UiTTa: Online Test-Time Adaptation by User Interaction

Anonymous authors
Paper under double-blind review

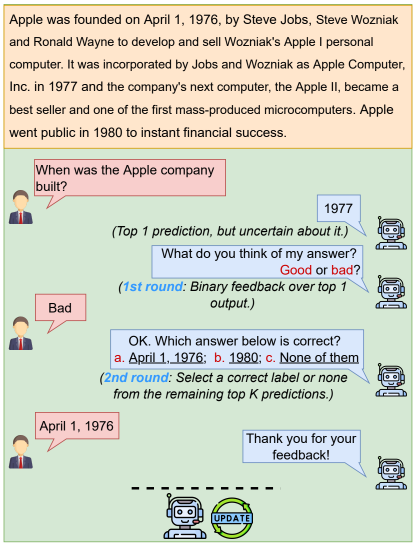
Abstract

We explore user interaction-based test-time adaptation (UiTTa), which adapts a model to shifted test distributions with supervision signals from model-user interactions. Model adaptation in TTA can fail since models learn from the noisy pseudo-labels of the test data. UiTTa achieves better adaptation from user feedback on top-K predictions within two rounds of simulated interactions. To have real-time adaptation, we further accelerate model optimization by reducing the cost of gradient backpropagation, through random dropping of backward paths. Simulation experiments on cross-lingual transfer, domain generalization, and corruption robustness show that low-cost user feedback can significantly boost TTA in performance, even competing with online active learning which however needs expensive human annotation. By accelerating pre-trained language models, we reduce 70% – 90% backpropagation cost with only a small drop in performance.

1 Introduction

Real-world machine learning systems suffer from mismatched distributions during training time and test time (Belinkov & Bisk, 2018; Gan & Ng, 2019; Wang et al., 2021c; Rychalska et al., 2019; Hendrycks & Dietterich, 2019; Tu et al., 2020). In the last few years, test-time adaptation (TTA) that adapts to arbitrary test distributions has been a consistent theme of machine learning research (Wang et al., 2021a). The dominant paradigm to achieve test-time adaptation is based on self-training (Lee et al., 2013). Insofar as self-training is an effective method for generating pseudo-labels (Xie et al., 2020; Sohn et al., 2020; Pham et al., 2021), a mismatched test-time distribution renders the pseudo-labels less applicable, since it may inject too much noise as illustrated in Fig. 2. In addition, a changing test distribution over a long time horizon makes the situation worse (Wang et al., 2022).

That the training set and test set are drawn i.i.d from the same distribution is an in-built assumption in machine learning. It points us to another alternative approach, active learning (Ein-Dor et al., 2020; Settles, 2009), to reason about the changing distribution. Active learning adopts an active learner to query an oracle (user or expert) to label the unlabeled test data to reduce noises. However, it is prohibitively expensive to run an on-the-fly data labeling process.


Apple was founded on April 1, 1976, by Steve Jobs, Steve Wozniak and Ronald Wayne to develop and sell Wozniak's Apple I personal computer. It was incorporated by Jobs and Wozniak as Apple Computer, Inc. in 1977 and the company's next computer, the Apple II, became a best seller and one of the first mass-produced microcomputers. Apple went public in 1980 to instant financial success.

When was the Apple company built?

1977

(Top 1 prediction, but uncertain about it.)

What do you think of my answer? Good or bad?

(1st round: Binary feedback over top 1 output.)

Bad

OK. Which answer below is correct?
a. April 1, 1976; b. 1980; c. None of them

(2nd round: Select a correct label or none from the remaining top K predictions.)

April 1, 1976

Thank you for your feedback!

Figure 1: Illustration of a two-round interaction between the user and a QA system.

Motivated by the aforementioned difficulties, we propose user interaction-based test time adaptation (UiTTa) that involves users in the adaptation loop. UiTTa aims to identify and possibly correct the noises by model-user interactions and explores an efficient way to reduce the user cost (as in Fig. 1). UiTTa requires the model to learn to query data whose predictions it is uncertain about. It explores a low-cost labeling paradigm on data instances, where user feedback is sought on the top K (e.g., 2) model predictions. Such a labeling approach has potential since the gold label is likely

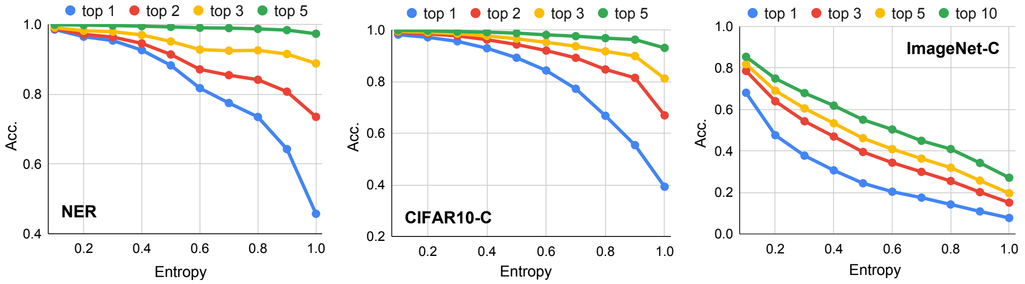


Figure 2: Model accuracy measured from top K predictions. For data with high entropy, low accuracy is observed in top one predictions, but accuracy improves significantly in top 3, 5 and 10 predictions.

to be found in the top few model predictions as supported by Fig. 2. The cost of user annotation can be further reduced through a well-designed user interface (Gao et al., 2022; Kratzwald et al., 2020) to fit the scenario of on-the-fly data labeling. As evidenced by Shuster et al. (2022), user feedback (e.g., binary feedback or free-form text) can be adequately collected in practice to improve an open-domain dialogue system.

The user feedback is incorporated as supervision signals to update the model, together with the confident unsupervised signals from the model. However, model optimization could be costly in TTA, especially when there a large number of parameters in the model, e.g., a pre-trained language model (PLM) (Devlin et al., 2019). Since lower layers in deep neural networks learn general knowledge, instead of task-specific knowledge learnt by the upper layers (Merchant et al., 2020; Yosinski et al., 2014), we accelerate model updating by reducing the cost of gradient backpropagation, where the backpropagation is randomly stopped when it reaches a certain layer. We focus on speeding up the training of PLMs built on transformer layers (Vaswani et al., 2017) in this work.

We simulate the model-user interaction of UiTTA on cross-lingual transfer (Hu et al., 2020), domain generalization in machine reading comprehension (Fisch et al., 2019), and robustness to image corruptions (Hendrycks & Dietterich, 2019). We empirically look into how the choice of K affects the adaptation performance, the generalization of user feedback to changes in test distributions, the efficiency of involving humans in the loop, and the effects of perturbation in user feedback.

Contributions

- We study an under-explored setting which is test-time adaptation from model-user interaction.
- UiTTA emphasizes the efficiency of user involvement and explores how to reduce the cost of involving users during interactions.
- We further propose a simple but effective method to reduce the cost of model optimization in pre-trained language models.
- Extensive simulation experiments are conducted. We find that UiTTA significantly boosts the performance over TTA baselines with only 30% to 40% user involvement. Top-K feedback is even comparable to active learning which however needs costly human annotation.
- By saving up to 70% to 90% of backpropagation cost, our acceleration method achieves results comparable to those without speed-up.

2 Related Work

Test-Time Adaptation TTA adapts a source model from a training distribution to a test distribution, happening during test time. Extensive methods have been proposed to learn on test data for adaptation, including entropy minimization (Wang et al., 2021a), test-time classifier adjustment (Iwasawa & Matsuo, 2021), and batch norm estimation (Nado et al., 2020), which needs no modification to the training-time loss compared to TTT (Sun et al., 2020), TTT+ (Liu et al., 2021b), and MT3 (Bartler et al., 2022). Some other works consider more issues in TTA. For example, Niu et al. (2022) study how to prevent the model from forgetting source knowledge during model adaptation, Wang et al. (2022) consider a more challenging setting where the test distributions change over time, and Khurana et al. (2021) study how to update the model with only one test sample on the fly.

	Data Usage		Optimization Loss		Feedback
Setting	# Train	# Test	# Train	# Test	# Test
Unsupervised Domain Adaptation	$x^{\mathcal{S}}, y^{\mathcal{S}}$	$x^{\mathcal{T}}$	$\mathcal{L}(x^{\mathcal{S}}, y^{\mathcal{S}}) + \mathcal{L}(x^{\mathcal{T}})$	-	None
Test-time Adaptation (TTA)[†]	-	$x^{\mathcal{T}}$	-	$\mathcal{L}(x^{\mathcal{T}})$	None
Online Active Learning (OAL)[♭]	-	$x^{\mathcal{T}}$	-	$\mathcal{L}(x^{\mathcal{T}}) + \mathcal{L}(x^{\mathcal{T}}, y^{\mathcal{T}})$	$(x^{\mathcal{T}}, y^{\mathcal{T}})$
User Interaction-based TTA (UiTTA)	-	$x^{\mathcal{T}}$	-	$\mathcal{L}(x^{\mathcal{T}}) + r \cdot \mathcal{L}(x^{\mathcal{T}}, y'^{\mathcal{T}})$	$\langle r, (x^{\mathcal{T}}, y'^{\mathcal{T}}) \rangle$

Table 1: Compared settings. $y^{\mathcal{T}}$ is a gold label and $y'^{\mathcal{T}}$ is a noisy label. r is a binary value: 1 for correct label, and 0 for wrong label. UiTTA and OAL both request users to annotate a model's uncertain data, but UiTTA only needs feedback on top K outputs instead of full annotation. [†]Wang et al. (2021a). [♭] Settles (2009).

Learning with Human Feedback There are many tasks that have explored human-in-the-loop processing to reduce the cost of data annotation and further improve the model. User feedback has been explored in semantic parsing (Lawrence & Riezler, 2018; Elgohary et al., 2021; Yao et al., 2020; Elgohary et al., 2020), machine translation (Mendonça et al., 2021; Kreutzer & Riezler, 2019; Nguyen et al., 2017), document summarization (Stiennon et al., 2020; Gao et al., 2018), and question answering (Kratzwald et al., 2020; Gao et al., 2022), where most of them focus on how to train a model from scratch after deployment. BlenderBot-3 (Shuster et al., 2022) is a dialogue system that interacts with users to improve itself continually, where it exploits various feedback types such as binary feedback, feedback in human language, etc. Our work focuses on a model robustness to distribution shift, by adapting through a novel two-round interaction.

Robustness in NLP Building robust NLP models attracts more attention in recent years (Wang et al., 2021c). Ribeiro et al. (2020) propose to measure model robustness in NLP. Some work (Goel et al., 2021; Wang et al., 2021b) provides platforms to evaluate NLP models. Some work aims to determine a model's weakness on specific NLP tasks such as dialogue understanding (Liu et al., 2021a), machine translation (Belinkov & Bisk, 2018), question answering (Gan & Ng, 2019), etc. Related benchmarks study cross-lingual transfer (Hu et al., 2020), domain generalization or adaptation (Fisch et al., 2019), and robustness to language corruptions (Ravichander et al., 2021).

3 Problem Definition

Online TTA Test-time adaptation (TTA) (Wang et al., 2021a) adapts a source model f_{θ_0} trained from a training distribution $\mathcal{S}$ to a test distribution $\mathcal{T}$, by learning from the unlabeled test data. We focus on online TTA in this work, which means that at each time t, the model f_{θ_t} first returns its prediction on the input x_t and then updates itself with x_t on the fly. The updated model parameters θ'_t will be carried over to the next time instance $t+1$: $\theta_{t+1} \leftarrow \theta'_t$.

UiTTA vs. OAL Different from previous settings (as summarized in Table 1), we study user interaction-based test-time adaptation (UiTTA) which tries to gather user feedback to achieve better adaptation. Online active learning (OAL) (Settles, 2009) selects a model's uncertain data for annotation (a gold label is assigned to a data instance) over a stream of data. OAL is similar to UiTTA, but OAL needs full human annotation instead of explicit user feedback, so OAL costs user much more than UiTTA. We consider OAL to be the upper bound of UiTTA.

We study two different adaptation scenarios: (1) **Constant Adaptation:** The test distribution stays unchanged all the time; (2) **Continual Adaptation:** As studied in Wang et al. (2022), the test distribution changes over time, which is harder than the first scenario.

4 Method

4.1 Online Test-time Adaptation by User Interaction

UiTTA aims to identify and possibly correct the noises caused by self-training-based methods. UiTTA tries to reduce the user cost during interaction through two mechanisms which are discussed in this section: learning to identify a model's uncertain data and incorporating binary user feedback on top K model predictions. Specifically, at time t, the input is x_t. The model generates a list of top

Algorithm 1 User Interaction-based Test-time Adaptation (UITTA)

```
Input: Source model $f_{\theta_0}$; $\alpha$ to rescale the threshold; $K$ top predictions; $T$ gradient steps.
Do warm-up to obtain $\bar{l}$;
for $x_t \in \{x_1, x_2, \cdots\}$ do  # each $x_t$ is a batch
    $\theta_t \leftarrow \theta'_{t-1}$;
    Generate top $K$ predictions $Y_t$ as in Eq. 1 and $y_t^{(0)} \in Y_t$;
    if $\mathcal{L}\big(f_{\theta_t}(x_t), y_t^{(0)}\big) < \alpha\bar{l}$ then
        ▷ Model is certain about its prediction.
        $l_t(\theta_t) \leftarrow \mathcal{L}\big(f_{\theta_t}(x_t), y_t^{(0)}\big)$;
    else
        ▷ Model is uncertain about the prediction and starts the interaction.
        $\langle r_t, (x_t, y_t)\rangle \leftarrow$ INTERACTION$(x_t, Y_t)$;
        $l_t(\theta_t) \leftarrow r_t \cdot \mathcal{L}\big(f_{\theta_t}(x_t), y_t\big)$;
    end if
    Obtain $\theta'_t$ by updating model with $l_t(\theta_t)$ for $T$ gradient steps;
end for
function INTERACTION$(x, Y)$  # Binary feedback repeats on each prediction from Y
    return $\langle 1, (x, y^{(0)})\rangle$ if user leaves good feedback to $y^{(0)} \in Y$;  ▷ 1st round interaction
    Show predictions $Y - \{y^{(0)}\}$ with $K-1$ outputs to the user;  ▷ 2nd round interaction
    return $\langle 1, (x, y^{(1)})\rangle$ if user selects $y^{(1)}$ from $Y - \{y^{(0)}\}$ else $\langle 0, (x, y^{(0)})\rangle$;
end function
```

K predictions of x_t (denoted as Y_t) as follows:

$$Y_t = \underset{Y'_t : |Y'_t| \le K}{\arg\max} \sum_{i \in Y'_t} f_{\theta_t}(x_t)[i] \tag{1}$$

where $y_t^{(0)} = \arg\max_i f_{\theta_t}(x_t)[i]$ is the top one prediction.

Learning to Query The model first decides whether or not it can trust the top one prediction $y_t^{(0)}$ based on cross-entropy loss, similar to least confidence selection used in active learning (Settles, 2009). Only data that are uncertain to the model can trigger user involvement. Mask m_t denotes whether the model is certain ($m_t = 1$) or uncertain ($m_t = 0$) about the prediction $y_t^{(0)}$:

$$m_t = 1 \;\; \textbf{if} \;\; \mathcal{L}\big(f_{\theta_t}(x_t), y_t^{(0)}\big) < \alpha\bar{l} \;\; \textbf{else} \;\; 0 \tag{2}$$

where $\mathcal{L}\big(f_{\theta_t}(x_t), y_t^{(0)}\big)$ is the cross-entropy loss, and only prediction with a loss smaller than the threshold is treated as certain. To determine the threshold, we perform warm-up at the start of model adaptation. That is, in the first several adaptation steps, all the data are presented to the user to request feedback. Then the cross-entropy losses of the data with *good* feedback are averaged to obtain $\bar{l}$, which approximately determines the loss of gold labels. α is a hyper-parameter to rescale $\bar{l}$ and it is set to 1 in most of the experiments. Only 1% or 5% of the test data is used to perform warm-up in our experiments.

User Feedback on Top K Predictions After identifying the model's uncertain data, the model starts its interaction with the user. We denote the returned user feedback after interaction as $\langle r_t, (x_t, y_t)\rangle$ where r_t is a binary value 1 or 0 and y_t is a label for the input x_t decided by the user. We simulate two rounds of interactions. **1st Round:** The user provides binary feedback to the predicted label $y_t^{(0)}$. If the user provides *good* feedback, then $\langle 1, (x_t, y_t^{(0)})\rangle$ is returned. **2nd Round:** If *bad* feedback is provided, then the model shows the remaining outputs which are $Y_t - \{y_t^{(0)}\}$ to the user, where the user has to select a satisfying prediction or none from the list. Suppose the user selects a satisfying label $y_t^{(1)}$ from the list, then $\langle 1, (x_t, y_t^{(1)})\rangle$ is returned; otherwise, $\langle 0, (x_t, y_t^{(0)})\rangle$ is returned.

Model Adaptation After interaction, the model incorporates the user feedback as supervision signal to update itself. On the model's certain data, the model updates like self-training (Lee et al., 2013). Specifically, the model at time t updates with the following loss:

$$l_t(\theta_t) = m_t \cdot \mathcal{L}\big(f_{\theta_t}(x_t), y_t^{(0)}\big) + (1 - m_t) \cdot r_t \cdot \mathcal{L}\big(f_{\theta_t}(x_t), y_t\big) \tag{3}$$

If the model is certain about the prediction ($m_t = 1$), the model updates with its predicted label; otherwise ($m_t = 0$), the model updates with the user feedback $\langle r_t, (x_t, y_t)\rangle$ from interaction. $l_t(\theta_t)$ can be optimized for T gradient steps to obtain an updated model θ'_t which will be carried over to the next time instance $t+1$: $\theta_{t+1} \leftarrow \theta'_t$. Detailed pseudocode is shown in Algorithm 1.

4.2 Fast UiTTa with Pre-trained Language Models

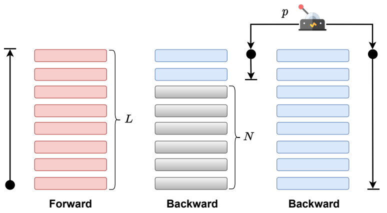


Figure 3: Random dropping of gradient backpropagation from one certain layer.

To fulfill real-time adaptation, we reduce the cost of model optimization during model adaptation. Based on the observation that lower layers in deep neural networks capture general information but upper layers learn task-specific information (Yosinski et al., 2014), we reduce the update frequency in lower layers by randomly freezing these layers in each updating step. In this way, the cost of gradient backpropagation could be substantially reduced.

We focus on optimizing pre-trained language models (PLMs) (Devlin et al., 2019), since PLMs have been widely used in NLP as foundation models (Bommasani et al., 2021) but with a large number of parameters. Specifically, suppose there are L layers in total, i.e., transformer layers (Vaswani et al., 2017) in PLMs. We pre-define a layer number N and a drop probability p. As shown in Fig. 3, during gradient backpropagation, when the gradient reaches layer N, backpropagation is randomly cut off (layers less than or equal to N are frozen) with drop probability p. Suppose the backward cost is 1, we can calculate the approximate expected saved *backward* cost q during model optimization as:

$$q(L, N, p) = p \cdot \frac{N}{L} \tag{4}$$

We only calculate the cost of gradient backpropagation in transformer layers other than layers such as the embedding layer and extra linear layers built for task-specific classification. Note that the embedding layer would also be frozen if backward gradients are cut off in transformer layers.

5 Experiments

We conduct simulation experiments to investigate UiTTa on constant and continual adaptation. Then model acceleration is evaluated, and noisy user feedback is then discussed.

5.1 Settings

Datasets We study cross-lingual transfer, domain generalization, and corruption robustness.

For cross-lingual transfer, we use the benchmark XTREME (Hu et al., 2020), where Wikiann (Pan et al., 2017), Universal Dependencies v2.5 (Nivre et al., 2018), XQuAD (Artetxe et al., 2020), and MLQA (Lewis et al., 2020) are evaluated. Models are trained from English corpus, and transferred to target languages such as Germany, Japanese, etc.

For domain generalization, we use the datasets from MRQA (Fisch et al., 2019), consisting of datasets in 6 domains for machine reading comprehension. We use SQuAD (Rajpurkar et al., 2016) as source, and the datasets in MRQA as targets, which are HotpotQA (Yang et al., 2018), NaturalQA (Kwiatkowski et al., 2019), NewsQA (Trischler et al., 2017), SearchQA (Dunn et al., 2017), and TriviaQA (Joshi et al., 2017).

For corruption robustness, CIFAR10-C, CIFAR100-C, and ImageNet-C (Hendrycks & Dietterich, 2019) are evaluated, for the task of image classification. For each dataset, there are 15 corruption types and the highest corruption severity level 5 is studied in our work.

The datasets are summarized in Table 2. The models are trained with source datasets, then evaluated (adapted) on each subset of each target set. On each target dataset, the average results over all subsets are reported, but the result of MRQA takes the average of the 6 target sets.

Baselines We compare the following baselines with our method UiTTa in different setups:

Distribution Shift	**Dataset**	**Task**	**Source**	**Target**	**|Subset|**	**|Test|**
Cross-lingual Transfer	XTREME	NER	en	Wikiann	25	10,000-13,165
		POS	en	Universal Dependencies	14	2,414-22,362
		MRC	SQuAD	XQuAD	11	1,190
		MRC	SQuAD	MLQA	7	4,517–11,590
Domain Generalization	MRQA	MRC	SQuAD	HotpotQA	1	5,901
				NaturalQA	1	12,836
				NewsQA	1	4,212
				SearchQA	1	16,980
				TriviaQA	1	7,785
Corruption Robustness	-	IC	CIFAR10	CIFAR10-C	15	10,000
			CIFAR100	CIFAR100-C	15	10,000
			ImageNet	ImageNet-C	15	5,000

Table 2: Datasets evaluated in this work. MRC: machine reading comprehension. IC: image classification. Models are first trained on source datasets, and then adapted on each subset of each target dataset. |Subset| is the number of subsets in the target set. |Test| shows the range of the subset size.

- **Source** The model is trained with the source distribution without adaptation at test time.
- **PL** The test-time adaptation baseline which predicts pseudo-labels on the test data and updates itself with such labels using cross-entropy loss (Lee et al., 2013).
- **Tent** TTA baseline which is similar to PL but optimizes entropy loss (Wang et al., 2021a).
- **UiTTa-K** Our method by generating top K predictions. We study UiTTa with different Ks. When K is equal to 1, then only the 1st round of interaction is conducted which requests the user to leave binary feedback to the top 1 prediction.
- **OAL** The baseline of online active learning which is the same as UiTTa, but requires full annotation from the user. The model obtains a gold label for the model uncertain data after interaction. OAL serves as an upper bound of UiTTa-K.

For UiTTa-K and OAL, the models need to identify model uncertain data to receive user feedback. Here, we compare them to the setting where all data are annotated by users during adaptation.

Evaluation For UiTTa and OAL, we evaluate the model performance with its top 1 prediction, not the one corrected by users from the top K predictions during interaction.

Model Setups For experiments of cross-lingual transfer and domain generalization, the source models are all based on a pre-trained language model which is XLMR-base (Conneau et al., 2020). For POS and NER tasks, user feedback is applied at token level. For QA tasks, the start and end positions are treated independently and each receives user feedback. For experiments on image corruption robustness, we take the model settings from Wang et al. (2022). For all the baselines, we try to maintain the batch size, learning rate, etc. that are independent from the algorithm to be the same. More training details are shown in Appendix 7.1.

5.2 Constant Adaptation

In constant adaptation, the test distribution stays the same during adaptation. Model adaptation is not accelerated here. The experimental results are shown in Table 3 and Fig 4.

TTA baselines cannot consistently improve the adaptation results. For image corruption, PL and Tent can consistently improve over source by a large margin, but such consistent improvements do not hold on NLP datasets, where PL and Tent drop a lot in POS and MRQA. PL is better than Tent in NLP tasks, in contrast to the CV datasets.

UiTTa consistently improves the adaptation performance, and larger K leads to better results. In POS, XQuAD, MLQA, and MRQA which are hard for PL and Tent to improve, UiTTa-3 brings around 8-point increase on average. For image corruption, UiTTa-5 improves over Tent by around 3 points. Comparing UiTTa with different Ks, we find larger K leads to better performance. There is an especially large performance improvement from $K = 1$ to $K = 2, 3, 5$, since $K = 1$ can only identify the noises which cannot be corrected without other top predictions (Gao et al., 2022).

UiTTa is efficient in exploiting user participation. UiTTa can achieve better performance with less user involvement by learning to identify the data on which it is not confident. As shown in

Metric	**NER** F1	**POS** F1	**XQuAD** EM / F1	**MLQA** EM / F1	**MRQA** EM / F1	Metric	**C10-C** Err.	**C100-C** Err.	**IN-C** Err.
Source	61	75.5	56.0 / 72.2	48.9 / 66.0	40.1 / 52.8	Source	43.5	46.4	82.4
PL	65.3	73.0	57.9 / 73.1	49.1 / 65.8	37.1 / 48.4	PL	19.8	32.2	66.9
Tent	64.6	66.7	56.9 / 72.6	48.0 / 65.0	35.0 / 46.1	Tent	18.6	31.0	64.6
Model Uncertainty									
UiTTa-1	70.6	80.5	60.4 / 75.0	51.1 / 67.6	38.5 / 51.0	UiTTa-1	16.1	30.2	64.5
|*HITL*|	*31*	*26*	*28*	*30*	*21*	|*HITL*|	*22*	*42*	*60*
UiTTa-2	72.4	81.9	64.9 / 78.2	53.4 / 69.8	44.1 / 56.4	UiTTa-3	15.4	29.7	63.1
|*HITL*|	*34*	*28*	*33*	*35*	*26*	|*HITL*|	*29*	*43*	*58*
UiTTa-3	73.2	82.9	65.8 / 79.2	54.1 / 70.6	49.8 / 62.2	UiTTa-5	15.1	29.5	62.6
|*HITL*|	*36*	*29*	*35*	*39*	*44*	|*HITL*|	*31*	*44*	*58*
OAL	73.6	83.7	66.1 / 80.3	54.3 / 71.2	52.6 / 64.6	OAL	15.1	29.4	61.4
|*HITL*|	*38*	*32*	*43*	*54*	*66*	|*HITL*|	*33*	*45*	*58*
Full									
UiTTa-1	73.3	81.4	61.2 / 75.6	52.0 / 68.6	39.1 / 51.6	UiTTa-1	16.0	30.0	63.2
UiTTa-2	73.4	83.8	66.6 / 79.8	54.2 / 70.9	49.6 / 62.0	UiTTa-3	15.2	29.9	62.0
UiTTa-3	73.5	84.3	67.0 / 80.6	54.5 / 71.3	50.5 / 63.0	UiTTa-5	15.0	29.6	61.5
OAL	73.8	84.3	66.2 / 80.8	54.3 / 71.4	52.7 / 64.9	OAL	14.9	29.6	61.0

Table 3: Results of constant adaptation. *Model Uncertainty*: model uncertain data is identified for users to annotate. *Full*: all the test data is annotated by users. |*HITL* (human-in-the-loop)| means the proportion of tokens (NER and POS) or sum of start and end positions (XQuAD and MLQA) or images (image datasets) in test data that is annotated. On the left, average results over 3 random runs are reported except *Source*.

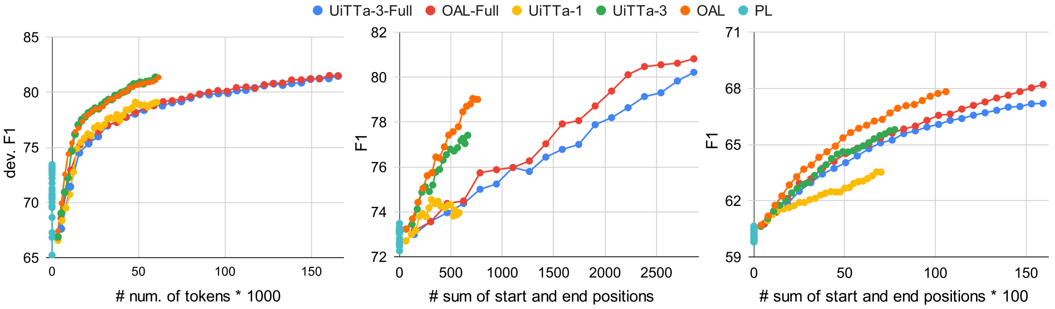


Figure 4: Constant adaptation results w.r.t. the cost of user annotation. Left to right: target language of ru from NER, target language of el from XQuAD, and target domain of HotpotQA from MRQA.

Table 3, in the setting of *Model Uncertainty*, using around 30% to 40% user involvement can achieve comparable results to the setting of *Full*, which is consistent with the result of Fig. 4.

UiTTa with larger K can achieve results comparable to OAL. Contrary to full annotation used in OAL, UiTTa only requires user feedback on top K outputs where the data may not receive a gold label after interaction. However, UiTTa with larger K (3 or 5) can compete with OAL, since the gold label usually exists in the top predictions and there is no need to search the full label space, which is also evidenced by Fig. 2 where the top-3 and -5 accuracies are much better than top-1.

5.3 Continual Adaptation

Continual adaptation requires the model to adapt to test distributions changing over time without stopping. The results are presented in Table 4, where model acceleration is not applied.

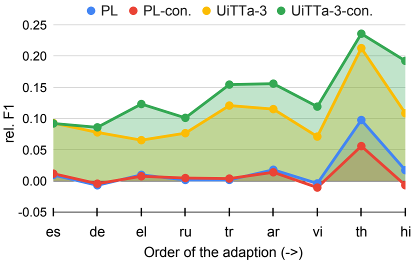
PL and Tent perform worse in continual adaptation. As studied in Wang et al. (2022), continual adaptation is more challenging than constant adaptation, which causes TTA baselines to perform worse in such a case. In this work, we further verify this conclusion on some NLP tasks, shown by the results in the left part of Table 4.

UiTTa generalizes well to continual adaptation. As Table 4 indicates, changing of test distributions cannot result in worse results with UiTTa. Instead, it even improves the results in most of the datasets, especially in image corruption, on which UiTTa-5-con. further improves over UiTTa-5 by about 3 to 6 points. As in Fig. 5, UiTTa-con. outperforms UiTTa during adaptation. Two reasons explain this phenomenon. First, UiTTa eases the problem of overfitting to noisy labels during

Metric	NER F1	POS F1	XQuAD F1	MLQA F1	MRQA F1	Metric	C10-C Err.	C100-C Err.	IN-C Err.
Source	62.9	74.2	70.9	66	52.8	Source	43.5	46.4	82.4
PL	66.1	71.0	72.0	65.8	48.4	Tent	18.6	31.0	64.6
PL-con.	$64.5_{0.8}$	$59.8_{5.2}$	$71.5_{0.3}$	$65.2_{0.3}$	$47.1_{3.3}$	Tent-con.	$20.2_{1.1}$	$62.8_{3.2}$	$66.5_{0.6}$
CoTTa	-	-	-	-	-	CoTTa	$16.3_{0.1}$	$32.6_{0.2}$	$63.7_{1.9}$
UiTTa-1	71.5	79.3	75.0	67.6	51	UiTTa-1	16.1	30.2	64.5
UiTTa-1-con.	$73.4_{0.6}$	$82.2_{3.0}$	$74.7_{0.3}$	$68.2_{0.1}$	$52.0_{0.5}$	UiTTa-1-con.	$15.5_{0.1}$	$29.0_{0.2}$	$60.1_{0.9}$
UiTTa-3	73.3	82.0	78.2	70.6	62.2	UiTTa-5	15.1	29.5	62.6
UiTTa-3-con.	$76.5_{0.1}$	$84.9_{0.3}$	$80.7_{0.2}$	$71.5_{0.2}$	$60.8_{3.7}$	UiTTa-5.con.	$12.6_{0.1}$	$26.4_{0.2}$	$56.9_{0.6}$
OAL	73.4	82.8	80.2	71.2	64.6	OAL	15.1	29.4	61.4
OAL-con.	$76.6_{0.1}$	$85.3_{0.4}$	$81.6_{0.2}$	$71.8_{0.1}$	$65.5_{0.5}$	OAL-con.	$12.5_{0.1}$	$26.0_{0.1}$	$55.3_{0.4}$

Table 4: Results of continual adaptation. Methods with *con.* denote working in the setting of continual adaptation. For each target dataset, we construct 5 random orders of the subsets from the dataset to set the test scenario of changing over time. The average results and standard deviation over these orders are reported. CoTTa is a baseline proposed for continual adaptation for image classification from Wang et al. (2022).

Figure 5: Continual adaptation in one specific order in XQuAD. Relative F1 gains over the source baseline are calculated.



Data		UiTTa-3	N = 12		
			p = 0.9	p = 0.8	p = 0.5
POS	Wall-clock Time (s)	*839.7*	*528.3*	*581.2*	*743.3*
	F1	82.9	79.8	80.6	81.4
XQuAD	Wall-clock Time (s)	*632.0*	*418.1*	*443.9*	*551.4*
	F1	79.2	75.4	76.2	78.1
MRQA	Wall-clock Time (s)	*312.5*	*182.8*	*206.6*	*273.0*
	F1	64.5	61.7	62.6	63.6

Table 5: Results of wall-clock time (in seconds) used to train the model on each dataset based on XLMR-base. On each dataset, the total time used is summed up over all subsets.

adaptation. Second, UiTTa can make use of the knowledge learned from the past distributions, which may benefit learning in the current and future adaptation.

5.4 Fast Adaptation

In this section, we show the effectiveness of our method to accelerate model optimization. We first evaluate XLMR-base (Conneau et al., 2020), which has 12 transformer layers. We study our method UiTTa-3. We show the results w.r.t. saved backward cost (calculated by Eq. 4) in Fig. 6, and the wall-clock times used to run the model are also calculated in Table 5. More results evaluated on XLMR-large with UiTTa-1 and -3 are shown in Fig. 8, 9, 10 in the Appendix.

Permanent freezing cannot compete with random freezing. To compare with random freezing, we freeze the layers lower than 6, 9, and 12 permanently (the rightmost points on each curve are the corresponding results). As Fig. 6 indicates, freezing lower layers starting from no. 6 permanently would have 50% saved cost, but freezing lower 9 or 12 layers randomly, the performance can be improved with the same cost or maintained with less cost. Freezing layers under 9 and 12 permanently would be close to source without adaptation, though around 80% or 100% cost is saved.

Maintaining the same performance, larger N would have more cost saved. From the results of Fig. 6, we find that using larger N can have more backward cost saved. The reason may be that though more lower layers are frozen when N is large, the higher layers can still learn the task information well, which also indicates the large capacity of the pre-trained language model.

Saving up to 70% – 90% backward cost, the performance only drops a bit. For example, in XQuAD and MRQA, with 70% backward cost saved, the F1 result drops by around 2 points in XQuAD and 1.5 points in MRQA. XQuAD drops from 79.2 to 76 when the saved cost further reaches 80%, and MRQA can still maintain at 62 compared to original result 64.5 with 90% cost saved. We further calculate the actual wall-clock time used in model adaptation, in which the forward and backward time costs are both counted. The experiments are conducted on one A100 GPU. As shown in Table 5, we can achieve a good trade-off between performance and optimization cost, e.g., setting N to 12 and p to 0.8. Similar observation can be seen in XLMR-large shown in Table 8.

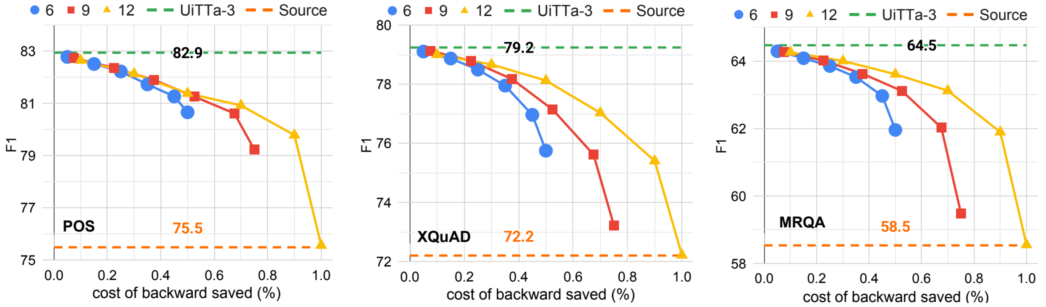


Figure 6: Constant adaptation results on XLMR-base w.r.t. backward cost saved. Various combinations of the layer number N and drop probability p are evaluated, where N is selected from $\{6, 9, 12\}$ and p from $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. $p = 1$ means freezing the lower layers for the whole time of adaptation.

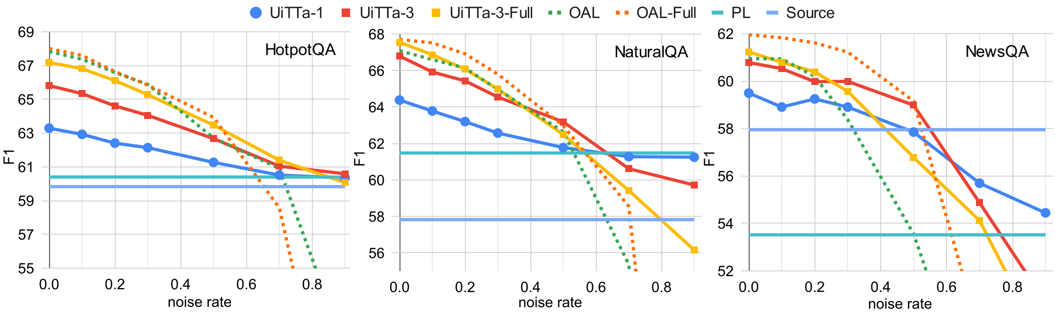


Figure 7: Constant adaptation results with noisy user feedback. Noise rate is the probability of generating the two types of noises which are combined here. *-Full* means all the data are annotated by users.

5.5 Analysis of Perturbation in User Feedback

Users may leave noisy/wrong feedback to the model. In our case, we can consider two types of noise that the user may generate, where the user regards a wrong label as correct (feedback is $\langle 1, (x, y^{(1)}) \rangle$ but $y^{(1)}$ is a wrong label), or the user cannot recognize the gold label existing in the top-K list and the final feedback is $\langle 0, (x, y^{(0)}) \rangle$. To simulate the first type of noise, we randomly return a label from the top-K predictions. For the second type of noise, feedback with $r = 1$ is randomly set to 0. We study constant adaptation and the results are in Fig. 7, where the noisy rate is the probability to generate the two types of noises which are combined.

UiTTa can tolerate a large noise rate. The noisy feedback can bring down the performance of UiTTa. However, when the noise rate is smaller than 20%, UiTTa still has a large margin over PL and source. Furthermore, we find UiTTa-1 and UiTTa-3 can still be better than the baselines of PL and source under noise rate up to 40%.

However, extremely noisy rates can fail UiTTa and larger K would have worse performance. When the noise rate reaches around 50%, UiTTa would have no advantages over PL and source, and could have even worse results when exceeding 50%. Larger K will result in worse performance after UiTTa starts to be worse than PL and source, since larger K would cause a high probability to return a wrong label from the top-K predictions. In practice, users can be confused by the many predictions returned by the model and would make a mistake more easily.

6 Conclusion

We study test-time adaptation with model-user interaction. UiTTa improves the efficiency of user interaction to make online data labeling less costly. UiTTa explores user feedback on top K model predictions, which is cheaper than full annotation adopted in active learning. Simulation experiments demonstrate its effectiveness. We also propose a simple but effective method to speed up model optimization.

REFERENCES

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4623–4637, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.421. URL https://aclanthology.org/2020.acl-main.421.

Alexander Bartler, Andre Bühler, Felix Wiewel, Mario Döbler, and Bin Yang. MT3: meta test-time training for self-supervised test-time adaption. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pp. 3080–3090. PMLR, 2022. URL https://proceedings.mlr.press/v151/bartler22a.html.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=BJ8vJebC-.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *ArXiv preprint*, abs/1704.05179, 2017. URL https://arxiv.org/abs/1704.05179.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7949–7962, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.638. URL https://aclanthology.org/2020.emnlp-main.638.

Ahmed Elgohary, Saghar Hosseini, and Ahmed Hassan Awadallah. Speak to your parser: Interactive text-to-sql with natural language feedback. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 2065–2077. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.187. URL https://doi.org/10.18653/v1/2020.acl-main.187.

Ahmed Elgohary, Christopher Meek, Matthew Richardson, Adam Fourney, Gonzalo Ramos, and Ahmed Hassan Awadallah. NL-EDIT: Correcting semantic parse errors through natural language interaction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5599–5610, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.444. URL https://aclanthology.org/2021.naacl-main.444.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pp. 1–13, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5801. URL https://aclanthology.org/D19-5801.

Wee Chung Gan and Hwee Tou Ng. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6065–6075, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1610. URL https://aclanthology.org/P19-1610.

Ge Gao, Eunsol Choi, and Yoav Artzi. Simulating bandit learning from user feedback for extractive question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 5167–5179. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.acl-long.355.

Yang Gao, Christian M. Meyer, and Iryna Gurevych. APRIL: interactively learning to summarise by combining active preference learning and reinforcement learning. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 4120–4130. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1445. URL https://doi.org/10.18653/v1/d18-1445.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. Robustness gym: Unifying the NLP evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pp. 42–55, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-demos.6. URL https://aclanthology.org/2021.naacl-demos.6.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4411–4421. PMLR, 2020. URL http://proceedings.mlr.press/v119/hu20b.html.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 2427–2440, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/1415fe9fea0fa1e45dddcff5682239a0-Abstract.html.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Ansh Khurana, Sujoy Paul, Piyush Rai, Soma Biswas, and Gaurav Aggarwal. SITA: single image test-time adaptation. *CoRR*, abs/2112.02355, 2021. URL https://arxiv.org/abs/2112.02355.

Bernhard Kratzwald, Stefan Feuerriegel, and Huan Sun. Learning a cost-effective annotation policy for question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 3051–3062. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.246. URL https://doi.org/10.18653/v1/2020.emnlp-main.246.

Julia Kreutzer and Stefan Riezler. Self-regulated interactive sequence-to-sequence learning. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 303–315. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1029. URL https://doi.org/10.18653/v1/p19-1029.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Carolin Lawrence and Stefan Riezler. Improving a neural semantic parser by counterfactual learning from human bandit feedback. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 1820–1830. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1169. URL https://aclanthology.org/P18-1169/.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7315–7330, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.653. URL https://aclanthology.org/2020.acl-main.653.

Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. Robustness testing of language understanding in task-oriented dialog. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2467–2480, Online, 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.192. URL https://aclanthology.org/2021.acl-long.192.

Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34, 2021b.

Vânia Mendonça, Ricardo Rei, Luísa Coheur, Alberto Sardinha, and Ana Lúcia Santos. Online learning meets machine translation evaluation: Finding the best systems with the least human effort. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 3105–3117. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.242. URL https://doi.org/10.18653/v1/2021.acl-long.242.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What happens to BERT embeddings during fine-tuning? In Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupala, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad (eds.), *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2020, Online, November 2020*, pp. 33–44. Association for Computational Linguistics, 2020.

doi: 10.18653/v1/2020.blackboxnlp-1.4. URL https://doi.org/10.18653/v1/2020.blackboxnlp-1.4.

Zachary Nado, Shreyas Padhy, D. Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *CoRR*, abs/2006.10963, 2020. URL https://arxiv.org/abs/2006.10963.

Khanh Nguyen, Hal Daumé III, and Jordan L. Boyd-Graber. Reinforcement learning for bandit neural machine translation with simulated human feedback. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 1464–1474. Association for Computational Linguistics, 2017. doi: 10.18653/v1/d17-1153. URL https://doi.org/10.18653/v1/d17-1153.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 2022. URL https://proceedings.mlr.press/v162/niu22a.html.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, et al. Universal dependencies 2.2. 2018.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1946–1958, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1178. URL https://aclanthology.org/P17-1178.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 11557–11568. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01139. URL https://openaccess.thecvf.com/content/CVPR2021/html/Pham_Meta_Pseudo_Labels_CVPR_2021_paper.html.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Abhilasha Ravichander, Siddharth Dalmia, Maria Ryskina, Florian Metze, Eduard Hovy, and Alan W Black. NoiseQA: Challenge set evaluation for user-centric question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2976–2992, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.259. URL https://aclanthology.org/2021.eacl-main.259.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442.

Barbara Rychalska, Dominika Basaj, Alicja Gosiewska, and Przemysław Biecek. Models in the wild: On corruption robustness of neural nlp systems. In *International Conference on Neural Information Processing*, pp. 235–247. Springer, 2019.

Burr Settles. Active learning literature survey. 2009.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *CoRR*, abs/2208.03188, 2022. doi: 10.48550/arXiv.2208.03188. URL `https://doi.org/10.48550/arXiv.2208.03188`.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/06964dce9addb1c5cb5d6e3d9838f733-Abstract.html`.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize from human feedback. *CoRR*, abs/2009.01325, 2020. URL `https://arxiv.org/abs/2009.01325`.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020. URL `http://proceedings.mlr.press/v119/sun20b.html`.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 191–200, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2623. URL `https://aclanthology.org/W17-2623`.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633, 2020. doi: 10.1162/tacl_a_00335. URL `https://aclanthology.org/2020.tacl-1.40`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL `https://openreview.net/forum?id=uXl3bZLkr3c`.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. *CoRR*, abs/2203.13591, 2022. doi: 10.48550/arXiv.2203.13591. URL `https://doi.org/10.48550/arXiv.2203.13591`.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pp. 347–355, Online, 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-demo.41. URL https://aclanthology.org/2021.acl-demo.41.

Xuezhi Wang, Haohan Wang, and Diyi Yang. Measure and improve robustness in nlp models: A survey. *ArXiv preprint*, abs/2112.08313, 2021c. URL https://arxiv.org/abs/2112.08313.

Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 10684–10695. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01070. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Xie_Self-Training_With_Noisy_Student_Improves_ImageNet_Classification_CVPR_2020_paper.html.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259.

Ziyu Yao, Yiqi Tang, Wen-tau Yih, Huan Sun, and Yu Su. An imitation game for learning semantic parsers from user interaction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6883–6902, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.559. URL https://aclanthology.org/2020.emnlp-main.559.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 3320–3328, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/375c71349b295fbe2dcdca9206f20a06-Abstract.html.

	NER	**POS**	**XQuAD**	**MLQA**	**CIFAR10-C**	**CIFAR100-C**	**ImageNet-C**
PL	LR: 5e-7 T: 5 BATCH_SIZE: 32	LR: 5e-7 T: 5 BATCH_SIZE: 32	T: 5 BATCH_SIZE: 8 LR: 1e-6	T: 3 BATCH_SIZE: 16 LR: 1e-6	T: 1	T: 1	T: 5 (constant) 1 (continual)
Tent	LR: 5e-7 T: 5 BATCH_SIZE: 32	LR: 5e-7 T: 5 BATCH_SIZE: 32	T: 5 BATCH_SIZE: 8 LR: 1e-6	T: 3 BATCH_SIZE: 16 LR: 1e-6	T: 1	T: 1	T: 5 (constant) 1 (continual)
UiTTa	warm-up: 0.01 α: 1 K: 1,2,3 LR: 5e-7 T: 5 BATCH_SIZE: 32	warm-up: 0.01 α: 1 K: 1,2,3 LR: 5e-7 T: 5 BATCH_SIZE: 32	T: 5 warm-up: 0.01 α: 1 K: 1,2,3 BATCH_SIZE: 8 LR: 1e-6	T: 3 warm-up: 0.01 α: 1 K: 1,2,3 BATCH_SIZE: 16 LR: 1e-6	warm-up: 0.05 K: 1,3,5 α: 1 T: 5	warm-up: 0.05 K: 1,3,5 α: 1 T: 5	warm-up: 0.05 K: 1,3,5 α: 1 T: 5
OAL	warm-up: 0.01 α: 1 K: inf LR: 5e-7 T: 5 BATCH_SIZE: 32	warm-up: 0.01 α: 1 K: inf LR: 5e-7 T: 5 BATCH_SIZE: 32	T: 5 warm-up: 0.01 α: 1 K: inf BATCH_SIZE: 8 LR: 1e-6	T: 3 warm-up: 0.01 α: 1 K: inf BATCH_SIZE: 16 LR: 1e-6	warm-up: 0.05 K: inf α: 1 T: 5	warm-up: 0.05 K: inf α: 1 T: 5	warm-up: 0.05 K: inf α: 1 T: 5

Table 6: Hyper-parameters used for each baseline. For CIFAR10-C, CIFAR100-C and ImageNet-C, other hyper-parameters can be found in Wang et al. (2022).

MRQA	**HotpotQA**	**NaturalQA**	**NewsQA**	**SearchQA**	**TriviaQA**
PL	T: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 BATCH_SIZE: 32 LR: 1e-6
Tent	T: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 BATCH_SIZE: 32 LR: 1e-6
UiTTa	T: 1 warm-up: 0.01 α: 1 K: 1,2,3 BATCH_SIZE: 32 LR: 1e-6	T: 1 warm-up: 0.01 α: 1 K: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 warm-up: 0.01 α: 1 K: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 warm-up: 0.01 α: 0.3 K: 1 BATCH_SIZE: 32 LR: 1e-6	T: 1 warm-up: 0.01 α: 0.3 K: 1 BATCH_SIZE: 32 LR: 1e-6
OAL	T: 1 warm-up: 0.01 α: 1 K: inf BATCH_SIZE: 32 LR: 1e-6	T: 1 warm-up: 0.01 α: 1 K: inf BATCH_SIZE: 32 LR: 1e-6	T: 1 warm-up: 0.01 α: 1 K: inf BATCH_SIZE: 32 LR: 1e-6	T: 1 warm-up: 0.01 α: 0.3 K: inf BATCH_SIZE: 32 LR: 1e-6	T: 1 warm-up: 0.01 α: 0.3 K: inf BATCH_SIZE: 32 LR: 1e-6

Table 7: Hyper-parameters in MRQA.

7 Appendix

7.1 Training Details

For experiments of cross-lingual transfer and domain generation, the source models are all based on a pre-trained language model which is XLMR-base (Conneau et al., 2020). For UiTTa and OAL, 1% test data is used to do warm-up to obtain the threshold $\bar{l}$. For POS and NER tasks, the user feedback is applied to token levels. And for QA tasks, the start and end positions are treated independently and receive the user feedback separately. For experiments on corruption robustness, we take the model settings from Wang et al. (2022). The source models for CIFAR10-C, CIFAR100-C and ImageNet-C are WideResNet-28, Hendrycks2020AugMix-ResNeXt and Standard-R50, respectively. The proportion of test data to do warm-up is 5%. For all of the baselines, we try to maintain the batch size, learning rate and etc. that are independent from the algorithm itself the same.

We show the hyper-parameters for model training in Table 6 and Table 7.

7.2 More Experimental Results

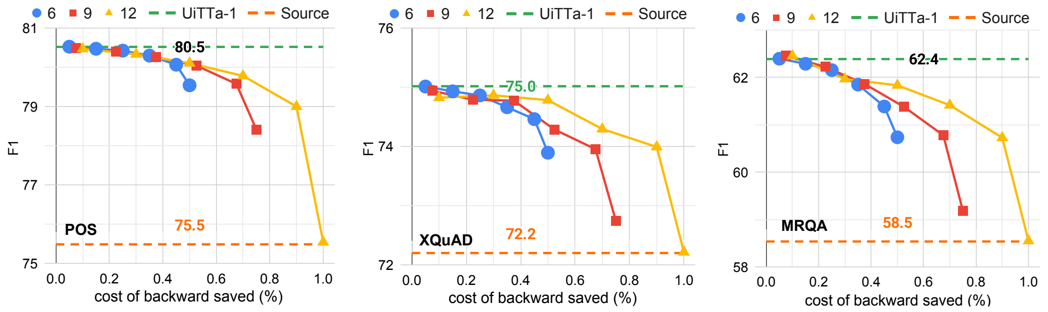


Figure 8: Constant adaptation results w.r.t. cost of backward saved based on UITTA-1. XLMR-base is accelerated. Various combinations of the layer N and drop probability p are evaluated, where N is selected from $\{6, 9, 12\}$ and p from $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. $p = 1$ means freezing the lowers layers for the whole time of adaptation.

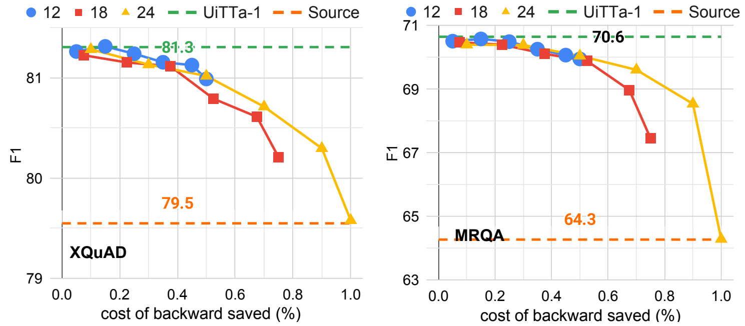


Figure 9: Constant adaptation results w.r.t. cost of backward saved based on UITTA-1. XLMR-large is accelerated. Various combinations of the layer N and drop probability p are evaluated, where N is selected from $\{12, 18, 24\}$ and p from $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. $p = 1$ means freezing the lowers layers for the whole time of adaptation.

Data		UITTA-1	$N = 24$		
			$p = 0.9$	$p = 0.7$	$p = 0.5$
XQuAD	Wall-clock time (s)	1,079.9	587.7	698.2	811.1
	F1	81.3	80.3	80.7	81.0
MRQA	Wall-clock time (s)	1,167.4	590.4	720.3	857.0
	F1	70.6	68.5	69.6	70.0
Data		UITTA-3	$N = 24$		
			$p = 0.9$	$p = 0.7$	$p = 0.5$
XQuAD	Wall-clock time (s)	1,080.8	603.0	706.4	817.7
	F1	85.4	81.7	83.4	84.2
MRQA	Wall-clock time (s)	1,166.4	593.4	721.4	854.3
	F1	74.2	70.4	72.0	72.8

Table 8: The wall-clock time (measured in seconds) used to train the model on each dataset, where XLMR-large is evaluated. On each dataset, the total used time is summed over all subsets.

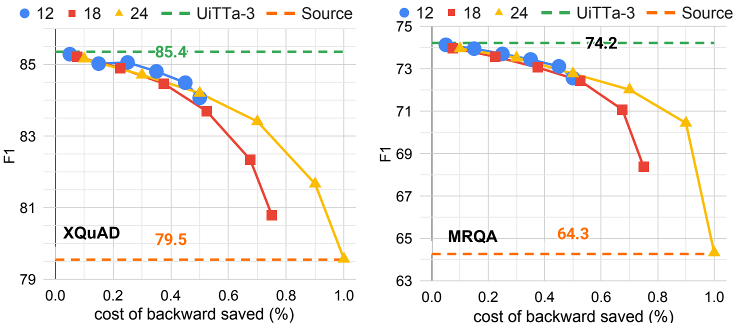


Figure 10: Constant adaptation results w.r.t. cost of backward saved based on UITTA-3. XLMR-large is accelerated. Various combinations of the layer N and drop probability p are evaluated, where N is selected from $\{12, 18, 24\}$ and p from $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. $p = 1$ means freezing the lowers layers for the whole time of adaptation.