A Channel Coding Benchmark for Meta-Learning

Rui Li[1], **Ondrej Bohdal**[2], **Hyeji Kim**[3], **Da Li**[1, 2], **Nicholas D. Lane**[1, 4], **and Timothy Hospedales**[1, 2]

[1] Samsung AI Center, Cambridge, UK [2] University of Edinburgh, UK
[3] University of Texas at Austin, US [4] University of Cambridge, UK
rui.li@samsung.com, ondrej.bohdal@ed.ac.uk, hyeji.kim@austin.utexas.edu, dali.academic@gmail.com,
ndl32@cam.ac.uk, t.hospedales@ed.ac.uk

Abstract

Meta-learning provides a popular and effective family of methods for data-efficient learning of new tasks. However, performance degrades in real-world settings where meta-learners must learn from a wide and potentially multi-modal distribution of training tasks; and when distribution shift exists between meta-train and meta-test task distributions. These issues are hard for the research community to study since the shape of task distributions, and shift between them are not straightforward to measure or control in standard benchmarks. To this end we consider the *channel coding* problem as a benchmark for meta-learning. Channel coding is a real-world application where task distributions naturally arise, and fast adaptation to new tasks is practically valuable. We show how to evaluate the variability of meta-learner performance with task distribution breadth and shift, which can be controlled in a coding benchmark, thus providing a tool for the community to drive research on practically robust and effective meta-learning methods going forward.

1 Introduction

Meta-learning, or learning-to-learn, aims to provide data-efficient learning of new tasks by training improved learning algorithms using a distribution over tasks. The promise of such data efficient learning has long inspired research (Thrun and Pratt 1998; Schmidhuber, Zhao, and Wiering 1996), and recently grown into a thriving research area in which rapid progress is being made (Finn, Abbeel, and Levine 2017; Zintgraf et al. 2019; Flennerhag et al. 2020; Hospedales et al. 2020). While performance has improved steadily, particularly on simple image recognition benchmarks, several fundamental outstanding challenges have been identified (Baz et al. 2021; Hospedales et al. 2020). In particular, state of the art meta-learners have been shown to suffer in realistic settings (Yu et al. 2019; Triantafillou et al. 2020) when required to generalize across a diverse rather than artificially narrow range of tasks – i.e. the task distribution is broad and multi modal; and when there is distribution shift between the (meta)training and (meta)testing tasks. These conditions are almost inevitable in real-world applications where, for example, robots should generalize to across the range of manipulation tasks of interest to humans (Yu et al. 2019), and image recognition systems should cover a realistically wide range of image types (Triantafillou et al. 2020). However, systematic study of these issues is hampered because conventional benchmarks do not provide a way to quantitatively measure or control the complexity or similarity of task distributions: Does an image recognition benchmark covering birds and airplanes provide a more or less complex task distribution to meta-learn than one covering flowers and vehicles? Is there greater task-shift if a robot trained to pick up objects must adapt to opening a drawer or throwing a ball? In this paper, we contribute to the future study of these issues by introducing a *channel coding* meta-learning benchmark, which enables finer control of task-distribution complexity and shift for study.

Channel coding is a classic problem in communications theory of how to encode/decode data to be transmitted over a capacity limited noisy channel so as to maximise the fidelity of the received transmission. While there is extensive theory on optimal codes for analytically tractable (e.g., Gaussian) channels, recent work has shown that codecs obtained by deep learning provide clearly superior performance on more complex realistic channels (Kim et al. 2018b,a). In this paper, we focus on learning the *decoder* for a fixed encoder[1]. Best deep channel coding however is achieved by training codecs tuned to the noise properties of a given channel. Thus, a highly practical meta-learning problem arises: Meta-learning a channel code learner on a distribution of training channels, which can rapidly adapt to the characteristics of a new channel. By way of example, the goal of meta-learning is now to enable the codec of a user's mobile wireless device to rapidly adapt its coding algorithm for best reception as she traverses different environments or switches on/off other sources of interference.

In this paper we introduce channel coding problems (Kim et al. 2018a,b) as tasks to study the impact of task distributions on contemporary meta-learners, as a complement to existing conventional meta-learning benchmarks (Yu et al. 2019; Triantafillou et al. 2020). We introduce benchmarks to study the dependence of meta-learning performance on the

[1]This is the practically relevant setting as communications standards defining the encoding protocol are not easy to change, but decoders can be upgraded without changing the standard.

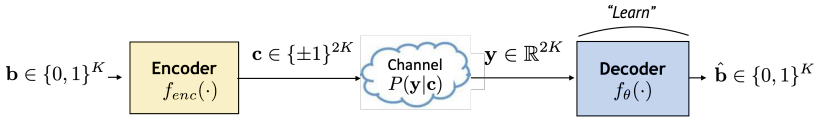

Figure 1: An illustration of the channel coding problem. We learn a channel decoder for a fixed encoder under various channel models.

width and complexity of training task distributions, as well as the degree of meta-train/meta-test task distribution shift. Furthermore, as these problems are relatively fast to train, while being of configurable difficulty (unlike other saturated starter benchmarks such as omniglot), we hope that this will provide a valuable prototyping tool for the meta-learning research community going forward.

2 Methods

2.1 Channel Coding Background

Channel coding is one of the key elements in a communication system, the role of which is to introduce controlled redundancy so that the receiver can reliably and efficiently recover the message from a *corrupted* received signal.

A typical channel coding system consists of an encoder and a decoder, as illustrated in Fig. 1. In this example a rate 1/2 channel encoder maps K message bits $\mathbf{b} \in \{0,1\}^K$ to a length-$2K$ transmitted signal $\mathbf{c} \in \{-1,1\}^{2K}$. In a more general setting a rate $1/r$ encoder maps $\mathbf{b} \in \{0,1\}^K$ to $\mathbf{c} \in \{\pm 1\}^{rK}$. Given the transmitted signal, a channel $p(\mathbf{y}|\mathbf{c})$ models the noise effect experienced by the signal in the communication medium based on mathematically expressed conditional distributions and outputs a noisy signal $\mathbf{y} \in \mathbb{R}^{2K}$. The decoder will in turn take the noisy signal as input and output estimations of the original message, i.e. $\hat{\mathbf{b}} = f_\theta(\mathbf{y}) \in \{0,1\}^K$. The reliability an encoder and decoder pair is measured by the probability of error, namely, Bit Error Rate (BER) defined as $\mathbb{P}(\hat{b}_k \neq b_k)$. We treat the decoding problem as a K-dimensional binary classification task for each of the ground-truth *message bits* b_k.

Neural decoder for convolutional codes We focus on learning a decoder for a fixed rate 1/2 convolutional encoder, as illustrated in Fig. 7 in the Appendix A. We are interested in this setting for two reasons. Firstly, the sequential nature of the convolutional encoding naturally aligns with convolutional neural networks. Secondly, efficient optimal decoders (e.g. Viterbi and BCJR algorithms) are known for a special class of channels, i.e. additive white Guassian noise (AWGN) channels. Practically, reliable and efficient decoders form an essential part of almost all kinds of communication systems, from wireline to wireless communications including both Wi-Fi and cellular. There has been significant interest in applying deep learning for channel decoding (and coding itself) (O'Shea and Hoydis 2017).

Adaptive neural decoder The channel $p(\mathbf{y}|\mathbf{c})$ can vary over time, and is unknown to the decoder. To help the decoder estimate the channel, pilot signals that are known messages $\mathbf{b}_{known}$ to the decoder are sent before the transmission begins, so that the decoder can extract channel information from $\mathbf{y}$ and $\mathbf{b}_{known}$. When modeling the decoder as a neural network instead of an analytical algorithm, the common practice is, for any given channel, to train the decoder using pilot signals $(\mathbf{y}, \mathbf{b}_{known})$ as ground-truths and their corresponding noisy received values as inputs. The optimization goal is to minimize a loss $\mathcal{L}$, which is typically in form of binary cross-entropy, with respect to decoder as f_θ

$$\theta^* = \text{argmin}_\theta \, \mathbb{E}_{\mathbf{b}_{known},\mathbf{y}} \mathcal{L}(\mathbf{b}_{known}, f_\theta(\mathbf{y})) \quad (1)$$

To ensure good performance as channel characteristics $p(\mathbf{y}|\mathbf{c})$ change due to e.g. weather or moving users, which always happen in realistic communications, adapting neural decoder f_θ to the evolving channel is necessary. Meta-learning is therefore a particularly promising tool to enable rapid decoder adaptation with few pilot codes, as confirmed by early evidence (Jiang et al. 2019). Conversely, channel coding provides a lightweight benchmark playground for contemporary meta-learners, allowing control of the task complexity and distribution-shift, thanks to the mathematical representability and tractability of channel models.

2.2 Meta-Learning

Meta-learning usually considers distributions over tasks for training and testing $p_{tr}(\mathcal{T})$ and $p_{te}(\mathcal{T})$. Each task $\mathcal{T}_i$ is associated with a dataset $D_i = \{\mathbf{x}_i^j, \mathbf{y}_i^j\}_{j=1}^J$, which we split into $D_i = D_i^{tr} \cup D_i^{val}$. We are interested in learning models f_θ of the form $\hat{\mathbf{y}} = f_\theta(\mathbf{x})$ using some algorithm $\mathcal{A}$ that minimizes a loss function $\mathcal{L}(\theta, D)$ on data D with respect to parameters θ. The algorithm itself is paramaterized by meta-parameter ϕ, i.e., $\theta^* = \mathcal{A}(D, \mathcal{L}, \phi)$. The goal of meta-learning is to find the parameters ϕ of algorithm $\mathcal{A}$ that leads to strong validation performance after learning.

$$\phi^* = \text{argmin} \, \mathbb{E}_{\substack{\mathcal{T} \sim p(\mathcal{T}) \\ (\mathcal{D}^{tr}, \mathcal{D}^{val}) \in \mathcal{T}}} \mathcal{L}(\mathcal{A}(D^{tr}, \mathcal{L}, \phi), D^{val}) \quad (2)$$

When datasets D^{tr} are small, this leads to meta-optimization for a data-efficient learner, as pioneered by MAML (Finn, Abbeel, and Levine 2017), which choses meta-parameter ϕ as the initial condition of the optimization for θ by $\mathcal{A}$. Once meta-learning is complete, we can draw a new task $\mathcal{T}' \sim p_{te}(\mathcal{T})$, and solve it efficiently as

$$\theta^* = \mathcal{A}(D', \mathcal{L}, \phi^*) \quad (3)$$

2.3 Coding benchmark for meta-learning

Constructing Task Distributions We consider four common *families* (also called *modalities*) of channel models and corresponding decoding tasks, namely, Additive White Gaussian Noise (AWGN), Bursty, Memory noise, and Multipath interference channels as detailed in Appendix B.

To define task distributions, we consider uni-modal and multi-modal settings. In the *single-family, uni-modal* case, a task distribution p corresponds to a specific channel class as discussed above, paramaterized by a continuous channel

parameter ω (e.g., the variance of additive noise or multi-path strength). The distribution of tasks in this family then depends on the prior over channel parameter ω, $p(\mathcal{T}) = \int_\omega p(\mathcal{T}|\omega)p(\omega)$. We can control the width of a task distribution by varying the width of the, e.g. uniform distributed, prior $p(\omega)$. In the *multi-family, multi-modal* case we can define a more complex task distribution as a mixture over multiple channel types p_k, each with its own distribution over channel parameters ω, $p(\mathcal{T}) = \sum_k \int_\omega \pi_k p_k(\mathcal{T}|\omega)p_k(\omega)$.

Quantifying task distribution shift and breadth We quantify the train-test task shift distance (Definition 1) and diversity (Definition 2) of each task, based on information theoretic measures. In a coding benchmark, we can control these scores by choosing appropriate set of channel models, which allows us to evaluate the variability of meta-learning with the task distribution breadth and shift. We demonstrate such examples in Section 3 (Fig. 3, 6).

Definition 1 (Train-Test Task-Shift $S(p_a(\mathcal{T}), p_b(\mathcal{T}))$) Simply *measuring* the shift between training and testing task distributions has previously been an open problem in meta-learning. However this becomes feasible to define for the channel coding problem. We quantify the distance of a test distribution $p_a(\mathcal{T})$ from a training distribution $p_b(\mathcal{T})$ using the Kullback–Leibler divergence (KLD) a.k.a. the relative entropy (Kullback and Leibler 1951). The KLD-based shift distance score is defined as:

$$S(p_a(\mathcal{T}), p_b(\mathcal{T})) := \mathbb{E}_{\mathbf{c}}[D_{KL}(p_a(\mathbf{y}_a|\mathbf{c})||p_b(\mathbf{y}_b|\mathbf{c}))],$$

where $p_a(\mathbf{y}_a|\mathbf{c})$ and $p_b(\mathbf{y}_b|\mathbf{c})$ denote the channels associated with $\mathcal{T}_a$ and $\mathcal{T}_b$, respectively. The distance is large if a testing distribution p_a introduces a very different distribution over received messages $\mathbf{y}$ for a given code $\mathbf{c}$ compared to training p_b, and zero if they induce the same distribution.

Definition 2 (Diversity Score $D(\mathcal{T})$) The diversity score of a task distribution $p(\mathcal{T})$ is defined as mutual information between the channel parameter ω and the received signal $\mathbf{y}$:

$$D(\mathcal{T}) = \mathbb{E}_{\mathbf{c}}[I(\omega; \mathbf{y}|\mathbf{c})],$$

where ω denotes the channel parameter (latent variable) for the task distribution, i.e., $p(\mathbf{y}|\mathbf{c}) = \int_\omega p(\mathbf{y}|\mathbf{c}, \omega)p_\omega(\omega)$. We will see that this metric will quantify amenability to meta-learning. Intuitively, decoding can benefit from meta-learning when knowing the task (channel parameter) conveys more information about the messages $\mathbf{y}$.

Estimation of scores In order to estimate shift-distance and diversity scores, we generate samples according to the corresponding channel distributions and use Kraskov-Stögbauer-Grassberger (KSG) estimator (Kraskov, Stögbauer, and Grassberger 2004), the k-nearest neighbor based estimator for entropy and mutual information (Kraskov, Stögbauer, and Grassberger 2004; Steeg 2014) to compute each of these scores given the samples.

Discussion While we denote each channel to learn as a '*task*', we note that in our scenario each task shares the same encoding and label-space of messages to recognize. As such our goal could also be understood as few-shot supervised domain adaptation by meta-learning. Thus we will consider the simple domain generalisation baseline of conventionally learning a decoder on all data from $p_{tr}(\cdot)$ and applying it directly to tasks in $p_{te}(\cdot)$ without adaptation.

3 Experiments

We first evaluate the impact of training distribution diversity on meta-learning performance, followed by that of train-test task distribution shift. Before analysing the results, we introduce the details of our experiments.

Dataset and Task Design We consider a wide range of channel scenarios that are described in Appendix C. To facilitate evaluation, we create a dataset of codewords and multiple transmitted messages under each channel model. For each of the benchmark scenarios, we have created a dataset with 200 randomly sampled noise setups, from the noise family specific to the scenario. Each noise setup has 50 randomly generated true codes ("classes") with 50 examples (noisy received messages) for each type. When generating a meta-training task, we randomly sample a noise set-up and then randomly sample N codes with K support examples and L target or query examples. This makes the tasks similar to the standard N-way K-shot problems, but instead of a classification problem, we solve a fast-adaptation problem. For meta-testing, we have another dataset with 50 manually specified noise setups and randomly generated 50 codes with 50 examples each. The meta-testing dataset is shared across various scenarios. Meta-testing tasks are generated in the same way as meta-training tasks. All of the datasets are small enough to easily fit into the GPU memory, allowing fast experimentation.

Meta-Learning Algorithms We have evaluated a variety of meta-learning approaches, both popular ones such as MAML and Reptile, but also state-of-the-art ones such as MetaCurvature or KFO. We have used the implementations provided by *learn2learn* library (Arnold et al. 2020). In particular, we compare the following algorithms:

1. **Vanilla** does not use meta-learning and directly trains a conventional model on the union of meta-training tasks. It can be considered as the *domain generalization* baseline for our benchmark, where simply aggregating all source data is a strong baseline (Li et al. 2017a).
2. **MAML** (Finn, Abbeel, and Levine 2017) is a meta-learner that aims to learn an initial condition for few-shot optimisation by backpropagating through a few steps of gradient descent, exploiting higher-order gradients.
3. **MAML FO** is the First-Order approximation to MAML introduced in Finn, Abbeel, and Levine (2017), which saves computation by avoiding higher order gradients.
4. **Reptile** (Nichol, Achiam, and Schulman 2018) is an efficient first-order alternative to MAML that is based on moving the initial weights towards the weights obtained after fine-tuning on a task. On some problems it is able to match MAML performance.
5. **ANIL** (Almost No Inner Loop, (Raghu et al. 2020)) is a simplification of MAML where only the task-specific network head (classifier) is included in the inner-loop updates. If ANIL performs similarly well as MAML, it sug-

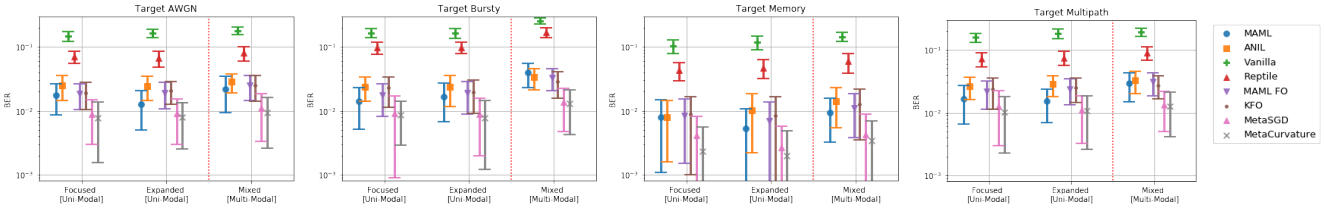

Figure 2: Meta-testing on AWGN, Bursty, Memory and Multipath task families (subplots). Y-axis is Bit Error Rate (BER, lower is better). Bars indicate meta-test standard-deviations. Left of the dashed line: Within-family setting compares uni-modal training distributions focused or widely distributed around the testing distribution. Right of the dashed line: Across-family setting where the training distributions is a mixture of all four task families.

gests that feature-reuse is the dominant factor, rather than rapid-tuning from the meta-learned initialization.

6. **MetaSGD** (Li et al. 2017b) is an extended version of MAML where the meta-learner learns to update the direction as well as the learning rate for each parameter together with the initialization of the neural network.
7. **KFO** (Meta Kronecker Factorized Optimizer, Arnold, Iqbal, and Shal (2019)) uses Kronecker factorization to transform the gradients and obtain more expressive and non-linear meta-optimizers. This transformation has helped improve the performance of MAML on various computer vision benchmarks.
8. **MetaCurvature** (Park and Oliva 2019) builds on MAML and learns a curvature matrix together with initial parameters of the model. MetaCurvature has been shown to outperform MAML and MetaSGD in generalization.

Note that our channel-coding problem is well-suited for gradient-based meta-optimizers. In contrast, it would be difficult to use metric-based approaches such as Prototypical Nets (Snell, Swersky, and Zemel 2017) or Relation Nets (Sung et al. 2018) because we do not solve a new classification problem where we could create a class prototype to which we would compare query examples. Instead, we focus on the problem of rapid adaptation to a new type of channel.

Hyperparameters and Architecture All approaches used the same hyperparameters and architecture for consistency. We used Adam optimizer with a meta-learning rate of 0.001 for the outer-loop, SGD with fine-tuning learning rate of 0.1 for the inner-loop consisting of 5 adaptation steps, 10 tasks in a meta-batch and 50000 meta-training iterations. Each task consisted of 5 different adaptation types ("classes"), with 5 support and 15 target examples. The ground truth messages are 10 bits long, encoded by the $1/2$ rate convolutional encoder. Hence the message input to the decoder has shape $1 \times 10 \times 2$. We fix the decoder architecture as a CNN with 4 layers, 64 filters, kernel size 3, and stride 2. The CNN is followed by a linear fully-connected layer of size 64×10.

3.1 Impact of Training Distribution Diversity on Meta-Learning Performance

In this first section we investigate how meta-learners cope with task distributions of varying breadth and complexity, since previous studies have suggested that capacity could be a limiting factor of existing meta-learners (Yu et al. 2019; Rusu et al. 2019). We would like meta-learners to be capable of learning from a broad range of auxiliary tasks, without requiring the auxiliary task distribution to be carefully constructed in advance for similarity to each given target task.

Setup In these experiments, we fix the meta-testing task distribution $p_{te}(\mathcal{T})$ to ensure comparability, and then evaluate performance when the training distribution $p_{tr}(\mathcal{T})$ is focused around the testing condition '*focused*' vs when it is spread more broadly around the testing distribution '*expanded*'. To expand the training distribution in the single-family/uni-modal case, we use a wider prior on the channel parameter $p_{tr}^{expand}(\omega) = \text{Unif}(a-\delta, b+\delta)$ vs $p_{te}(\omega) = \text{Unif}(a, b)$ when constructing task distributions as discussed in Section 2.3. In the multi-modal case, we use a single channel family for p_{te} and a multi-modal mixture of families for p_{tr} composed of all the channel models introduced in Section 2.3 including the testing distribution p_{te}.

Results Fig. 2 summarizes the results for meta-learning on training distributions of varying widths for both uni-modal (left or red line) and multi-modal (right of red line) conditions, and for different target channels (subplots). From the results, see we can that: (i) All meta-learners surpass the vanilla baseline in every case, confirming the value of meta-learning based adaptation to channel type. (ii) Among the meta-learners, Reptile is worst; MAML, FO-MAML, KFO and ANIL perform similarly in the middle; and MetaSGD and MetaCurvature perform best indicating the value of learning not just initial condition but update direction. (iii) In the within-family case (left group), focusing the meta-training distribution on the meta-testing condition (x-axis: focused) vs a diverse meta-training regime (x-axis: expanded) does not visibly meta-testing performance on our log-performance scale. In the across-family case (right, x-axis: mixed), transferring from a multi-modal training distribution to a specific testing distribution incurs a small but visible difference to performance. This suggests that meta-learner capacity for fitting a multi-modal training distribution does impact performance (Vuorio et al. 2019), albeit in a minor way.

To further understand these results we compute the breadth of each training regime as measured by its *diversity* (Section 2.3). Note that we can measure the diversity of both focused/expanded (uni-modal) and mixed (multi-modal) training regimes with the same metric. As expected,

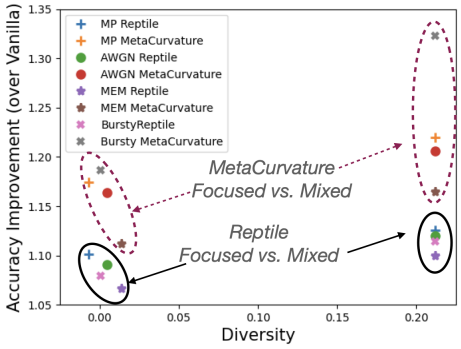

Figure 3: Connection between task distribution diversity score and benefit of meta-learning for the best and worst meta-learners, i.e. MetaCurvature and Reptile. Symbols: Different target channels. X-axis: Diversity scores of the channel; Y-axis: accuracy gain over vanilla. Results are grouped by circles on a per-learner (dashed circle for MetaCurvature and solid circle for Reptile) and per-train regime basis (Focused on the left and Mixed on the right).

the mixed regimes lead to higher diversity. Fig. 3 plots the margin between example meta-learners and the vanilla non-adaptive baseline against the diversity score of the channel. We can see that *the benefit provided by meta-learning increases with more diverse training regimes*. Intuitively, the more the channel configuration parameters determine the output message distribution, the more potential benefit there is from meta-learning how to adapt to a given channel.

3.2 Impact of Train-Test Distribution Shift on Meta-Learning Performance

In this section, we study how each of the baseline learners' performance depends on distribution shift between training $p_{tr}(\mathcal{T})$ and testing $p_{te}(\mathcal{T})$ task distributions.

Within-Family Setup We first consider the uni-modal within task family case, where we create distribution shift by setting $p_{tr}(\omega) \neq p_{te}(\omega)$. Specifically, we define two task distributions using two different non-overlapping uniform priors on ω corresponding to different channel SNRs in the Bursty channel. We then train our meta-learners on each, and evaluate them on a range of task parameters ω that are both in- and out-of-domain with respect to the training distribution.

Results Fig. 4 shows the results of generalizing across a range of meta-testing tasks (x-axis), for models learned within the each of the two specified training domains. Note that the 'difficulty' of the shown testing tasks is non-uniform i.e. higher SNR tasks are easier. This means that other things being equal we expect worse performance toward the left of the graphs; and that the models trained on the 'Low" SNR range (blue) and models trained on the 'High' SNR range (orange) have been exposed to the hardest/easiest training regime respectively.

From the plots we can see that: (i) Overall MetaCurvature is best, followed by MAML, Reptile and Vanilla. (ii) The meta-learners' performance is clearly better when operating within-domain than when operating with train-test distribution shift, indicated by the crossing of the lines corresponding to the two training conditions. (iii) However, considering the stronger MetaCurvature, even when it is evaluated in task-shift condition (orange line in low-SNR regime on the left; blue line in high-SNR regime on the right), it is better than the non-meta learned Vanilla decoder.

Across-Family Setup We next consider the across-family setting. In this case we create task distributions defined by each channel type and use them for meta-training. We then consider several channel types for meta-testing and evaluate pairs of matched and mis-matched train/test regimes.

Results From the results in Fig. 5, we can see that: (i) Meta-learning is generally most successful in the within-family. IE: When source channel on the x-axis matches the target channel of the sub-plot, as indicated by the dashed box. (ii) Some across-channel family conditions also perform quite well, such as Multipath $\rightarrow$ Bursty; but not others, such as Bursty $\rightarrow$ AWGN. (iii) However, some specific channel families such as Bursty cannot be succesfully addressed when transferring from any other cross-family training distribution.

Summary We have seen in the previous two experiments that meta-learning performance is best when $p_{te} = p_{tr}$, with performance degrading smoothly when there is small deviation between them (Fig. 4), and sometimes dropping significantly when they are entirely different task families (Fig. 5). A key feature of channel coding as a meta-learning benchmark is the ability to measure the distance between task distributions in a systematic manner, as explained in Section 2.3. We can thus aggregate our results across experiments and plot normalized BER against meta-train meta-test task distribution distance as shown in Fig. 6. Here, each dot on the scatter plot is an experiment, and we fit lines for each model to show how different meta-learners respond to increasingly different train-test task distributions. Going forward, the evaluation shown in Fig. 6 can provide a metric to benchmark the robustness of meta-learners to train-test distribution shift, and can be used as an overall robustness metric to complement the standard approach of evaluating within-task-distribution performance.

4 Discussion

We set out to study the significance of training task-distribution complexity and train-test task distribution shift on meta-learning. Overall we saw only a mild degradation in performance under complex task distributions compared to simple task distributions (Figure 2). In contrast to prior work (Vuorio et al. 2019; Yu et al. 2019), this suggests that meta-underfitting, or lack of meta-learner capacity is not a major bottleneck in this application. This may also be due to our setting requiring fast adaptation to new domains rather than new tasks. However, our analysis does show that performance degrades as train-test distribution shift increases, both within and across task-family (Figures 4-5). Since these distributions can almost certainly never be perfectly aligned in real applications (Yu et al. 2019; Triantafillou et al. 2020), it is important to develop meta-learners that are robust to such distribution shift. Our ability to measure performance under

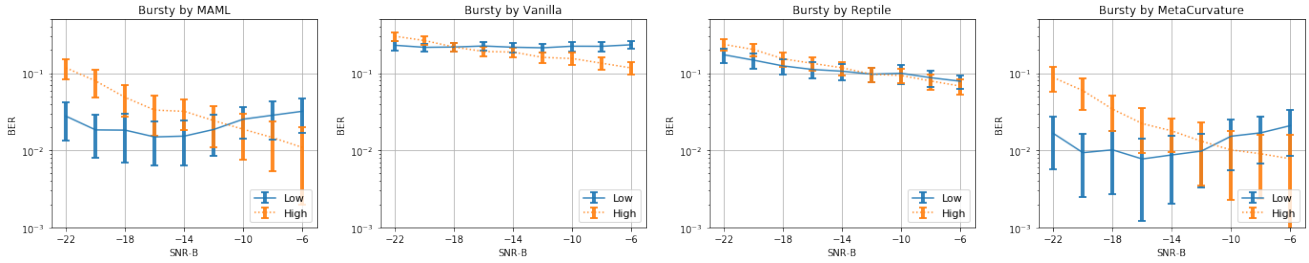

Figure 4: Impact of train-test distribution shift on decoding performance, within-family condition. X-axis: Meta-testing distribution parameter. Lines: Meta-training regime. "Low" training regime corresponds to lower SNR sampled from range (-2.5,3.5) and SNR-B from (-23, -17) for Bursty channel, and "High" training regime corresponds to SNR and SNR-B sampled from (8.5,13.5) and (-11, -5), respectively. Meta-learners (MAML, MetaCurvature) are better than a non-adaptive learned decoder (vanilla). Their performance degrades relatively smoothly as decoders are evaluated in increasingly out-of-domain conditions.

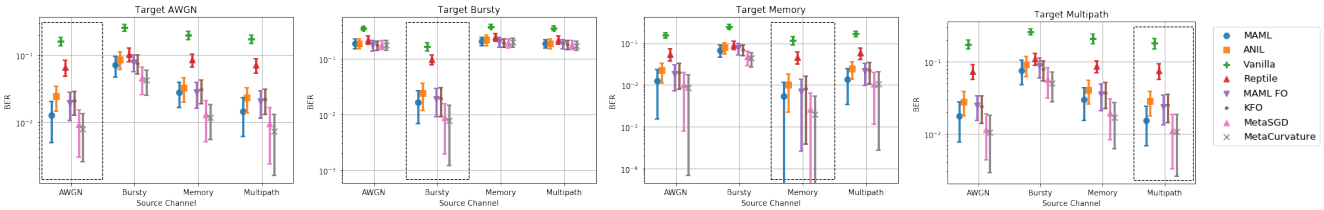

Figure 5: Impact of train-test distribution shift on decoding performance, across-family condition. Each sub-plot shows results of meta-testing on one of the AWGN, Bursty, Memory, Multipath task families, after learning on different meta-training channels (lines). Lines correspond to meta-test standard deviations of each algorithm. Boxes indicate when meta-training and meta-testing task families align, i.e., the within-family condition. Overall meta-learning works reliably within-task distribution (boxes), and occasionally across task distribution.

different degrees of distribution shift provides an excellent metric to drive this research in future.

A second contribution of this work is the introduction of quantifiable metrics for the diversity of a task distribution, and the distance between training and testing task distributions. Our synthesis of results showed that task distributions with greater diversity can benefit more from meta-learning (Figure 3), and that while absolute performance drops with train-test task distribution distance, the benefit provided by meta-learning over the non-meta baseline can actually increase (Figure 6).

Future Work In future, we will investigate the application of these task distribution metrics to standard meta-learning benchmarks such as image recognition. We have left for future work evaluating the impact of the *number* of unique meta-training tasks on meta-learning performance, but this is straightforward to explore within our framework. In the current evaluation we only considered gradient-based meta-learners, however feed-forward meta-learners could also be applied to channel coding and evaluated in the same framework. In our evaluation, we considered a single instance of adaptation to new tasks, however in real applications with moving users the channel becomes a continuously varying environment. Thus coding could also serve as a systematic benchmark for continuous meta-learning (Al-Shedivat et al. 2018), which has thus far suffered from lack of a simple and standardized benchmark. Meta-learning specifically in support of direct cross-domain robustness is also a topical branch of meta-learning (Balaji, Sankaranarayanan, and Chellappa 2018), and such methods can be that can be mapped directly onto our benchmark for evaluation as improvements to the "Vanilla" baseline in the case where no adaptation is allowed on target channels.

5 Conclusion

We presented a new meta-learning benchmark based on channel coding, a real-world and practically important problem that lends itself to meta-learning. We used it to evaluate the ability of state of the art meta-learners to learn complex task distributions without underfitting, and their ability to generalize across meta-train/test task distribution shift both within and across task families. Overall, we believe this benchmark setup will prove fruitful to help the community study these issues going forward.

References

Al-Shedivat, M.; Bansal, T.; Burda, Y.; Sutskever, I.; Mordatch, I.; and Abbeel, P. 2018. Continuous Adaptation via Meta-Learning in Nonstationary and Competitive Environments. *ICLR* .

Arnold, S. M.; Iqbal, S.; and Shal, F. 2019. When MAML Can Adapt Fast and How to Assist When It Cannot. *arXiv preprint arXiv:1910.13603* .

Arnold, S. M. R.; Mahajan, P.; Datta, D.; Bunner, I.; and Zarkias, K. S. 2020. learn2learn: A Library for Meta-Learning Research URL http://arxiv.org/abs/2008.12284.

Balaji, Y.; Sankaranarayanan, S.; and Chellappa, R. 2018.

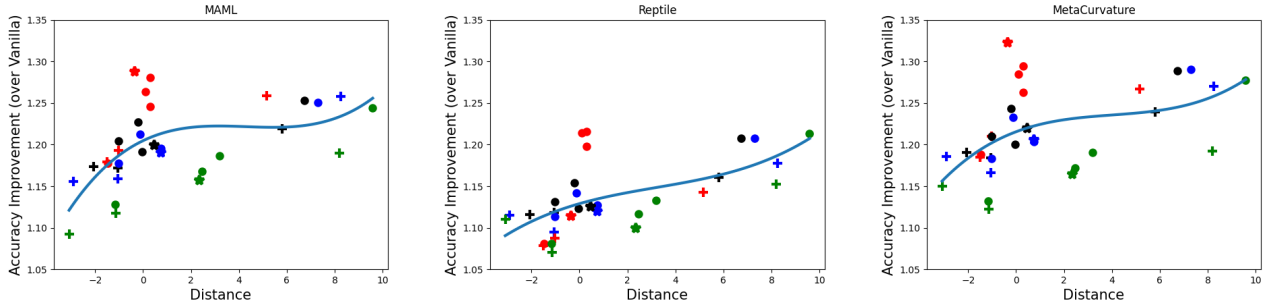

Figure 6: Impact of train-test task distribution distance on accuracy for (left) MAML, (middle) Reptile, and (right) MetaCurvature. X-axis denotes the KLD distance score between train and test distributions, and Y-axis denotes accuracy gain over vanilla. Each dot on the scatter plot corresponds to an experiment in Figure 2 ($*$, shift from mixed channels), Figure 4 (+, shift within family) and Figure 5 (o, shift across-family).

MetaReg: Towards Domain Generalization using Meta-Regularization. In *NIPS*.

Baz, A. E.; Guyon, I.; Liu, Z.; van Rijn, J. N.; Treguer, S.; and Vanschoren, J. 2021. AAAI Workshop on Meta-Learning. URL https://metalearning.chalearn.org.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-learning for Fast Adaptation of Deep Networks. In *ICML*.

Flennerhag, S.; Rusu, A. A.; Pascanu, R.; Visin, F.; Yin, H.; and Hadsell, R. 2020. Meta-Learning with Warped Gradient Descent. In *ICLR*.

Hospedales, T.; Antoniou, A.; Micaelli, P.; and Storkey, A. 2020. Meta-Learning in Neural Networks: A Survey.

Jiang, Y.; Kim, H.; Asnani, H.; and Kannan, S. 2019. Mind: Model independent neural decoder. In *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 1–5. IEEE.

Kim, H.; Jiang, Y.; Kannan, S.; Oh, S.; and Viswanath, P. 2018a. Deepcode: Feedback Codes via Deep Learning. *NIPS* .

Kim, H.; Jiang, Y.; Rana, R.; Kannan, S.; Oh, S.; and Viswanath, P. 2018b. Communication algorithms via deep learning. *ICLR* .

Kraskov, A.; Stögbauer, H.; and Grassberger, P. 2004. Estimating mutual information. *Phys. Rev. E* 69: 066138.

Kullback, S.; and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics* 22(1): 79–86.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017a. Deeper, Broader and Artier Domain Generalization. In *ICCV*.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017b. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835* .

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On First-Order Meta-Learning Algorithms. In *arXiv*.

O'Shea, T.; and Hoydis, J. 2017. An Introduction to Deep Learning for the Physical Layer. *IEEE Transactions on Cognitive Communications and Networking* 3(4): 563–575. ISSN 2372-2045. doi:10.1109/TCCN.2017.2758370.

Park, E.; and Oliva, J. B. 2019. Meta-curvature. In *Advances in Neural Information Processing Systems*, 3314–3324.

Raghu, A.; Raghu, M.; Bengio, S.; and Vinyals, O. 2020. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *arXiv*.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2019. Meta-Learning with Latent Embedding Optimization. *ICLR* .

Schmidhuber, J.; Zhao, J.; and Wiering, M. 1996. Simple principles of metalearning. *Technical report IDSIA* 69: 1–23.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical Networks for Few-shot Learning. In *NIPS*.

Steeg, G. V. 2014. Non-parametric Entropy Estimation Toolbox (NPEET) URL http://github.com/gregversteeg/NPEET.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *CVPR*.

Thrun, S.; and Pratt, L., eds. 1998. *Learning to Learn*. Kluwer Academic Publishers.

Triantafillou, E.; Zhu, T.; Dumoulin, V.; Lamblin, P.; Evci, U.; Xu, K.; Goroshin, R.; Gelada, C.; Swersky, K.; Manzagol, P.-A.; and Larochelle, H. 2020. Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. In *ICLR*.

Vuorio, R.; Sun, S.-H.; Hu, H.; and Lim, J. J. 2019. Multimodal Model-Agnostic Meta-Learning via Task-Aware Modulation. In *NeurIPS*.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; and Levine, S. 2019. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CORL*.

Zintgraf, L.; Shiarli, K.; Kurin, V.; Hofmann, K.; and Whiteson, S. 2019. Fast Context Adaptation via Meta-Learning. In *ICML*.