Two Lessons I have recently learned
(1) Parallel Decoding of a Sequence
(2) Meta-Learning

Kyunghyun Cho

NYU Courant Institute (Computer Science) and Center for Data Science

Facebook AI Research

Parallel Decoding of a Sequence

Non-autoregressive neural sequence modeling

Jason Lee, **Elman Mansimov** and Kyunghyun Cho. Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement. 2018 (under review)

Neural sequence modeling

- An arbitrary input X: e.g., another sequence, image, video, …
- A sequence output $Y = (y_1, y_2, \ldots, y_T)$
  - e.g., natural language sentence
  - Discrete $y_t \in V$
- Use a neural network to estimate a distribution over sequences

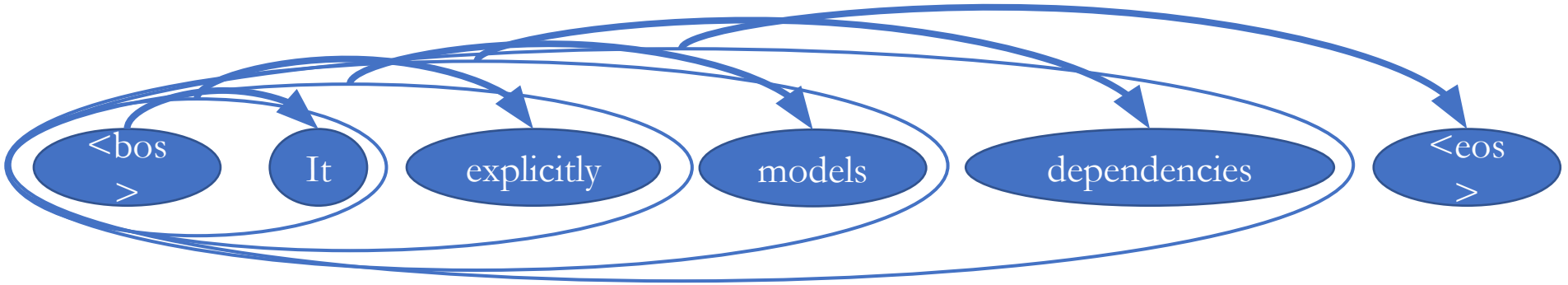
$$\log p_\theta(Y|X)$$

- Machine translation, automatic speech recognition, …

Neural autoregressive sequence modeling

- Unlike classification, complex, strong dependencies among y_t's
  - More than half of residents in Korea speak _____.
  - Among millions of possible tokens, only one word (Korean) is likely above.
- Neural autoregressive sequence modelling

$$\log p(Y|X) = \sum_{t=1}^{T} \log p(y_t|y_{<t}, X)$$

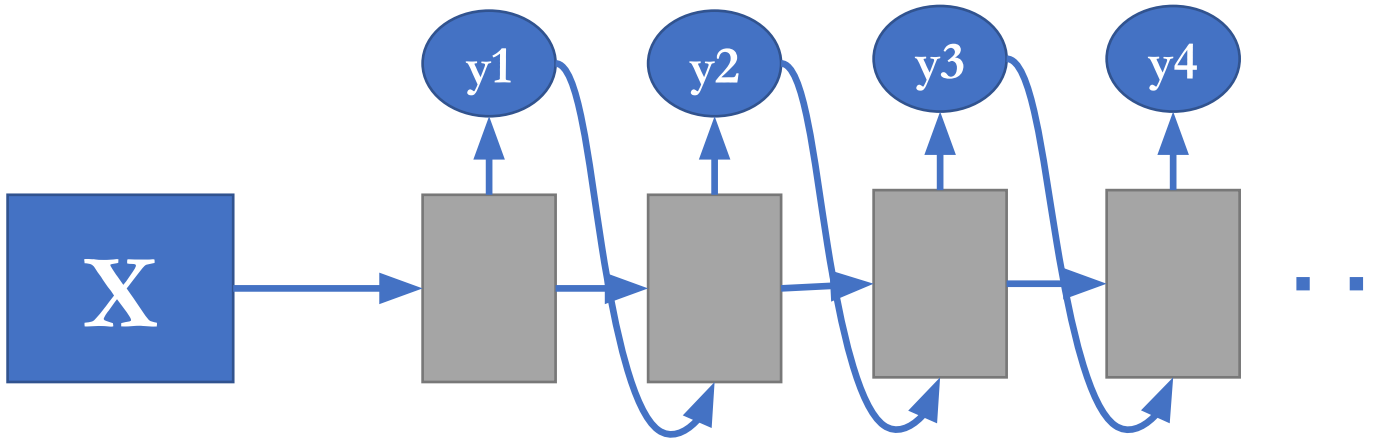


Neural autoregressive sequence modeling

- Decoding is problematic

1. Exact decoding is intractable $\arg\max_Y \sum_{t=1}^{T} \log p(y_t|y_{<t}, X) = ?$

2. Decoding is inherently sequential $O(kT)$

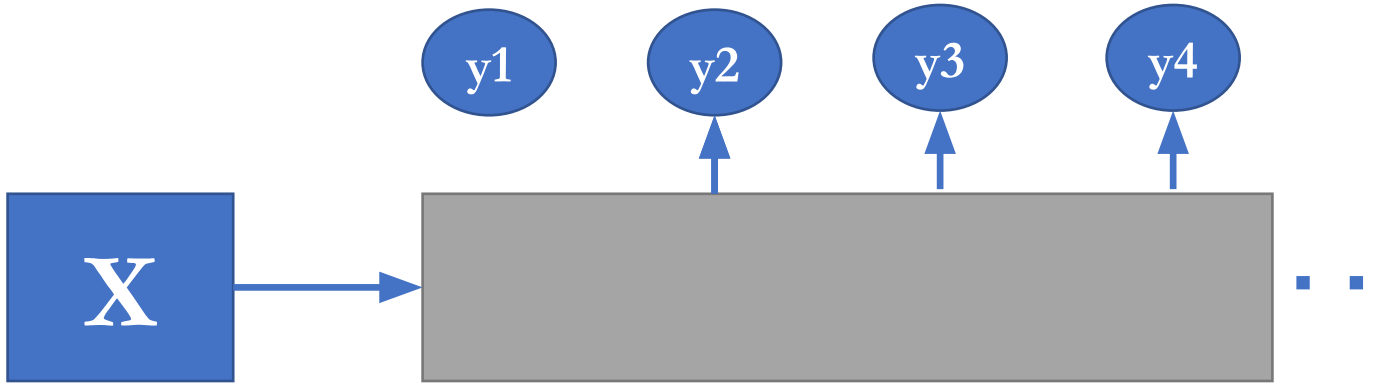


Neural **non**-autoregressive sequence modeling

- Conditional independence among y_t's

$$\log p(Y|X) = \sum_{t=1}^{T} \log p(y_t|y_{<t}, X) \Rightarrow \log p(Y|X) = \sum_{t=1}^{T} \log p(y_t|X)$$

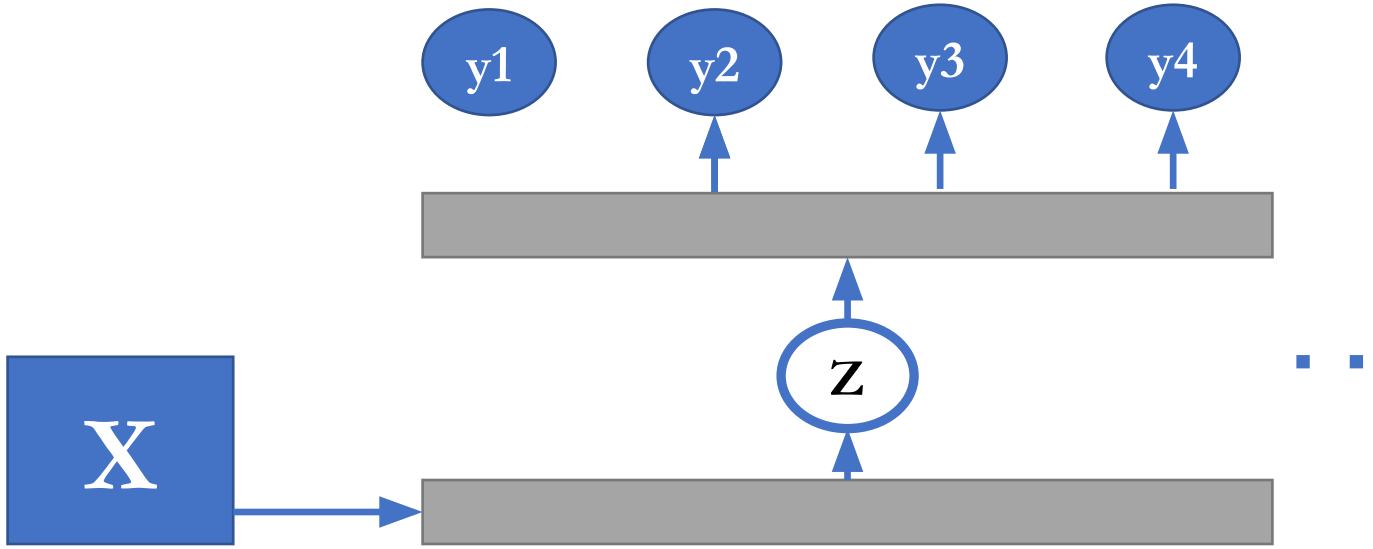
- Exact decoding is tractable $\hat{y}_t = \arg\max_{y_t} \log p(y_t|X)$
- Decoding is highly parallelizable



Neural **non**-autoregressive sequence modeling

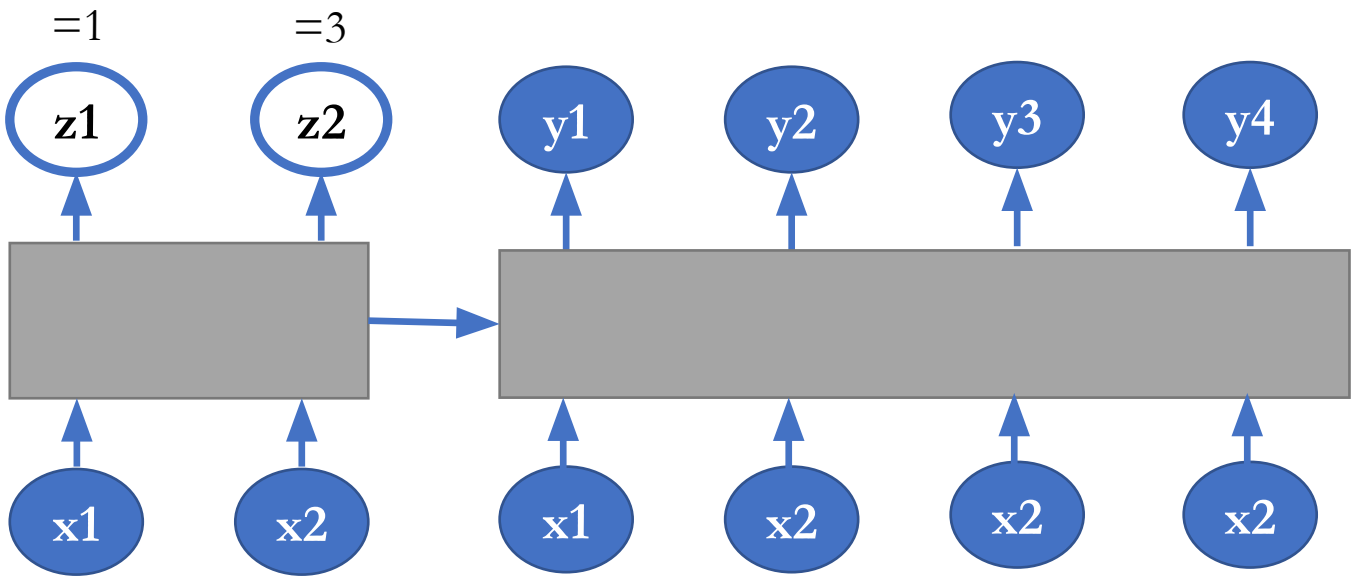
- *Too good to be true*: dependencies must be modelled somehow
- Introduce a set of latent variables [Gu et al., 2018 ICLR]

$$\log p(Y|X) = \sum_{t=1}^{T} \log \sum_{Z} p(y_t|Z, X)p(Z|X)$$



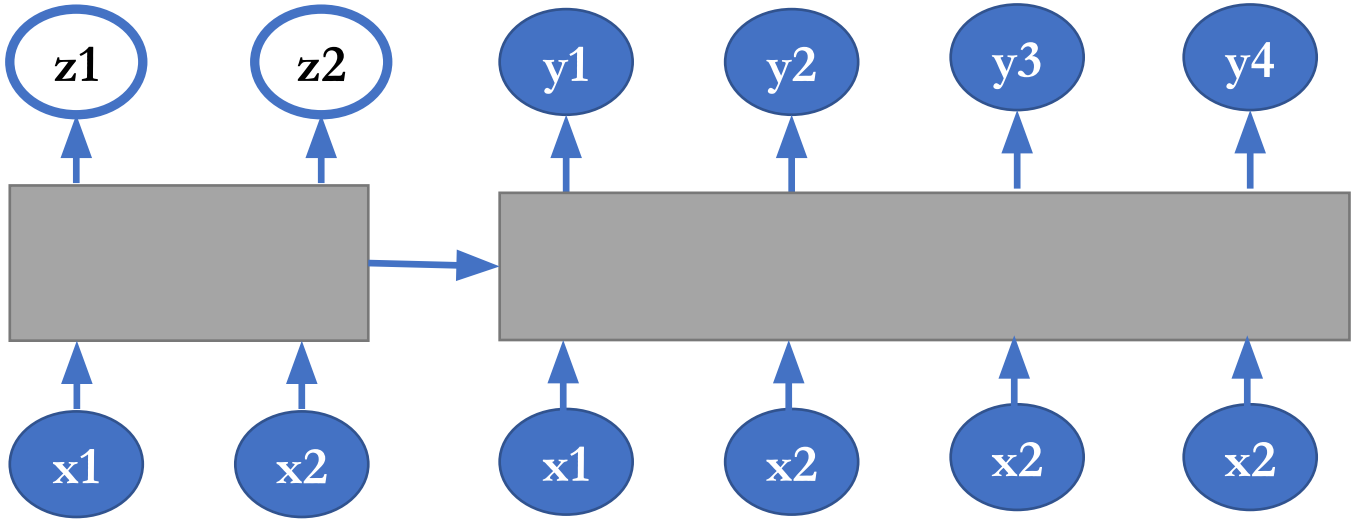
Neural **non**-autoregressive sequence modeling

- Repetition as a latent variable [Gu et al., 2018 ICLR]
- Each latent variable z_t: # of repetitions of the input symbol x_t
  - $|Z| = |X$



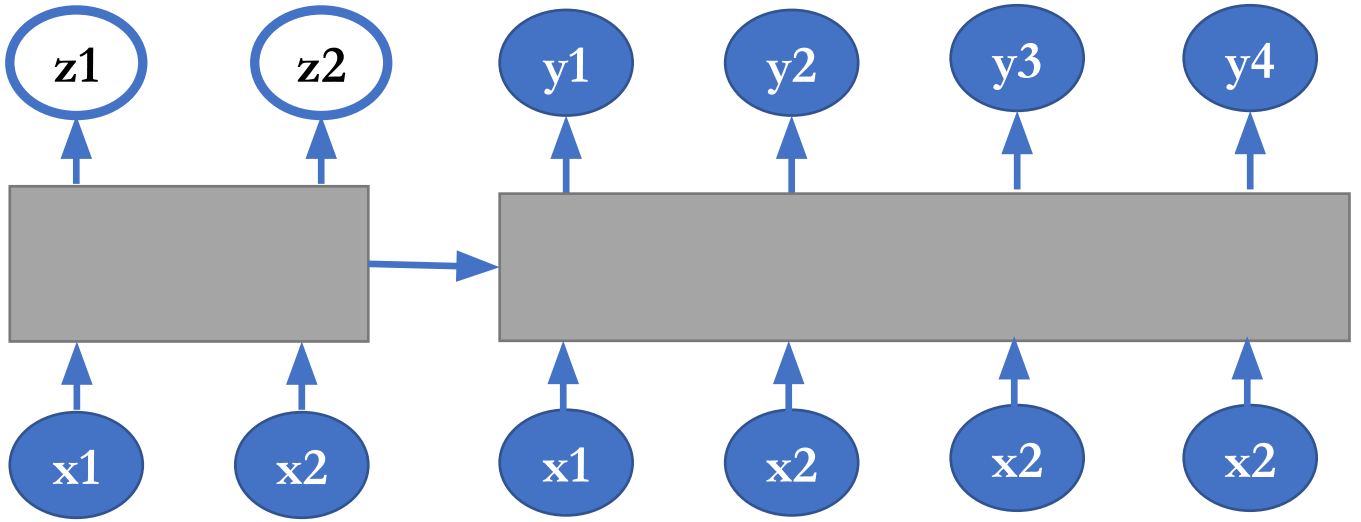
Neural **non**-autoregressive sequence modeling

- Repetition as a latent variable [Gu et al., 2018 ICLR]
- Each latent variable z_t: # of repetitions of the input symbol x_t
- Monte Carlo approximation with rescoring
  1. $Z_m \sim Z|X$, $Y_m = \arg\max_Y \log p(Y|Z_m, X)$
  2. Pick Y_m with the high score by another model.



Neural **non**-autoregressive sequence modeling

- Repetition as a latent variable [Gu et al., 2018 ICLR]
- Each latent variable z_t : # of repetitions of the input symbol x_t
- For training: use an auxiliary task to train $p(Z|X)$
  1. Word alignment models: use fast_align [Dyer et al., 2013]



Neural **non-**autoregressive sequence modeling

- **First** convincing result!! [Gu et al., 2018 ICLR]
- IWSLT'16 En→De

Non-Autoregressive?	Decoding	BLEU	Sentence Latency (ms)
No	Greedy	28.89	408ms
	Beam search (4)	**29.70**	**607ms**
Yes	argmax	25.20	39ms
	MC+Rescoring (10)	27.44	79ms
	MC+Rescoring (100)	**28.16**	**257ms**

Non-autoregressive modeling by iterative refinement [Lee, Mansimov & Cho, 2018]

- What are these latent variables?

$$\log p(Y|X) = \sum_{t=1}^{T} \log \sum_{Z} p(y_t|Z, X)p(Z|X)$$

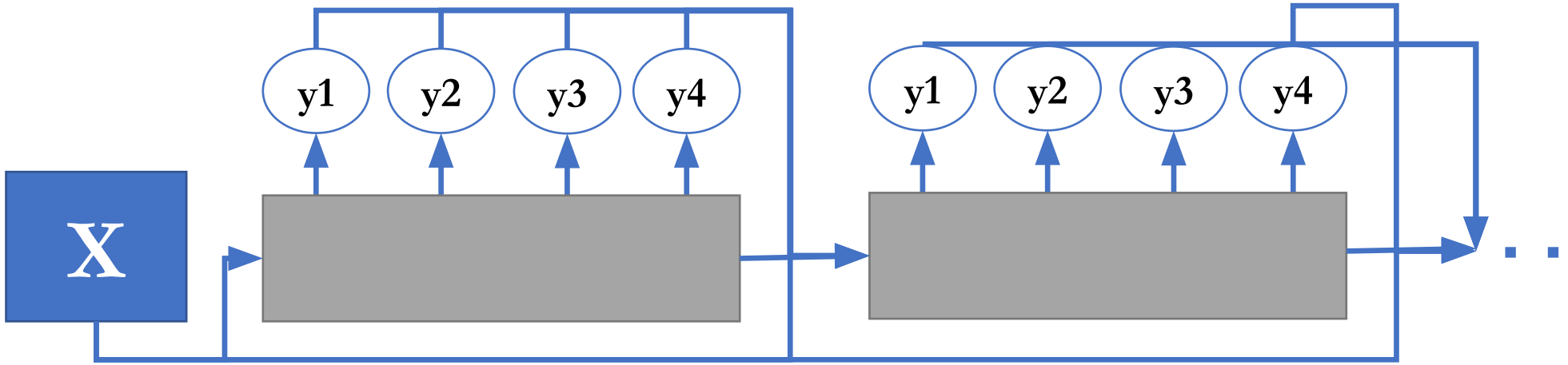
- We impose that latent variables share the output semantics
  - They share the same vocabulary $z_t \in V, y_t \in V$
- Multiple layers of the latent variables

$$p(Y|X) = \sum_{Z^1,\dots,Z^L} \left(\prod_{t=1}^{T} p(y_t|Z^L, X)\right) \left(\prod_{t=1}^{T} p(z_t^L|Z^{L-1}, X)\right) \cdots \left(\prod_{t=1}^{T} p(z_t^1|X)\right)$$

- Shared conditional distributions

Non-autoregressive modeling by iterative refinement

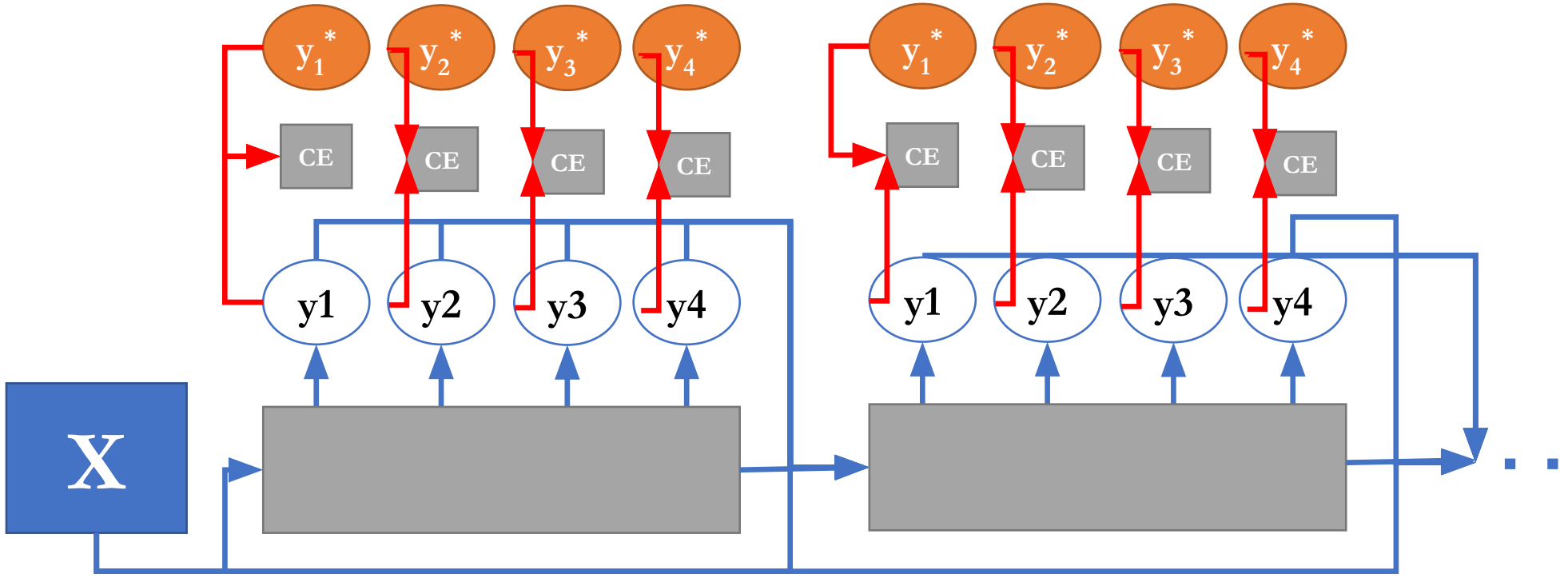
- Generative story: Iterative refinement
  1. Refine*: Generate an intermediate translation Y^l given a previous translation Y^{l-1} and the source sentence X
  2. Repeat 1 for L iterations (or until convergence)



* As the latent variables share the semantics with the output, we can use Z and Y exchangingly.

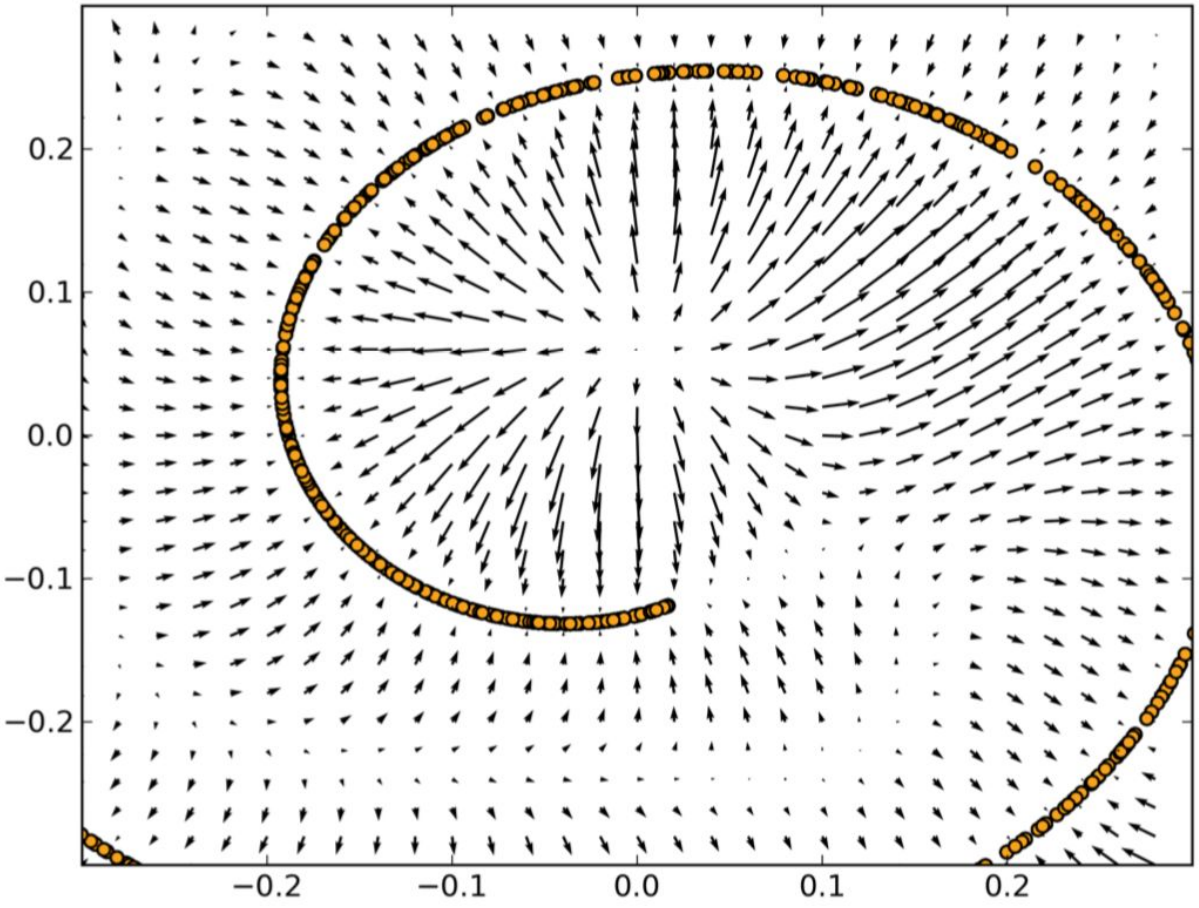
Non-autoregressive modeling by iterative refinement

- Training 1: end-to-end training
- The output of each iteration is encouraged to be the correct answer



Non-autoregressive modeling by iterative refinement

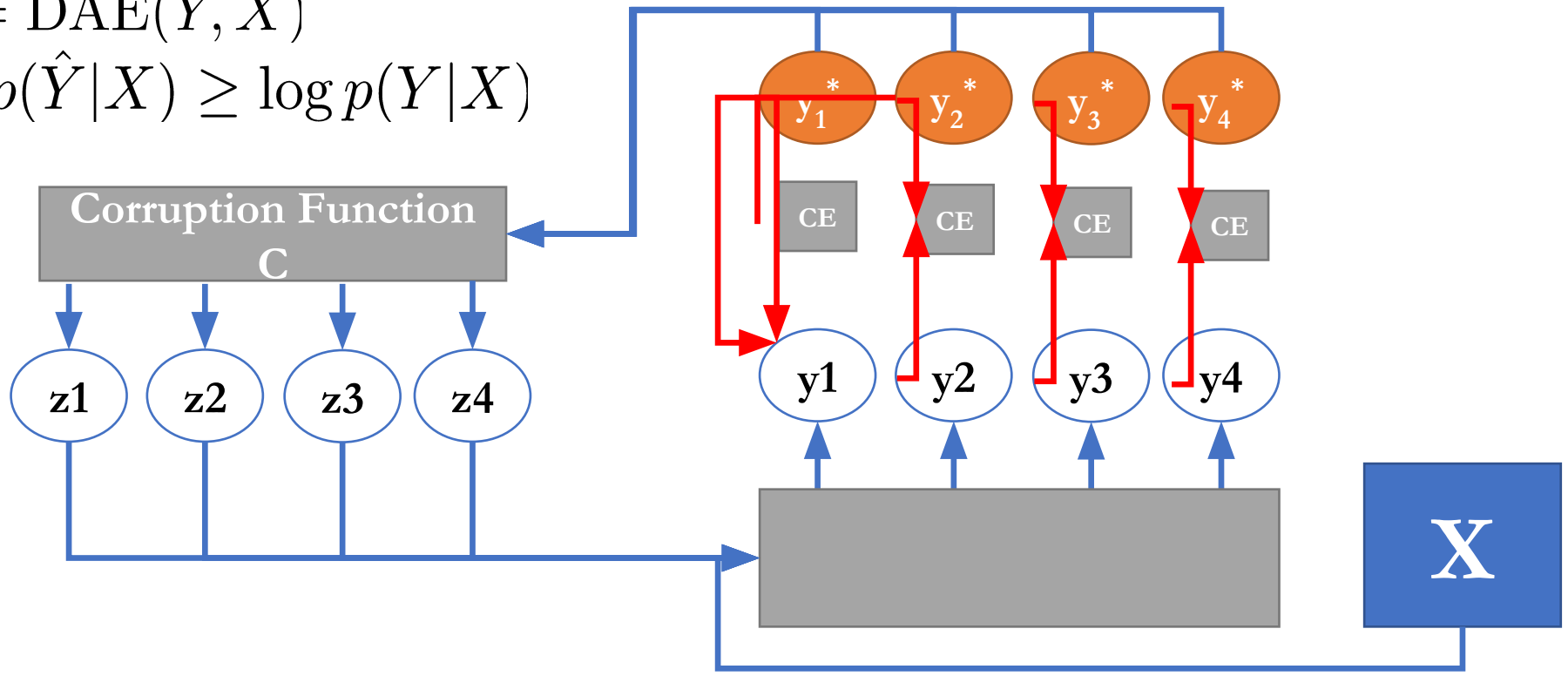
- Training 2: Conditional denoising autoencoder
- A denoising autoencoder learns to hill climb [Alain & Bengio, 2013]
  - $\hat{Y} = \text{DAE}(Y, X)$
  - $\log p(\hat{Y}|X) \geq \log p(Y|X)$



(b) $r(x) - x$ vector field, close-up

Non-autoregressive modeling by iterative refinement

- Training 2: Conditional denoising autoencoder
- A denoising autoencoder learns to hill climb [Alain & Bengio, 2013]
  - $\hat{Y} = \mathrm{DAE}(Y, X)$
  - $\log p(\hat{Y}|X) \geq \log p(Y|X)$



Non-autoregressive modeling by iterative refinement

- Lower-bound maximization & Conditional Denoising

$$J_{\text{LVM}}(\theta) = -\sum_{l=0}^{L+1}\left(\sum_{t=1}^{T}\log p_\theta(y_t^l = y_t^*|\hat{Y}^{l-1}, X)\right), \quad J_{\text{DAE}}(\theta) = -\sum_{t=1}^{T}\log p_\theta(y_t = y_t^*|\tilde{Y}, X)$$

- Mixed Training Objective
  - Consider L+1 iterations.
  - At each iteration, stochastically choose one of the two objectives.
  - Joint training from scratch

$$J(\theta) = -\sum_{l=0}^{L+1}\left(\alpha_l\sum_{t=1}^{T}\log p_\theta(y_t^*|\hat{Y}^{l-1}, X) + (1-\alpha_l)\sum_{t=1}^{T}\log p_\theta(y_t^*|\tilde{Y}, X)\right)$$

Experiments – Machine Translation

- En↔Ro (WMT'16): low-resource machine translation

Non-Autoregressive?	Decoding	En→Ro (BLEU)	Ro→En (BLEU)	Speed (toks/sec)	
				CPU	GPU
No	Greedy	**31.93**	**31.55**	**15.7**	**55.6**
	Beam (4)	**32.40**	**32.06**	**7.3**	**43.3**
Yes	Iter 1	24.45	25.73	98.6	694.2
	Iter 2	27.10	28.15	62.8	332.7
	Iter 5	28.86	29.72	29.0	194.4
	Iter 10	29.32	30.19	14.8	93.1
	adaptive	**29.66**	**30.30**	**16.5**	**118.3**

- 91.5% translation quality with up to 4x decoding speed-up (on GPU)

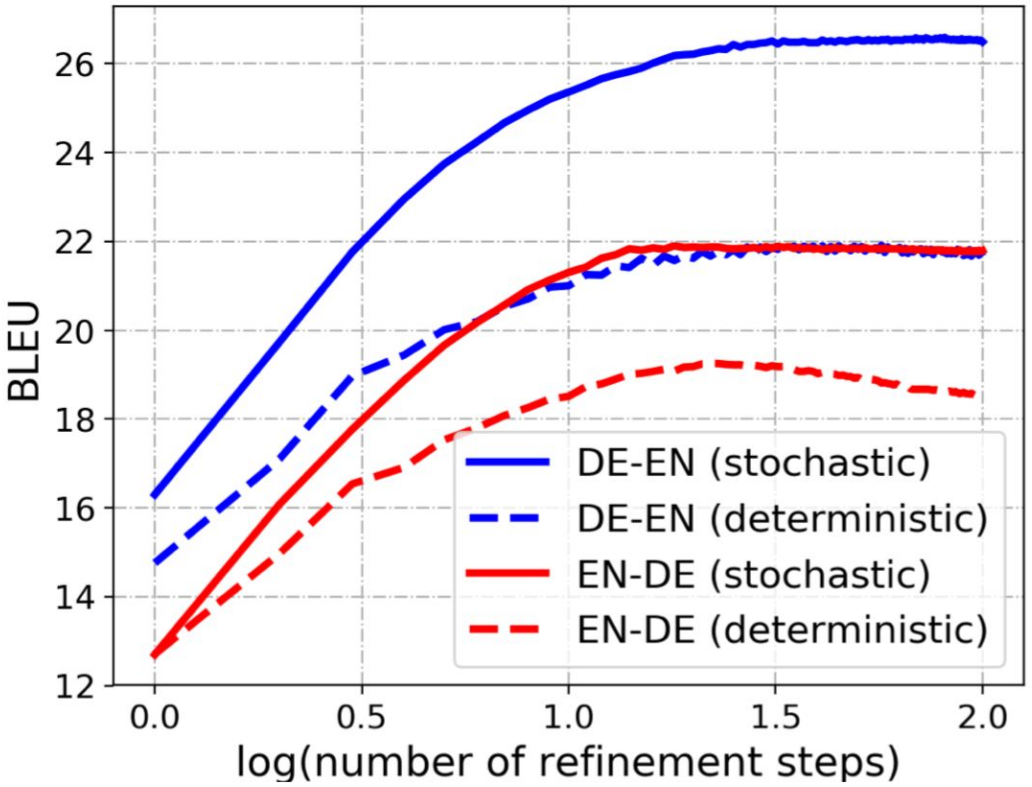
Experiments – Machine Translation

- En↔De (WMT'14): moderate-scale machine translation

Non-Autoregressive?	Decoding	En→De (BLEU)	De→En (BLEU)	Speed (toks/sec)	
				CPU	GPU
No	Greedy	**23.77**	**28.15**	**15.8**	**54.0**
	Beam (4)	**24.57**	**28.47**	**7.0**	**44.9**
Yes	Iter 1	13.91	16.77	83.3	511.4
	Iter 2	16.95	20.39	49.6	393.6
	Iter 5	20.26	23.86	23.1	139.7
	Iter 10	21.61	25.48	12.3	90.4
	adaptive	**21.54**	**25.43**	**20.3**	**107.2**

- 80% translation quality with up to 2x decoding speed-up (on GPU)

Experiments – Machine Translation



- Iterative refinement improves translation quality (almost) monotonically
  - intermediate latent variables (translations) are successfully capturing dependencies.
- Quality degradation with large data
- Significant speed-up in decoding on GPU
  - Perhaps more suitable for brains? ☺

Experiments – Machine Translation

***Src**: seitdem habe ich sieben Ha ̈user in der Nachbarschaft mit den Lichtern versorgt und sie funktionierenen wirklich gut .*

***Iter 1**: and I ’ve been seven homes since in neighborhood with the lights and they ’re really functional .*

***Iter 4**: and I ’ve been seven homes in neighborhood with the lights , and they ’re a really functional .*

***Iter 8**: and I ’ve been providing seven homes in the neighborhood with the lights and they ’re a really functional .*

***Ref**: since now , I ’ve set up seven homes around my community , and they ’re really working .*

***Src**: er sah sehr glu ̈cklich aus , was damals ziemlich ungewo ̈hnlich war , da ihn die Nachrichten meistens deprimierten .*

***Iter 1**: he looked very happy , which was pretty unusual the , because the news was were usually depressing .*

***Iter 4**: he looked very happy , which was pretty unusual at the , because news was mostly depressing .*

***Iter 8**: he looked very happy , which was pretty unusual at the time because the news was mostly depressing .*

***Ref**: there was a big smile on his face which was unusual then , because the news mostly depressed him .*

Experiments – Image Caption Generation

- MS COCO: image caption generation

Non-Autoregressive?	Decoding	BLEU	Speed (toks/sec)	
			CPU	GPU
No	Greedy	**23.47**	**4.3**	**2.1**
	Beam (4)	**24.78**	**3.6**	**1.0**
Yes	Iter 1	20.12	17.1	8.9
	Iter 2	20.88	12.0	5.7
	Iter 5	21.12	6.2	2.8
	Iter 10	21.24	2.0	1.2
	adaptive	**21.12**	**10.8**	**4.8**

- 85% caption quality with up to 5x decoding speed-up (on GPU)

Non-autoregressive modeling by iterative refinement

Input X

Y^1 a woman standing on playing tennis on a tennis racquet .

Y^2 a woman standing on a tennis court a tennis racquet .

Y^3 a woman standing on a tennis court a a racquet .

Y^4 a woman standing on a tennis court holding a racquet .

Non-autoregressive modeling by iterative refinement

Input X

Y^1 a yellow bus parked on parked in of parking road .

Y^2 a yellow and black on parked in a parking lot .

Y^3 a yellow and black bus parked in a parking lot .

Y^4 a yellow and black bus parked in a parking lot.

Part 1: Conclusion

- Latent variables capture output dependencies more efficiently.
- Different interpretation → Different learning/decoding algorithms
  - Gu et al. [2018]: fertility → auxiliary supervision + noisy parallel decoding
  - Lee+Mansimov+Cho [2018]: iterative refinement → conditional denoising
  - Kaiser et al. [2018]: latent sequence → hidden autoregressive inference
  - What else?
- Generation quality closely tracks the autoregressive models'.
- Decoding is significantly faster especially with GPU.
  - Potentially even faster decoding with a specialized hardware.

Part 1: Conclusion – Future Directions

- Mix of non-autoregressive and autoregressive paradigms
  - Autoregressive modeling followed by iterative refinement? [Xia et al., 2017; Grangier & Auli, 2017]
  - Autoregressive generation of segments and non-autoregressive generation within each segment [Kaiser et al., 2018; Huang et al., 2018], or
  - Non-autoregressive generation of segments and autoregressive generation within each segment?
- Beyond sentence-level generation
  - Efficiency of the non-autoregressive model may enable document-level generation.
- Many exciting future directions!

Meta-Learning of Low-Resource Neural Machine Translation

Model-**a**gnostic **M**eta-learning for **neural machine translation**

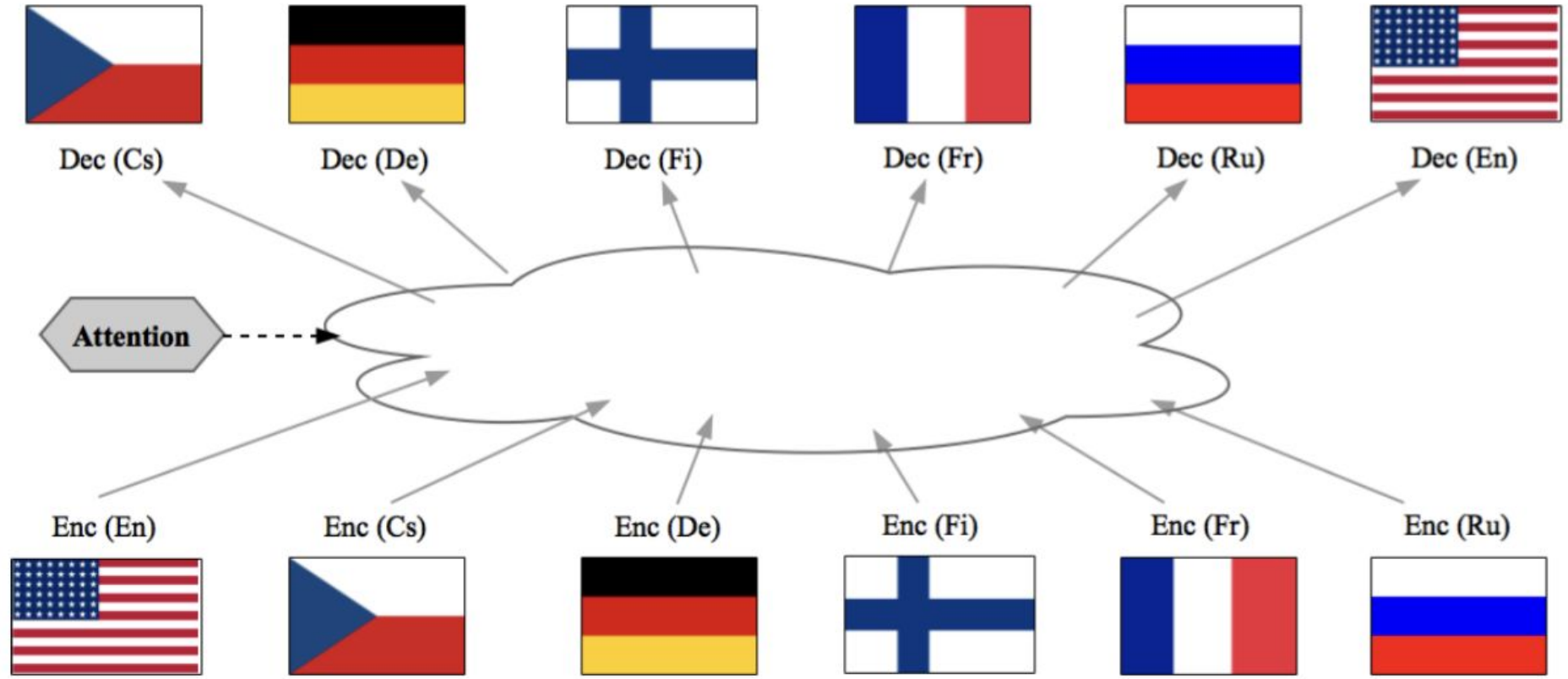
Jiatao Gu, **Yong Wang**, **Yun Chen**, Victor O. K. Li and Kyunghyun Cho.
Meta-Learning for Low-Resource Neural Machine Translation. 2018 (under review)

Multilingual Translation – (1)

- Traditionally,
  - If a parallel corpus exists, one system for each language pair.
    - Parallel corpus: $D^{a \to b} = \left\{ (X_1^a, Y_1^b), \ldots, (X_N^a, Y_N^b) \right\}$
    - Translation system: $\log p(Y^b | X^a)$
  - If no direct parallel corpus exists, a pivot-based translation.
    - No direct parallel corpus: $D^{a \to b} = \emptyset$
    - But, $|D^{a \to c}| > 0, |D^{c \to b}| > 0$
    - Then, $\log p(Y^b | \hat{X}^c)$, where $\hat{X}^c = \arg\max_X \log p(X^c | X^a)$
    - *c* is a pivot language (often, English.)
  - No knowledge transfer between different language pairs.

Multilingual Translation as Multitask Learning – (2)

- Now, [Firat et al., 2016a; Firat et al., 2016b; Johnson et al., 2016; Ha et al., 2016; Lee et al., 2017]



Multilingual Translation as Multitask Learning – (3)

- Separate encoder/decoders
  - [Firat et al., 2016a; Firat et al., 2016b]
- One encoder per source *l*

$$f^l_{\text{enc}} : V_l \times \cdots \times V_l \to \mathbb{R}^d \times \cdots \times \mathbb{R}^d$$

- One decoder per target *l'*

$$\log p^{l'}(Y^{l'}|H)$$

- For each pair (l, l'),

$$\log p^{l'}(Y^{l'}|H = f^l_{\text{enc}}(X^l))$$

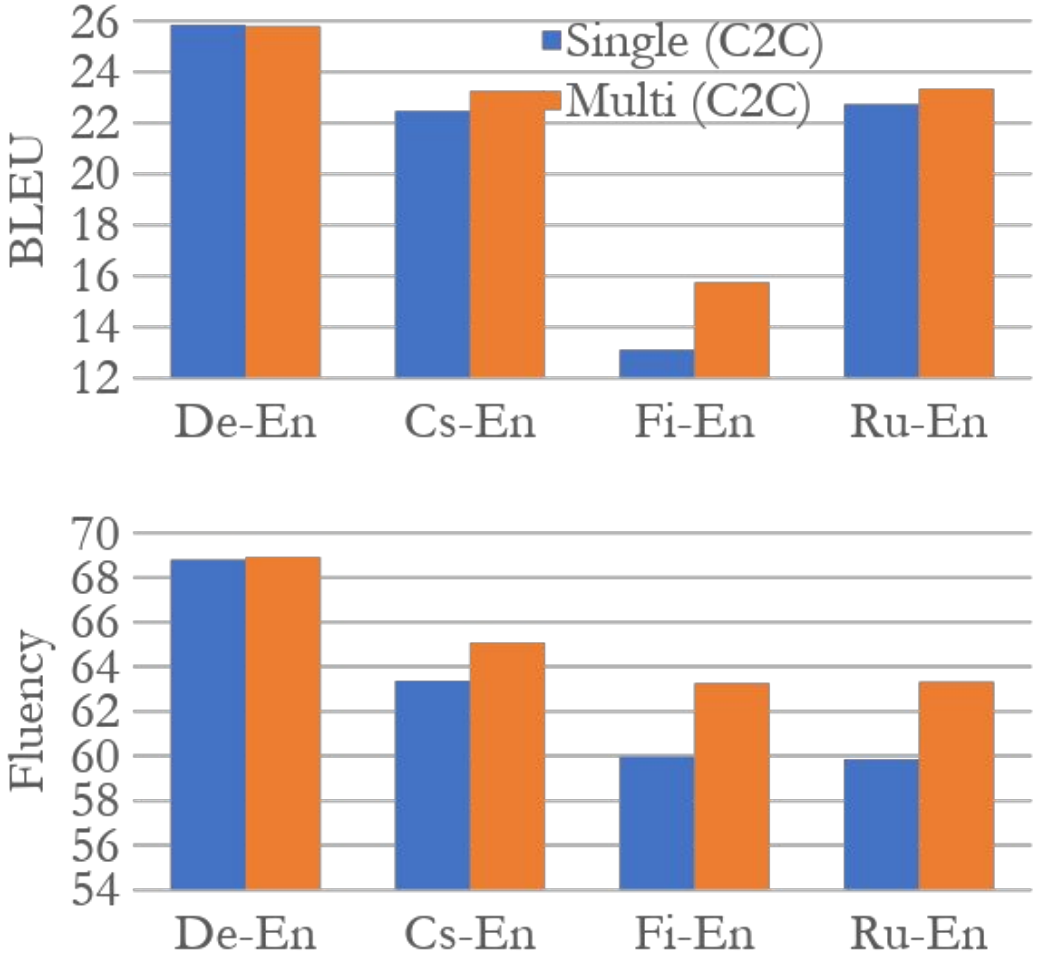
- Train using all available language pairs

- Universal encoder/decoders
  - [Johnson et al., 2016; Ha et al., 2016; Lee et al., 2017; Gu et al., 2018]
- Shared lexicons $f^l_{\text{lex}} : V_l \to V$
  - A shared vocabulary of language-agnostic tokens [J, 2016; H, 2016; L, 2017]
  - Universal lexical representation [G, 2018]
- One encoder-decoder for all pairs

$$f^{-l'}_{\text{lex}}(\arg\max_Y \log p(Y|f^l_{\text{lex}}(X^l)))$$

Multilingual Translation as Multitask Learning – (4)

- Does it work?
- Single-pair Systems
  De→En, Cs→En, Fi→En, Ru→En
- Multilingual System
  {De, Cs, Fi, Ru}→En
- The latter has 1/4x parameters
- Better translation quality on low-resource languages (Fi & Ru)



Lee, Cho and Hoffman (2017)

Multilingual Translation as Multitask Learning – (5)

- Does it work? – **Yes!***
- Single-pair Systems vs. Multilingual System
- Works with intra-sentence code-switching

(e) Multilingual

Multi src	Bei der Metropolitního výboru pro dopravu für das Gebiet der San Francisco Bay erklärten Beamte , der Kongress könne das Problem банкротство доверительного Фонда строительства шоссейных дорог einfach durch Erhöhung der Kraftstoffsteuer lösen .
EN ref	At the Metropolitan Transportation Commission in the San Francisco Bay Area , officials say Congress could very simply deal with the bankrupt Highway Trust Fund by raising gas taxes .
bpe2char	During the Metropolitan Committee on Transport for San Francisco Bay , officials declared that Congress could solve the problem of bankruptcy by increasing the fuel tax bankrupt .
char2char	At the Metropolitan Committee on Transport for the territory of San Francisco Bay , officials explained that the Congress could simply solve the problem of the bankruptcy of the Road Construction Fund by increasing the fuel tax .

** It often fails to translate between a pair of languages not seen during training*

Lee, Cho and Hoffman (2017)

Limitations of Multitask Learning – (1)

- Tricky when the availability of data drastically differs across languages.
  - *overfitting* on low-resource pairs, while *underfitting* on high-resource pairs.

$$L(\theta) = \sum_l \frac{1}{N^l} \sum_{n=1}^{N^l} \log p_\theta(Y_n^l | X_n^l)$$

  - Extremely low-resource pairs can easily be *ignored*.

$$L(\theta) = \sum_l \sum_{n=1}^{N^l} \log p_\theta(Y_n^l | X_n^l)$$

  - See [Firat et al., 2016a] and [Lee et al., 2017] for more discussion.
  - *It is really horrible to figure out how to tackle this in practice…*

Limitations of Multitask Learning – (2)

- Assumes the availability of all language pairs in advance.
  - The entire model must be re-trained each time a new language is introduced.
- Transfer Learning [Zoph et al., 2016; Nguyen & Chiang, 2017]
  - Only re-train a subset of parameters on a new language pair.
  - Many possible strategies, but no clear winning strategy.

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

Zoph et al., 2016

Limitation of Multitask Learning – (3)

- Inconvenient truths about multitask+transfer learning
  - Relies on our intuition that all languages/tasks share common underlying structures: *true?*
  - Assumes multitask learning can capture those underlying structures and share across multiple languages/tasks: *true?*
  - Assumes multitask-learned parameters are a good initialization for further training: *true?*

- Is there a more satisfying approach?

Meta-Learning: MAML [Finn et al., 2018] – (1)

- Model-agnostic meta-learning [Finn et al., 2018]
- Two-stage learning
  1. Simulated learning

$$\begin{aligned}\text{Learn}(D_{\mathcal{T}};\theta^0) &= \arg\max_{\theta} \mathcal{L}^{D_{\mathcal{T}}}(\theta) \\ &= \arg\max_{\theta} \sum_{(X,Y)\in D_{\mathcal{T}}} \log p(Y|X,\theta) - \beta\|\theta-\theta^0\|^2,\end{aligned}$$

  2. Meta-learning

$$\mathcal{L}(\theta) = \mathbb{E}_k \mathbb{E}_{D_{\mathcal{T}^k}, D'_{\mathcal{T}^k}} \left[\sum_{(X,Y)\in D'_{\mathcal{T}^k}} \log p(Y|X;\text{Learn}(D_{\mathcal{T}^k};\theta)) \right],$$

Meta-Learning: MAML [Finn et al., 2018] – (2)

1. Simulated learning
   - Given a small subset $D_{\mathcal{T}}$ of the training set of task $\mathcal{T}$, update the model parameters $N = 1$ times.

$$\begin{aligned}\text{Learn}(D_{\mathcal{T}}; \theta^0) &= \arg\max_{\theta} \mathcal{L}^{D_{\mathcal{T}}}(\theta) \\ &= \arg\max_{\theta} \sum_{(X,Y)\in D_{\mathcal{T}}} \log p(Y|X,\theta) - \beta\|\theta - \theta^0\|^2, \\ &= \theta_0 + \eta\nabla_{\theta}\mathcal{L}^{D_{\mathcal{T}^k}}(\theta_0)\end{aligned}$$

   - Clip the update so that $\eta\nabla_{\theta}\mathcal{L}^{D_{\mathcal{T}^k}}(\theta_0)$ does not deviate too much from θ_0.
   - *It simulates finetuning on a target task with a limited resource.*

Meta-Learning: MAML [Finn et al., 2018] – (3)

2. Meta-Learning
   - Randomly select a task k and select a training subset $D = D_{\mathcal{T}^k}$
   - Randomly select a validation subset $D' = D'_{\mathcal{T}^k}$ for evaluation.
   - Update the meta-parameter θ_0 by gradient descent:

$$\theta_0 \leftarrow \theta_0 + \eta_0 \nabla_{\theta_0} \mathcal{L}^{D'}(\theta_0),$$

   where

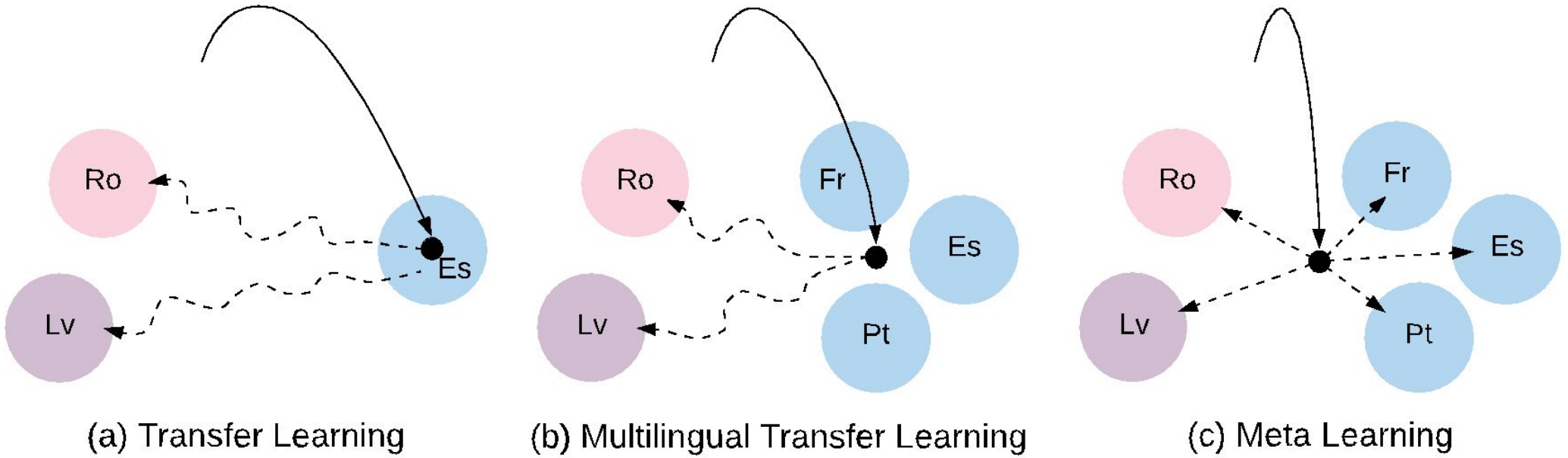
$$\begin{aligned}\nabla_\theta \mathcal{L}^{D'}(\theta') &= \nabla_{\theta'} \mathcal{L}^{D'}(\theta') \nabla_\theta (\theta - \eta \nabla_\theta \mathcal{L}^{D}(\theta)) \\ &= \nabla_{\theta'} \mathcal{L}^{D'}(\theta') - \eta \nabla_{\theta'} \mathcal{L}^{D'}(\theta') H_\theta(\mathcal{L}^{D}(\theta))\end{aligned}$$

   - *Update the meta-parameter so that N-step GD on the k-th task works well.*

Meta-Learning: MAML [Finn et al., 2018] – (4)

3. Fast adaptation to a new task
   - Given a small training set D of the new target task, SGD starting from the meta-parameter θ_0
   - Early stopping based on $\|\theta - \theta_0\|^2$.

Multitask learning vs. Meta-learning



a) Transfer learning does not take into account subsequent learning.

b) Multilingual learning does not take into account new, future tasks.

c) Meta-learning considers subsequent learning on new, future tasks.

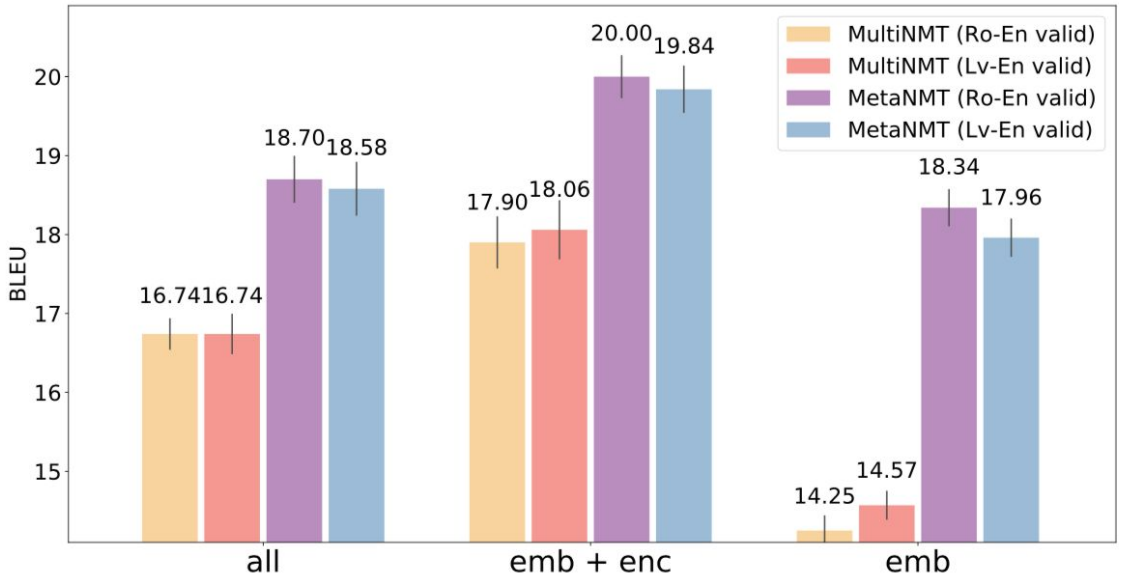
Extension to Neural Machine Translation

- I/O mismatch between different tasks
  - Vocabulary mismatch among different languages
- Multilingual word embedding [Artetxe et al., 2017; Conneau et al., 2018; and more]
  - Project each token into a continuous vector space $f^l : V^l \rightarrow \mathbb{R}^d$
  - Ensure that they are compatible:
    $\|f^l(v^l) - f^{l'}(v^{l'})\|^2 < \epsilon$, iff v^l and $v^{l'}$ have the same meaning.
- Universal lexical representation [Gu et al., 2018]

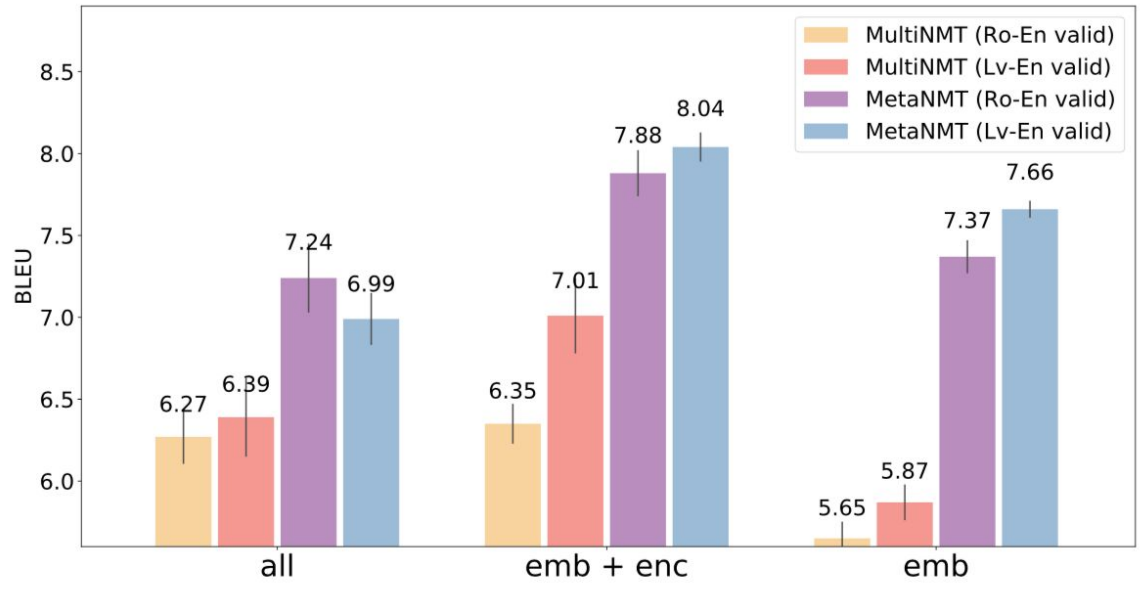
- *Meta-NMT!*

Experiments

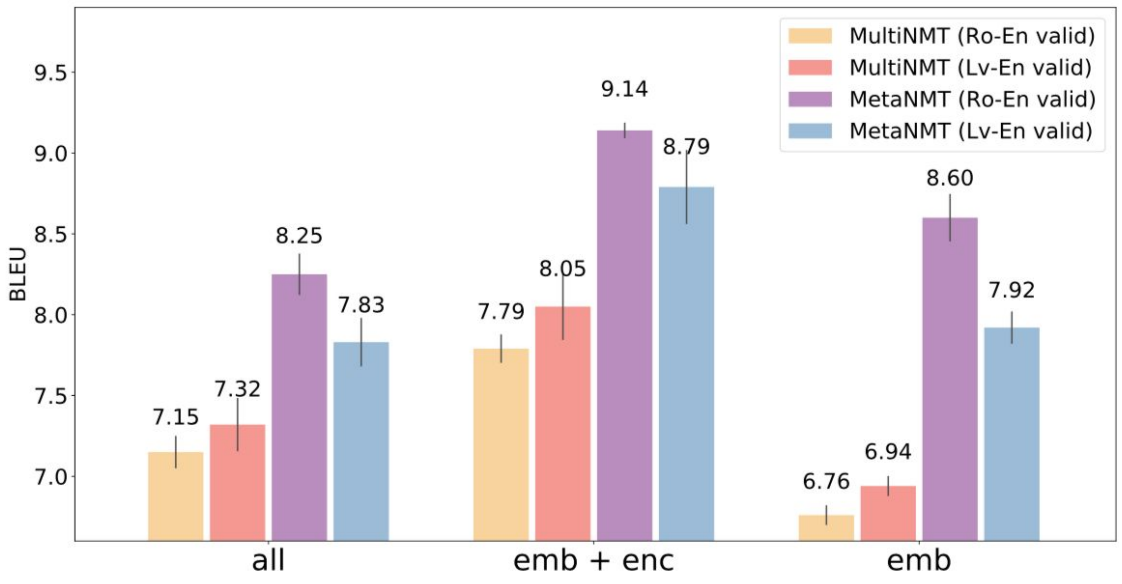
- Source tasks: all the languages from Europarl + Russian
  - Bg→En, Cs→En, Da→En, De→En, El→En, Es→En, Et→En, Fr→En, Hu→En, It→En, Lt→En, Nl→En, Pl→En, Pt→En, Sk→En, Sl→En, Sv→En and Ru→En.
  - Reasonable high-resource language pairs.
- Target tasks: (simulated) low-resource language pairs
  - Ro→En, Lv→En, Fi→En, Tr→En and Ko→En
  - Approximately 16k target tokens (English side): roughly 800 sentence pairs.
- Universal lexical representation: obtained from Wikipedia.
- Early stopping of meta-learning: either Ro-En or Lv-En



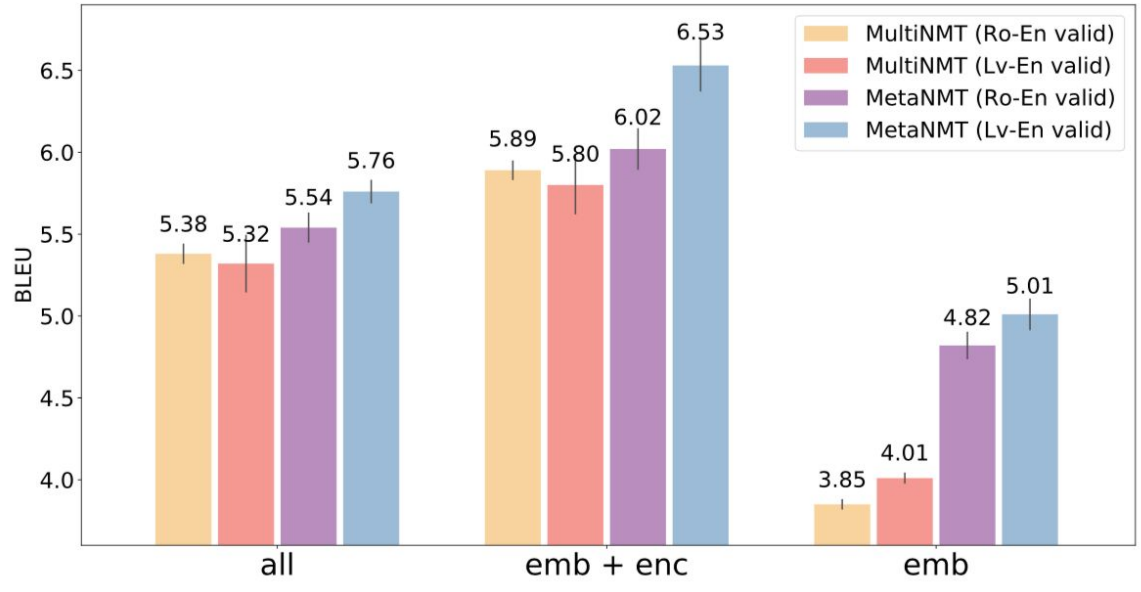
(a) Ro-En



(b) Lv-En



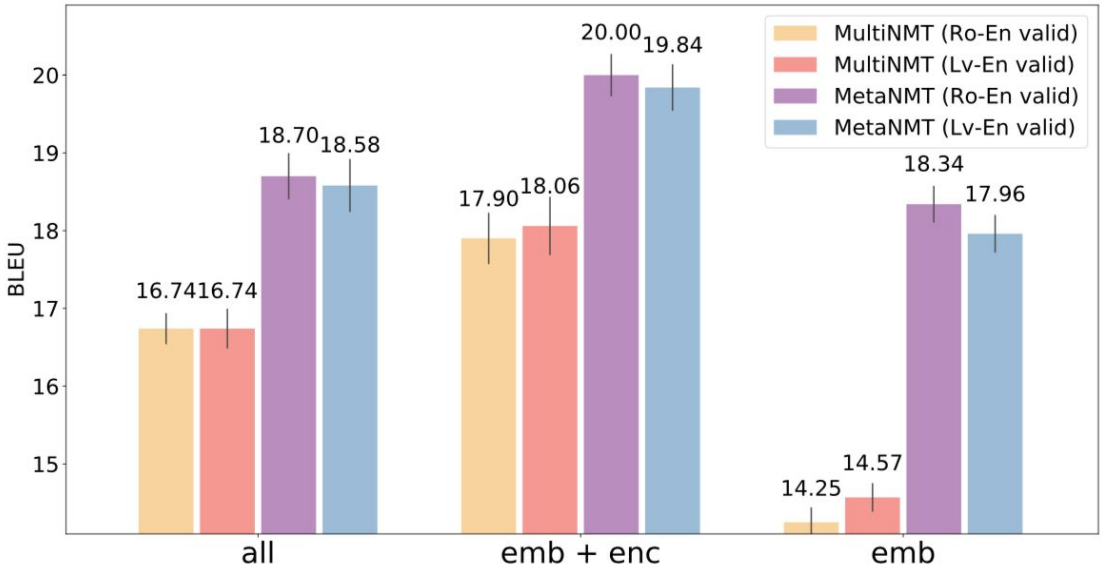
(c) Fi-En



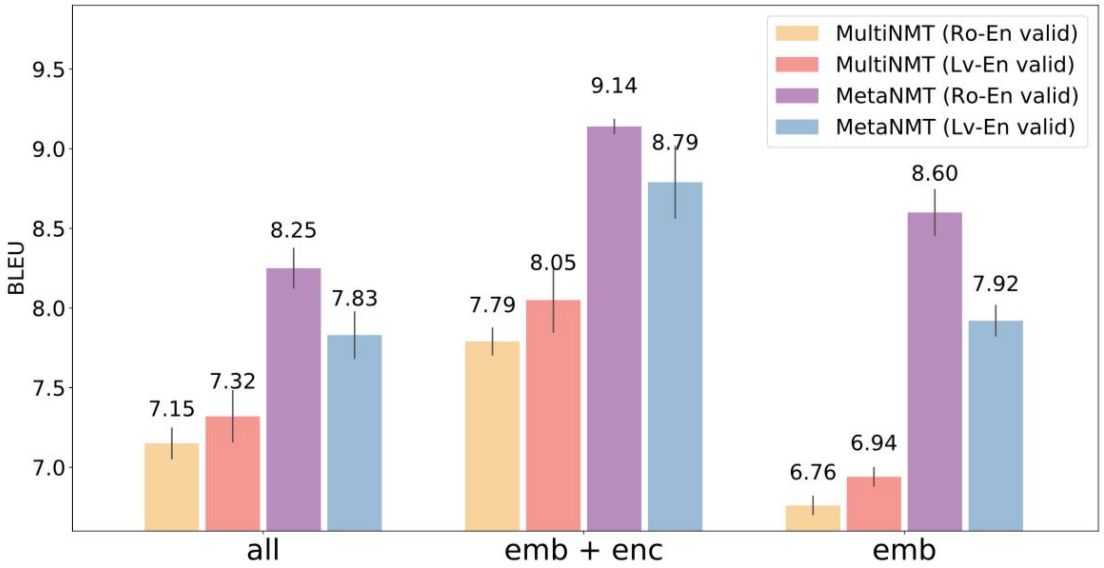
(d) Tr-En

Experiments – (1)

- Meta-learning outperforms multitask learning across all the target languages and across different finetuning strategies.
- Using only 800 examples, reaches up to 65% of fully-supervised models in terms of BLEU.



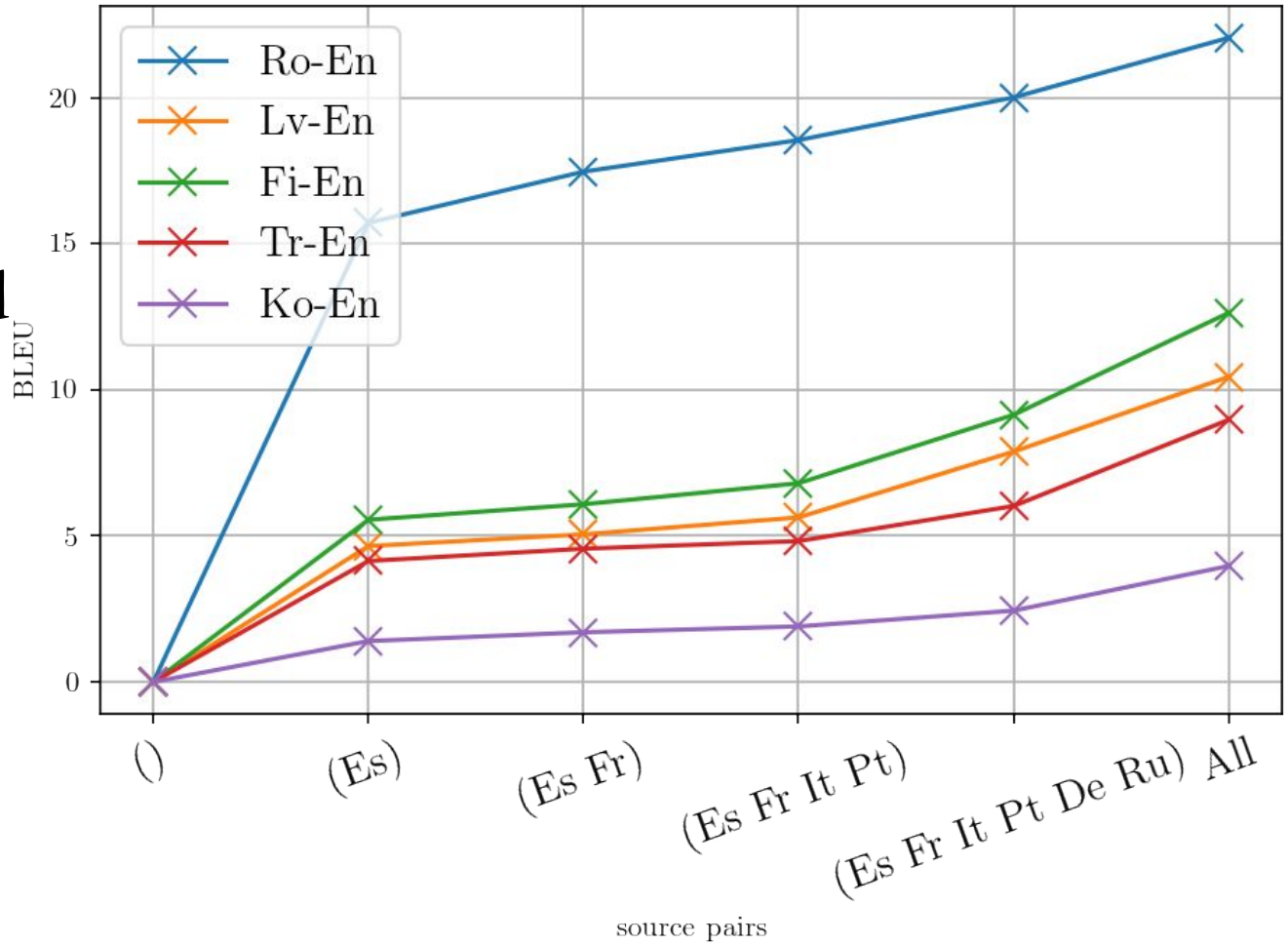
(a) Ro-En



(c) Fi-En

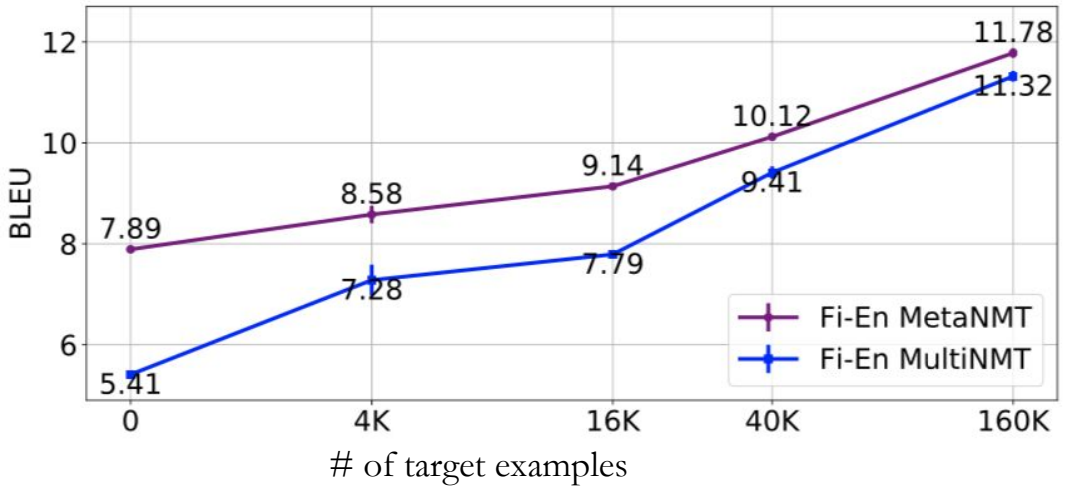
Experiments – (2)

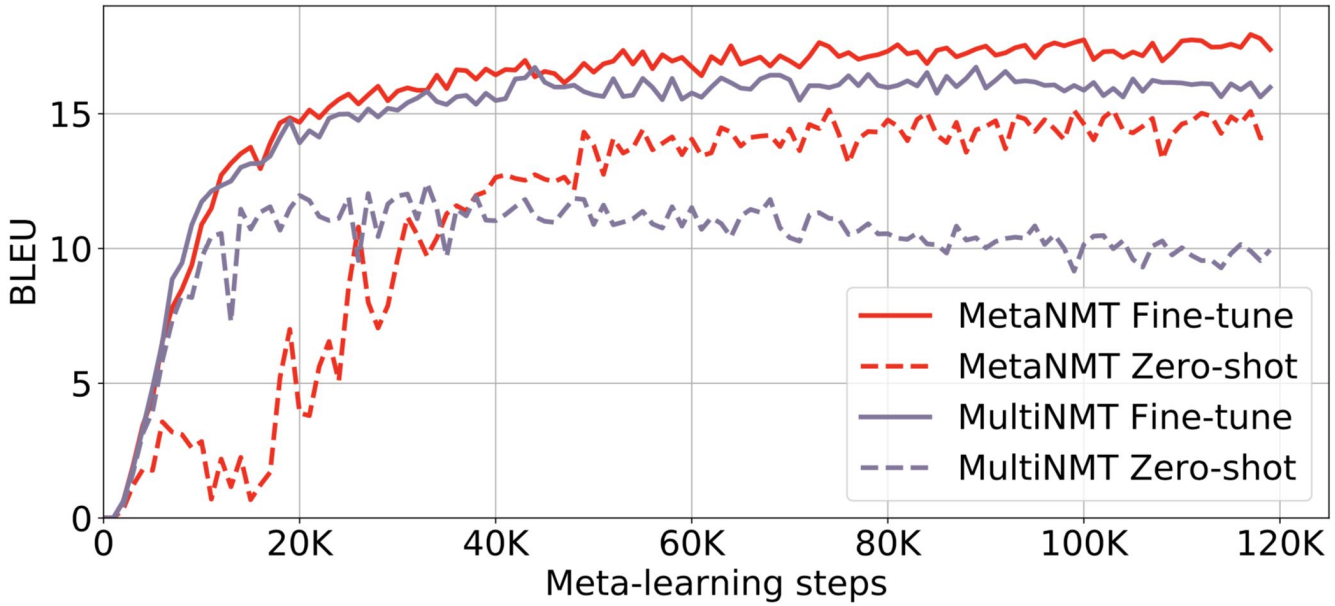
- More source tasks lead to greater improvements.
- The similarity between source and target asks matters.



Experiments – (3)

- Multi-task learning over-adapts to the source tasks.
  - Performance on the target task degrades with longer multi-task learning.
- Meta-learning does not over-adapt.
  - The meta-learning objective explicitly takes into account finetuning on a target task.
  - It requires less target examples.





Experiments – (4) Sample Translations

Source (Tr)	google mülteciler için 11 milyon dolar toplamak üzere bağış eşleştirme kampanyasını başlattı .
Target	google launches donation-matching campaign to raise $ 11 million for refugees .
Meta-0	google refugee fund for usd 11 million has launched a campaign for donation .
Meta-16k	google has launched a campaign to collect $ 11 million for refugees .
Source (Ko)	이번에 체포되어 기소된 사람들 중에는 퇴역한 군 고위관리 , 언론인 , 정치인 , 경제인 등이 포함됐다
Target	among the suspects are retired military officials , journalists , politicians , businessmen and others .
Meta-0	last year , convicted people , among other people , of a high-ranking army of journalists in economic and economic policies , were included .
Meta-16k	the arrested persons were included in the charge , including the military officials , journalists , politicians and economists .

Part 2: Conclusion

- Meta-learning allows us to exploit many high-resource tasks for *extremely low-resource* target tasks.
- Gradual shift toward higher-order learning
  - Learning to optimize [Andrychowicz et al., 2017; and others]
  - Multi-agent modelling (theory of mind) [Foerster et al., 2018 LOLA; and others]
  - Neural architecture search [Zoph & Le, 2016; and others]
  - Hyperparameter search [Luketina et al., 2016; and others]
  - And more on the horizon…

Lessons learned

- Lesson 1
  - I thought sequential decoding of a sequence was *the* answer.
  - I thought multitask, transfer learning was *the* answer.
  - In both cases, I have been so wrong and will probably be wrong again.

- Lesson 2
  - Denoising (iterative refinement) for structured output prediction
  - Second-order learning for meta-learning
  - So many (yet unknown) learning algorithms/regimes are out there.

Thank you!