Fast-Electra For Efficient Pre-Training

Chengyu Dong[1], Liyuan Liu[2], Hao Cheng[2], Jingbo Shang[1],
Jianfeng Gao[2], Xiaodong Liu[2]

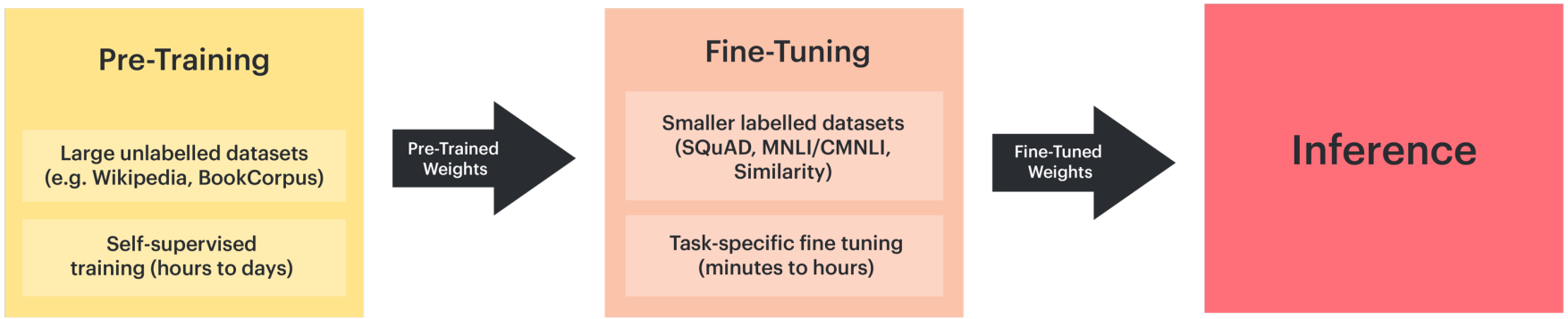
[1] University of California, San Diego
[2] Microsoft Research

Contributions

- We decouple the trainings of the auxiliary model and the main model in ELECTRA-style pretraining, to enable large-scale pretraining with ELECTRA

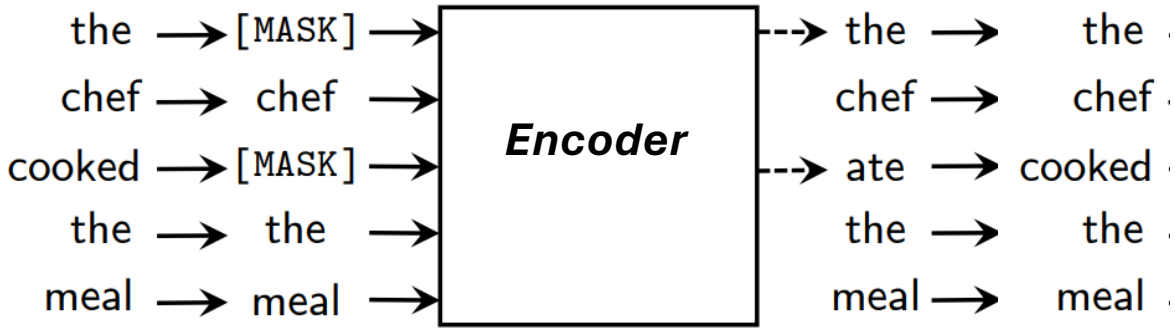
Language Model Pretraining

- Two-staged training of language models
  - (Pretraining) Self-supervised training on large unlabeled datasets
  - (Fine-tuning) Supervised training on labelled datasets for specific downstream tasks



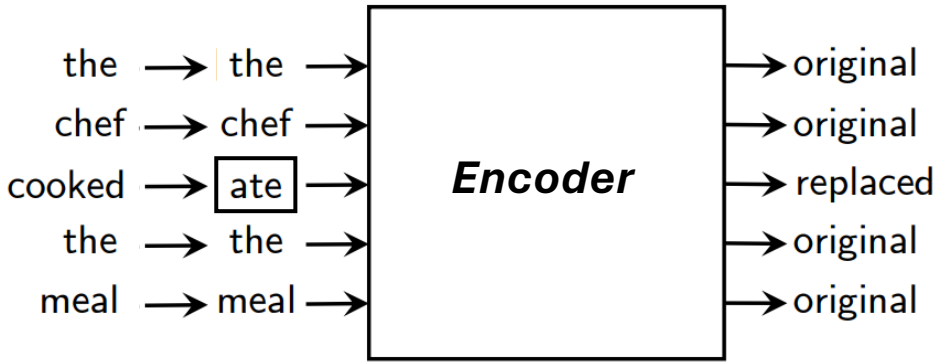
Pretraining methods: BERT-style vs. ELECTRA-style

- BERT-style: Masked Language Modeling (MLM)
  - Train the model to reconstruct masked tokens in a sequence



Pretraining methods: BERT-style vs. ELECTRA-style

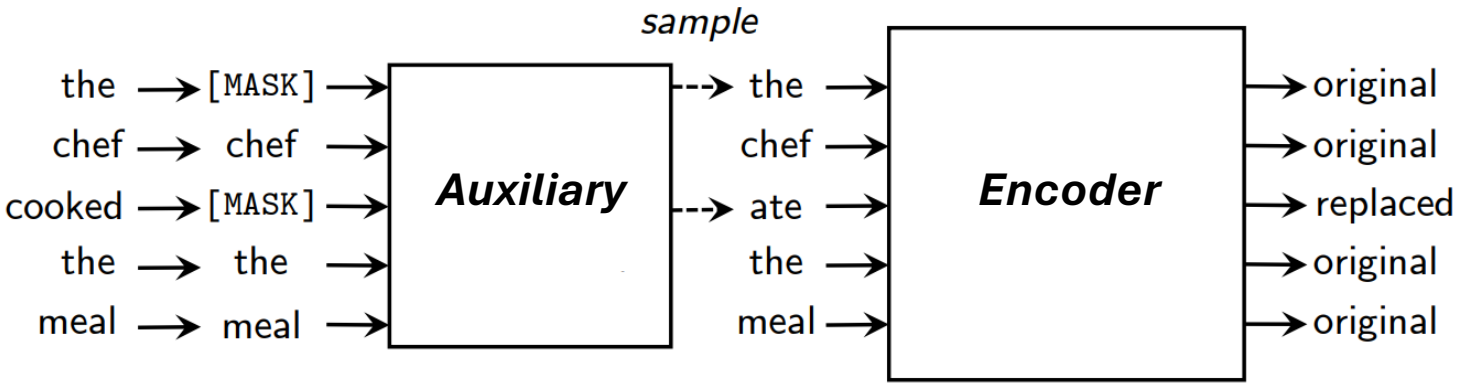
- ELECTRA-style: Replaced Token Detection (RTD)
  - Train the model to detect replaced tokens in a sequence



Clark et al., Electra: Pre-training text encoders as discriminators rather than generators. 2020.

Pretraining methods: BERT-style vs. ELECTRA-style

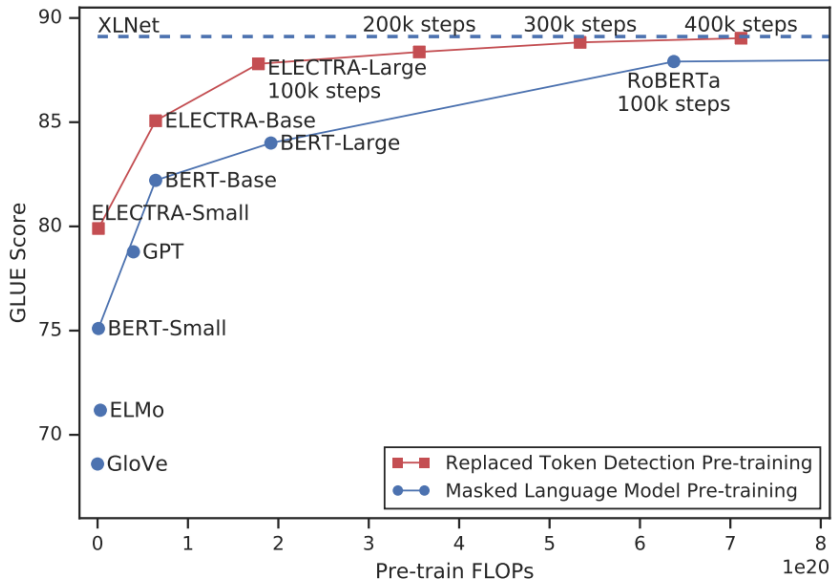
- ELECTRA-style: Replaced Token Detection (RTD)
  - Train the model to detect replaced tokens in a sequence
  - The replaced tokens are generated by an auxiliary model (“Generator”) trained with MLM



Clark et al., Electra: Pre-training text encoders as discriminators rather than generators. 2020.

Advantage of ELECTRA-style Pretraining

- Training efficiency and model efficiency
  - Achieve similar performance with less pretraining steps
  - Achieve similar performance with a smaller model size

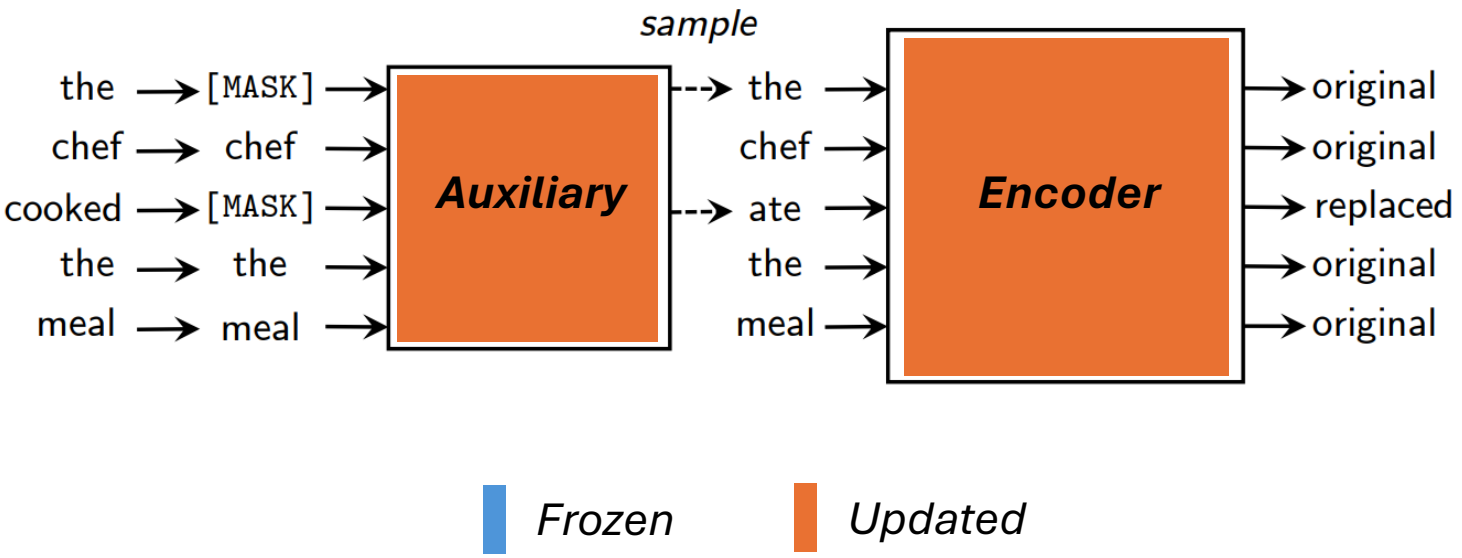


Clark et al., Electra: Pre-training text encoders as discriminators rather than generators. 2020.

Difficulty in Scaling up ELECTRA-style Pretraining

- The auxiliary model is jointly trained with the main language model

$$\min_{\theta,\theta_{\text{aux}}} L_{\text{MLM}}(\theta_{\text{aux}}) + \lambda L_{\text{RTD}}(\theta)$$



Clark et al., Electra: Pre-training text encoders as discriminators rather than generators. 2020.

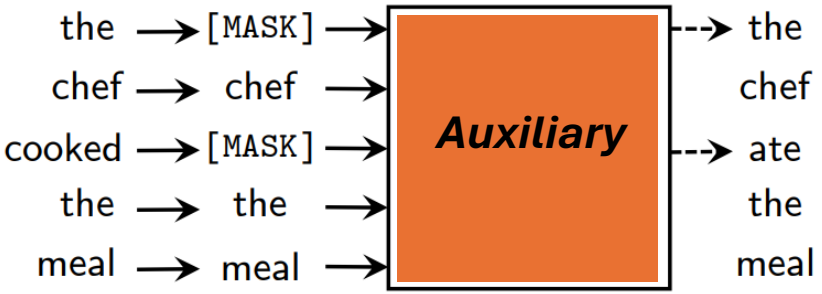
Difficulty in Scaling up ELECTRA-style Pretraining

- The auxiliary model is jointly trained with the main language model
- Why difficult?
  - High peak memory cost
  - Wasted computation on the auxiliary model
  - Sensitive to the balance between optimizations of the auxiliary model and main model
  - Unstable training

Fast-ELECTRA

- Decouple the trainings of the auxiliary model and main model
  1. Train the auxiliary model, or use an existing language model

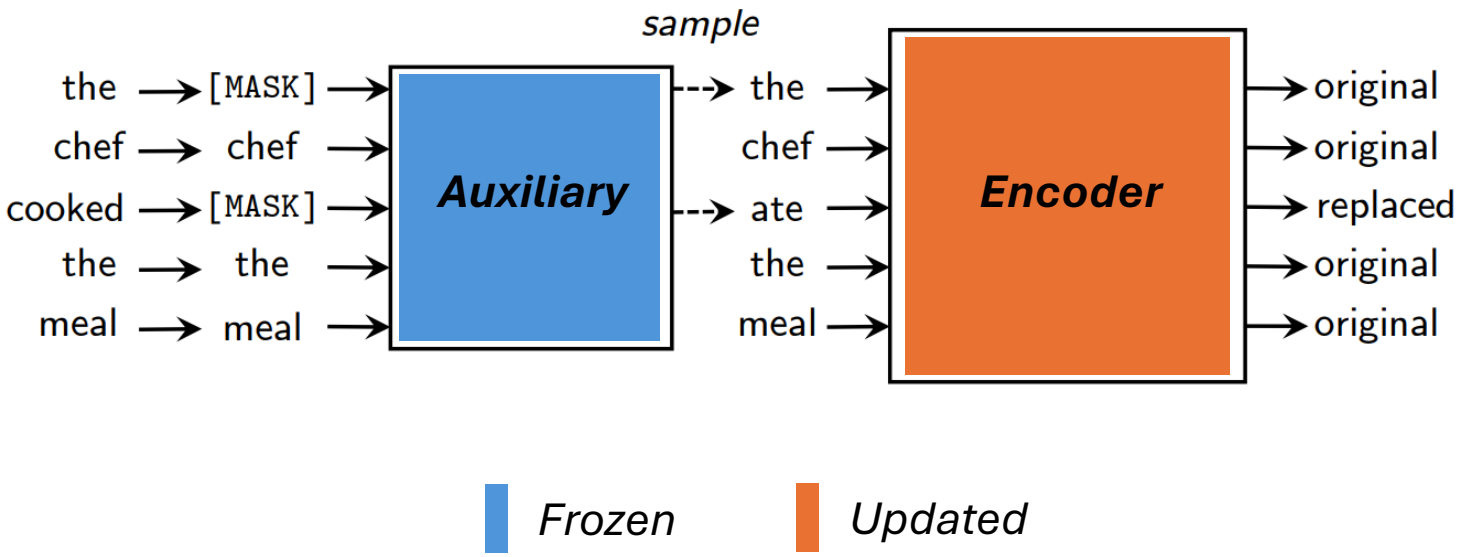
$$\min_{\theta_{\text{aux}}} L_{\text{MLM}}(\theta_{\text{aux}})$$



Fast-ELECTRA

- Decouple the trainings of the auxiliary model and main model
  1. Train the auxiliary model, or use an existing language model
  2. Train the main model

$$\min_\theta L_{\text{RTD}}(\theta)$$



Fast-ELECTRA

- Decouple the trainings of the auxiliary model and main model
- Construct a learning curriculum for the main model by “simulated annealing”

$$\hat{x}_i \sim \text{Softmax}\left(\frac{\log p_{\text{aux}}(\cdot|i, \boldsymbol{x}_{\text{masked}})}{T}\right)$$

$$T = 1 + (T_0 - 1) \cdot \exp(-u/\tau)$$

Fast-ELECTRA reduces the peak memory

- The auxiliary model is only used for inference when training the main model

Model	Method	Memory (GB)		
		Main	Auxiliary	Total
Base	Original	23.0	7.0	30.0
	Fast-ELECTRA	23.0	0.25	23.3
	Ratio	1.0	0.04	0.77
Large	Original	60.2	14.4	74.6
	Fast-ELECTRA	60.2	0.42	60.6
	Ratio	1.0	0.03	0.81

Fast-ELECTRA avoids wasted computation

- The auxiliary model only needs to be trained for once

Model	Method	Computation (GFLOPs)		
		Main	Auxiliary	Total
Base	Original	591.9	398.6	990.5
	Fast-ELECTRA	591.9	132.9	724.8
	Ratio	1.0	0.33	0.73
Large	Original	1407.7	653.9	2061.6
	Fast-ELECTRA	1407.7	218.0	1625.6
	Ratio	1.0	0.33	0.79

Fast-ELECTRA is less sensitive to hyperparameters

- Less sensitive to the choice of the auxiliary model size
- Less sensitive to the learning curriculum

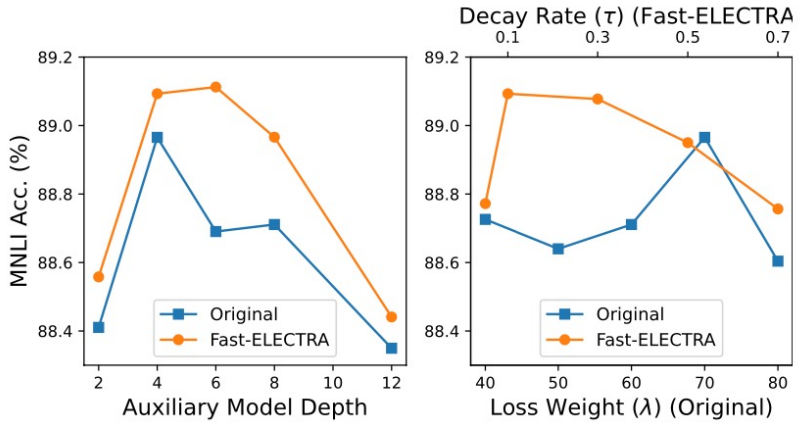


Figure 2: Downstream task performance (MNLI accuracy, Avg m/mm) versus the depth of the auxiliary model (Left) and the hyper-parameters (Right) used in the "Original" ELECTRA or "Fast-ELECTRA".

Fast-ELECTRA is more stable in training

- Fast-ELECTRA now supports larger learning rates without training divergence

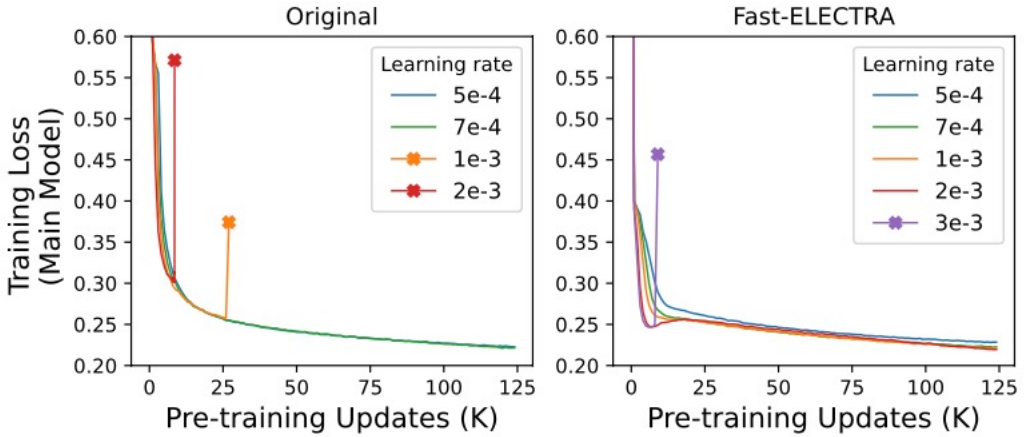


Figure 3: Training loss curves of the main model when pre-training with the original ELECTRA (*Left*) and Fast-ELECTRA (*Right*). "×" indicates the divergence of the training.

Fast-ELECTRA is as effective

Table 1: Results on GLUE development set. "-" indicates that no public reports are available. "†" indicates the model is pre-trained for 1M updates with batch size of 256. "‡" indicates the model is pre-trained for 100K updates with batch size of 8K.

Model	MNLI-(m/mm) (Acc.)	QQP (Acc.)	QNLI (Acc.)	SST-2 (Acc.)	CoLA (Mat. Corr.)	RTE (Acc.)	MRPC (Acc.)	STS-B (Spear. Corr.)	Average Score
Base Setting									
BERT (Devlin et al., 2019)	84.5/ -	91.3	91.7	93.2	58.9	68.6	87.3	89.5	83.1
RoBERTa (Liu et al., 2019b)	85.8/85.5	91.3	92.0	93.7	60.1	68.2	87.3	88.5	83.3
XLNet (Yang et al., 2019)	85.8/85.4	-	-	92.7	-	-	-	-	-
DeBERTa (He et al., 2020)	86.3/86.2	-	-	-	-	-	-	-	-
TUPE (Ke et al., 2020)	86.2/86.2	91.3	92.2	93.3	63.6	73.6	89.9	89.2	84.9
ELECTRA (Clark et al., 2020)	86.9/86.7	91.9	92.6	93.6	66.2	75.1	88.2	89.7	85.5
MC-BERT (Xu et al., 2020)	85.7/85.2	89.7	91.3	92.3	62.1	75.0	86.0	88.0	83.7
COCO-LM (Meng et al., 2021)	88.5/88.3	92.0	93.1	93.2	63.9	84.8	**91.4**	90.3	87.2
AMOS (Meng et al., 2022)	88.9/88.7	**92.3**	93.6	94.2	**70.7**	**86.6**	90.9	**91.6**	88.6
DeBERTaV3 (He et al., 2021)	**89.3/89.0**	-	-	-	-	-	-	-	-
METRO (Bajaj et al., 2022)	89.0/88.8	92.2	93.4	**95.0**	70.6	86.5	91.2	91.2	88.6
$\text{METRO}_{\text{ReImp}}$	89.0/88.9	92.0	93.4	94.4	70.1	86.3	**91.4**	91.2	88.5
Fast-ELECTRA	89.4/88.8	92.1	**93.8**	94.5	**71.4**	85.6	**91.4**	**91.6**	**88.7**
Large Setting									
BERT†	86.6/ -	-	-	-	-	-	-	-	-
RoBERTa‡	89.0/ -	91.9	93.9	**95.3**	66.3	84.5	90.2	91.6	87.8
XLNet†	88.4/ -	91.8	93.9	94.4	65.2	81.2	90.0	91.1	87.0
TUPE†	88.2/88.2	91.7	93.6	95.0	67.5	81.7	90.1	90.7	87.3
$\text{METRO}_{\text{ReImp}}$	89.9/90.2	**92.5**	**94.5**	94.3	69.7	**88.8**	**91.9**	91.6	89.2
Fast-ELECTRA	**90.1/90.2**	92.4	**94.5**	95.1	**72.1**	87.4	90.7	**91.9**	**89.3**