Revisiting Single-gated Mixture of Experts

Qualcomm AI research

BMVC 2022

Amélie Royer
Ilia Karmanov
Andrii Skliar
Babak Ehteshami Bejnordi
Tijmen Blankevoort

Qualcomm Technologies Netherlands, B.V.

{aroyer, ikarmano, askliar,behtesha,tijmen}@qti.qualcomm.com

1. Motivation

Mixture of Experts (**MoE**) are rising in popularity as a means to train extremely large models yet allowing for a reasonable computational cost at inference time. However, state-of-the-art approaches either:

- *(large-scale MoE)* Utilize many experts and routing decisions that have to be trained jointly, which leads to **training instabilities** and can make it hard to implement the routing in practice
- *(hierarchical classifiers)* Define rigid per-class routing that might **not be optimal subsets** of the data to train on

We propose to revisit the single-gate MoE and improve its accuracy-efficiency trade-off, as well as training practicality. Key to our work are:

- A full base model branch acting both as an early-exit (**efficiency**) and an ensembling regularization scheme (**accuracy**)
- A simple and efficient **asynchronous training pipeline** without router collapse issues
- An automatic per-sample clustering-based initialization.

2. Our Improved Single-gated MoE Design

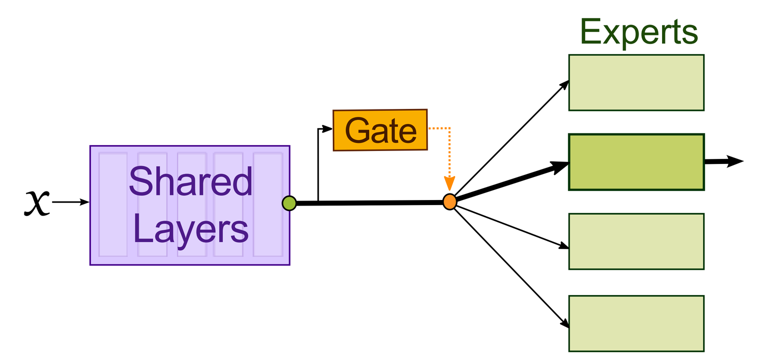

Figure 1: Single-gated MoE contain a shared branch of *few* layers (the **base model** ϕ) and a set of K (here, K = 4) separate **experts**. Each sample is routed to a *unique* expert during execution to produce the final model predictions. The routing is decided by a small lightweight **gate** module, which takes as inputs the base model output features.

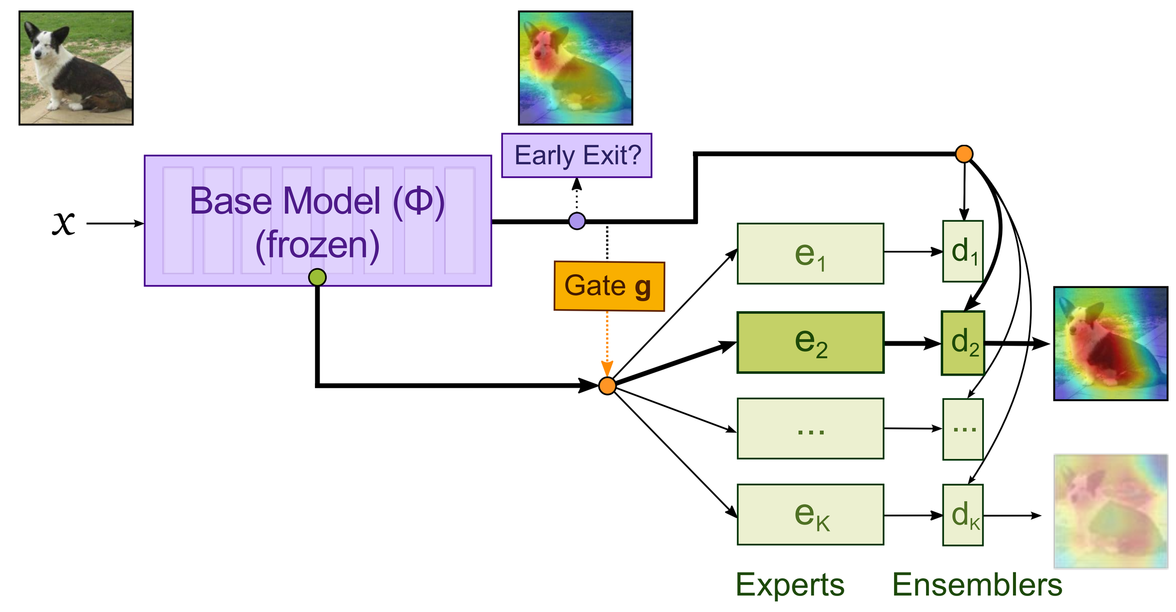

Figure 2: We propose improvementsto the traditional single-gated MoE:

- **Accuracy improvement:** We use a full network as our base model, and use its logit outputs to regularize the experts via *ensembling*
- **Efficiency improvement:** By design, we can use the base model's output as *early exit* at inference time, avoiding the computational cost of the experts
- **Training:** We propose an efficient asynchronous and stable training scheme: The gate is initialized by clustering the base models features, then frozen. Experts can thus be trained separately, and the gate does not risk mode collapse

3. Training Scheme

The components of our model are:

The base model ϕ is network trained on the whole dataset, and is executed for every input. It captures **shared generic knowledge**.

Experts e_k take as input an intermediate feature map of the base model. At inference, the most probable expert is executed. They capture **specialized knowledge**.

Ensemblers d_k combines outputs of the base model and selected expert. We experiment with several ensembling designs and use bagging in practice: d_k **(x) =** ϕ**(x) +** e_k **(x)**

Asynchronous Training algorithm

Step 1: Train the base model φ (or use off-the-shelf) then **freeze**

Step 2 (init routing): Cluster the base model embeddings using **K-Means**, obtaining cluster centers $c_{1...K}$

Define target gate g* to route samples to the closest centroid

Step 3 (train): Train the gate g by minimizing **KL(g, g*)** then **freeze**

For **k** = 1 to K (asynchronous) do

- **Initialize** k-th expert from the base model's weights
- **Sample** training example set D_k by following the distribution given by g + ϵ , where ϵ is **regularization** noise
- Train the k-th **expert** on D_k

4. Any-time Performance for Mazimized Efficiency

Default Behavior (static): Select the top-1 expert chosen by the gate

Early-exiting (dynamic): Exit after the base model forward pass

Top-k experts (dynamic): Select more than one expert and combines their output via ensembling

We find that we can implement both dynamic behavior with a simple **thresholding rule** and achieve good performance. More complex (e.g., learned) early-exiting strategies did not help.

$$\alpha_k = g(k\,|x)\,(1\, - \max_y \phi(y\,|x))$$

Combined gate and base model confidence

$$ee(x) = 1\; iff\; \forall k, \alpha_k(x) < \tau$$

Early exit if no expert is confident enough

$$\text{out}^{anytime}(x) = \boldsymbol{ee(x)}\phi(x) + \left(1\, - \boldsymbol{ee(x)}\right) \sum_k \mathbf{1}_{\alpha_k \geq \tau}\, g(k\,|x)\, d_k(e_k(y\,|x))$$

5. Results on Image Classification

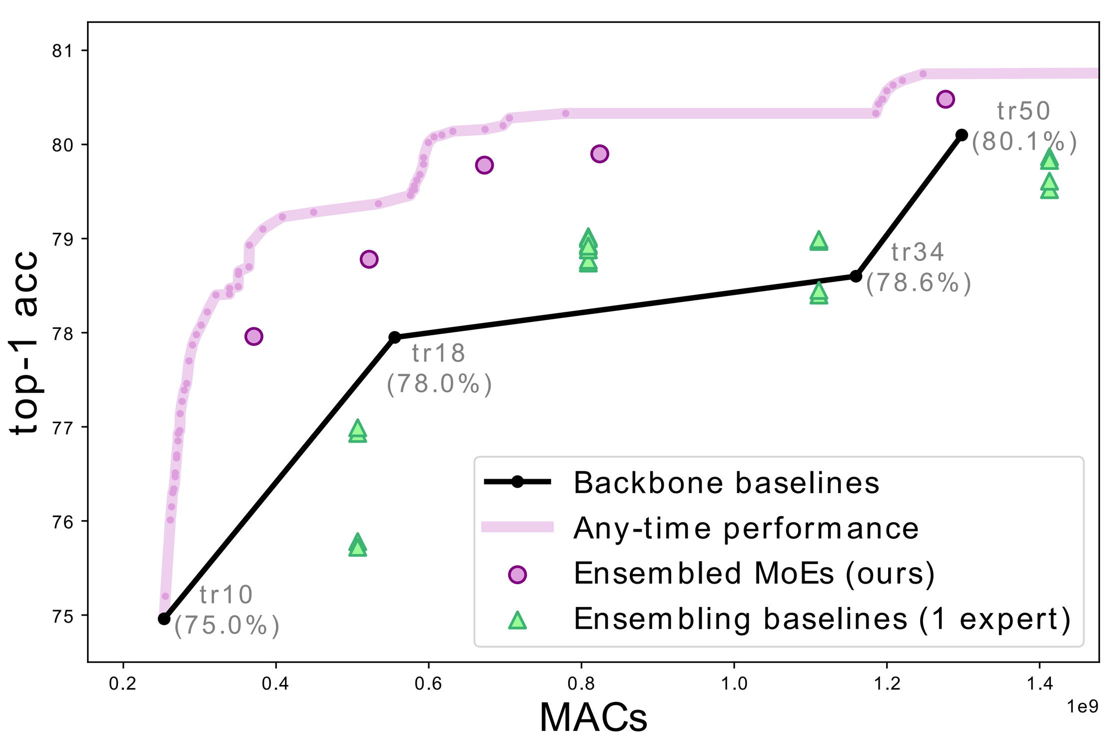

Figure 3: MACs (efficiency) vs Accuracy results on **CIFAR100** on ResNet18 (○), compared against different widths of ResNets (•) and a one expert baseline (△)

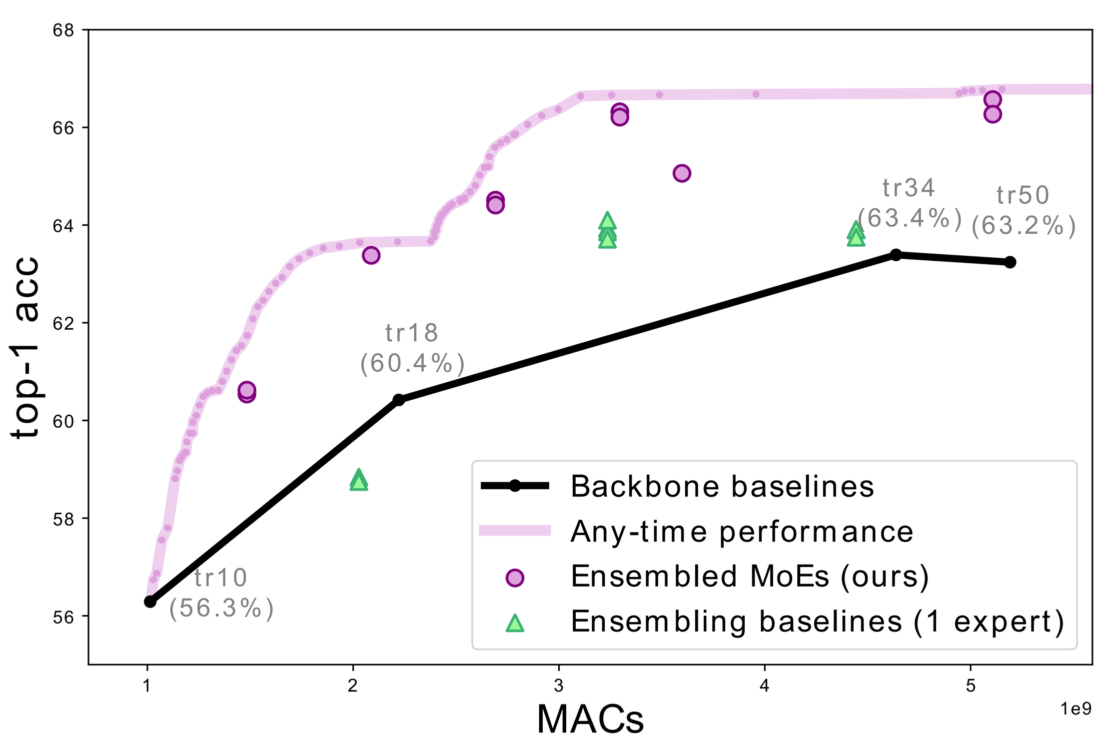

Figure 4: MACs versus Accuracy results on **tiny-ImageNet** on ResNet18

ResNet18	No early exit	τ = 0.75	τ = 0.5
1-expert baseline	71.50	71.50	71.13
4 experts	72.17	72.11	71.68
20 experts	**72.38**	**72.38**	**71.73**
MACs x1e9	2.64	2.18	2.03

Table 1: **ImageNet** results with ResNet18 base model (69.76% accuracy, 1.82 GMACs). Experts are implemented as 2 residual blocks + 1 linear layer

MobileNet	No early exit	τ = 0.75	τ = 0.5
1-expert baseline	68.06	68.13	68.15
4 experts	**68.60**	**68.59**	68.44
20 experts	68.58	68.53	**68.46**
MACs x1e7	8.13	6.83	6.36

Table 2: **ImageNet** results with MobileNetv3 base model (67.67% accuracy, 5.65e7 MACs). Experts are implemented as 4 inverted residual blocks + 1 linear layer

Comparison to multi-gated MoEs	# gates	Acc	GMACs	# train params
Ours	1	72.17	2.64	5.1e9
τ = 0.75	1	72.11	2.18	5.1e9
DeepMoE [1]	17	70.95	1.81	7.0e9

Table 3: Comparison to **DeepMoE [1]** baseline: [1] trains a twice wide ResNet alongside a gate in each layer that selects half of the channels as inactive: *The inference cost is that of ResNet-18, but the training cost is of a twice as wide network*

6. Per-sample Assignment

The per-sample routing uncovers meaningful intra-class variations. This shows **the limits of per-class routing** (e.g. Hierarchical classification) as it can sometimes be too rigid to capture data diversity

The class king-penguin (*left*) co-occurs with other animals (*right*) for full-view images.

but is grouped with e.g., bell pepper when the image is a close-up of its orange beak

Comparing per-sample vs per-class routing	Per-sample (Ours)	Per-class	Per-class + Oracle
With ensemblers	65.7	63.9	68.0
Without ensemblers	63.1	62.5	68.8

Conclusions

- We augment MoE with a novel **ensembling scheme** and a simple **asynchronous** and stable training pipeline leveraging a per-sample clustering-based initialization.
- Our model consistently reaches higher accuracy than hierarchical classifiers and a 1-expert ensembling baseline, revealing the benefits of training specialized experts with **per-sample routing**.
- Finally, maintaining the base model as an independent branch allows us to further save computations at inference time using a **simple threshold-based conditional** rule to adapt the computational budget without retraining.

[1] Deep Mixture of Experts via Shallow Embeddings, published in UAI 2019