A APPENDIX

A.1 SUMMARY OF ACRONYMS

Acronyms of datasets and descriptions can be found below in section A.3

Table 6: List of acronyms used in this paper.

Acronym	Description
ARLM	Autoregressive Language Models
CA-MTL	Conditional Adaptive Multi-Task Learning: our architecture
CFF	Conditional Feed-Forward: a feed forward layer modulated by a conditioning vector
CLN	Conditional Layer Normalization in section 2.3
EDM	Evolutionary Data Measures (Collins et al., 2018): a task difficulty estimate
GLUE	General Language Understanding Evaluation Wang et al. (2018): a benchmark with multiple datasets
QA	Question Answering
MT	Multi-Task
MTAL	Multi-Task Active Learning: finding the most informative instance for multiple learners (or models)
MLM	Masked Language Model: BERT Devlin et al. (2018) is an example of an MLM
MTL	Multi-Task Learning: "learning tasks in parallel while using a shared representation" (Caruana, 1997)
MRQA	Machine Reading for Question Answering Fisch et al. (2019): a benchmark with multiple datasets
NLP	Natural Language Processing
SOTA	State of the art
ST	Single Task finetuning: all weights are typically updated
ST-A	ST with Adapter modules: one adapter per task is trained and pretrained weights are optionally updated

A.2 UNCERTAINTY SAMPLING: ALGORITHM AND ADDITIONAL RESULTS

Algorithm 1: Multi-task Uncertainty Sampling

Input: Training data D_t for task $t \in [1, \ldots, T]$; batch size b; C_t possible output classes for task t; $f := f_{\phi(\mathbf{z}_i),\theta_i}$ our model with weights ϕ, θ_i;

Output: $\mathcal{B}'$ - multi-task batch of size b

$\mathcal{B} \leftarrow \emptyset$

for $t \leftarrow 1$ **to** T **do**

  Generate $\mathbf{x}_t := \{x_{t,1}, \ldots, x_{t,b}\} \overset{\text{i.i.d.}}{\sim} D_t$

  for $i \leftarrow 1$ **to** b **do**

    $\mathcal{H}_{t,i} \leftarrow -\sum_{c=1}^{C_i} p_c(f(x_{t,i}))\ \log\ p_c(f(x_{t,i}))$ ▷ Entropy of each sample

  end

  Compute $\bar{\mathcal{H}}_t \leftarrow \frac{1}{b}\sum_{\mathbf{x}\in\mathbf{x}_i} \mathcal{H}_{t,i}$ ▷ Average entropy for task t

  Compute $H'_t \leftarrow -\sum_{c=1}^{C_t} \frac{1}{C_t}\ \log\left[\frac{1}{C_t}\right]$ ▷ Max entropy (uniform distribution)

  $\mathcal{B} \leftarrow \mathcal{B} \cup \mathbf{x}_t$ and $D_t \leftarrow D_t \setminus \mathbf{x}_t$

  if $D_t = \emptyset$ **then**

    Reload D_t

  end

  for $i \leftarrow 1$ **to** b **do**

    Compute: $\mathcal{U}_{t,i} \leftarrow \mathcal{H}_{t,i}/H'_t$ ▷ Uncertainty normalized with max entropy

  end

end

Compute $\hat{\mathcal{H}} \leftarrow \max_{i\in\{1,\ldots,T\}}[\bar{\mathcal{H}}_t]$ ▷ Entropy of task with highest average entropy

Update $\mathcal{U}_{t,i} \leftarrow \mathcal{U}_{t,i}/\hat{\mathcal{H}}$ ▷ Normalize each sample's uncertainty measure

$\mathcal{B}' \leftarrow \text{top\_b}(\{\mathcal{U}_{t,i} | t \in [1, \ldots, T], i \in [1, \ldots, b]\})$ ▷ b samples w/ highest uncertainty

Return: With $\mathcal{B}'$, solve eq. 1 with gradient descent; updated model f

An advantage of our MT-Uncertainty Sampling approach is its ability to manage task difficulty. This is highlighted in Figure 6. In this experiment, we estimated task difficulty using the Evolutionary Data Measures (EDM)[4] proposed by Collins et al. (2018). The task difficulty estimate relies on multiple dataset statistics such as the data size, class diversity, class balance and class interference. Interestingly, estimated task difficulty correlates with the first instance that the selection of a specific task occurs. Supposing that QNLI is an outlier, we notice that peaks in the data occur whenever tasks are first selected by MT Uncertainty sampling. This process follows the following order: 1. MNLI 2. CoLA 3. RTE 4. QQP 5. MRPC 6.SST-2, which is the order from highest task difficulty to lowest task difficulty using EDM. As opposed to Curriculum Learning (Bengio et al., 2009), MT-Uncertainty dynamically prioritizes the most difficult tasks. As also discovered in MTL vision work (Guo et al., 2018), this type of prioritization on more difficult tasks may explain MT-Uncertainty's improved performance over other task selection methods.

While the EDM difficulty measure, is shown to correlate well with model performance, it lacks precision. As reported in Collins et al. (2018), the average score achieved on the Yahoo Answers dataset is 69.9% and its difficulty is 4.51. The average score achieved on Yelp Full is 56.8%, 13.1% less than Yahoo Answers and its difficulty is 4.42. The authors mention that "This indicates that the difficulty measure in its current incarnation may be more effective at assigning a class of difficulty to datasets, rather than a regression-like value".

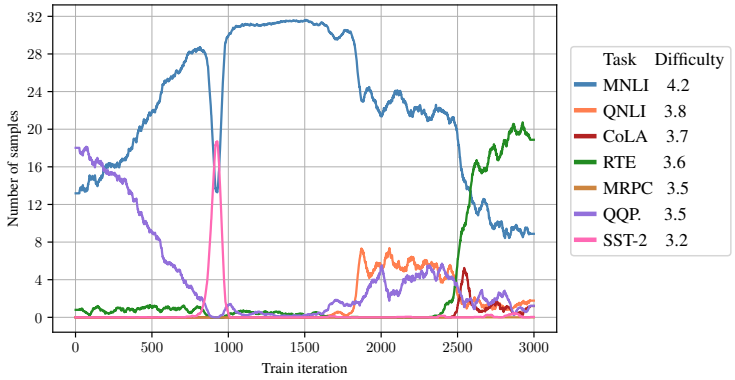


Figure 6: Task composition of MT-Uncertainty sampling and estimated task difficulty using EDM: number of training samples per task at each iteration for batch size of 32. The occurrence of first peaks and estimated difficulty follow the same order: From highest to lowest: MNLI > CoLA > RTE > QQP = MRPC > SST-2.

A.3 Dataset Description

The datasets that were used for the domain adaption experiments were SciTail[5] and SNLI[6]. We *jointly* trained a CA-MTL$_{\text{RoBERTa-LARGE}}$ model on 9 GLUE tasks, 8 Super-GLUE[7] tasks, 6 MRQA[8] tasks, and on WNUT2017[9] (Derczynski et al., 2017).

All GLUE tasks are binary classification, except STS-B (regression) and MNLI (three classes). We used the same GLUE data preprocessing as in Devlin et al. (2018).

[4]https://github.com/Wluper/edm
[5]https://allenai.org/data/scitail; Leaderboard can be found at: https://leaderboard.allenai.org/scitail/submissions/public
[6]https://nlp.stanford.edu/projects/snli/
[7]https://super.gluebenchmark.com/tasks
[8]https://github.com/mrqa/MRQA-Shared-Task-2019
[9]https://github.com/leondz/emerging_entities_17

Table 7: GLUE (Wang et al., 2018) dataset description.
References: [1]Warstadt et al. (2018), [2]Socher et al. (2013), [3]Dolan & Brockett (2005), [4]Cer et al. (2017), [5]Williams et al. (2018), [6]Wang et al. (2018), [7]Levesque (2011)

Acronym	Corpus	|Train|	Task	Domain
CoLA[1]	Corpus of Linguistic Acceptability	8.5K	acceptability	miscellaneous
SST-2[2]	Stanford Sentiment Treebank	67K	sentiment detection	movie reviews
MRPC[3]	Microsoft Research Paraphrase Corpus	3.7K	paraphrase detection	news
STS-B[4]	Semantic Textual Similarity Benchmark	7K	textual similarity	miscellaneous
QQP	Quora Question Pairs	364K	paraphrase detection	online QA
MNLI[5]	Multi-Genre NLI	393K	inference	miscellaneous
RTE[6]	Recognition Textual Entailment	2.5K	inference/entailment	news, Wikipedia
WNLI[7]	Winograd NLI	634	coreference	fiction books

Table 8: Super-GLUE (Wang et al., 2019b) dataset description. References: [1]Clark et al. (2019a), [2]de Marneffe et al. (2019), [3]Gordon et al. (2012), [4]Khashabi et al. (2018), [5]Zhang et al. (2018), [6]Wang et al. (2019b), [7]Poliak et al. (2018), [8]Levesque (2011)

Acronym	Corpus	|Train|	Task	Domain
BoolQ[1]	Boolean Questions	9.4K	acceptability	Google queries, Wikipedia
CB[2]	CommitmentBank	250	sentiment detection	miscellaneous
COPA[3]	Choice of Plausible Alternatives	400	paraphrase detection	blogs, encyclopedia
MultiRC[4]	Multi-Sentence Reading Comprehension	5.1K	textual similarity	miscellaneous
ReCoRD[5]	Reading Comprehension and Commonsense Reasoning	101K	paraphrase detection	news
RTE[6]	Recognition Textual Entailment	2.5K	inference	news, Wikipedia
WiC[7]	Word-in-Context	6K	word sense disambiguation	WordNet, VerbNet
WSC[8]	Winograd Schema Challenge	554	coreference resolution	fiction books

Table 9: MRQA (Fisch et al., 2019) dataset description. References: [1]Rajpurkar et al. (2016a), [2]Trischler et al. (2017), [3]Joshi et al. (2017), [4]Dunn et al. (2017), [5]Yang et al. (2018), [6]Kwiatkowski et al. (2019)

Acronym	Corpus	|Train|	Task	Domain
SQuAD[1]	Stanford QA Dataset	86.6K	crowdsourced questions	Wikipedia
NewsQA[2]	NewsQA	74.2K	crowdsourced questions	news
TriviaQA[3]	TriviaQA	61.7K	trivia QA	web snipets
SearchQA[4]	SearchQA	117.4K	Jeopardy QA	web snipets
HotpotQA[5]	HotpotQA	72.9K	crowdsourced questions	Wikipedia
Natural Questions[6]	Natural Questions	104.7K	search logs	Wikipedia

SuperGLUE has a more diverse task format than GLUE, which is mostly limited to sentence and sentence-pair classification. We follow the same preprocessing procedure as in Wang et al. (2019b). All tasks are binary classification tasks, except CB (three classes). Also, WiC and WSC are span based classification tasks. We used the same modified MRQA dataset and preprocessing steps that were used in Joshi et al. (2019). All MRQA tasks are span prediction tasks which seeks to identify start and end tokens of an answer span in the input text.

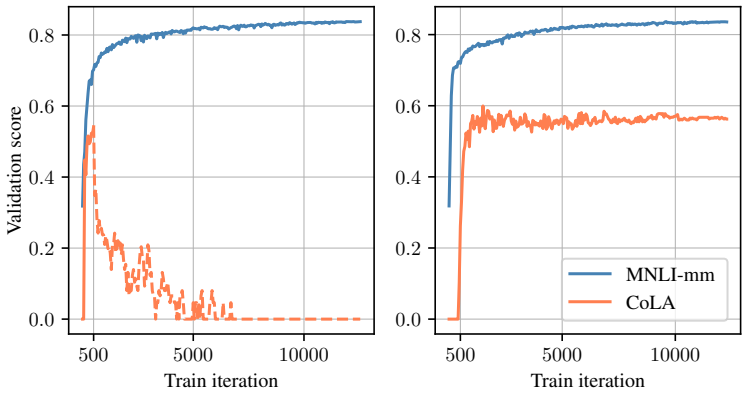
Table 10: SNLI (Bowman et al., 2015) and SciTail (Khot et al., 2018) datasets description.

Acronym	Corpus	|Train|	Task	Domain
SNLI[1]	Stanford Natural Language Inference	550.2k	inference	human-written English sentence pairs
SciTail[2]	Science and Entailment	23.5K	entailment	Science question answering

SNLI is a natural inference task where we predict three classes. Examples of three target labels are: Entailment, Contradiction, and Neutral (irrelevant). SciTail is a textual entailment dataset. The hypotheses in SciTail are created from multiple-choice science exams and the answers candidates (premise) are extracted from the web using information retrieval tools. SciTail is a binary true/false classification tasks that seeks to predict whether the premise entails the hypothesis. The two datasets are used only for domain adaptation in this study (see section A.5 for the details of our approach).

A.4 CATASTROPHIC FORGETTING

The datasets in the GLUE benchmark offers a wide range of dataset sizes. In MTL, heuristics to balance tasks during training is typically done by weighting each task's loss differently. We have investigated in preceding section MT-Uncertainty was able to prioritize task difficulty. Now, we see if MT-Uncertainty can help keep a low resource task performance steady and avoid catastrophic forgetting. Our experimental set-up is the same as in section 4.1. In Figure 6, we compare our method with Random sampling (see equation 6). With Random sampling, CoLA's dataset is seen completely by iteration 500 and the task performance starts to decrease. On the other hand, MT-Uncertainty samples the task whenever it's Shannon Entropy is high.



(a) Random (b) MT-Uncertainty

Figure 7: Illustrating catastrophic forgetting with two tasks in the first epoch: With a random sampling strategy, all of CoLA's tasks are sampled by iteration 500, at which point the larger MNLI dataset overtakes the learning process. With MT-Uncertainty sampling, CoLA is sampled whenever Shannon entropy is high but not necessarily at every iteration, allowing lower resource tasks to avoid catastrophic forgetting.

A.5 ZERO-SHOT RESULTS ON SCITAIL AND SNLI

Table 11: CA-MTL is flexible and extensible to new tasks. However, CA-MTL is sensitive to the new task's embedding. We tested multiple task embeddings that worked best on either SciTail or SNLI by checking performance in a zero shot setting or using 0% of the data.

Initialization of new task embedding layer	SciTail 0% of data	SNLI 0% of data
CoLA init	43.0	34.0
MNLI init	24.2	33.0
MRPC init	34.5	**45.5**
STS-B init	**46.9**	33.2
SST-2 init	25.8	34.2
QQP init	31.7	37.3
QNLI init	32.0	38.0
RTE init	32.3	40.6
WNLI init	29.0	30.4
Average init	28.7	37.7
Random init	46.8	34.0
Xavier init	29.8	37.6

Before testing models on domain adaptation in section 4.2, we ran zero-shot evaluations on the development set of SciTail and SNLI. Table 11 outlines CA-MTL$_{\text{BERT-BASE}}$'s zero-shot transfer abilities when pretrained on GLUE with our MTL approach. We expand the task embedding layer to accommodate an extra task and explore various embedding initialization. We found that reusing STS-B and MRPC task embeddings worked best for SciTail and SNLI respectively.

A.6 NAMED ENTITY RECOGNITION (NER) RESULTS

We report NER task results on the WNUT2017 dataset in table 12. As with the other 23 tasks (see section 4.2) that we *jointly* trained our 24-task CA-MTL$_{\text{RoBERTa-LARGE}}$ model, we did not use fine-tuning or assemble methods on WNUT2017. We compare with the latest state-of-the art models. Note that Nguyen et al. (2020) used RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019c) and XLM-R$_{\text{LARGE}}$ (Conneau et al., 2020) as large model baselines. CA-MTL$_{\text{RoBERTa-LARGE}}$ outperforms XLM-R$_{\text{LARGE}}$ by 1.6% WNUT2017 F1 score. Except for the BLSTM-CRF-MTL Aguilar et al. (2019) model and our method, all methods use single task fine-tuning.

Table 12: WNUT2017 test F1 results (entity level) on the NER task. Results taken from: [1]Aguilar et al. (2019), [2]Zhou et al. (2019), [3]Nguyen et al. (2020)

SOTA Models	F1
BLSTM-CRF-MTL[1]	41.9
DATNet[2]	42.3
BERTweet[3]	56.5
RoBERTa$_{\text{LARGE}}$[3]	56.9
XLM-R$_{\text{LARGE}}$[3]	57.1
CA-MTL$_{\text{RoBERTa-LARGE}}$ (ours)	**58.0**

A.7 MORE EXPERIMENTAL DETAILS

For Figure 5 and Table 5, all BERT-based model have half their layers frozen (untrained) for a fair comparison of ablation results.