A Multi-armed bandits

The Multi-Armed Bandit (**MAB**) setting is a problem from machine learning where a learner interacts with an environment over N rounds by following a policy π. At each round t the learner chooses one of the environment's K arms, $a \in \mathcal{A}$ where $K = |\mathcal{A}|$, after which the environment provides a reward R_t. Rewards for unplayed arms are not observed. The goal of the learner is to adopt a policy π that selects actions that lead to the largest cumulative reward over N rounds, $R = \sum_{t=1}^{N} R_t$. In this work we assume a finite K and that the underlying reward distribution of each arm may have a variety of properties (e.g. stochasticity or stationarity) depending on the exact scenario, leading to different optimal policies [34].

Adversarial MAB. The adversarial MAB setting assumes that the reward-generating process is controlled by an adversary. This assumption allows for modelling non-stationary and highly stochastic reward signals. We will later show why our FLAD formulation fits into this setting. Under this setting, it is assumed that an adversary is given access to the learner's policy π and determines the sequence of rewards, $(R_{a,t})_{t=1}^{N}$, for each arm prior to play [38]. At each turn π determines a distribution over actions, $p(\mathcal{A})$, and an action is sampled from the distribution, $a \sim p(\mathcal{A})$. See Lattimore & Szepesvári [34] for further details.

The EXP3 algorithm. The EXP3 algorithm ("*Exp*onential-weight algorithm for *Exp*loration and *Exp*loitation") targets the adversarial multi-armed bandit problem by choosing arms according to a Gibbs distribution based on the empirically determined importance-weighted rewards of arms [21]. To allow for exploration, EXP3 mixes the Gibbs distribution with a uniform distribution.

Formally, let the exploration rate be $\gamma \in (0, 1]$. At round t, π defines the probability of selecting a given arm, $a \in \mathcal{A}$, as a linear combination of Gibbs and uniform distributions

$$p_t(a) = (1 - \gamma)\frac{\exp(\gamma \hat{R}_{a,t-1}/K)}{\sum_{a'} \exp(\gamma \hat{R}_{a',t-1}/K)} + \frac{\gamma}{K} \tag{1}$$

where the importance weighted reward $\hat{R}_{a,t}$ is calculated as

$$\hat{R}_{a,t} = \hat{R}_{a,t-1} + \frac{R_{a,t}}{p_{t-1}(a)} \tag{2}$$

and $R_{a,t}$ denotes the observed reward. All unplayed arms, $a' \neq a$ have unchanged importance weighted rewards; $\hat{R}_{a',t} = \hat{R}_{a',t-1}$.

Algorithmically, EXP3 takes the following steps at each round: First, calculate the sampling distribution p_t and sample an arm from the distribution. Then a reward $R_{a,t}$ is observed and the algorithm updates the importance weighted reward $\hat{R}_{a,t}$ for the played arm.

Informally, the use of an importance-weighted estimated reward compensates the rewards of actions that are less likely to be chosen, guaranteeing that the expected estimated reward is equal to the actual reward for each action. EXP3 is designed to be nearly optimal in the worst case, but due to the exploration rate it will select "bad" actions at a rate of γ/K. The exploration of EXP3 combined with importance-weighting allows the policy to handle non-stationary reward-generating processes.

The UCB1 algorithm. While the adversarial setting makes almost no assumptions about the reward-generating process and therefore maintains its performance guarantees under almost any circumstances, it can be outperformed in settings that *are* constrained. In this section we assume that the reward-generating processes are stationary Gaussian distributions. A common policy used to solve this MAB setting is the Upper Confidence Bound (UCB1) algorithm, which assigns each arm a value called the upper confidence bound based on Hoeffding's inequality [22]. The UCB1 algorithm is based on the principle of *optimism in the face of uncertainty*, meaning that with high probability the upper confidence bound assigned to each arm is an overestimate of the unknown mean reward.

Formally, let the estimated mean reward of arm a after being played n_a times be $\hat{R}_a$ and the true mean reward be R_a, then

$$\mathbb{P}\left(R_a \geq \hat{R}_a + \sqrt{\frac{2\ln(1/\delta)}{n_a}}\right) \leq \delta \quad \forall \delta \in (0, 1)$$

derived from Hoeffding's inequality (following equation 7.1 of Lattimore & Szepesvári [34]), where δ is the confidence level that quantifies the degree of certainty in the arm. In this work we let $\delta = 1/t$ where t is the number of rounds played, shrinking the confidence bound over rounds. Thus, we define the upper confidence bound for arm a at turn t as

$$UCB_{a,t} = \begin{cases} \infty, & \text{if } n_a = 0 \\ \hat{R}_a + \sqrt{\frac{2 \ln t}{n_a}}, & \text{otherwise} \end{cases} \tag{3}$$

Algorithmically, UCB1 takes the following steps at each round. First, the UCB1 policy plays the arm with largest upper confidence bound, $a^* = \arg\max_{a \in \mathcal{A}} UCB_{a,t}$. Next, a reward $R_{a^*,t}$ is observed and the algorithm updates $\hat{R}_{a^*}$ (the estimated mean reward for a^*) and the upper confidence bounds for all a. Informally, this algorithm suggests that the learner should play arms more often if they either 1. have large expected reward, $\hat{R}$, or 2. n_a is small because the arm is not well explored.

B Pseudo-code

We include here pseudo-code for our 2 proposed algorithms. Algorithm 1 contains the pseudo-code for EXP3-FLAD, and Algorithm 2 contains the pseudo-code for UCB1-FLAD.

Algorithm 1 EXP3-FLAD

Require: $\mathcal{D}_{\mathcal{A}}, \mathcal{D}_{\mathcal{T}}$: Auxiliary and target datasets
Require: f_θ: Parameterized model
Require: G: Gradient accumulation steps
Initialize: $K = |\mathcal{A}|$; $\mathcal{E}_0 = \frac{1}{K}$;
  $\forall a \in \mathcal{A} : \nabla_a = 0, \hat{R}_a = 1$
for $t = 1, 2, \ldots, N$ **do**
  $\mathcal{E}_t = \min\left\{ \frac{1}{K}, \sqrt{\frac{\ln K}{K \cdot t}} \right\}$
  $\forall a \in \mathcal{A} : \pi(a) \leftarrow (1 - K\mathcal{E}_t) \frac{\exp(\mathcal{E}_{t-1} \hat{R}_a)}{\sum_{a'} \exp(\mathcal{E}_{t-1} R_{a'})} + \mathcal{E}_t$
  Sample $a \sim \pi(\mathcal{A})$ and batch $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_a$
  $\nabla_a \leftarrow \nabla_a + \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$
  if $t \pmod G \equiv 0$ **then**
    $\nabla_{\mathcal{T}} \leftarrow \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_{\mathcal{T}})$
    Update model parameters w.r.t. $\nabla_{\mathcal{T}} + \sum_a \nabla_a$
    for all $\{a \in \mathcal{A} | \nabla_a \neq 0\}$ **do**
      $\hat{R}_a \leftarrow \hat{R}_a + \frac{R_{a,t}}{\pi(a)}$
      $\nabla_a \leftarrow 0$
    end for
  end if
end for

Algorithm 2 UCB1-FLAD

Require: $\mathcal{D}_{\mathcal{A}}, \mathcal{D}_{\mathcal{T}}$: Auxiliary and target datasets
Require: f_θ: Parameterized model
Require: G: Gradient accumulation steps
Require: β: Smoothing factor

Initialize:
  $\forall a \in \mathcal{A} : n_a = 1,$
    $\hat{R}_a = \cos(\nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_{\mathcal{T}}), \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_a))$
for $t = 1, 2, \ldots, N$ **do**
  $a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \ \hat{R}_a + \sqrt{\frac{2 \ln t}{n_a}}$
  Sample batch $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_{a^*}$
  $\nabla_{a^*} \leftarrow \nabla_{a^*} + \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$
  $n_{a^*} \leftarrow n_{a^*} + 1$
  if $t \pmod G \equiv 0$ **then**
    $\nabla_{\mathcal{T}} \leftarrow \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_{\mathcal{T}})$
    Update model parameters w.r.t. $\nabla_{\mathcal{T}} + \sum_a \nabla_a$
    for all $\{a \in \mathcal{A} | \nabla_a \neq 0\}$ **do**
      $\hat{R}_a \leftarrow (1 - \beta)\hat{R}_a + \beta R_{a,t}$
      $\nabla_a \leftarrow 0$
    end for
  end if
end for

C Training details

We train all models (FLAD and non-FLAD) on 40Gb A100s.

For all experiments, we use validation-based early stopping, and train for a maximum of 10,000 gradient update steps. In practice, we find that early-stopping leads to significantly fewer than 10,000 updates, usually between 50-150 for direct fine-tuning, and 1-2,000 for other methods.

For the smoothing factor, β, in UCB1-FLAD we ran preliminary experiments using values of $\{0.99, 0.9, 0.75, 0.5\}$ and found 0.9 to work well across datasets. All reported scores use $\beta = 0.9$.

In preliminary experiments we consider rewards using gradients from multiple model partitions: the full model, encoder-only, decoder-only, and language modelling head (token classifier). We find that using the parameters from the LM head provides best performance, followed by the decoder-only, encoder-only, and full model gradients. The differential from best to worst method was $\sim 3\%$ relative performance. Recall that with a gradient accumulation factor of G, our algorithms need to store at most $G + 1$ gradients at any time. So not only does using the LM head provide performance improvements, but also saves memory. For the models we use, the LM head contains only 2.3% of the full model parameters.

D Full results

The full results of experiments on target-only fine-tuning, explore-only, exploit-only, EXP3-FLAD, and UCB1-FLAD are found on the next page.

Table 2: Detailed results from the main experiment including direct fine-tuning, exploration-only, exploitation-only baselines and our proposed methods, EXP3-FLAD and UCB1-FLAD.

	Dataset	Anli-r1	Anli-r2	Anli-r3	CB	COPA	HellaSwag	RTE	Story Cloze	WiC	Winogrande	WSC	Average
T5-3B	Direct Fine-Tuning	37.6	36.2	35.0	83.2	53.8	51.0	54.2	75.9	51.6	49.6	53.1	52.8
	Loss-Scaling (GA)	35.7	36.4	35.3	82.5	58.0	51.8	59.0	79.6	49.8	50.4	46.9	53.2
	Loss-Scaling (GMS)	37.8	37.6	36.0	80.0	76.4	52.6	55.3	85.7	50.6	52.0	51.7	56.0
T0Mix	Exploration-Only	38.1	40.3	36.7	88.6	85.6	51.2	67.6	88.8	51.0	55.5	47.7	59.2
	Exploitation-Only	38.8	40.5	38.0	86.1	86.0	51.1	69.4	89.5	52.8	59.2	46.3	59.8
	EXP3-FLAD ($\mathcal{R}^{GA}$)	40.6	39.9	36.9	86.1	89.8	52.0	76.7	90.8	50.5	60.3	52.9	61.5
	UCB1-FLAD ($\mathcal{R}^{GA}$)	41.8	39.0	38.0	85.4	87.0	52.0	79.1	91.4	49.7	62.7	56.2	62.0
	EXP3-FLAD ($\mathcal{R}^{GMS}$)	42.0	40.2	36.6	87.1	87.2	52.4	77.5	90.9	51.1	61.9	51.9	61.7
	UCB1-FLAD ($\mathcal{R}^{AGG}$)	41.3	39.7	38.0	82.5	89.8	51.0	76.6	90.5	51.0	62.0	56.0	61.7
	EXP3-FLAD ($\mathcal{R}^{AGG}$)	38.6	39.8	39.1	86.8	91.2	51.2	78.8	90.4	50.7	63.0	52.9	62.0
	UCB1-FLAD ($\mathcal{R}^{AGG}$)	42.0	41.0	36.6	88.2	86.8	51.0	77.3	90.3	51.1	63.3	55.4	62.1
P3	Loss-Scaling (GA)	38.7	39.5	34.8	80.7	64.4	52.7	62.9	80.1	50.3	51.9	51.2	55.2
	Loss-Scaling (GMS)	39.2	38.7	36.4	85.0	67.8	51.9	62.4	84.8	50.3	51.8	52.1	56.4
	Exploration-Only	40.1	37.7	36.0	85.4	83.6	52.1	77.3	89.1	51.5	57.2	57.1	60.6
	Exploitation-Only	40.4	37.2	37.3	87.1	84.4	51.0	78.6	90.3	51.3	56.2	51.5	60.5
	EXP3-FLAD ($\mathcal{R}^{GA}$)	46.9	38.8	40.2	89.6	88.0	51.5	76.9	91.2	53.4	66.2	61.9	64.1
	UCB1-FLAD ($\mathcal{R}^{GA}$)	49.1	38.8	40.1	88.6	88.2	51.6	83.7	90.2	54.3	68.0	68.3	65.5
	EXP3-FLAD ($\mathcal{R}^{GMS}$)	46.2	40.6	39.4	88.9	90.4	51.6	85.1	91.3	54.4	65.8	67.5	65.6
	UCB1-FLAD ($\mathcal{R}^{GMS}$)	48.1	40.1	39.1	87.5	89.4	52.0	83.7	89.4	51.7	65.8	70.6	65.2
	EXP3-FLAD ($\mathcal{R}^{AGG}$)	47.6	40.6	40.6	90.0	90.6	51.4	84.5	91.0	53.2	66.7	64.0	65.5
	UCB1-FLAD ($\mathcal{R}^{AGG}$)	47.1	39.0	41.2	86.8	90.4	51.5	85.5	91.1	52.7	66.3	70.6	65.6
T0Mix	Direct Fine-Tuning	40.9	39.1	37.1	79.6	66.4	43.5	67.1	83.2	52.5	54.6	56.7	56.4
	Loss-Scaling (GA)	41.3	40.0	36.9	81.8	78.0	51.2	76.5	86.9	50.7	54.7	56.2	59.5
	Loss-Scaling (GMS)	40.5	40.5	37.8	81.1	79.0	52.0	77.0	88.8	52.7	55.0	60.8	60.5
	Exploration-Only	44.4	40.3	37.0	82.5	85.6	47.9	77.6	90.1	52.1	58.6	56.9	61.2
	Exploitation-Only	42.5	39.3	37.2	84.3	82.8	48.1	79.7	88.8	52.8	57.8	56.3	60.9
	EXP3-FLAD ($\mathcal{R}^{GA}$)	46.2	41.5	37.7	83.9	87.6	49.4	80.0	90.1	52.6	63.4	59.0	62.9
	UCB1-FLAD ($\mathcal{R}^{GA}$)	43.7	40.8	37.6	86.1	85.4	48.6	80.5	91.3	53.4	63.5	61.0	62.9
	EXP3-FLAD ($\mathcal{R}^{GMS}$)	43.4	41.1	38.2	84.6	86.6	49.1	81.0	90.6	53.0	63.1	59.8	62.8
	UCB1-FLAD ($\mathcal{R}^{GMS}$)	43.2	41.2	38.7	86.4	86.6	48.4	82.8	91.4	52.2	61.0	59.4	62.8
	EXP3-FLAD ($\mathcal{R}^{AGG}$)	43.8	41.6	38.0	83.9	87.8	48.9	81.9	90.7	52.5	62.3	59.8	62.8
	UCB1-FLAD ($\mathcal{R}^{AGG}$)	44.0	41.6	38.3	85.4	87.4	48.6	81.1	90.6	53.0	63.1	59.2	62.9
T0-3B	Loss-Scaling (GA)	44.0	40.4	38.9	86.4	77.6	51.0	75.1	86.8	51.7	55.6	59.8	60.7
	Loss-Scaling (GMS)	43.8	38.6	39.3	82.5	79.2	50.6	80.6	89.1	51.6	56.6	56.0	60.7
	Exploration-Only	45.4	40.3	38.0	82.5	87.8	50.6	82.2	88.8	52.4	61.8	60.6	62.8
	Exploitation-Only	45.5	40.0	38.8	87.5	82.2	49.9	79.6	90.9	52.2	60.1	64.8	62.9
	EXP3-FLAD ($\mathcal{R}^{GA}$)	50.4	40.0	41.2	87.9	88.4	49.7	86.1	91.6	52.8	67.5	70.4	66.0
P3	UCB1-FLAD ($\mathcal{R}^{GA}$)	48.2	41.8	41.2	90.0	86.6	50.0	86.1	91.5	53.6	65.6	74.6	66.3
	EXP3-FLAD ($\mathcal{R}^{GMS}$)	49.5	40.8	39.5	87.1	89.2	49.4	85.8	91.4	53.9	65.4	68.7	65.5
	UCB1-FLAD ($\mathcal{R}^{GMS}$)	48.2	41.8	40.5	89.6	88.0	49.6	83.2	91.6	52.6	66.1	74.6	66.0
	EXP3-FLAD ($\mathcal{R}^{AGG}$)	51.1	40.3	39.9	89.6	91.4	49.0	86.5	91.6	52.6	66.4	76.7	66.8
	UCB1-FLAD ($\mathcal{R}^{AGG}$)	49.8	39.9	40.8	86.8	88.4	49.6	84.7	91.0	53.2	68.0	76.9	66.3

E Probing the reward generating processes.

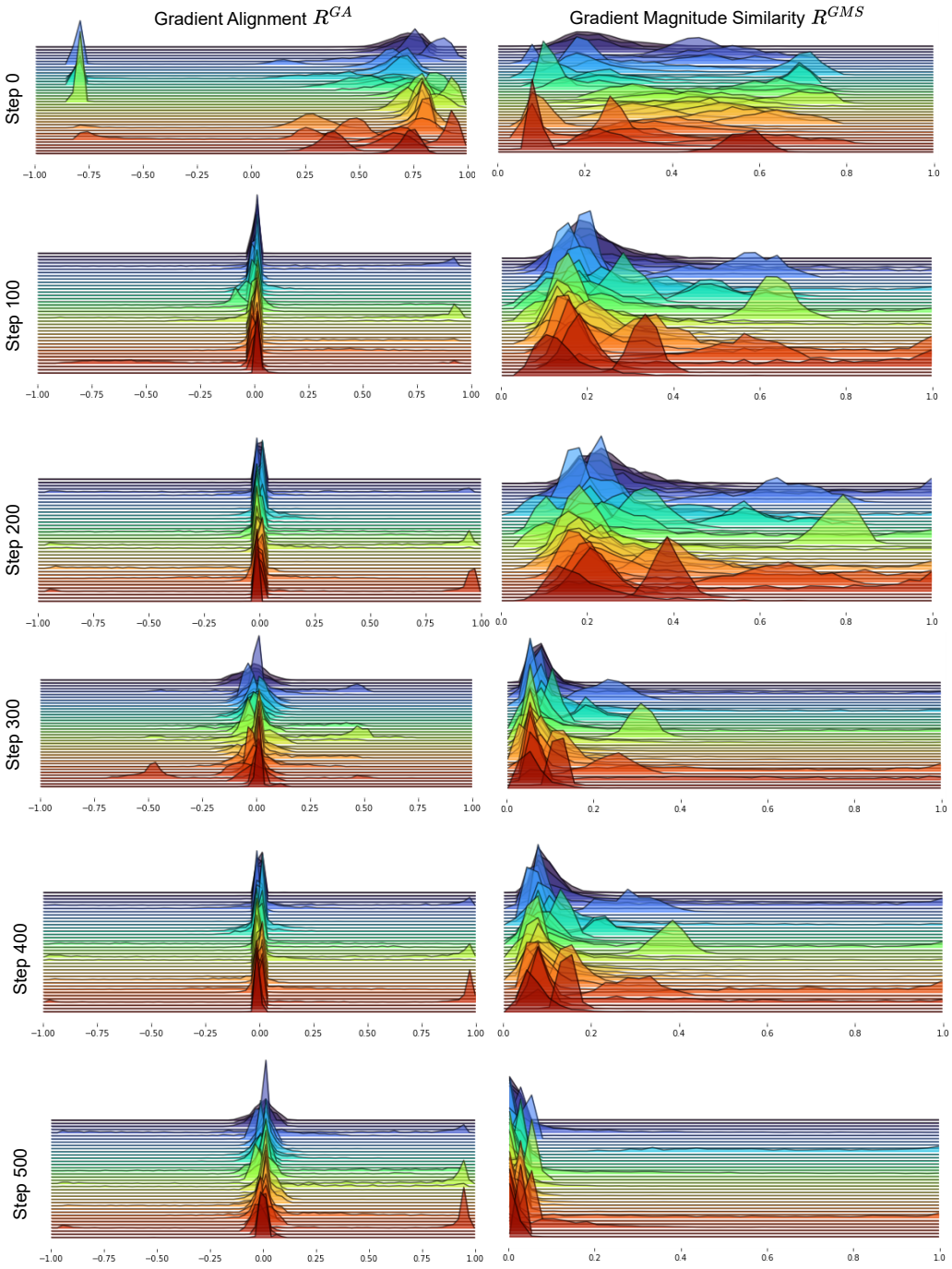

Figure 4: **Reward distributions** of $\mathcal{R}^{GA}$ and $\mathcal{R}^{GMS}$ prior to training and every 100 gradient updates thereafter. We probe the reward distributions using the T5-XL model with the T0Mix auxiliary dataset and WSC [52] as the target dataset.

F EXP3-FLAD and UCB1-FLAD training dynamics

The following 4 pages include a case study on the training dynamics of EXP3-FLAD and UCB1-FLAD when training T5-XL using T0Mix as the auxiliary data. First, we find datasets where EXP3-FLAD and UCB1-FLAD improve significantly over the baseline FLAD methods, but also where either EXP3-FLAD or UCB1-FLAD clearly outperforms the other. The two datasets that fulfill our interests are RTE and COPA.

We find that UCB1-FLAD outperforms EXP3-FLAD on RTE, and show their respective training dynamics in Figure 5 (UCB1) and Figure 6 (EXP3).

We find that EXP3-FLAD outperforms UCB1-FLAD on COPA, and show their respective training dynamics in Figure 7 (UCB1) and Figure 8 (EXP3).

We include details and takeaways in the caption for each figure. For EXP3-FLAD figures, we include charts of the cumulative estimated reward, empirical gradient alignment, instantaneous sampling distribution determined by the policy, and the empirical sampling distribution determined by the total number of samples seen per dataset as a fraction of the total samples seen. For UCB1-FLAD figures, we include charts of the upper confidence index, estimated gradient alignment, and the empirical sampling distribution.

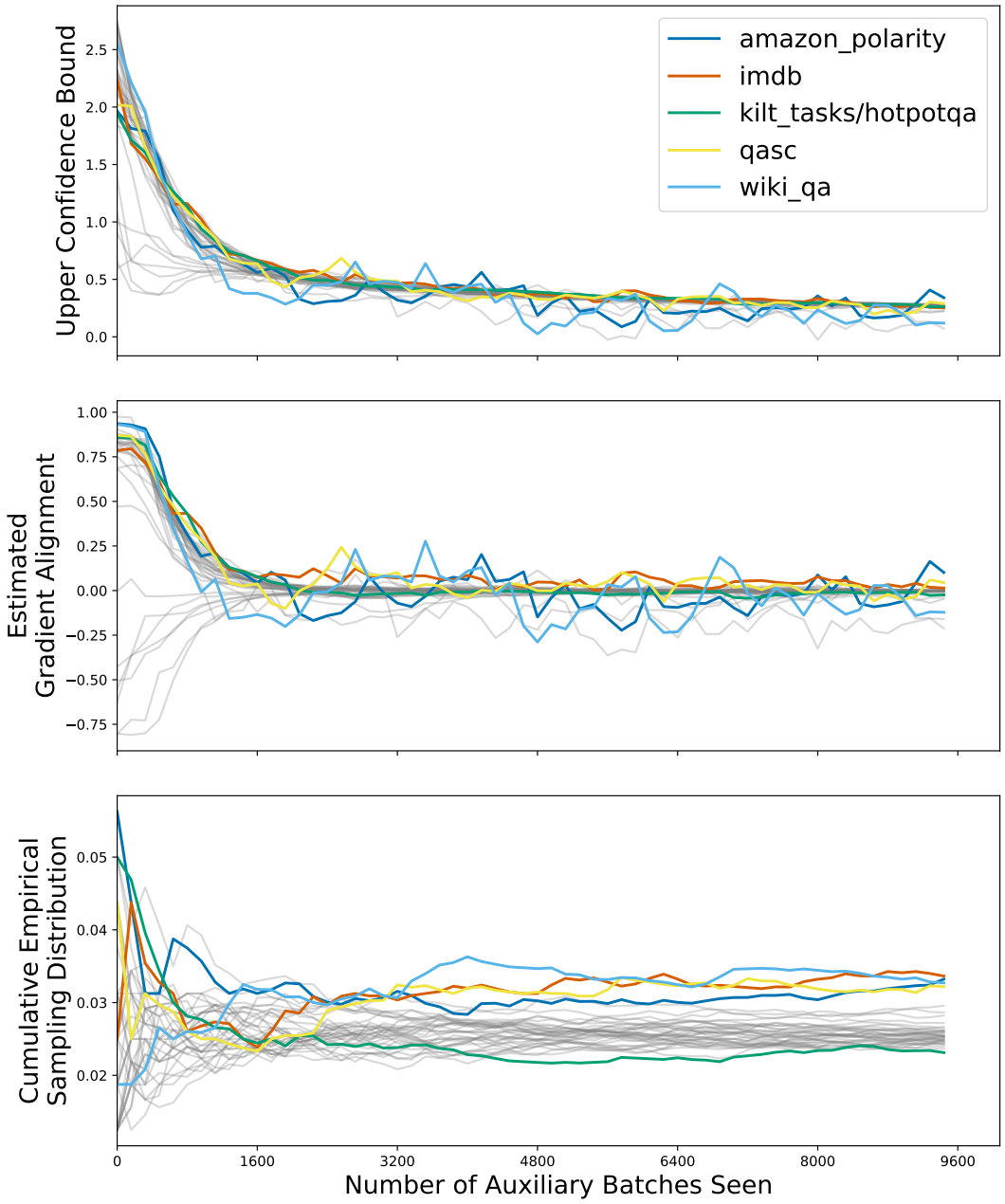

Figure 5: Training dynamics of UCB1-FLAD, a case study using RTE as target dataset and T0Mix as auxiliary data, where UCB1-FLAD outperforms EXP3-FLAD. Colored lines are a sample of auxiliary datasets with interesting properties, the remaining datasets are shown in grey. We find that even though wiki_qa's estimated gradient alignment falls to below 0 (middle), UCB1 does not abandon sampling from it in the future, finding that between 3200 and 4800 batches, it becomes the dataset with largest upper confidence bound (top). Similarly, we see that UCB1 alternates between wiki_qa, amazon_polarity, and qasc as the datasets with higher gradient alignment and upper confidence bounds. kilt_tasks/hotpotqa has a very high gradient alignment prior to training, but UCB1 samples very infrequently from it, due to it'ls lower upper confidence bound. This is a failure case for transfer learning-based methods. Interestingly, UCB1 never estimates imdb to have a negative gradient, and gradually samples from it more and more frequently over the course of training.

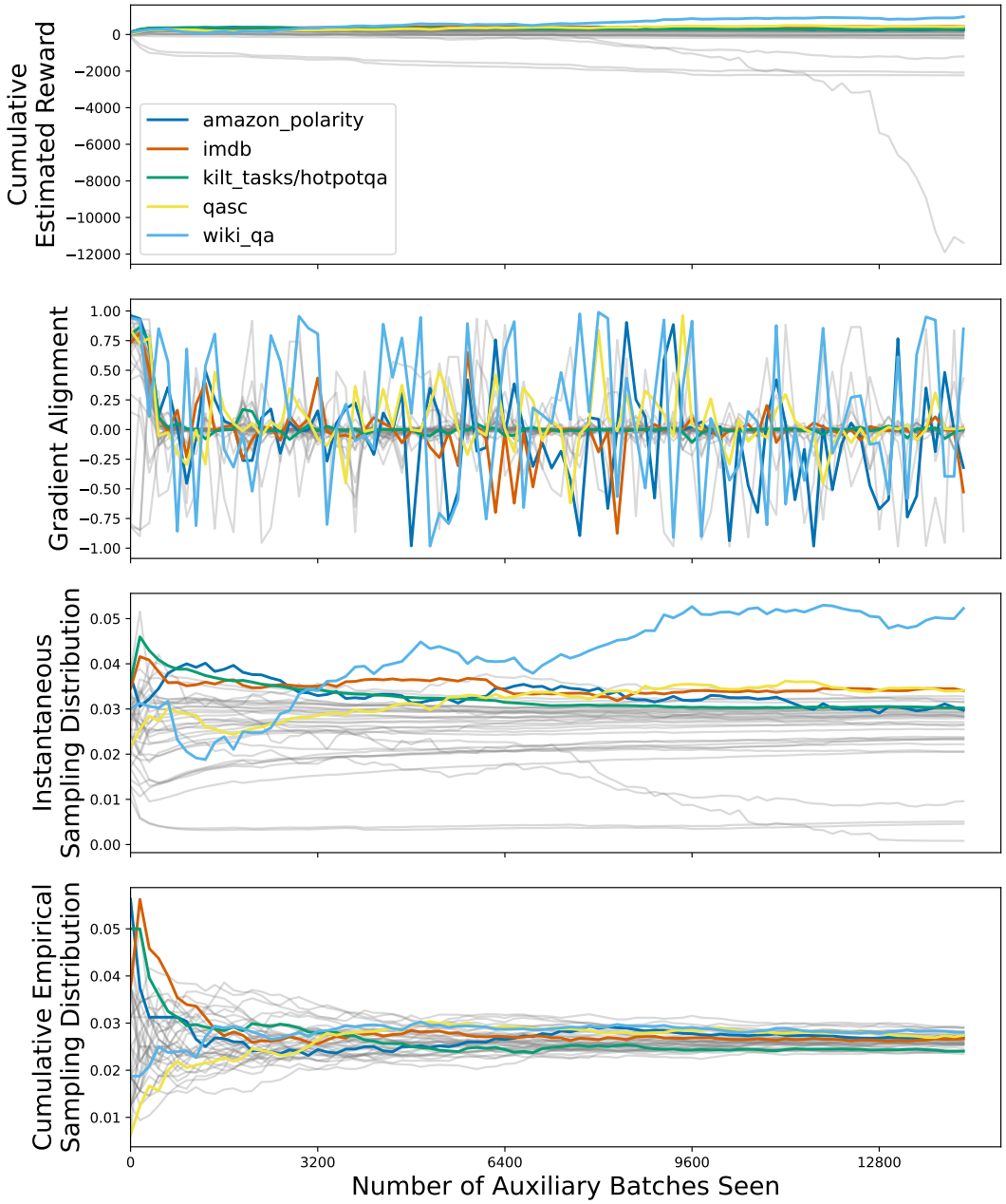

Figure 6: Training dynamics of EXP3-FLAD, a case study using RTE as target dataset and T0Mix as auxiliary data, where UCB1-FLAD outperforms EXP3-FLAD. Colored lines are a sample of auxiliary datasets with interesting properties, the remaining datasets are shown in grey. We find that the gradient alignment signal is particularly noisy for EXP3-FLAD, possibly leading to it's slightly worse performance on RTE. All five highlighted auxiliary datasets have high instantaneous sampling probability, but over the course of training, the empirical sampling distribution is very condensed across the full set of auxiliary datasets, unlike UCB1 which is able to find better separation.

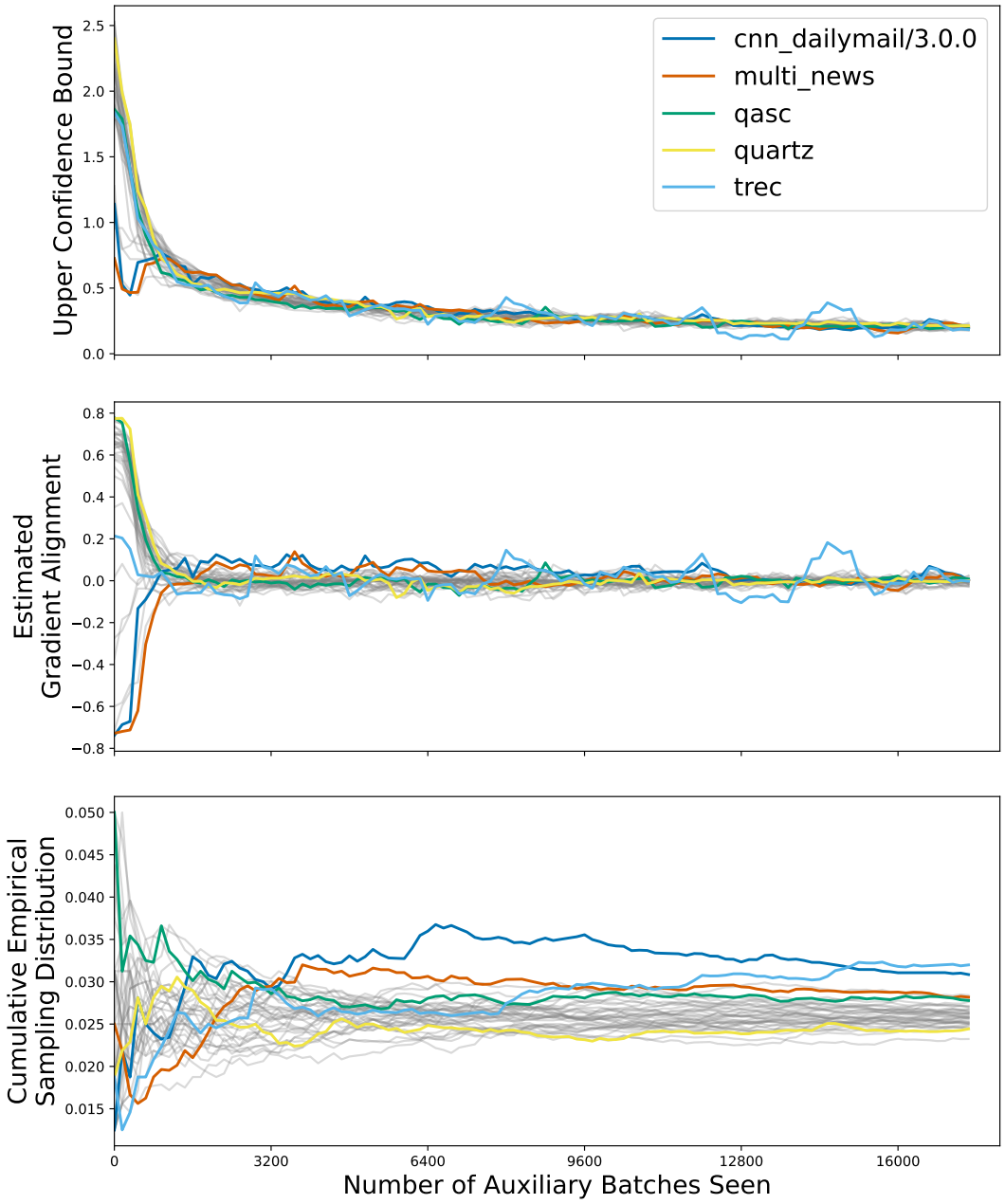

Figure 7: Training dynamics of UCB1-FLAD, a case study using COPA as target dataset and T0Mix as auxiliary data, where EXP3-FLAD outperforms UCB1-FLAD. Colored lines are a sample of auxiliary datasets with interesting properties, the remaining datasets are shown in grey. We find that although qasc and quartz start with very high gradient alignment, they very quickly fall to negative alignment (middle figure, green and yellow). In the end, we find that the algorithm samples much more from qasc than from quartz (bottom figure). Interestingly, we find that although both cnn_dailymail and multi_news start off with very negative gradient alignment, they quickly become the most aligned with the target task (middle figure, blue and red). We find that the three auxiliary datasets with highest upper confidence index (top figure) and largest sampling percent (bottom figure) are cnn_dailymail, multi_news, and trec even though these all considered dissimilar to the target prior to training.

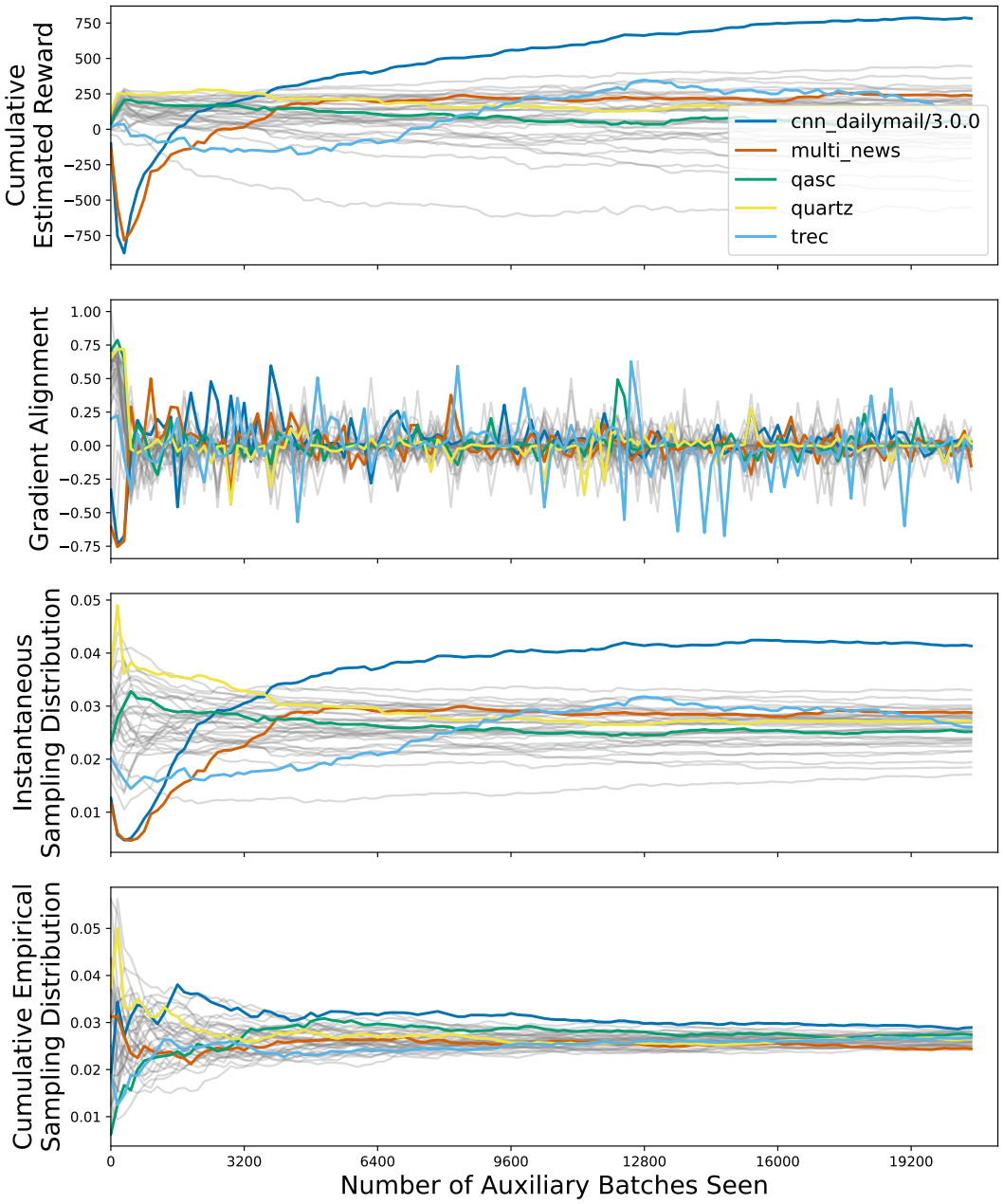

Figure 8: Training dynamics of EXP3-FLAD, a case study using COPA as target dataset and T0Mix as auxiliary data, where EXP3-FLAD outperforms UCB1-FLAD. Colored lines are a sample of auxiliary datasets with interesting properties, the remaining datasets are shown in grey. This is an impressive example of the importance-weighted estimated reward. We see that cnn_dailymail and multi_news both start with very negative alignment, but EXP3 quickly updates it's estimated reward once their alignment becomes positive. Similar to RTE, we see that EXP3 never makes large separations in the empirical sampling distribution, possibly a reason why UCB1 outperforms EXP3 overall. Compared to RTE, we find that gradient alignments are much less variable, with a maximum alignment close to 0.5 and minimum alignment close to -0.5. Whereas in RTE, alignments regularly reach close to 1.0 and -1.0.

G Auxiliary Datasets

Here we include the full list of auxiliary datasets from P3 [23] used to train models for the ANLI target task. Other target datasets have slightly different auxiliary datasets, but are generally the same. Datasets are listed by their name as found in HuggingFace Datasets[2].

Zaid/quac_expanded, acronym_identification, ade_corpus_v2/Ade_corpus_v2_classification, ade_corpus_v2/Ade_corpus_v2_drug_ade_relation, ade_corpus_v2/Ade_corpus_v2_drug_dosage_relation, adversarial_qa/adversarialQA, adversarial_qa/dbert, adversarial_qa/dbidaf, adversarial_qa/droberta, aeslc, ag_news, ai2_arc/ARC-Challenge, ai2_arc/ARC-Easy, amazon_polarity, amazon_reviews_multi/en, amazon_us_reviews/Wireless_v1_00, ambig_qa/light, app_reviews, aqua_rat/raw, art, asset/ratings, asset/simplification, banking77, billsum, bing_coronavirus_query_set, biosses, blbooksgenre/title_genre_classifiction, blended_skill_talk, cbt/CN, cbt/NE, cbt/P, cbt/V, cbt/raw, cc_news, circa, climate_fever, cnn_dailymail/3.0.0, codah/codah, codah/fold_0, codah/fold_1, codah/fold_2, codah/fold_3, codah/fold_4, code_x_glue_tc_text_to_code, common_gen, commonsense_qa, conv_ai, conv_ai_2, conv_ai_3, cord19/metadata, cos_e/v1.0, cos_e/v1.11, cosmos_qa, covid_qa_castorini, craffel/openai_lambada, craigslist_bargains, crows_pairs, dbpedia_14, discofuse/discofuse-sport, discofuse/discofuse-wikipedia, discovery/discovery, docred, dream, drop, duorc/ParaphraseRC, duorc/SelfRC, e2e_nlg_cleaned, ecthr_cases/alleged-violation-prediction, emo, emotion, enriched_web_nlg/en, esnli, evidence_infer_treatment/1.1, evidence_infer_treatment/2.0, fever/v1.0, fever/v2.0, financial_phrasebank/sentences_allagree, freebase_qa, generated_reviews_enth, gigaword, glue/ax, glue/cola, glue/mnli, glue/mnli_matched, glue/mnli_mismatched, glue/mrpc, glue/qnli, glue/qqp, glue/rte, glue/sst2, glue/stsb, glue/wnli, google_wellformed_query, great_code, guardian_authorship/cross_genre_1, guardian_authorship/cross_topic_1, guardian_authorship/cross_topic_4, guardian_authorship/cross_topic_7, gutenberg_time, hans, hate_speech18, head_qa/en, health_fact, hlgd, hotpot_qa/distractor, hotpot_qa/fullwiki, humicroedit/subtask-1, humicroedit/subtask-2, hyperpartisan_news_detection/byarticle, hyperpartisan_news_detection/bypublisher, imdb, jfleg, kelm, kilt_tasks/hotpotqa, kilt_tasks/nq, lama/trex, lambada, liar, limit, math_dataset/algebra__linear_1d, math_dataset/algebra__linear_1d_composed, math_dataset/algebra__linear_2d, math_dataset/algebra__linear_2d_composed, math_qa, mc_taco, mdd/task1_qa, mdd/task2_recs, mdd/task3_qarecs, medal, medical_questions_pairs, meta_woz/dialogues, mocha, movie_rationales, multi_news, multi_nli, multi_x_science_sum, mwsc, narrativeqa, ncbi_disease, neural_code_search/evaluation_dataset, newspop, nlu_evaluation_data, nq_open, numer_sense, onestop_english, openai_humaneval, openbookqa/additional, openbookqa/main, paws-x/en, paws/labeled_final, paws/labeled_swap, paws/unlabeled_final, piqa, poem_sentiment, pubmed_qa/pqa_labeled, qa_srl, qa_zre, qasc, qed, quac, quail, quarel, quartz, quora, quoref, race/all, race/high, race/middle, riddle_sense, ropes, rotten_tomatoes, samsum, scan/addprim_jump, scan/addprim_turn_left, scan/filler_num0, scan/filler_num1, scan/filler_num2, scan/filler_num3, scan/length, scan/simple, scan/template_around_right, scan/template_jump_around_right, scan/template_opposite_right, scan/template_right, scicite, scientific_papers/arxiv, scientific_papers/pubmed, sciq, scitail/snli_format, scitail/tsv_format, scitldr/Abstract, selqa/answer_selection_analysis, sem_eval_2010_task_8, sem_eval_2014_task_1, sent_comp, sick, sms_spam, snips_built_in_intents, snli, social_i_qa, species_800, squad, squad_adversarial/AddSent, squad_v2, squadshifts/amazon, squadshifts/new_wiki, squadshifts/nyt, sst/default, stsb_multi_mt/en, subjqa/books, subjqa/electronics, subjqa/grocery, subjqa/movies, subjqa/restaurants, subjqa/tripadvisor, super_glue/axb, super_glue/axg, super_glue/boolq, super_glue/multirc, super_glue/record, swag/regular, tab_fact/tab_fact, tmu_gfm_dataset, trec, trivia_qa/unfiltered, turk, tweet_eval/emoji, tweet_eval/emotion, tweet_eval/hate, tweet_eval/irony, tweet_eval/offensive, tweet_eval/sentiment, tweet_eval/stance_abortion, tweet_eval/stance_atheism, tweet_eval/stance_climate, tweet_eval/stance_feminist, tweet_eval/stance_hillary, tydiqa/primary_task, tydiqa/secondary_task, web_questions, wiki_bio, wiki_hop/masked, wiki_hop/original, wiki_qa, wiki_split, wino_bias/type1_anti, wino_bias/type1_pro, wino_bias/type2_anti, wino_bias/type2_pro, winograd_wsc/wsc273, winograd_wsc/wsc285, wiqa, xnli/en, xquad/xquad.en, xquad_r/en, xsum, yahoo_answers_qa, yahoo_answers_topics, yelp_polarity, yelp_review_full, zest

[2] https://huggingface.co/datasets