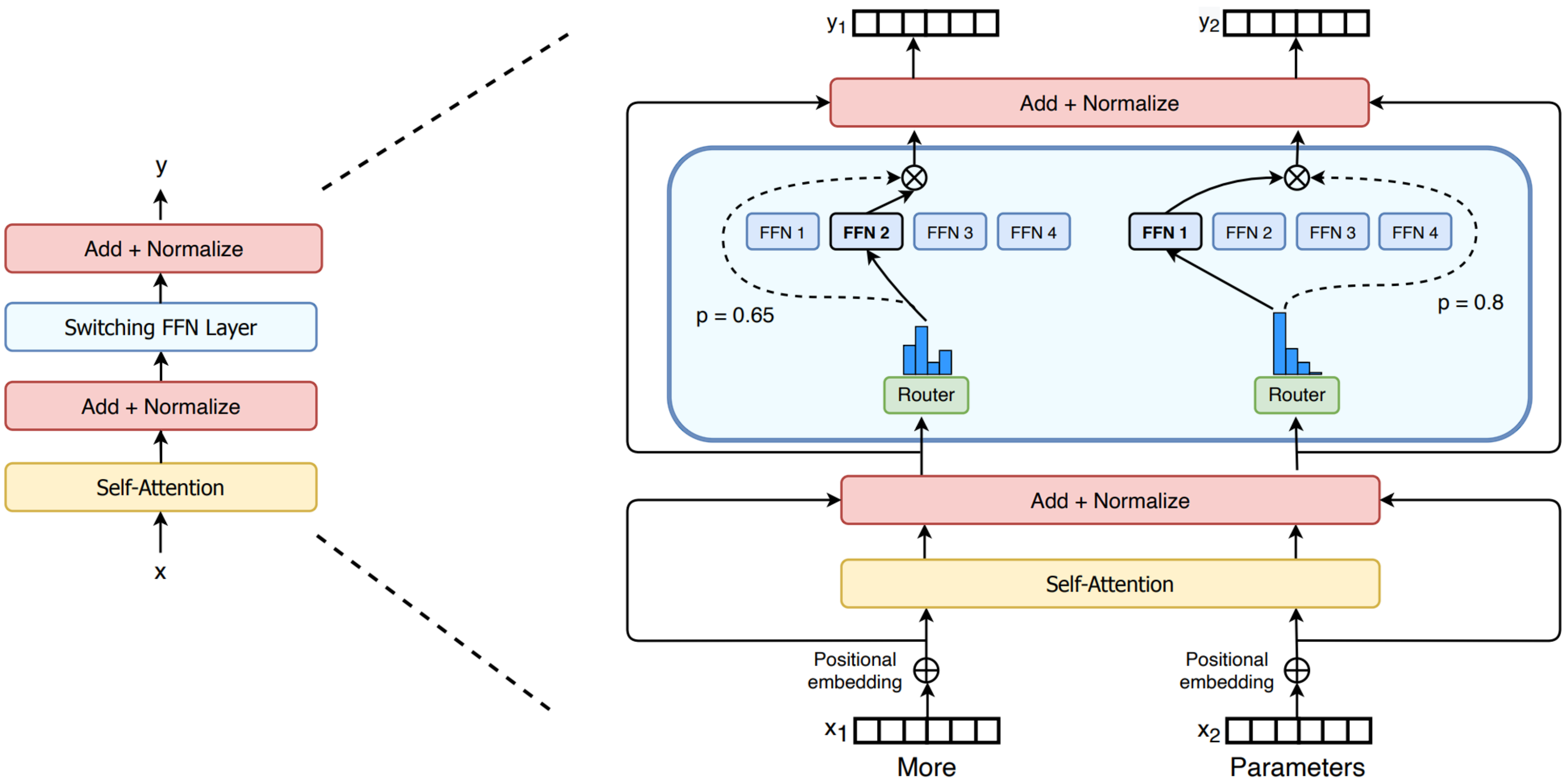
Sparse Backpropagation for MoE Training

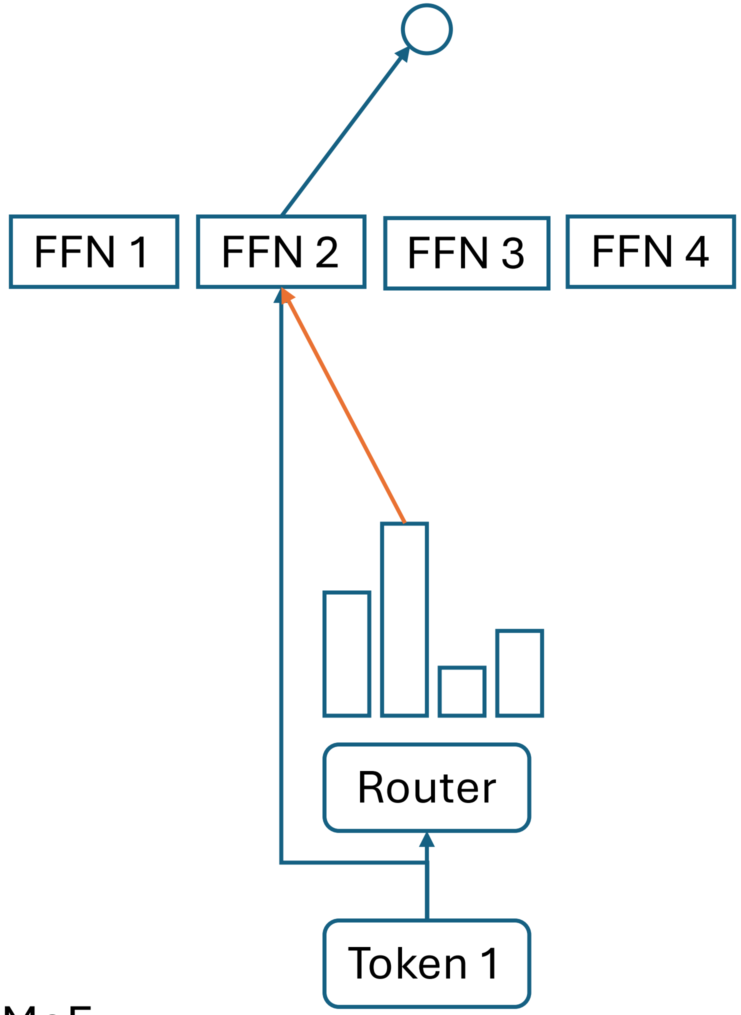
Liyuan Liu[§], Jianfeng Gao[§], Weizhu Chen[‡]

[§]Microsoft Research [‡]Microsoft Azure AI

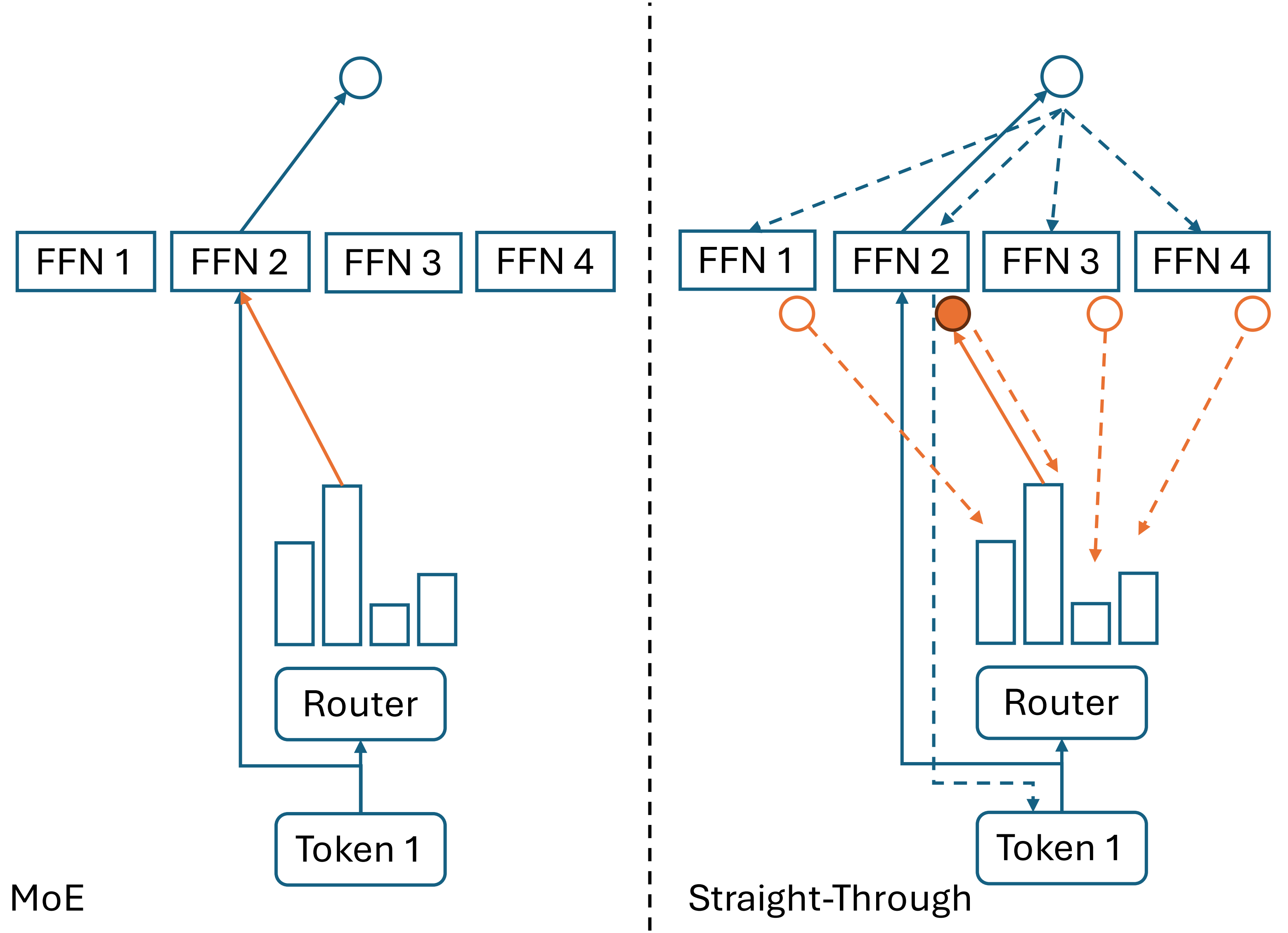
Mixture-of-Expert



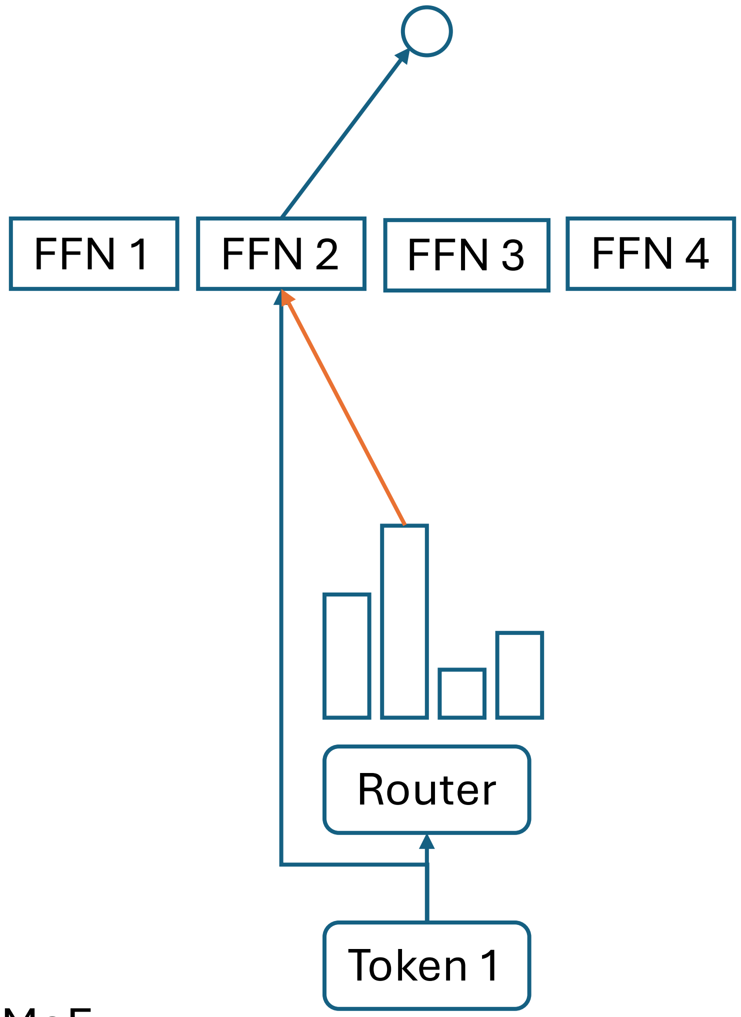
Gradient Estimation for Expert Routing



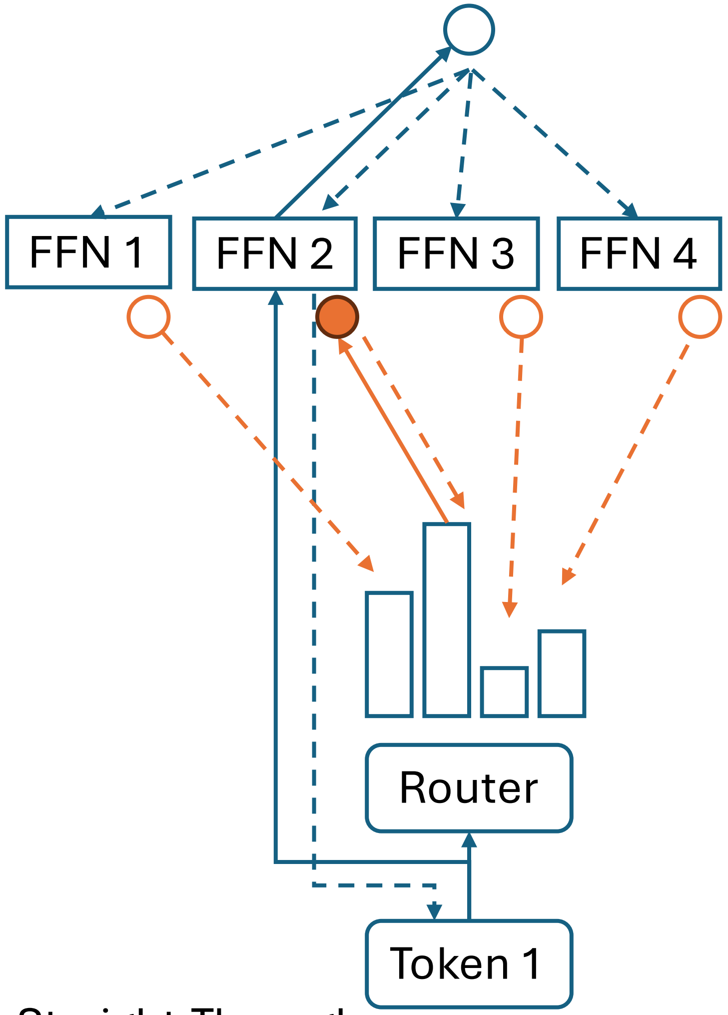
Gradient Estimation for Expert Routing



Gradient Estimation for Expert Routing



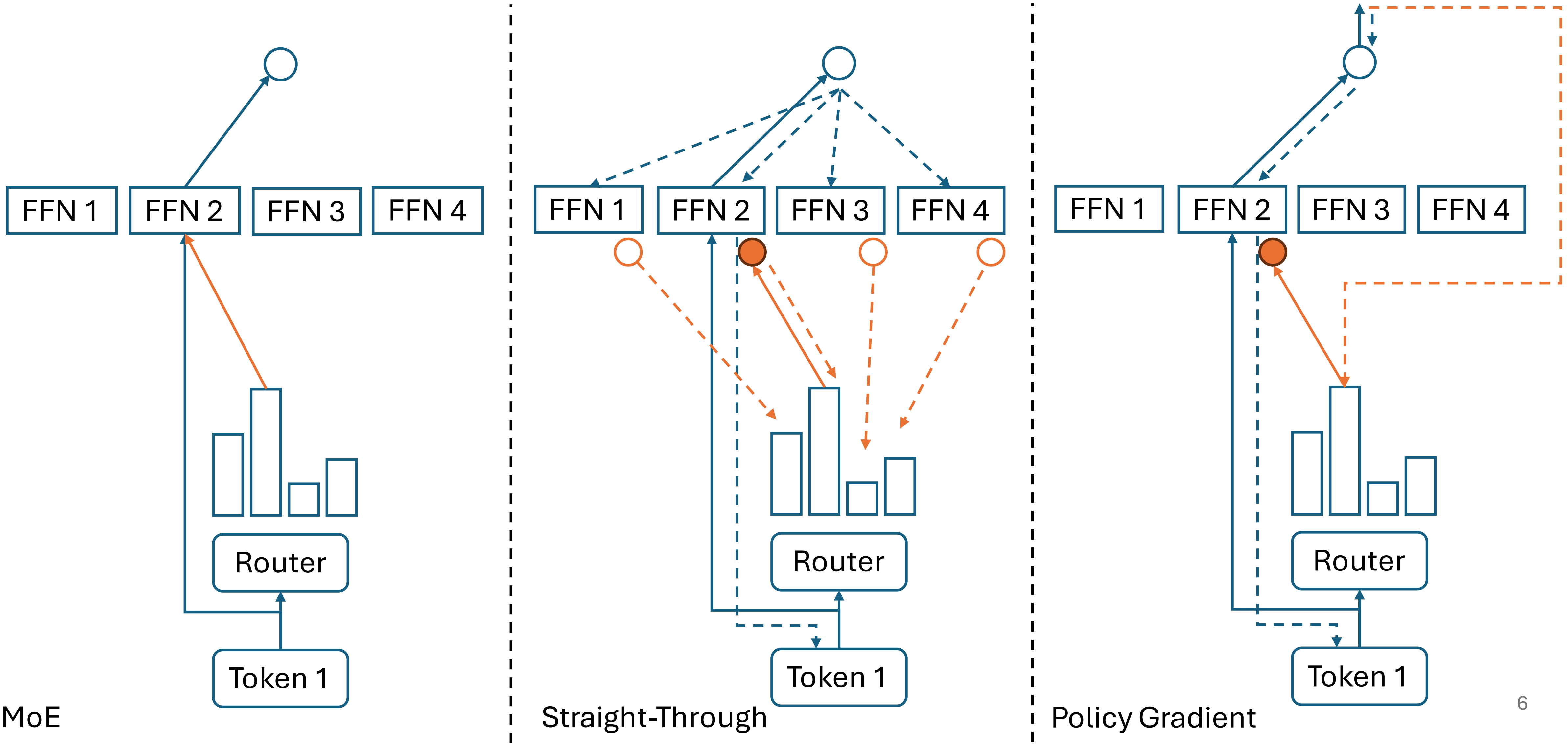
MoE



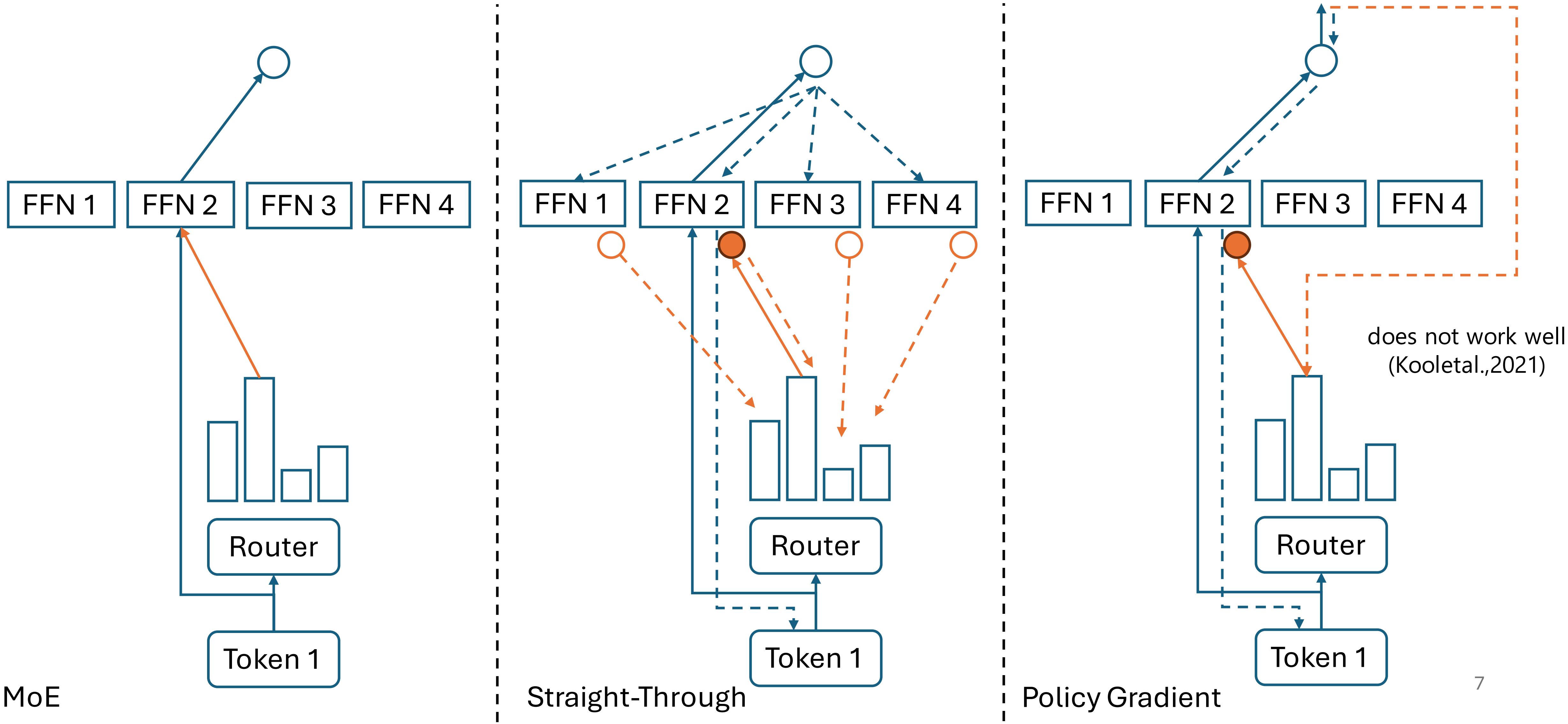
Straight-Through

Applying Straight-Through to expert routing necessitates the output of all experts

Gradient Estimation for Expert Routing



Gradient Estimation for Expert Routing

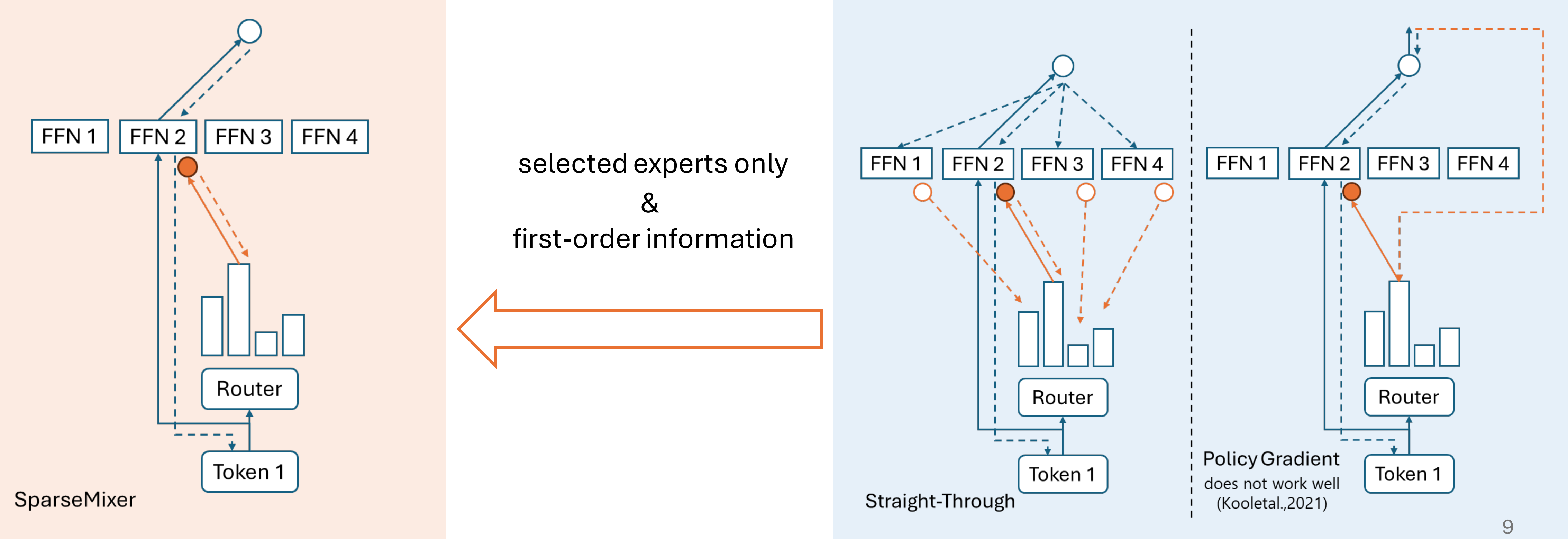


Backpropagation Made Sparse

We propose SparseMixer to provide sound gradient estimations for expert routing, without requiring outputs from all experts

Backpropagation Made Sparse

We propose SparseMixer to provide sound gradient estimations for expert routing, without requiring outputs from all experts



Underlying Mechanism of Switch Transformer

Underlying Mechanism of Switch Transformer

> Expert Sampling during Training

Underlying Mechanism of Switch Transformer

> Expert Sampling during Training

Softmax

$\mathrm{D} \sim \mathrm{softmax}(\theta)$

$\mathrm{D} \sim \mathrm{argmax}_{I_i}\, \theta_{I_i} + G_{I_i}$,
where $G_{I_i} \overset{\text{iid}}{\sim} \mathrm{Gumbel}(0,1)$

Switch Transformer

$\mathrm{D} = \mathrm{argmax}_{I_i}\, \theta_{I_i} \cdot u_{I_i}$, where
$u_{I_i} \overset{\text{iid}}{\sim} \mathrm{Uniform}(1 - r, 1 + r)$

Underlying Mechanism of Switch Transformer

> Expert Sampling during Training

Softmax

$\mathrm{D} \sim \mathrm{softmax}(\theta)$

$\mathrm{D} \sim \mathrm{argmax}_{I_i}\, \theta_{I_i} + G_{I_i}$,
where $G_{I_i} \overset{\text{iid}}{\sim} \mathrm{Gumbel}(0,1)$

Switch Transformer

$\mathrm{D} = \mathrm{argmax}_{I_i}\, \theta_{I_i} \cdot u_{I_i}$, where
$u_{I_i} \overset{\text{iid}}{\sim} \mathrm{Uniform}(1 - r, 1 + r)$

⇩ Important Property

Mark $\theta^* \coloneqq \max_{I_i} \theta_{I_i}$, then I_i will <u>never be</u> sampled if
$\theta^* - \theta_{I_i} > r \cdot (|\theta^*| + |\theta_{I_i}|)$

Underlying Mechanism of Switch Transformer

> Expert Sampling during Training

Softmax

$$\mathrm{D} \sim \mathrm{softmax}(\theta)$$

$$\mathrm{D} \sim \mathrm{argmax}_{I_i}\, \theta_{I_i} + G_{I_i},$$
where $G_{I_i} \overset{\text{iid}}{\sim} \mathrm{Gumbel}(0,1)$

Switch Transformer

$\mathrm{D} = \mathrm{argmax}_{I_i}\, \theta_{I_i} \cdot u_{I_i}$, where
$u_{I_i} \overset{\text{iid}}{\sim} \mathrm{Uniform}(1 - r, 1 + r)$

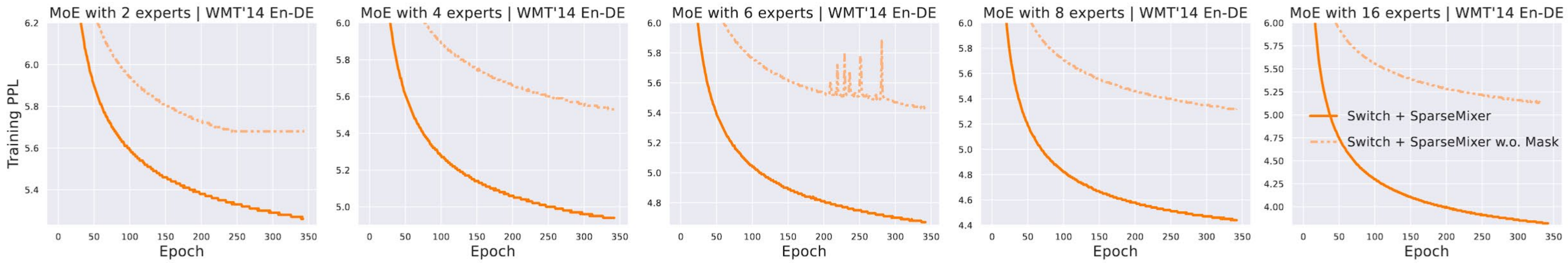
Important Property

Mark $\theta^* \coloneqq \max_{I_i} \theta_{I_i}$, then I_i will never be sampled if
$\theta^* - \theta_{I_i} > r \cdot (|\theta^*| + |\theta_{I_i}|)$

Adapt

Masked-Softmax

$$\mathrm{D} \sim \frac{\exp(\theta_i) \cdot \Delta_i}{\sum_j \exp(\theta_j) \cdot \Delta_j} \qquad \Delta_i = \delta(\theta^* - \theta_{I_i} > r \cdot (|\theta^*| + |\theta_{I_i}|))$$



Underlying Mechanism of Switch Transformer

> Expert Sampling during Training → prefers a “margin-like” distribution

Underlying Mechanism of Switch Transformer

> Expert Sampling during Training → prefers a “margin-like” distribution

> Expert Output Scaling

In Switch Transformer $\mathrm{G_D} \leftarrow \pi_{\mathrm{D}} \cdot \mathrm{g_D}(\mathrm{x})$

Underlying Mechanism of Switch Transformer

> Expert Sampling during Training → prefers a "margin-like" distribution

> Expert Output Scaling

In Switch Transformer $G_D \leftarrow \pi_D \cdot g_D(x)$

Why we need π_D? $G_D \leftarrow \pi_D \cdot g_D(x)$ or $G_D \leftarrow g_D(x)$?

Underlying Mechanism of Switch Transformer

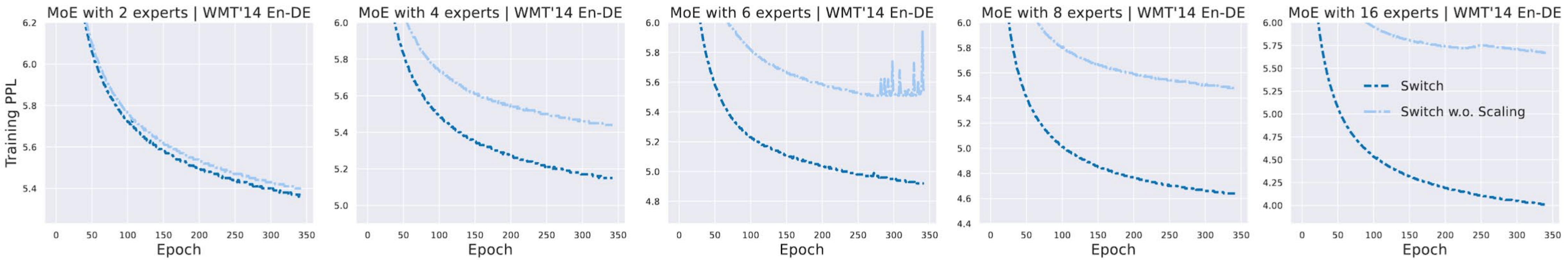
> Expert Sampling during Training → prefers a "margin-like" distribution

> Expert Output Scaling

In Switch Transformer $\quad G_D \leftarrow \pi_D \cdot g_D(x)$

Why we need π_D? $G_D \leftarrow \pi_D \cdot g_D(x)$ or $G_D \leftarrow g_D(x)$?

With the masked-softmax parameterization, π_D scaling leads to a dynamic learning rate adaptation on expert model learning.



Underlying Mechanism of Switch Transformer

> Expert Sampling during Training → prefers a “margin-like” distribution

> Expert Output Scaling → acts as a dynamic learning rate adapter

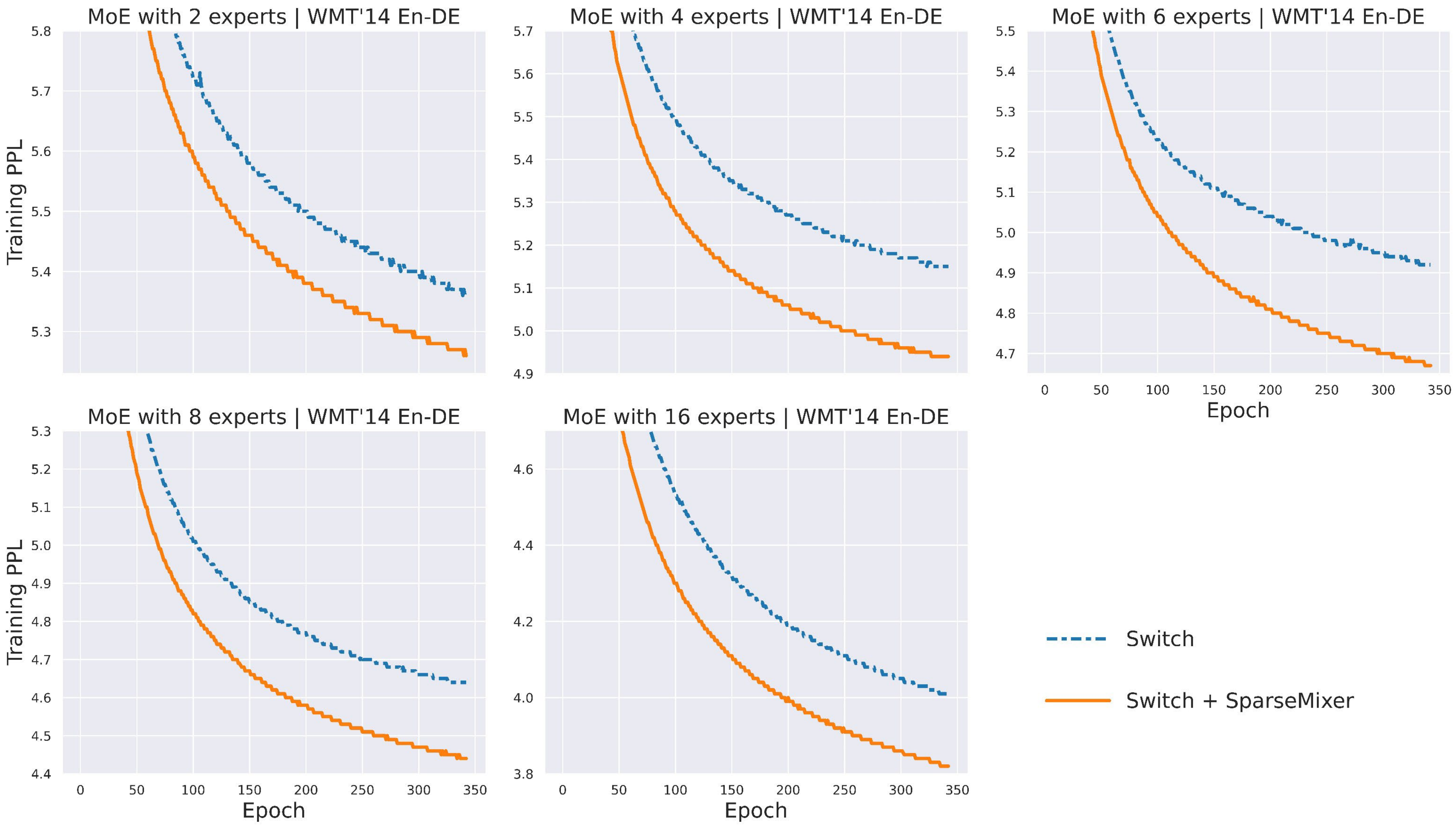
Underlying Mechanism of Switch Transformer

> Expert Sampling during Training → prefers a “margin-like” distribution

> Expert Output Scaling → acts as a dynamic learning rate adapter

Applying SparseMixer on Switch Transformer Training

Neural Machine Translation



Neural Machine Translation

Table 1: BLEU score on WMT'14 En-De (N refers to the number of experts).

	Dense	Mixture-of-Expert				
		$N = 2$	$N = 4$	$N = 6$	$N = 8$	$N = 16$
Transformer-base	28.33	/	/	/	/	/
Switch	/	28.17	28.05	27.96	27.99	27.81
Switch+SparseMixer	/	**28.72**	**28.61**	**28.32**	**28.12**	**28.08**

Electra Pretraining

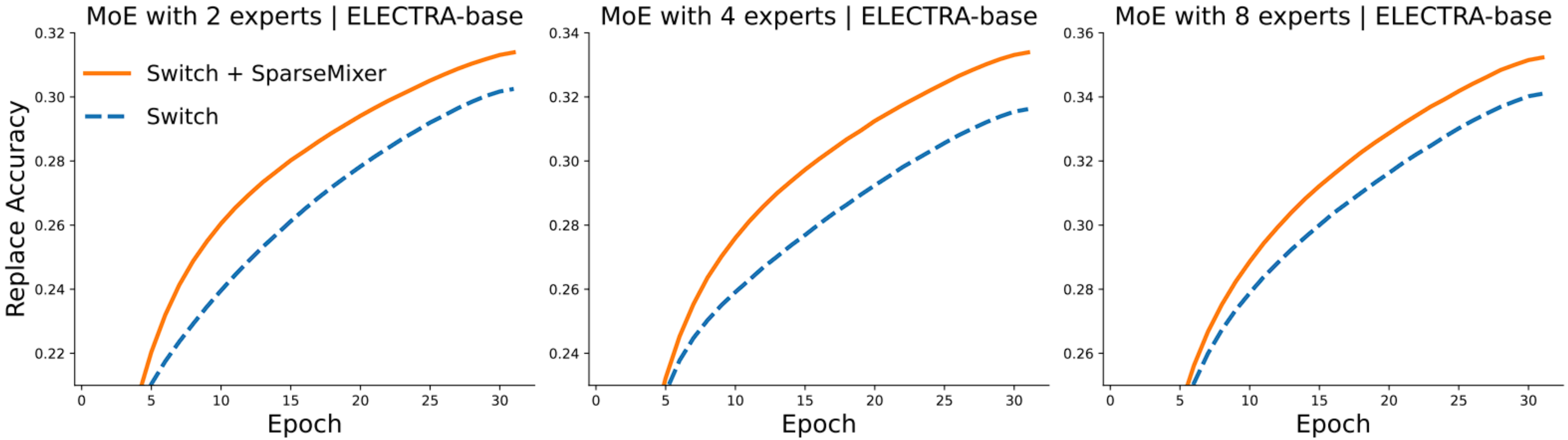


Figure 2: Training curves of Switch Transformer on ELECTRA-base training.

Pretraining

Table 2: Results on the GLUE development set. S refers to Switch and S+S refers to Switch+SparseMixer. AVG is the average score across eight tasks.

N	Model	AVG	MNLI-(m/mm) (Acc.)	QQP (Acc.)	QNLI (Acc.)	SST-2 (Acc.)	CoLA (Mat. Corr.)	RTE (Acc.)	MRPC (Acc.)	STS-B (Spear. Corr.)
1	Dense	87.37	88.72/88.40	91.90	93.36	93.35	68.71	82.31	89.95	90.83
2	S	87.62	88.55/88.34	91.86	93.52	94.27	67.90	83.76	90.69	90.52
	S+S	88.31	89.06/88.78	91.98	93.54	94.38	69.96	85.20	91.67	90.81
4	S	87.02	88.12/88.40	91.73	93.21	93.92	70.89	77.26	90.44	90.49
	S+S	87.63	88.97/88.41	91.92	93.54	94.04	71.00	80.87	90.69	90.72
8	S	87.27	88.43/88.22	91.78	93.23	94.84	68.06	80.87	90.44	90.62
	S+S	87.71	88.69/88.47	92.03	93.41	94.15	69.00	83.76	89.95	90.81

Take Aways

SparseMixer provides sound gradient estimations for expert routing, without requiring outputs from all experts

Our study sheds insights into the mechanism of switch transformer