Optimizing Data Usage via Differentiable Rewards

Xinyi Wang*, Hieu Pham*, Paul Michel, Antonios Anastasopoulos, Jaime Carbonell, Graham Neubig

***: equal contribution**

Jaime G. Carbonell (1953 - 2020)

REMEMBERING OUR CO-AUTHOR

Motivation

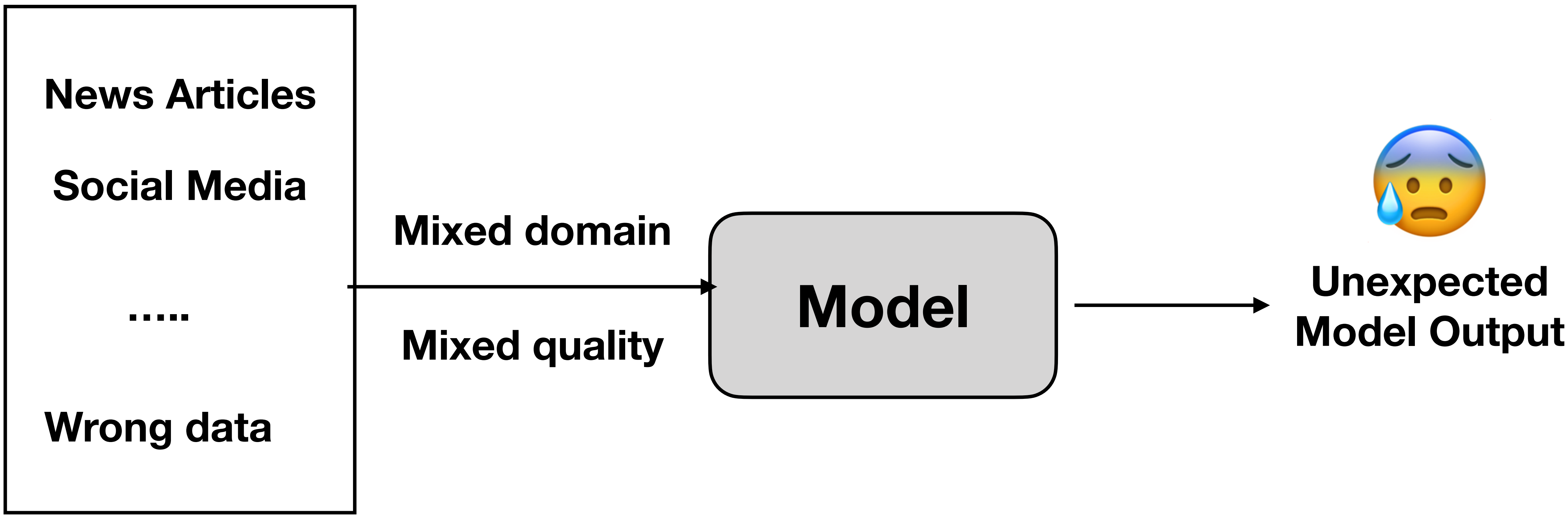
- Mismatch in training data distribution and real distribution:

We want: $\theta* = argmin_\theta \mathbb{E}_{x,y \sim P(X,Y)}[\ell(x, y; \theta)]$

But we do: $\theta* = argmin_\theta \mathbb{E}_{x,y \sim Uniform(D_{train})}[\ell(x, y; \theta)]$

- Many things can go wrong in D_{train}

Motivation



- Deep learning models are sensitive to the domain/quality of training data

Existing methods

- Data filtering/curriculum learning based on **hand-designed heuristics**
- Learning the data usage schedule for **specific applications**
    - Noisy data for classification (Jiang et al.)
    - Learning curriculum for NMT (Kumar et al.)
- Teacher-student framework (Fan et al.)
    - Trains a teacher data selector for multiple runs based on student network's final dev set performance
    - Very **sparse/unstable feedback** at the end of training, requires **multiple training runs** to train the teacher

Learning to optimize data usage

Standard Training

Training Data $\xrightarrow{(x, y) \sim \textbf{Uniform}(D_{\textbf{train}})}$ $\mathscr{L}_{Normal} = \ell(x, y; \theta)$

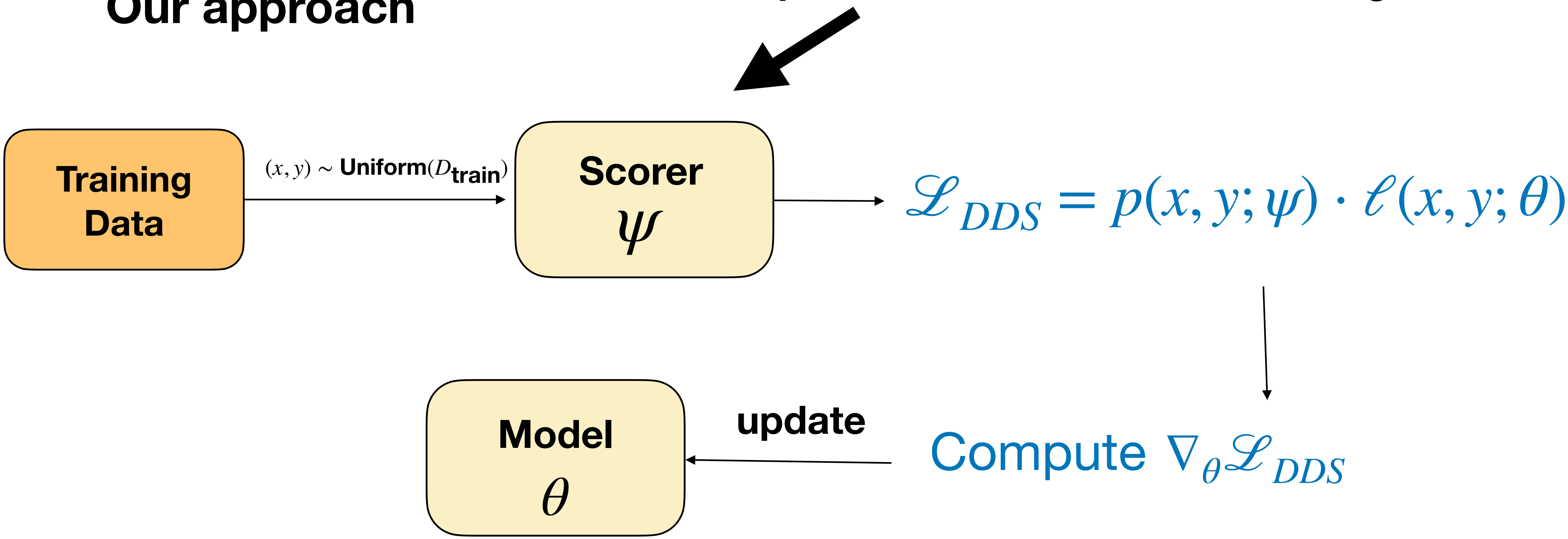
$\downarrow$

Model θ $\xleftarrow{\textbf{update}}$ Compute $\nabla_\theta \mathscr{L}_{Normal}$

Learning to optimize data usage

Simple/General Formulation:
- **Input: training data**
- **Output: distribution over the training data**

Our approach

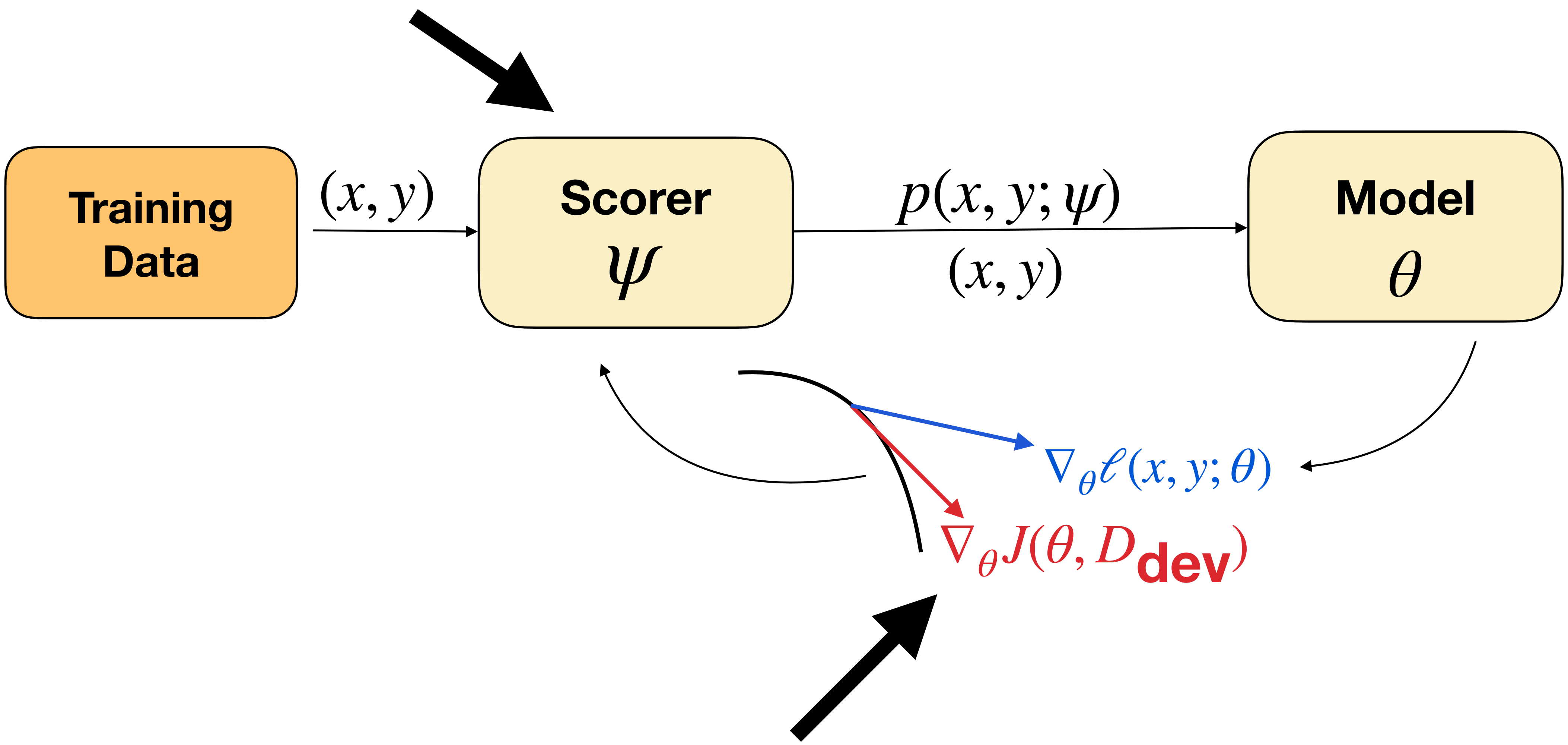


Hope: $p(x, y; \psi)$ makes the training data distribution closer to the real distribution

Differentiable Data Selection

Simple/General Formulation:

- **Input: training data**
- **Output: distribution over the training data**



Reward: gradient alignment with the dev data

Deriving the Rewards Via Direct Differentiation

- The gradient alignment reward can be derived as a solution of a bi-level optimization problem (Colson et. al.)

$$\theta^* = \text{argmin}_\theta \mathbb{E}_{x,y\sim P(X,Y;\psi)}[\ell(x,y;\theta)]$$

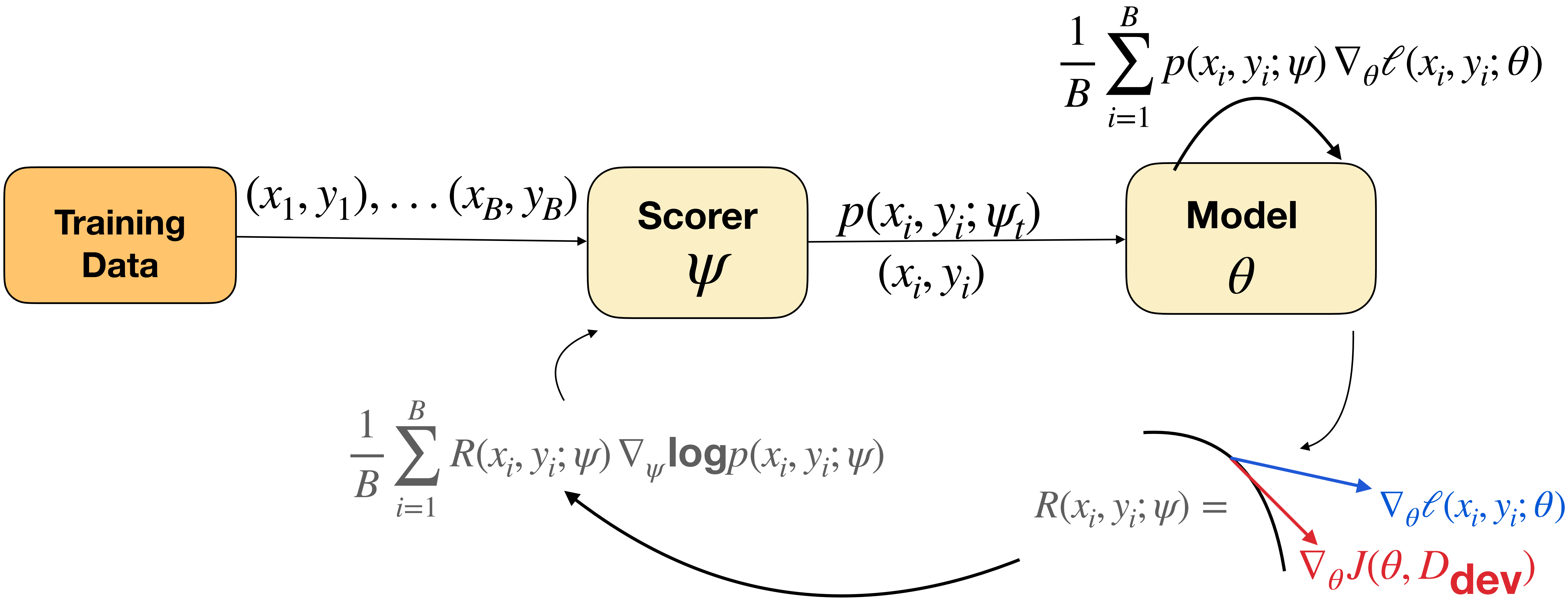
$$\psi^* = \text{argmin}_\psi J(\theta^*(\psi), D_{\text{dev}})$$

- Chain rule and Markov assumption

$$\nabla_\psi J(\theta_t, D_{\textbf{dev}}) \approx -\mathbb{E}_{x,y\sim P(X,Y;\psi)}[\underbrace{\nabla_\theta J(\theta_t, D_{\textbf{dev}})^\top \nabla_\theta \ell(x,y;\theta_{t-1}))}_{\text{gradient alignment}} \nabla_\psi \textbf{log} P(x,y;\psi)]$$

DDS for Image Classification

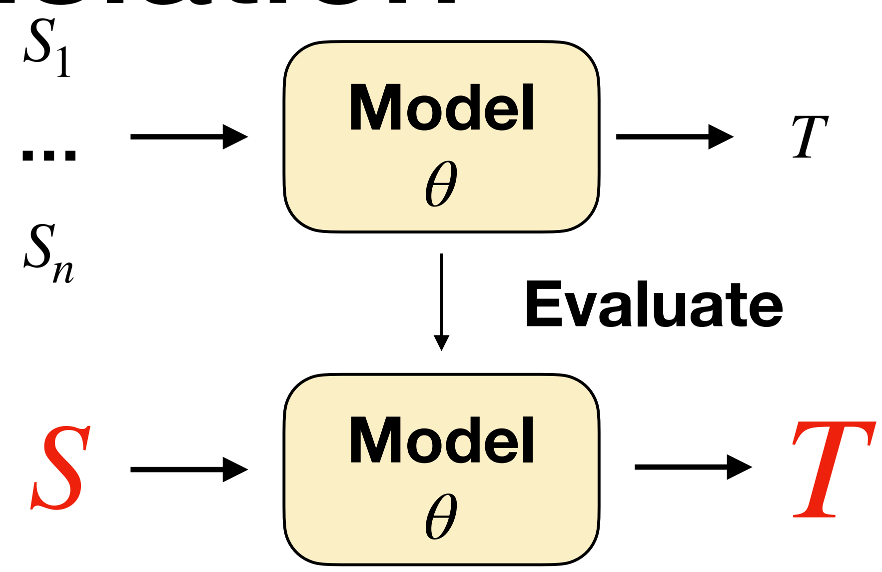
- Generic classification, applicable to a variety of tasks
- Given $D_{\textbf{train}}$, $D_{\textbf{dev}}$, find the optimal parameters θ^*
- For each training step



DDS for Multilingual Neural Machine Translation

- Given $D_{\textbf{train}} = (S_1 - T, \ldots, S_n - T)$

- find θ^* that translates from S to T where $D_{\textbf{dev}} = S - T$



- Several design choices for the specific problem
  - Scorer defined over training source languages
  - Directly sample data according to the scorer
  - Only update scorer once in a while during training

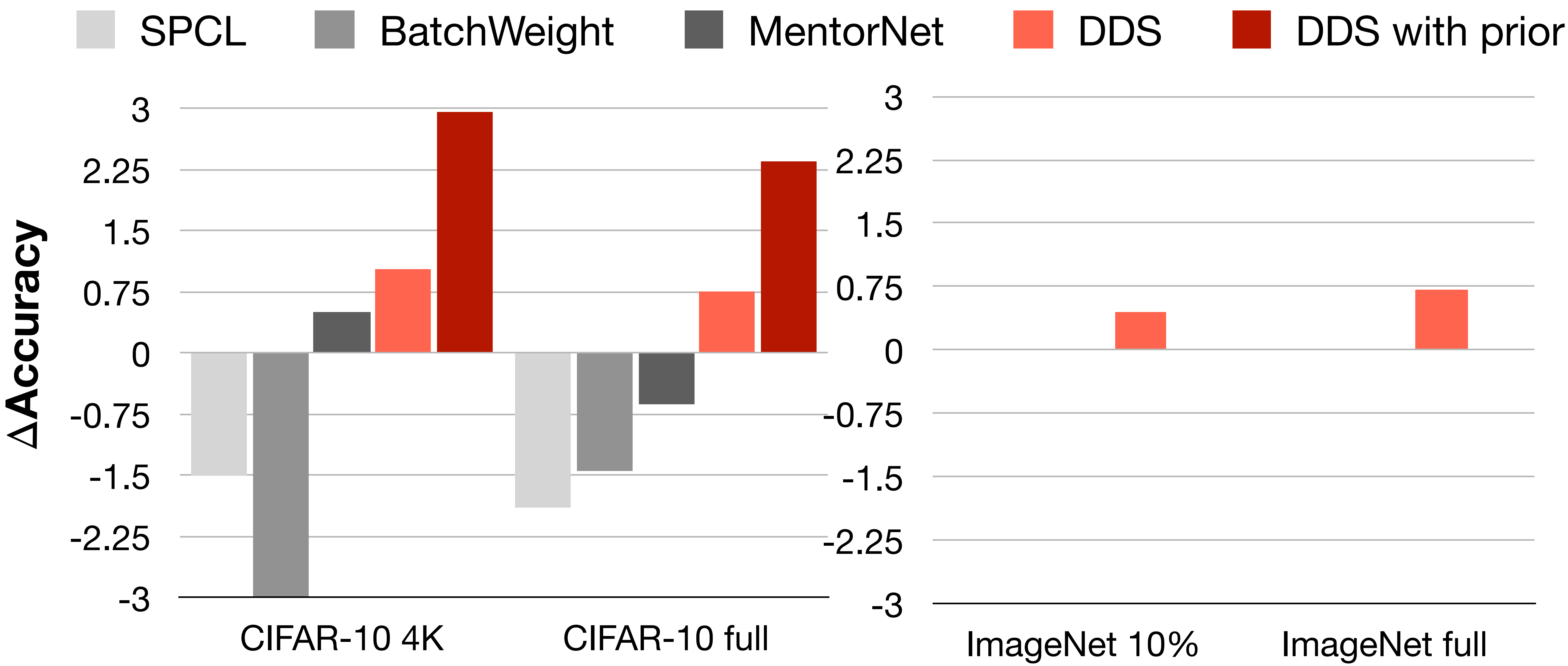
Dataset and Setup

- **Image Classification**
  - CIFAR10, ImageNet
  - First 10%, Full Dataset
- **Multilingual NMT**
  - 58-languages-to-English TED dataset
  - Train on 8 pairs of languages
    - Evaluate model on 4 low-resource languages (LRL) Azerbaijani (aze), Belarusian (bel), Galician (glg), and Slovak (slk)
    - The other 4 are their corresponding related high-resource languages (HRL) Turkish (tur), Russian (rus), Portugese (por), and Czech (ces)

Baselines and Ours

- **Baselines**
  - Uniform
  - SPCL (Jiang et al.): dynamically update the training curriculum
  - Other data selection methods
    - Classification: BatchWeight (Ren et al.), MentorNet (Jiang et al.)
    - NMT: Related (Neubig & Hu), TCS (Wang et al.)
- **Ours**
  - DDS
  - DDS with prior knowledge
    - Classification: Retrained DDS
    - NMT: TCS+DDS

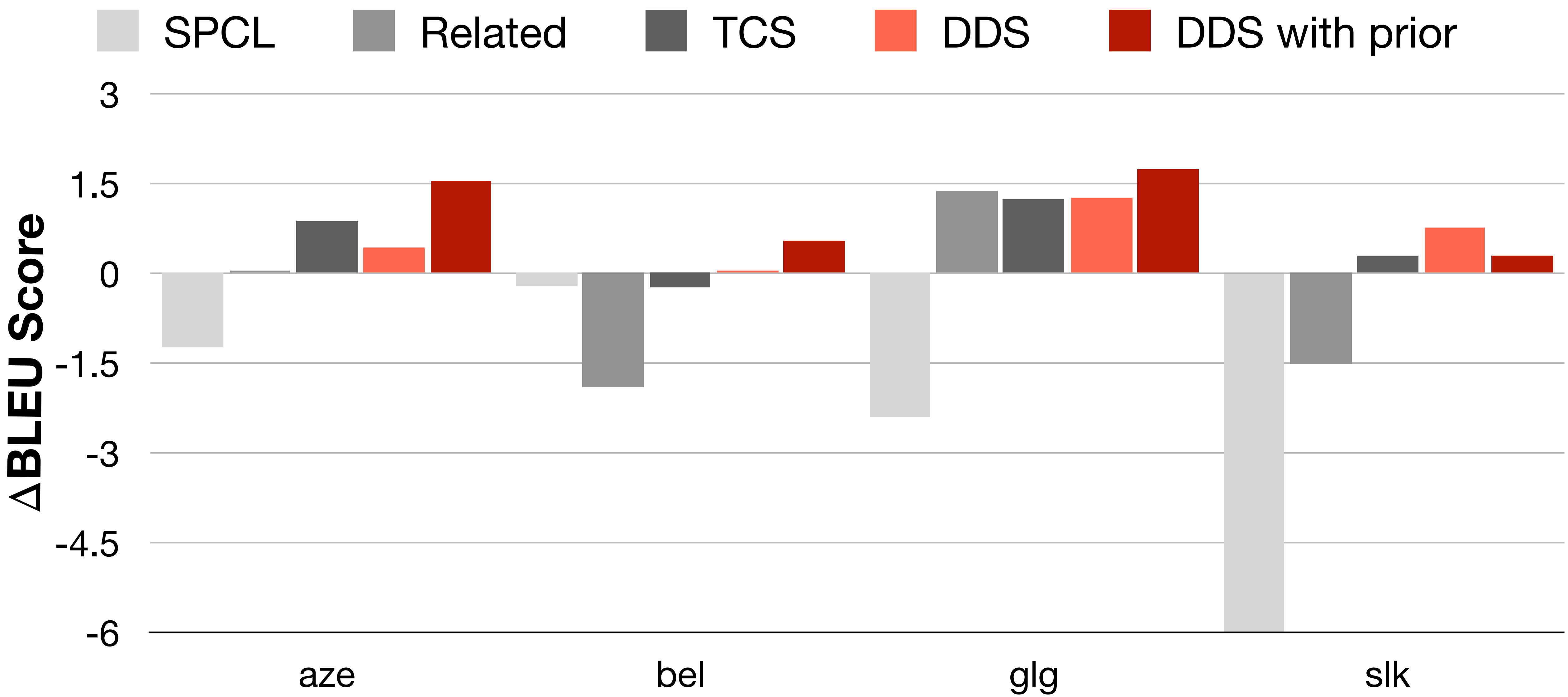
Results



- DDS performs the best of all strategies
- Adding a prior to DDS further improves

Figure: difference from Uniform sampling

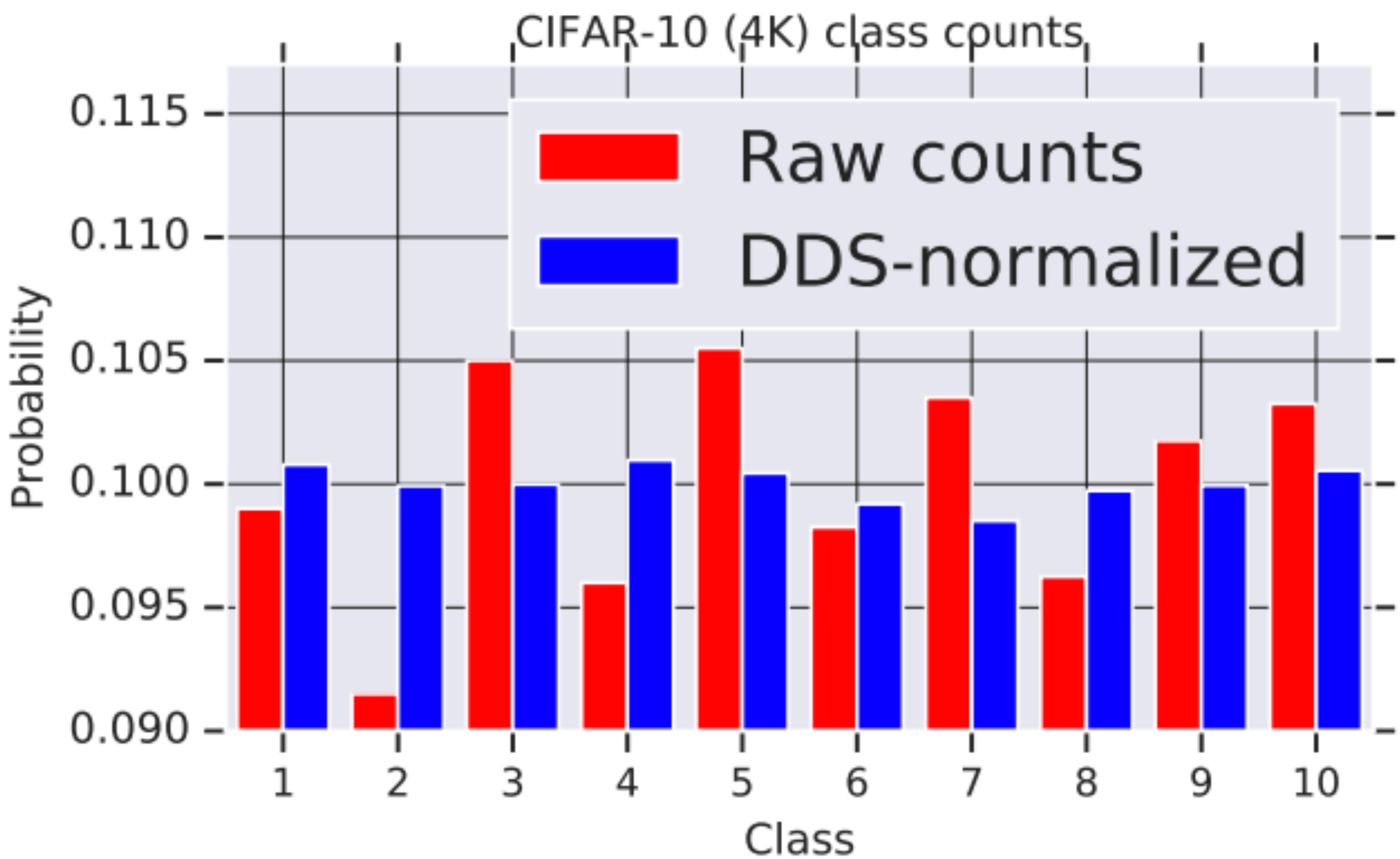
Results



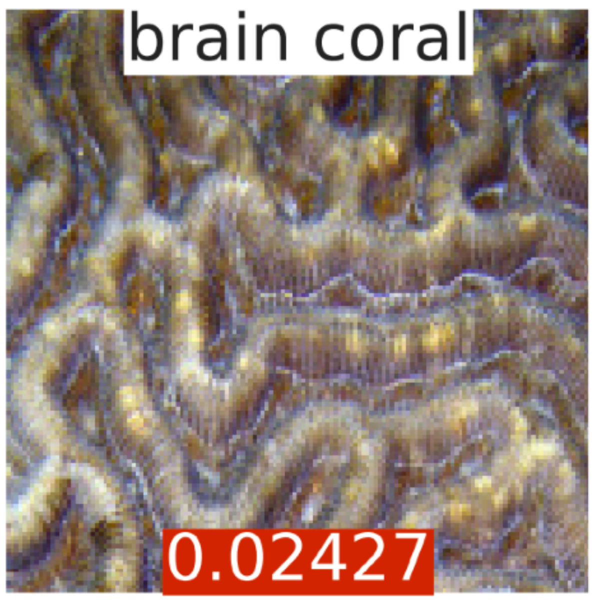
- SPCL is not competitive: ignores relevance to dev set
- DDS performs the best for all settings

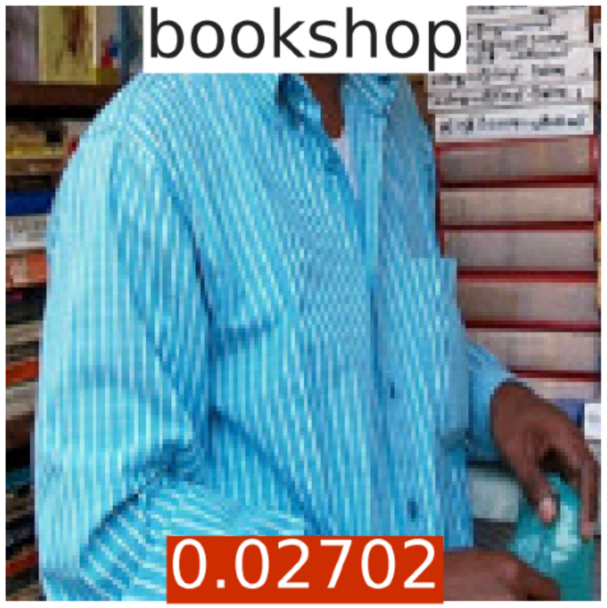
Figure: difference from Uniform sampling

Why does DDS work?: Learns to rebalance the class distribution

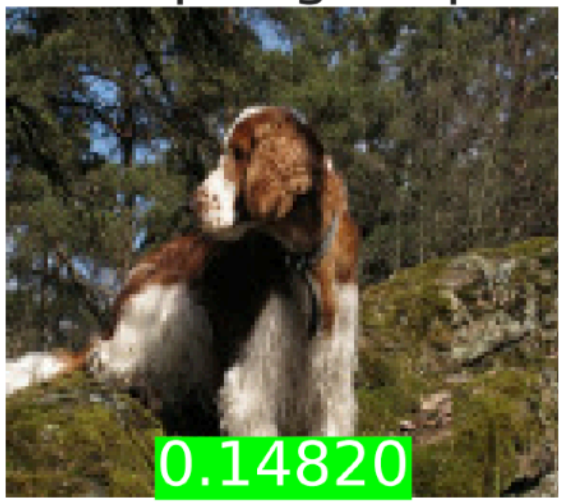


Why doe DDS work: Assigns higher scores to images with clearer content





Welsh springer spaniel



Why does DDS work: Learns to upweight the most related language

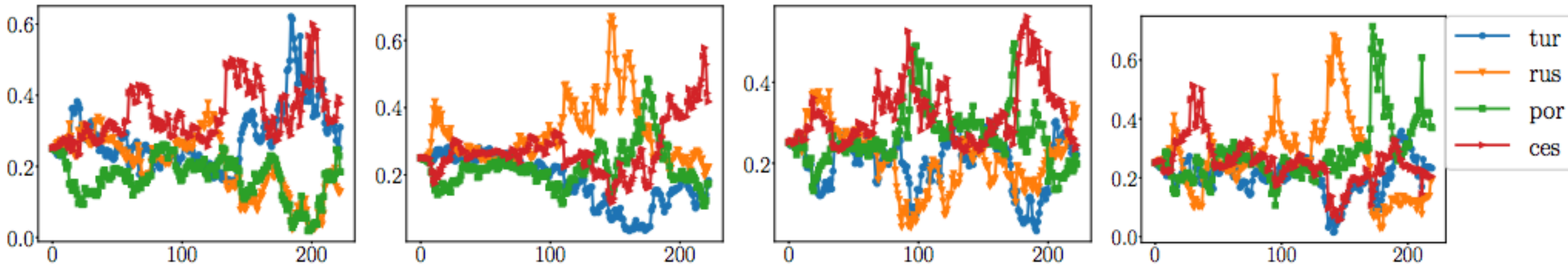


Figure 5: Language usage for DDS by training step. *From left to right*: `aze`, `bel`, `glg`, `slk`.

- Data distribution changes significantly over the course of training

Conclusion

- We present **Differentiable Data Selection**, which optimizes a data scorer network during training with **an intuitive reward function**
- Formulate two algorithms under DDS for **two realistic and very different tasks**
- DDS is a **flexible framework** that is potentially useful for many other tasks

Thanks for listening!

Questions can be emailed to: xinyiw1@cs.cmu.edu or hyhieu@cmu.edu