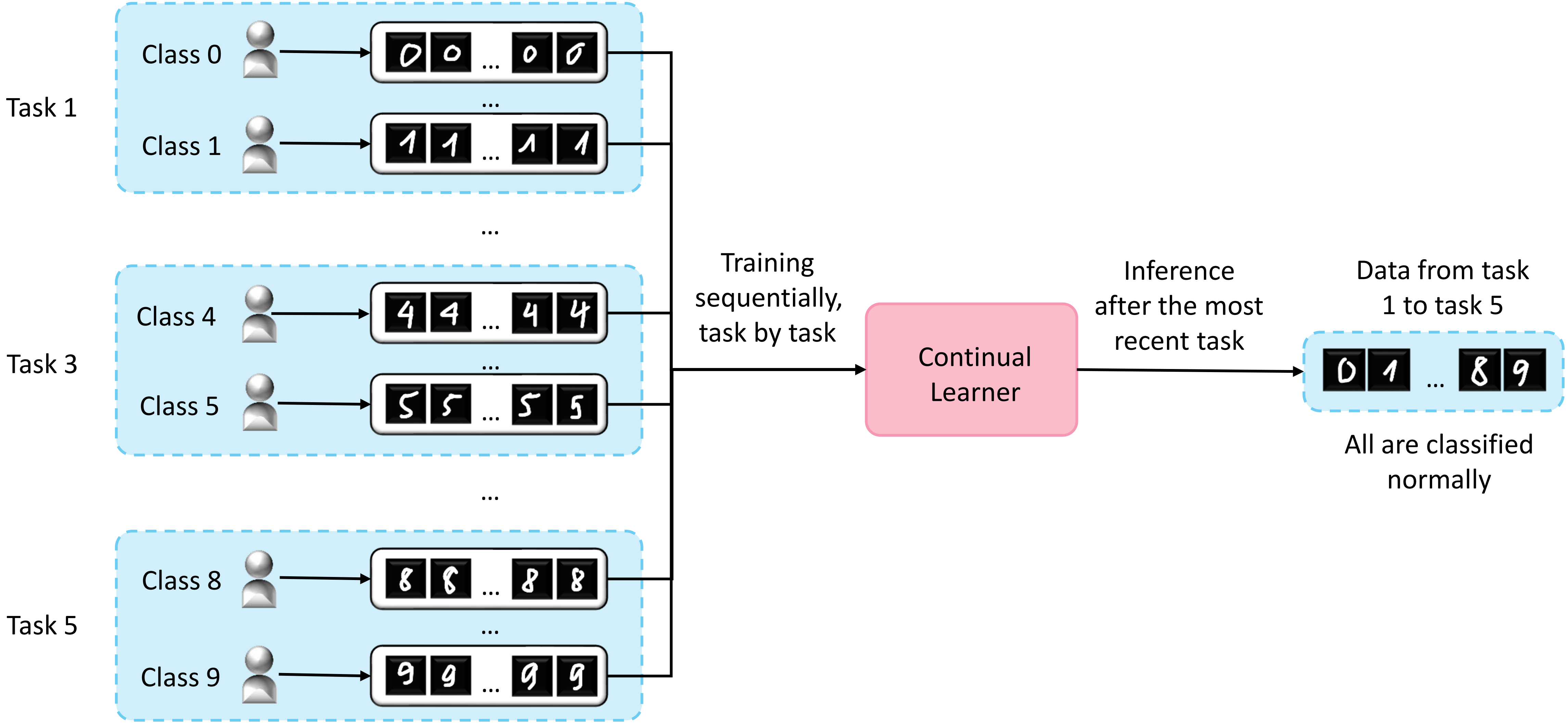
Mixture of experts meets Prompt-based Continual Learning

Minh Le[3], An Nguyen[2]*, Huy Nguyen[1]*, Trang Nguyen[3]*,

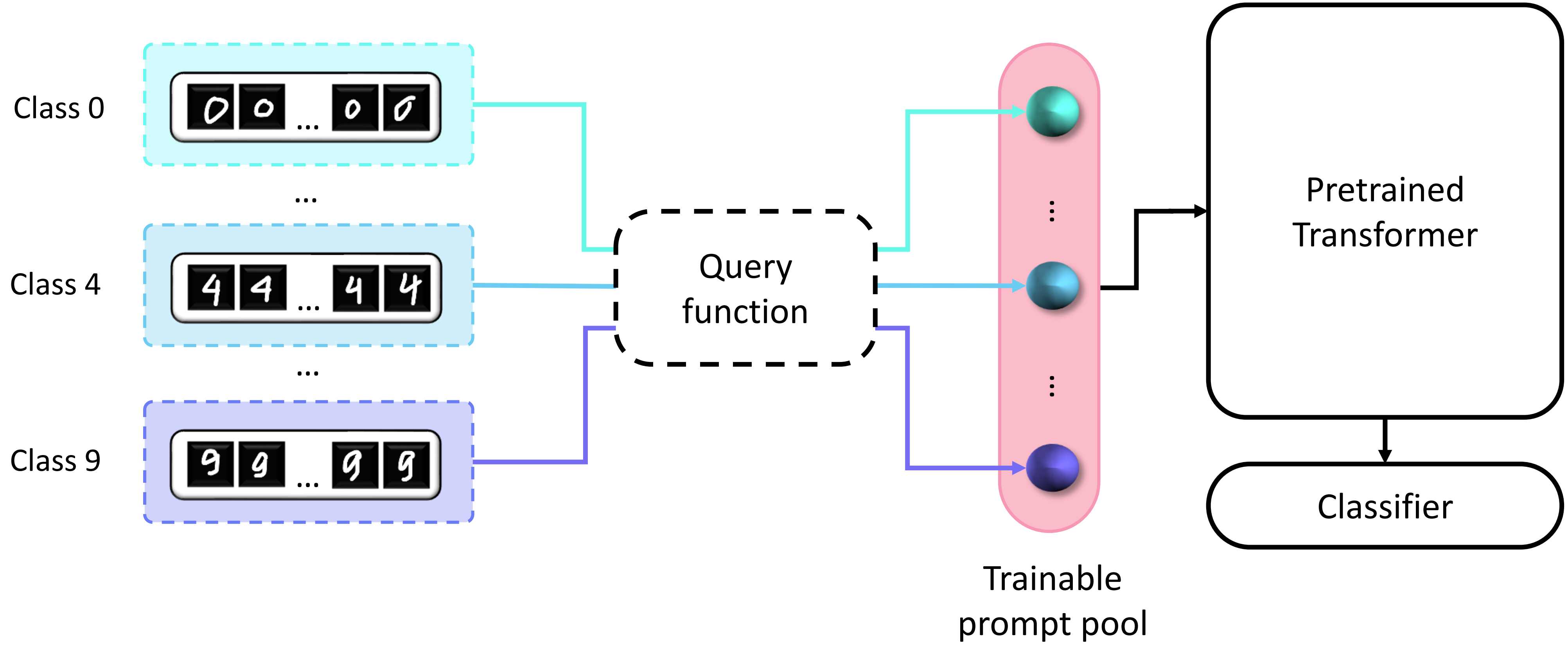
Trang Pham[3]*, Linh Van Ngo[2], Nhat Ho[1]

[1] University of Texas at Austin
[2] Hanoi University of Science and Technology
[3] VinAI Research

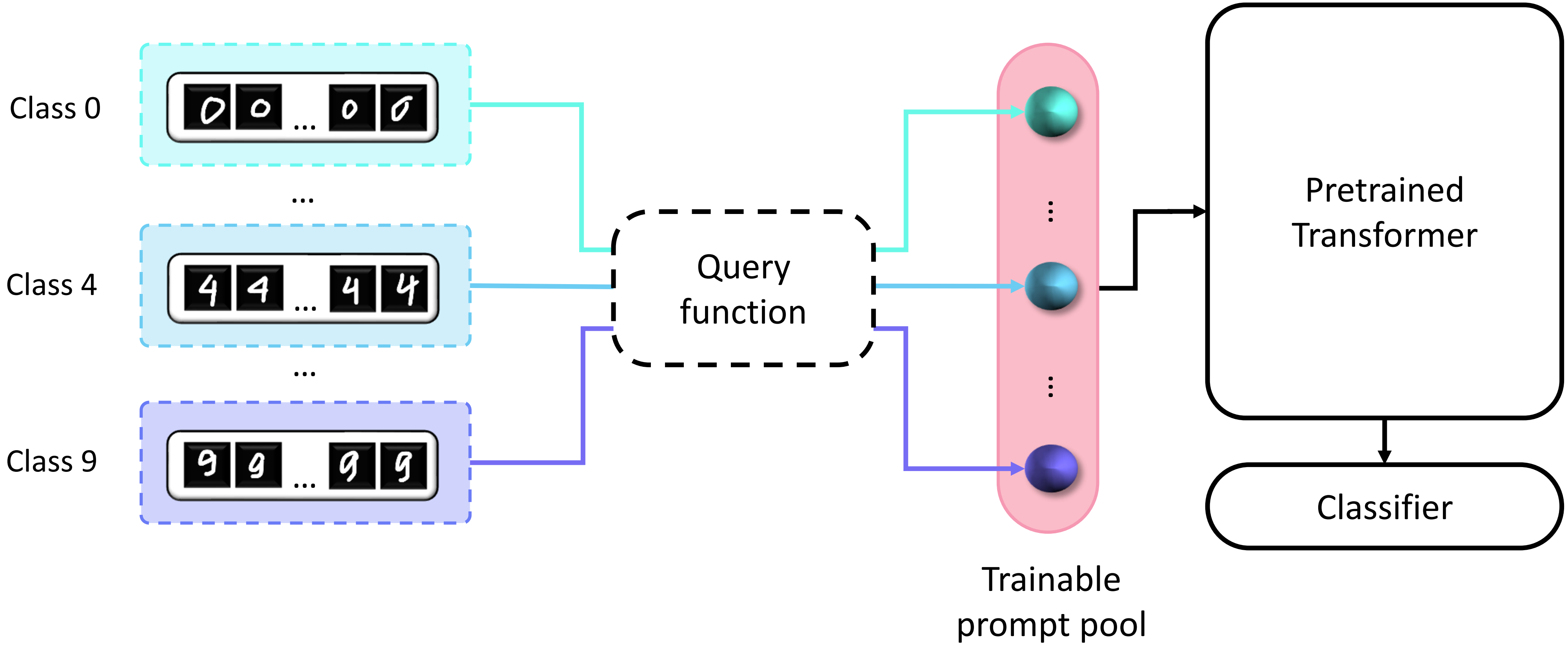
Continual Learning



Prompt-based Continual Learning



Prompt-based Continual Learning



Efficient, astonishing performance

BUT LACKS A THEORETICAL FOUNDATION!

Mixture of Experts

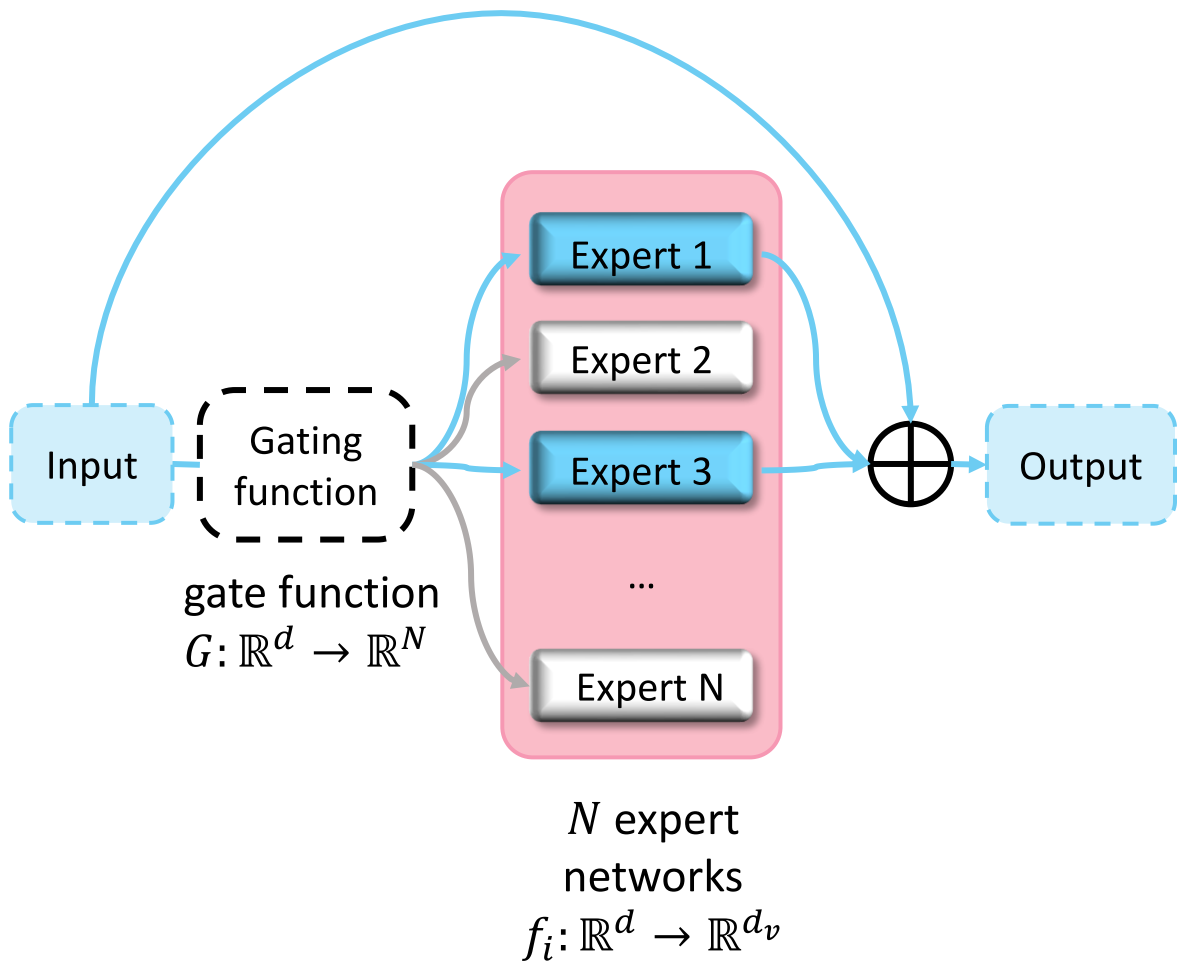
An MoE model consists of:

- a group of N expert networks $f_i: \mathbb{R}^d \to \mathbb{R}^{d_v}$
- a gate function $G: \mathbb{R}^d \to \mathbb{R}^N$
- a learned score function $s_i: \mathbb{R}^d \to \mathbb{R}$

Given an input $h \in \mathbb{R}^d$, its MoE output is computed as:

$$y \coloneqq \sum_{j=1}^{N} G(\boldsymbol{h})_j \cdot f_j(\boldsymbol{h}) \coloneqq \sum_{j=1}^{N} \frac{\exp\left(s_j(\boldsymbol{h})\right)}{\sum_{l=1}^{N} \exp(s_l(\boldsymbol{h}))} \cdot f_j(\boldsymbol{h}),$$

where $G(h) \coloneqq softmax\left(s_1(\boldsymbol{h}), \ldots, s_N(\boldsymbol{h})\right)$.



Mixture of Experts

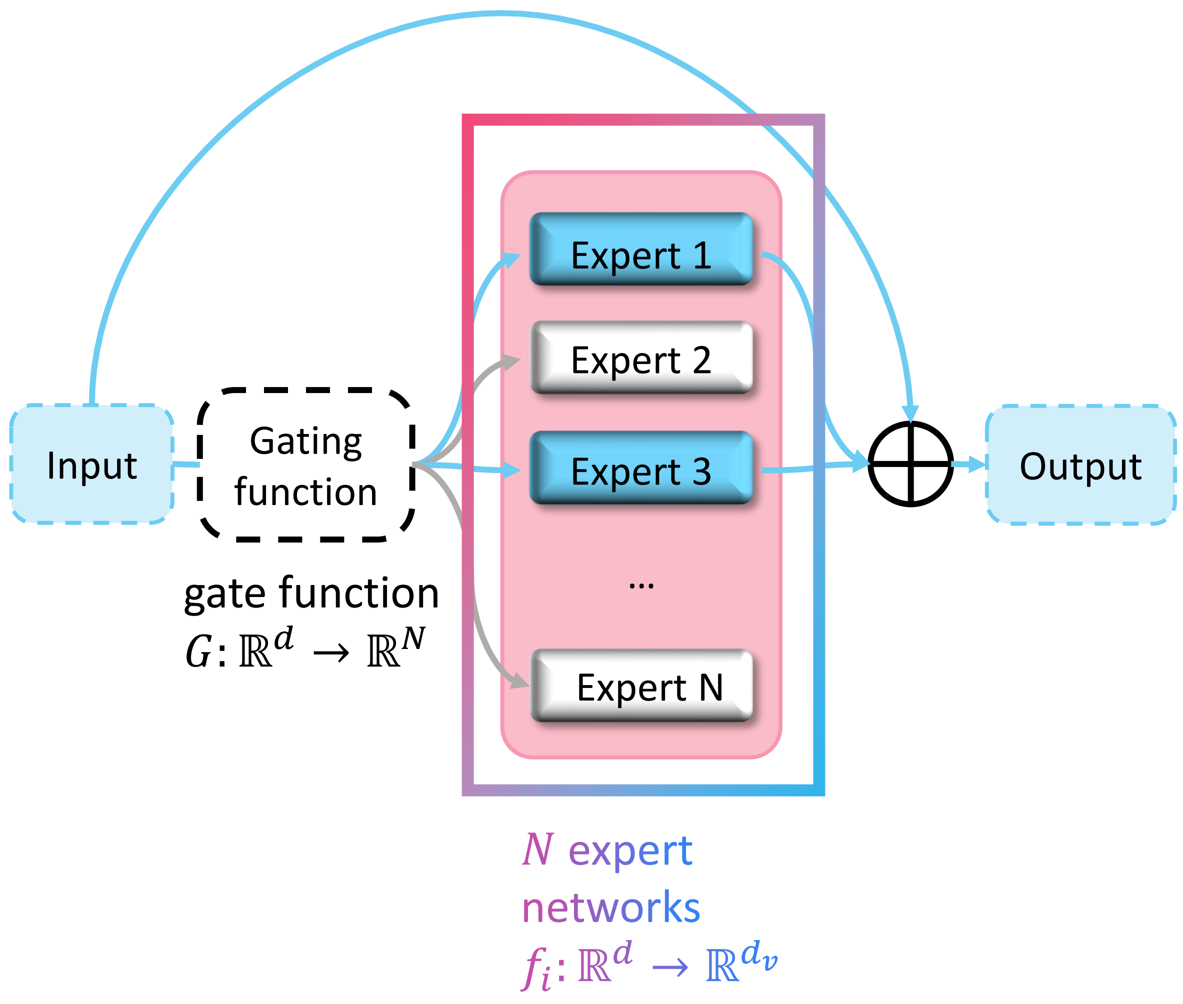
An MoE model consists of:

- a group of N expert networks $f_i: \mathbb{R}^d \to \mathbb{R}^{d_v}$, for all $i \in [N]$
- a gate function $G: \mathbb{R}^d \to \mathbb{R}^N$
- a learned score function $s_i: \mathbb{R}^d \to \mathbb{R}$

Given an input $h \in \mathbb{R}^d$, its MoE output is computed as:

$$y \coloneqq \sum_{j=1}^{N} G(\boldsymbol{h})_j \cdot f_j(\boldsymbol{h}) \coloneqq \sum_{j=1}^{N} \frac{\exp\left(s_j(\boldsymbol{h})\right)}{\sum_{l=1}^{N} \exp(s_l(\boldsymbol{h}))} \cdot f_j(\boldsymbol{h}),$$

where $G(h) \coloneqq softmax\left(s_1(\boldsymbol{h}), \ldots, s_N(\boldsymbol{h})\right)$.



Mixture of Experts

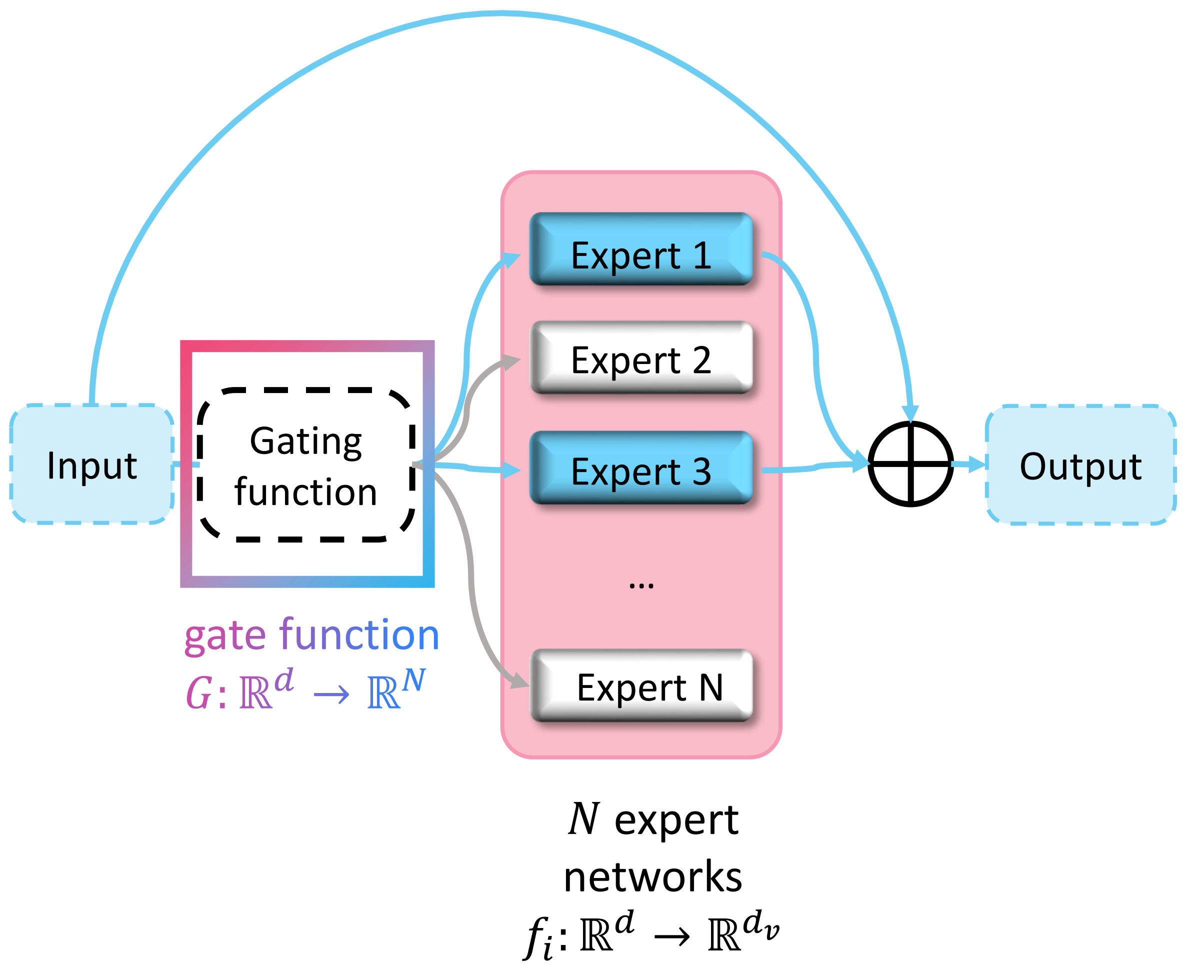
An MoE model consists of:

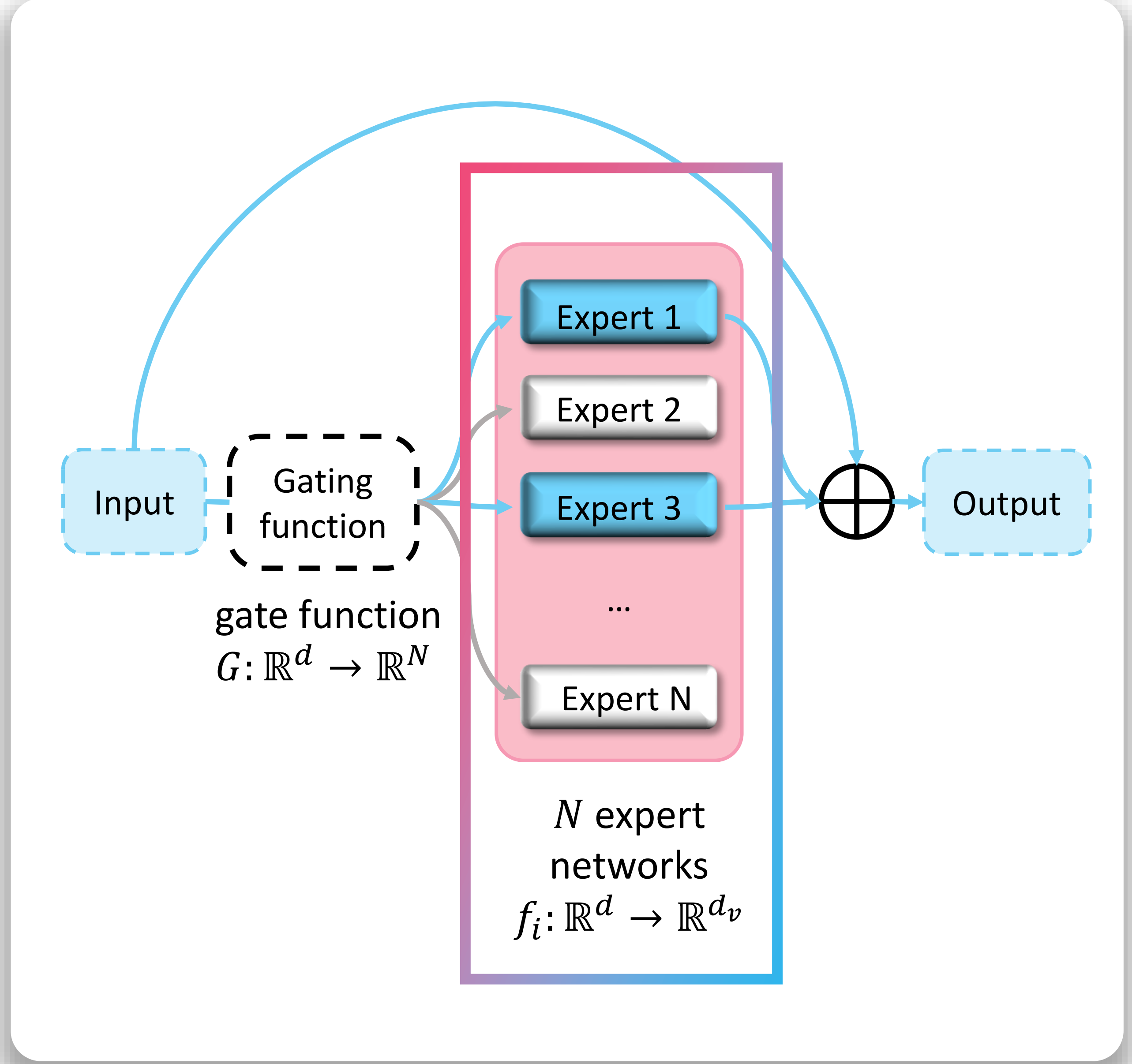
- a group of N expert networks $f_i: \mathbb{R}^d \to \mathbb{R}^{d_v}$, for all $i \in [N]$
- a gate function $G: \mathbb{R}^d \to \mathbb{R}^N$
- learned score function $s_i: \mathbb{R}^d \to \mathbb{R}$

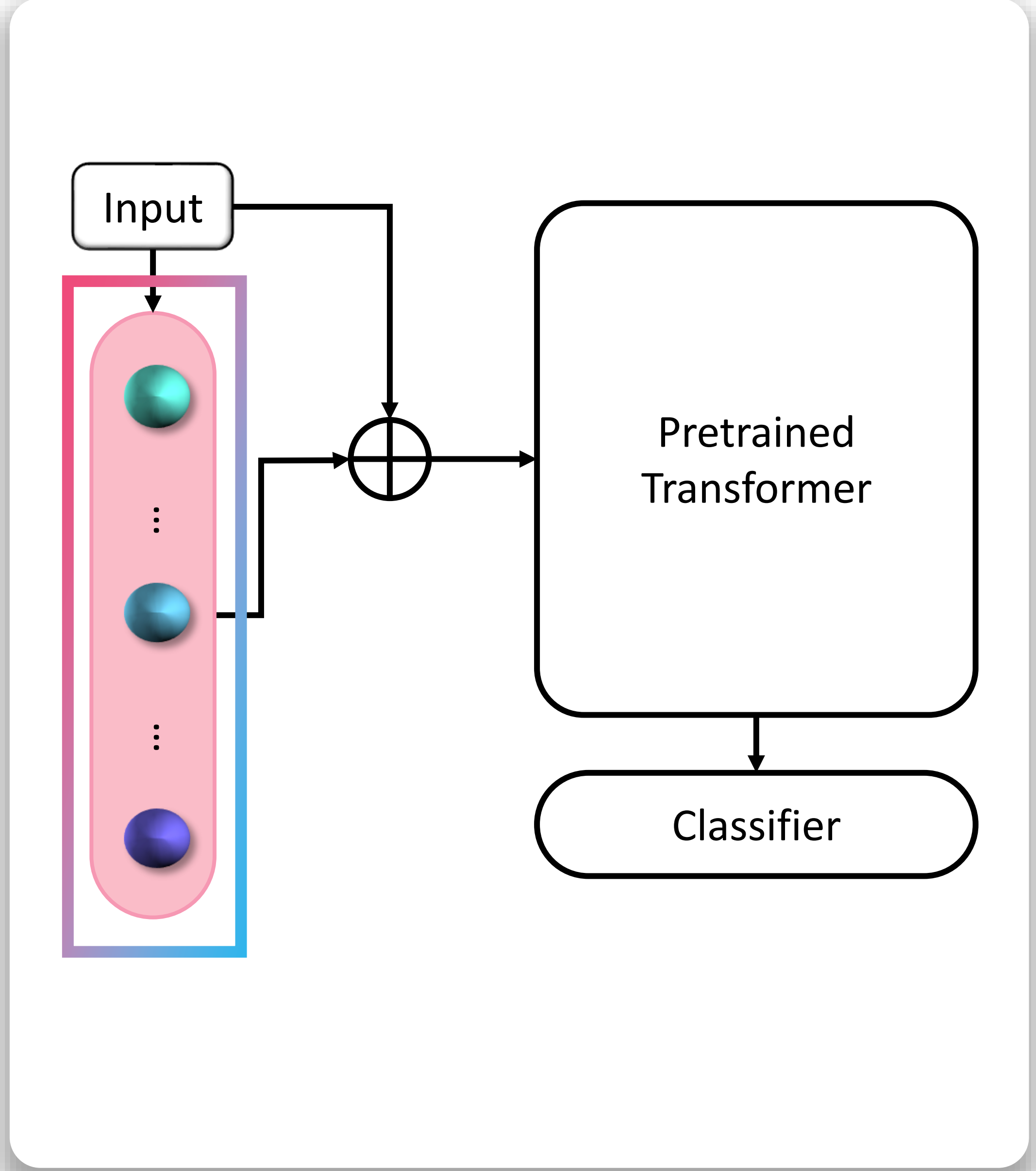
Given an input $h \in \mathbb{R}^d$, its MoE output is computed as:

$$y \coloneqq \sum_{j=1}^{N} G(\boldsymbol{h})_j \cdot f_j(\boldsymbol{h}) \coloneqq \sum_{j=1}^{N} \frac{\exp\left(s_j(\boldsymbol{h})\right)}{\sum_{l=1}^{N} \exp\left(s_l(\boldsymbol{h})\right)} \cdot f_j(\boldsymbol{h}),$$

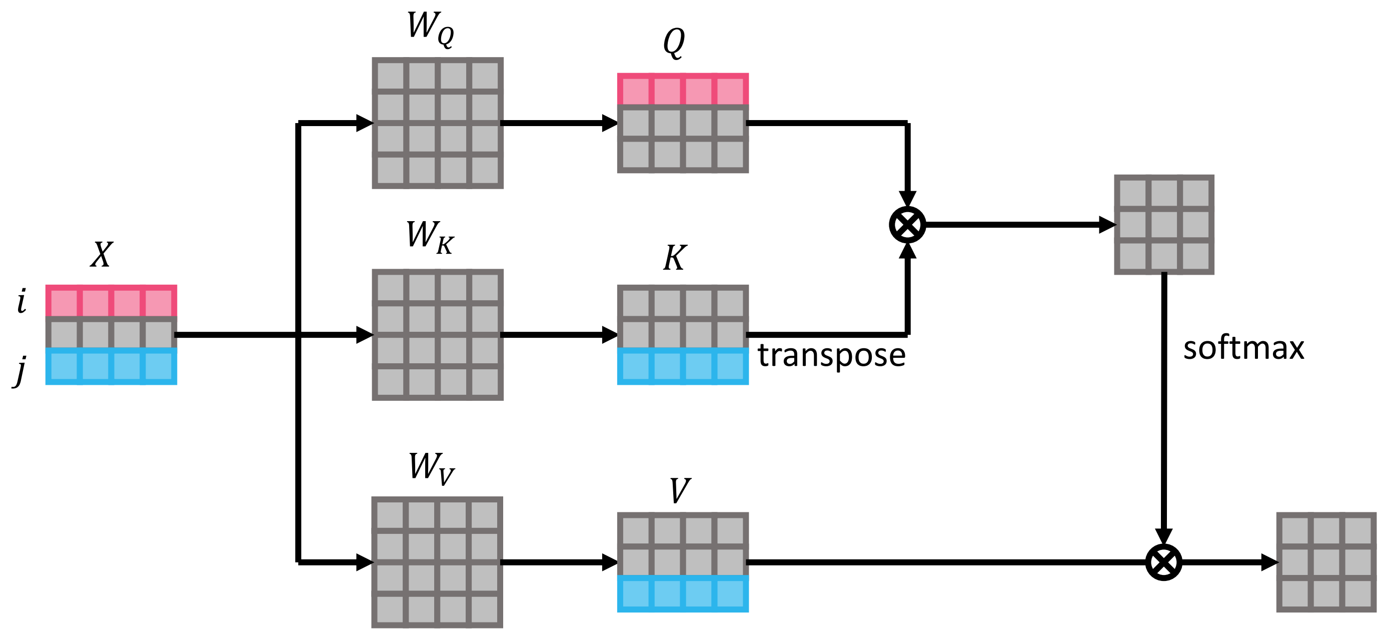
where $G(h) \coloneqq softmax\left(s_1(\mathbf{h}), \ldots, s_N(\mathbf{h})\right)$.




Input
Gating function
gate function
$G: \mathbb{R}^d \rightarrow \mathbb{R}^N$
Expert 1
Expert 2
Expert 3
...
Expert N
N expert networks
$f_i: \mathbb{R}^d \rightarrow \mathbb{R}^{d_v}$
Output

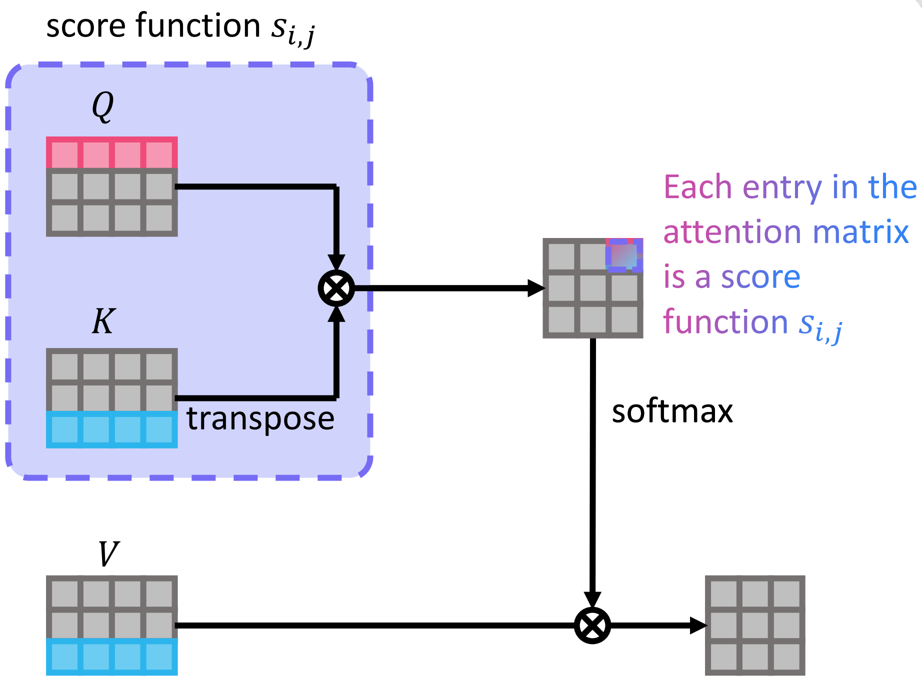

Input
Pretrained Transformer
Classifier

Mixture of Experts and Self-Attention



Attention Mechanism

Mixture of Experts and Self-Attention

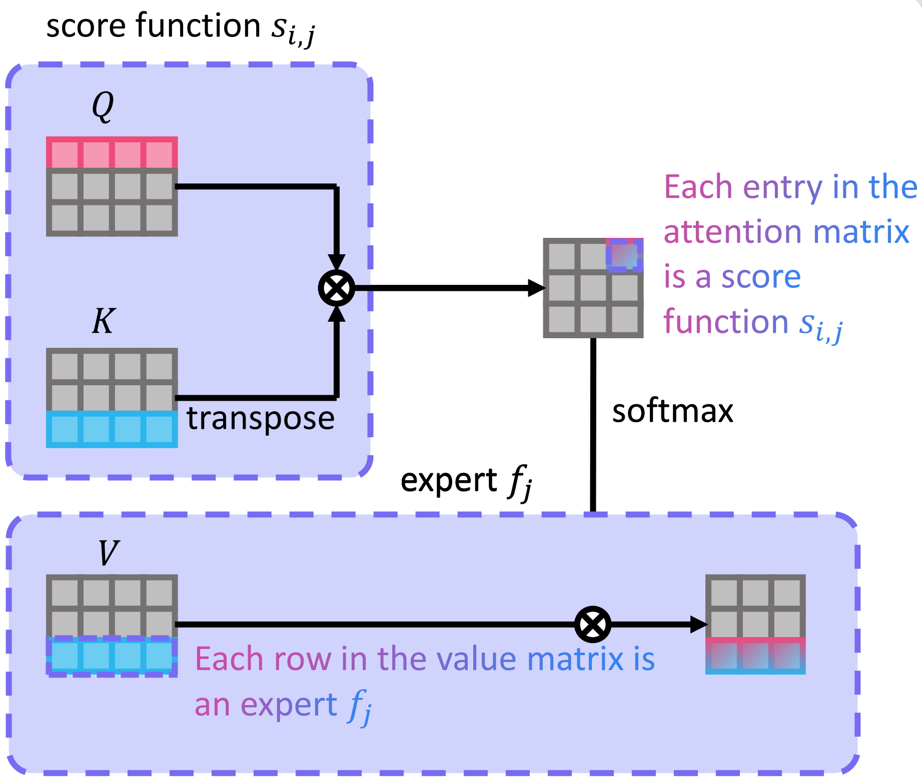


- Define N gating function G_i with the score function for the j^{th} expert of the i^{th} gating $s_{i,j}$:

$$s_{i,j}(X) := \frac{\boldsymbol{X}^\top E_i^\top W_l^Q {W_l^K}^\top E_j \boldsymbol{X}}{\sqrt{d_v}} = \frac{\boldsymbol{X}_i^\top W_l^Q {W_l^K}^\top \boldsymbol{X}_j}{\sqrt{d_v}}$$

for $i,j \in [N]$.

Mixture of Experts and Self-Attention



- Define N gating function G_i with the score function for the j^{th} expert of the i^{th} gating $s_{i,j}$:

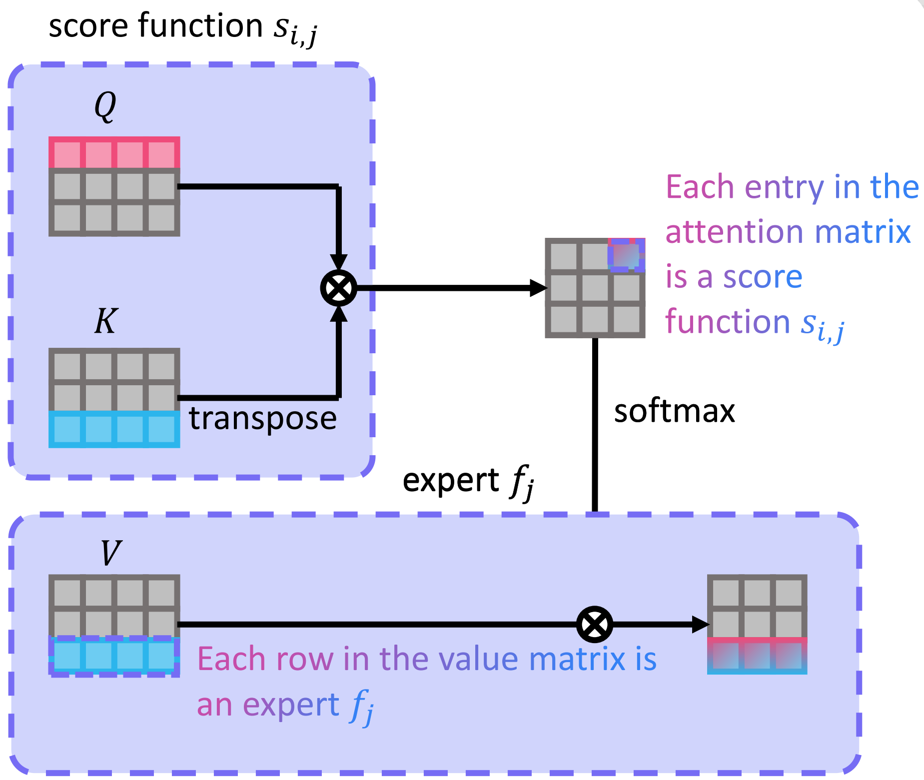
$$s_{i,j}(X) := \frac{\boldsymbol{X}^\top E_i^\top W_l^Q {W_l^K}^\top E_j \boldsymbol{X}}{\sqrt{d_v}} = \frac{\boldsymbol{x}_i^\top W_l^Q {W_l^K}^\top \boldsymbol{x}_j}{\sqrt{d_v}}$$

for $i,j \in [N]$.

- Define N experts f_j:

$$f_j(X) := {W_l^V}^\top E_j \boldsymbol{X} = {W_l^V}^\top \boldsymbol{x}_j$$

Mixture of Experts and Self-Attention



We can express the output of the l^{th} head as follows:

$$\boldsymbol{h}_l = [\boldsymbol{h}_{l,1}, \ldots, \boldsymbol{h}_{l,N}]^{\mathsf{T}} \in \mathbb{R}^{N \times d_v}$$

$$\boldsymbol{h}_{l,i} = \sum_{j=1}^{N} \frac{\exp\left(s_{i,j}(\boldsymbol{X})\right)}{\sum_{k=1}^{N} \exp\left(s_{i,k}(\boldsymbol{X})\right)} f_j(\boldsymbol{X})$$

We can interpret each head in a multi-head self-attention layer as a multi-gate mixture of experts architecture.

Prefix Tuning via the Perspective of Mixture of Experts

- Prefix tuning can be interpreted as the introduction of new experts to customize the pre-trained model for a specific task

$$\boldsymbol{p}^K = \left[\boldsymbol{p}_1^K, \dots, \boldsymbol{p}_L^K\right]^\top \in \mathbb{R}^{L\times d}, \boldsymbol{p}^V = \left[\boldsymbol{p}_1^V, \dots, \boldsymbol{p}_L^V\right]^\top \in \mathbb{R}^{L\times d}$$

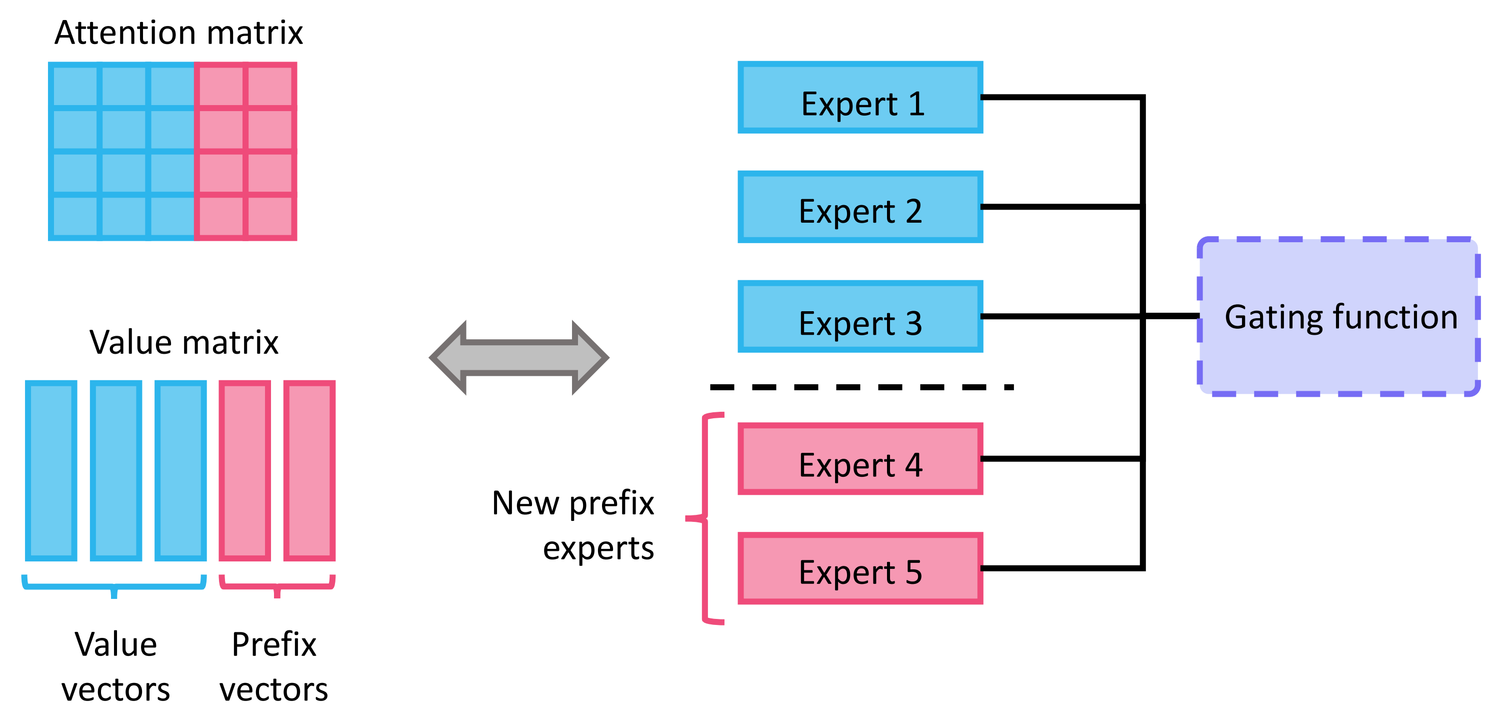
- Define new prefix experts along with their corresponding new score functions:

$$f_{N+j}(\boldsymbol{x}) \coloneqq {W_l^V}^\top \boldsymbol{p}_j^V,$$

$$s_{i,N+j}(x) \coloneqq \frac{\boldsymbol{x}^\top E_i^\top W_l^Q {W_l^K}^\top \boldsymbol{p}_j^K}{\sqrt{d_v}} = \frac{\boldsymbol{x}_i^\top W_l^Q {W_l^K}^\top \boldsymbol{p}_j^K}{\sqrt{dv}}$$

for $i \in [N]$ and $j \in [L]$.

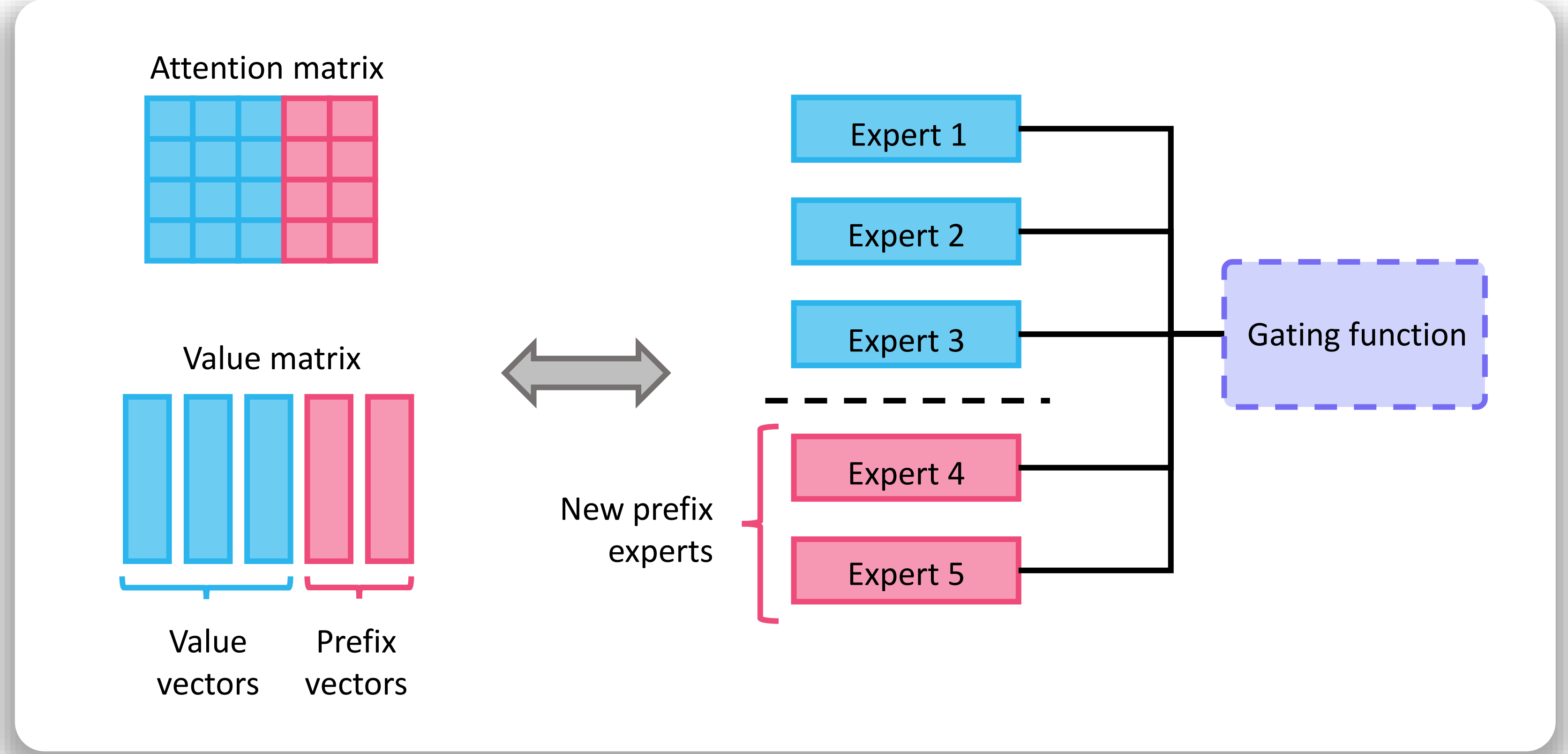
Linear gating prefix MoE model



$$\tilde{\boldsymbol{h}}_l = \mathrm{Attention}\left(\boldsymbol{X}^Q W_l^Q, \begin{bmatrix} \boldsymbol{p}^K \\ \boldsymbol{X}^K \end{bmatrix} W_l^K, \begin{bmatrix} \boldsymbol{p}^V \\ \boldsymbol{X}^V \end{bmatrix} W_l^V \right) = \left[\tilde{\boldsymbol{h}}_{l,1}, \dots, \tilde{\boldsymbol{h}}_{l,N}\right]^\top \in \mathbb{R}^{N \times d_v},$$

$$\begin{aligned}
\tilde{\boldsymbol{h}}_{l,i} = & \sum_{j=1}^{N} \frac{\exp(s_{i,j}(\boldsymbol{X}))}{\sum_{k=1}^{N} \exp(s_{i,k}(\boldsymbol{X})) + \sum_{k'=1}^{L} \exp(s_{i,N+k'}(\boldsymbol{X}))} f_j(\boldsymbol{X}) \quad \Big\} \text{Pretrained experts} \\
& + \sum_{j'=1}^{L} \frac{\exp(s_{i,N+j'}(\boldsymbol{X}))}{\sum_{k=1}^{N} \exp(s_{i,k}(\boldsymbol{X})) + \sum_{k'=1}^{L} \exp(s_{i,N+k'}(\boldsymbol{X}))} f_{N+j'}(\boldsymbol{X}) \quad \Big\} \text{New experts}
\end{aligned}$$

Linear gating prefix MoE model



Parameter estimation rate is $O(1/\log(n)^{\tau})$.

Requires HUGE amount of data!

$$\tilde{\boldsymbol{h}}_l = \text{Attention}\left(\boldsymbol{X}^Q W_l^Q, \begin{bmatrix} \boldsymbol{p}^K \\ \boldsymbol{X}^K \end{bmatrix} W_l^K, \begin{bmatrix} \boldsymbol{p}^V \\ \boldsymbol{X}^V \end{bmatrix} W_l^V\right) = \left[\tilde{\boldsymbol{h}}_{l,1}, \dots, \tilde{\boldsymbol{h}}_{l,N}\right]^\top \in \mathbb{R}^{N \times d_v},$$

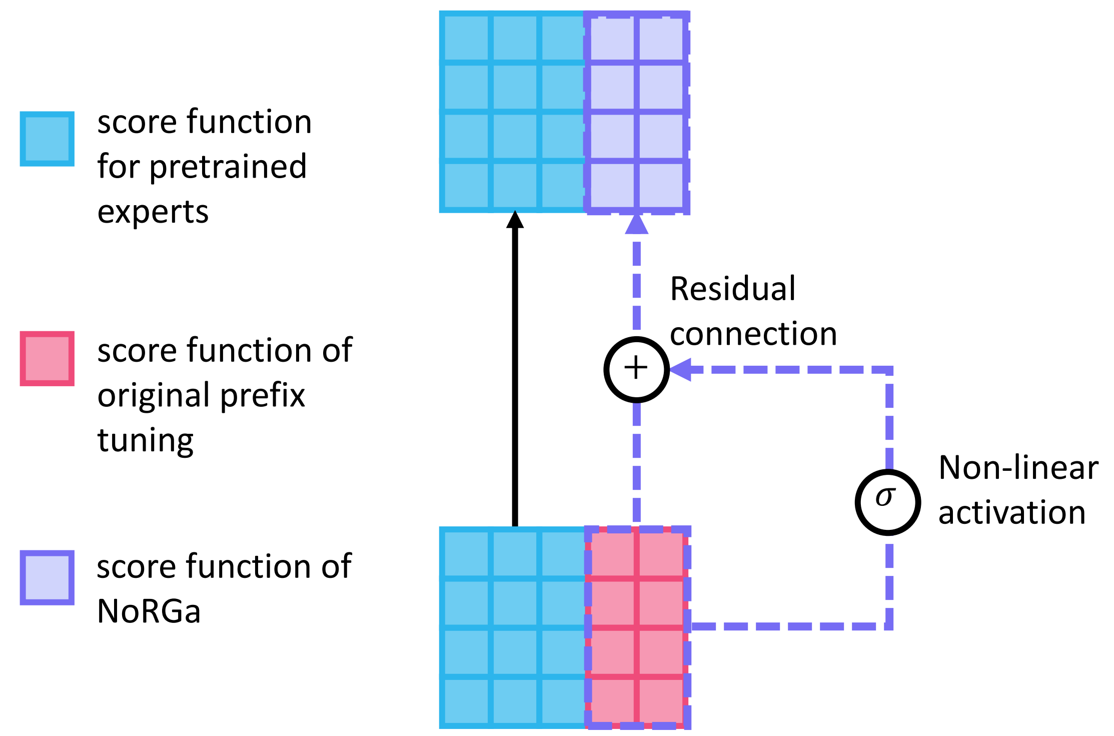
$$\begin{aligned} \tilde{\boldsymbol{h}}_{l,i} = & \left.\sum_{j=1}^{N} \frac{\exp(s_{i,j}(\boldsymbol{X}))}{\sum_{k=1}^{N} \exp(s_{i,k}(\boldsymbol{X})) + \sum_{k'=1}^{L} \exp(s_{i,N+k'}(\boldsymbol{X}))} f_j(\boldsymbol{X}) \right\} \text{Pretrained experts} \\ & \left. + \sum_{j'=1}^{L} \frac{\exp(s_{i,N+j'}(\boldsymbol{X}))}{\sum_{k=1}^{N} \exp(s_{i,k}(\boldsymbol{X})) + \sum_{k'=1}^{L} \exp(s_{i,N+k'}(\boldsymbol{X}))} f_{N+j'}(\boldsymbol{X}) \right\} \text{New experts} \end{aligned}$$

NoRGa – Non-linear Residual Gate

- Modify the linear gating prefix MoE model:

$$\hat{s}_{i,N+j}(\boldsymbol{X}) = s_{i,N+j}(\boldsymbol{X}) + \alpha \cdot \sigma\left(\tau \cdot s_{i,N+j}(\boldsymbol{X})\right), \quad i \in [N],\ j \in [L]$$

where α, τ are scalar factors, σ is a non-linear activation function

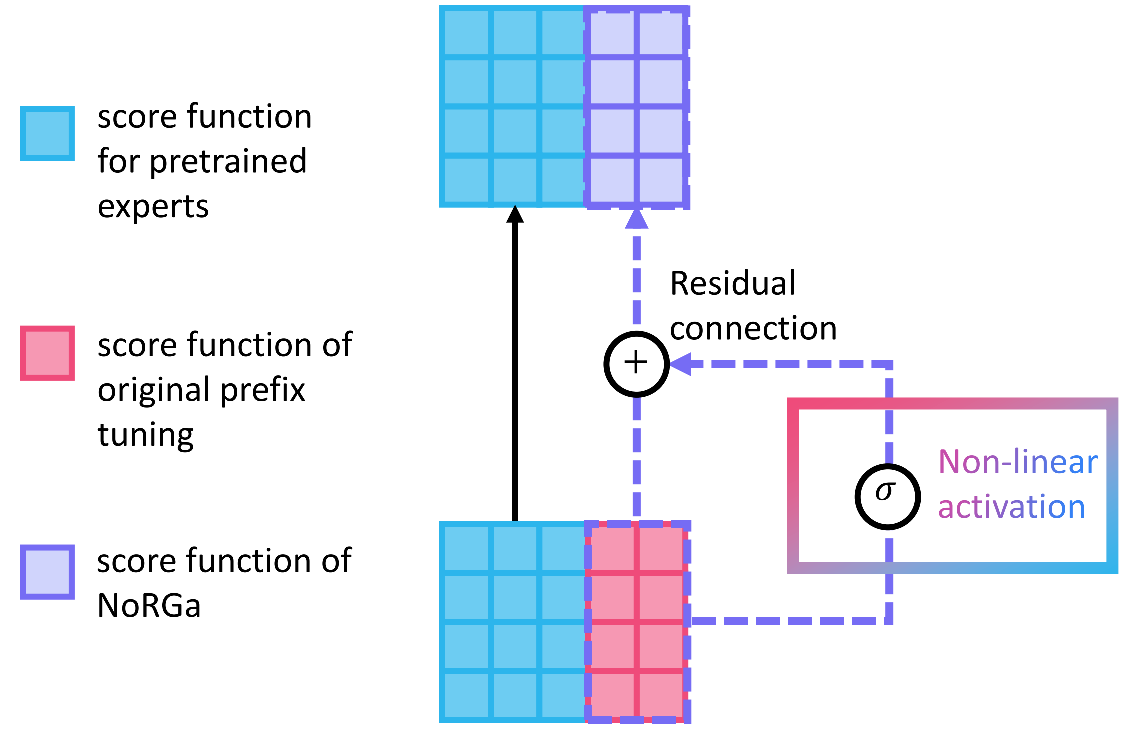


NoRGa – Non-linear Residual Gate

- Modify the linear gating prefix MoE model:

$$\hat{s}_{i,N+j}(\boldsymbol{X}) = s_{i,N+j}(\boldsymbol{X}) + \alpha \cdot \sigma\left(\tau \cdot s_{i,N+j}(\boldsymbol{X})\right),$$
$$i \in [N],\ j \in [L]$$

where α, τ are scalar factors, σ is a non-linear activation function

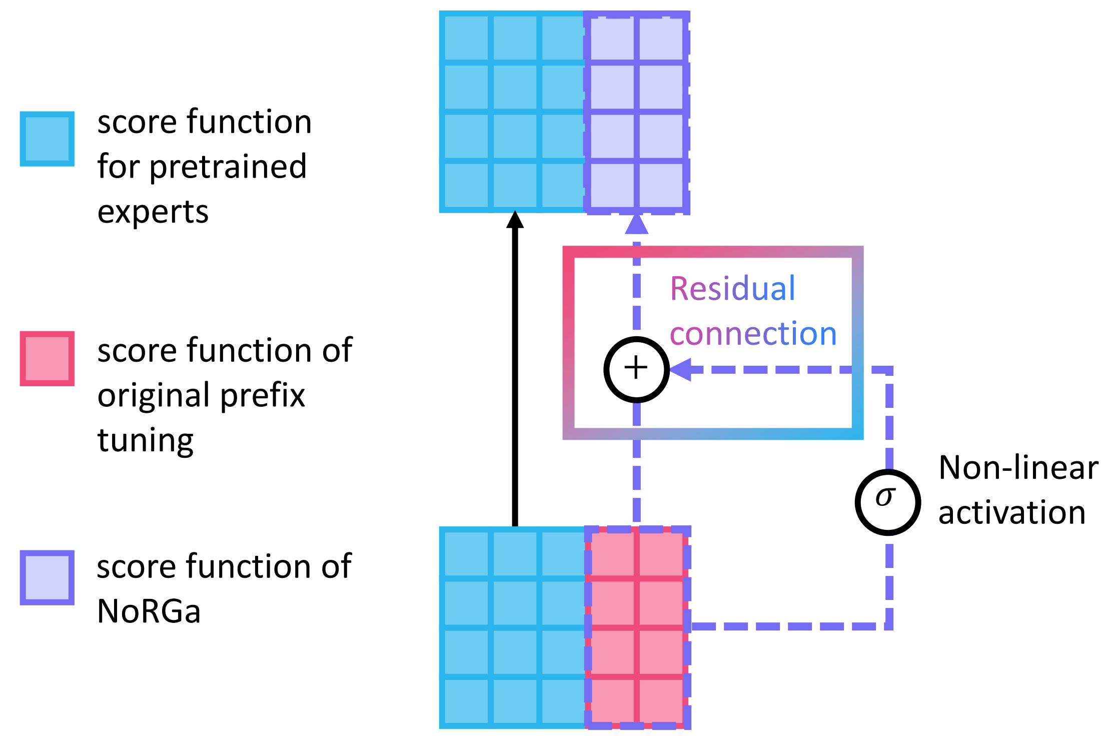


NoRGa – Non-linear Residual Gate

- Modify the linear gating prefix MoE model:

$$\hat{s}_{i,N+j}(\boldsymbol{X}) = s_{i,N+j}(\boldsymbol{X}) + \alpha \cdot \sigma\left(\tau \cdot s_{i,N+j}(\boldsymbol{X})\right), \quad i \in [N],\ j \in [L]$$

where α, τ are scalar factors, σ is a non-linear activation function



NoRGa – Non-linear Residual Gate

- We prove that estimating parameters in the non-linear residual gating prefix MoE model is statistically efficient in terms of the number of data.

Model	Parameter estimation rate	Number of data
Linear gating prefix MoE	$O(1/\log(n)^{\tau})$	Exponential $\exp(\varepsilon^{-\tau})$
Non-linear residual gating prefix MoE	$O\left(\sqrt[4]{\log(n)/n}\right)$	Polynomial ε^{-4}

Experiments

PTM	Method	Split CIFAR-100			Split ImageNet-R		
		FA (↑)	CA (↑)	FM (↓)	FA (↑)	CA (↑)	FM (↓)
Sup-21K	L2P	83.06 ± 0.17	88.27 ± 0.71	5.61 ± 0.32	67.53 ± 0.44	71.98 ± 0.52	5.84 ± 0.38
	DualPrompt	87.30 ± 0.27	91.23 ± 0.65	3.87 ± 0.43	70.93 ± 0.08	75.67 ± 0.52	5.47 ± 0.19
	S-Prompt	87.57 ± 0.42	91.38 ± 0.69	3.63 ± 0.41	69.88 ± 0.51	74.25 ± 0.55	4.73 ± 0.47
	CODA-Prompt	86.94 ± 0.63	91.57 ± 0.75	4.04 ± 0.18	70.03 ± 0.47	74.26 ± 0.24	5.17 ± 0.22
	HiDe-Prompt	92.61 ± 0.28	94.03 ± 0.01	1.50 ± 0.28	75.06 ± 0.12	76.60 ± 0.01	**4.09 ± 0.13**
	NoRGa (Ours)	**94.48 ± 0.13**	**95.83 ± 0.37**	**1.44 ± 0.27**	**75.40 ± 0.39**	**79.52 ± 0.07**	4.59 ± 0.07
iBOT-21K	L2P	79.13 ± 1.25	85.13 ± 0.05	7.50 ± 1.21	61.31 ± 0.50	68.81 ± 0.52	10.72 ± 0.40
	DualPrompt	78.84 ± 0.47	86.16 ± 0.02	8.84 ± 0.67	58.69 ± 0.61	66.61 ± 0.67	11.75 ± 0.92
	S-Prompt	79.14 ± 0.65	85.85 ± 0.17	8.23 ± 1.15	57.96 ± 1.10	66.42 ± 0.71	11.27 ± 0.72
	CODA-Prompt	80.83 ± 0.27	87.02 ± 0.20	7.50 ± 0.25	61.22 ± 0.35	66.76 ± 0.37	9.66 ± 0.20
	HiDe-Prompt	93.02 ± 0.15	94.56 ± 0.05	**1.26 ± 0.13**	70.83 ± 0.17	73.23 ± 0.08	**6.77 ± 0.23**
	NoRGa (Ours)	**94.76 ± 0.15**	**95.86 ± 0.31**	1.34 ± 0.14	**73.06 ± 0.26**	**77.46 ± 0.42**	6.88 ± 0.49
iBOT-1K	L2P	75.51 ± 0.88	82.53 ± 1.10	6.80 ± 1.70	59.43 ± 0.28	66.83 ± 0.92	11.33 ± 1.25
	DualPrompt	76.21 ± 1.00	83.54 ± 1.23	9.89 ± 1.81	60.41 ± 0.76	66.87 ± 0.41	9.21 ± 0.43
	S-Prompt	76.60 ± 0.61	82.89 ± 0.89	8.60 ± 1.36	59.56 ± 0.60	66.60 ± 0.13	8.83 ± 0.81
	CODA-Prompt	79.11 ± 1.02	86.21 ± 0.49	7.69 ± 1.57	66.56 ± 0.68	73.14 ± 0.57	7.22 ± 0.38
	HiDe-Prompt	93.48 ± 0.11	95.02 ± 0.01	1.63 ± 0.10	71.33 ± 0.21	73.62 ± 0.13	7.11 ± 0.02
	NoRGa (Ours)	**94.01 ± 0.04**	**95.11 ± 0.35**	**1.61 ± 0.30**	**72.77 ± 0.20**	**76.55 ± 0.46**	**7.10 ± 0.39**

Experiments

Method	Split CIFAR-100		Split CUB-200	
	Sup-21K	iBOT-21K	Sup-21K	iBOT-21K
HiDe-Prompt	92.61	93.02	86.56	78.23
NoRGa tanh	94.36	**94.76**	90.87	**80.69**
NoRGa sigmoid	**94.48**	94.69	**90.90**	80.18
NoRGa gelu	94.05	94.63	90.74	80.54

Conclusion

- Reveals a novel connection between **Prefix Tuning**, a popular prompt implementation technique, and **Mixture of Experts**.
- Proposes **Non-linear Residual Gates (NoRGa)**, an innovative gating mechanism.
- Achieves state-of-the-art performance across various **continual learning benchmarks** and pretraining settings.

NEURAL INFORMATION PROCESSING SYSTEMS

THANK YOU FOR YOUR ATTENTION!

Mixture of experts meets Prompt-based Continual Learning

Minh Le[3], An Nguyen[2]*, Huy Nguyen[1]*, Trang Nguyen[3]*,

Trang Pham[3]*, Linh Van Ngo[2], Nhat Ho[1]

[1] University of Texas at Austin
[2] Hanoi University of Science and Technology
[3] VinAI Research