CS11-711 Advanced NLP

Multi-task, Multi-domain, and Multi-lingual Learning

Graham Neubig

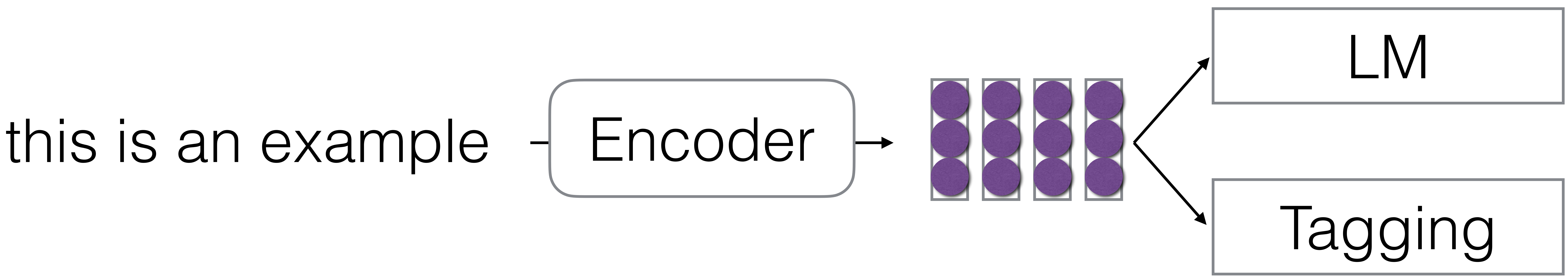
Carnegie Mellon University
Language Technologies Institute

Site
https://phontron.com/class/anlp2022/

Multi-task Learning

(Caruana 1997)

- Train representations to do well on multiple tasks at once



Applications of Multi-task Learning

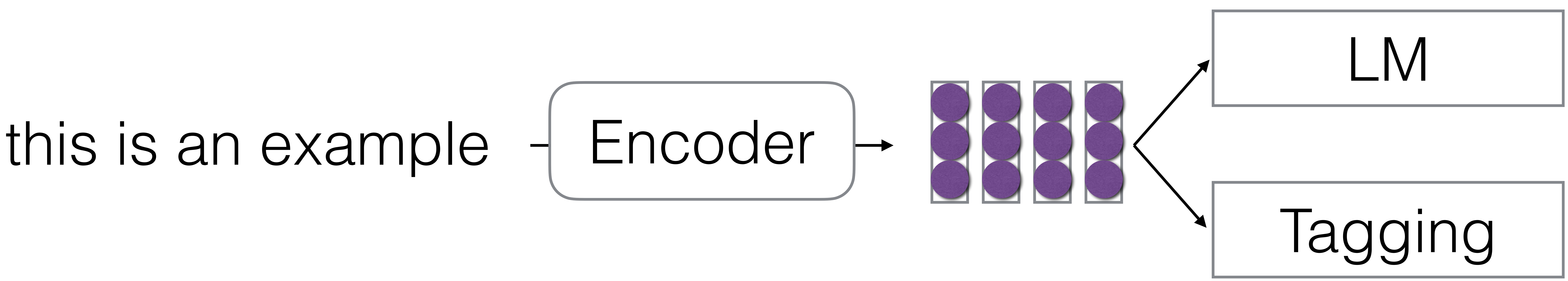
- Perform multi-tasking when one of your two tasks has fewer data

- **Plain text → labeled text**
  (e.g. LM -> parser)

- **General domain → specific domain**
  (e.g. web text → medical text)

- **High-resourced language → low-resourced language**
  (e.g. English → Telugu)

Advanced Multi-tasking Methodology

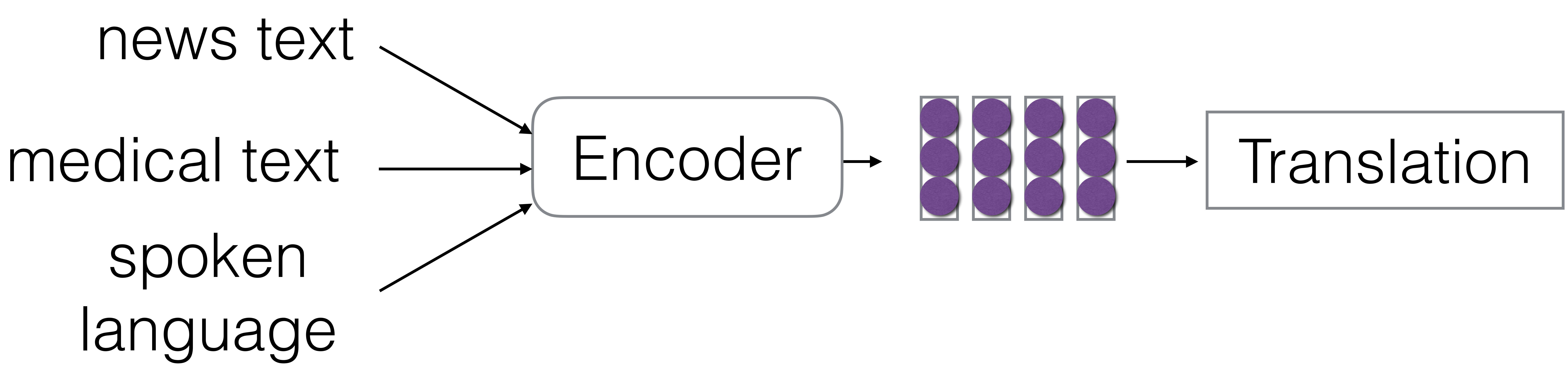


- What parameters do we update and how?
- How do we sample/weight our different tasks?

Domain Adaptation

Domains in NLP

- One task, but incoming data could be from very different distributions



- Sometimes domains are labeled, sometimes they are not

What's in a "Domain"

(Stewart 2019)

- Mathematically, joint distribution over inputs and outputs differs over domains 1 and 2

$$P_{d1}(X, Y) \neq P_{d2}(X, Y)$$

- In practice:
  - **Content**, what is being discussed
  - **Style**, the way in which it is being discussed
  - **Labeling Standards,** the way that the same data is labeled

Types of Domain Shift

- **Covariate Shift:** The input changes but not the labeling

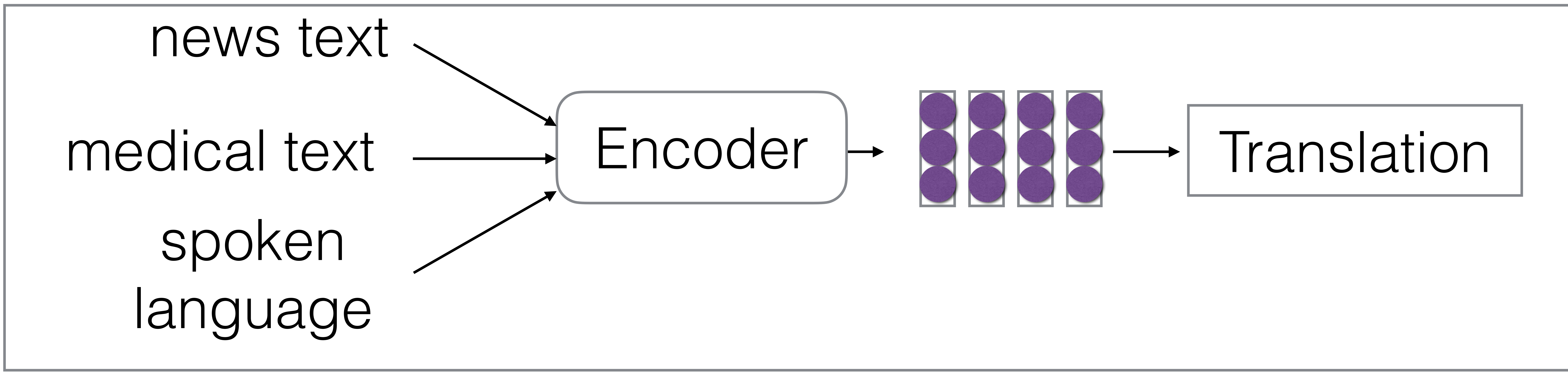
$$P_{d1}(X) \neq P_{d2}(X) \qquad P_{d1}(Y|X) = P_{d2}(Y|X)$$

- **Concept Shift:** The conditional distribution of labels changes (e.g. different labeling standards)

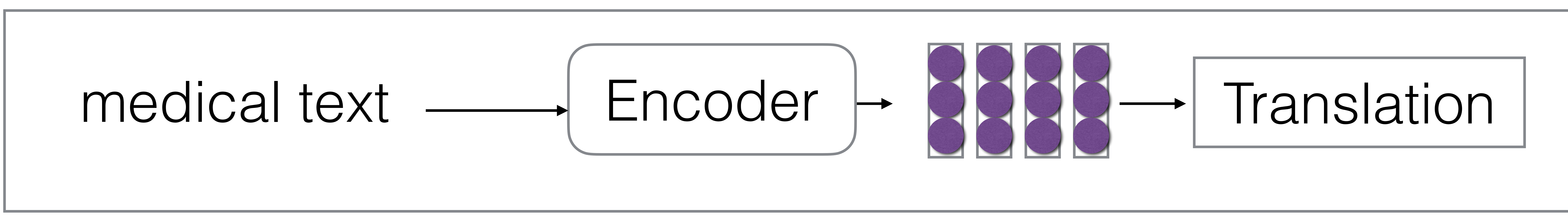
$$P_{d1}(Y|X) \neq P_{d2}(Y|X)$$

Domain Adaptation

- Train on many domains, or a high-resourced domain



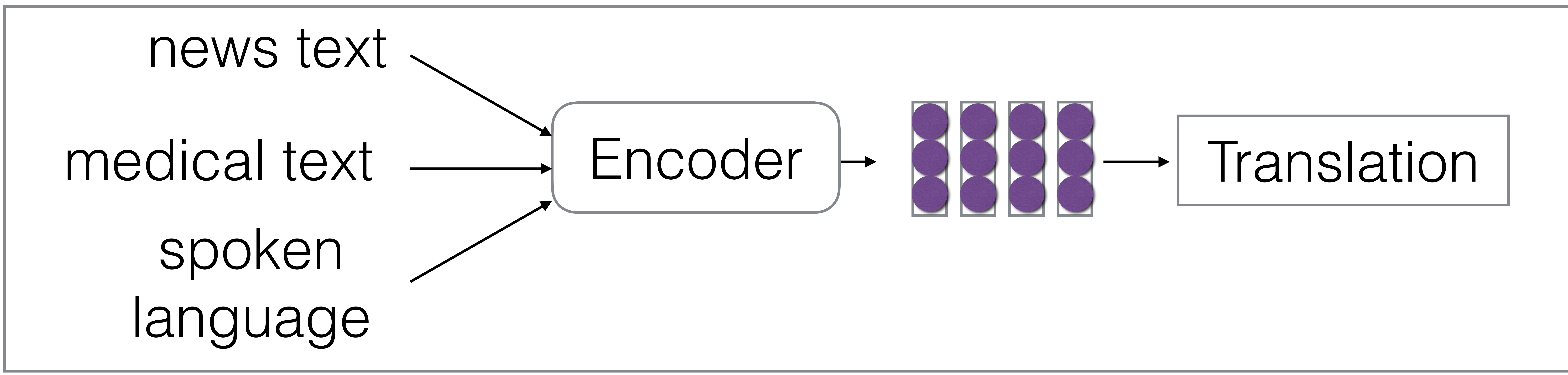
- Test on a low-resourced domain



- **Supervised** or **unsupervised** adaptation

Domain Robustness

- Train on many domains and do well on all of them



- Robustness to **minority domains**

- **Zero-shot** robustness to domains not in training data

Multilingual Learning

http://endangeredlanguages.com/

Similarity Across Languages

- Many languages share similar word roots

Cognates (joint origin)

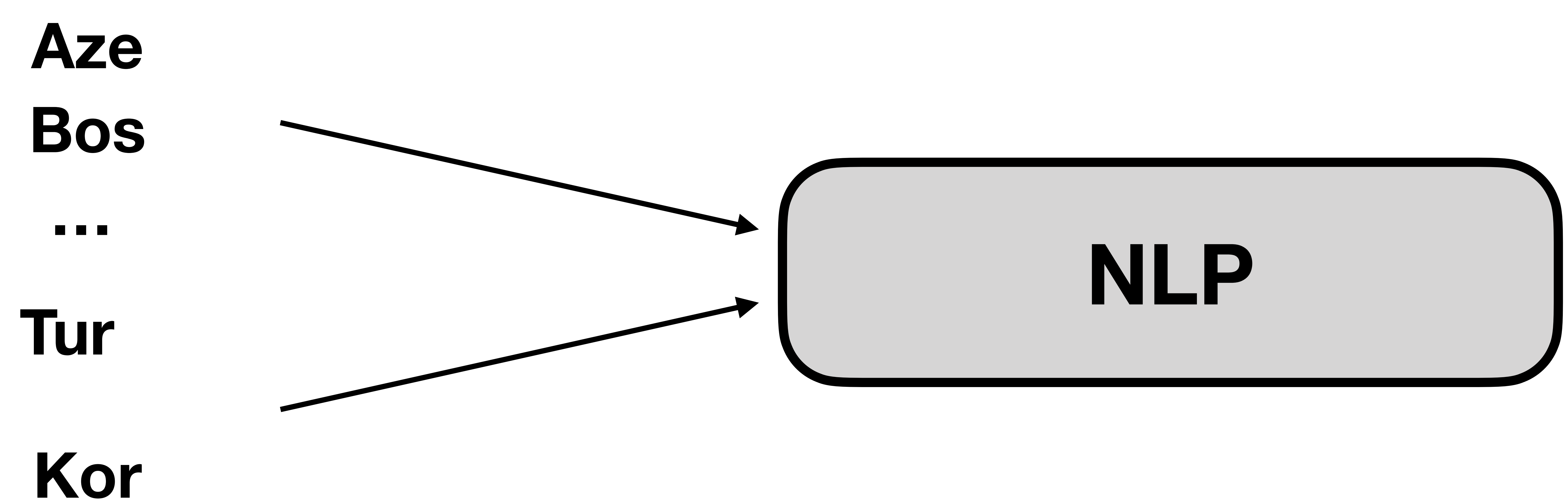
English:	night
French:	nuit
Russian:	noch
Bengali:	nishi

Loan Words (borrowed from another)

Arabic:	qahwa
Turkish:	kahveh
French:	coffee
Japanese:	kohi
Chinese:	kafei

- Languages share a considerable amount of underlying structure, e.g. word order, grammar.

Multilingual Training



Now our best tool for applying methods to low-resourced languages

Languages as Domains

- Multilingual learning is an extreme variety,
  different language = different domain
  - **Adaptation:** Improve accuracy on lower-resource languages by transferring from higher-resource languages
  - **Robustness:** Use one model for all languages, instead of one for each
- At the same time, much more complexity!
  → Requires modeling similarities/differences in lexicon, morphology, syntax, semantics, culture

Parameter Sharing Methods

How to Share Parameters?

- Share **all** parameters
  - e.g. single model for all domains
- Share **some** model components, not others
  - e.g. share encoder, separate decoder
- **Very small number** of unshared parameters
  - e.g. a single embedding specifying the domain

Full Parameter Sharing

- Ignore domain differences, just train a single model
  → Standard first step in multi-domain learning
- Also done multi-lingually
  - Multilingual MT into English (Neubig and Hu 2018)
  - Multi-lingual pre-trained LMs (Devlin et al. 2019, Wu and Dredze 2019)
- Cannot achieve ideal accuracy under **concept shift**

Simple Parameter Decoupling: Domain Tag

- Append a domain tag to input (Chu et al. 2017)

<news> news text

<med> medical text

- Translate into several languages by adding a tag about the target language (Johnson et al. 2017)

<fr> this is an example → ceci est un exemple

<ja> this is an example → これは例です

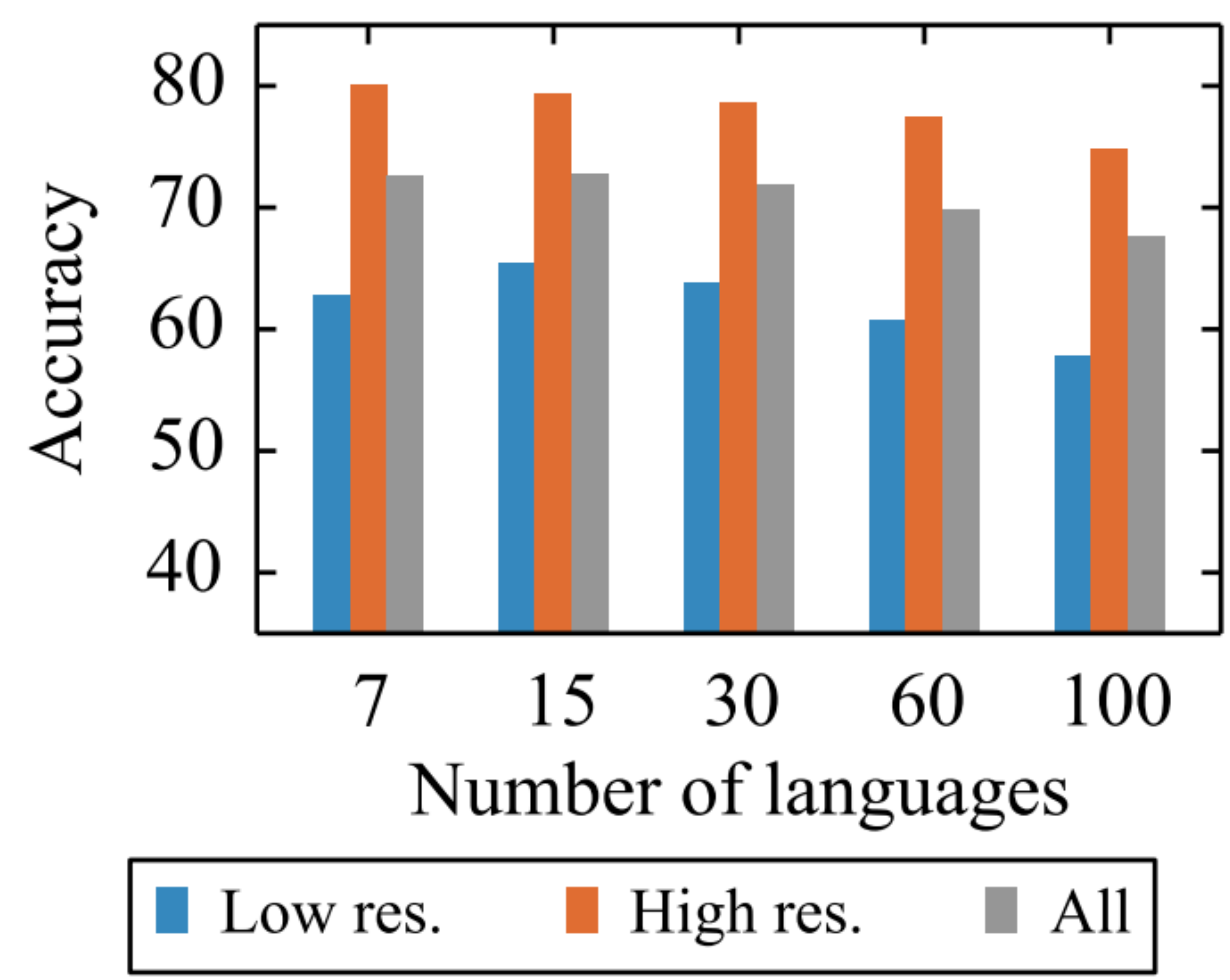
- Introduces a small number of parameters (=embedding size) for each domain

Minimal Parameter Decoupling Often Insufficient

- E.g. in multilingual learning

- In a fixed sized model, the per-language capacity decreases as we increase the number of languages

- Increasing the number of languages
—> decrease in the quality of all language accuracy (Conneau et al. 2019)

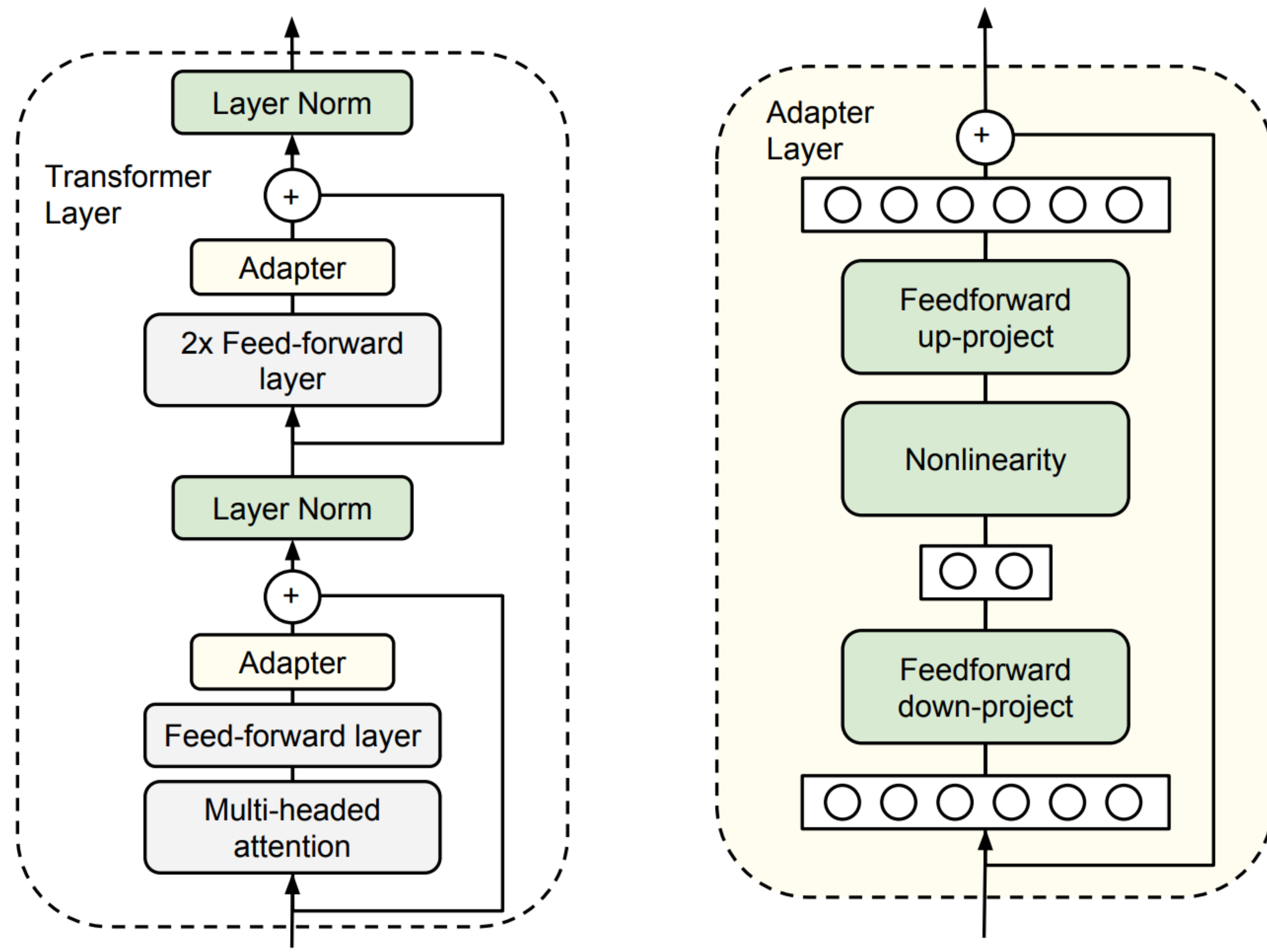


Aggressive Parameter Decoupling

- E.g. in multilingual MT, one encoder or decoder per language (Firat et al. 2016)
- Problems:
  - Can't share when languages/domains are legitimately similar
  - Explosion in number of parameters

Minimal Parameter Decoupling Example: Adapters

- Add a small layer per task to an already-trained model
- Transformer architecture example from Houlsby et al. (2019)



Different Types of Adapters

- Adapter: add between layers
- Prefix tuning: add learnable prefix
- LoRA: Learn low-rank approximation of weight matrix

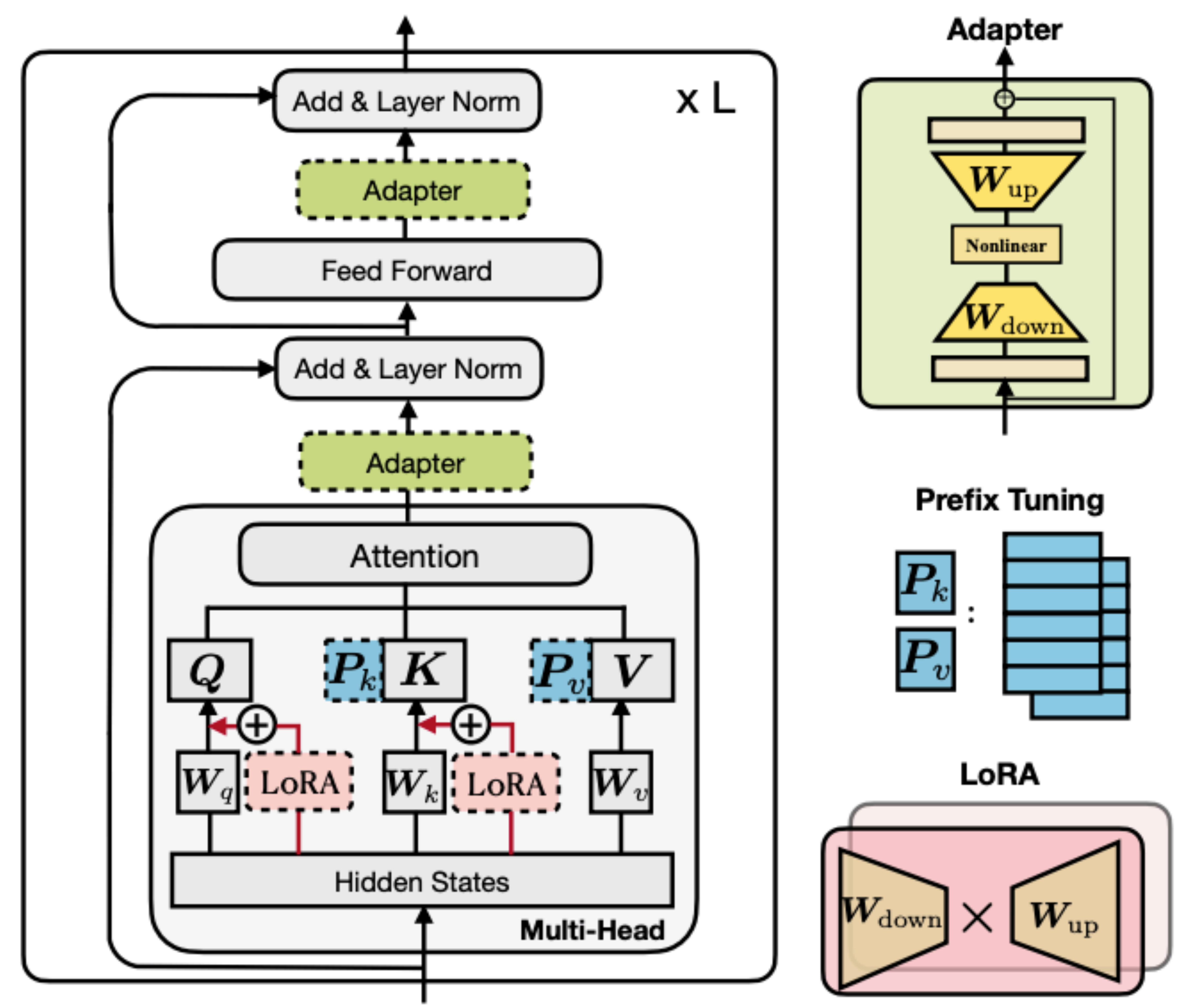


Figure 1: Illustration of the transformer architecture and several state-of-the-art parameter-efficient tuning methods. We use blocks with dashed borderlines to represent the added modules by those methods.

Regularization Methods for Adaptation (e.g. Barone et al. 2017)

- Pre-training relies on the fact that we won't move too far from the initialized values
- We need some form of regularization to ensure this
  - **Early stopping:** implicit regularization — stop when the model starts to overfit
  - **Explicit regularization:** L2 on difference from initial parameters

$$\theta_{adapt} = \theta_{pre} + \theta_{diff} \qquad \ell(\theta_{adapt}) = \sum_{\langle X,Y\rangle \in \langle \mathcal{X},\mathcal{Y}\rangle} -\log P(Y \mid X; \theta_{adapt}) + ||\theta_{diff}||$$

  - **Dropout:** Also implicit regularization, works pretty well

Soft Parameter Tying for Multi-task Learning

- It is also possible to share parameters loosely between various tasks
- Parameters are regularized to be closer, but not tied in a hard fashion (e.g. Duong et al. 2015)

Selective Parameter Adaptation

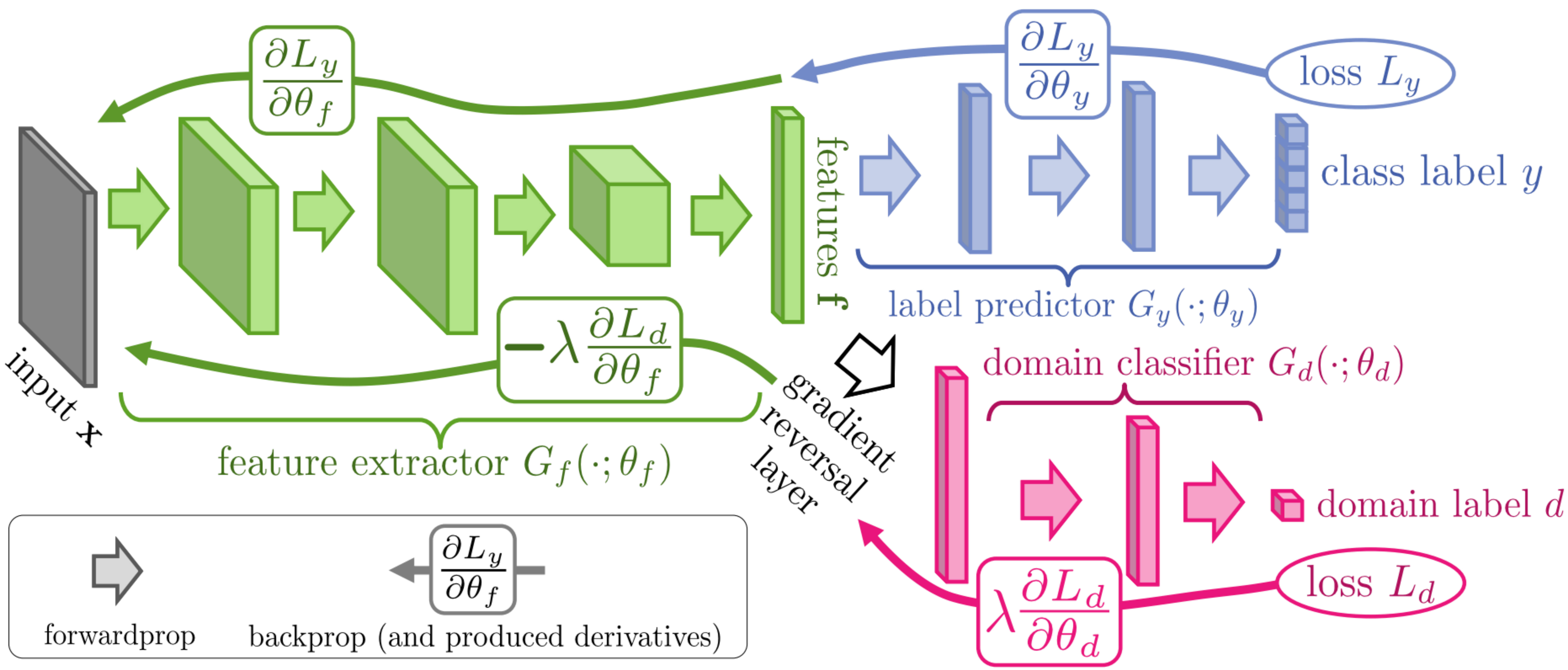
- Sometimes best to adapt subset of parameters
- e.g. cross-lingual transfer for neural MT (Zoph et al. 2016)

Setting	Dev BLEU	Dev PPL
No retraining	0.0	112.6
Retrain source embeddings	7.7	24.7
+ source RNN	11.8	17.0
+ target RNN	14.2	14.5
+ target attention	**15.0**	13.9
+ target input embeddings	14.7	**13.8**
+ target output embeddings	13.7	14.4

- Share sub-networks of the Transformer (Sachan and Neubig 2018)

Feature Space Regularization

- Try to regularize the features spaces learned to be closer to each-other (e.g. Ganin et al. 2016)



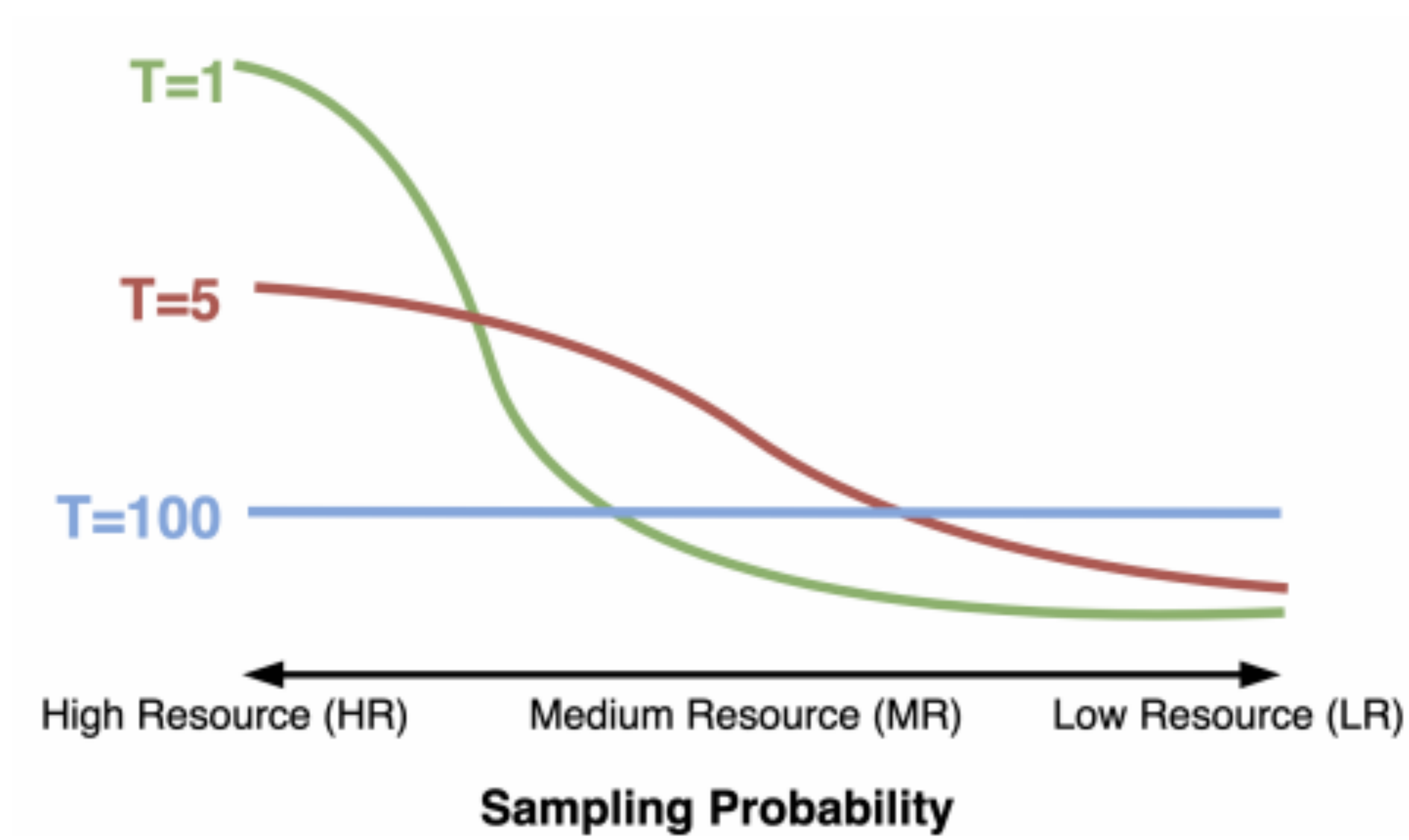
Task Weighting

Handling Different Tasks in Learning

- *How much* to learn on each task?
  - **Task Weighting:** Differently weight loss functions from different tasks
  - **Task Sampling:** Similar to weighting, modify sampling proportion
- *When* to learn on each task?
  - **Curriculum Learning:** Choose the ordering of tasks

Simple Task Weighting Strategies

- **Uniform:** Sample/weight all tasks with equal probability
- **Proportional:** Sample/weight tasks according to data size
- **Temperature-based:** Sample tasks according to data size exponentiated by 1/τ (Arivazhagan et al. 2019)

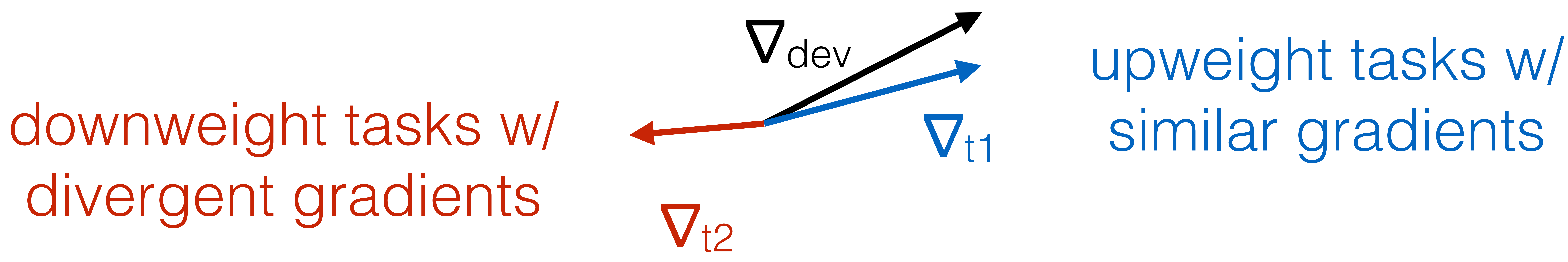


Data-driven Task Weighting

- **Loss Scaling:** Scale the loss according to variance w/ regularizer (Kendall et al. 2018)

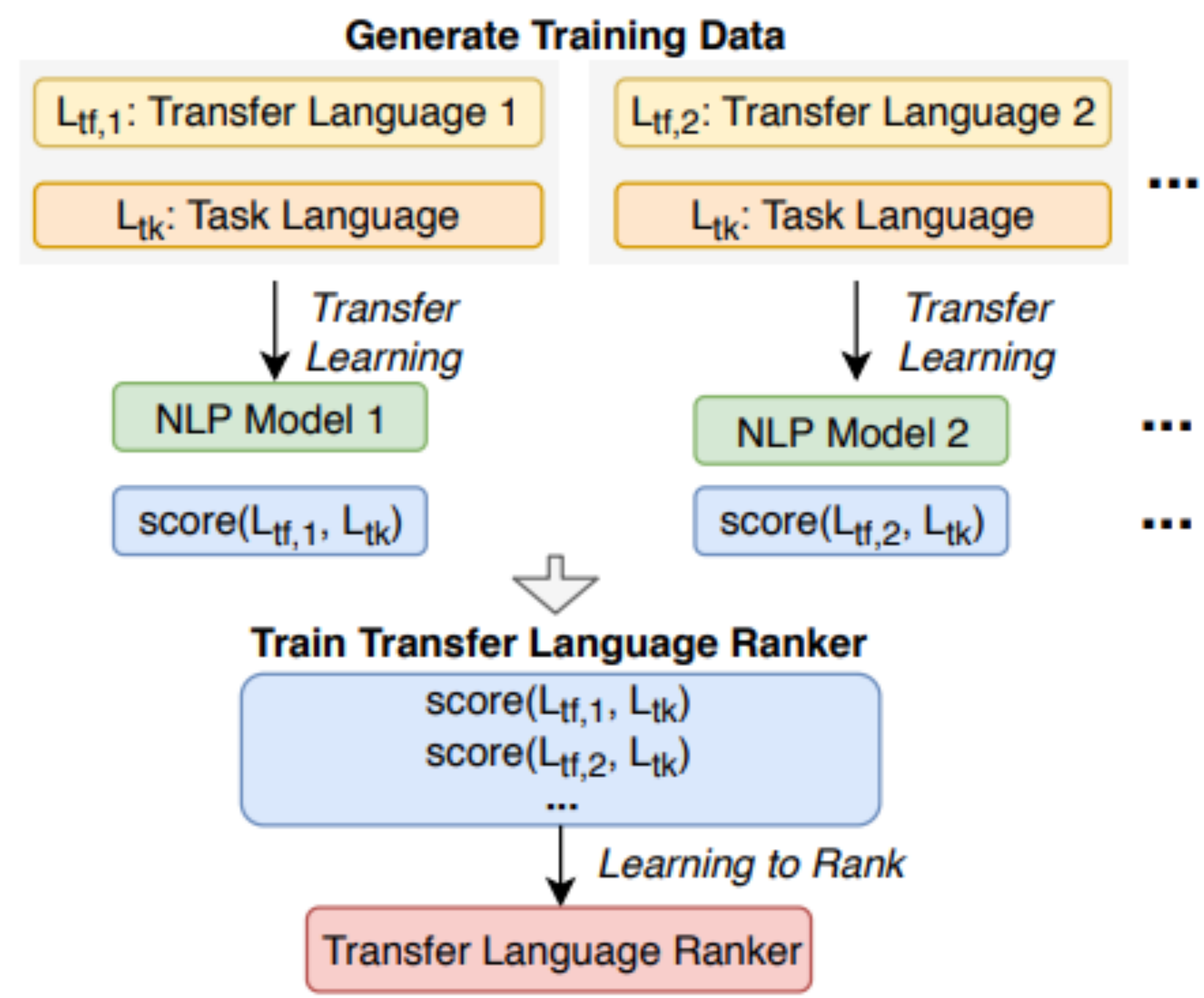
$$\mathcal{L}_{\text{total}} = \sum_{u} \frac{\mathcal{L}_i}{2\sigma_i} + \log \sigma_i$$

- **Task Weight Optimization:** Optimize weights of each task to improve accuracy on a development set (e.g. Dery et al. 2021)



Choosing Transfer Tasks

- We have many tasks that we could be choosing from!
- **Intuitive selection:** more similar task benefit more
- **Empirical selection:** run many transfer experiments and deduce rules
  - Choosing transfer languages (Lin et al. 2019)
  - Multi-task learning on one language (Vu et al. 2020)



Distributionally Robust Optimization

- We'd like to find find a model that does well over multiple domains

- **Distributionally robust optimization** optimizes the *worst-case* loss (loss on the worst task)

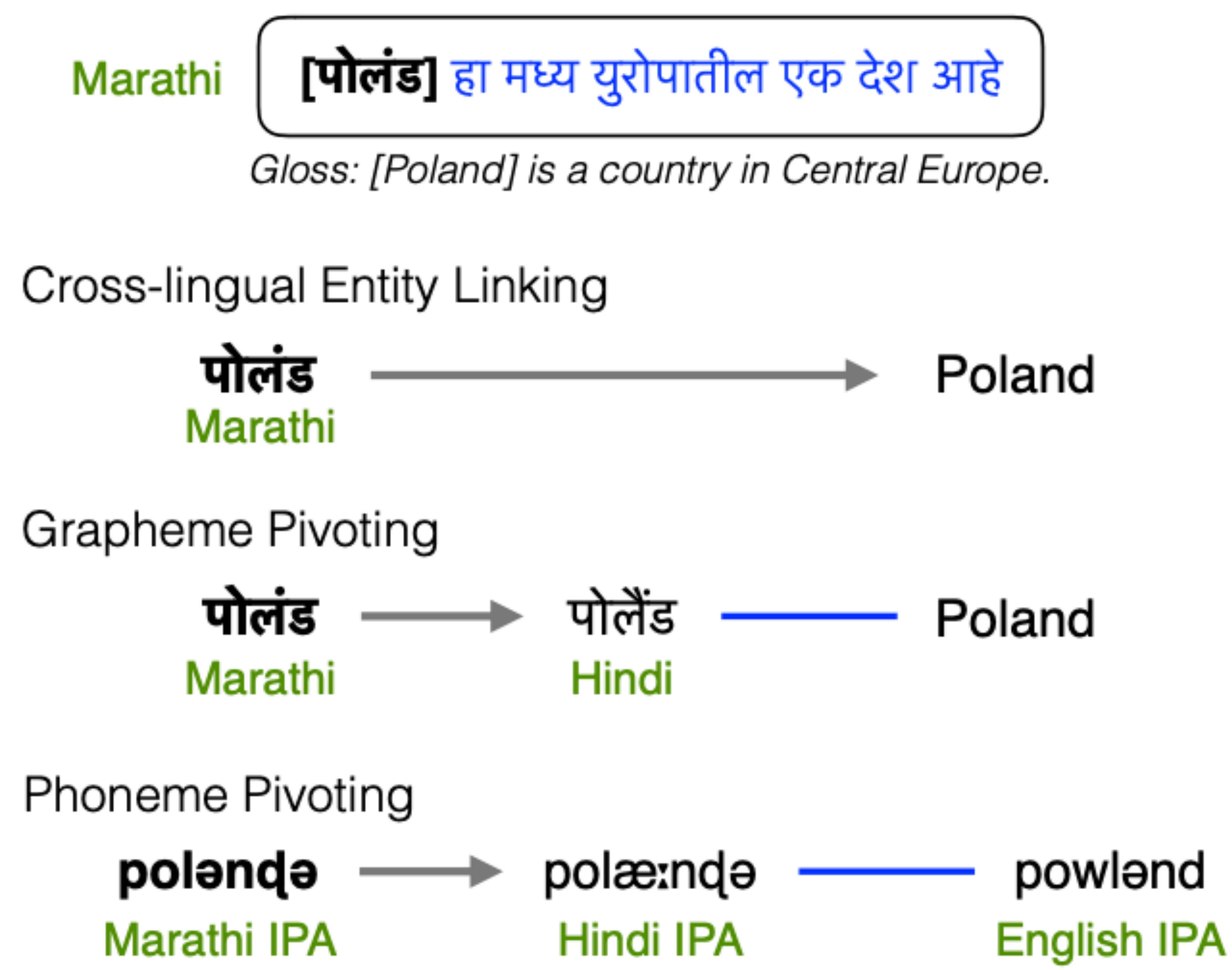
$$\mathcal{L} = \operatorname*{argmin}_{\theta} \max_{\tilde{\mathcal{L}}} \tilde{\mathcal{L}}(\theta)$$

- NLP applications to LM across domains (Oren et al. 2019) and MT across languages (Zhou et al. 2021)

Inherently Multilingual Considerations

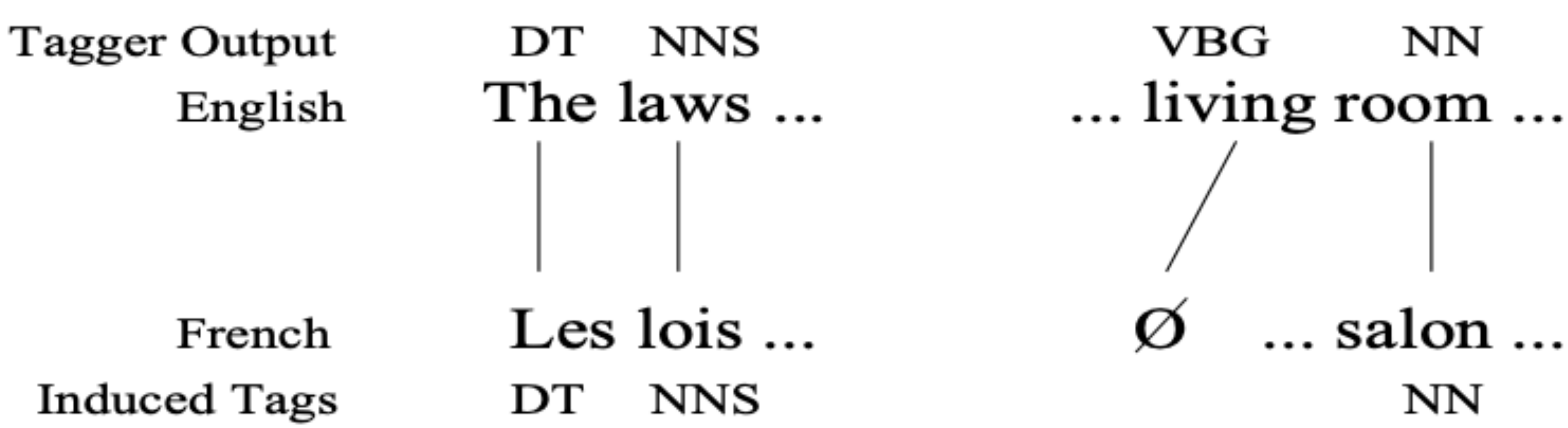
Handling Different Scripts

- Use phonological representations to make the similarity between languages apparent.
- E.g. Rijhwani et al (2019) link between entities in different languages in pronunciation space



Using Parallel Data

- Often we have translations in multiple languages
- Annotation projection: induce annotations in the target language using parallel data or bilingual dictionary (Yarowsky et al, 2001).



Questions?