Regression-Free Model Updates for Spoken Language Understanding

Andrea Caciolai
Amazon Alexa AI
andccl@amazon.it

Verena Weber
Amazon Alexa AI
wverena@amazon.de

Tobias Falke
Amazon Alexa AI
falket@amazon.de

Alessandro Pedrani
Amazon Alexa AI
pedrana@amazon.it

Davide Bernardi
Amazon Alexa AI
dvdbe@amazon.it

Abstract

In real-world systems, an important requirement for model updates is to avoid regressions in user experience caused by flips of previously correct classifications to incorrect ones. Multiple techniques for that have been proposed in the recent literature. In this paper, we apply one such technique, focal distillation, to model updates in a goal-oriented dialog system and assess its usefulness in practice. In particular, we evaluate its effectiveness for key language understanding tasks, including sentence classification and sequence labeling tasks, we further assess its effect when applied to repeated model updates over time, and test its compatibility with mislabeled data. Our experiments on a public benchmark and data from a deployed dialog system demonstrate that focal distillation can substantially reduce regressions, at only minor drops in accuracy, and that it further outperforms naive supervised training in challenging mislabeled data and label expansion settings.

1 Introduction

Machine learning models that are deployed in real-world applications typically require regular updates to accommodate data distribution shifts or changes to the output label space. The retraining process, even if it leads to stable or improved overall accuracy, can result in different sample-level predictions due to its stochastic nature. In an application setting, that in turn can change (or even break) specific functionalities. A key requirement for model updates in real-world applications is therefore to minimize regressions in user experience.

For classification models, Yan et al. (2021) formalized this requirement as minimizing the number of *negative flips* of a model, defined as the number of previously correct classifications that turn incorrect for a new model. Previous work proposed several methods towards that goal (Shen et al., 2020; Yan et al., 2021; Zhao et al., 2022; Träuble et al., 2021) that rely on knowledge distillation, model ensembling or Bayesian learning.

In this work, we focus on model update-caused regressions in goal-oriented dialog systems, and in particular on updates of spoken language understanding models. In real-world dialog systems, a negative flip would mean that a request that was previously correctly understood is now interpreted as a different intent (or with different slots) and therefore leads to a regression in user experience.

While previous work explored the reduction of negative flips on various tasks, spoken language understanding remains unexplored (see section 2). We therefore apply *focal distillation* (Yan et al., 2021), the most applicable existing technique, to this use case. Moreover, the use in a real-world goal-oriented dialog system raises additional questions that we address. Specifically, we study the following:

Effectiveness for DC and IC: We test focal distillation on domain classification (DC) and intent classification (IC), two key tasks in spoken language understanding, using public data as well as internal datasets from a real-world dialog system.

Applicability to SL: We further test the effectiveness for slot labeling (SL), a sequence labeling task that requires an extension of focal distillation to handle tasks with token-level supervision.

Repeated Model Updates: We simulate multiple iterations of retraining with focal distillation to study its long-term effect, in particular, whether the coupling of new and old model via distillation restricts the model's ability to learn new features.

Noisy Labels: Finally, we also study the effect of mislabeled data. In the presence of annotation errors, focal distillation bears the risk that it enforces prediction consistency on samples that have supposedly correct classifications in the old model, but are actually mislabeled, preventing the new model to predict the true correct label.

We run extensive experiments for DC, IC and

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 5: Industry Track, pages 538–551

SL tasks on SLURP (Bastianelli et al., 2020), a public benchmark, and internal datasets from our real-world goal-oriented dialog system. We find that focal distillation is effective for DC and IC and reduces negative flips by up to 30% relative at no or only marginal decreases in accuracy. For SL, a naive application as a token-level loss is effective as well and brings 8% relative reduction on average. When simulating repeated retraining over time, focal distillation can restrict the model's ability to learn new labels, but this can be remedied by warm-starting the model with the previous model's weights. Finally, we also show that focal distillation is beneficial even under annotation errors, and can be made even more robust by adding noise-awareness to the loss.

2 Related Work

Enabling regression-free model updates is a relatively recent line of research. Shen et al. (2020) first studied it for computer vision problems with the goal of learning backwards-compatible image representations. Yan et al. (2021) introduced the notion of negative flips for classification tasks and coined the minimization of them as positive-congruent training. They proposed focal knowledge distillation, a variant of traditional teacher-student distillation (Hinton et al., 2015), and model ensembling as techniques to achieve positive-congruent training. Zhao et al. (2022) continued this line of work by extending and combining the distillation and ensembling ideas into a single method called ELODI. With a slightly different focus, namely accepting or rejecting the predictions of a new model rather than training it, Träuble et al. (2021) proposed a Bayesian approach to reduce negative flips. In our work, we focus on Yan et al.'s (2021) focal distillation method as it is most applicable to our real-world use case where we cannot afford the use of model ensembles because of their computation, storage and latency overhead.

Xie et al. (2021) first applied the methods to NLP tasks. They found that negative flips are also prevalent during model updates for NLP tasks and demonstrated mitigations with distillation and ensembling methods in line with the earlier work. Concurrent to our work, Cai et al. (2022) extended positive-congruent training ideas to structured prediction tasks like parsing, which require extensions such as sequence distillation (Kim and Rush, 2016) or reranking. Also concurrent to our work, Schumann et al. (2023) introduced an importance-weighted interpolation method that they find to outperform focal distillation on intent classification benchmarks. We plan to incorporate their findings in our future work.

Continual learning (also known as incremental learning, sequential learning or lifelong learning) is closely related to our work (McCloskey and Cohen, 1989; Silver and Mercer, 2002; Biesialska et al., 2020). While we focus specifically on avoiding negative flips, continual learning is more general and studies continuous training of models on evolving data and tasks, with a particular focus on avoiding catastrophic forgetting. The latter is a challenge if data for previously learned features is no longer available; it is however less relevant for our application scenario, a real-world goal-oriented dialog system, with ongoing user interactions covering all features.

3 Methods

Application Scenario Spoken language understanding models are a core component in many goal-oriented dialog systems. They map a natural language request to a machine-readable meaning representation that the system can act upon to fulfill the request. In our experiment setup, this is modelled as a combination of domain classification (DC), intent classification (IC) and slot labeling (SL). Consider the example *Play Michael Jackson*. DC recognizes this request as a *Music* request, IC detects a *PlayMusic* intent and SL identifies *Michael Jackson* as *Artist* slot, whereas *Play* does not represent a slot in this case.

Negative Flips Let $x \in X$ be a model input (e.g. an utterance), $y \in Y$ its ground truth label (e.g. an intent label) and $p(y|x)$ a model that can be used to predict $\hat{y}_i = \text{argmax}_y\, p(y|x_i)$. A negative flip occurs if a new model incorrectly predicts a sample that the previous model predicted correctly, i.e. if $\hat{y}_i^{new} \neq y_i$ and $\hat{y}_i^{old} = y_i$. The *negative flip rate (NFR)* measures the fraction of samples where a correct prediction turns incorrect between two models in a dataset with size N. Yan et al. (2021) define it as

$$NFR = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(\hat{y}_i^{new} \neq y_i \wedge \hat{y}_i^{old} = y_i) \quad (1)$$

Focal Distillation (FD) Focal distillation, as introduced by Yan et al. (2021) and illustrated in Figure 1, aims to reduce negative flips by minimizing

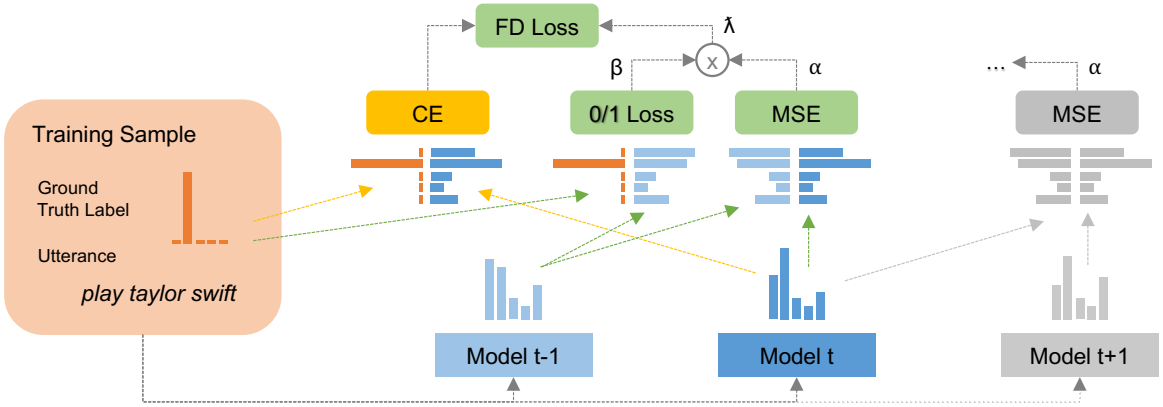

Figure 1: Illustration of focal distillation (FD): When training model t, a cross-entropy (CE) loss against ground truth labels is combined with a mean-squared error (MSE) loss against logits of model $t-1$, weighted by $t-1$'s sample-level accuracy to focus the distillation. This is applied iteratively, when training model $t+1$, model t acts as the reference model for distillation (indicated in gray).

the loss

$$\mathcal{L}_{CE}(\hat{y}^{new}, y) + \lambda \mathcal{L}_{FD}(p^{new}(y|x), p^{old}(y|x)) \tag{2}$$

where $\mathcal{L}_{CE}$ denotes the standard cross entropy (CE) loss between new model and ground truth and $\mathcal{L}_{FD}$ is the additional focal distillation (FD) loss term that discourages negative flips, with a trade-off parameter λ. This loss term is formally defined as

$$\begin{aligned} \mathcal{L}_{FD} &= -\mathcal{F}(x,y) \cdot \mathcal{D}(p^{new}, p^{old}) \\ \mathcal{F}(x,y) &= \alpha + \beta \cdot \mathbb{1}(\hat{y}^{old} = y), \end{aligned} \tag{3}$$

where $\mathcal{D}$ is a distance between the output distributions of the new and old model, and $\mathcal{F}(x,y)$ is a "filtering" function. It applies a weight α to all samples in the training set and an additional weight β to the samples correctly predicted by the old model. When $\alpha = 1$ and $\beta = 0$, focal distillation reduces to ordinary distillation. When $\alpha = 0$ and $\beta > 0$, we are only applying the distillation objective to the training samples predicted correctly by the old model.

In their work, Yan et al. (2021) experiment with two choices for $\mathcal{D}$: Kullback-Leibler (KL) divergence between temperature-scaled $p(y|x)$ and mean-squared error (MSE) between pre-softmax logits. Since the latter performed better in their experiments, we adopt it. Hence, $\mathcal{D}$ is defined as

$$\mathcal{D}(p^{new}, p^{old}) = \frac{1}{K} \sum_{j=1}^{K} (z_j^{new}(x) - z_j^{old}(x))^2 \tag{4}$$

where $z_j(x)$ is element j of the K-dimensional pre-softmax logit vector for x.

FD for Slot Labeling In slot labeling, a sequence of labels has to be predicted instead of just a single label as in DC or IC. Naturally, there are two options to apply distillation in that case: either apply the loss independently to each token or use the reference model's sequence-level decision for supervision. For the latter, Wang et al. (2020) proposed multiple techniques. In this work, we resort to the simpler token-level distillation for now and leave sequence-level distillation for future work.

We compute the FD loss for each token j in the sequence i with length M as

$$\mathcal{L}_{Tok}^{FD} = -\sum_{j=1}^{M} \mathcal{F}(x_j, y_j) \mathcal{D}\left(p_j^{new}, p_j^{old}\right) \tag{5}$$

Notice that this formulation works both if the models perform token-level decisions, and if they perform sequence-level decisions. In the latter case, when training the new model, the token-level FD loss is summed to the sequence-level loss.

FD with Noisy Labels FD biases the new model towards the old model's predictions when those predictions are correct. To discern correct predictions we rely on accurate labels. However, real-world data is often noisy. Therefore we investigate the combination of FD with label noise detection. We experiment with Area Under the Margin (AUM), a method suggested by Pleiss et al. (2020). The method leverages the observation that mislabeled data hurts generalization, and thus monitors the training dynamics to define the *margin* of a sample. The margin M at epoch t of sample (x, y) mea-

sures how much larger the assigned logit is than the largest other logit. Let $z^t(x) \in \mathbb{R}^c$ be the logit vector of sample (x, y) at epoch t. Then M at t is

$$M^{(t)}(x, y) = z_y^{(t)}(x) - max_{k \neq y}\, z_k^{(t)}(x) \quad (6)$$

where logit $z_k^{(t)}$ corresponds to class k. The first term corresponds to the assigned logit, while the second is the largest other logit. If a sample is mislabelled, the assigned logit tends to receive weaker gradient updates due to the tension between generalization from similar, correctly labeled samples and memorization of the sample itself. For instance, an utterance that is semantically similar to others labelled as *PlayMusic*, but is incorrectly labelled as *GetWeather*, results in the model predicting the true class with more confidence (higher logit) and assigning lower logit (confidence) to the incorrect label. As a consequence, a correctly labeled sample will have a larger margin than a mislabeled sample in expectation. Each sample's margin is measured during training and averaged over all epochs T:

$$AUM(x, y) = \frac{1}{T} \sum_{t=1}^{T} M^t(x, y) \quad (7)$$

We then use this measure as an additional term in the FD objective to re-weight the FD loss contributions of mislabeled samples:

$$\mathcal{L}_{AUM}^{FD} = - \underbrace{g(AUM(x, y))}_{\text{noise-aware weight}} \underbrace{\mathcal{F}(x, y)\mathcal{D}(p^{new}, p^{old})}_{\text{standard FD loss}} \quad (8)$$

where $g(\cdot)$ simply rescales AUM into $[0, 1]$.

4 Experimental Setup

We run experiments on both public and internal data. For our experiments on public data, we use SLURP (Bastianelli et al., 2020), an English multi-domain dataset for NLU spanning across 18 domains, 60 intents and 55 slot types (ca. 16,000 utterances). In addition, we present results on our internal datasets for English and German. These datasets comprise live traffic utterances, de-identified and anonymized for privacy reasons, then annotated to enable supervised training. For the internal datasets, the number of slot types is domain-specific. In the experiments for SL we employ three domain-specifc internal datasets, referred to as *INT-G*, *INT-M* and *INT-S*, that have 88, 101 and 35 slot types, respectively. Results on public data are averaged across 5 seeds, while we only train once on internal data. In our experiments we examine two settings: (i) A *data update* scenario, in which we only update the training data leaving the model architecture unchanged. In this scenario, 50% of the samples are left out when training the old model, while the complete dataset is used when training the new model, either with the baseline approach or FD. (ii) A *label introduction* scenario, in which we gradually introduce a new label in the dataset, training n models in sequence on datasets in which we uniformly increase the support for that label. For implementation details, we refer the reader to Appendix C. Across all experiments, we compare FD with the *baseline* approach of simply retraining the model on the whole training data, without any additional signal from the previous model. All models employ a BERT-based (Devlin et al., 2018) architecture: a pre-trained encoder extracts contextualized semantic word embeddings, then fed either to a Multi-Layer Perceptron (MLP) in case of DC and IC, or to a Conditional Random Field (CRF) (Lafferty et al., 2001; Lample et al., 2016) in case of SL, to obtain either sequence-level or word-level predictions. We experiment also with the introduction of *warm start* for the new model, i.e. the model's weights are initialized with those of the previous model. See Appendix A for more details.

Task	Method	Accuracy ↑ abs.	Accuracy ↑ rel.	NFR ↓ abs.	NFR ↓ rel.
DC	Baseline	$91.36_{\pm 0.35}$	–	$2.17_{\pm 0.16}$	–
	FD	$90.67_{\pm 0.59}$	-0.75	$1.57_{\pm 0.59}$	-27.64
IC	Baseline	$88.63_{\pm 0.45}$	–	$2.47_{\pm 0.41}$	–
	FD	$88.24_{\pm 0.51}$	-0.44	$1.63_{\pm 0.53}$	-33.79

Table 1: Test results for applying FD to DC and IC on SLURP under data update.

Task	Dataset	Accuracy ↑	NFR ↓
DC	English	+0.12	-54.94
	German	-0.07	-9.31
IC	English Cross-Domain	0.39	-35.59
	German INT-M	-0.02	-3.31

Table 2: Test results (rel. change to baseline) for applying FD to DC and IC on internal data under data update.

5 Experimental Results

We present results for each of the research questions raised in the introduction.

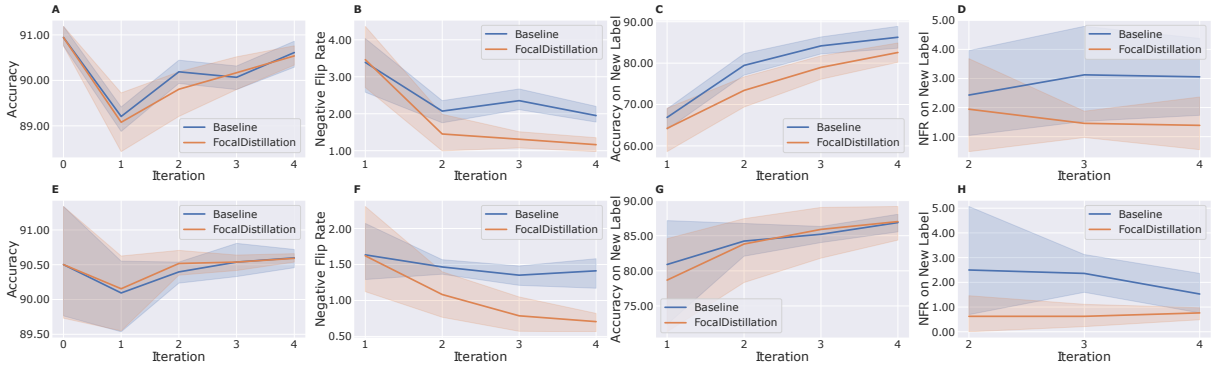

Figure 2: Metrics for repeated application of FD versus baseline for training DC models. Bottom row includes warm start. **A**, **E**: Overall accuracy. **B**, **F**: Overall NFR. **C**, **G**: Accuracy on the new label. **D**, **H**: NFR on the new label. Graphs for the new label, and those for NFR, skip the first iteration(s) as there is no previous model to compare to.

Task	Method	Accuracy ↑		NFR ↓	
		abs.	rel.	abs.	rel.
SL	Baseline	$92.9_{\pm 0.2}$	-	$1.6_{\pm 0.1}$	-
	FD	$92.4_{\pm 0.1}$	-0.46	$1.5_{\pm 0.1}$	-7.93

Table 3: Test results for applying FD to slot labeling on SLURP under data update.

Task	Dataset	Accuracy ↑	NFR ↓
SL	English INT-G	+0.09	-15.94
	English INT-M	-0.01	-3.66
	English INT-S	-0.01	-2.86

Table 4: Test results (rel. change to baseline) for applying FD to slot labeling on internal data.

Is FD effective for DC and IC? Results on public data are displayed in Table 1. FD reduces NFR for both DC and IC in similar magnitude, by 27.64% for DC and by 33.79% for IC, while only decreasing accuracy slightly by 0.75% for DC and 0.44% for IC. Results on internal data are shown in Table 2. On internal data, we can only disclose relative changes to baseline, no absolute metrics. For experiments on German data we use the full dataset and the production model. While we consider all domains for DC, we only consider intents within a single domain for IC since training is expensive and time-consuming. For English we only use 10% of the full training set and a surrogate model from Huggingface to speed up the experiments (see Appendix A). Also on internal data, FD reduces NFR for both DC and IC. Again accuracy is only slightly reduced, for DC on English data we even see a slight increase in accuracy. NFR reduces less significantly on German data, which can be explained by the fact that the training set for the old and the new model are the same and negative flips only stem from randomness in training. For the English dataset, we simulate an increase in training data, as explained in Section 4.

Can FD be used for token-level SL? As mentioned above, we employ models with a CRF layer for SL, able to make structured predictions. However, we experiment with token-level FD (see Equation (5)) that takes as input the token logits directly, instead of the top-scoring label path from the CRF. Therefore, the CRF layer of the student model is not affected by the additional distillation objective. Results on public data are shown in Table 3. With only a slight decrease in accuracy of -0.46%, NFR can be reduced by 7.93%. Results on internal data are reported in Table 4. Here we see an even larger reduction of 15.94% in NFR on the INT-G dataset, while a less significant reduction is observed on the other datasets: -3.66% on INT-M and -2.86% on INT-S. For all datasets, changes in accuracy are negligible. We conclude that FD on token-level SL reduces NFR without harming accuracy.

Does repeated FD restrict learning? Figure 2 reports the comparison of the baseline approach with FD in the label introduction scenario (spanning 5 iterations). We can observe how FD does not seem to negatively influence the overall accuracy over time of the model; on the contrary, the additional loss term seems to be moderately beneficial in helping the model learn the task compared to the baseline. The approach also behaves well in reducing the overall NFR of the new model. Interestingly, standard CE is slightly superior in absolute terms with respect to FD in learning the new label distribution. However, the gap remains fixed over time, therefore FD is not hindering the ability of

Task	Approach	Original		20% Noise		40% Noise		60% Noise	
		Accuracy ↑	NFR ↓	Accuracy ↑	NFR ↓	Accuracy ↑	NFR ↓	Accuracy ↑	NFR ↓
DC	Baseline	–	–	–	–	–	–	–	–
	FD	**0.34%**	**-53.02%**	**0.46%**	-43.07%	**0.10%**	**-32.84%**	**0.75%**	**-47.81%**
	FD+AUM	0.11%	-24.01%	**0.46%**	**-43.46%**	-0.21%	6.13%	0.72%	-31.53%
IC	Baseline	–	–	–	–	–	–	–	–
	FD	0.09%	-30.05%	-0.04%	30.56%	**0.05%**	**-37.49%**	**0.12%**	**-35.68%**
	FD+AUM	**0.20%**	**-34.33%**	**0.09%**	**-8.37%**.	0.02%	-18%	**0.12%**	-32.88%

Table 5: Test results (rel. change to baseline) for applying FD with AUM both on original internal dataset and on internal dataset with artificially added noise.

Task	Approach	Original		20% Noise		40% Noise		60% Noise	
		Accuracy ↑	NFR ↓	Accuracy ↑	NFR ↓	Accuracy ↑	NFR ↓	Accuracy ↑	NFR ↓
DC	Baseline	0.90742	2.2416	0.7215	2.5331	0.505	5.9067	0.3692	3.5194
	FD	-0.06%	**-52.50%**	-0.08%	-21.24%	**6.91%**	**-77.42%**	**3.79%**	**-63.38%**
	FD+AUM	**0.03%**	**-52.50%**	**0.29%**	**-29.65%**	6.24%	-63.19%	2.65%	-42.36%
IC	Baseline	**0.8807**	2.5106	**0.6518**	**4.2319**	0.4913	4.1913	**0.2887**	**4.1034**
	FD	-0.01%	**-25.89%**	-6.15%	4%	-3.26%	22.97%	-3.05%	31.91%
	FD+AUM	-0.28%	-12.59%	-1.58%	2.06%	**0.98%**	**-14.15%**	-6.44%	12.48%

Table 6: Test results (rel. change to baseline) for applying FD with AUM both on SLURP original dataset and on SLURP with artificially added noise.

the model to learn, but only introducing an initial delay. Remarkably, FD is able to reduce regression on the newly introduced label already with a handful of samples, and consistently remains lower than the baseline on the NFR metric. Interestingly but not surprisingly, warm start helps both approaches in both metrics, with respect to the non-warm start alternative. This suggests that, in general, warm start is a useful strategy for retaining model performance during an update. However, it is clear from the results how FD benefits more from warm-start than the baseline, in terms of both accuracy improvement and NFR reduction. Further results for this setting (and the specular one of gradual removal of a label) are reported in Appendix E.

Can FD cope with noisy labels? In order to verify the extent to which FD coupled with AUM is capable of dealing with increasing level of noise we experiment both on the public SLURP dataset as well as on the internal English dataset, and we also test the approach on specific versions of those datasets manipulated to artificially introduce varying levels of noise: 20%, 40%, 60%. The algorithm used to generate noise, together with a study of how AUM is able to detect it, is reported in Appendix D. Results on the internal dataset and SLURP are reported in tables 5 and 6, respectively. Overall we observe that integrating AUM into FD does not lead to significant improvement over vanilla FD. We believe the reason behind the lack of improvement is twofold: first, there might be a more effective way to integrate the AUM signal into the FD objective; secondly, the models trained with the baseline, especially on internal data, already exhibit low NFR, therefore there is little margin for improvement. On the other hand, FD with AUM is not detrimental, neither on original nor the noisy datasets: when the level of label noise is significant, AUM helps FD recovering its performance; when the label noise if less present (if at all), AUM does not significantly decrease FD performance.

6 Conclusions

In this paper, we presented an extensive set of experiments to evaluate the effectiveness of focal distillation to reduce negative flips in a real-world goal-oriented dialog system. We found the technique to be effective in DC, IC and SL with only minor accuracy drops. When used repeatedly over multiple updates, the effect remains while still allowing the model to learn new labels. In addition, the method is also robust to labeling errors in the training data. As future work, we plan to extend our experiments to alternative techniques for negative flip reduction, in particular those proposed concurrent to our work, and to experiment with potentially more powerful sequence-level distillation for slot labeling.

Limitations

A first limitation of our contribution stems from the fact that to compute the focal distillation term in the loss, predictions from the old model are required. This additional stream of information will therefore cause a slight increase in the required computational power.

In this work, we only experimented with FD based on mean-squared error between pre-softmax logits as that approach yielded best results in the paper our experiments are based on, leaving experiments using FD with Kullback-Leibler divergence between temperature-scaled softmax outputs for future research. Due to inference time limitations in a production setting, we did not investigate the reduction of negative flips with ensembles either.

Finally, we have not tested more principled approaches for NER distillation and focused on token-level distillation leaving sequence-level distillation for future work.

References

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. SLURP: A spoken language understanding resource package. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7252–7262, Online. Association for Computational Linguistics.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Deng Cai, Elman Mansimov, Yi-An Lai, Yixuan Su, Lei Shu, and Yi Zhang. 2022. Measuring and reducing model update regression in structured prediction for nlp. In *NeurIPS 2022*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

WA Falcon. 2019. Pytorch lightning.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056.

Raphael Schumann, Elman Mansimov, Yi-An Lai, Nikolaos Pappas, Xibin Gao, and Yi Zhang. 2023. Backward compatibility during data updates by weight interpolation.

Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. 2020. Towards backward-compatible representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6368–6377.

Daniel L. Silver and Robert E. Mercer. 2002. The task rehearsal method of life-long learning: Overcoming impoverished data. In *Advances in Artificial Intelligence*, pages 90–101, Berlin, Heidelberg. Springer Berlin Heidelberg.

F. Träuble, J. von Kügelgen, M. Kleindessner, F. Locatello, B. Schölkopf, and P. Gehler. 2021. Backward-compatible prediction updates: A probabilistic approach. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pages 116–128.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Fei Huang, and Kewei Tu. 2020. Structure-level knowledge distillation for multilingual sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3317–3330, Online. Association for Computational Linguistics.

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

Yuqing Xie, Yi-An Lai, Yuanjun Xiong, Yi Zhang, and Stefano Soatto. 2021. Regression bugs are in your model! measuring, reducing and analyzing regressions in NLP model updates. 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)*, pages 6589–6602, Online. Association for Computational Linguistics.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. 2021. Positive-congruent training: Towards regression-free model updates. In *Proceedings of the 2021 Conference on Computer Vision and Pattern Recognition*, pages 14299–14308.

Yue Zhao, Yantao Shen, Yuanjun Xiong, Shuo Yang, Wei Xia, Zhuowen Tu, Bernt Schiele, and Stefano Soatto. 2022. Elodi: Ensemble logit difference inhibition for positive-congruent training.

A Experiment Details

The experiments have been run on `p3.16xlarge` EC2 instances[1], equipped with eight NVIDIA Tesla V100 GPUs[2]. As optimization framework, PyTorch (Paszke et al., 2019) (version 1.10.0) has been used, along with PyTorch Lightning (Falcon, 2019) (version 1.8.6) for easier development and faster experimental iterations.

Across all the experiments on English corpora, the encoder is based on pre-trained BERT (Devlin et al., 2018) models from HuggingFace (Wolf et al., 2019), with their weights unfrozen during training, hence allowing their fine-tuning. For DC and IC experiments on German corpora, a custom pre-trained `tiny-bert` (Jiao et al., 2020) is used. All models feature a two-layer, fully-connected MLP mapping word or sentence embeddings into label-space. Additionally, the model for SL employs a CRF layer to make structured predictions about the label sequence, taking estimated label-label transition probabilities into account. For the SL experiments, we obtain subword token-level embeddings on the English corpora by summing the hidden states of the last 3 layers of the encoder. On the German corpora, the model considers the last hidden states. In the latter, word-level embeddings (aligned with slot labels) are obtained performing an average subword pooling, i.e. for each input text token we take the average embedding of all its corresponding subword tokens. In the former, the last subword token embedding is considered.

For models trained with Focal Distillation, we follow the authors' suggestion in Yan et al. (2021) and set $\alpha = 1, \beta = 5$ and $\lambda = 1$ for all experiments. Table 7 reports the hyperparameters used to train the models across all the experiments.

All models are trained to convergence using early stopping, monitoring model performance on a held-out validation set as convergence condition.

Parameter	Value
Learning Rate	5e-5
Optimizer	Adam
Max epochs	20
Embedding size	768
Hidden size	256
Dropout	0.1
Activation	ReLU
Validation split	0.1
Early stopping metric	Validation F1 score
Early stopping delta	1e-3
Early stopping patience	5 epochs
Focal Distillation α	1
Focal Distillation β	5
FD trade-off λ	1

Table 7: Hyperparameter values for models used in the experiments.

B Data Update Scenario

Over time, the data available to train a predictive model can change for various reasons. In a supervised learning setting, one simple reason may be the acquisition of more labeled data: human annotators review existing unlabeled instances and assign labels to them, enlarging the training corpus. In this work, we refer to this event as a *data update*, and study the impact of applying FD in this scenario.

The results presented in section 5 examine in particular a scenario in which the amount of available training data is doubled for the new model. This is realized by simply training the old model on 50% of the overall training dataset, then training the new model (with either the baseline or FD) on 100% of the training samples.

C New Label Scenario

Another possible reason for a change in the training data is the addition of data supporting new classes. New classes appearing in the training dataset of an already deployed model may be the result of the definition of a new downstream feature that the model has to support. In this work, we refer to this event as a *label introduction*, and study the impact of applying FD in this scenario.

Usually, data supporting a new feature is not readily available, but rather comes in batches as human annotators work to provide new labeled data based on the feature definition. For this reason, in this work we study the impact of FD on a *gradual* introduction of a new label. In particular, the scenario is implemented as follows: (i) a label is chosen for scenario simulation and completely removed from the dataset, i.e. all the samples belonging to that class are removed; (ii) a *schedule* for introducing the label in the following n "re-

[1] https://aws.amazon.com/ec2/instance-types/p3/

[2] https://www.nvidia.com/en-us/data-center/v100/

leases" of the dataset is stated. For simplicity, we assume the rate at which newly labelled data becomes available is constant over time, and therefore the schedule simply dictates that a fixed amount of labelled data is reintroduced at each iteration. To do so, data pertaining to the removed labels is evenly partitioned in n batches, and the i-th dataset is simply the union of the previous dataset in the sequence and the i-th batch.

In this work we set the number of releases to $n = 5$. We run experiments on the SLURP dataset for Domain Classification using the qa domain and for Intent Classification using the news_query intent. This choice reflects two competing needs: on one hand, we want to reflect the observed reality of new features not becoming the predominant classes in the dataset in terms of data, even after a long time; on the other hand, to report statistically significant results we need more than a handful of samples to be removed. As a result, we choose labels that are neither the prevalent classes nor the scarcest, but are averagely represented.

When a label is introduced for the first time, we set the MSE loss in FD to zero for samples with the new label as the previous model cannot provide useful information for those. That means we zero out the logits for the new label coming from the old model by concatenating a zero tensor to the logits coming from the old model. As a result, the contribution to the MSE loss in FD is zero for the new label, falling back to only Cross Entropy loss for samples with the new label.

D Area Under the Margin and Noise Generation Procedure

Pleiss et al. (2020) introduce the concept of Area Under the Margin (AUM), and demonstrate its ability to identify mislabelled samples in synthetically-mislabeled versions of popular Computer Vision datasets, such as CIFAR10. Their approach makes no assumption about the specific task under consideration, but only draws on the insight that a neural network's training dynamics contain salient signals about noisy data and generalization. In this work, however, before testing the interaction of AUM with FD in a noisy data setting we test the ability of AUM of spotting noise in our Natural Language Understanding (NLU) setting to begin with. To do so, we repeat the synthetically-mislabeled experiment on our datasets.

Algorithm 1 Label noise generation

Input: true labels Y, noise level $n_l \in [0, 1]$
Output: assigned labels $\tilde{Y}$

$N \leftarrow \mid Y \mid$
$L \leftarrow \{y \mid y \in Y\}$
$N_{flip} \leftarrow \lceil n_l \cdot N \rceil$
for $i = 0 \rightarrow N_{flip}$ **do**
    $y_i \leftarrow$ sample an item uniformly at random from Y without replacement
    $\tilde{L} \leftarrow \{y \mid y \in L \wedge y \neq y_i\}$
    $\tilde{y}_i \leftarrow$ sample a label uniformly at random from $\tilde{L}$
    change y_i into $\tilde{y}_i$ in Y
end for

Table 8 reports the noise levels estimated in the synthetically-mislabeled datasets. Pleiss et al. (2020) introduce *threshold samples*, purposefully mislabeled samples belonging to an extra class, to identify a AUM upper bound that isolates mislabeled data (see algorithm 2). In particular, they establish that the 99^{th} percentile of threshold AUM values separates correctly- and mislabeled data. Notice that this mechanism would introduce additional complexity for coupling the AUM approach with FD, since we do not wish for the extra class to be present in the output distribution of the new model trained with FD. Therefore, beside testing vanilla AUM in the NLU setting, we test whether simply observing the *sign* of the AUM values is a satisfying proxy metric of the true AUM metric. Synthetic noise is injected using algorithm 2 for the former, and algorithm 1 for the latter.

We can see how standard AUM is able to estimate the noise level quite accurately, with an average (absolute) estimation error of 1.71%. The simpler variant is less competitive in estimating noise levels, reporting an average estimation error of 3.70%. Interestingly, the variant consistently overestimates noise levels for the SLURP dataset in the IC setting, exhibiting a sensitivity to the ratio between label space dimension and dataset size. Indeed, moving from the DC task to the IC task, the number of samples remains constant but the label space nearly triples in dimension. While the same holds roughly true also for the INT-G dataset, its size is considerably larger than SLURP. We hypothesize this is due to the approach having to rely on fewer samples to observe training dynamics, leading to a less informative metric computation.

Algorithm 2 Label noise generation when using threshold samples

Input: true labels Y, noise level $n_l \in [0, 1]$
Output: assigned labels $\tilde{Y}$, threshold samples $\bar{I}$

$N \leftarrow \mid Y \mid$
$L \leftarrow \{y \mid y \in Y\}$
$N_{flip} \leftarrow \lceil n_l \cdot N \rceil$
$N_{threshold} \leftarrow \lceil N \cdot (\mid L \mid + 1) \rceil$
$\bar{y} \leftarrow (\mid L \mid + 1)$ ▷ new class for threshold samples
$\bar{I} \leftarrow \{\ \}$
for $i = 0 \rightarrow N_{flip}$ **do**
    $y_i \leftarrow$ sample an item uniformly at random from Y without replacement
    $\tilde{L} \leftarrow \{y \mid y \in L \wedge y \neq y_i\}$
    $\tilde{y}_i \leftarrow$ sample a label uniformly at random from $\tilde{L}$
    change y_i into $\tilde{y}_i$ in Y
end for
for $i = 0 \rightarrow N_{threshold}$ **do**
    $y_i \leftarrow$ sample an item uniformly at random from Y without replacement
    change y_i into $\bar{y}_i$ in Y
    $\bar{I} \leftarrow \bar{I} \cup \{i\}$
end for

E Additional Results

Figure 3 reports results for repeated application of FD to the IC task, in contrast with the DC task reported in fig. 2.

Figures 4 and 5 present instead results for a specular setting to the "label introduction" one, in which we gradually remove data supporting a label.

In figs. 6 to 9 we investigate the influence of model size on repeatedly applying FD. In particular, a `tiny-bert` encoder is used. Results suggest that FD becomes detrimental when the older model does not have sufficient "capacity" to accurately provide a distillation signal for the new model

Task	Dataset	20% Noise		40% Noise		60% Noise	
		AUM $< 99p$	**AUM** < 0	**AUM** $< 99p$	**AUM** < 0	**AUM** $< 99p$	**AUM** < 0
DC	SLURP	23.80 %	26.42 %	39.53 %	42.14 %	58.36 %	61.89 %
	English INT-G	23.32 %	25.02 %	40.25 %	40.59 %	57.24 %	58.60 %
IC	SLURP	21.84 %	32.08 %	36.87 %	47.56 %	57.22 %	67.32 %
	English INT-G	21.27 %	21.81 %	40.10 %	39.77 %	60.17 %	61.23 %

Table 8: Datasets noise estimation on synthetically-mislabelled dataset. In the first column, we consider standard AUM, in which we estimate that a sample is noisy when its AUM value is lower than than the 99-th percentile of the AUM values of the threshold samples. In the second column we consider the simpler variant, in which we estimate that a sample is noisy when its AUM value is negative.

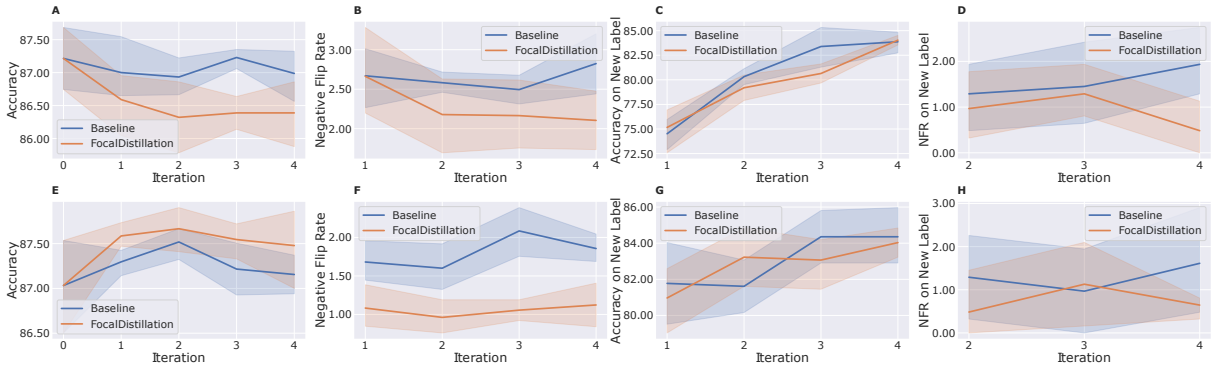

Figure 3: Metrics for repeated application of FD versus baseline for training IC models. Bottom row includes warm start. **A**, **E**: Overall accuracy. **B**, **F**: Overall NFR. **C**, **G**: Accuracy on the new label. **D**, **H**: NFR on the new label. Graphs for the new label, and those for NFR, skip the first iteration(s) as there is no previous model to compare to.

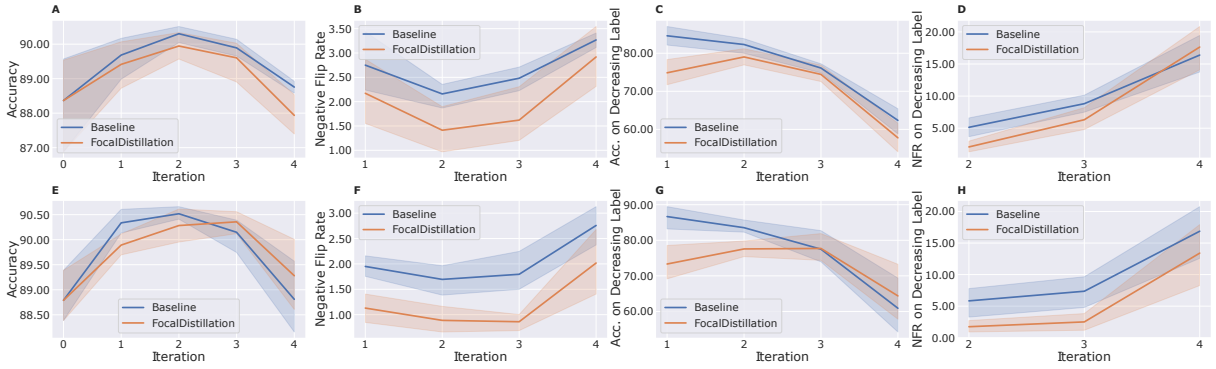

Figure 4: Metrics for repeated application of FD versus baseline for training DC models when gradually removing data for a label. Bottom row includes warm start. **A**, **E**: Overall accuracy. **B**, **F**: Overall NFR. **C**, **G**: Accuracy on the label being removed. **D**, **H**: NFR on the label being removed. Graphs for the new label, and those for NFR, skip the first iteration(s) as there is no previous model to compare to.

Task	Approach	Original		20% Noise		40% Noise		60% Noise	
		Accuracy ↑	**NFR** ↓	**Accuracy** ↑	**NFR** ↓	**Accuracy** ↑	**NFR** ↓	**Accuracy** ↑	**NFR** ↓
DC	Baseline	0.9074	2.2416	0.7215	2.5331	0.505	5.9067	0.3692	3.5194
	FD	0.9069	**1.0647**	0.7209	1.9951	**0.5399**	**1.3338**	**0.3832**	**1.2889**
	FD+AUM	**0.9077**	**1.0647**	0.7236	**1.7821**	0.5365	2.1744	0.379	2.0287
IC	Baseline	**0.8807**	2.5106	**0.6518**	**4.2319**	0.4913	4.1913	**0.2887**	**4.1034**
	FD	0.8806	**1.8605**	0.6117	4.4013	0.4753	5.154	0.2799	5.4127
	FD+AUM	0.8782	2.1968	0.6415	4.3192	**0.4961**	**3.5982**	0.2701	4.6157

Table 9: Absolute results for the experiment on applying FD with AUM on the noisy labels setting on SLURP.

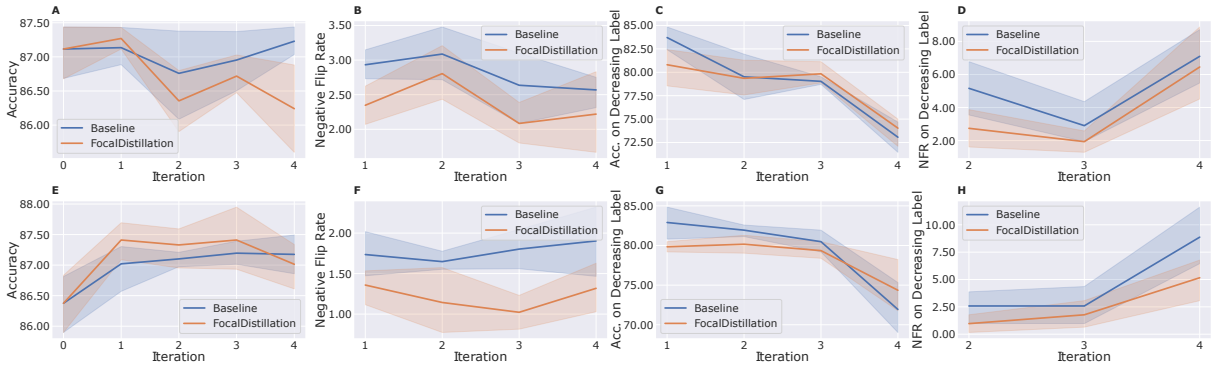

Figure 5: Metrics for repeated application of FD versus baseline for training IC models when gradually removing data for a label. Bottom row includes warm start. **A**, **E**: Overall accuracy. **B**, **F**: Overall NFR. **C**, **G**: Accuracy on the new label. **D**, **H**: NFR on the new label. Graphs for the new label, and those for NFR, skip the first iteration(s) as there is no previous model to compare to.

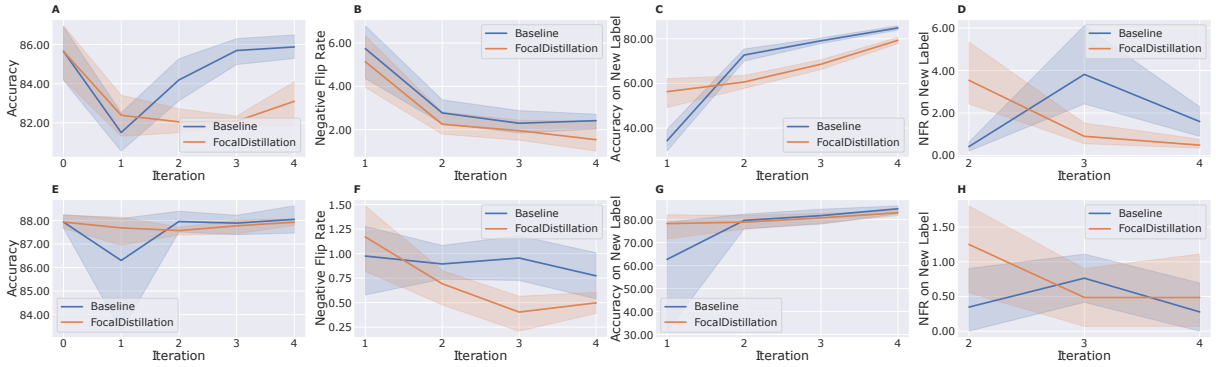

Figure 6: Metrics for repeated application of FD versus baseline for training DC models when using a smaller BERT model. Bottom row includes warm start. **A**, **E**: Overall accuracy. **B**, **F**: Overall NFR. **C**, **G**: Accuracy on the new label. **D**, **H**: NFR on the new label. Graphs for the new label, and those for NFR, skip the first iteration(s) as there is no previous model to compare to.

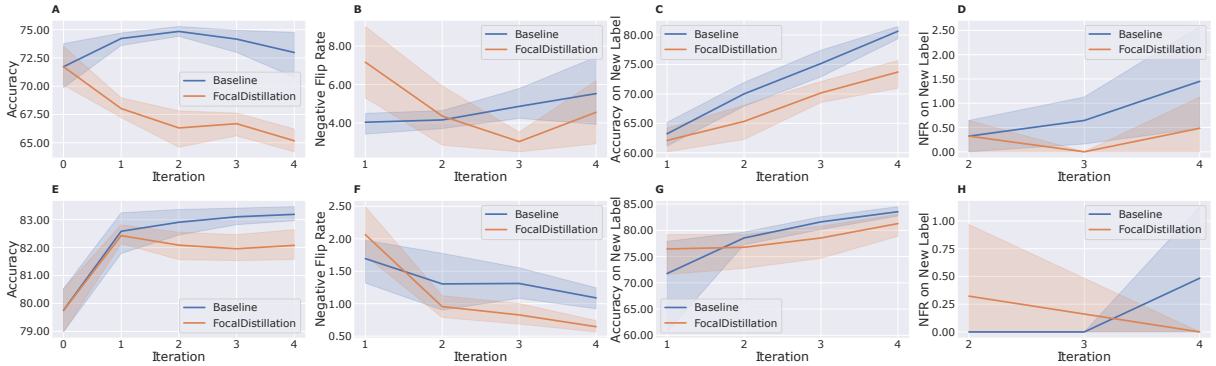

Figure 7: Metrics for repeated application of FD versus baseline for training IC models when using a smaller BERT model. Bottom row includes warm start. **A**, **E**: Overall accuracy. **B**, **F**: Overall NFR. **C**, **G**: Accuracy on the new label. **D**, **H**: NFR on the new label. Graphs for the new label, and those for NFR, skip the first iteration(s) as there is no previous model to compare to.

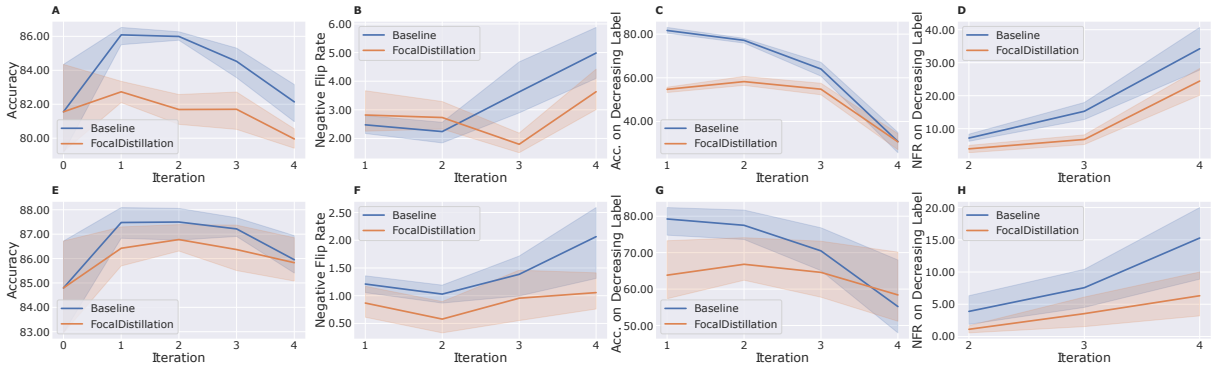

Figure 8: Metrics for repeated application of FD versus baseline for training DC models when using a smaller BERT model and gradually removing a label. Bottom row includes warm start. **A**, **E**: Overall accuracy. **B**, **F**: Overall NFR. **C**, **G**: Accuracy on the label being removed. **D**, **H**: NFR on the label being removed. Graphs for the new label, and those for NFR, skip the first iteration(s) as there is no previous model to compare to.

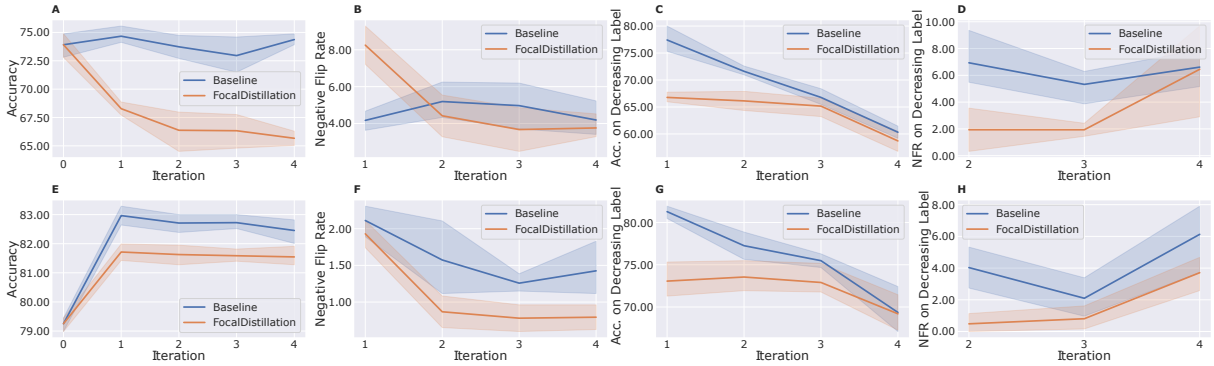

Figure 9: Metrics for repeated application of FD versus baseline for training IC models when using a smaller BERT model and gradually removing a label. Bottom row includes warm start. **A**, **E**: Overall accuracy. **B**, **F**: Overall NFR. **C**, **G**: Accuracy on the new label. **D**, **H**: NFR on the new label. Graphs for the new label, and those for NFR, skip the first iteration(s) as there is no previous model to compare to.