torchmeta (Deleu et al., 2019), a meta-learning library. We experimented with different activation functions, and SiLU (Ramachandran et al., 2017) yielded the best results.

In our experiments, we use the CUB-200-2011 (Wah et al., 2011) and mini-ImageNet (Vinyals et al., 2016) datasets. For CUB, we use the split of Chen et al. (2019) and also experiment with the captions collected by Reed et al. (2016). For mini-ImageNet, we use the setting proposed by Ravi and Larochelle (2016), with synthetic captions generated using an open-source implementation of a Transformer (Vaswani et al., 2017) for image captioning.[1]

Our implementation is publicly available on Github.[2]

---

[1] https://github.com/saahiluppal/catr
[2] https://github.com/jordiae/simpaux-release