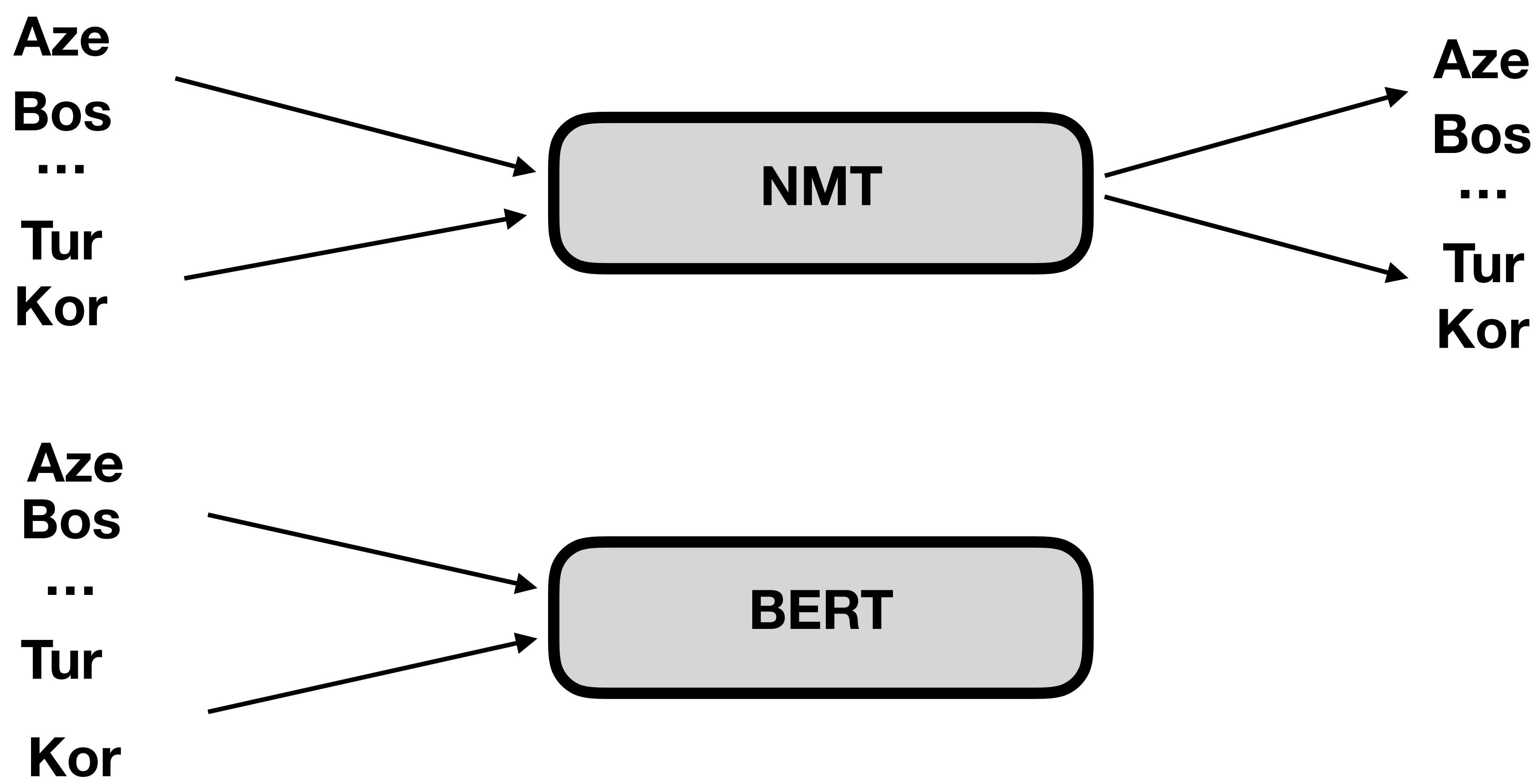
Balancing Training for Multilingual Neural Machine Translation

Xinyi Wang, Yulia Tsvetkov, Graham Neubig

Language Technologies Institute

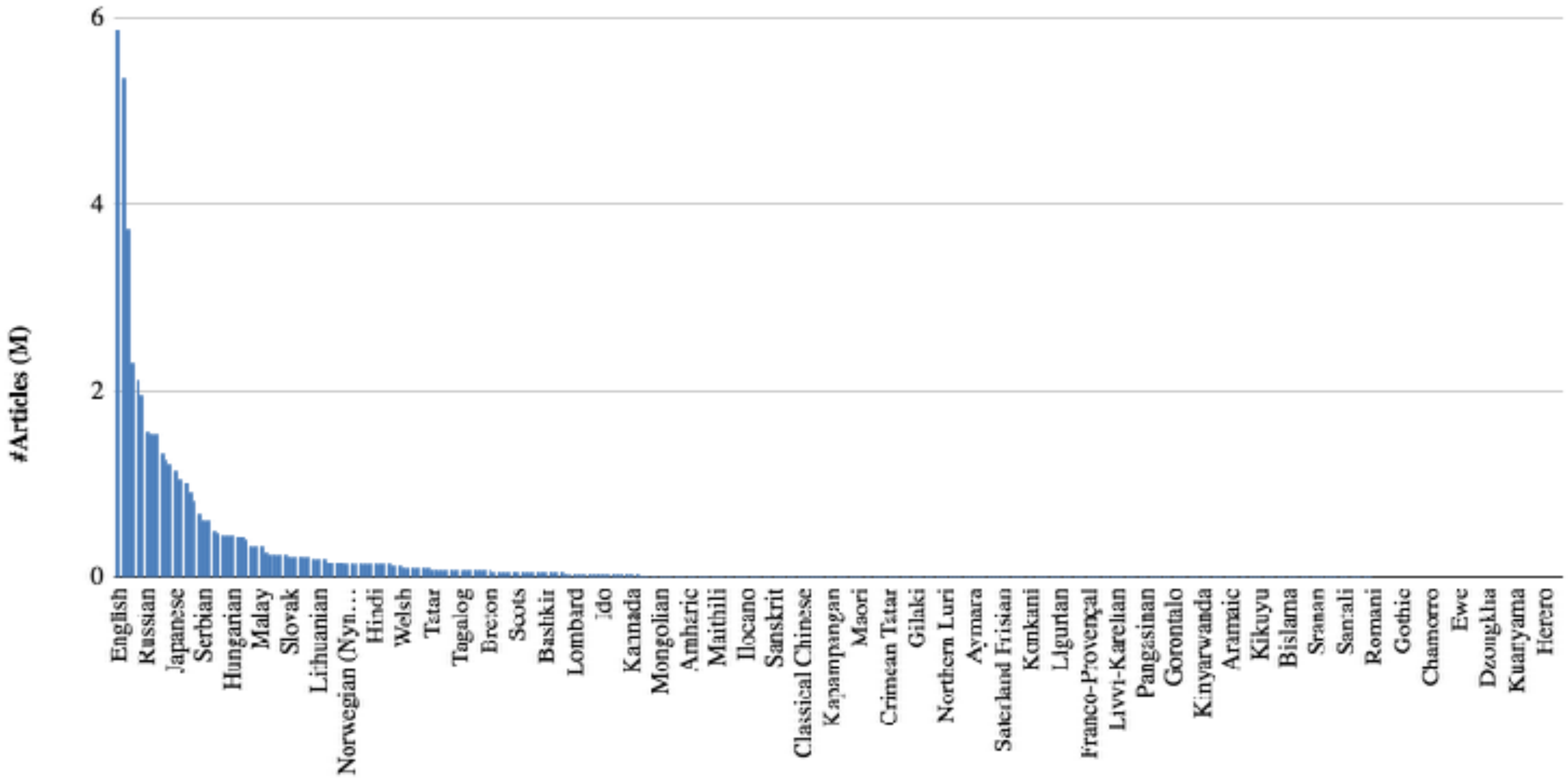
Carnegie Mellon University

Multilingual Training



- Resource efficient, easy to deploy
- Accuracy benefit from cross-lingual transfer

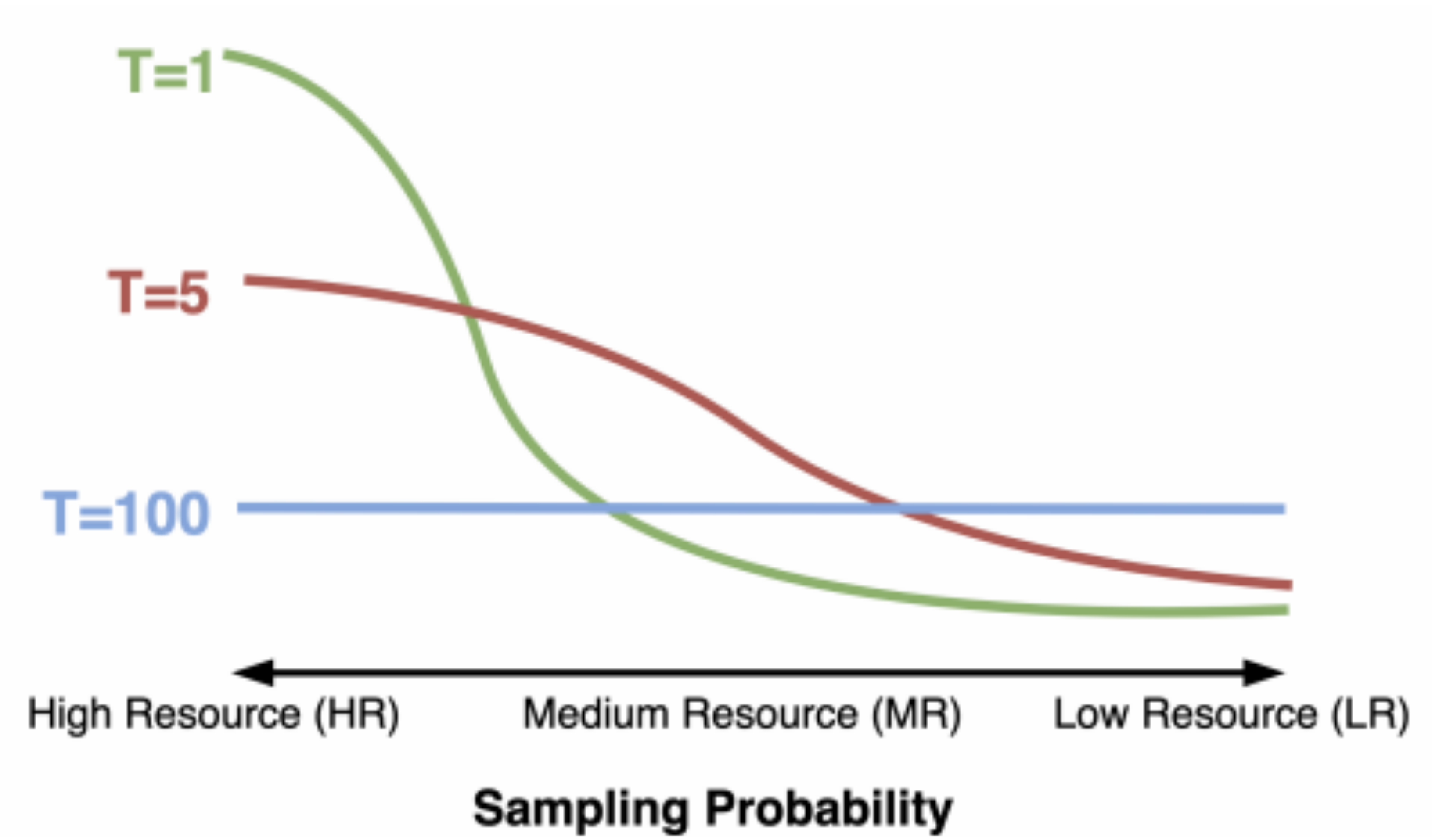
Multilingual Data are Imbalanced



- Need to upsample LRL data

Data Source: Wikipedia articles from different languages

Heuristic Sampling of Data



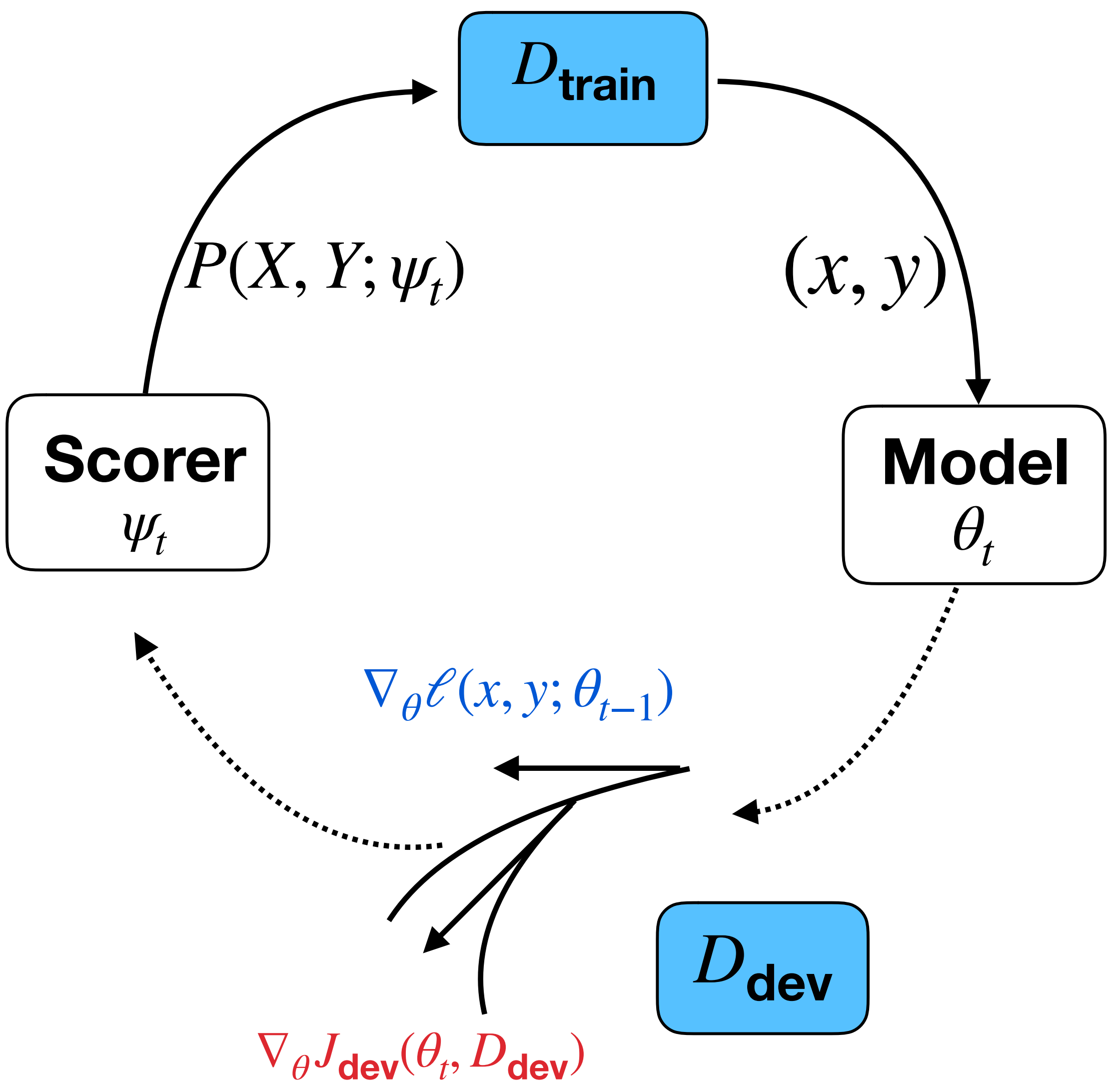
- Used in SOTA Multilingual BERT (Conneau et al. 2019) and Multilingual NMT (Arivazhagan et al. 2019, Aharoni et al., 2019)

- Can we **learn** the data sampling strategy directly?

Picture From: Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges, Arivazhagan et. al. 2019

Differentiable Data Selection

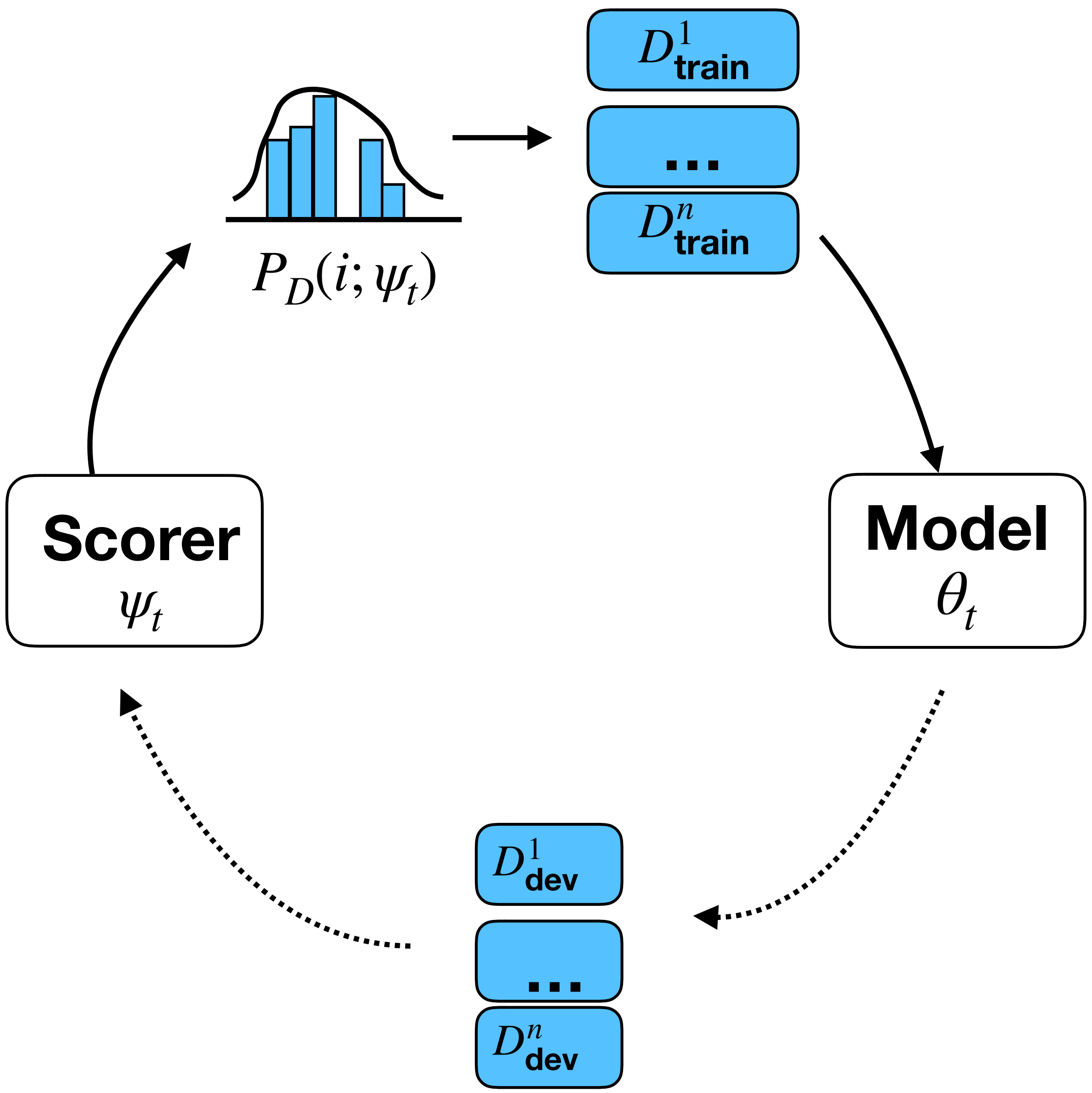
Learn a
data sampling strategy



- A general purpose ML method to learn weighting of training data to optimize a separate held-out data (Wang et al. 2019)
- Learns data scorer $P(x, y; \psi)$ to minimize dev loss $J(\theta; D_{\text{dev}})$
- Main idea: scorer should up-weight data with similar gradient as the dev data

$$R(x, y; \theta) \approx \cos\left(\underbrace{\nabla_\theta \big(J(\theta_t, D_{\text{dev}})\big)}_{\text{dev gradient}}, \underbrace{\nabla_\theta \ell(x, y; \theta_{t-1})}_{\text{train gradient}} \right)$$

DDS for Multilingual Data Usage



- Existing Approach: temperature based heuristic sampling

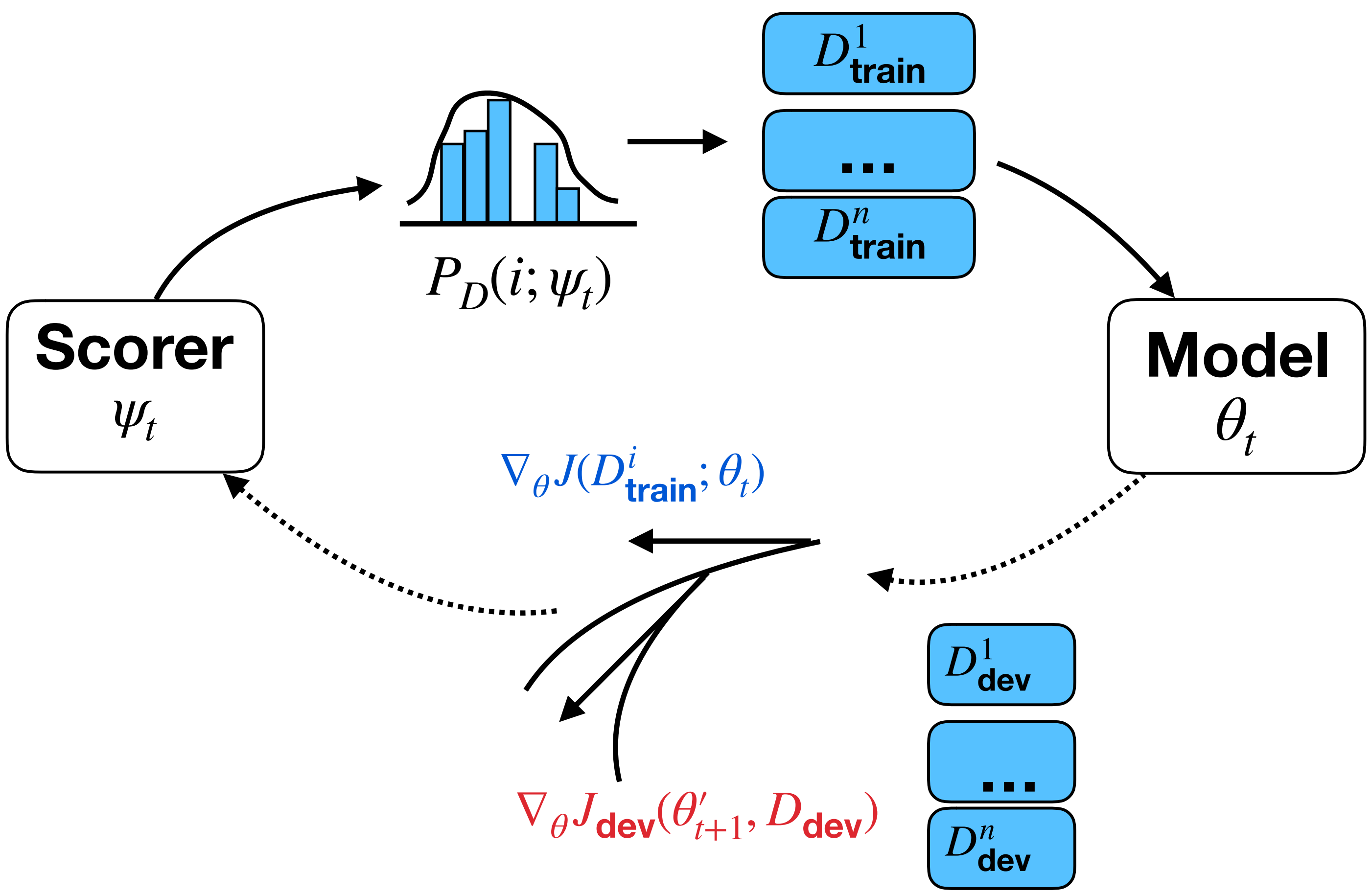
$$P_D(i) = \frac{|D^i_{\text{train}}|^{1/\tau}}{\sum_{k=1}^{n} |D^k_{\text{train}}|^{1/\tau}}$$

- How to use DDS?
  - Directly parameterize data scorer over the standard dataset sampling distribution

$$P_D(i; \psi) = e^{\psi_i} / \sum_{k=1}^{n} e^{\psi_k}$$

  - Optimize over the **multilingual dev set**

MultiDDS



- Update Model

$$\theta_t \leftarrow \theta_{t-1} - \nabla_\theta \mathbb{E}_{i\sim P_D(i;\psi)}\left[\ell(D^i_{\text{train}};\theta)\right]$$

- Update Scorer

$$\psi_{t+1} \leftarrow \psi_t + \nabla_\psi R(i;\theta)\cdot \log P(i;\psi)$$

Effect of D^i_{train} on all languages

$$R(i;\theta_t) \approx \cos\left(\underbrace{\frac{1}{n}\sum_{k=1}^{n}\nabla_\theta J(\theta_t, D^k_{\text{dev}})}_{\text{multilingual dev gradient}}, \underbrace{\nabla_\theta J(\theta_{t-1}, D^i_{\text{train}})}_{\text{train gradient}}\right)$$

Stabilizing the Reward

$$R(i, \theta) = \textbf{cos}\left(\frac{1}{n}\sum_{k=1}^{n} \nabla_\theta J(\theta_t, D^k_{\textbf{dev}}), \nabla_\theta J(\theta_{t-1}, D^i_{\textbf{train}})\right)$$

- Aggregate dev gradient, then calculate cosine alignment
  - The reward to update scorer has large variance when number of dev sets is large

$$R(i, \theta) \approx \frac{1}{n}\sum_{k=1}^{n} \textbf{cos}\left(\nabla_\theta J(\theta_t, D^k_{\textbf{dev}}), \nabla_\theta J(\theta_{t-1}, D^i_{\textbf{train}})\right)$$

- **MultiDDS-S**: trick to stabilize the reward
  - Calculate cosine distance for each dev set, then aggregate the alignment

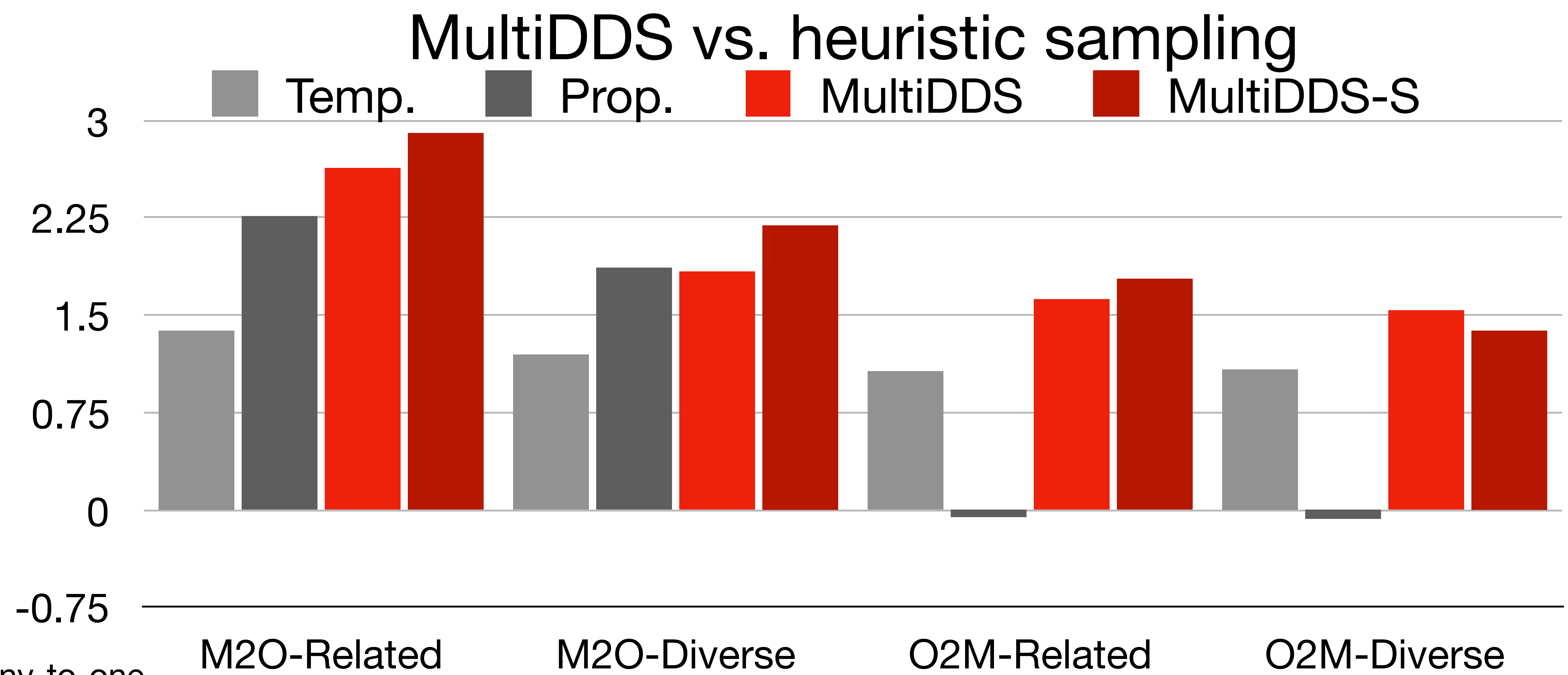
Avoids high variance in aggregated gradient

Gradients of different languages won't cancel out

Experiment Setup

- Dataset: Multilingual TED Talks (Qi et al. 2018)
- Two sets of languages
  - Related: 4 LRLs (Azerbaijani: **aze**, Belarusian: **bel**, Glacian: **glg**, Slovak: **slk**) and a related HRL for each LRL (Turkish: **tur**, Russian: **rus**, Portuguese: **por**, Czech: **ces**)
  - Diverse: picked without consideration for relatedness (Bosnian: **bos**, Marathi: **mar**, Hindi: **hin**, Macedonian: **mkd**, Greek: **ell**, Bulgarian: **bul**, French: **fra**, Korean: **kor**)
- Two NMT settings
  - Many-to-One (M2O)
  - One-to-Many (O2M)

Main Results



M2O: many-to-one
O2M: one-to-many

- Baselines: there is no consistently strong strategy
- MultiDDS consistently outperforms the baseline in all settings

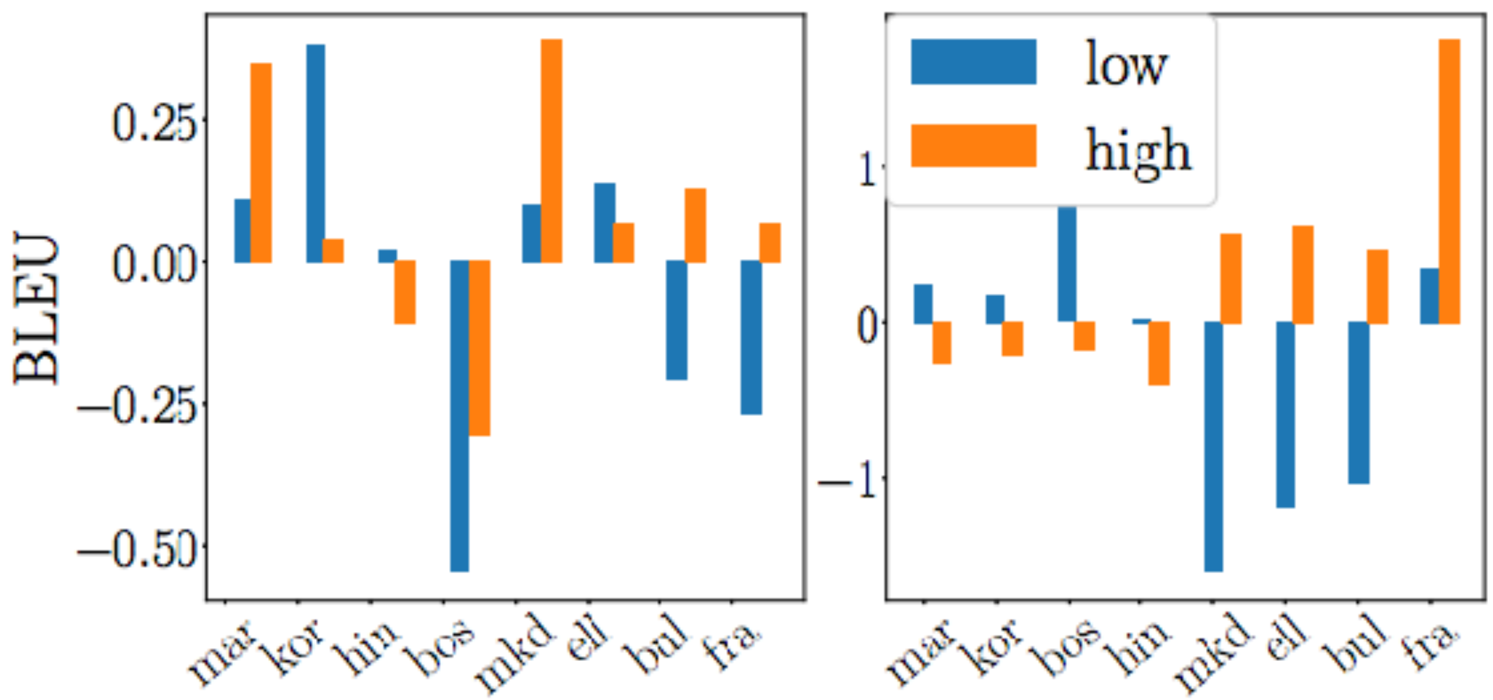
Figure plots performance difference from Uniform sampling

Prioritizing What to Optimize

- Prior work only focused on average performance
- What if we care about certain languages more?
- Fine-tune after 10 epochs using different aggregation methods
    - **Regular**: average performance
    - **Low** (egalitarian system): prioritize low-performing languages
    - **High** (specialized system): prioritize high-performing languages

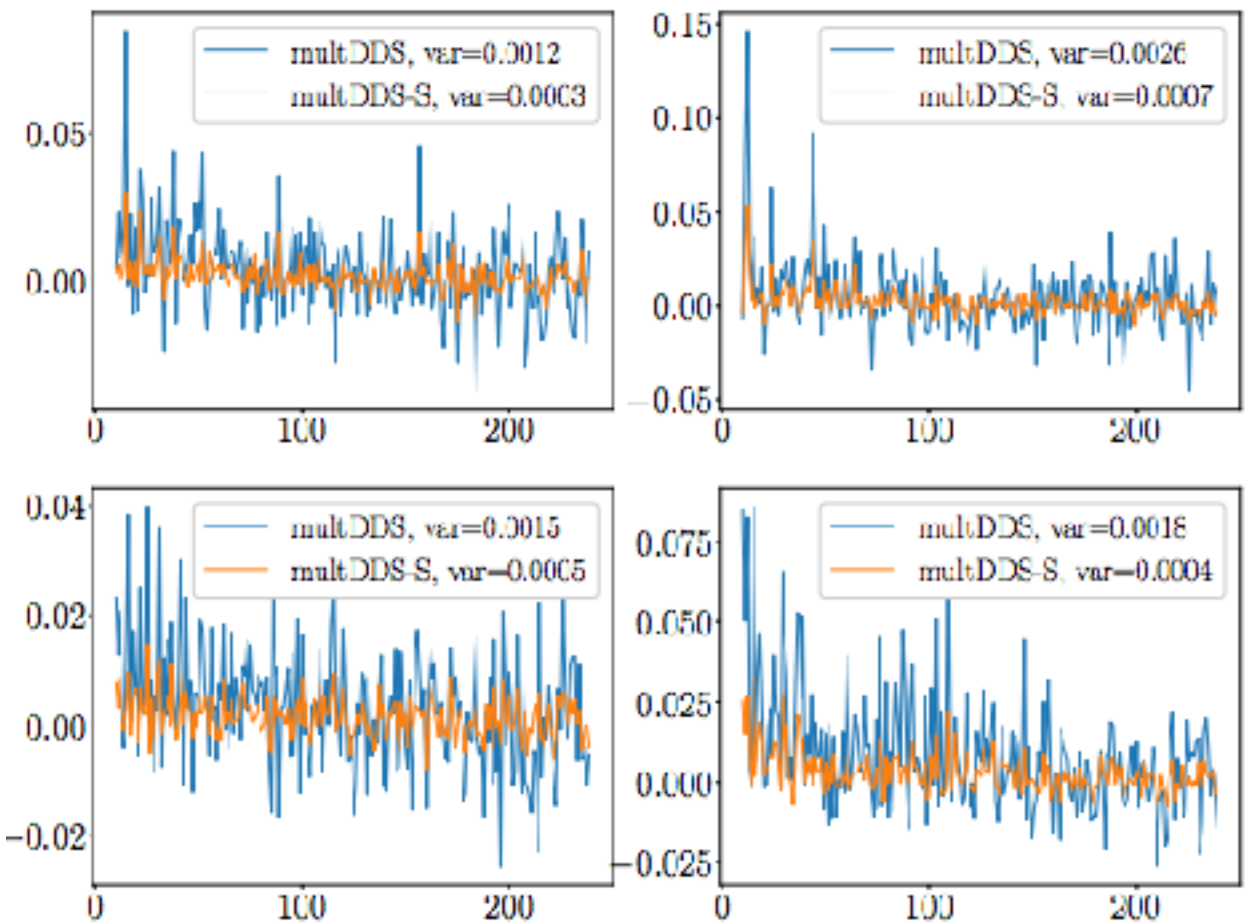
Prioritizing What to Optimize

Setting	Baseline	MultiDDS-S		
		Regular	Low	High
M2O	26.68	27.00	26.97	27.08
O2M	17.94	18.24	17.95	18.55



- MultiDDS of three different priorities always outperform the baseline in terms of average BLEU
- MultiDDS successfully optimizes for different priorities

Effect of Stabilized Reward



Method	M2O Mean	M2O Var.	O2M Mean	O2M Var.
MultiDDS	26.85	0.04	18.20	0.05
MultiDDS-S	26.94	0.02	18.24	0.02

- Reward of MultiDDS-S has less variance
- MultiDDS-S leads to smaller variance in model performance

Future Directions

- Extend to **other multilingual tasks** other than NMT
- Clearly define and experiment with **other multilingual optimization objectives** other than average performance

Thanks for listening!
Additional questions can be emailed to xinyiw1@cs.cmu.edu
Link to code: https://github.com/cindyxinyiwang/fairseq/tree/multiDDS