DReCa: A General Task Augmentation Strategy for Few-Shot Natural Language Inference

Shikhar Murty **Tatsunori B. Hashimoto** **Christopher D. Manning**
Computer Science Department, Stanford University
{smurty,thashim,manning}@cs.stanford.edu

Abstract

Meta-learning promises few-shot learners that quickly adapt to new distributions by repurposing knowledge acquired from previous training. However, we believe meta-learning has not yet succeeded in NLP due to the lack of a well-defined task distribution, leading to attempts that treat datasets as tasks. Such an ad hoc task distribution causes problems of quantity and quality. Since there's only a handful of datasets for any NLP problem, meta-learners tend to overfit their adaptation mechanism and, since NLP datasets are highly heterogeneous, many learning episodes have poor transfer between their support and query sets, which discourages the meta-learner from adapting. To alleviate these issues, we propose DReCa (**D**ecomposing datasets into **Re**asoning **Ca**tegories), a simple method for discovering and using latent reasoning categories in a dataset, to form additional high quality tasks. DReCa works by splitting examples into label groups, embedding them with a finetuned BERT model and then clustering each group into reasoning categories. Across four few-shot NLI problems, we demonstrate that using DReCa improves the accuracy of meta-learners by 1.5–4%.

1 Introduction

A key desideratum for human-like understanding is few-shot adaptation. Adaptation is central to many NLP applications since new concepts and words appear often, leading to distribution shifts. People can effortlessly deal with these distribution shifts by learning these new concepts quickly and we would like our models to have similar capabilities. While finetuning large pre-trained transformers is one way to facilitate this adaptation, this procedure requires thousands of samples where humans might require only a few.

Can these pre-trained transformers be made to achieve few-shot adaptation? One promising direction is meta-learning (Schmidhuber, 1987; Bengio et al., 1997). Meta-learning promises few-shot classifiers that can adapt to new tasks by repurposing skills acquired from training tasks. An important prerequisite for successful application of meta-learning is a task-distribution from which a large number of tasks can be sampled to train the meta-learner. While meta-learning is very appealing, applications in NLP have thus far proven challenging due to the absence of a well-defined set of tasks that correspond to re-usable skills. This has led to less effective ad hoc alternatives, like treating entire datasets as tasks.

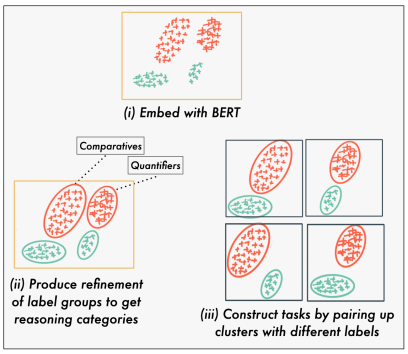


Figure 1: Overview of our approach. We embed all examples with BERT, and then cluster within each label group separately (red and green correspond to *entailment* and *not_entailment* respectively). Then, we group clusters from distinct label groups to form tasks.

Treating entire datasets as tasks has two major issues. The first issue is learner overfitting (Rajendran et al., 2020), where a meta-learner overfits its adaptation mechanism to the small number of training tasks, since there's only a small number of supervised datasets available for any NLP problem. Second, the heterogeneity of NLP datasets can lead to learning episodes that encourage memorization overfitting (Yin et al., 2020; Rajendran et al., 2020), a phenomenon where a meta-learner ignores the support set, and doesn't learn to adapt.

To improve the quality and quantity of tasks, we propose the simple approach of **D**ecomposing

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1113–1125
June 6–11, 2021.

datasets into **Re**asoning **Ca**tegories or DRECA. DRECA is a *meta* data augmentation strategy that takes as input a set of tasks (entire datasets), and then decomposes them to approximately recover some of the latent reasoning categories underlying these datasets, such as various syntactic constructs within a dataset, or semantic categories such as quantifiers and negation. These reasoning categories are then used to construct additional few-shot classification tasks, augmenting the original task distribution. We illustrate these steps in Fig. 1. DRECA first embeds the examples using a BERT model finetuned over all the datasets. We then run k-means clustering over these representations to produce a refinement of the original tasks.

Experiments demonstrate the effectiveness of our simple approach. As a proof of concept, we adapt the classic sine-wave regression problem from Finn et al. (2017) to mimic the challenges of the NLP setting, and observe that standard meta-learning procedures fail to adapt. However, a model that meta-learns over the underlying reasoning types shows a substantial improvement. Then, we consider the problem of natural language inference (NLI). We show that meta-learners augmented with DRECA improve over baselines by 1.5–4 accuracy points across four separate NLI few-shot problems without requiring domain-specific engineering or additional unlabeled data.

2 Related Work

Few-shot learning in NLP. The goal of learning from few examples has been studied for various NLP applications. Common settings include few-shot adaptation to new relations (Han et al., 2018), words (Holla et al., 2020), domains (Bao et al., 2020; Yu et al., 2018; Geng et al., 2019), and language pairs (Gu et al., 2018). Since these applications come with well-defined task distributions, they do not have the same overfitting challenges. On the other hand, many works deal with few-shot adaptation in settings with no clear task distribution (Dou et al., 2019; Bansal et al., 2020a) but do not address meta-overfitting, and thus are complementary to our work.

Overfitting and Task Augmentation. The memorization problem in meta-learning is studied in Yin et al. (2020) who propose a meta-regularizer to mitigate memorization overfitting, but don't study learner overfitting. Task augmentation for mitigating overfitting in meta-learners is first studied in Rajendran et al. (2020) in the context of few-shot label adaptation. Hsu et al. (2019) propose CACTUs, a clustering-based approach for unsupervised meta-learning in the context of few-shot label adaptation for images. While also based on clustering, CACTUs creates meta-learning tasks where the goal is to predict cluster membership of images, whereas our work is focused on using clusters to subdivide pre-existing tasks for mitigating meta-overfitting in NLP. Most closely related to our work is the SMLMT method from Bansal et al. (2020b). SMLMT creates new self-supervised tasks that improve meta-overfitting but this does not directly address the dataset-as-tasks problem we identify. In contrast, we focus on using clustering as a way to subdivide and fix tasks that already exist. This approach allows us to mitigate meta-overfitting *without* additional unlabeled data. In Section 6, we compare our model against SMLMT, and demonstrate comparable or better performance.

3 Setting

3.1 NLI

We consider the problem of Natural Language Inference or NLI (MacCartney and Manning, 2008; Bowman et al., 2015), also known as Recognising Textual Entailment (RTE) (Dagan et al., 2005). Given a sentence pair $x = (p, h)$ where p is referred to as the premise sentence, and h is the hypothesis sentence, the goal is to output a binary label[1] $\hat{y} \in \{0, 1\}$ indicating whether the hypothesis h is entailed by the premise p or not. For instance, the sentence pair *(The dog barked, The animal barked)* is classified as entailed, whereas the sentence pair *(The dog barked, The labrador barked)* would be classified as not entailed. As shown in Table 1, NLI datasets typically encompass a broad range of linguistic phenomena. Apart from the reasoning types shown in Table 1, examples may also vary in terms of their genre, syntax, annotator writing style etc. leading to extensive linguistic variability. Taken together, these factors of variation make NLI datasets highly heterogeneous.

3.2 Meta-Learning

The goal of meta-learning is to output a meta-learner $f : (\mathcal{S}_i, x_q^i) \mapsto \hat{y}$ that takes as input a *support* set $\mathcal{S}_i$ of labeled examples and a query point

[1] Since many of the NLI datasets we experiment with are 2-way NLI, we choose this formulation instead of 3-way NLI.

Example	Reasoning Category
A boy with the green jacket went back $\implies$ *A boy went back*	Restrictive Modifiers
A white rabbit ran $\implies$ *A rabbit ran*	Intersective Adjectives
Bill is taller than Jack $\not\Longrightarrow$ *Jack is taller than Bill*	Comparatives
The dog barked $\not\Longrightarrow$ *The dog did not bark*	Negation
The man went to the restaurant since he was hungry $\implies$ *The man was hungry*	Coreference Resolution
Bill is taller than Jack $\implies$ *Jack is not taller than Bill*	Negated Comparatives

Table 1: Some common reasoning types within NLI. These can also be composed to create new types.

x_q^i and returns a prediction $\hat{y}$. In the usual meta-learning setting, these support and query sets are defined as samples from a task $\mathcal{T}^i$, which is a collection of labeled examples $\{(x^i, y^i)\}$. In N-way k-shot adaptation, each $\mathcal{T}^i$ is an N-way classification problem, and f is given k examples per label to adapt. A simple baseline for meta-learning is to train a supervised model on labeled data from training tasks, and then finetune it at test time on the support set. This can be powerful, but is ineffective for very small support sets. A better alternative is episodic meta-learning, which explicitly trains models to adapt using training tasks

Episodic Training. In the standard setup for training episodic meta-learners, we are given a collection of training tasks. We assume that both train and test tasks are i.i.d. draws from a task distribution $p(\mathcal{T})$. For each training task $\mathcal{T}_i^{\text{tr}} \sim p(\mathcal{T})$, we create *learning episodes* which are used to train the meta-learner. Each learning episode consists of a support set $\mathcal{S}_i = \{(x_s^i, y_s^i)\}$ and a query set $\mathcal{Q}_i = \{(x_q^i, y_q^i)\}$. The goal of episodic meta-learning is to ensure that the meta-learning loss $\mathcal{L}(f(\mathcal{S}_i, x_q^i), y_q^i)$ is small on training tasks $\mathcal{T}_i^{\text{tr}}$. Since train tasks are i.i.d. with test tasks, this results in meta-learners that achieve low loss at test time.

Several algorithms have been proposed for meta-learning that follow this general setup, such as Matching Networks (Vinyals et al., 2016), MANN (Santoro et al., 2016), Prototypical Networks (Snell et al., 2017) and MAML (Finn et al., 2017). In this work, we use MAML as our meta-learner.

MAML. In MAML, the meta-learner f takes the form of gradient descent on a model $h_\theta \colon x \mapsto y$ using the support set,

$$f(\mathcal{S}_i, x_i^q) = h_{\theta_i'}(x_i^q) \tag{1}$$

where θ_i' denotes *task-specific* parameters obtained after gradient descent. The goal of MAML is to produce an initialization θ, such that after performing gradient descent on h_θ using $\mathcal{S}_i$, the updated model $h_{\theta_i'}$ can make accurate predictions on $\mathcal{Q}_i$. MAML consists of an *inner loop* and an *outer loop*. In the inner loop, the support set $\mathcal{S}_i$ is used to update model parameters θ, to obtain task-specific parameters θ_i',

$$\theta_i' = \theta - \alpha \nabla_\theta \sum_{(x_s^i, y_s^i) \in \mathcal{S}_i} \mathcal{L}(h_\theta(x_s^i), y_s^i). \tag{2}$$

These task-specific parameters are then used to make predictions on $\mathcal{Q}_i$. The outer loop takes gradient steps over θ such that *task-specific* parameters θ_i' perform well on $\mathcal{Q}_i$. Since θ_i' is itself a differentiable function of θ, we can perform this outer optimization using gradient descent,

$$\theta \leftarrow \text{Opt}\left(\theta, \nabla_\theta \sum_{(x_q^i, y_q^i) \in \mathcal{Q}_i} \mathcal{L}(h_{\theta_i'}(x_q^i), y_q^i)\right). \tag{3}$$

where Opt is an optimization algorithm typically chosen to be Adam. The outer loop gradient is typically computed in a mini-batch fashion by sampling a batch of episodes from distinct training tasks. The gradient $\nabla_\theta \mathcal{L}(h_{\theta_i'}(x_q^i), y_q^i)$ involves back-propagation through the adaptation step which requires computing higher order gradients. This can be computationally expensive so a first order approximation (FoMAML),

$$\nabla_\theta \mathcal{L}(h_{\theta_i'}(x_q^i), y_q^i) \approx \nabla_{\theta_i'} \mathcal{L}(h_{\theta_i'}(x_q^i), y_q^i) \tag{4}$$

is often used instead (Finn et al., 2017).

3.3 Meta-Learning for NLI

As mentioned earlier, training tasks in NLP are often entire datasets, leading to a small number of heterogeneous training tasks. Thus, to train a meta-learner for NLI, our training tasks $\mathcal{T}_i^{\text{tr}}$ are NLI *datasets*. At test time, we are given new datasets that we must adapt to, given a support set of randomly drawn examples from the dataset.

Meta Overfitting. Consider learning episodes sampled from an NLI dataset (Table 2). NLI datasets consist of a wide range of linguistic phenomena, and so we expect an episode to be comprised of a diverse set of reasoning categories. Such *heterogeneous* episodes can lead to scenarios where the support and query sets do not have any overlap in reasoning skills, causing the model to ignore the support set. This is known as memorization overfitting. Moreover, since we have a limited number of datasets, the meta-learner is exposed to a very small number of tasks at meta-training time causing it to generalize poorly to test tasks. This is known as learner overfitting (Rajendran et al., 2020).

NLI Example	Reasoning Category
Everyone has visited every person $\not\Longrightarrow$ *Jeff didn't visit Byron*	Negation, Quantifier
Generally, LC mail is lighter than AO mail $\Longrightarrow$ *AO mail is almost always heavier than LC mail*	Comparative, Quantifier
They've had their house that long $\not\Longrightarrow$ *They don't own the house and have never lived there*	Negation
Then he strolled gently in the opposite direction $\Longrightarrow$ *He wasn't walking in the same direction*	Negation
Query	
A white rabbit ran $\Longrightarrow$ *A rabbit ran*	Intersective adjective

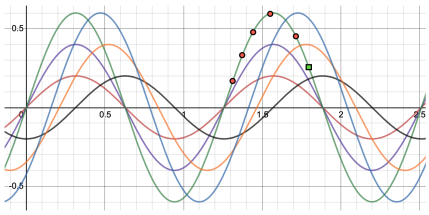
Table 2: Illustration of an episode sampled from a heterogeneous task. We can observe that there is no overlap between the support and query reasoning categories, leading to limited transfer.

4 An Illustration of Overfitting in Meta-Learning

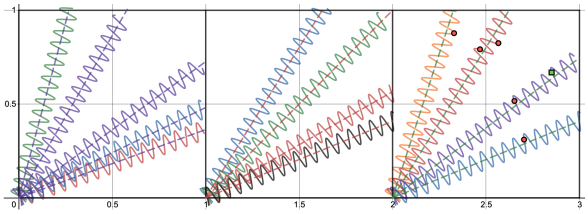
We illustrate meta overfitting challenges by modifying the classic sine-wave toy example for meta-learning from Finn et al. (2017).

Dataset. Consider the sine-wave regression problem from Finn et al. (2017) where each task corresponds to learning a sine wave mapping with a fixed amplitude and phase offset. As shown in Fig. 2(a), each support and query set consists of points drawn from the same sine wave mapping. The key observation here is that since support and query examples are drawn from the same mapping, we might expect a meta-learner to use the support set for adaptation. In the NLP case, since tasks are heterogeneous, support and query examples may belong to different reasoning categories. We instantiate this by letting support and query points come from different sine waves (Fig. 2(b)).

More formally, our construction consists of multiple datasets. Each dataset is defined as a unit square sampled from a 10×10 grid over $x_1 = [-5, 5]$ and $x_2 = [-5, 5]$. Within each dataset, we construct multiple reasoning categories by defining each reasoning category to be a sine wave with a distinct phase offset. This is illustrated in Fig. 2(b) where each unit square represents a dataset, and sine waves along distinct rays correspond to reasoning categories. The target label y for the regression task is defined for each category by a randomly sampled phase $\phi \in [0.1, 2\pi]$ and $y = \sin(\|x - \lfloor x \rfloor\|_2 - \phi)$. At meta-training time, we sample a subset of these 100 squares as our training datasets, and then evaluate few-shot adaptation to reasoning categories from held out datasets at meta-test time.



(a) 1D sine wave regression (Finn et al., 2017). Each task is a sine-wave with a fixed amplitude and phase offset.



(b) Three datasets from our 2D sine wave regression. Each dataset is a unit square with multiple reasoning categories; A reasoning category is a distinct sinusoid along a ray that maps $x = (x_1, x_2)$ to the value of the sine-wave y at that point.

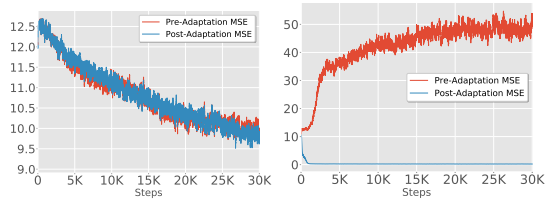
Figure 2: Comparing the classic 1D sine wave regression with our setting. For a randomly sampled episode, red dots mark support examples and the green square marks a query example. Notice how in 2(a), the support and query come from the same sine wave while in 2(b) they often come from different sine waves. This makes adaptation challenging, leading to memorization overfitting.

Experiments. We use similar hyperparameters as Finn et al. (2017) elaborated in Appendix A.1.

We start by considering MAML-BASE, a meta-learner that is trained directly over a dataset-based task distribution. Concretely, we define each training task as a dataset and randomly sample episodes to train the meta-learner. Note that since episodes are drawn uniformly at random from an entire dataset, we expect support and query sets to often contain points from disjoint reasoning categories

(Fig. 2(b)), making adaptation infeasible. Thus, we expect pre and post adaptation losses to be similar, which is indeed reflected in the learning curves in Fig. 3(a). We observe that the orange and blue lines, corresponding to pre and post adaptation losses respectively, almost overlap. In other words, the meta-learner ignores the support set entirely. This is what we mean by *memorization overfitting*.

Next we consider MAML-ORACLE, a meta-learner that is trained on tasks based on the underlying reasoning categories—distinct sine waves. Consequently, support and query sets are both drawn from the *same* sine wave, similar to Finn et al. (2017) making adaptation feasible. From Fig. 3(b), we observe large gaps between pre and post adaptation losses which indicates that memorization overfitting has been mitigated.



(a) (b)

Figure 3: Learning curves for MAML-BASE (a) and MAML-ORACLE (b). The lack of a gap between pre-adaptation (orange) and post-adaptation (blue) losses for MAML-BASE indicates strong memorization overfitting. A big gap for MAML-ORACLE indicates that this model learns to adapt.

These experiments confirm our hypothesis about the challenges of meta-learning with heterogeneous task distributions. Since NLI datasets require a wide range of skills, we might expect similar challenges on few-shot NLI as well.

5 DRECA

In this section, we introduce our approach for extracting reasoning categories for NLI. The key observation here is that high quality sentence pair representations, such as those obtained from a finetuned BERT model, can bring out the microstructure of NLI datasets. Indeed, the fact that pretrained transformers can be used to create meaningful clusters has been shown in other recent works (c.f. Aharoni and Goldberg (2020); Joshi et al. (2020)).

At a high level, the goal of DRECA is to take a heterogeneous task (such as a dataset) and produce a decomposed set of tasks. In doing so, we hope to obtain a large number of relatively homogeneous tasks that can prevent meta overfitting.

Given a training task $\mathcal{T}_i^{\text{tr}}$, we first group examples by their labels, and then embed examples within each group with an embedding function $\texttt{EMBED}(.)$. Concretely, for each N-way classification task $\mathcal{T}_i^{\text{tr}}$ we form groups $g_l^i = \{(\texttt{EMBED}(x_i^p), y_i^p) \mid y_i^p = l\}$. Then, we proceed to refine each label group into K clusters via k-means clustering to break down $\mathcal{T}_i^{\text{tr}}$ into groups $\{C^j(g_l^i)\}_{j=1}^K$ for $l = 1, 2, \ldots, N$.

These cluster groups can be used to produce K^N potential DRECA tasks.[2] Each task is obtained by choosing one of K clusters for each of the N label groups, and taking their union. At meta-training time, learning episodes are sampled uniformly at random from DRECA tasks with a probability λ and from one of the original tasks with probability $1 - \lambda$. Since our clustering procedure is based on finetuned BERT vectors, we expect the resulting clusters to roughly correspond to distinct reasoning categories. Indeed, when the true reasoning categories are known, we show in Section 7.2 that DRECA yields clusters that recover these reasoning categories almost exactly.

6 NLI Experiments

6.1 Datasets

We evaluate DRECA on 4 NLI few-shot learning problems which we describe below (more details in Appendix A.2.1). The first problem is based on synthetic data, while the other 3 problems are on real datasets and hence a good demonstration of the utility of our proposal.

HANS-FEWSHOT is a few-shot classification problem over HANS (McCoy et al., 2019), a synthetic diagnostic dataset for NLI. Each example in HANS comes from a hand-designed syntactic template which is associated with a fixed label (*entailment* or *not_entailment*). The entire dataset consists of 30 such templates which we use to define 15 reasoning categories. We then hold out 5 of these for evaluation, and train on the remaining 10. While this is a simple setting, it allows us to compare DRECA against an "oracle" with access to the underlying reasoning categories.

[2]Note that we do not instantiate the K^N tasks. Instead, we simply sample an episode from random chosen clusters from each label group.

COMBINEDNLI consists of a combination of 3 NLI datasets—MultiNLI (Williams et al., 2018), Diverse Natural Language Inference Collection (DNC; Poliak et al. (2018)) and Semantic Fragments (Richardson et al., 2020) for training. These training datasets cover a broad range of NLI phenomena. MultiNLI consists of crowdsourced examples, DNC consists of various semantic annotations from NLP datasets re-cast into NLI and Semantic fragments is a synthetic NLI dataset covering logical and monotonicity reasoning. Our objective is to train a single meta-learner that can then be used to make predictions on diverse NLP problems recast as NLI. To this end, we evaluate models trained on COMBINEDNLI on 2 datasets. In COMBINEDNLI-RTE, we evaluate on the RTE datasets (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009) as provided in GLUE (Wang et al., 2019). The RTE datasets consist of various IE and QA datasets recast as NLI. Second, we consider the QANLI dataset (Demszky et al., 2018) which recasts question answering into NLI. In particular, we consider RACE (Lai et al., 2017) and use gold annotations provided in Demszky et al. (2018) to transform it into an NLI dataset.

GLUE-SciTail where we train on all NLI datasets from GLUE (Wang et al., 2019) and evaluate on SciTail (Khot et al., 2018). This setting is comparable to Bansal et al. (2020b) with the difference that we only meta-train on the NLI subset of GLUE, whereas they meta-train on all GLUE tasks. We follow the same evaluation protocol as Bansal et al. (2020b) and report 2-way 4-shot accuracy.

6.2 Models

Non-Episodic Baselines. All non-episodic baselines train h_θ on the union of all examples from each $\mathcal{T}_i^{\text{tr}}$. In MULTITASK (FINETUNE), we additionally finetune the trained model on the support set of each test task. In MULTITASK (K-NN), each query example in the test task is labeled according to the nearest neighbor of the example in the support set. Finally, in MULTITASK (FINETUNE + K-NN), we first finetune the trained model on the support set and then label each query example based on its nearest neighbor in the support set.

Episodic Meta-learners. MAML-BASE is a MAML model where every task corresponds to a dataset. In the HANS-FEWSHOT setting where underlying reasoning categories are known, we also compare with an oracle model MAML-ORACLE which is trained over a mixture of dataset-based tasks as well as oracle reasoning categories. Finally, MAML-DRECA is our model which trains MAML over a mixture of the original dataset-based tasks as well as the augmented tasks from DRECA.

Evaluation. To control for variations across different support sets, we sample 5–10 random support sets for each test task. We finetune each of our models on these support sets and report means and 95% confidence intervals assuming the accuracies follow a Gaussian.

Training Details. We use first order MAML (FoMAML) for computational efficiency. We use BERT-base as provided in the transformers library (Wolf et al., 2019) as the parameterization for h_θ and $\texttt{EMBED}(;)$. The meta-training inner loop optimization involves 10 gradient steps with Adam, with a support set of 2 examples (2-way 1-shot) for all except GLUE-SciTail where the support set size is 8 (2-way 4-shot). We experiment with 4-shot adaptation on GLUE-SciTail to match the evaluation setup from Bansal et al. (2020b). The mixing weight λ is set to 0.5 for all our experiments. More details can be found in Appendix A.2.2.

Results. We report results on the synthetic HANS-FEWSHOT setting in Table 4, where we find that DRECA improves over all baselines. In particular, we observe an improvement of +6.94 over MULTITASK (FINETUNE + K-NN) and +4.3 over MAML-BASE. Moreover, we observe that MAML-DRECA obtains a comparable accuracy as MAML-ORACLE.

Next, we report results on our 3 real NLI settings in Table 3. Again, we find that DRECA improves model performance across all 3 settings: MAML-DRECA improves over MAML-BASE by +2.5 points on COMBINEDNLI-QANLI, +2.7 points on COMBINEDNLI-RTE and +1.6 points on GLUE-SciTail. On GLUE-SciTail, we compare against SMLMT (Bansal et al., 2020b) and find that MAML-DRECA improves over it by 1.5 accuracy points. However, we note that the confidence intervals of these approaches overlap, and also that (Bansal et al., 2020a) consider the entire GLUE data to train the meta-learner whereas we only consider NLI datasets within GLUE.

Model	CombinedNLI-QANLI	CombinedNLI-RTE	GLUE-SciTail
Multitask (Finetune)	69.66 ± 0.39	65.47 ± 3.19	75.80 ± 2.58
Multitask (k-NN)	68.97 ± 1.26	63.69 ± 6.65	69.76 ± 3.74
Multitask (Finetune + k-NN)	67.38 ± 2.61	66.52 ± 5.48	76.44 ± 1.77
MAML-Base	69.43 ± 0.81	72.61 ± 0.85	76.38 ± 1.25
SMLMT (Bansal et al., 2020b)	–	–	76.75 ± 2.08
MAML-DReCa	**71.98 ± 0.79**	**75.36 ± 0.69**	**77.91 ± 1.60**

Table 3: Results on NLI few-shot learning. We report the mean and 95% confidence intervals assuming accuracies follow a Gaussian. Bolded cells represent the best mean accuracy for the particular dataset. For all settings except GLUE-SciTail, we consider 2 way 1 shot adaptation. For GLUE-SciTail, we consider 2 way 4 shot adaptation. SMLMT numbers are taken directly from Bansal et al. (2020b).

Model	HANS-fewshot
Multitask (Finetune)	80.76 ± 1.83
Multitask (k-NN)	70.35 ± 2.29
Multitask (Finetune + k-NN)	80.59 ± 1.63
MAML-Base	82.64 ± 1.80
MAML-Oracle	86.74 ± 1.06
MAML-DReCa	**87.53 ± 2.38**

Table 4: Results on HANS-fewshot. We report the mean and 95% confidence intervals assuming accuracies follow a Gaussian.

Dataset	#Reasoning Categories	Cluster purity
HANS-fewshot	10	85.6%

Table 5: Measuring cluster purity. Our model is effective at recovering underlying reasoning types.

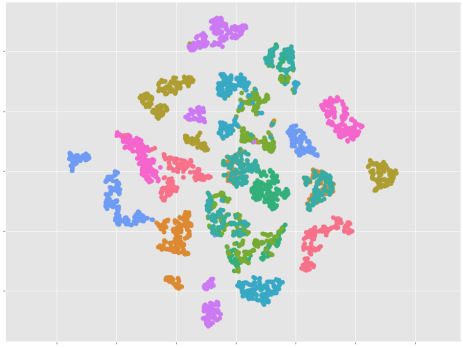


Figure 4: t-SNE plot of BERT vectors after finetuning on HANS. We see distinct clusters corresponding to the various reasoning categories.

7 Analysis

7.1 Visualizing the geometry of finetuned BERT on HANS-fewshot

We start by visualizing finetuned BERT embeddings used by DReCa for HANS-fewshot. As mentioned earlier, HANS consists of 30 manually defined syntactic templates which can be grouped into 15 reasoning categories. Following the procedure for `EMBED()` (details in Appendix A.2.2), we finetune BERT (Devlin et al., 2019) for 5000 randomly chosen examples from HANS. To obtain a vector representation for each example $x = (p, h)$, we concatenate the vector at the [CLS] token, along with a mean pooled representation of the premise and hypothesis. We then use t-SNE (Maaten and Hinton, 2008) to project these representations onto 2 dimensions. Each point in Fig. 4 is colored with its corresponding reasoning category, and we can observe a clear clustering of examples according to their reasoning category.

7.2 Evaluating DReCa Cluster Purity

To understand if reasoning categories can be accurately recovered with our approach, we measure the purity of DReCa clusters for HANS-fewshot where true reasoning categories are known. This is evaluated by computing the number of examples belonging to the majority reasoning type for each cluster and then dividing by the total number of examples. From Table 5, we observe high cluster purity which provides evidence that DReCa is able to recover true reasoning categories.

7.3 Distribution of linguistic phenomena across clusters

We seek to understand how different linguistic phenomena present in the overall population are distributed among various clusters. To perform this analysis, we focus on MultiNLI annotation tags from Williams et al. (2018). A subset of examples in MultiNLI are assigned tags based on the presence of certain keywords, e.g., *time words* like days of the week; *quantifiers* like *every, each, some;*

negation words like *no, not, never.* Additionally, certain tags are assigned based on the PTB (Marcus et al., 1993) parses of examples, e.g., presence or absence of adjectives/adverbs etc. For each annotation tag, we compute the fraction of examples labeled with that tag in each cluster. We visualize this for 10 annotation tags and indicate statistically significant deviations from the averages in Fig. 5. Statistical significance is measured with binomial testing with a Bonferroni correction to account for multiple testing.

For every annotation tag, we shade all clusters that contain a statistically significant deviation from the mean. For instance, there is a positive cluster with 2.5 fold enrichment in *Negation* tags compared to the average, and a negative cluster that contains over 4 times the population average of *Negation (Hyp only)* tags. Similarly, among *Conditionals*, we have positive clusters that contain 1.4 times the population average and a negative cluster containing half the population average. Interestingly, we find most positive clusters to be significantly poverished in *Adverb (Hyp only)* tags, while most negative clusters are enriched in these tags. This analysis presents evidence that clusters used by DRECA localize linguistic phenomena to a small number of clusters.

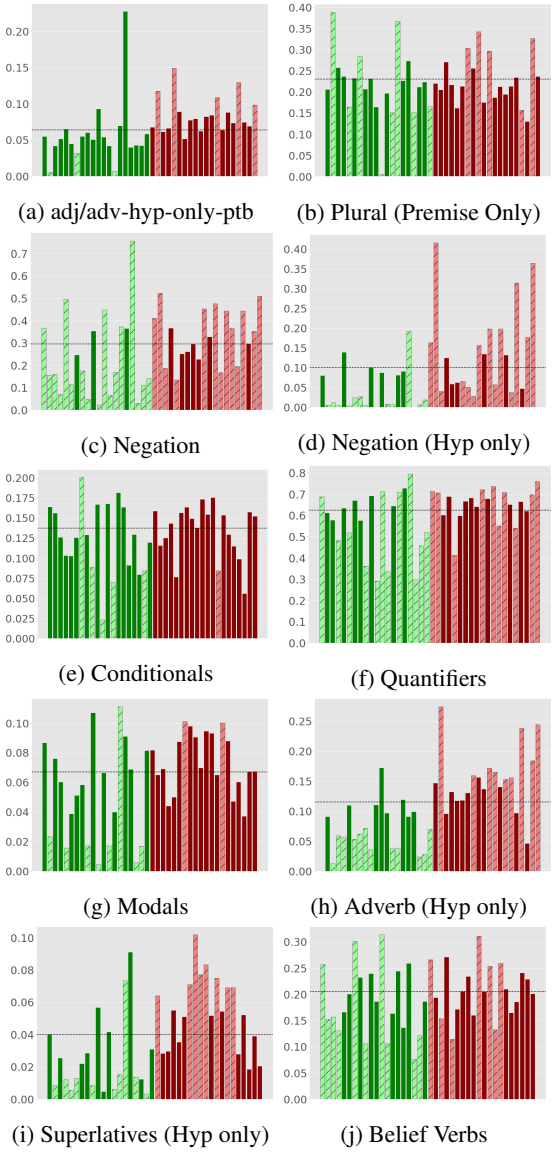


Figure 5: Fraction of cluster examples belonging to each linguistic annotation tag. Cluster groups corresponding to entailment (non-entailment) are colored green (red). The fraction of examples in the overall population is marked with a dashed line. We observe that many clusters have a statistically significant oversampling / undersampling of certain tags (drawn with a ligher color), according to a binomial test with a p-value of 0.05 under a Bonferroni correction for multiple testing.

8 Discussion

Comparing with CACTUs. Our work is most similar to CACTUs from Hsu et al. (2019). Apart from differences in the modality considered (text vs images), we differ in the following ways. Conceptually, Hsu et al. (2019) consider a fully unsupervised meta-learning setting where no labels are provided and use cluster IDs to induce labels, while our goal is to produce *additional* tasks in a supervised meta-learning setting. Second, CACTUs tasks are constructed by directly applying k-means on the entire training dataset while we apply k-means separately on each label group and construct tasks by choosing a cluster from each label group, leading to tasks with uniform label distribution. Finally, while CACTUs uses constructed tasks directly, our work using them to augment the original task distribution.

Number of examples in support set. All evaluation in this work considers small support sets where number of examples per label range from 1–4. This setting is somewhat restrictive since in practice, one might be able to get a few hundred examples for the target domain. These moderately sized support sets could *themselves* be heterogeneous where adapting a single learner might be hard. In such cases, we can use a similar clustering approach to separate out the support set into homogeneous tasks and adapt a separate learner for each task. These learners could then be plugged into a mixture of experts framework for making predictions.

Using k-means to produce task refinements. While we are able to get sufficiently homogeneous clusters with k-means, we note one shortcoming with this approach. Any input has multiple attributes / factors of variations and it may be possible to create a clustering for each factor. The current k-means based approach doesn't model this since we only produce a single clustering of the data. For instance, x_1 = *The man was walking in the park* $\implies$ *The man is not at home* and x_2 = *He went with his friends to the mall* $\implies$ *He is not at work* can belong to the same cluster if the underlying metric is based on reasoning types. At the same time, it could also be clustered with x_3 = *The man was walking in the park* $\not\Rightarrow$ *The woman is in the park* if the distance metric is based on lexical similarity. A promising direction for future work is to explore these *multi-clusterings* based on the various factors of variation present in the training data.

Non-meta learning based few-shot adaptation. In this work, we use tools from meta-learning to directly optimize for few-shot behavior. While not directly comparable to us, there have been many recent approaches to few-shot adaptation for NLP that do not use meta-learning. Brown et al. (2020) show impressive few-shot adaptation in large language models through "in-context learning" which is presumably acquired only through its language modeling objective,. Schick and Schütze (2020) train multiple models on lexical variations of a small support set and use these to label additional unlabeled examples from the target domain. These "self-labeled" examples are used to train a second model which can then make predictions on query examples. Finally, Gao et al. (2020) explore in-context learning of smaller language models for few-shot adaptation. In particular, they introduce a pipeline to identify useful prompts for the target domain, along with informative labeled examples to prepend as context for the LM.

9 Conclusion

Many papers point out fundamental challenges in creating systems that achieve human-like understanding of tasks like NLI. Here, we studied conditions under which systems can learn from extremely few samples. We believe that such systems would complement and enhance further study into more sophisticated challenges such as model extrapolation.

One of the main ingredients for successful application of meta-learning is a large number of high quality training tasks to sample learning episodes for the meta-learner. We observe that such a task distribution is usually not available for important NLP problems, leading to less desirable ad hoc attempts that treat entire datasets as tasks. In response, we propose DRECA as a simple and general purpose task-augmentation strategy. Our approach creates a refinement of the original set of tasks (entire datasets) that roughly correspond to linguistic phenomena present in the dataset. We show that training on a task distribution augmented with DRECA leads to consistent improvements on 4 NLI few-shot classification problems, matching other approaches that require additional unlabeled data and well as oracles that have access to the true task distribution.

10 Acknowledgements

We are grateful to Eric Mitchell, Robin Jia, Alex Tamkin, John Hewitt, Pratyusha Sharma and the anonymous reviewers for helpful comments. The authors would also like to thank other members of the Stanford NLP group for feedback on an early draft of the paper. This work has been partially supported by JD.com American Technologies Corporation ("JD") under the SAIL-JD AI Research Initiative and partially by Toyota Research Institute ("TRI"). This article solely reflects the opinions and conclusions of its authors and not JD, any entity associated with JD.com, TRI, or any other Toyota entity. Christopher Manning is a CIFAR Fellow.

11 Reproducibility

Code and model checkpoints will be available at https://github.com/MurtyShikhar/DReCA.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2020a. Learning to few-shot learn across diverse natural language classification tasks. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020b. Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 522–534.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In *International Conference on Learning Representations*.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–9.

Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. 1997. On the optimization of a synaptic learning rule. In Daniel S. Levine and Wesley R. Elsberry, editors, *Optimality in Biological and Artificial Networks?*, pages 265–287. Routledge.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proc. Text Analysis Conference (TAC'09*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *ArXiv*, abs/1809.02922.

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.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1192–1197, Hong Kong, China. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of Machine Learning Research*, 70:1126–1135.

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

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-07 Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. 2019. Generalized inner loop meta-learning. *arXiv preprint arXiv:1910.01727*.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Learning to learn to disambiguate: Meta-learning for few-shot word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4517–4533, Online. Association for Computational Linguistics.

Kyle Hsu, Sergey Levine, and Chelsea Finn. 2019. Unsupervised learning via meta-learning. In *International Conference on Learning Representations*.

Pratik Joshi, S. Aditya, Aalok Sathe, and M. Choudhury. 2020. TaxiNLI: Taking a ride up the NLU hill. *ArXiv*, abs/2009.14505.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *AAAI 2018*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

L. V. D. Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of COLING*, pages 521–528, Manchester, UK.

M. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Comput. Linguistics*, 19:313–330.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting Diverse Natural Language Inference Problems for Sentence Representation Evaluation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Janarthanan Rajendran, Alexander Irpan, and Eric Jang. 2020. Meta-learning requires meta-augmentation. In *Advances in Neural Information Processing Systems*, volume 33, pages 5705–5715. Curran Associates, Inc.

Kyle Richardson, H. Hu, L. Moss, and A. Sabharwal. 2020. Probing natural language inference models through semantic fragments. *ArXiv*, abs/1909.07521.

Adam Santoro, Sergey Bartunov, M. Botvinick, Daan Wierstra, and T. Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *ICML*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. ArXiv 2009.07118.

Jurgen Schmidhuber. 1987. Evolutionary principles in self-referential learning. on learning how to learn. Diploma thesis, Technische Universitat Munchen, Germany.

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

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. 2020. Meta-learning without memorization. In *International Conference on Learning Representations*.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. 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 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

A.1 2D Sine Wave Regression: Training Details

We use a two layer neural network with 40 dimensional hidden representations and ReLU nonlinearity as the parameterization of f. Following Finn et al. (2017), we take a single gradient step on the support set at meta-training time, and take 10 gradient steps at meta-test time. The MAML weights are optimized with Adam and the inner loop adaptation is done with SGD with a learning rate of 1e-2. For each outer loop update, we sample 5 tasks, and each episode consists of a support set of size 5, i.e., we consider 5 shot adaptation.

A.2 NLI Experiments

A.2.1 Dataset Generation Details

We describe details of how our datasets are generated below. Note that all our datasets are in English.

HANS-FEWSHOT. The reasoning categories we use are in Table 6. We randomly split these 15 reasoning categories in HANS into training and test tasks. For each task, we sample 500 examples split equally among `entailment` and `not_entailment`.

COMBINEDNLI. We first convert MultiNLI and Semantic Fragments into 2-way (`entailment` vs `not_entailment`) NLI problems by collapsing both `contradiction` and `neutral` labels into `not_entailment`, and resampling such that the dataset is balanced between the 2 label classes. To evaluate on QANLI, we use the RACE QA dataset and transform it into NLI as in Demszky et al. (2018). For RTE, we create a test set ourselves by randomly sampling examples from the RTE dataset provided by Wang et al. (2019). Dataset statistics can be found in Table 7.

GLUE-SciTail. We use MultiNLI, RTE, QNLI and SNLI as training data, following a similar procedure to convert 3-way NLI datasets into 2-way NLI. For evaluation, we use SciTail. Dataset statistics are in Table 8.

A.2.2 Training Details

Hyperparameters for all MAML models can be found in Table 9. We implement MAML in Pytorch using the higher library (Grefenstette et al., 2019). We take the BERT-base implementation from the huggingface library (Wolf et al., 2019) as the parameterization for h_θ, which has 110 million parameters

Task	Syntactic Templates
1	ce_adverb, cn_adverb
2	ce_embedded_under_verb, cn_embedded_under_verb
3	ce_conjunction, cn_disjunction
4	ce_after_since_clause, cn_after_if_clause
5	ce_embedded_under_since, cn_embedded_under_if
6	le_conjunction, ln_conjunction
7	le_passive, ln_passive
8	le_relative_clause, ln_relative_clause
9	le_around_prepositional_phrase, ln_preposition
10	le_around_relative_clause, ln_subject/object_swap
11	se_PP_on_obj, sn_PP_on_subject
12	se_relative_clause_on_obj, sn_relative_clause_on_subject
13	se_adjective, sn_NP/S
14	se_conjunction, sn_NP/Z
15	se_understood_object, sn_past_participle

Table 6: The 15 reasoning categories constructed from 30 HANS syntactic templates. For each reasoning category, we select 2 syntactic templates corresponding to `entailment` and `not_entailment` labels, giving us 15 binary classification tasks.

Dataset	#examples
Training Datasets	
MultiNLI	261798
DNC	440456
Semantic Fragments	33170
Test Datasets	
RTE	554
QANLI	1990

Table 7: Dataset statistics for COMBINEDNLI

DRECA. We first finetune BERT on the entire training dataset for 5 epochs. Then, we embed each example by concatenating the embeddding at the `[CLS]` token along with the mean pooled representation of the premise and the hypothesis to get a 2304-dimensional vector. Next, we apply PCA to select a subset of dimensions that explain 99% of the variance. We then apply k-means clustering after standardizing the resulting embeddings.

Dataset	#examples
Training Datasets	
MultiNLI	261798
SNLI	366832
QNLI	104732
RTE	2482
Test Dataset	
SciTail	2126

Table 8: Dataset statistics for GLUE-SciTail

Hyperparameter	Value
inner loop learning rate	5e-5
outer loop learning rate	5e-5
inner loop adaptation steps	10
inner / outer loop optimizer	Adam
max number of iterations	20000
episode size	32
episode batch size	5

Table 9: Hyperparameters for MAML based models.

B Discovering Reasoning Categories in 2D Sine Wave Regression

To discover latent reasoning categories for the 2D Sine Wave Regression dataset, we train a feedforward neural net (paramaterized similarly as h_θ) on the union of all the datasets, and use the final layer representation to cluster examples. We then use these clusters instead of the true reasoning categories to augment the original task distribution.

We now show learning curves on held out test tasks in Fig. 6. As expected, MAML-BASE fails to adapt to new reasoning categories, indicating that it was unable to acquire the required skill from its training tasks. On the other hand, MAML-ORACLE is able to adapt very well, which confirms our hypothesis that a large number of high quality tasks helps. Finally, we see that using MAML trained on the augmented task distribution is able to match the performance of the oracle.

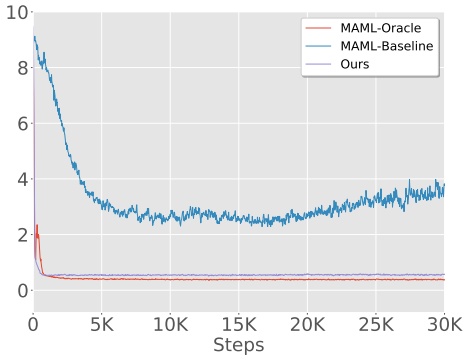


Figure 6: Learning curves on the 2D sine-wave regression task. We observe that the oracle meta-learner outperforms the baseline, and our proposed approach is able to bridge the gap.