Improving Universal Access to Modern Speech Technology

Stanford | NLP

Martijn Bartelds

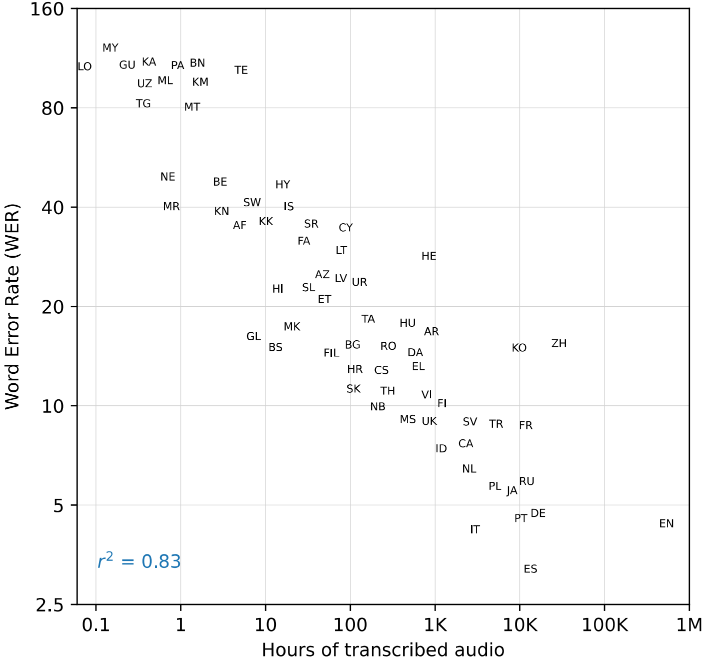
Stanford NLP Group

bartelds@stanford.edu

Increasingly powerful speech models promise
“universal” speech processing

Robust Speech Recognition via Large-Scale Weak Supervision

Alec Radford *[1] Jong Wook Kim *[1] Tao Xu [1] Greg Brockman [1] Christine McLeavey [1] Ilya Sutskever [1]



"Whisper's speech recognition performance is still quite poor on many languages."
(Radford et al. 2023)

Scaling Speech Technology to 1,000+ Languages

Vineel Pratap* **Andros Tjandra*** **Bowen Shi*** **Paden Tomasello**
Arun Babu **Sayani Kundu**[†] **Ali Elkahky**[‡] **Zhaoheng Ni**
Apoorv Vyas **Maryam Fazel-Zarandi** **Alexei Baevski** **Yossi Adi**
Xiaohui Zhang **Wei-Ning Hsu** **Alexis Conneau**[§] **Michael Auli***

	Whisper medium	Whisper large-v2	MMS L-61 noLM	MMS L-61 CC LM	MMS L-61 noLM LSAH	MMS L-61 CC LM LSAH	MMS L-1107 noLM	MMS L-1107 CC LM	MMS L-1107 noLM LSAH	MMS L-1107 CC LM LSAH
Amharic	229.3	140.3	48.7	30.7	52.4	32.5	52.9	30.1	53.3	31.1
Arabic	20.4	16.0	34.9	19.6	35.8	19.9	44.0	23.4	41.3	21.0
Assamese	102.3	106.2	29.5	18.8	28.4	18.6	37.6	21.2	30.5	19.2
Azerbaijani	33.1	23.4	40.7	21.3	38.3	19.8	45.0	21.2	40.1	19.1
Bengali	100.6	104.1	19.7	11.6	20.0	12.1	25.0	12.5	23.5	12.1
Bulgarian	21.4	14.6	23.4	13.1	23.9	13.3	27.9	12.9	25.5	13.5
Burmese	123.0	115.7	22.2	14.2	22.3	14.5	29.2	20.2	24.5	16.0
Catalan	9.6	7.3	18.1	11.0	18.1	11.0	25.9	11.5	20.1	10.8
Dutch	9.9	6.7	26.9	13.7	26.4	14.3	38.1	14.9	27.6	14.5

Addressing this challenge could improve the digital participation of many speakers worldwide

What do we need?

Better ways to **reliably measure** speech recognition model performance

What do we need?

- Better ways to **reliably measure** speech recognition model performance
- **New algorithms** for bridging the performance gap between languages

Evaluating

Interspeech 2024
1-5 September 2024, Kos, Greece

ML-SUPERB 2.0: Benchmarking Multilingual Speech Models Across Modeling Constraints, Languages, and Datasets

Jiatong Shi[1], *Shih-Heng Wang*[2,*], *William Chen*[1,*], *Martijn Bartelds*[3,*], *Vanya Bannihatti Kumar*[1], *Jinchuan Tian*[1], *Xuankai Chang*[1], *Dan Jurafsky*[3], *Karen Livescu*[3,4], *Hung-yi Lee*[2], *Shinji Watanabe*[1]

[1] Carnegie Mellon University, [2] National Taiwan University, [3] Stanford University,
[4] Toyota Technological Institute at Chicago

Background: Multilingual Speech Processing Benchmark

- Recent multilingual speech processing models
  - Have the capacity to model **hundreds of languages**

Background: Multilingual Speech Processing Benchmark

- Recent multilingual speech processing models
  - Have the capacity to model **hundreds of languages**
  - However, they are often evaluated using different setups, which **limits the extent to which they can be reliably compared**

Background: Multilingual Speech Processing Benchmark

- Recent multilingual speech processing models
  - Have the capacity to model **hundreds of languages**
  - However, they are often evaluated using different setups, which **limits the extent to which they can be reliably compared**

→This motivates the need for **multilingual speech processing benchmarks**

Background: Multilingual Speech Processing Benchmark

We observe great efforts in the community on spoken multilingual benchmarks:

- XTREME-S (Conneau et al. 2022)
- IndicSUPERB (Javed et al. 2023)
- ML-SUPERB (Shi et al. 2023)

Background: Multilingual Speech Processing Benchmark

- We observe great efforts in the community on spoken multilingual benchmarks:
  - XTREME-S (Conneau et al. 2022)
  - IndicSUPERB (Javed et al. 2023)
  - ML-SUPERB (Shi et al. 2023)

- ML-SUPERB is the most comprehensive benchmark in terms of language coverage, as it includes **143** languages and it evaluates models on:
  - Monolingual/multilingual automatic speech recognition (ASR)
  - Language identification (LID)
  - Joint ASR + LID

SUPERB

Limitations of ML-SUPERB

- Strictly constrained benchmark settings with self-supervised learning (SSL) pre-trained models
  - Efficient yet not generalizable enough to various settings (Zaiem et al. 2023; Arora et al. 2024)
  - Does not take application requirements or users’ budgets into account
- This motivates benchmarking with more **flexible constraints**

Limitations of ML-SUPERB

- Evaluation metric does not provide insight into performance variations between individual languages and datasets
- This motivates changes to the **evaluation metrics** to place greater focus on **robustness** across languages and datasets

Introduction of ML-SUPERB 2.0

- We revisit ML-SUPERB:
  - By **relaxing its fixed constraints**
  - By **improving fairness in its evaluation metrics** to focus on **robustness** across languages and **variation** across datasets

Experimental Design (General Setup)

- ML-SUPERB 2.0 evaluates joint multilingual LID/ASR
- We updated the ML-SUPERB dataset by **correcting** some mistakes*
- Some statistics:
  - 141 languages across 15 datasets
  - Around 300 hours in total (with 85 hours for validation + test sets)
  - We follow the 1-hour configuration presented in ML-SUPERB
  - 20 languages are reserved for few-shot learning experiments, each using 5 utterances for training

* Please refer to our paper for details about the updates to the dataset

Experimental Design (General Setup)

- Experimental codebases:
  - ESPnet (Watanabe et al. 2018)
  - S3PRL (Yang et al. 2021)
- Selected pre-trained self-supervised models:
  - XLS-R (Babu et al. 2022)
  - MMS (Pratap et al. 2024)
- In line with the original ML-SUPERB:
  - Limit the number of tunable parameters to 100 million

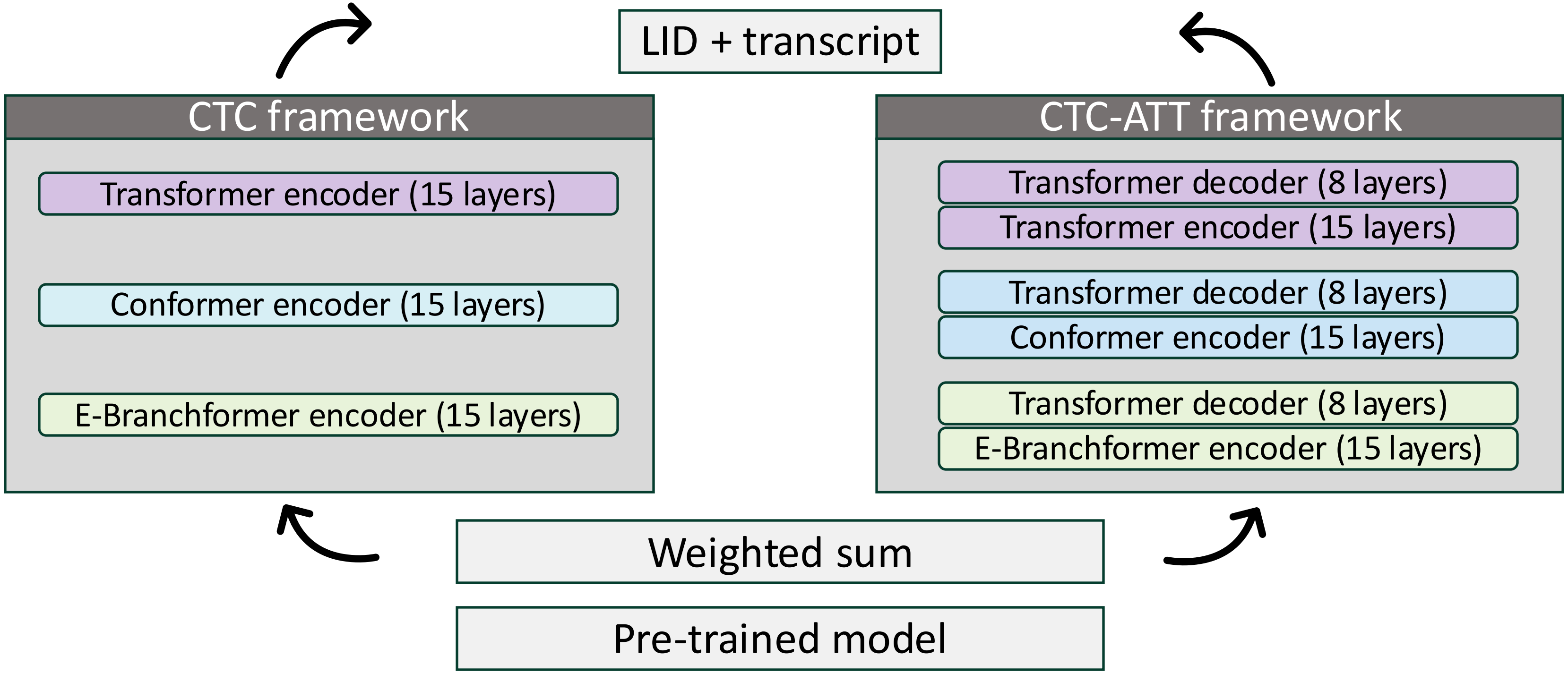
ESPnet

S3PRL
SPEECH TOOLKIT

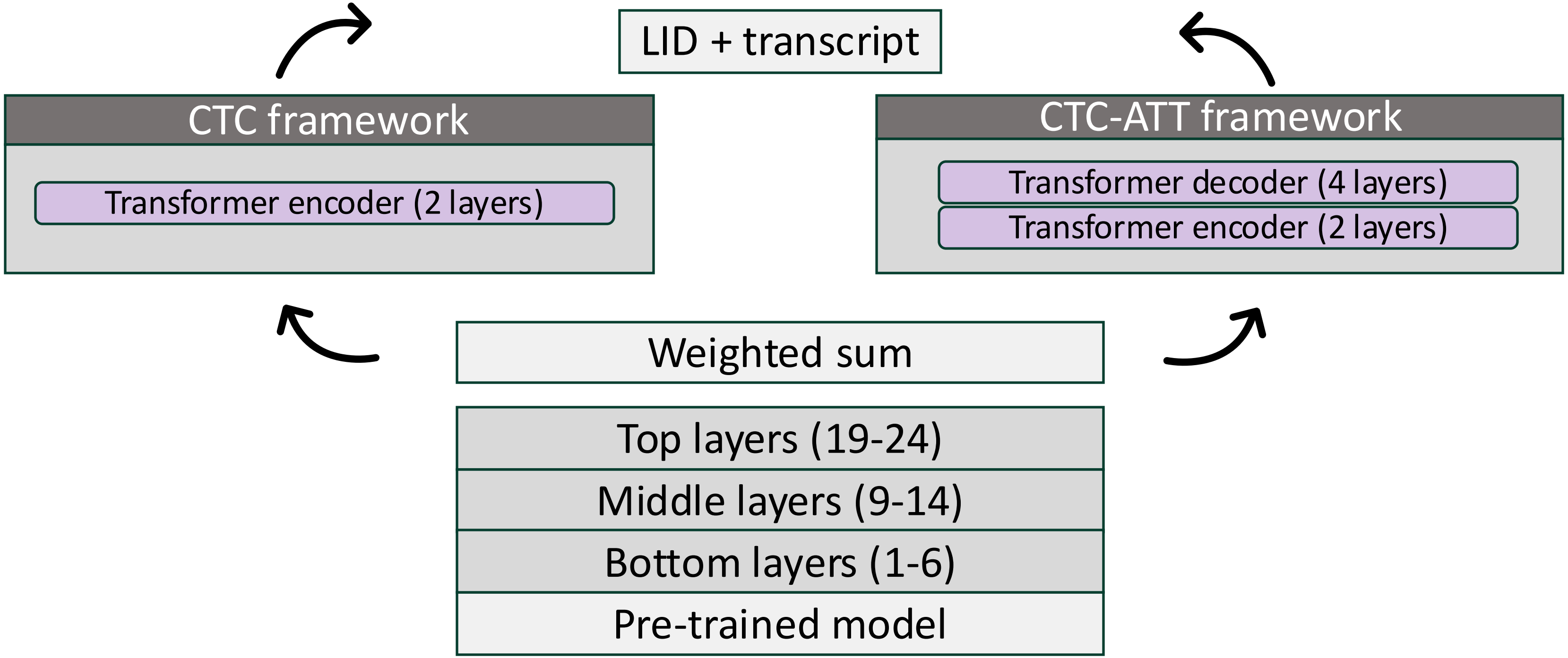
Experimental Design (General Setup)

- Specifically, we investigate **four new benchmark configurations**:
  - Larger-scale downstream models
  - SSL model fine-tuning
  - Efficient model adaptation strategies
  - Supervised pre-trained models

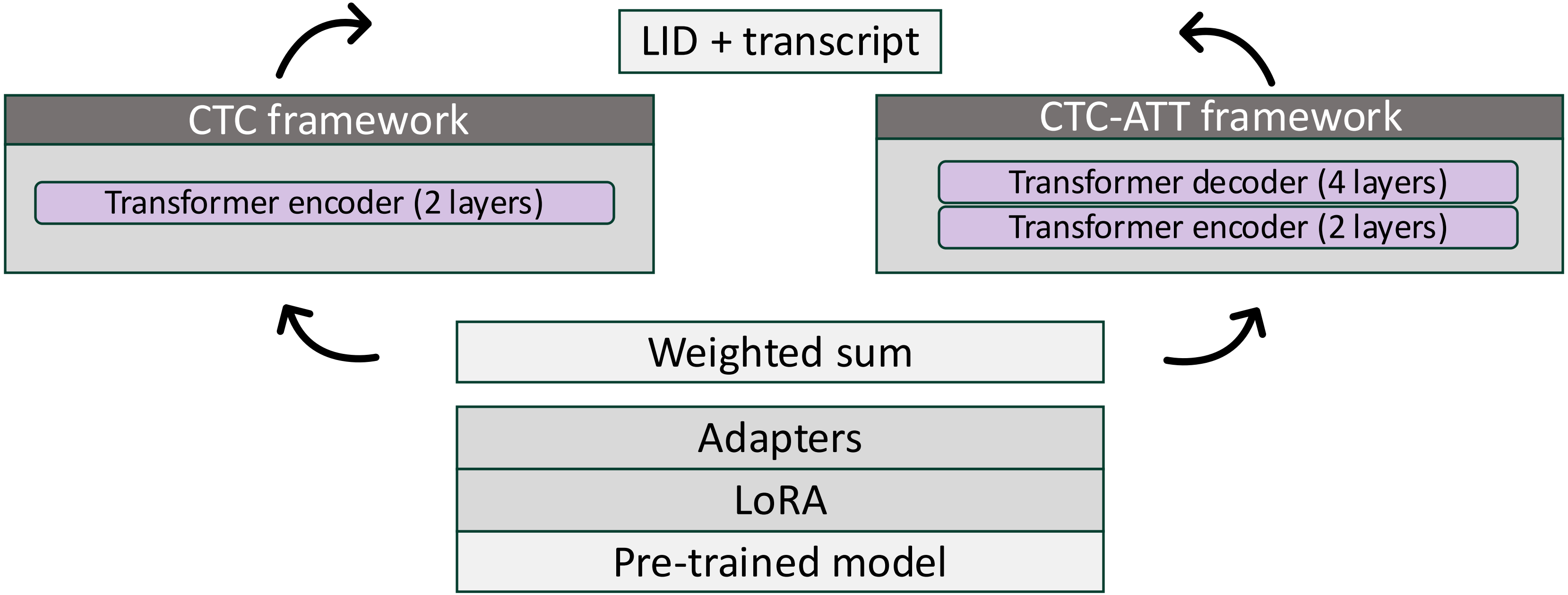
Larger-scale downstream models



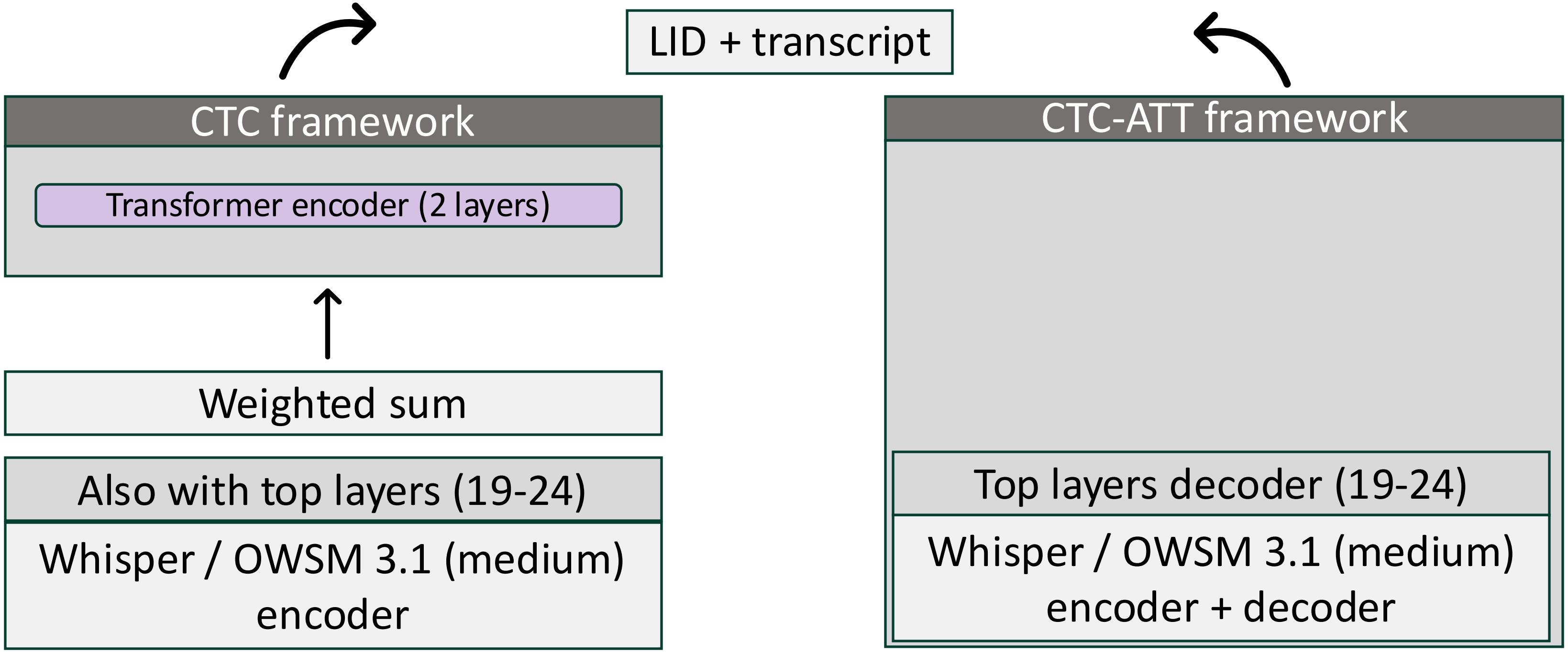
SSL model fine-tuning



Efficient model adaptation strategies



Supervised pre-trained models



Experimental Design (Configuration Setup)

- For the four benchmark configurations:
  - Hyperparameters follow prior works*
  - We tune the learning rate and select the best-performing model on the validation set

* Please refer to our paper for the complete list of prior works we refer to.

Experimental Design (Evaluation)

- Base metrics:
  - Accuracy for LID
  - Character error rate (CER) for ASR on two sets (normal and few-shot setting)

Experimental Design (Evaluation)

- **Place greater focus on measuring robustness:**
  - Macro-average over languages/datasets instead of micro-average CER
    - Compute per-language CER as the macro-average of CERs across all datasets per language
    - Compute the macro-average of the per-language CERs

    → Allows to better understand variation between languages and datasets

    → Languages with more samples do not disproportionally affect the CER

  - Standard deviation of language-specific CERs
  - Measure CER of the worst-performing language
  - Measure CER range between datasets in the same language

Experimental Results and Discussions

- Effect of introducing four benchmark configurations
- Model ranking for the benchmark configurations
- Supervised ASR versus SSL pre-trained models
- Variation across languages and datasets

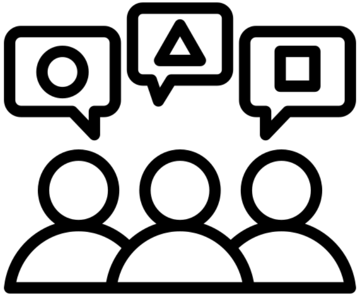
Due to the time limits, we present part of results in the presentation. Please refer to our paper for the full details.

Effect of Introducing Four Configurations

Configurations	Details	Accuracy	CER (Normal)
Original ML-SUPERB	MMS + Transformer CTC	90.3	24.7 ± 12.3
Larger Downstream	MMS + E-Branchformer ATT-CTC	95.2	16.6 ± 11.8
SSL Model Fine-tuning	MMS + 9-14 layers partial fine-tuning CTC	**95.6**	**15.5 ± 10.3**
Efficient Model Adaptation	MMS + LoRA + Transformer ATT-CTC	94.2	18.7 ± 11.5
Supervised Pre-trained Model	Whisper Encoder + Transformer CTC	91.7	21.0 ± 12.5

Compared to the original ML-SUPERB, we observe **better performance** for LID and ASR across **ALL configurations** (normal setting)

Model Ranking given Different Configurations



- ML-SUPERB 2.0 is a **better estimate** of model performance compared to the original ML-SUPERB
- However, when considering **different** training settings, the ranking of upstream models can be **different**

Model Ranking given Different Configurations (Larger-scale Downstream Models)

	Transformer	Conformer	E-Branchformer
CTC	XLS-R	MMS	XLS-R
ATT-CTC	MMS	MMS	MMS

XLS-R wins

MMS wins

Compared to the original ML-SUPERB, the performance of XLS-R and MMS **depends on the choice of the downstream model**

Model Ranking given Different Configurations (Model Fine-tuning)

	Bottom	Middle	Top
CTC	MMS	MMS	MMS
ATT-CTC	MMS	MMS	MMS

XLS-R wins

MMS wins

Compared to the downstream model configuration,
XLS-R and MMS **rank differently** when considering fine-tuning approaches

Model Ranking given Different Configurations (Efficient Model Adaptation)

	LoRA	Adapter
CTC	XLS-R	XLS-R
ATT-CTC	MMS	XLS-R

XLS-R wins

MMS wins

Compared to previous experimental settings,
XLS-R and MMS **rank differently** when considering efficient model adaptation approaches

Supervised ASR vs. SSL Pre-trained Models

- Original ML-SUPERB only focuses on SSL pre-trained models

- ML-SUPERB 2.0 also allows the use of supervised ASR models
  - As long as the test sets from the ML-SUPERB 2.0 dataset are not used in training

- In our paper, we introduce some preliminary analysis on the comparison between supervised ASR and SSL pre-trained models

Supervised ASR vs. SSL Pre-trained Models

Pre-trained Model (Module)	Downstream Learning Modules	Accuracy	CER (Normal)
XLS-R	Additional transformer encoder + CTC prediction head	93.7	20.7 ± 10.8
MMS	Additional transformer encoder + CTC prediction head	93.6	21.0 ± 11.2
Whisper Encoder	Additional transformer encoder + CTC prediction head	91.7	21.0 ± 12.5
Whisper Encoder	Partial parameters in Whisper encoder (top layers) and additional transformer encoder + CTC prediction head	83.9	26.8 ± 15.0
Whisper Encoder + Decoder	Partial parameters in Whisper decoder (top layers)	85.5	25.6 ± 19.4

In our experiments, SSL pre-trained models demonstrate slightly **superior performance** compared to supervised ASR pre-trained models

Variation across Languages and Datasets

- Large standard deviations in both normal and few shot settings

  → This shows that there is **substantial variation** among the language-specific CERs

Variations across Languages and Datasets

- Large standard deviations in both the normal and few-shot settings

  → This shows that there is **substantial variation** among the language-specific CERs

- The impact of language differences is also highlighted by the CER of the worst-performing languages
  - In most cases, Lao or Min Nan Chinese have a CER > 60%

Variations across Languages and Datasets

- Large standard deviations in both the normal and few-shot settings
  → This shows that there is **substantial variation** among language-specific CERs

- The large impact of language differences is also highlighted by the CER of the worst-performing languages
  - In most cases, Lao or Min Nan Chinese have a CER > 60%

- Large CER differences between datasets in the same language
  → This highlights the **impact of domain or acoustic differences**

Conclusion of ML-SUPERB 2.0

- We present **an updated benchmark** for multilingual speech pre-trained models, which builds upon ML-SUPERB
- We investigate **four configurations** that ML-SUPERB does not consider
- We introduce **a broader set of evaluation metrics** to measure variation across languages and datasets

Findings of ML-SUPERB 2.0

- All four configurations **show improvements** over the configuration used in the original ML-SUPERB, which was likely underestimating model performance
- Model ranking depends on the configuration of the benchmark
- There is no single way to evaluate an SSL model. It must always be measured in the context of a specific downstream model and task
- We encourage research on methods that improve language/dataset robustness

Can we develop robust optimization methods to address the performance gap?

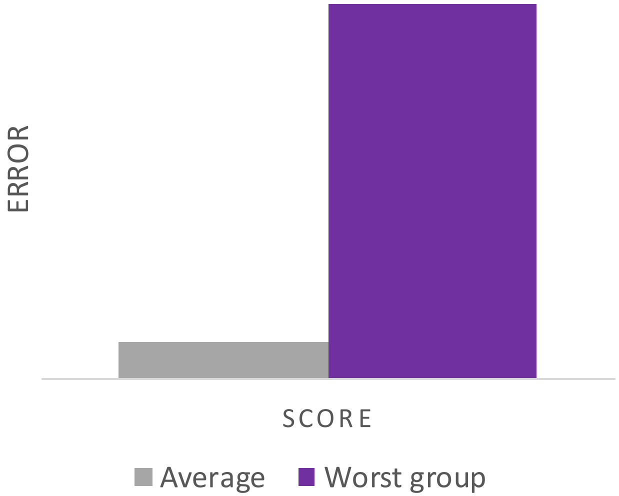
Standard approach: ERM

- Minimize the average loss on the training data

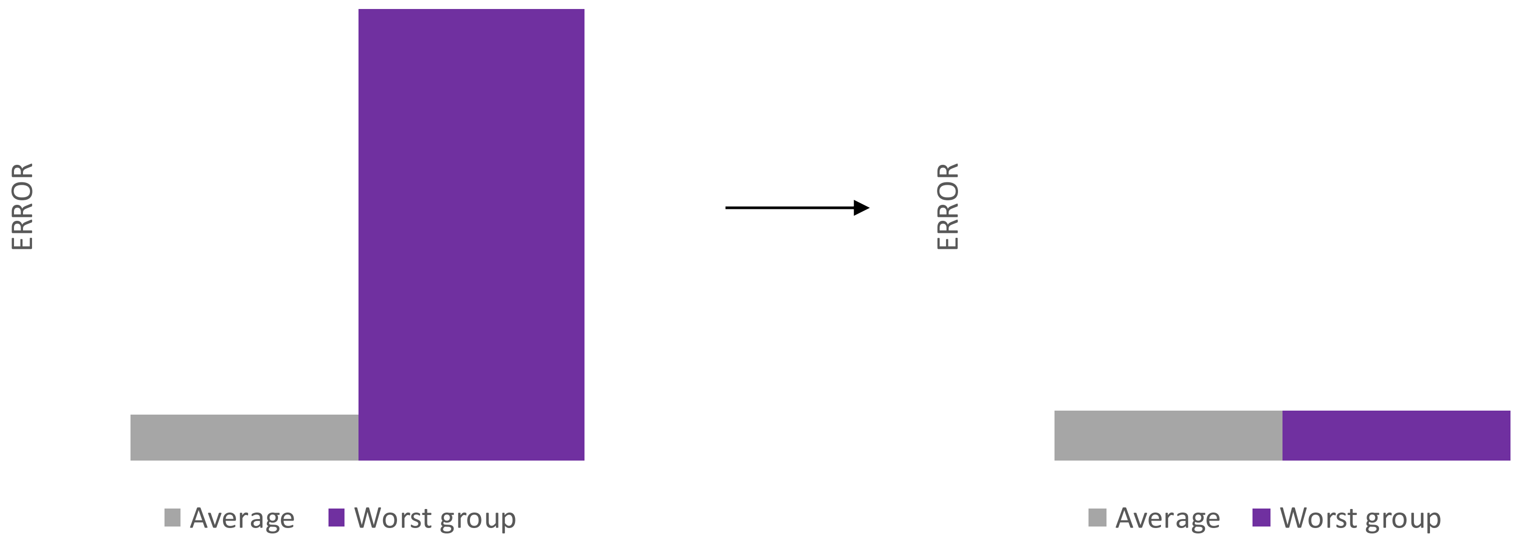
$$\hat{\theta}_{\text{ERM}} := \arg\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\hat{P}} \left[\ell(\theta; (x, y))\right]$$

Standard approach: ERM

- Minimize the average loss on the training data



Desired approach



Published as a conference paper at ICLR 2020

DISTRIBUTIONALLY ROBUST NEURAL NETWORKS FOR GROUP SHIFTS: ON THE IMPORTANCE OF REGULARIZATION FOR WORST-CASE GENERALIZATION

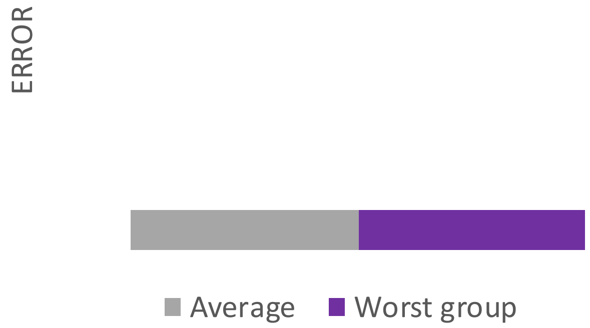
Shiori Sagawa*
Stanford University
ssagawa@cs.stanford.edu

Pang Wei Koh*
Stanford University
pangwei@cs.stanford.edu

Tatsunori B. Hashimoto
Microsoft
tahashim@microsoft.com

Percy Liang
Stanford University
pliang@cs.stanford.edu

Group Distributionally Robust Optimization



$$\hat{\theta}_{\text{DRO}} := \arg\min_{\theta\in\Theta} \left\{ \max_{g\in\mathcal{G}} \mathbb{E}_{(x,y)\sim\hat{P}_g} \left[\ell(\theta;(x,y))\right] \right\}$$

Minimize the worst-case expected loss over a set of pre-defined groups

Group DRO shows strong performance on image and text classification tasks but has not yet been successfully applied to speech

In practice

Algorithm 1 Online optimization algorithm for `group DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; loss function l; batch size B
Initialize $\theta^{(0)}$ $\boxed{\text{and } \{q_g\}}$
for $t = 1$ to T **do**
  $\mathcal{B} = \{\, (x_i, y_i, g_i)\,\}_{i=1}^{B}$
  for $g \in G$ **do**
    $\mathcal{L}_g \leftarrow 0$; $cnt_g \leftarrow 0$
    for $i = 1$ to B **do**
      if $g_i == g$ **then**
        $\mathcal{L}_g + = l(\theta^{(t-1)}; (x_i, y_i))$; $cnt_g + = 1$
      end if
    end for
    $\mathcal{L}_g \leftarrow \frac{\mathcal{L}_g}{cnt_g}$
    $q'_g \leftarrow q_g \exp(\eta_q \mathcal{L}_g)$
  end for
  for $g \in G$ **do**
    $q_g \leftarrow q'_g / \sum_{g'} q'_{g'}$ {normalize}
  end for
  $\mathcal{L} \leftarrow \sum_{g \in G} q_g \mathcal{L}_g$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta q_g^{(t)} \nabla \mathcal{L}$
end for

The training objective maintains a weight for each group, which are uniformly initialized and updated during training.

In practice

Algorithm 1 Online optimization algorithm for `group DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; loss function l; batch size B
Initialize $\theta^{(0)}$ and $\{q_g\}$
for $t = 1$ to T **do**
  $\mathcal{B} = \{(x_i, y_i, g_i)\}_{i=1}^{B}$
  for $g \in G$ **do**
    $\mathcal{L}_g \leftarrow 0$; $cnt_g \leftarrow 0$
    for $i = 1$ to B **do**
      if $g_i == g$ **then**
        $\mathcal{L}_g += l(\theta^{(t-1)}; (x_i, y_i))$; $cnt_g += 1$
      end if
    end for
    $\mathcal{L}_g \leftarrow \frac{\mathcal{L}_g}{cnt_g}$
    $q'_g \leftarrow q_g \exp(\eta_q \mathcal{L}_g)$
  end for
  for $g \in G$ **do**
    $q_g \leftarrow q'_g / \sum_{g'} q'_{g'}$ {normalize}
  end for
  $\mathcal{L} \leftarrow \sum_{g \in G} q_g \mathcal{L}_g$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta q_g^{(t)} \nabla \mathcal{L}$
end for

Compute the average training loss for each group in a batch and compute an exponential multiplicative update to the group weight vector.

In practice

Algorithm 1 Online optimization algorithm for `group DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; loss function l; batch size B
Initialize $\theta^{(0)}$ and $\{q_g\}$
for $t = 1$ to T **do**
  $\mathcal{B} = \{ (x_i, y_i, g_i) \}_{i=1}^{B}$
  for $g \in G$ **do**
    $\mathcal{L}_g \leftarrow 0$; $cnt_g \leftarrow 0$
    for $i = 1$ to B **do**
      if $g_i == g$ **then**
        $\mathcal{L}_g += l(\theta^{(t-1)}; (x_i, y_i))$; $cnt_g += 1$
      end if
    end for
    $\mathcal{L}_g \leftarrow \frac{\mathcal{L}_g}{cnt_g}$
    $q'_g \leftarrow q_g \exp(\eta_q \mathcal{L}_g)$
  end for
  for $g \in G$ **do**
    $q_g \leftarrow q'_g / \sum_{g'} q'_{g'}$ {normalize}
  end for
  $\mathcal{L} \leftarrow \sum_{g \in G} q_g \mathcal{L}_g$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta q_g^{(t)} \nabla \mathcal{L}$
end for

Normalize the group weight vector to form a valid probability distribution.

In practice

Algorithm 1 Online optimization algorithm for `group DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; loss function l; batch size B
Initialize $\theta^{(0)}$ and $\{q_g\}$
for $t = 1$ to T **do**
  $\mathcal{B} = \{\, (x_i, y_i, g_i)\,\}_{i=1}^{B}$
  for $g \in G$ **do**
    $\mathcal{L}_g \leftarrow 0$; $cnt_g \leftarrow 0$
    for $i = 1$ to B **do**
      if $g_i == g$ **then**
        $\mathcal{L}_g + = l(\theta^{(t-1)}; (x_i, y_i))$; $cnt_g + = 1$
      end if
    end for
    $\mathcal{L}_g \leftarrow \frac{\mathcal{L}_g}{cnt_g}$
    $q'_g \leftarrow q_g \exp(\eta_q \mathcal{L}_g)$
  end for
  for $g \in G$ **do**
    $q_g \leftarrow q'_g / \sum_{g'} q'_{g'}$ {normalize}
  end for
  $\mathcal{L} \leftarrow \sum_{g \in G} q_g \mathcal{L}_g$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta q_g^{(t)} \nabla \mathcal{L}$
end for

The loss used in the gradient descent update for the batch is then the sum of the group losses weighed by the group weights.

Challenges

Best-performing models on ML-SUPERB 2.0 are fine-tuned using CTC

Challenges

Best-performing models on ML-SUPERB 2.0 are fine-tuned using CTC

❗ Challenges optimizing CTC loss using group DRO

Challenges

Best-performing models on ML-SUPERB 2.0 are fine-tuned using CTC

❗ Challenges optimizing CTC loss using group DRO

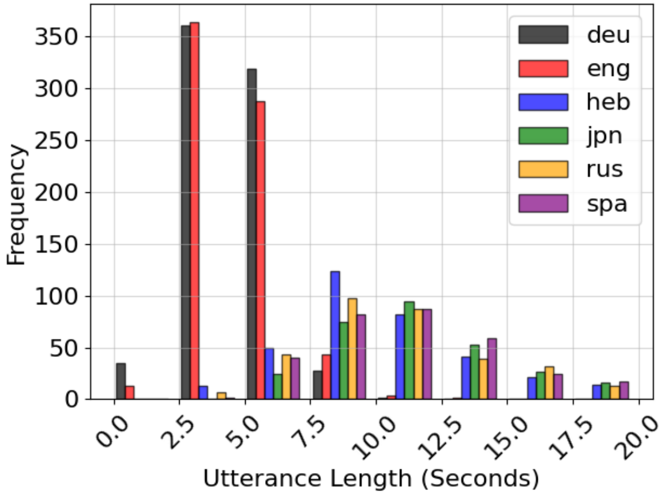
❗ Group DRO is restricted to applications where the losses between groups in the training data are comparable

Challenges

CTC loss scales with the length of the audio samples and the length of the corresponding transcriptions

Challenges

CTC loss scales with the length of the audio samples and the length of the corresponding transcriptions



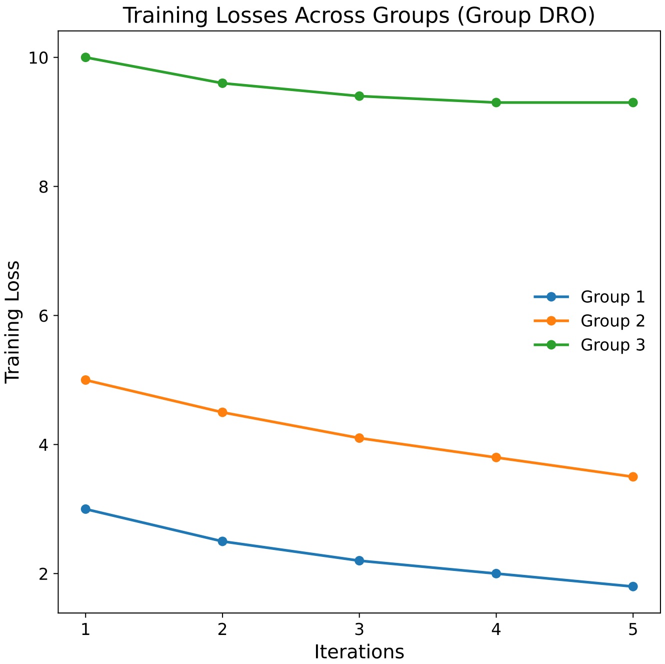
Challenges

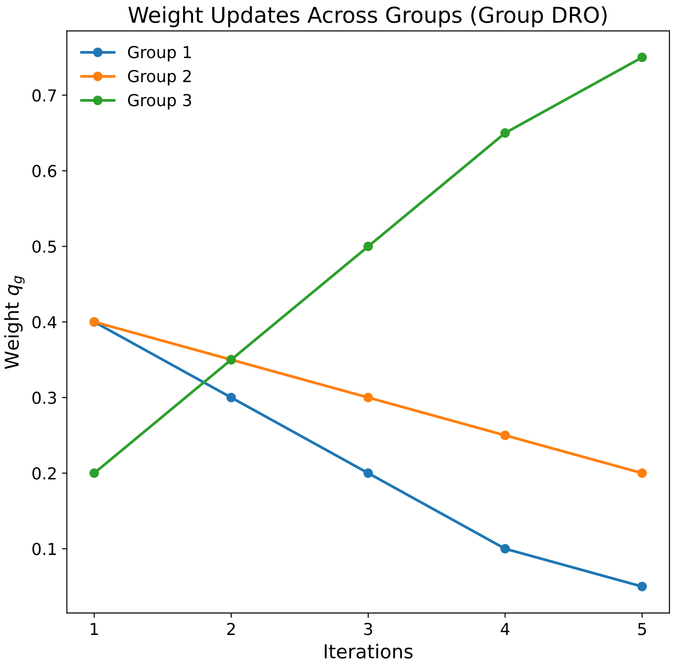
❗ Even when length differences would be taken into account, group losses might still not be comparable

❗ Audio samples can be from different speakers or domains

❗ This may lead to consistently higher or effectively irreducible losses for some groups

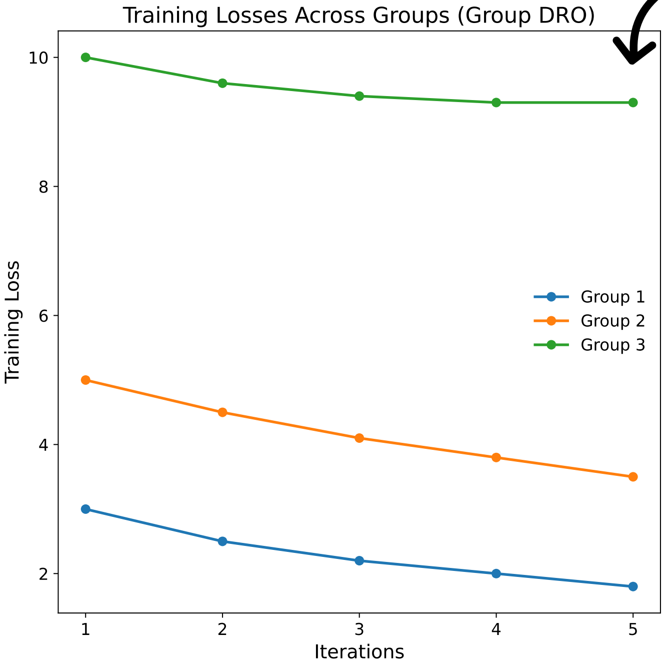
Challenges

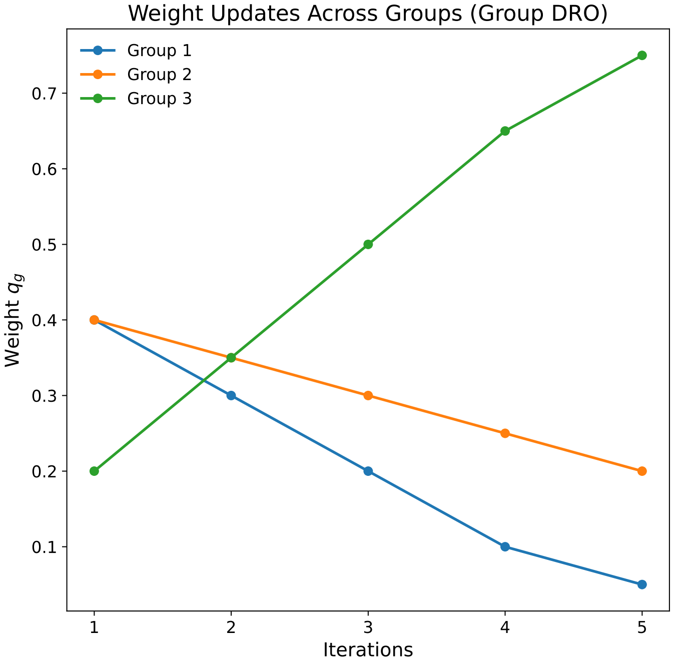

Training Losses Across Groups (Group DRO)
Training Loss
Iterations
Group 1
Group 2
Group 3
10
8
6
4
2
1
2
3
4
5


Weight Updates Across Groups (Group DRO)
Weight q_g
Iterations
Group 1
Group 2
Group 3
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1
2
3
4
5

Challenges

Focus on this group,
leading to undertraining of group 1 and 2





To address these limitations we present **CTC-DRO**

To address these limitations we present **CTC-DRO**

Duration-matched group losses

To address these limitations we present **CTC-DRO**

- Duration-matched group losses
- Group-based regularization

Duration-matched group losses

Algorithm 2 Optimization algorithm for `CTC-DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t

Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses

for $t = 1$ to T **do**

  $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;

  $\Sigma_{i=1}^{B_t}\ duration(x_i) = d$

  $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t

  $\text{gr\_losses}[g] \leftarrow \text{gr\_losses}[g] \cup \left\{\sum_{i=1}^{B_t} \ell_i\right\}$

  if $\text{gr\_losses}[g] \neq \emptyset\ \forall g$ **then**

    for each group g **do**

      $\bar{\ell}_g = \dfrac{\sum_{\mathcal{L} \in \text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}$

      $q'_g \leftarrow q'_g \times \exp\left(\dfrac{\eta_q \bar{\ell}_g}{q'_g + \alpha}\right)$

      $\text{gr\_losses}[g] \leftarrow \emptyset$

    end for

    for each group g **do**

      $q_g \leftarrow \dfrac{q'_g}{\sum_{g'} q'_{g'}}$

    end for

  end if

  $\tilde{\ell}_i = \ell_i \times q_g \times m \quad$ for $i = 1, \ldots, B_t$

  $\tilde{\mathcal{L}} = \dfrac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$

  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$

end for

To deal with the scaling properties of the CTC loss, we batch the same total duration of audio data for each group.

Duration-matched group losses

Algorithm 2 Optimization algorithm for `CTC-DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t
Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses
for $t = 1$ to T **do**
  $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;
  $\Sigma_{i=1}^{B_t}\ duration(x_i) = d$
  $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t
  $\text{gr\_losses}[g] \leftarrow \text{gr\_losses}[g] \cup \left\{\sum_{i=1}^{B_t} \ell_i\right\}$
  if $\text{gr\_losses}[g] \neq \emptyset\ \forall g$ **then**
    for each group g **do**
      $\boxed{\bar{\ell}_g = \dfrac{\sum_{\mathcal{L} \in \text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}}$
      $q'_g \leftarrow q'_g \times \exp\left(\dfrac{\eta_q \bar{\ell}_g}{q'_g + \alpha}\right)$
      $\text{gr\_losses}[g] \leftarrow \emptyset$
    end for
    for each group g **do**
      $q_g \leftarrow \dfrac{q'_g}{\sum_{g'} q'_{g'}}$
    end for
  end if
  $\tilde{\ell}_i = \ell_i \times q_g \times m \quad \text{for } i = 1, \ldots, B_t$
  $\tilde{\mathcal{L}} = \dfrac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$
end for

Instead of averaging losses within a batch, we sum them. This prevents artificially low or high averages for batches with many short utterances or few long ones. Since each batch has the same total duration, the sums remain comparable across groups.

Duration-matched group losses

Algorithm 2 Optimization algorithm for `CTC-DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t
Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses
for $t = 1$ to T **do**
  $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;
  $\Sigma_{i=1}^{B_t}\, duration(x_i) = d$
  $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t
  $\text{gr\_losses}[g] \leftarrow \text{gr\_losses}[g] \cup \left\{\sum_{i=1}^{B_t} \ell_i\right\}$
  if $\text{gr\_losses}[g] \neq \emptyset\ \forall g$ **then**
    for each group g **do**
      $\bar{\ell}_g = \dfrac{\sum_{\mathcal{L} \in \text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}$
      $q'_g \leftarrow q'_g \times \exp\left(\dfrac{\eta_q \bar{\ell}_g}{q'_g + \alpha}\right)$
      $\text{gr\_losses}[g] \leftarrow \emptyset$
    end for
    for each group g **do**
      $q_g \leftarrow \dfrac{q'_g}{\sum_{g'} q'_{g'}}$
    end for
  end if
  $\tilde{\ell}_i = \ell_i \times q_g \times m$ for $i = 1, \ldots, B_t$
  $\tilde{\mathcal{L}} = \dfrac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$
end for

Updates are done only after seeing all of the groups, simulating a larger batch containing all of the groups.

Group-based regularization

Algorithm 2 Optimization algorithm for `CTC-DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t

Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses

for $t = 1$ to T **do**

  $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;

  $\Sigma_{i=1}^{B_t}\ duration(x_i) = d$

  $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t

  gr_losses$[g] \leftarrow$ gr_losses$[g] \cup \left\{\sum_{i=1}^{B_t} \ell_i\right\}$

  if gr_losses$[g] \neq \emptyset\ \forall g$ **then**

    for each group g **do**

      $\bar{\ell}_g = \dfrac{\sum_{\mathcal{L} \in \text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}$

      $\boxed{q'_g \leftarrow q'_g \times \exp\left(\dfrac{\eta_q \bar{\ell}_g}{q'_g + \alpha}\right)}$

      gr_losses$[g] \leftarrow \emptyset$

    end for

    for each group g **do**

      $q_g \leftarrow \dfrac{q'_g}{\sum_{g'} q'_{g'}}$

    end for

  end if

  $\tilde{\ell}_i = \ell_i \times q_g \times m \quad \text{for } i = 1, \ldots, B_t$

  $\tilde{\mathcal{L}} = \dfrac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$

  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$

end for

We perform *softer* updates to the group weights q_g, which are now inversely proportional to the current q_g as well as proportional to the training loss.

Group-based regularization

Algorithm 2 Optimization algorithm for CTC−DRO, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t
Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses
for $t = 1$ to T **do**
  $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;
  $\Sigma_{i=1}^{B_t}\ duration(x_i) = d$
  $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t
  $\text{gr\_losses}[g] \leftarrow \text{gr\_losses}[g] \cup \left\{\sum_{i=1}^{B_t} \ell_i\right\}$
  if $\text{gr\_losses}[g] \neq \emptyset\ \forall g$ **then**
    for each group g **do**
      $\bar{\ell}_g = \dfrac{\sum_{\mathcal{L} \in \text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}$
      $q'_g \leftarrow q'_g \times \exp\left(\dfrac{\eta_q \bar{\ell}_g}{q'_g + \alpha}\right)$
      $\text{gr\_losses}[g] \leftarrow \emptyset$
    end for
    for each group g **do**
      $q_g \leftarrow \dfrac{q'_g}{\sum_{g'} q'_{g'}}$
    end for
  end if
  $\tilde{\ell}_i = \ell_i \times q_g \times m \quad \text{for } i = 1, \ldots, B_t$
  $\tilde{\mathcal{L}} = \dfrac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$
end for

Discourages groups from attaining very high q_g, mitigating group dro's issues with varying irreducibility of losses across groups

Group-based regularization

Algorithm 2 Optimization algorithm for CTC−DRO, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t

Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses

for $t = 1$ to T **do**

  $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;

  $\Sigma_{i=1}^{B_t}\ duration(x_i) = d$

  $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t

  $\text{gr\_losses}[g] \leftarrow \text{gr\_losses}[g] \cup \left\{\sum_{i=1}^{B_t} \ell_i\right\}$

  if $\text{gr\_losses}[g] \neq \emptyset\ \forall g$ **then**

    for each group g **do**

      $\bar{\ell}_g = \dfrac{\sum_{\mathcal{L} \in \text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}$

      $q'_g \leftarrow q'_g \times \exp\left(\dfrac{\eta_q \bar{\ell}_g}{q'_g + \alpha}\right)$

      $\text{gr\_losses}[g] \leftarrow \emptyset$

    end for

    for each group g **do**

      $q_g \leftarrow \dfrac{q'_g}{\sum_{g'} q'_{g'}}$

    end for

  end if

  $\tilde{\ell}_i = \ell_i \times q_g \times m \quad$ for $i = 1, \ldots, B_t$

  $\tilde{\mathcal{L}} = \dfrac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$

  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$

end for

Ensures groups with lower q_g receive larger updates when CTC losses are similar, helping them catch up during training

Algorithm 2 Optimization algorithm for `CTC-DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t
Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses
for $t = 1$ to T **do**
  $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;
  $\Sigma_{i=1}^{B_t} \, duration(x_i) = d$
  $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t
  $\text{gr\_losses}[g] \leftarrow \text{gr\_losses}[g] \cup \left\{ \sum_{i=1}^{B_t} \ell_i \right\}$
  if $\text{gr\_losses}[g] \neq \emptyset \; \forall g$ **then**
    for each group g **do**
      $\bar{\ell}_g = \dfrac{\sum_{\mathcal{L} \in \text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}$
      $q'_g \leftarrow q'_g \times \exp\left(\dfrac{\eta_q \bar{\ell}_g}{q'_g + \alpha} \right)$
      $\text{gr\_losses}[g] \leftarrow \emptyset$
    end for
    for each group g **do**
      $q_g \leftarrow \dfrac{q'_g}{\sum_{g'} q'_{g'}}$
    end for
  end if
  $\tilde{\ell}_i = \ell_i \times q_g \times m \quad$ for $i = 1, \ldots, B_t$
  $\tilde{\mathcal{L}} = \dfrac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$
end for

Group-based regularization

Prevents under-training by reducing divergence in DRO weights across groups

Group-based regularization

Algorithm 2 Optimization algorithm for CTC−DRO, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t

Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses

for $t = 1$ to T **do**

  $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;

  $\Sigma_{i=1}^{B_t}\ duration(x_i) = d$

  $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t

  $\text{gr\_losses}[g] \leftarrow \text{gr\_losses}[g] \cup \left\{\sum_{i=1}^{B_t} \ell_i\right\}$

  if $\text{gr\_losses}[g] \neq \emptyset\ \forall g$ **then**

    for each group g **do**

      $\bar{\ell}_g = \dfrac{\sum_{\mathcal{L}\in\text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}$

      $\boxed{q'_g \leftarrow q'_g \times \exp\left(\dfrac{\eta_q \bar{\ell}_g}{q'_g + \alpha}\right)}$

      $\text{gr\_losses}[g] \leftarrow \emptyset$

    end for

    for each group g **do**

      $q_g \leftarrow \dfrac{q'_g}{\sum_{g'} q'_{g'}}$

    end for

  end if

  $\tilde{\ell}_i = \ell_i \times q_g \times m \quad \text{for } i = 1, \ldots, B_t$

  $\tilde{\mathcal{L}} = \dfrac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$

  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$

end for

★ Higher values of the new hyperparameter α reduce the strength of this effect

Algorithm 2 Optimization algorithm for CTC−DRO, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t
Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses
for $t = 1$ to T **do**
  $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;
  $\Sigma_{i=1}^{B_t}\ duration(x_i) = d$
  $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t
  $\text{gr\_losses}[g] \leftarrow \text{gr\_losses}[g] \cup \left\{\sum_{i=1}^{B_t} \ell_i\right\}$
  if $\text{gr\_losses}[g] \neq \emptyset\ \forall g$ **then**
    for each group g **do**
      $$\bar{\ell}_g = \frac{\sum_{\mathcal{L} \in \text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}$$
      $$q'_g \leftarrow q'_g \times \exp\left(\frac{\eta_q \bar{\ell}_g}{q'_g + \alpha}\right)$$
      $\text{gr\_losses}[g] \leftarrow \emptyset$
    end for
    for each group g **do**
      $$q_g \leftarrow \frac{q'_g}{\sum_{g'} q'_{g'}}$$
    end for
  end if
  $\boxed{\tilde{\ell}_i = \ell_i \times q_g \times m \quad \text{for } i = 1, \ldots, B_t}$
  $$\tilde{\mathcal{L}} = \frac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$$
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$
end for

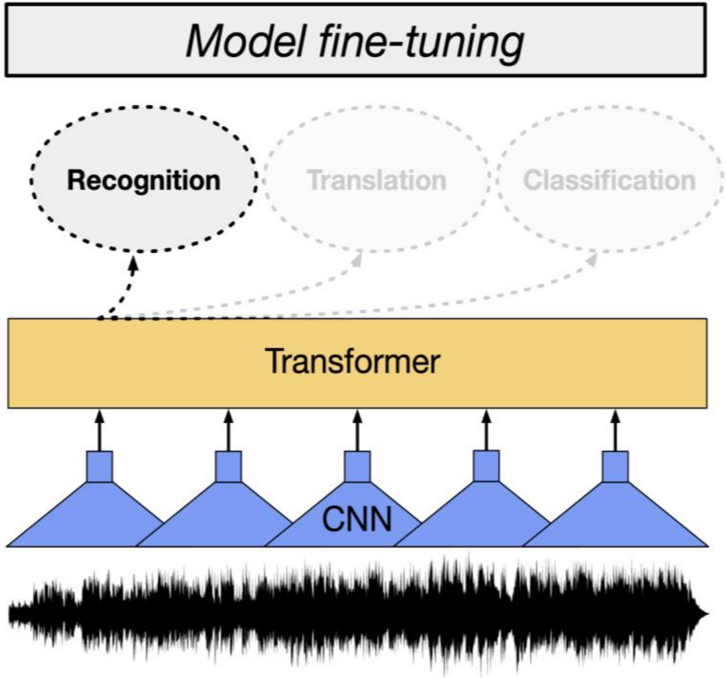
We multiply losses by the number of groups, which improves training stability. This way, losses are also comparable to models trained without CTC-DRO, removing the need to tune hyperparameters for both models.

Algorithm 2 Optimization algorithm for `CTC-DRO`, θ represents the model parameters.

Input: Step sizes η_q, η_θ; smoothing parameter α; number of groups m; loss function l; duration of each batch d; number of data points in t^{th} batch B_t
Initialize $\theta^{(0)}$, $\{q_g\}$ and gr_losses
for $t = 1$ to T **do**
    $\mathcal{B} = \{(x_i, y_i, g)\}_{i=1}^{B_t}$;
    $\Sigma_{i=1}^{B_t}\ duration(x_i) = d$
    $\ell_i = \ell(\theta^{(t-1)}; (x_i, y_i))$ **for** $i = 1$ to B_t
    $\text{gr\_losses}[g] \leftarrow \text{gr\_losses}[g] \cup \left\{\sum_{i=1}^{B_t} \ell_i\right\}$
    if $\text{gr\_losses}[g] \neq \emptyset\ \forall g$ **then**
        for each group g **do**
            $\bar{\ell}_g = \dfrac{\sum_{\mathcal{L} \in \text{gr\_losses}[g]} \mathcal{L}}{|\text{gr\_losses}[g]|}$
            $q'_g \leftarrow q'_g \times \exp\left(\dfrac{\eta_q \bar{\ell}_g}{q'_g + \alpha}\right)$
            $\text{gr\_losses}[g] \leftarrow \emptyset$
        end for
        for each group g **do**
            $q_g \leftarrow \dfrac{q'_g}{\sum_{g'} q'_{g'}}$
        end for
    end if
    $\tilde{\ell}_i = \ell_i \times q_g \times m \quad \text{for } i = 1, \ldots, B_t$
    $\tilde{\mathcal{L}} = \dfrac{1}{B_t} \sum_{i=1}^{B_t} \tilde{\ell}_i$
    $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta \nabla_\theta \tilde{\mathcal{L}}$
end for

We accumulate gradients across 16 batches before updating model parameters, simulating larger batches with multiple groups.

Experimental setup



Experimental setup

- MMS and XLS-R fine-tuned with and without **CTC-DRO** and with **group DRO**
- Groups in algorithm correspond to individual languages in training datasets

Experimental setup

- Two-layer Transformer encoder added on top of pre-trained models to predict characters using CTC
- All model weights updated during fine-tuning
- Learning rate tuned on development data
  - DRO models use same learning rate as baseline (i.e., non-DRO) models for clear comparison
- DRO-specific hyperparameters:
  - Step size η_q: 10^{-3} and 10^{-4}
  - Smoothing parameter α: 0.1, 0.5, and 1

Dataset: following ML-SUPERB 2.0

No.	Language	ISO	Corpus
1	Czech	ces	CV
	Mandarin	cmn	Fleurs
	Min Nan	nan	CV
	Polish	pol	M-AILABS
	Romanian	ron	Fleurs
	Spanish	spa	Voxforge
2	Cantonese	yue	Fleurs
	Croatian	hrv	Fleurs
	English	eng	LAD
	Italian	ita	Fleurs
	Persian	fas	CV
	Slovak	slk	Fleurs
3	Khmer	khm	Fleurs
	Korean	kor	Fleurs
	Northern Kurdish	kmr	CV
	Norwegian Nynorsk	nno	CV
	Southern Ndebele	nbl	NCHLT
	Tatar	tat	CV
4	Sindhi	snd	Fleurs
	Slovenian	slv	CV
	Southern Sotho	sot	Googlei18n
	Spanish	spa	M-AILABS
	Urdu	urd	Fleurs
	Western Mari	mrj	CV
5	English	eng	Voxforge
	German	deu	Voxforge
	Hebrew	heb	Fleurs
	Japanese	jpn	Fleurs
	Russian	rus	Fleurs
	Spanish	spa	Fleurs

Results

Model	Type	LID	ces	cmn	nan	pol	ron	spa	CER
MMS	Baseline	96.62	8.36	56.67	59.74	3.65	14.29	1.79	24.08
	`group DRO`	62.80	24.49	48.11	86.01	5.35	18.11	9.10	31.86
	`CTC-DRO`	97.58	10.36	45.08	56.15	3.61	14.09	1.94	21.87
XLSR	Baseline	77.78	26.56	187.93	84.11	11.16	30.17	11.21	58.53
	`group DRO`	86.82	27.43	86.55	82.74	11.64	25.47	7.76	40.27
	`CTC-DRO`	87.76	18.36	59.72	64.04	7.85	26.57	7.23	30.63

Results

Model	Type	LID	eng	fas	hrv	ita	slk	yue	CER
MMS	Baseline	98.43	0.18	21.97	10.58	4.57	10.64	45.17	15.52
	`group DRO`	97.32	10.79	29.87	12.43	8.90	12.43	56.91	21.89
	`CTC-DRO`	98.20	0.79	22.73	8.96	6.71	5.78	43.53	14.75
XLSR	Baseline	96.64	0.38	18.97	6.83	4.68	8.91	64.79	17.43
	`group DRO`	88.59	11.89	31.39	12.34	5.69	11.72	59.98	22.17
	`CTC-DRO`	96.38	0.78	21.79	11.93	5.78	8.30	43.67	15.38

Results

Model	Type	LID	khm	kmr	kor	nbl	nno	tat	CER
MMS	Baseline	99.17	32.07	11.60	36.60	8.09	2.44	9.86	16.78
	`group DRO`	98.84	30.72	18.90	33.39	18.52	10.01	13.54	20.85
	`CTC-DRO`	99.17	32.75	11.82	29.59	8.47	2.88	10.20	15.95
XLSR	Baseline	97.85	34.01	11.37	32.57	8.01	2.20	10.38	16.42
	`group DRO`	96.53	36.88	21.44	35.29	24.16	10.49	16.59	24.14
	`CTC-DRO`	96.53	31.38	11.94	32.43	8.36	2.97	12.60	16.61

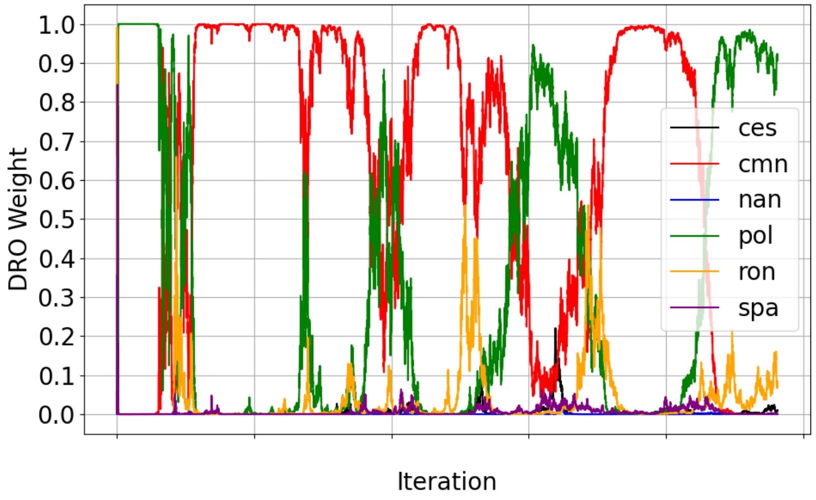
Results

Model	Type	LID	mrj	slv	snd	sot	spa	urd	CER
MMS	Baseline	89.86	10.34	12.68	20.37	14.89	4.53	28.38	15.20
	`group DRO`	92.08	9.35	13.67	19.86	16.09	4.34	30.53	15.64
	`CTC-DRO`	94.68	9.55	12.79	19.50	14.62	4.72	26.27	14.58
XLSR	Baseline	88.38	14.00	4.83	23.33	11.57	4.16	29.67	14.59
	`group DRO`	83.45	15.69	26.31	19.39	23.47	3.87	23.92	18.78
	`CTC-DRO`	88.91	11.95	6.69	20.97	13.80	4.79	24.22	13.74

Results

Model	Type	LID	deu	eng	heb	jpn	rus	spa	CER
MMS	Baseline	98.43	6.90	11.78	33.73	98.21	12.73	7.92	28.55
	group DRO	66.96	28.69	27.06	35.09	61.12	17.68	9.54	29.86
	CTC-DRO	98.85	9.99	14.13	31.87	52.98	13.94	8.67	21.93
XLSR	Baseline	89.00	5.22	11.43	37.98	120.94	11.84	7.85	32.54
	group DRO	57.71	29.24	27.44	44.66	98.11	17.66	11.20	38.05
	CTC-DRO	90.97	6.11	11.23	41.49	77.12	11.08	8.92	25.99

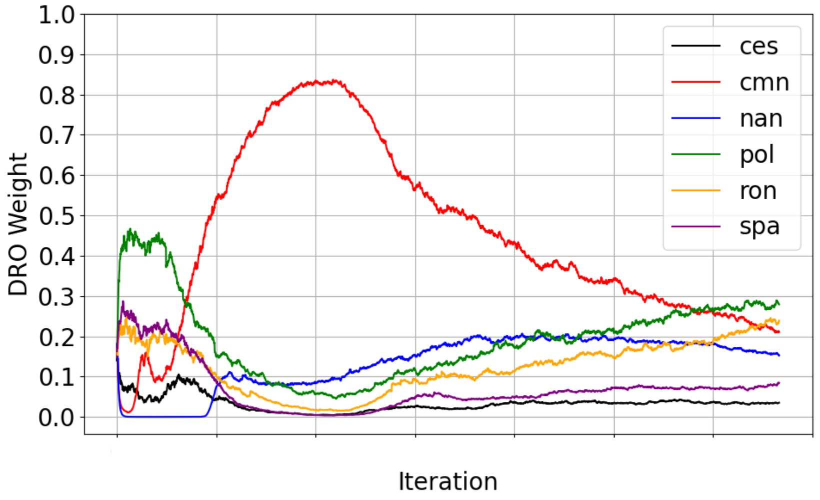
Analysis: group DRO



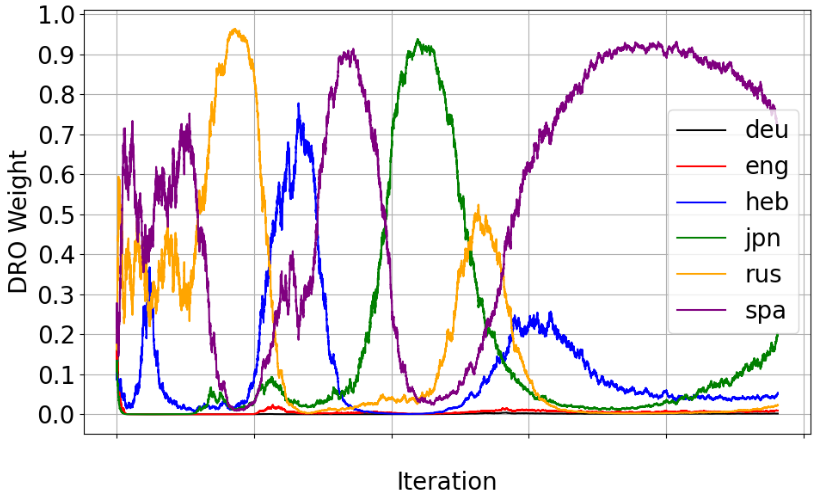
Weights fluctuate. During large portions of training, all of the DRO weight is concentrated on a single language, which is not the worst-performing language

Analysis: CTC-DRO

Weights fluctuate less, mitigating undertraining of any language. The worst-performing language has one of the largest weights.



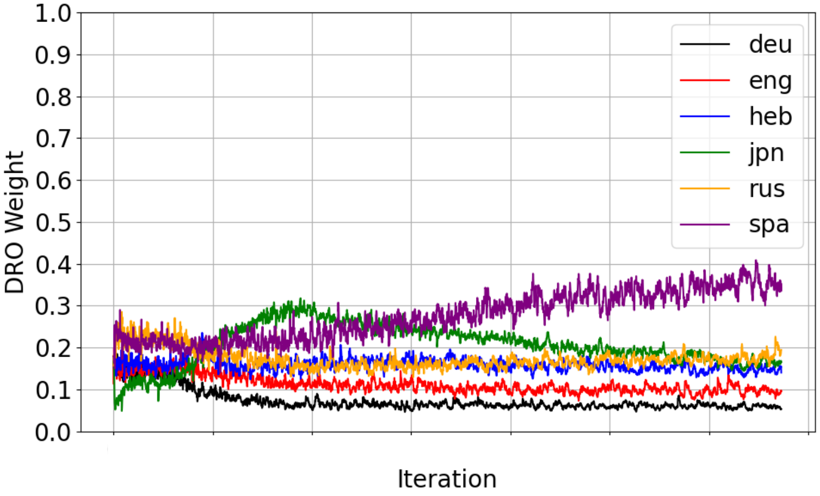
Analysis: group DRO



Weights fluctuate. During large portions of training, all of the DRO weight is concentrated on a single language, which is not the worst-performing language

Analysis: CTC-DRO

Weights are grouped much more tightly, mitigating undertraining of any language. The worst-performing language has one of the largest weights.



Analysis

Model	Type	LID	deu	eng	heb	jpn	rus	spa	CER
MMS	Baseline	98.43	6.90	11.78	33.73	98.21	12.73	7.92	28.55
	`CTC-DRO`	98.85	9.99	14.13	31.87	52.98	13.94	8.67	21.93
	`CTC-DRO` - duration-matched group losses	66.08	19.36	21.24	30.94	84.61	12.88	8.26	29.5
	`CTC-DRO` - group-based regularization	13.22	95.63	96.01	98.77	102.13	97.41	97.28	97.9
XLSR	Baseline	89.00	5.22	11.43	37.98	120.94	11.84	7.85	32.54
	`CTC-DRO`	90.97	6.11	11.23	41.49	77.12	11.08	8.92	25.99
	`CTC-DRO` - duration-matched group losses	51.54	35.60	36.54	72.91	115.23	27.43	15.90	50.6
	`CTC-DRO` - group-based regularization	43.17	18.52	24.49	69.85	194.20	41.21	19.88	61.4

Conclusion of CTC-DRO

- We find that **CTC-DRO** consistently reduced the worst-language CER and improved the average CER in most cases
- Future work will include different models, scale-up the number of languages, and handle multi-dimensional group definitions (e.g., language, gender, age)

Images are generated by DALL-E or directly from Flaticon.com

Conclusions

- ML-SUPERB 2.0 provides a way to **reliably measure** speech recognition model performance
- CTC-DRO **reduces the performance gap between languages** to help **improve universal access to modern speech technology**

Acknowledgements and contact information

bartelds@stanford.edu | martijnbartelds.nl | @barteldsmartijn