Google AI
Brain Team

Sparse MoEs meet efficient ensembles

Rodolphe Jenatton (rjenatton@google.com)

Amazon StatML Workshop, 3/28/2023

Google AI

Credits

James Urquhart Allingham[1,*] *jua23@cam.ac.uk*

Florian Wenzel[2,†] *fln.wenzel@gmail.com*

Zelda E Mariet, Basil Mustafa *{zmariet,basilm}@google.com*

Joan Puigcerver, Neil Houlsby *{jpuigcerver,neilhoulsby}@google.com*

Ghassen Jerfel[3,†] *ghassen@google.com*

Vincent Fortuin[1,4,*] *vbf21@cam.ac.uk*

Balaji Lakshminarayanan, Jasper Snoek *{balajiln,jsnoek}@google.com*

Dustin Tran, Carlos Riquelme, Rodolphe Jenatton *{trandustin,rikel,rjenatton}@google.com*

Google Research, Brain Team; [1] University of Cambridge; [2] no affiliation; [3] Waymo; [3] ETH Zürich

Credits

James Urquhart Allingham[1,*] *jua23@cam.ac.uk*

Florian Wenzel[2,†] *fln.wenzel@gmail.com*

Zelda E Mariet, Basil Mustafa *{zmariet,basilm}@google.com*

Joan Puigcerver, Neil Houlsby *{jpuigcerver,neilhoulsby}@google.com*

Ghassen Jerfel[3,†] *ghassen@google.com*

Vincent Fortuin[1,4,*] *vbf21@cam.ac.uk*

Balaji Lakshminarayanan, Jasper Snoek *{balajiln,jsnoek}@google.com*

Dustin Tran, Carlos Riquelme, Rodolphe Jenatton *{trandustin,rikel,rjenatton}@google.com*

Google Research, Brain Team; [1] University of Cambridge; [2] no affiliation; [3] Waymo; [3] ETH Zürich

Some parts of the talk reuse slides from:

- {balajiln, trandustin, jsnoek}@: NeurIPS tutorial 2020, Uncertainty and Out-of-Distribution Robustness in Deep Learning

Context and motivations

Google AI
Brain Team

Project at the intersection of two topics:

- **Sparse mixture of experts (sparse MoEs)**
- **Reliability in deep learning**
  - **Why and how to measure it?**
  - **Ensembles**

Context and motivations

Google AI Brain Team

Project at the intersection of two topics:

- **Sparse mixture of experts (sparse MoEs)**
- **Reliability in deep learning**
  - **Why and how to measure it?**
  - **Ensembles**

Sparse mixture of experts (sparse MoEs)

Google AI
Brain Team

In a nutshell:

- **Conditional computation** [Bengio et al., 2013]:
  - Only subpart of the model activated in an input-dependent fashion
  - (≠ standard “dense” models: all parameters used to process an input)

Sparse mixture of experts (sparse MoEs)

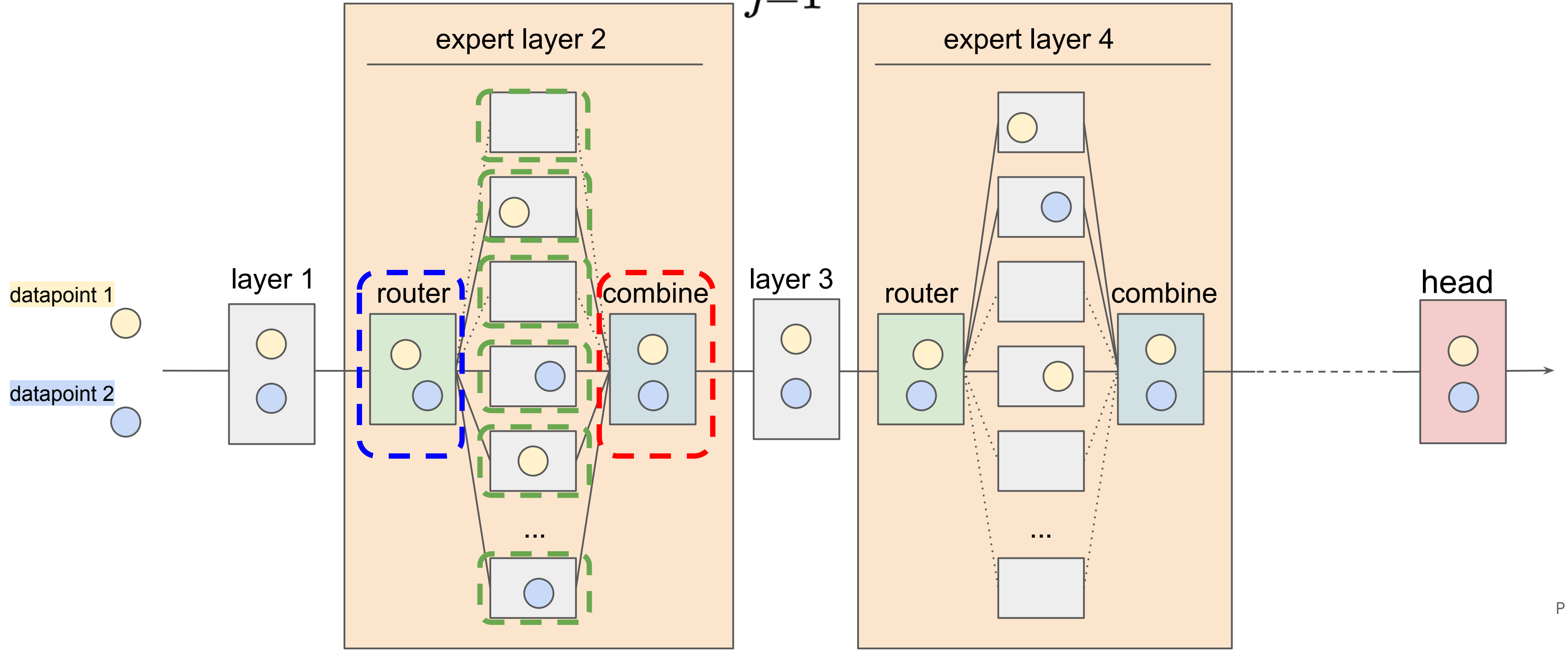
In a nutshell:

- **Conditional computation** [Bengio et al., 2013]:
    - Only subpart of the model activated in an input-dependent fashion
    - (≠ standard "dense" models: all parameters used to process an input)
- **Goal**: Grow model size while keeping compute ~ constant

Pictorial view of sparse MoEs

Google AI
Brain Team

$$f(\mathbf{x}) = \sum_{j=1}^{E} g_j(\mathbf{x}) \cdot \mathtt{expert}_j(\mathbf{x})$$



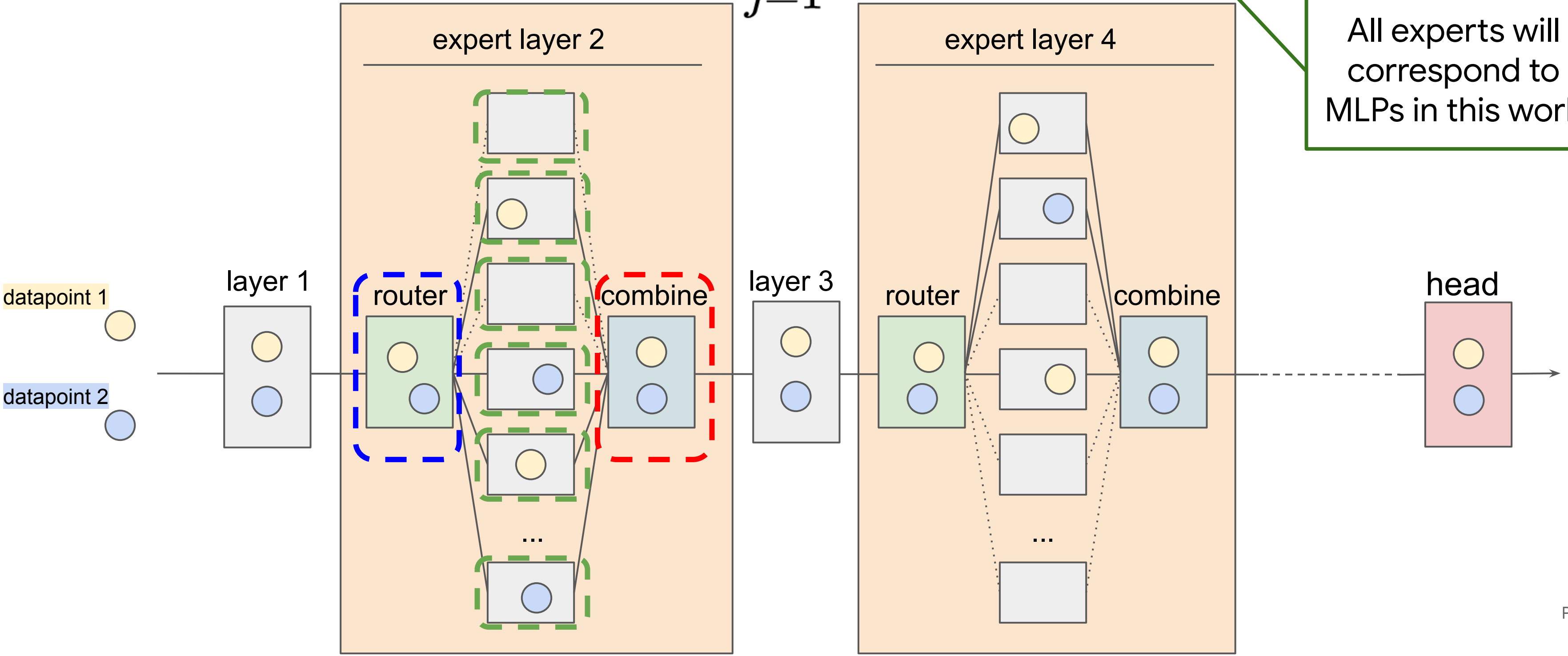
Pictorial view of sparse MoEs

#experts

Sparse gating weights (only **K** non-zero)

$$f(\mathbf{x}) = \sum_{j=1}^{E} g_j(\mathbf{x}) \cdot \mathtt{expert}_j(\mathbf{x})$$

All experts will correspond to MLPs in this work



Sparse MoEs successfully applied in NLP

Google AI
Brain Team

OUTRAGEOUSLY LARGE NEURAL NETWORKS:
THE SPARSELY-GATED MIXTURE-OF-EXPERTS LAYER

Noam Shazeer[1], Azalia Mirhoseini*†[1], Krzysztof Maziarz*[2], Andy Davis[1], Quoc Le[1], Geoffrey Hinton[1] and Jeff Dean[1]

[1]Google Brain, {noam,azalia,andydavis,qvl,geoffhinton,jeff}@google.com
[2]Jagiellonian University, Cracow, krzysztof.maziarz@student.uj.edu.pl

GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding

Dmitry Lepikhin lepikhin@google.com
HyoukJoong Lee hyouklee@google.com
Yuanzhong Xu yuanzx@google.com
Dehao Chen dehao@google.com
Orhan Firat orhanf@google.com
Yanping Huang huangyp@google.com
Maxim Krikun krikun@google.com
Noam Shazeer noam@google.com
Zhifeng Chen zhifengc@google.com

SWITCH TRANSFORMERS: SCALING TO TRILLION PARAMETER MODELS WITH SIMPLE AND EFFICIENT SPARSITY

William Fedus* Google Brain liamfedus@google.com
Barret Zoph* Google Brain barretzoph@google.com
Noam Shazeer Google Brain noam@google.com

PATHWAYS: ASYNCHRONOUS DISTRIBUTED DATAFLOW FOR ML

Paul Barham[1] Aakanksha Chowdhery[1] Jeff Dean[1] Sanjay Ghemawat[1] Steven Hand[1] Dan Hurt[1]
Michael Isard[1] Hyeontaek Lim[1] Ruoming Pang[1] Sudip Roy[1] Brennan Saeta[1] Parker Schuh[1]
Ryan Sepassi[1] Laurent El Shafey[1] Chandramohan A. Thekkath[1] Yonghui Wu[1]

ABSTRACT

We present the design of a new large scale orchestration layer for accelerators. Our system, PATHWAYS, is explicitly designed to enable exploration of new systems and ML research ideas, while retaining state of the art performance for current models. PATHWAYS uses a *sharded* dataflow graph of *asynchronous* operators that consume and produce futures, and efficiently gang-schedules *heterogeneous* parallel computations on thousands of accelerators while coordinating data transfers over their dedicated interconnects. PATHWAYS makes use of a novel

Carbon Emissions and Large Neural Network Training

David Patterson[1,2], Joseph Gonzalez[2], Quoc Le[1], Chen Liang[1], Lluis-Miquel Munguia[1], Daniel Rothchild[2], David So[1], Maud Texier[1], and Jeff Dean[1]
{davidpatterson, qvl, crazydonkey, llmunguia, davidso, maudt, jeff}@google.com, {pattrsn, jegonzal, drothchild}@berkeley.edu

Abstract: The computation demand for machine learning (ML) has grown rapidly recently, which comes with a number of costs. Estimating the energy cost helps measure its environmental impact and finding greener strategies, yet it is challenging without detailed information.

We calculate the energy use and carbon footprint of several recent large models—T5, Meena, GShard, Switch Transformer, and GPT-3—and refine earlier estimates for the neural architecture search that found Evolved Transformer.

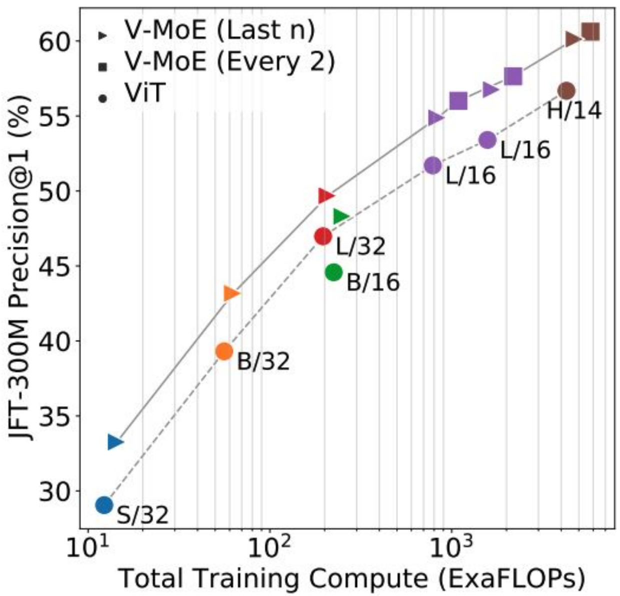
We highlight the following opportunities to improve energy efficiency and CO_2 *equivalent emissions* (CO_2e):

- Large but sparsely activated DNNs can consume <1/10th the energy of large, dense DNNs without sacrificing accuracy despite using as many or even more parameters.

Sparse MoEs *for vision*

[Riquelme, Puigcerver, Mustafa et al., 2021]:

- Inspired by successful applications of Sparse MoEs in NLP
- Strong performance vs. FLOPs trade-off



dense to sparse MoEs

Image credit: Riquelme et al., 2021

Context and motivations

Google AI
Brain Team

Project at the intersection of two topics:

- **Sparse mixture of experts (sparse MoEs)**
- **Reliability in deep learning**
  - **Why and how to measure it?**
  - **Ensembles**

Uncertainty & reliability in deep learning, why? (1/2)

Google AI
Brain Team

ML systems are being deployed in many *safety-critical* applications, e.g.,

- **Health applications** [Miotto et al., 2016; Rajkomar et al., 2018; Liu et al., 2020; Mckinney et al., 2020;...]
- **Self-driving cars** [Levinson et al., 2011; Sun et al., 2018]
- **Benefit claims & welfare issues** (e.g., the Guardian, 2019)
- **Dialog systems with LLMs**
- ...

Uncertainty & reliability in deep learning, why? (2/2)

Google AI
Brain Team

In those applications, we need robust uncertainty estimation

- Knowing when to trust model's predictions, e.g., under dataset shift
- Better decision making, e.g., with asymmetric costs

Uncertainty & reliability in deep learning, why? (2/2)

Google AI
Brain Team

In those applications, we need robust uncertainty estimation

- Knowing when to trust model's predictions, e.g., under dataset shift
- Better decision making, e.g., with asymmetric costs
- Open set recognition
- Lifelong learning
- Active learning, RL, Bayesian optimization

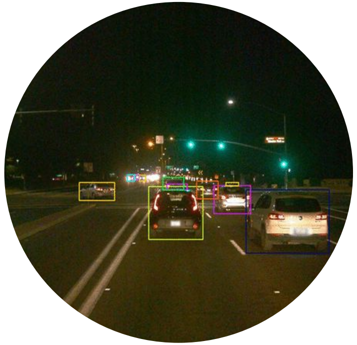
Example: Self-driving cars

Google AI
Brain Team

Dataset shift:

- Time of day / Lighting
- Geographical location (City vs suburban)
- Changing conditions (Weather / Construction)

Daylight



Night

Weather

Construction

Downtown

Suburban

Image credit: Sun et al, [Waymo Open Dataset]

Credits: slide from B. Lakshminarayanan, D. Tran, J. Snoek "Uncertainty and Out-of-Distribution Robustness in Deep Learning"

How can we measure reliability, robustness & uncertainty?

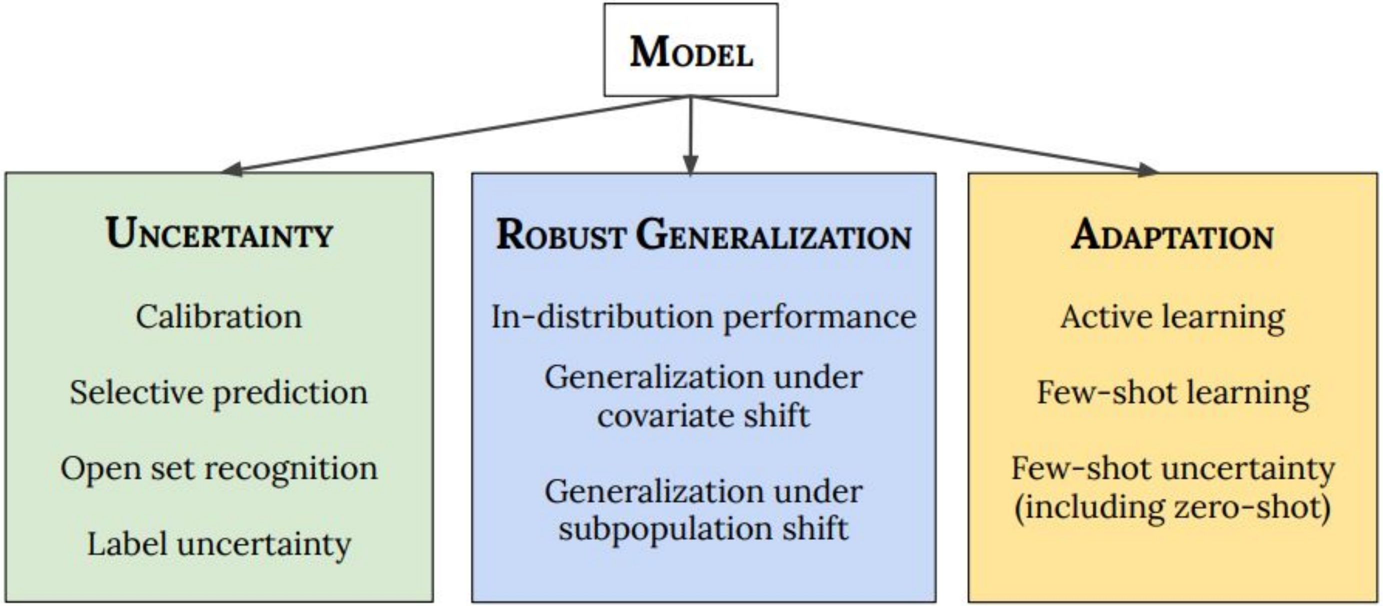


Image credit: Tran et al., 2022, Plex: Towards Reliability Using Pretrained Large Model Extensions

How can we measure reliability, robustness & uncertainty?

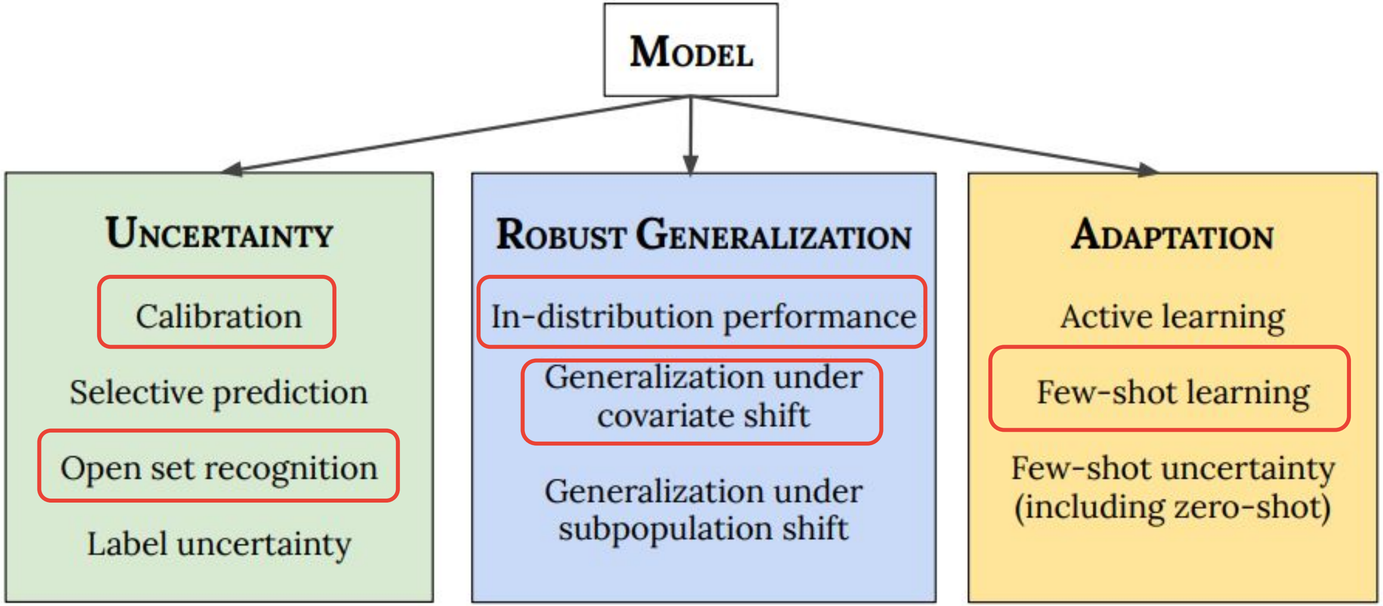


Image credit: [Tran et al., 2022, Plex: Towards Reliability Using Pretrained Large Model Extensions]

Calibration

Calibration Error = |Confidence - Accuracy|

predicted probability of correctness

observed frequency of correctness

Credits: slide from B. Lakshminarayanan, D. Tran, J. Snoek "Uncertainty and Out-of-Distribution Robustness in Deep Learning"

Calibration

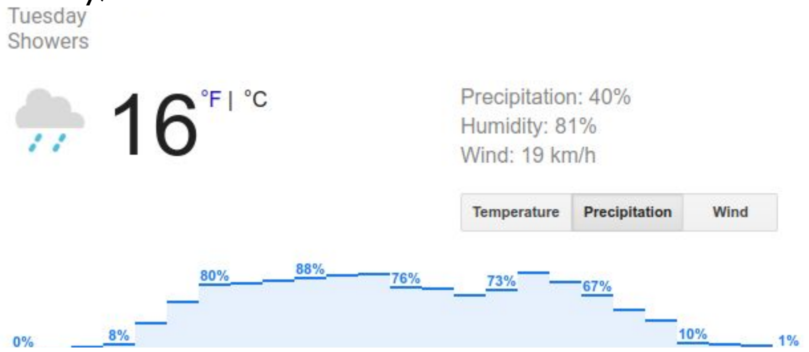
Google AI
Brain Team

Calibration Error = |Confidence - Accuracy|

predicted probability of correctness

observed frequency of correctness

Of all the days where the model predicted rain with 80% probability, what fraction did we observe rain?



- 80% implies perfect calibration
- Less than 80% implies model is *overconfident*
- Greater than 80% implies model is *under-confident*

Credits: slide from B. Lakshminarayanan, D. Tran, J. Snoek "Uncertainty and Out-of-Distribution Robustness in Deep Learning"

Calibration

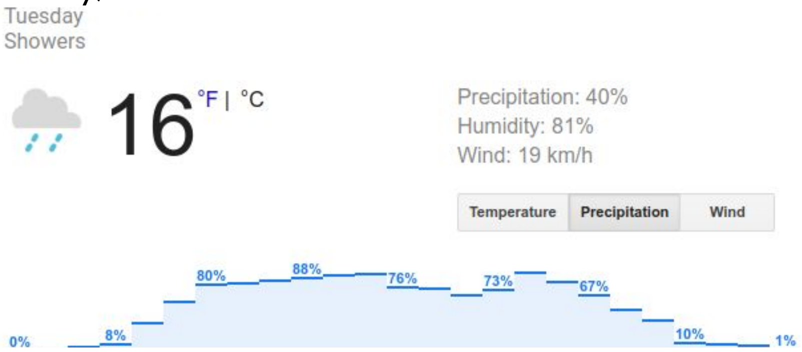
Google AI
Brain Team

Calibration Error = |Confidence - Accuracy|

predicted probability of correctness

observed frequency of correctness

Of all the days where the model predicted rain with 80% probability, what fraction did we observe rain?



- 80% implies perfect calibration
- Less than 80% implies model is *overconfident*
- Greater than 80% implies model is *under-confident*
- In practice, use some binning [Naeini et al., 2015]:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|$$

Credits: slide from B. Lakshminarayanan, D. Tran, J. Snoek "Uncertainty and Out-of-Distribution Robustness in Deep Learning"

Other ways to quantify reliability & uncertainty?

- **Proper scoring rules** [Gneiting & Raftery, 2007]: Log-likelihood, Brier score, ...

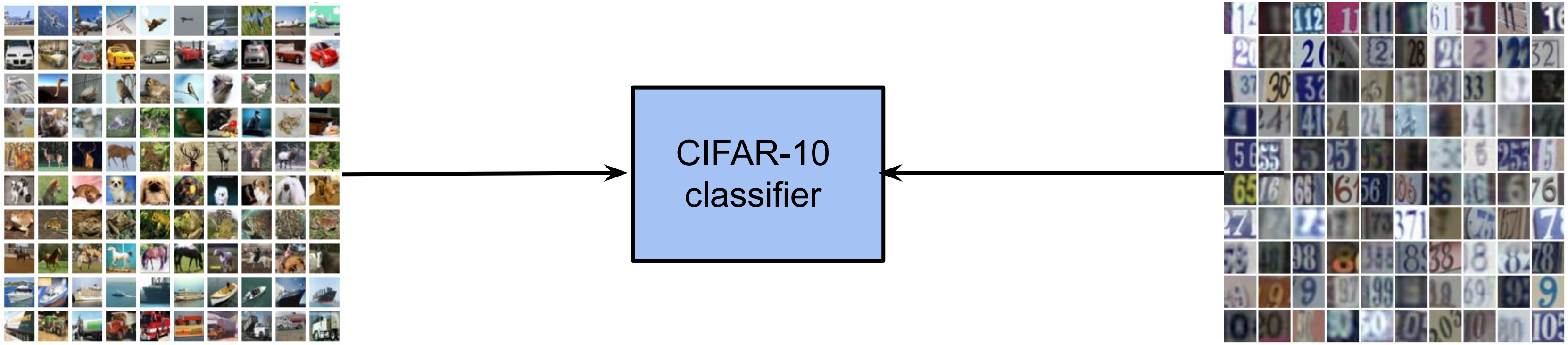
Other ways to quantify reliability & uncertainty?

Google AI
Brain Team

- **Proper scoring rules** [Gneiting & Raftery, 2007]: Log-likelihood, Brier score, ...
- **Out-of-distribution (OOD) detection:**

CIFAR-10 (test inputs)

SVHN (OOD inputs)

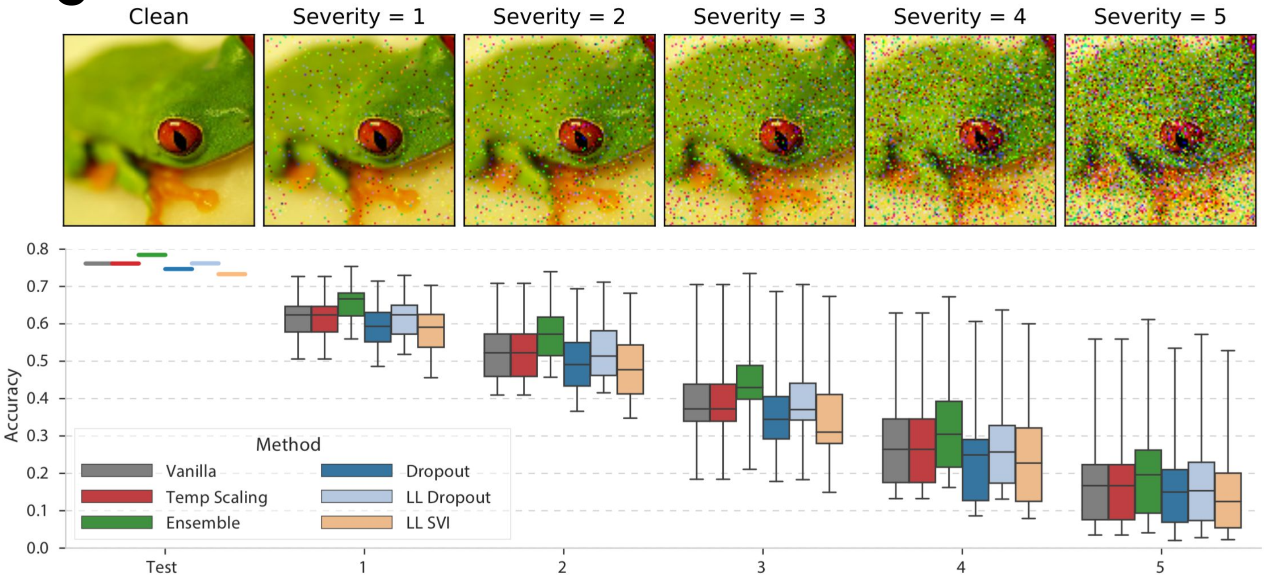


Confidence on test inputs > Confidence on OOD inputs ? (e.g., via AUC)

Credits: slide from B. Lakshminarayanan, D. Tran, J. Snoek "Uncertainty and Out-of-Distribution Robustness in Deep Learning"

Models accuracy degrades under dataset shift

Google AI Brain Team



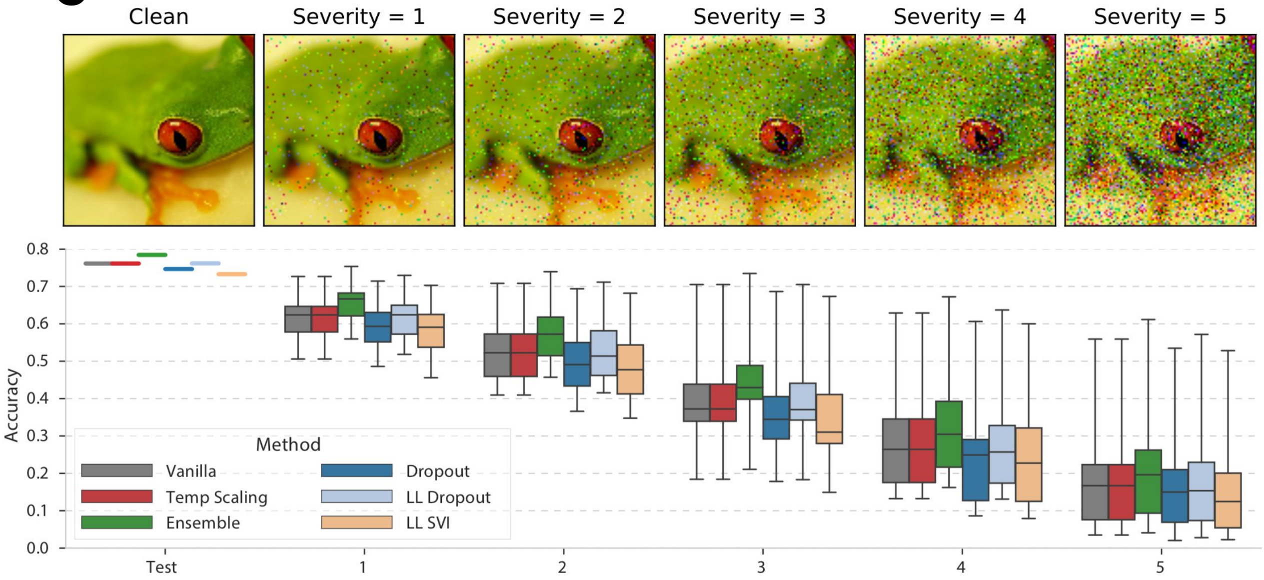
- **Accuracy drops** with increasing shift on Imagenet-C

Image source: [Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift?, Ovadia et al. 2019]

Credits: slide from B. Lakshminarayanan, D. Tran, J. Snoek "Uncertainty and Out-of-Distribution Robustness in Deep Learning"

Models accuracy degrades under dataset shift

Google AI
Brain Team



- **Accuracy drops** with increasing shift on Imagenet-C

- But do the models know that they are less accurate?

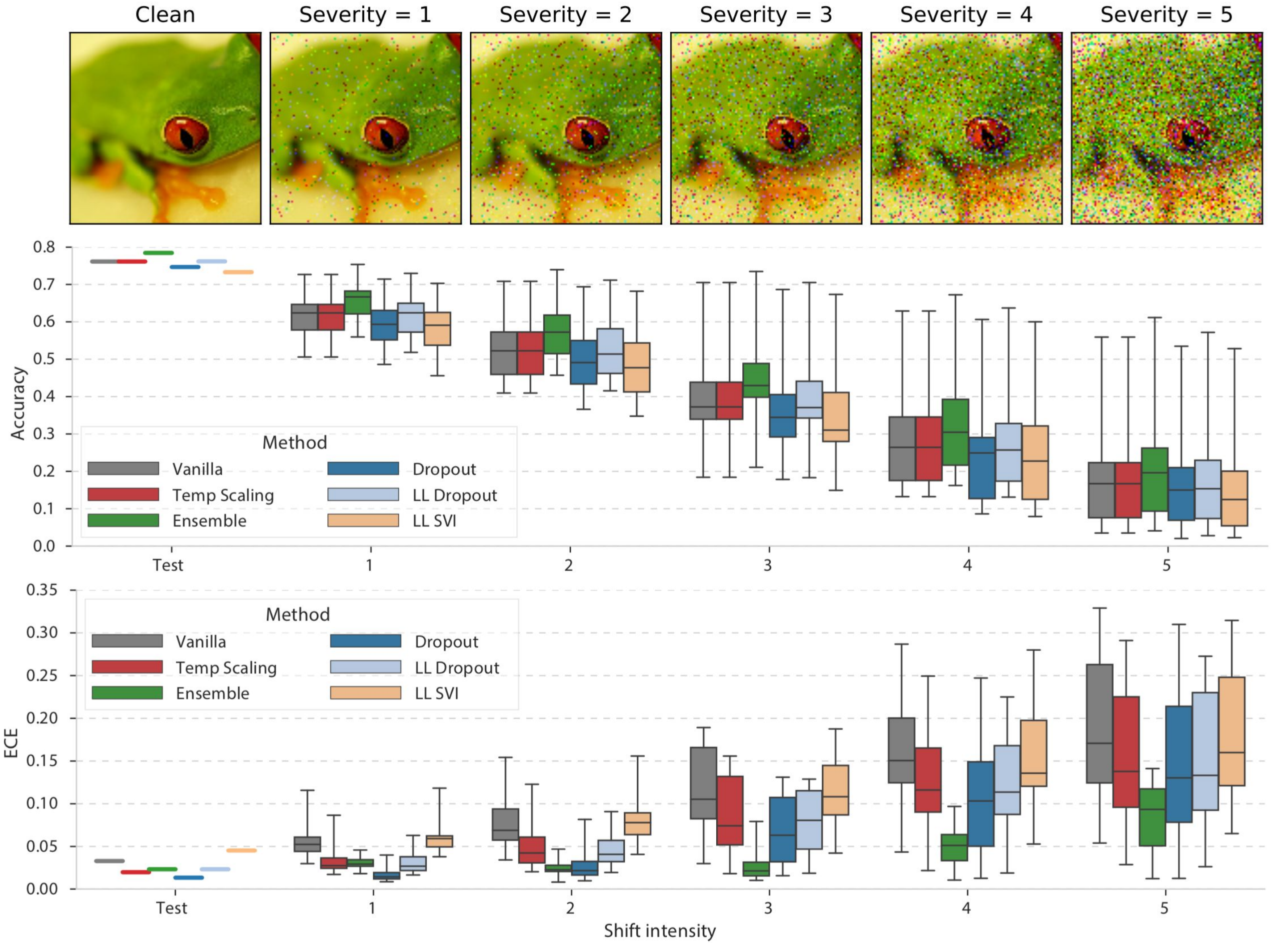
Image source: [Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift?, Ovadia et al. 2019]

Credits: slide from B. Lakshminarayanan, D. Tran, J. Snoek "Uncertainty and Out-of-Distribution Robustness in Deep Learning"

Models are not calibrated under dataset shift

- **Accuracy drops** with increasing shift on Imagenet-C

- **Calibration degrades** with shift: "*overconfident mistakes*"



Credits: slide from B. Lakshminarayanan, D. Tran, J. Snoek [“Uncertainty and Out-of-Distribution Robustness in Deep Learning”]

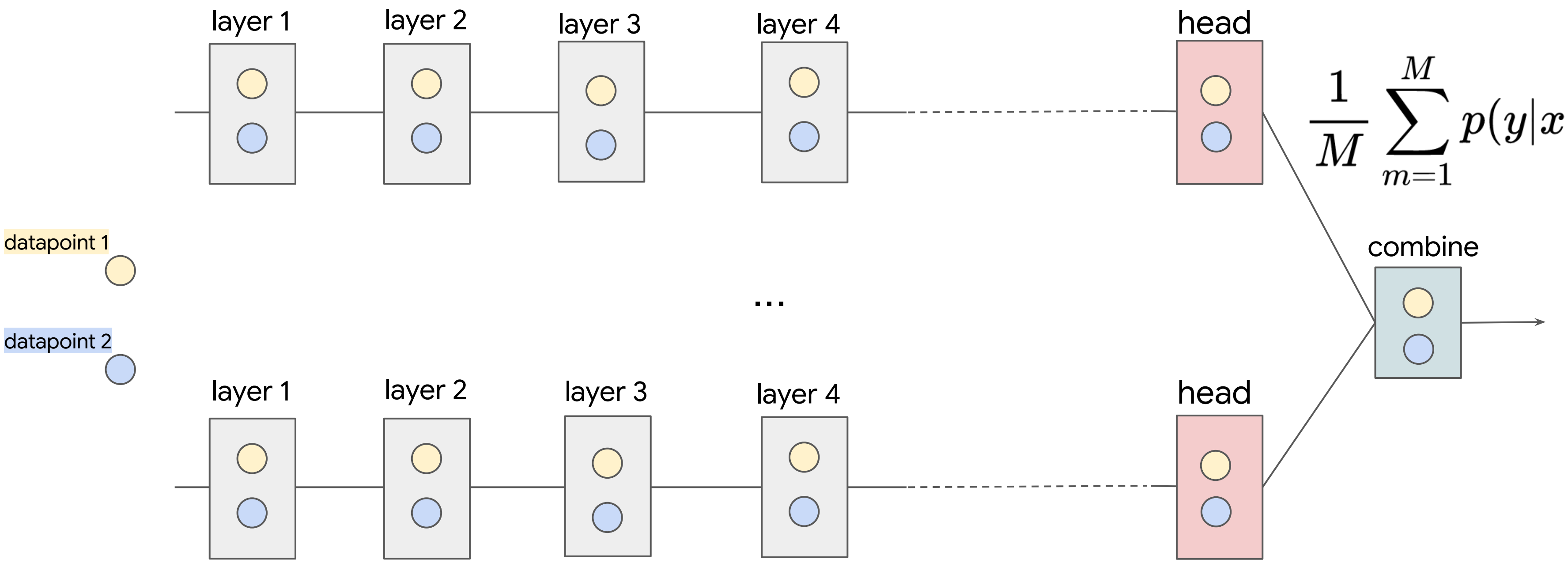
Context and motivations

Project at the intersection of two topics:

- **Sparse mixture of experts (sparse MoEs)**
- **Reliability in deep learning**
  - **Why and how to measure it?**
  - **Ensembles**

Primer on deep ensembles [Lakshminarayanan et al. 2017, ..., Hansen et al., 1990]

Google AI
Brain Team

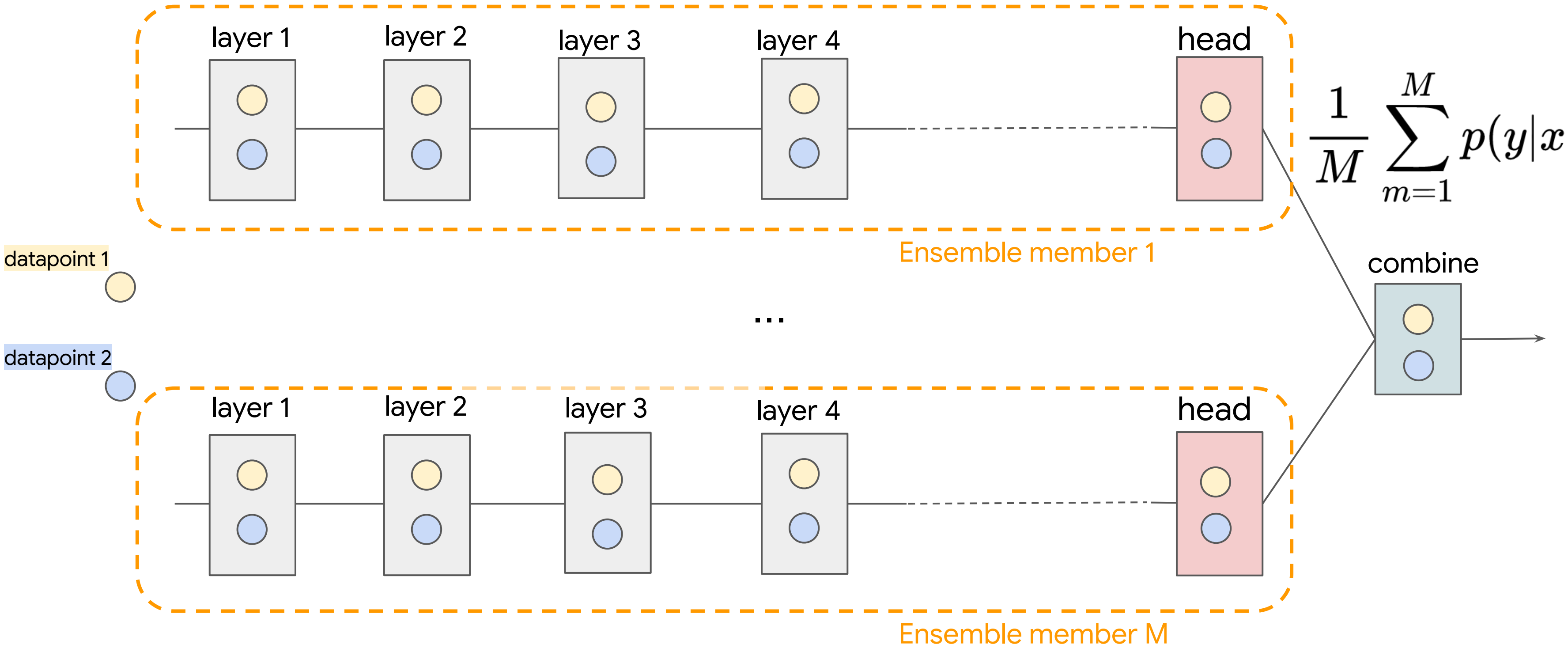


$$\frac{1}{M}\sum_{m=1}^{M} p(y|x;\theta_m)$$

- Multiple trainings from different seeds
- Average the predictions
- Simple...but expensive

Primer on deep ensembles [Lakshminarayanan et al. 2017, ..., Hansen et al., 1990]

Google AI
Brain Team

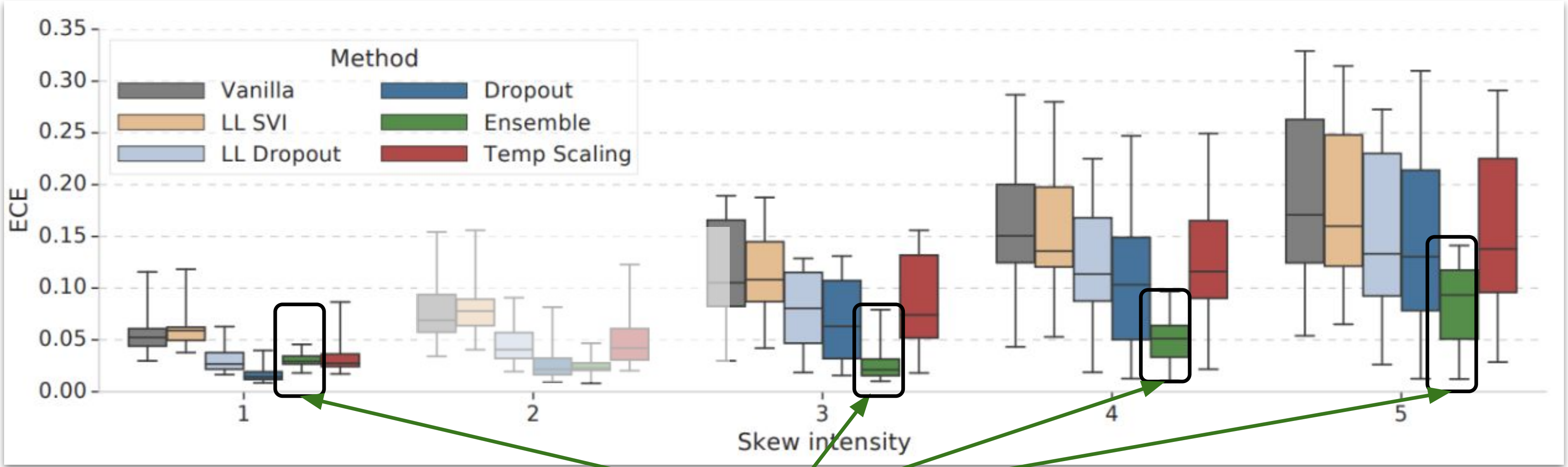


$$\frac{1}{M}\sum_{m=1}^{M} p(y|x;\theta_m)$$

- Multiple trainings from different seeds (**M** ensemble members)
- Average the predictions
- Simple...but expensive

Deep ensembles work surprisingly well in practice

Google AI
Brain Team



Deep ensembles are consistently among the best performing methods, especially under dataset shift

Credits: slide from B. Lakshminarayanan, D. Tran, J. Snoek "Uncertainty and Out-of-Distribution Robustness in Deep Learning"

Sparse MoEs vs. ensembles: Overview of properties

Sparse MoEs	Ensembles
Single prediction	Multiple predictions

Sparse MoEs vs. ensembles: Overview of properties

Sparse MoEs	Ensembles
Single prediction	Multiple predictions
Per-example adaptivity	Static combination

Sparse MoEs vs. ensembles: Overview of properties

Sparse MoEs	Ensembles
Single prediction	Multiple predictions
Per-example adaptivity	Static combination
Combination at activation level	Combination at prediction level

Sparse MoEs vs. ensembles: Overview of properties

Sparse MoEs	Ensembles
Single prediction	Multiple predictions
Per-example adaptivity	Static combination
Combination at activation level	Combination at prediction level
???	Robust to distribution shift Strong OOD detection Well-calibrated uncertainty
Strong few-shot performance	???

Sparse MoEs vs. ensembles: Overview of properties

Sparse MoEs	Ensembles
Single prediction	Multiple predictions
Per-example adaptivity	Static combination
Combination at activation level	Combination at prediction level
???	Robust to distribution shift Strong OOD detection Well-calibrated uncertainty
Strong few-shot performance	???
Compute ≈ standard NN	Compute ≫ standard NN

Sparse MoEs vs. ensembles: Overview of properties

Sparse MoEs	**Ensembles**
Single prediction	Multiple predictions
Per-example adaptivity	Static combination
Combination at activation level	Combination at prediction level
???	Robust to distribution shift Strong OOD detection Well-calibrated uncertainty
Strong few-shot performance	???
Compute ≈ standard NN	Compute ≫ standard NN

Goals:

- Understanding the interplay between those two classes of models
- Design approaches “taking the best of both worlds”

Interplay between sparse MoEs & ensembles

Google AI Brain Team

Some questions of interest:

- **Can we combine ensembles and sparse MoEs?**

Interplay between sparse MoEs & ensembles

Some questions of interest:

- **Can we combine ensembles and sparse MoEs?**
- **What are the most import factors?**
  - M: The number of ensemble members.
  - K: The sparsity, i.e., the number of selected experts.
  - (E: The total number of experts.)

$\left.\right\}$ **"Static" combination**

$\left.\right\}$ **"Adaptive" combination**

(Small digression: Experiment setup)

"**Upstream**", assume checkpoints:

- Pretrained on JFT-300M (~18k classes)
- Vision transformers (ViT) [Dosovitskiy et al., 2020]
- Sparse MoEs (V-MoE) [Riquelme et al., 2021]

"**Downstream**", fine-tuning:

- ImageNet, Cifar10, Cifar100,...
- ViT, V-MoE
- (all other models assumed compatible with checkpoints)

(Small digression: Experiment setup)

Image credit

Vision transformers with different backbone sizes: “`scale/patch_size`”

(Small digression: Experiment setup)

Image credit

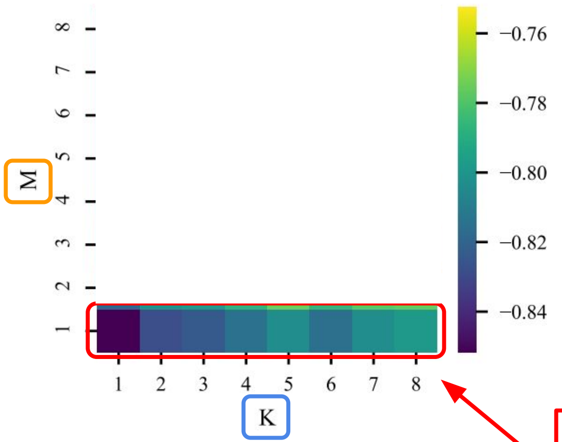
Vision transformers with different backbone sizes: "`scale/patch_size`"

- S/32, B/32, L/32, B/16, L/16, H/14 (from 36.5M to 2.7B params)

	Hidden dimension	MLP dimension	# layers
Small	512	2048	8
Base	768	3072	12
Large	1024	4096	24
Huge	1280	5144	32

“Static” vs. “adaptive” combination

- Deep ensembles of **M** x V-MoEs with sparsity **K**
- ImageNet fine-tuned performance

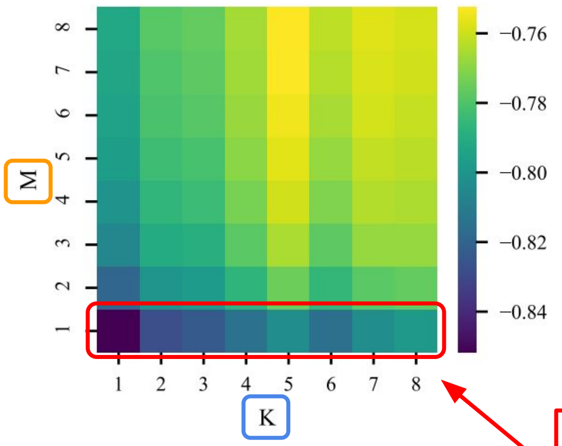


(a) Log-Likelihood

Standard V-MoE (no ensemble)
More selected experts help [Riquelme et al., 2021]

“Static” vs. “adaptive” combination

- Deep ensembles of **M** x V-MoEs with sparsity **K**
- ImageNet fine-tuned performance



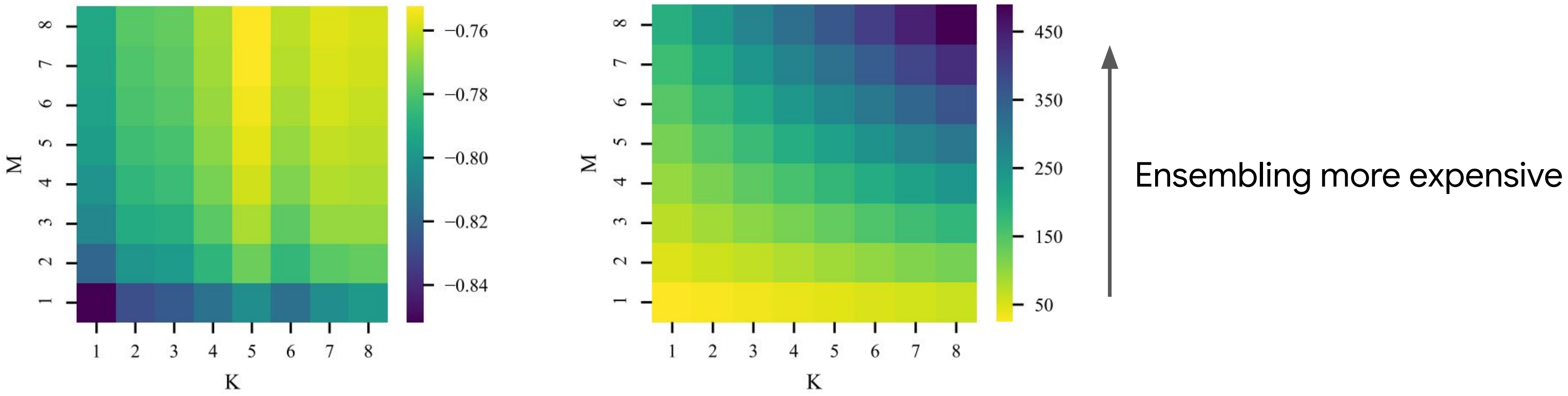
(a) Log-Likelihood

Ensembling beneficial at all sparsity levels

Standard V-MoE (no ensemble)
More selected experts help [Riquelme et al., 2021]

“Static” vs. “adaptive” combination

- What about the cost? (GFLOPs = 10^9 FLOPs)

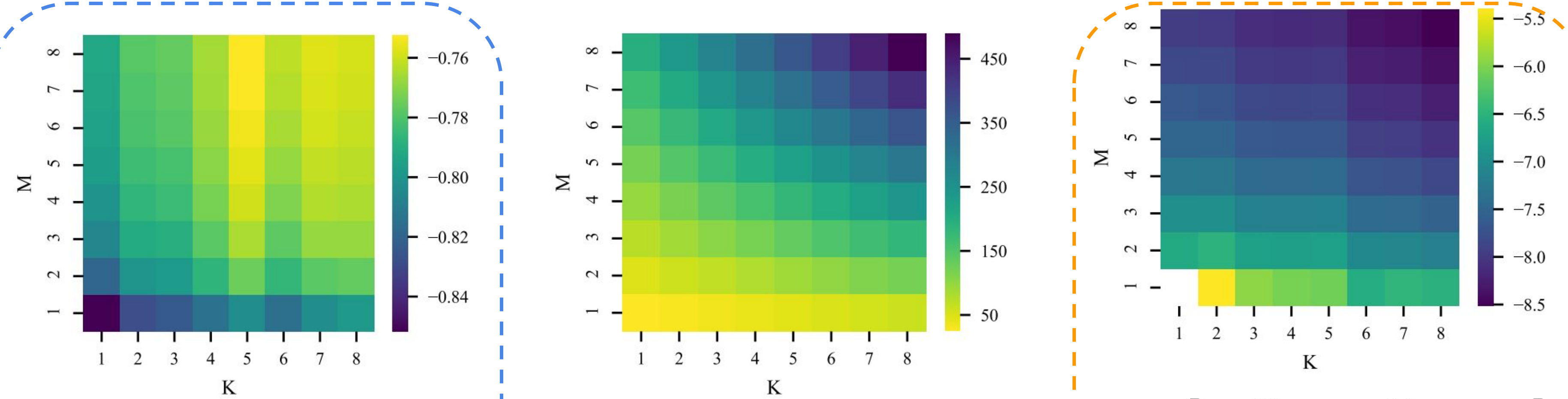


(a) Log-Likelihood

(b) (downstream) GFLOPs

“Static” vs. “adaptive” combination

- More cost-effective to use “adaptive” combination
- ...but best absolute performance requires both types of combinations

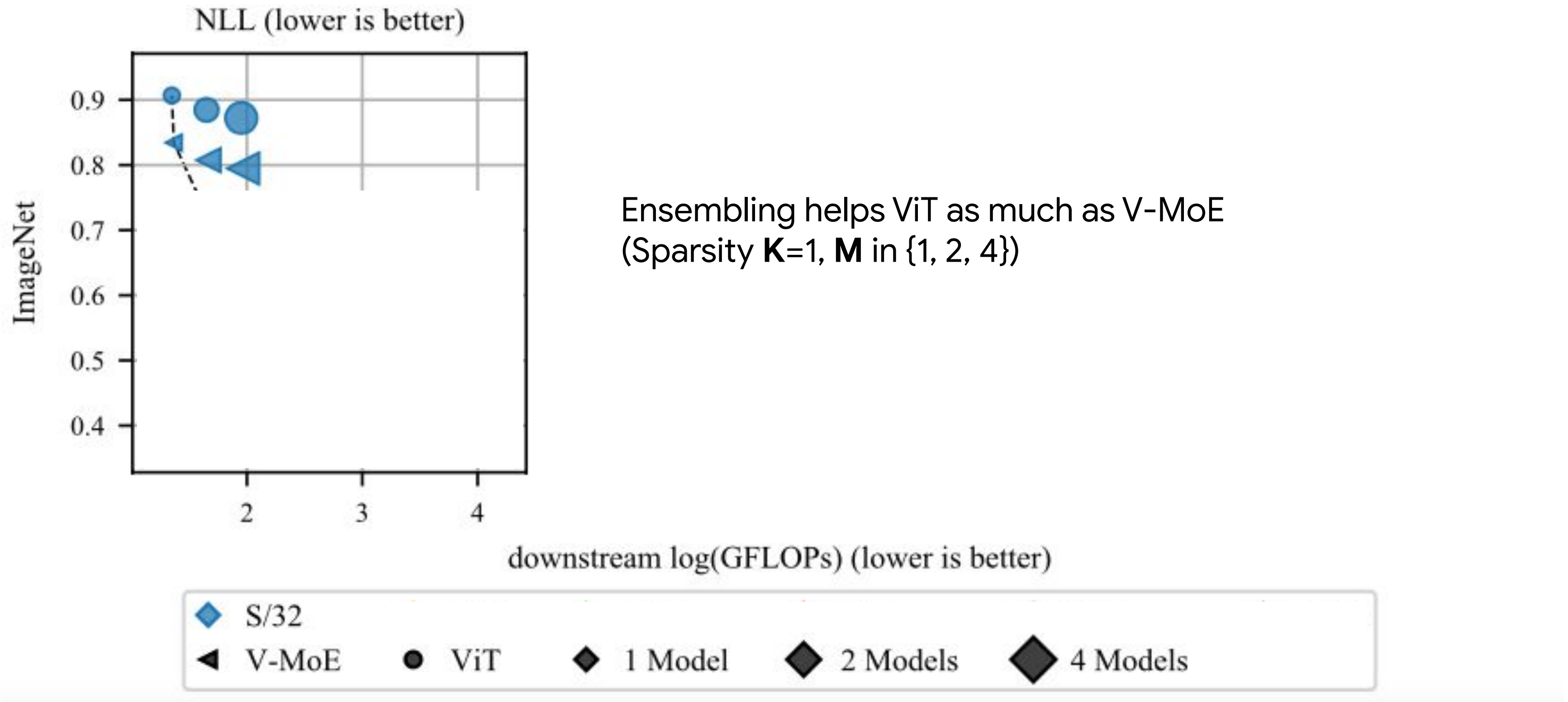


(a) Log-Likelihood

(b) (downstream) GFLOPs

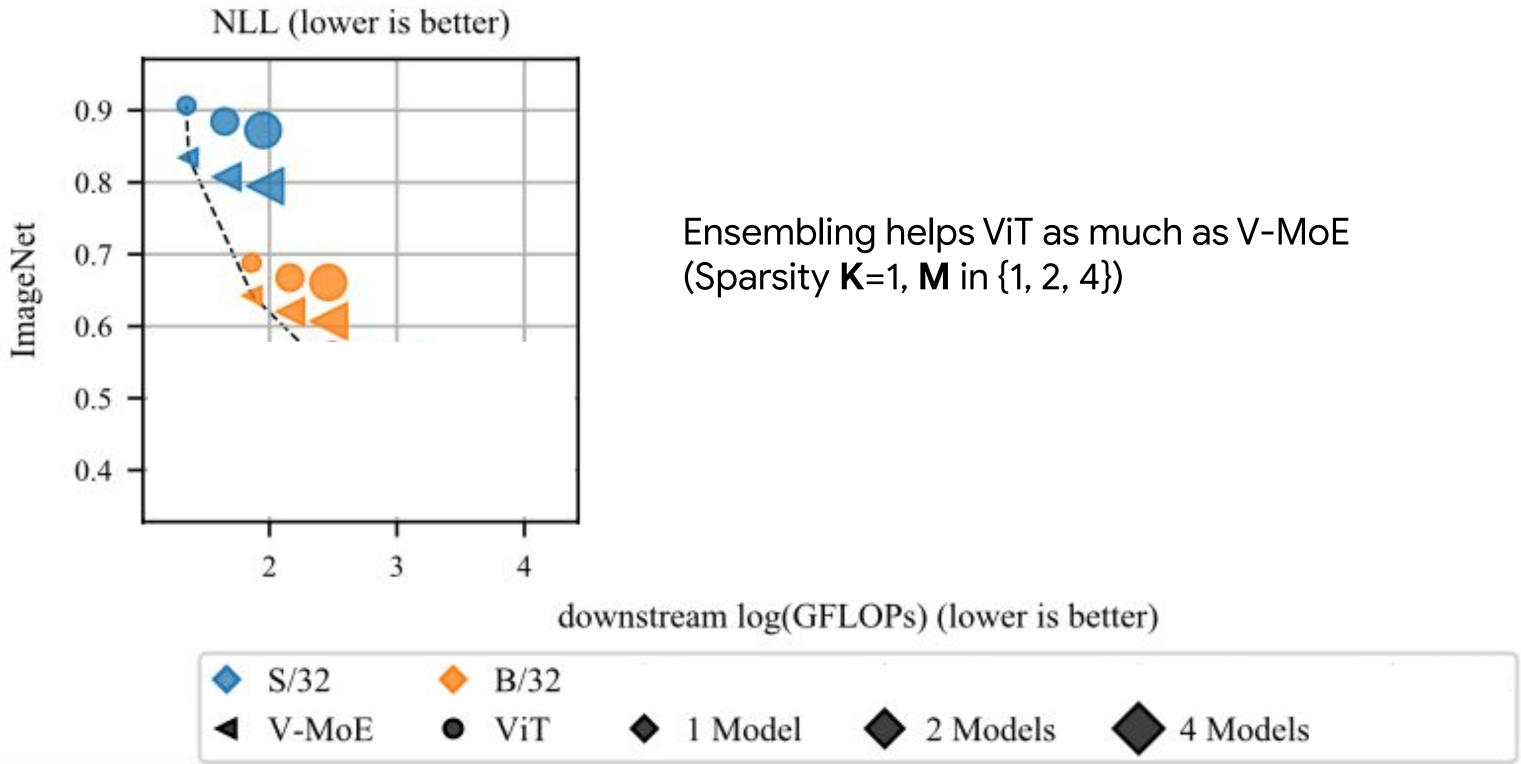
(c) $\log\left[\frac{\mathrm{LL}_{(K,M)} - \mathrm{LL}_{(1,1)}}{\mathrm{GFLOPs}_{(K,M)} - \mathrm{GFLOPs}_{(1,1)}}\right]$

Is ensembling equally helpful for ViT & V-MoE? (ImageNet)



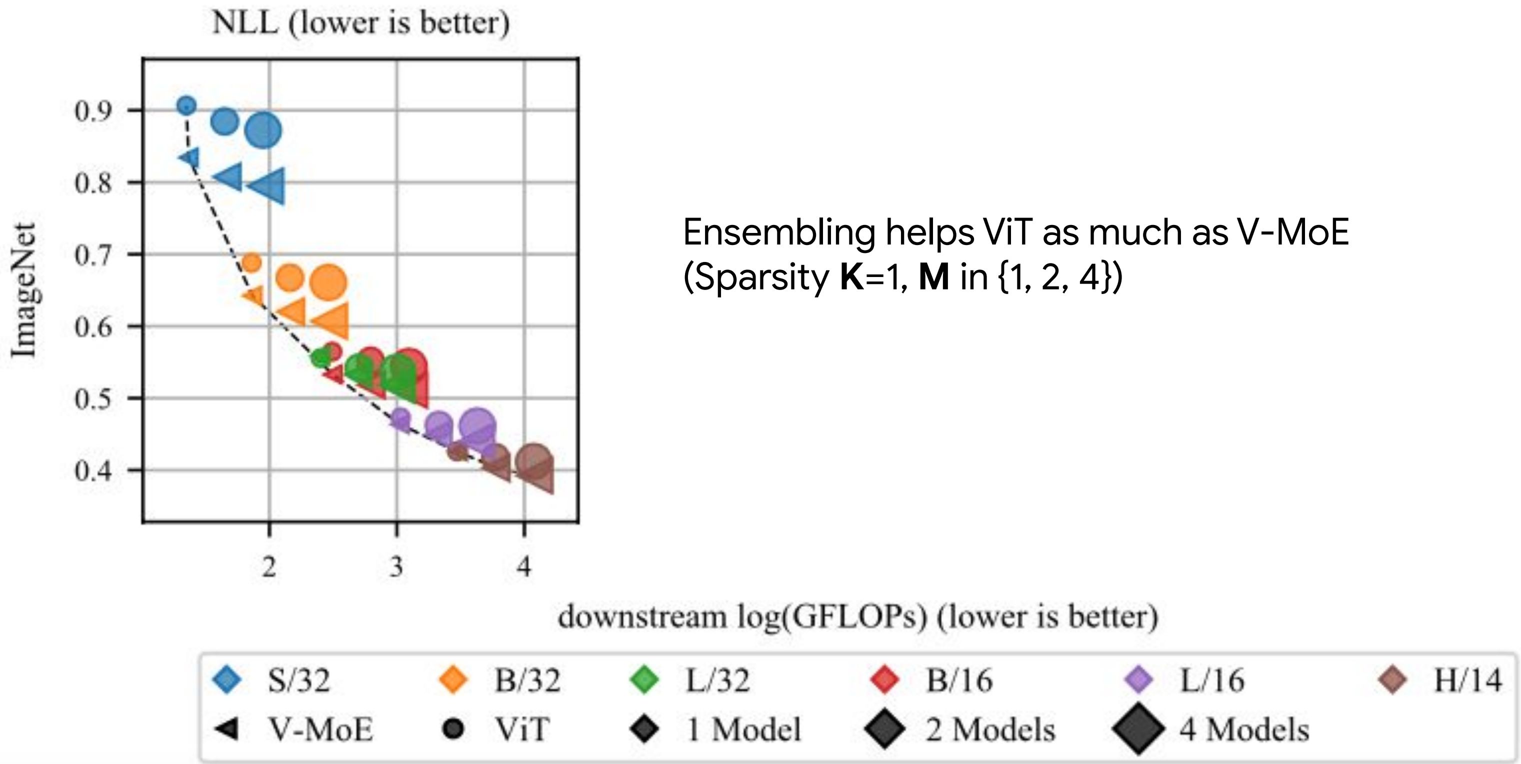
Ensembling helps ViT as much as V-MoE
(Sparsity **K**=1, **M** in {1, 2, 4})

Is ensembling equally helpful for ViT & V-MoE? (ImageNet)



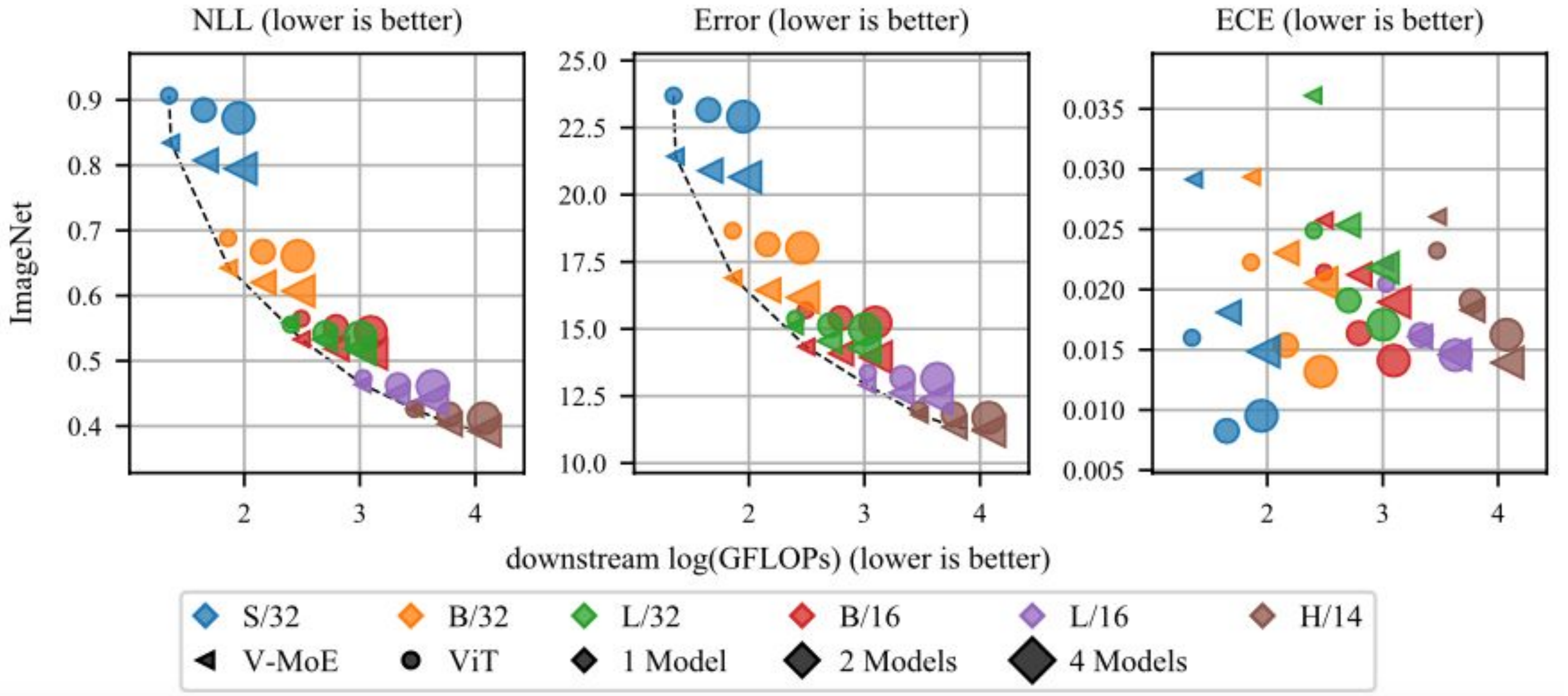
Ensembling helps ViT as much as V-MoE
(Sparsity **K**=1, **M** in {1, 2, 4})

Is ensembling equally helpful for ViT & V-MoE? (ImageNet)

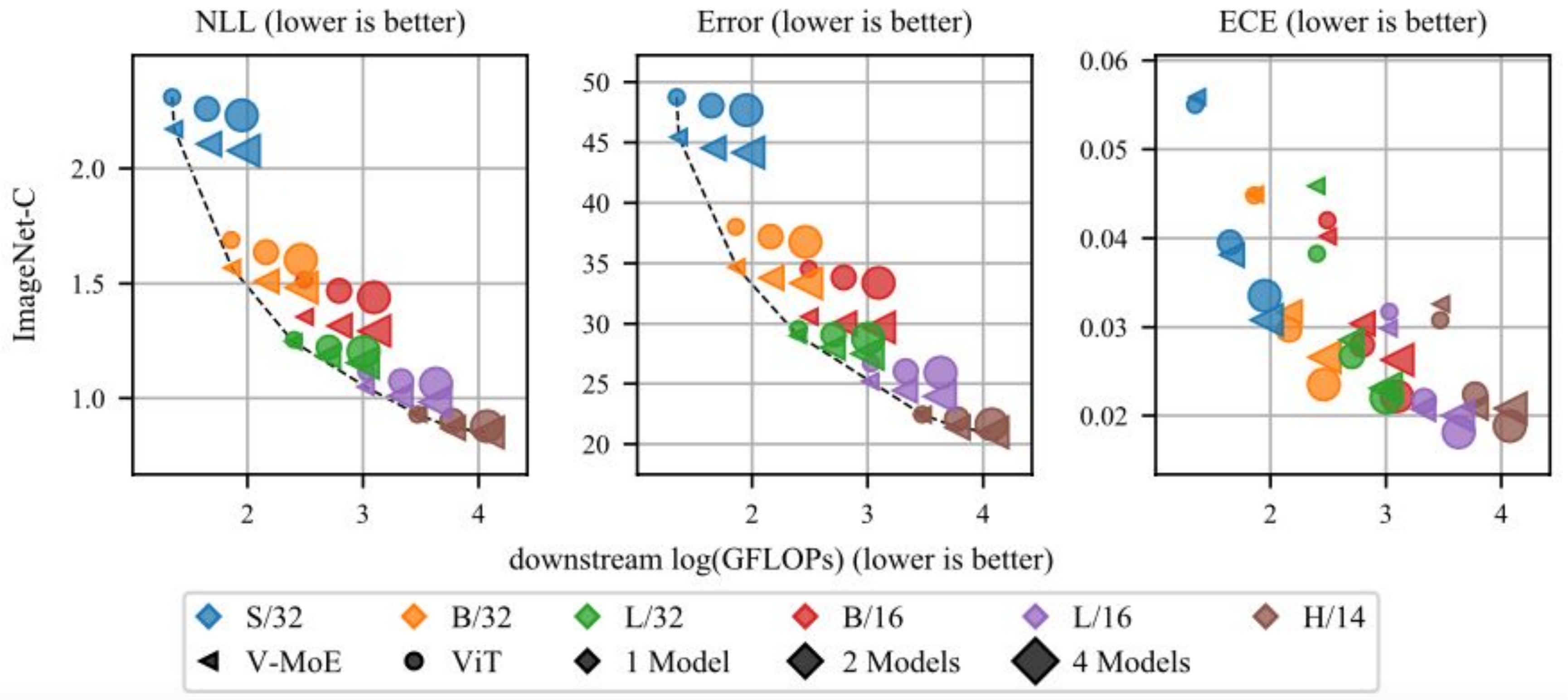


Ensembling helps ViT as much as V-MoE
(Sparsity **K**=1, **M** in {1, 2, 4})

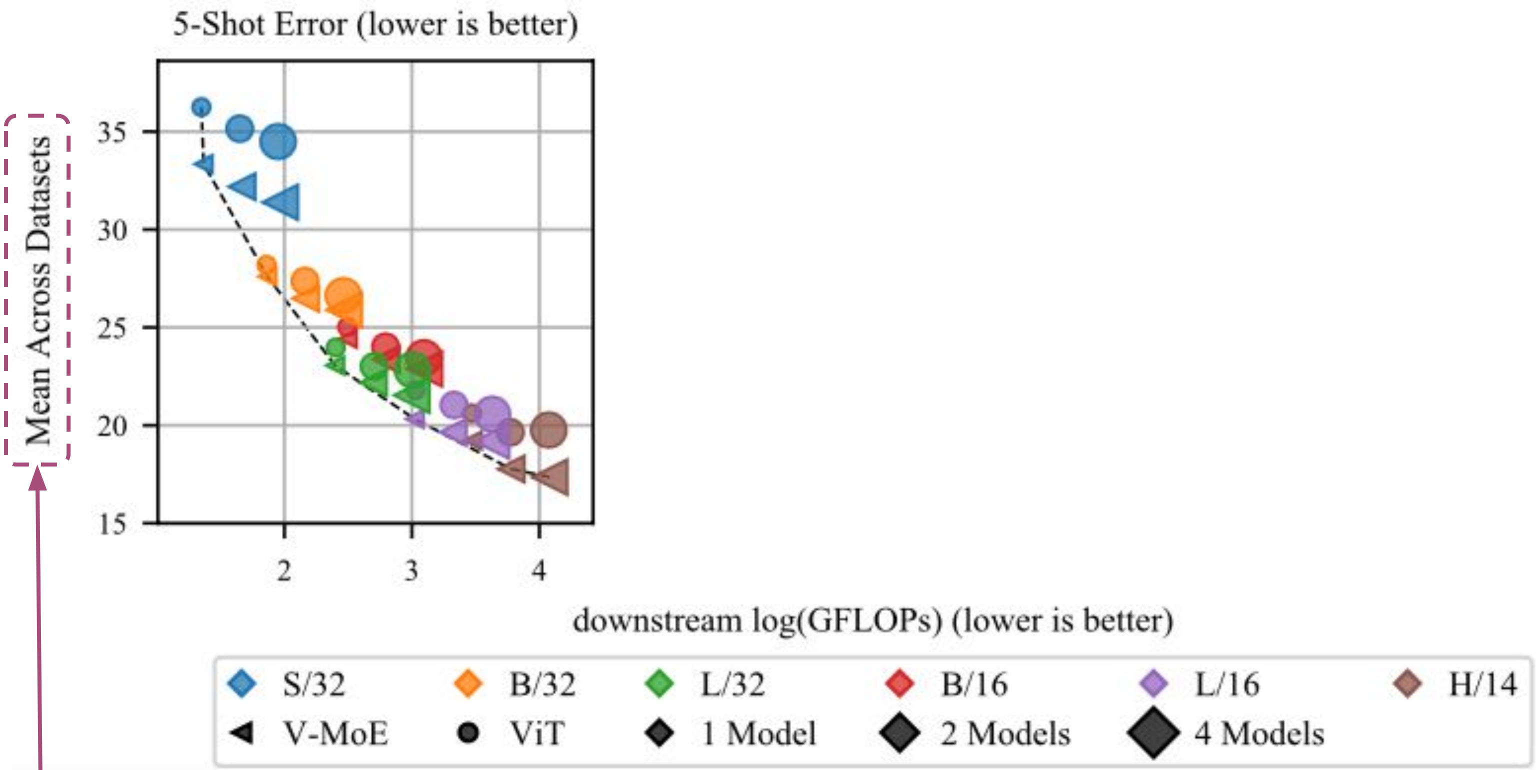
Is ensembling equally helpful for ViT & V-MoE? (ImageNet)



Is ensembling equally helpful for ViT & V-MoE? (ImageNet-C)

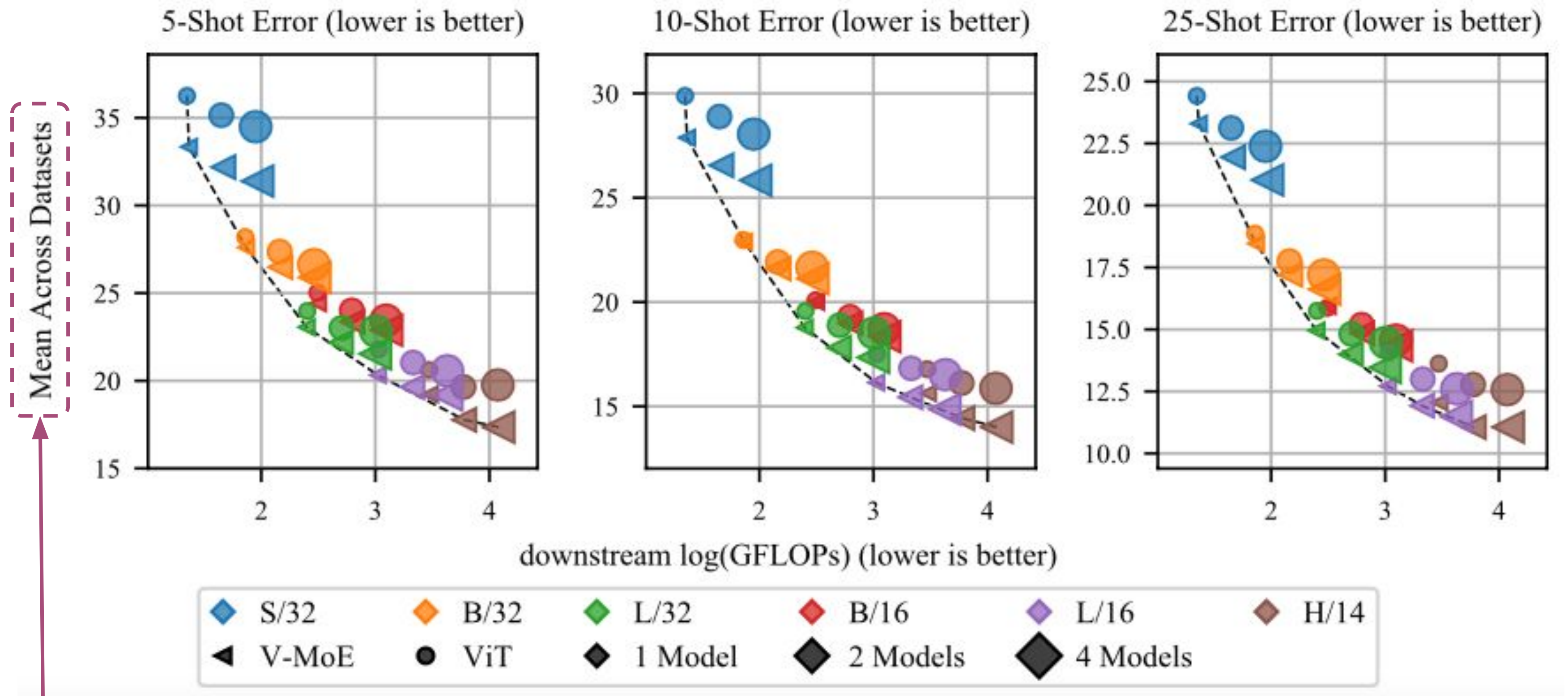


Is ensembling equally helpful for ViT & V-MoE? (few-shot)



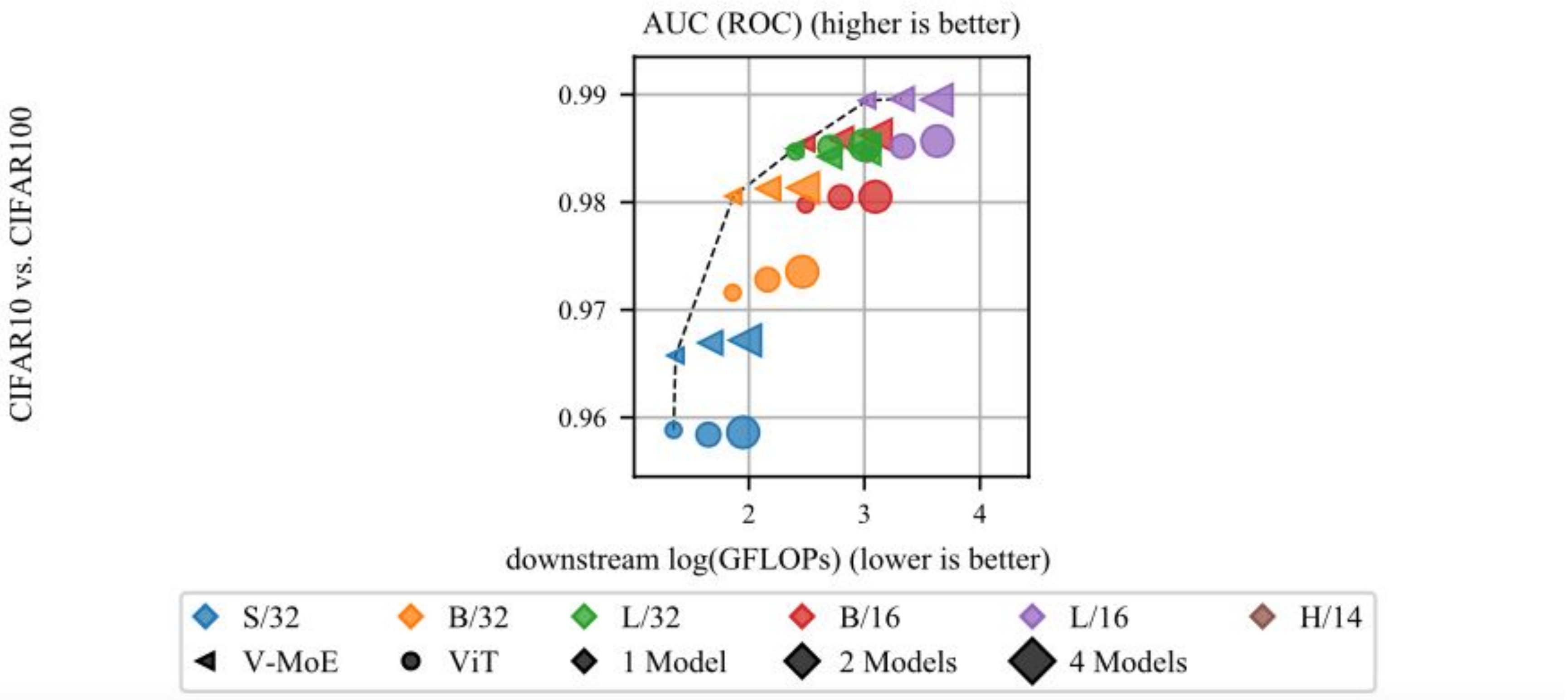
Linear few-shot [Dosovitskiy et al., 2020, Riquelme et al., 2021], aggregated over 8 datasets

Is ensembling equally helpful for ViT & V-MoE? (few-shot)

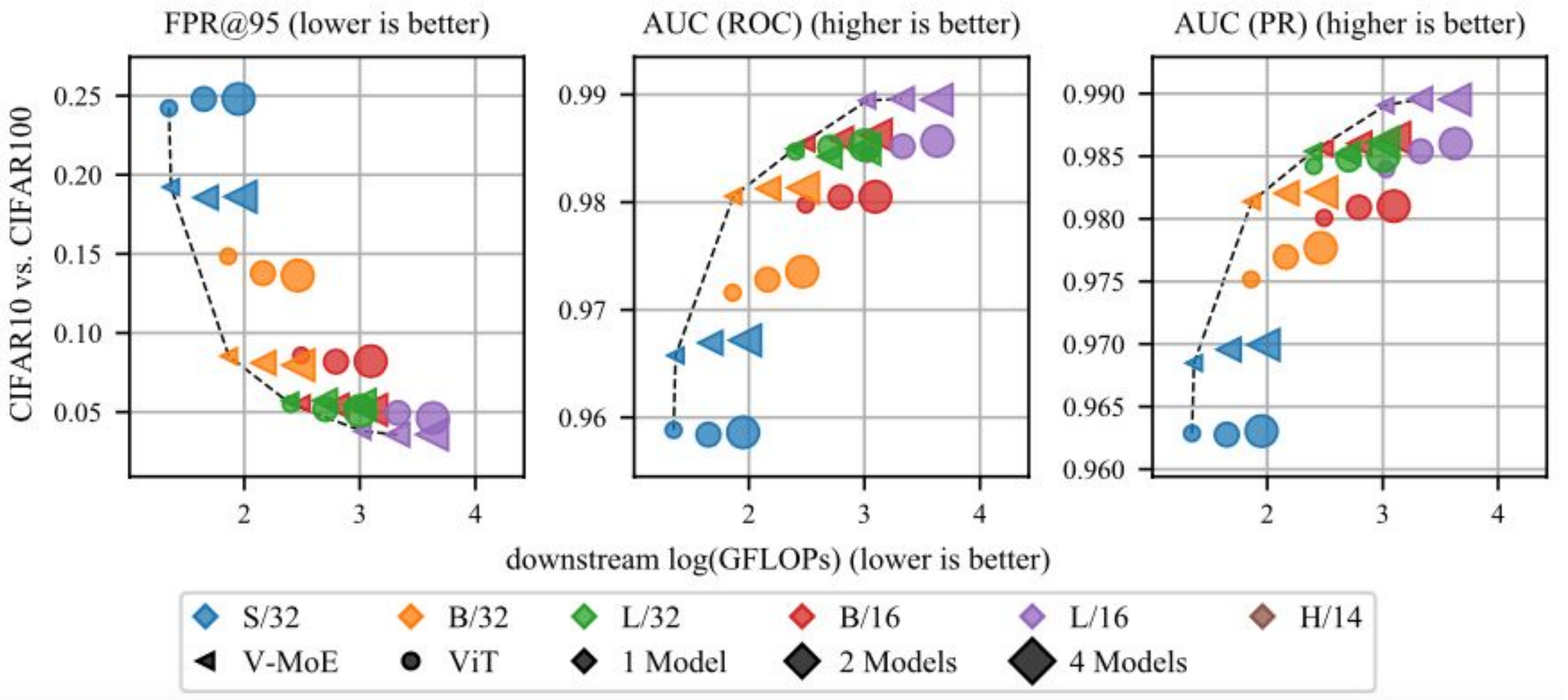


Linear few-shot [Dosovitskiy et al., 2020, Riquelme et al., 2021], aggregated over 8 datasets

Is ensembling equally helpful for ViT & V-MoE? (OOD detection)



Is ensembling equally helpful for ViT & V-MoE? (OOD detection)



Summary so far:

Some questions of interest:

- **Can we combine ensembles and sparse MoEs?** →  ... but costly 

Summary so far:

Some questions of interest:

- **Can we combine ensembles and sparse MoEs?** →  ... but costly 
- **What are the most import factors?**
    - M: The number of ensemble members.
    - K: The sparsity, i.e., the number of selected experts.

"Adaptive" combination more cost-effective but both "static" & "adaptive" are needed.

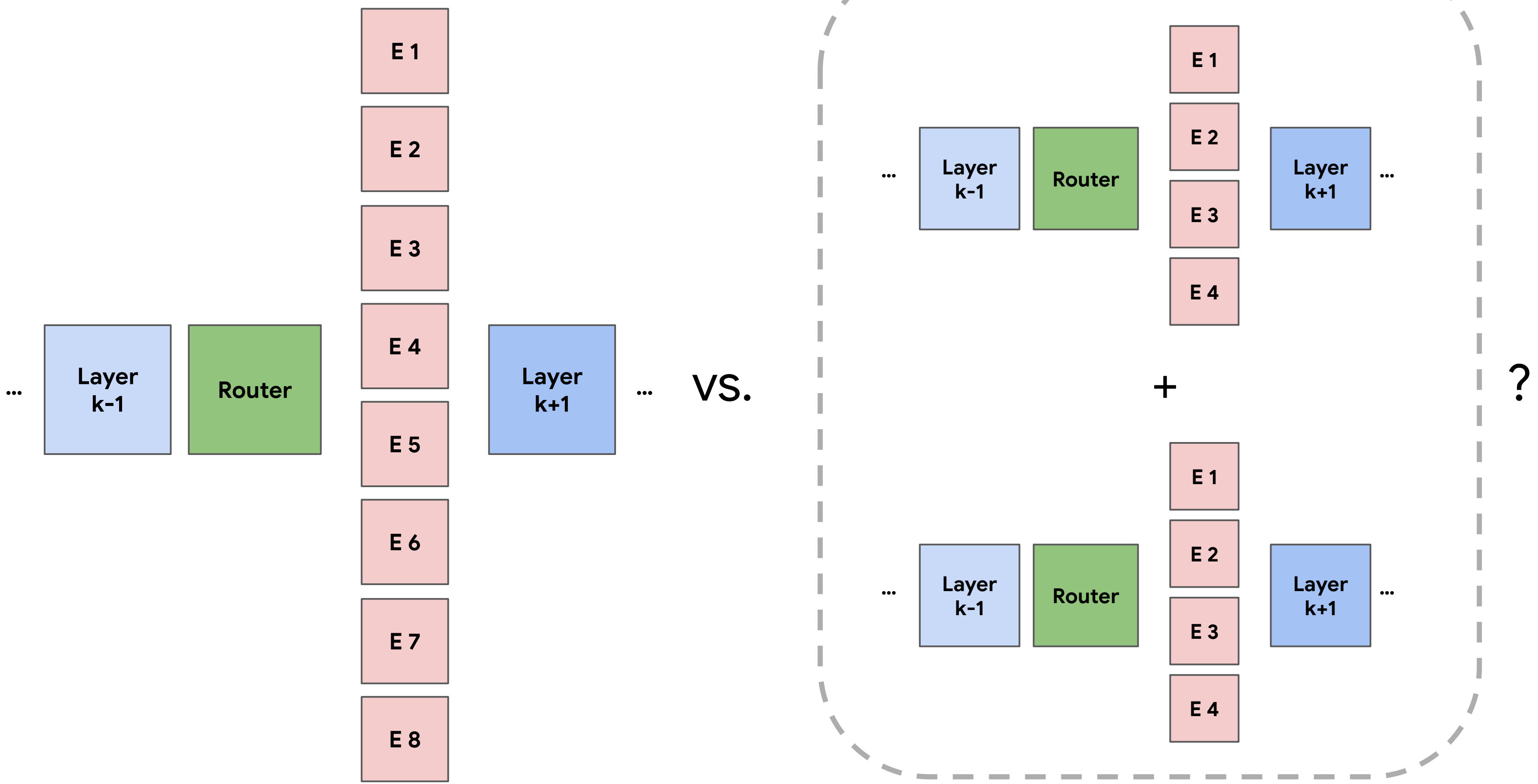
Summary so far:

Some questions of interest:

- **Can we combine ensembles and sparse MoEs?** →  ... but costly 
- **What are the most import factors?**
  - M: The number of ensemble members.
  - K: The sparsity, i.e., the number of selected experts.

  "Adaptive" combination more cost-effective but both "static" & "adaptive" are needed.
- **Can we do better than a naive ensemble of sparse MoEs?**

Motivating experiment



Motivating experiment: V-MoE-B/32 (K=1, E=32) on ImageNet

M	E	NLL ↓	ERROR ↓
1	32	0.642 ± 0.002	16.90 ± 0.05
2	32	0.588	15.74
4	32	0.561	15.10

Already observed:
Ensembling V-MoEs helps

Motivating experiment: V-MoE-B/32 (K=1, E=32) on ImageNet

M	E	NLL ↓	ERROR ↓
1	32	0.642 ± 0.002	16.90 ± 0.05
2	32	0.588	15.74
4	32	0.561	15.10
2	16	0.588	15.97
4	8	0.577	15.82

Main observation:
Ensemble of smaller sparse MoEs > One single larger sparse MoE

Efficient Ensemble of Experts (E^3)

Constructing an efficient ensembles of sparse MoEs:

- **Idea**: End-to-end training of **M** x sparse MoEs with **E/M** experts

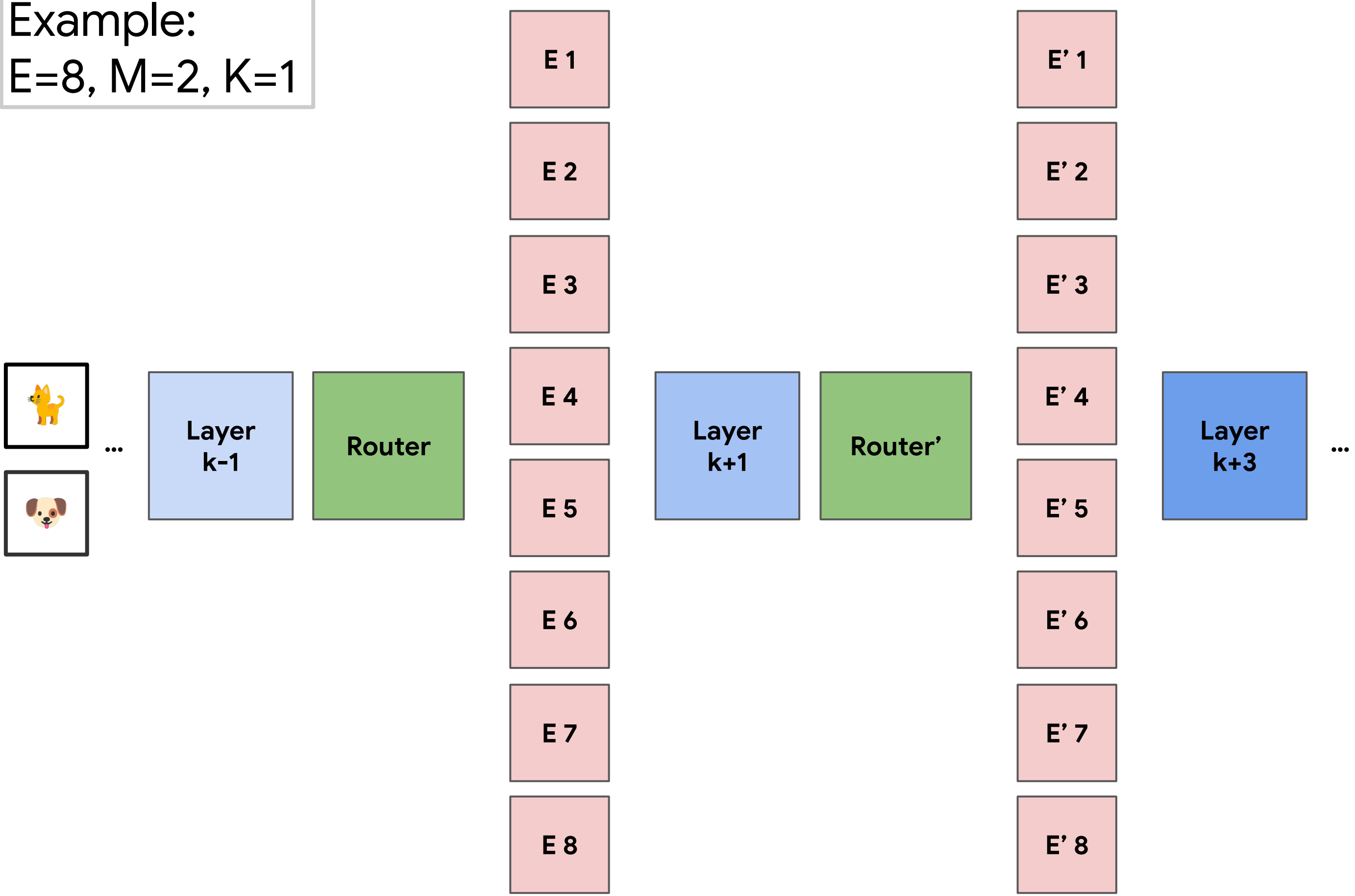
Efficient Ensemble of Experts (E^3)

Constructing an efficient ensembles of sparse MoEs:

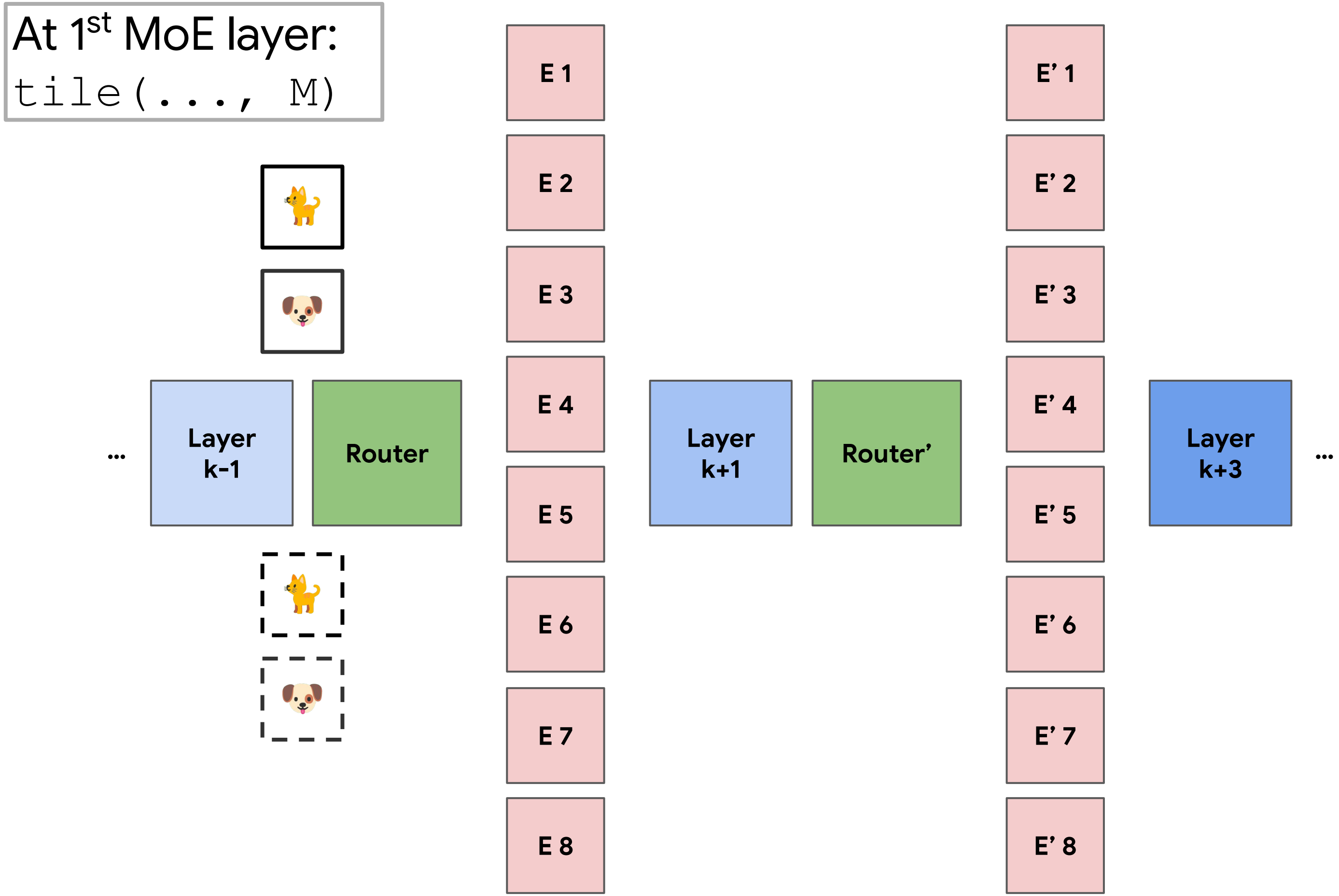
- **Idea**: End-to-end training of **M** x sparse MoEs with **E/M** experts
  - An ensemble member = one sparse MoEs with **E/M** experts
  - Efficient simultaneous training via a tiled representation
  - Sharing of parameters in non-expert layers
- Inspired by batch ensemble [Wen et al., 2019]

Pictorial view of Efficient Ensemble of Experts (E^3)

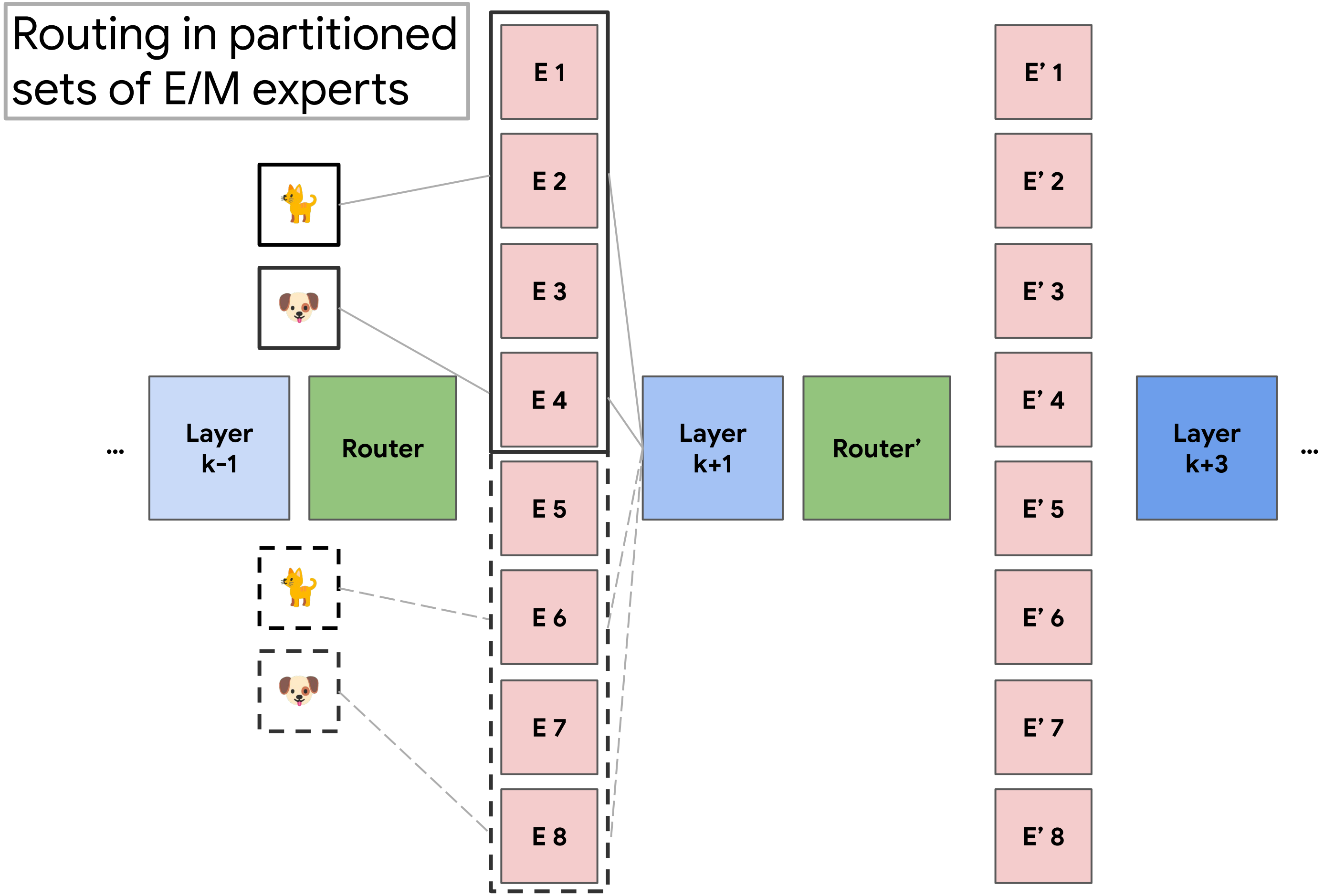
Example:
E=8, M=2, K=1



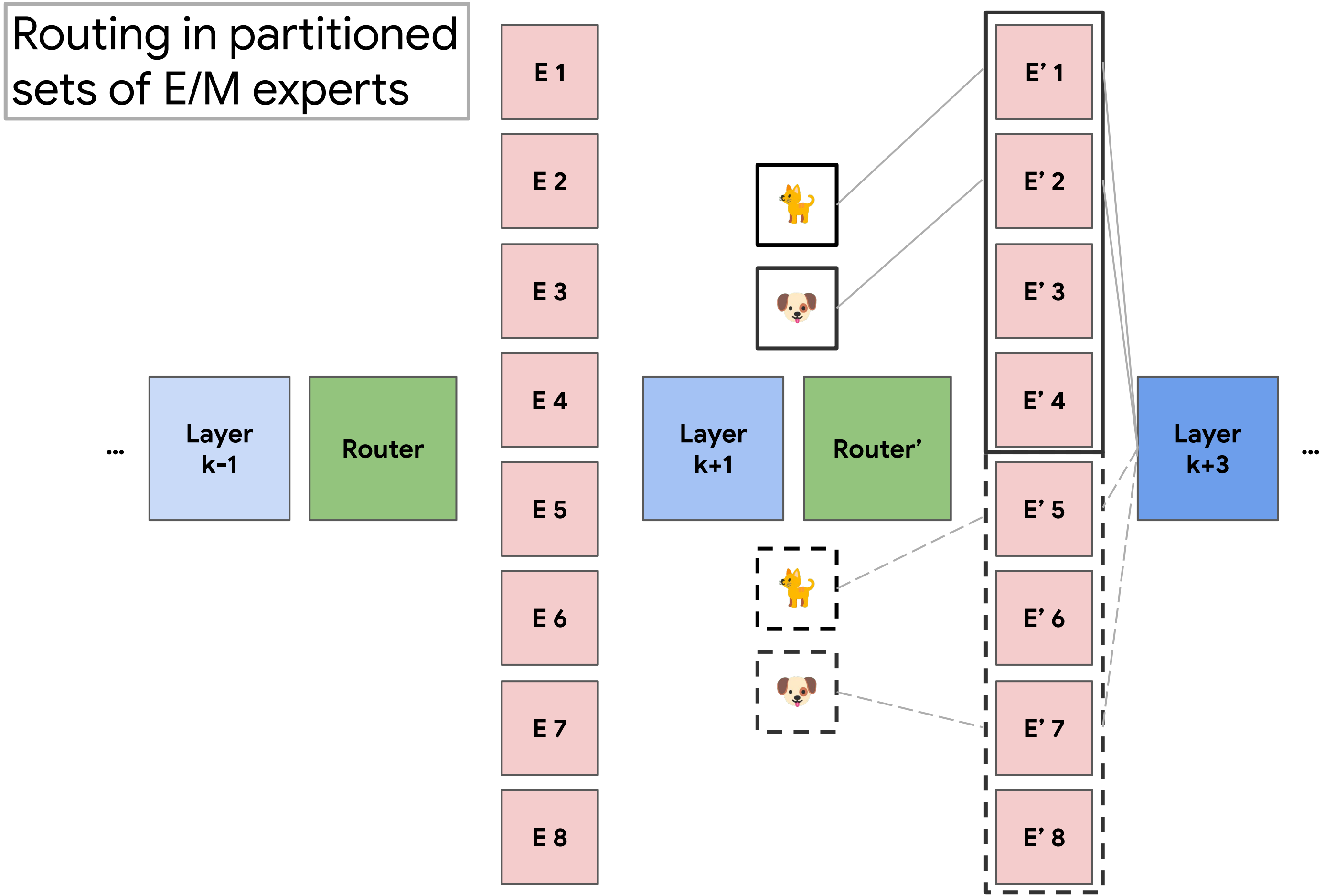
Pictorial view of Efficient Ensemble of Experts (E^3)



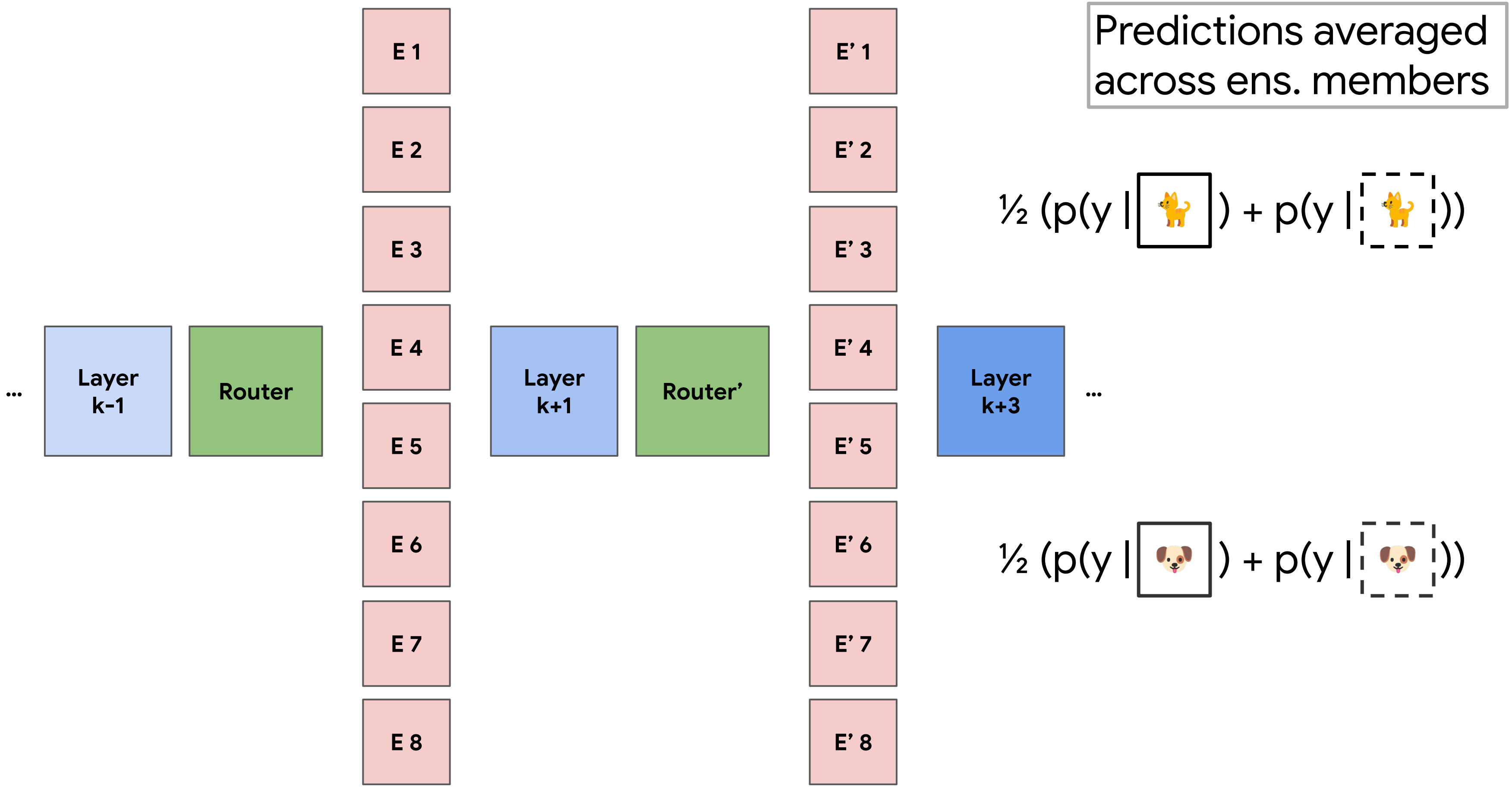
Pictorial view of Efficient Ensemble of Experts (E^3)



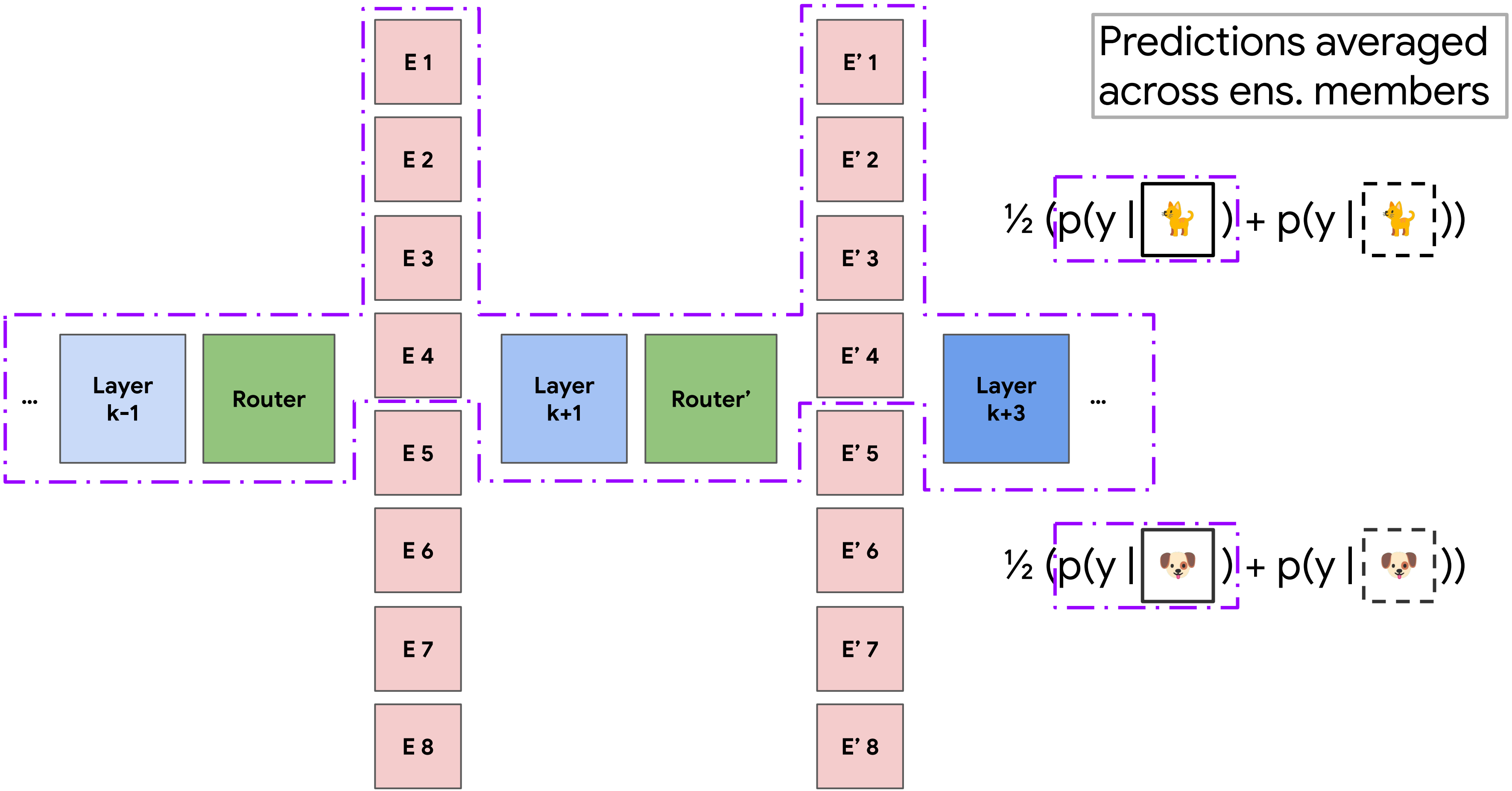
Pictorial view of Efficient Ensemble of Experts (E^3)



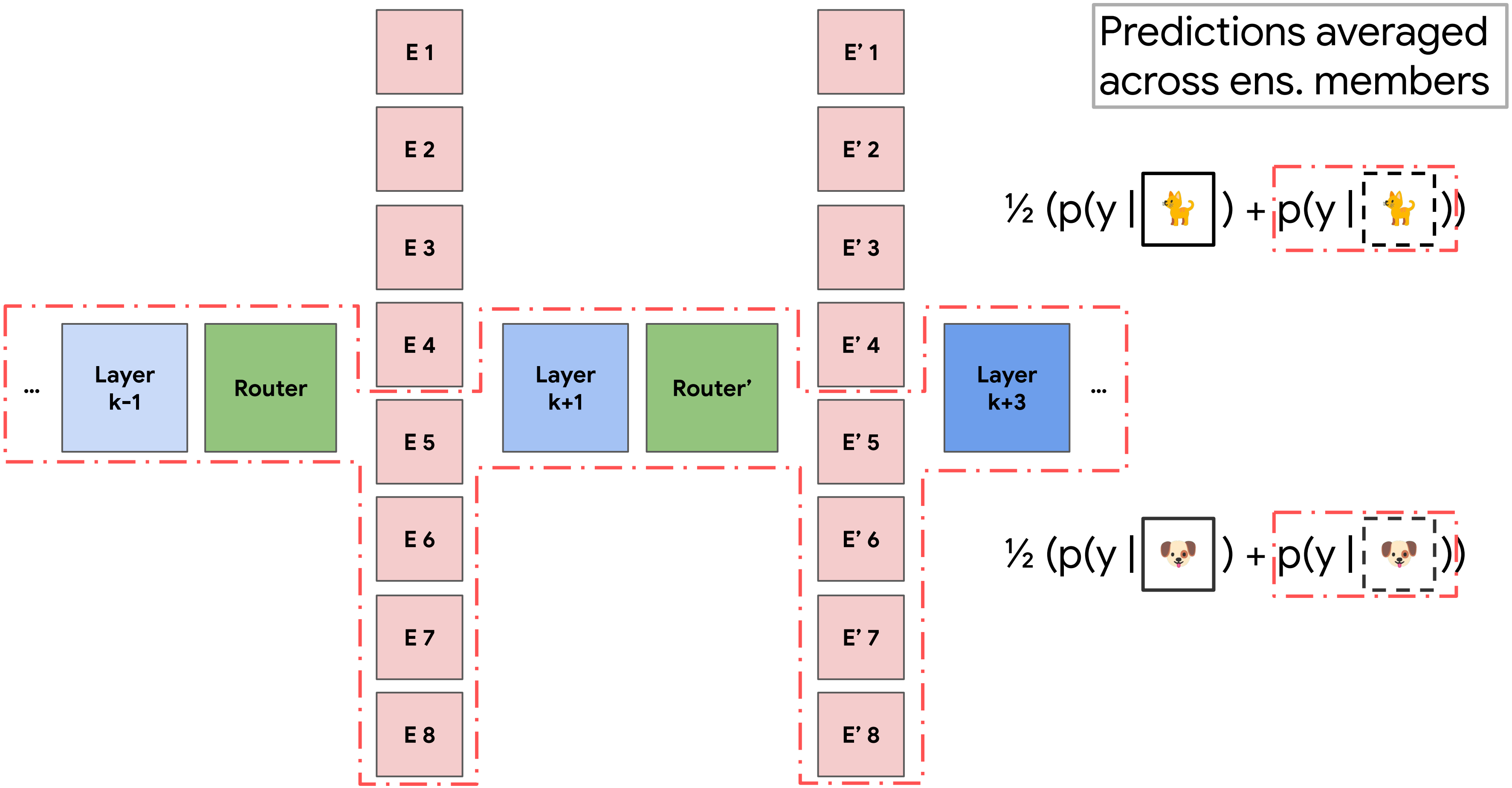
Pictorial view of Efficient Ensemble of Experts (E^3)



Pictorial view of Efficient Ensemble of Experts (E^3)

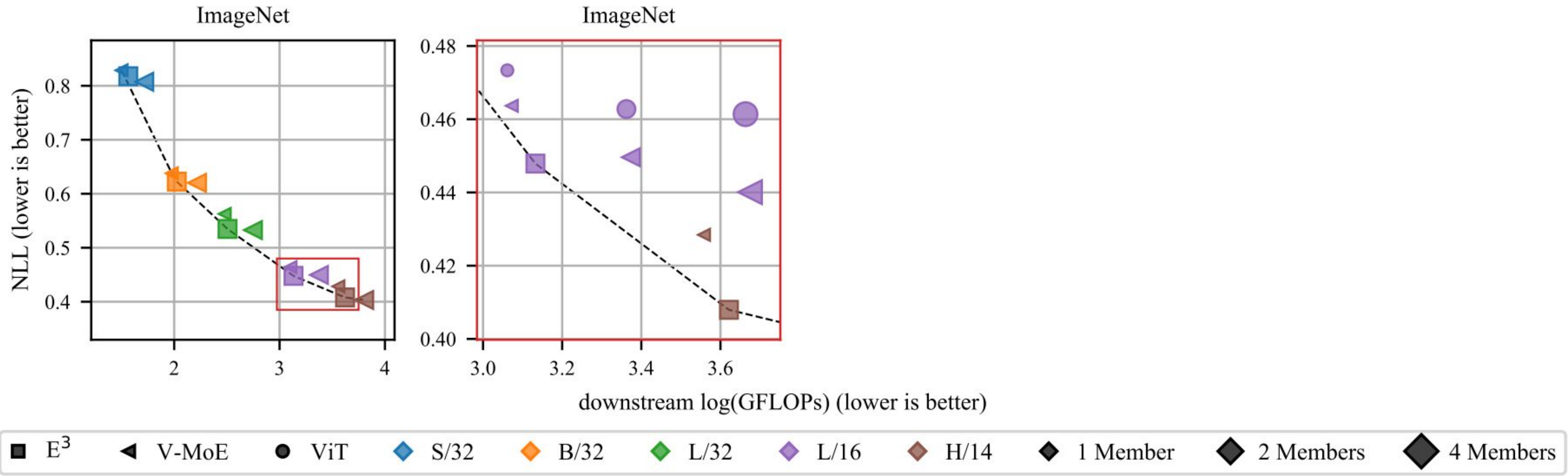


Pictorial view of Efficient Ensemble of Experts (E^3)



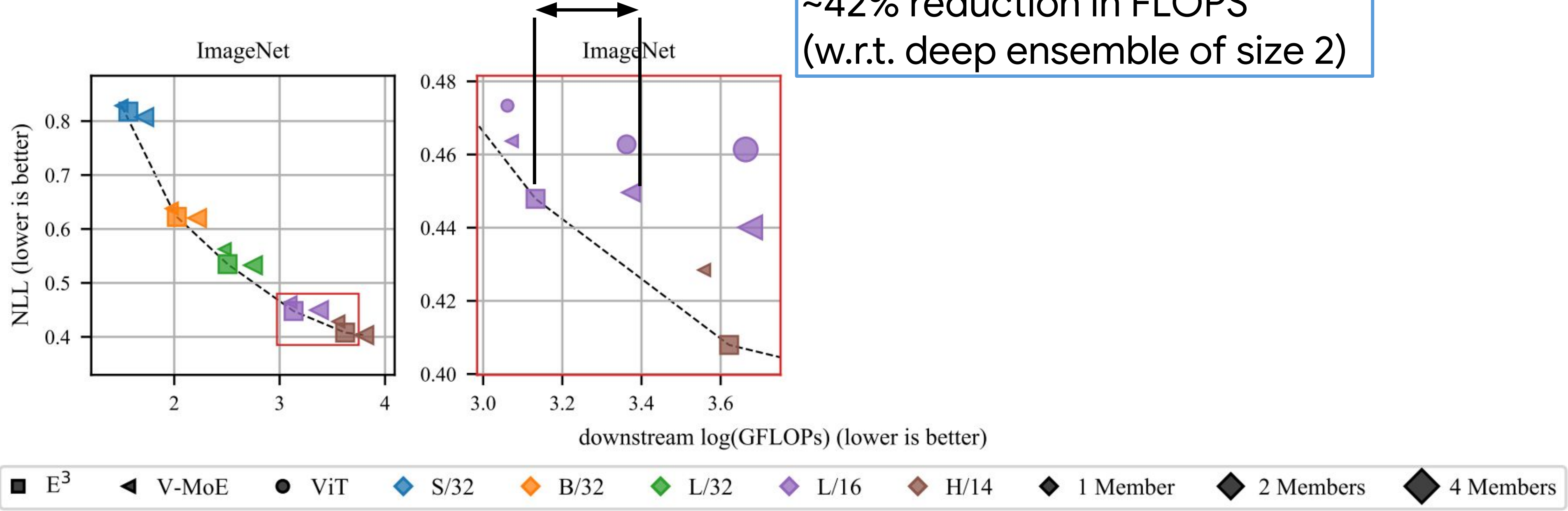
Evaluation of E^3 (ImageNet & few-shot)

Google AI
Brain Team



- E^3 tends to be on the frontier performance vs. FLOPs

Evaluation of E^3 (ImageNet & few-shot)

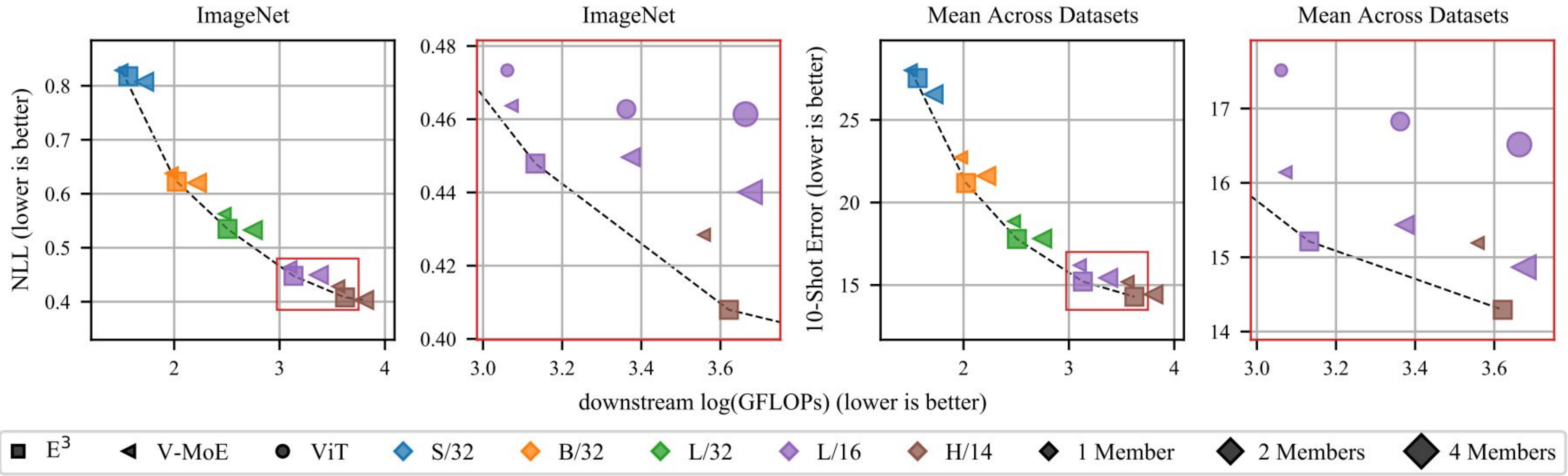


~42% reduction in FLOPS
(w.r.t. deep ensemble of size 2)

- E^3 tends to be on the frontier performance vs. FLOPs

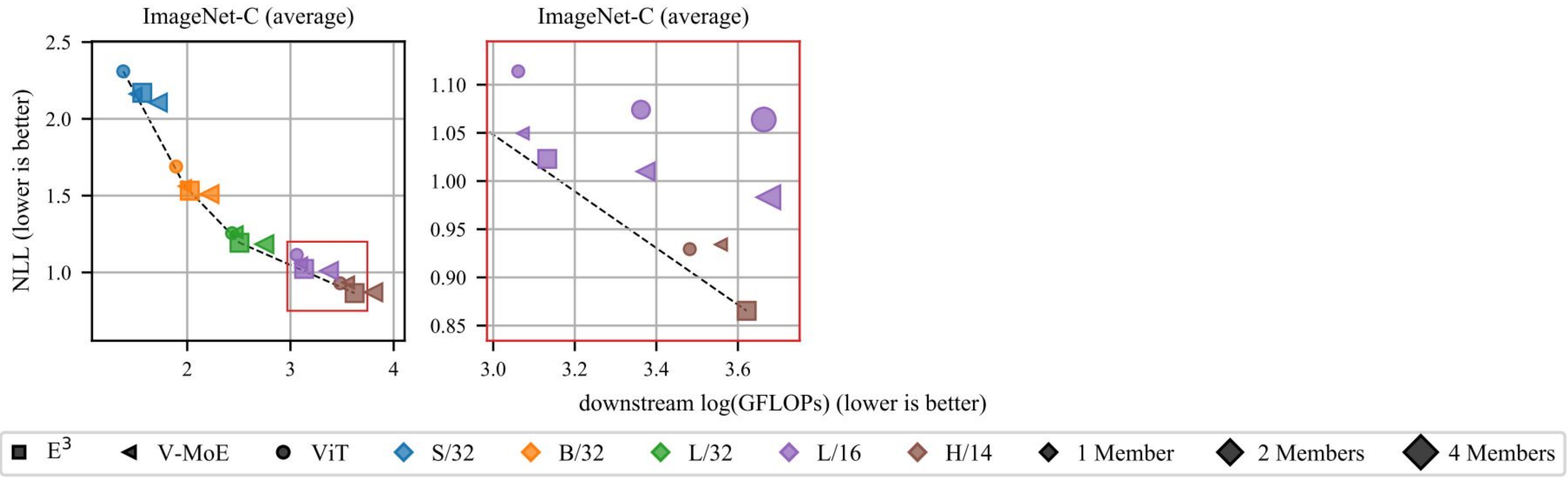
Evaluation of E^3 (ImageNet & few-shot)

Google AI Brain Team



- E^3 tends to be on the frontier performance vs. FLOPs
- E^3 works well on benchmark where either ensembles or sparse MoEs are known to perform well

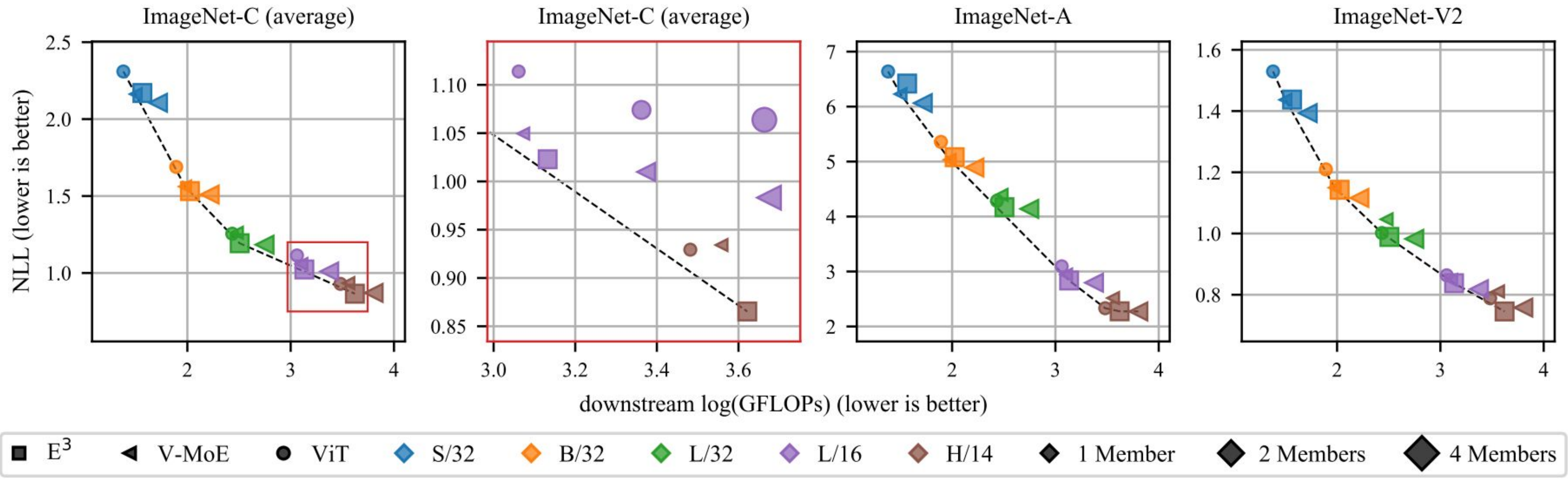
Evaluation of E^3 (ImageNet under dataset shifts)



- E^3 tends to be on the frontier performance vs. FLOPs

Evaluation of E^3 (ImageNet under dataset shifts)

Google AI
Brain Team



- E^3 tends to be on the frontier performance vs. FLOPs

What about other efficient ensemble approaches?

Google AI
Brain Team

Table : ImageNet performance (means ± standard errors over 8 seeds) of different efficient ensemble approaches based on a ViT-B/32 architecture.

		K	M	NLL ↓	ERROR ↓	ECE ↓	GFLOPs ↓
	ViT	–	–	0.688 ± 0.003	18.65 ± 0.08	0.022 ± 0.000	**78.0**
[Wen et al., 2019]	BE ViT	–	2	0.682 ± 0.003	18.47 ± 0.05	0.021 ± 0.000	97.1
	V-MoE	2	–	0.638 ± 0.001	**16.76** ± 0.05	0.033 ± 0.001	94.9
[Gal et al., 2015]	MC Dropout V-MoE	1	2	0.648 ± 0.002	17.10 ± 0.05	**0.019** ± 0.001	97.2
[Havasi et al., 2020]	MIMO V-MoE	2	2	0.636 ± 0.002	16.97 ± 0.04	0.028 ± 0.001	96.3
		2	4	0.672 ± 0.001	17.72 ± 0.04	0.037 ± 0.000	99.0
	E^3	1	2	**0.622** ± 0.001	**16.70** ± 0.03	**0.018** ± 0.000	105.9

What about other efficient ensemble approaches?

Google AI Brain Team

Table : ImageNet performance (means ± standard errors over 8 seeds) of different efficient ensemble approaches based on a ViT-B/32 architecture.

		K	M	NLL ↓	Error ↓	ECE ↓	KL ↑	GFLOPs ↓
	ViT	–	–	0.688 ± 0.003	18.65 ± 0.08	0.022 ± 0.000	–	**78.0**
[Wen et al., 2019]	BE ViT	–	2	0.682 ± 0.003	18.47 ± 0.05	0.021 ± 0.000	0.040 ± 0.001	97.1
	V-MoE	2	–	0.638 ± 0.001	**16.76** ± 0.05	0.033 ± 0.001	–	94.9
[Gal et al., 2015]	MC Dropout V-MoE	1	2	0.648 ± 0.002	17.10 ± 0.05	**0.019** ± 0.001	0.046 ± 0.000	97.2
[Havasi et al., 2020]	MIMO V-MoE	2	2	0.636 ± 0.002	16.97 ± 0.04	0.028 ± 0.001	0.000 ± 0.000	96.3
		2	4	0.672 ± 0.001	17.72 ± 0.04	0.037 ± 0.000	0.001 ± 0.000	99.0
	E^3	1	2	**0.622** ± 0.001	**16.70** ± 0.03	**0.018** ± 0.000	**0.217** ± 0.003	105.9

- E^3 tends to have diverse predictions

$$\frac{1}{M(M-1)} \sum_{m \neq m'}^{M} D_{\mathrm{KL}}(p_m(y|x) \| p_{m'}(y|x))$$

Some other results

In a nutshell:

- **Cifar10, Cifar10-C and Cifar100**: Even cleaner conclusions
- **OOD detection**:
  - In general, E^3 performs *worse* than V-MoE
  - Especially, “far” OOD detection task: Cifar* vs. {SVHN, Places365, DTD}
  - As the scale increases, E^3 performs better
- **Broader trend**: E^3 tends to perform better as the scale increases

Conclusions

- Sparse MoEs growing prevalence, e.g., in NLP [Patterson et al., 2021]
- Important, but also challenging, to study robustness at scale
- E^3 often on the "performance vs. FLOPs" frontier
- E^3 **simple** & **convenient**
  - Little code change
  - Can finetune standard checkpoints
- TMLR paper: https://openreview.net/pdf?id=i0ZM36d2qU
- Code: https://github.com/google-research/vmoe

Supplementary material

Detailed view of E^3

Google AI
Brain Team

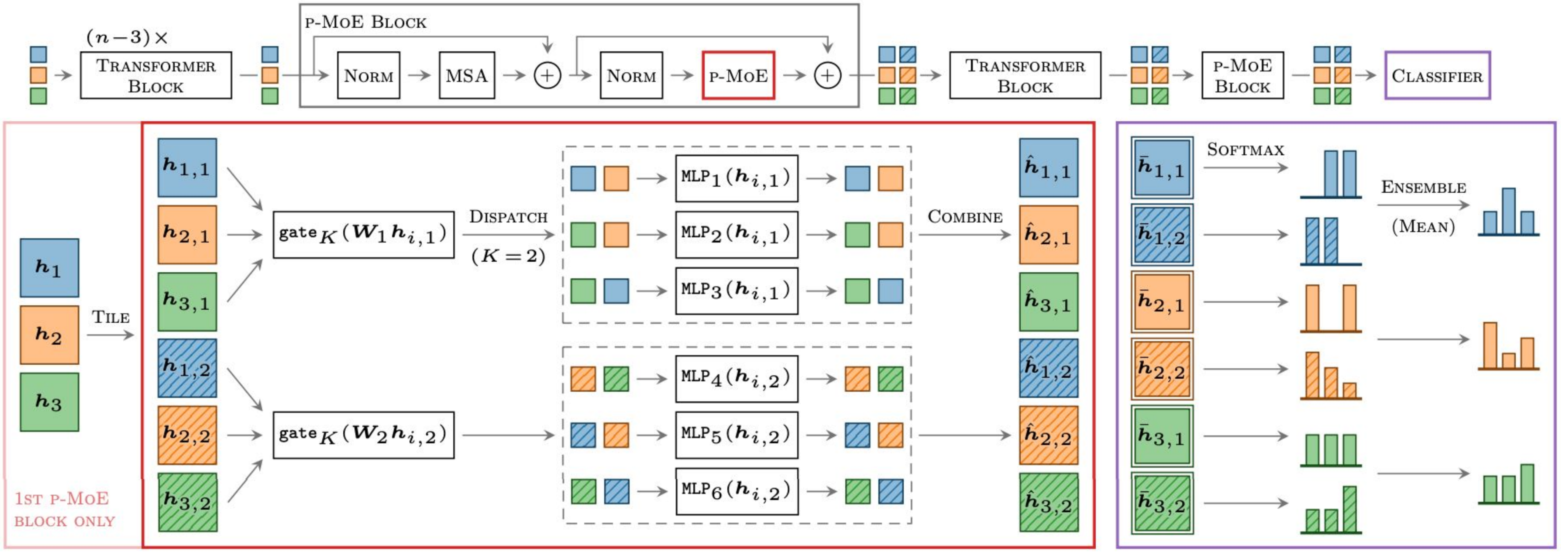
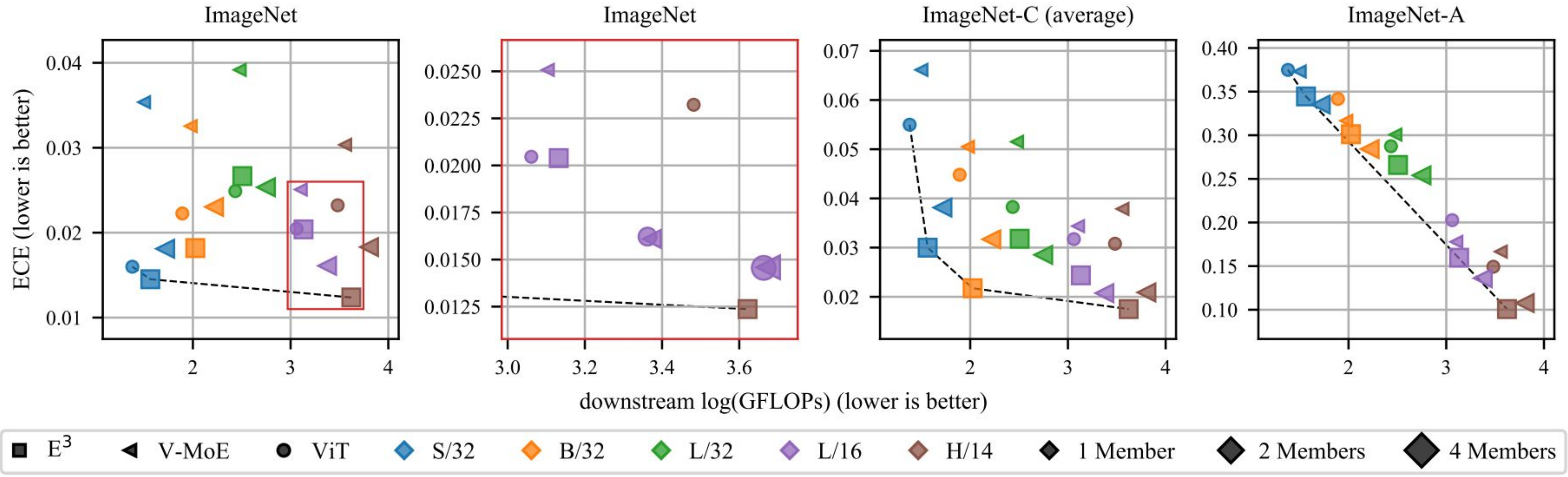


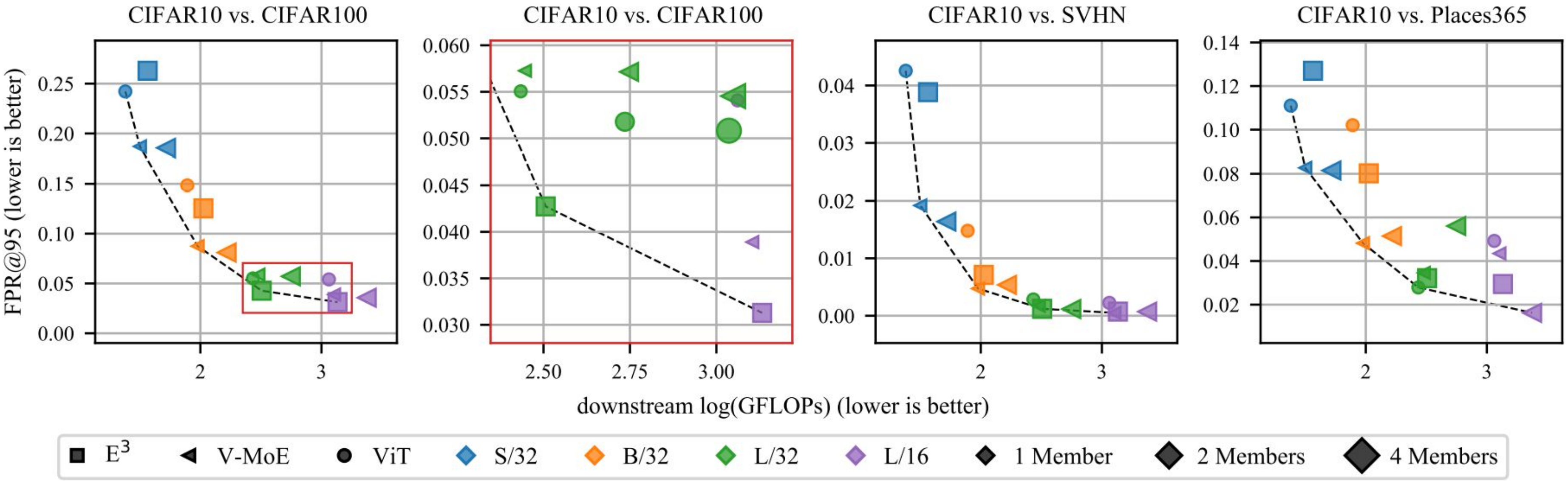
Figure 1: End-to-end overview of E^3 with $E = 6$ experts, partitioned into $M = 2$ groups, with sparsity of $K = 2$, and a "last-2" configuration. **Top**: E^3 contains a sequence of transformer blocks, followed by alternating transformer and p(artitioned)-MoE blocks. As in ViT, images are split into patches whose embeddings are processed by each block. Here, we show 1 embedding for each of three images (■, ■, ■). **Bottom left**: In a p-MoE block, we replace the transformer block's MLP with parallel partitioned expert MLPs, see (2). The effect of the routing weights is not depicted. Embeddings are tiled (▨) in the first p-MoE block only. **Bottom right**: The classifier averages predictions from the final tiled representations (▣).

Evaluation of E^3: ECE



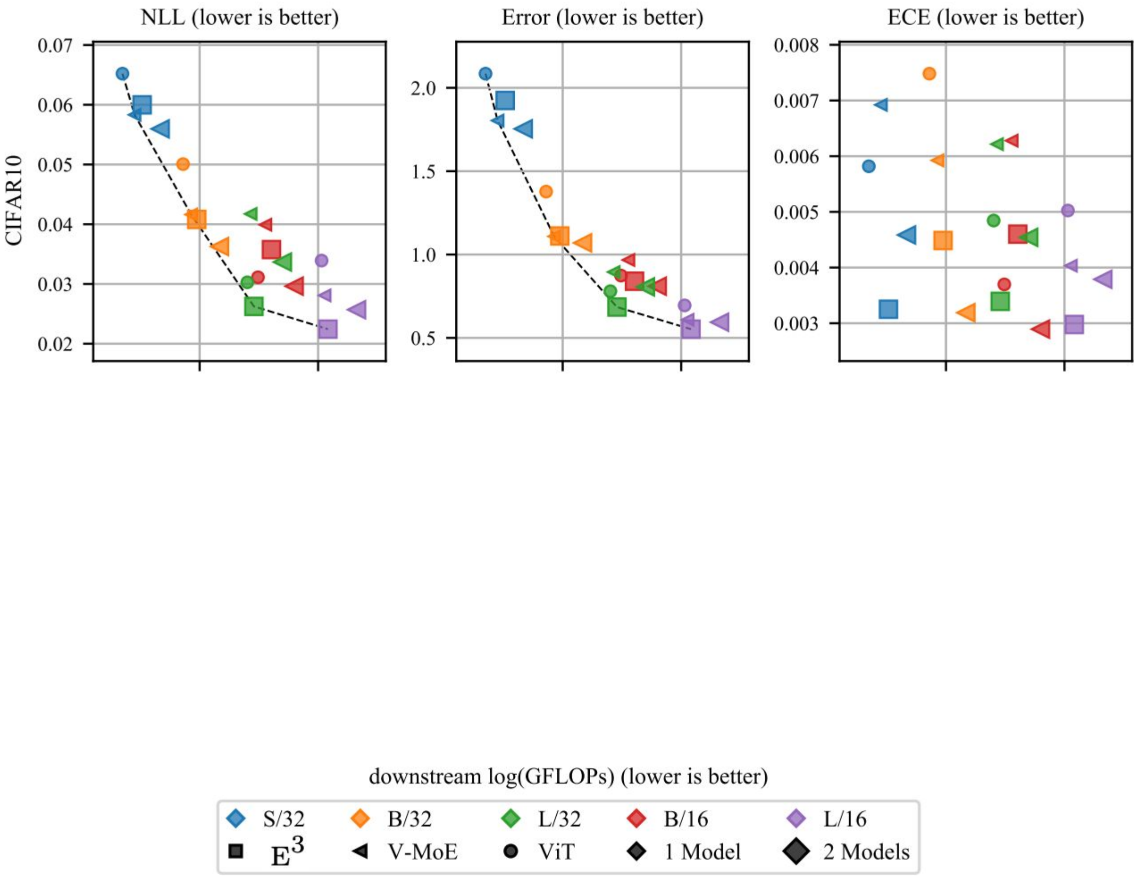
- ECE is not consistent for different ViT/V-MoE families
- E^3 improves ECE over ViT and V-MoE (& V-MoE provides poor ECE)

Evaluation of E^3: ODD detection



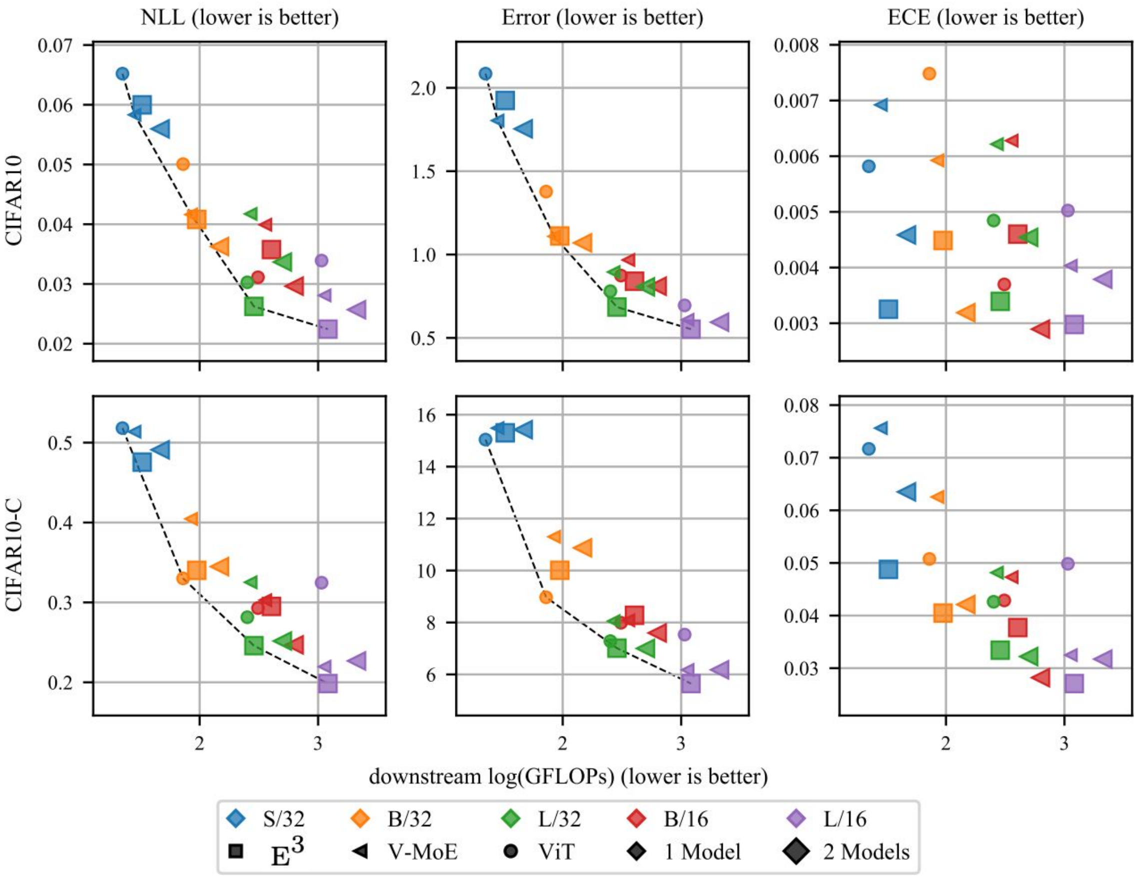
- E^3 does not provide consistent OOD detection performance
- Improvement at larger scales

Evaluation of E^3: Cifar10 & Cifar10-C



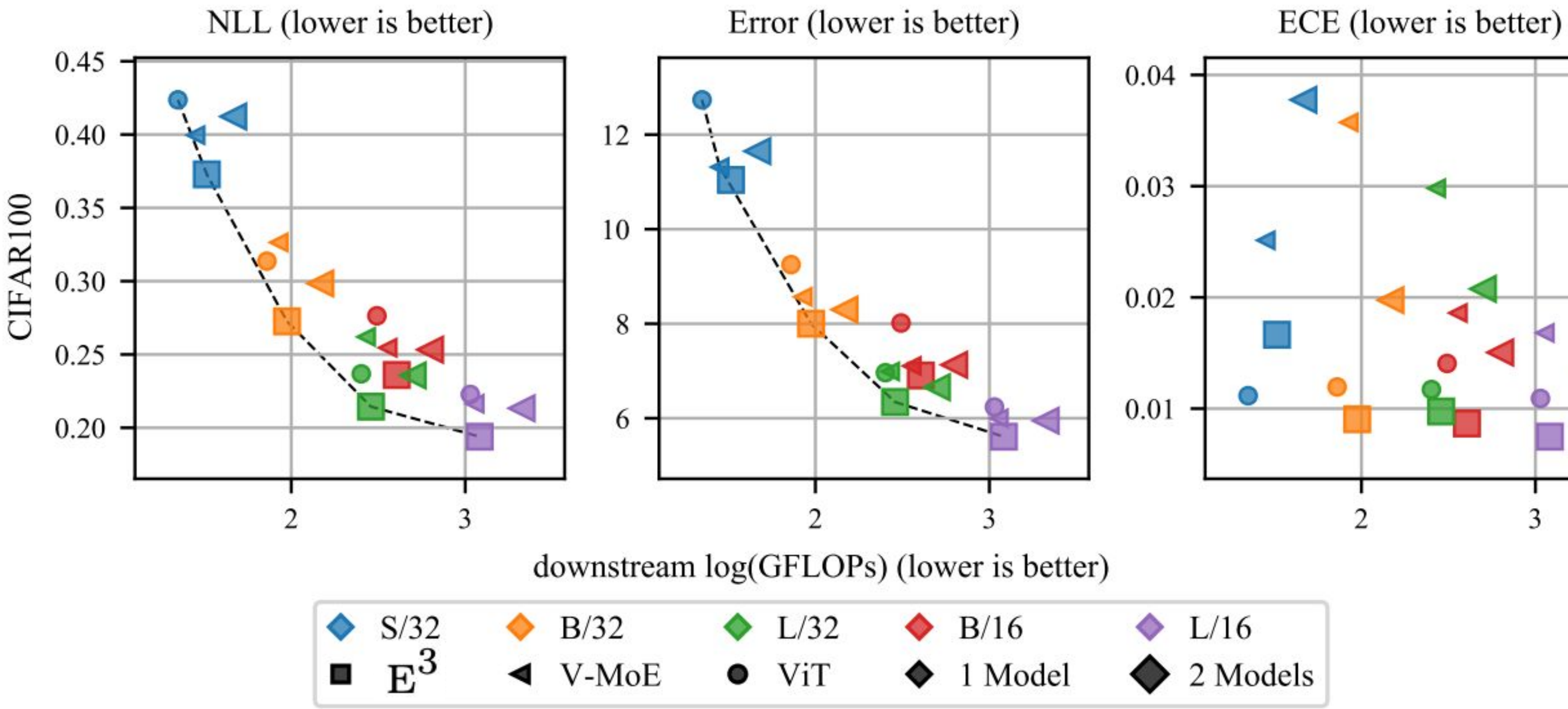
Evaluation of E^3: Cifar10 & Cifar10-C

Google AI Brain Team



Evaluation of E^3: Cifar100

Google AI
Brain Team



Ablation E^3: Tiling and partitioning

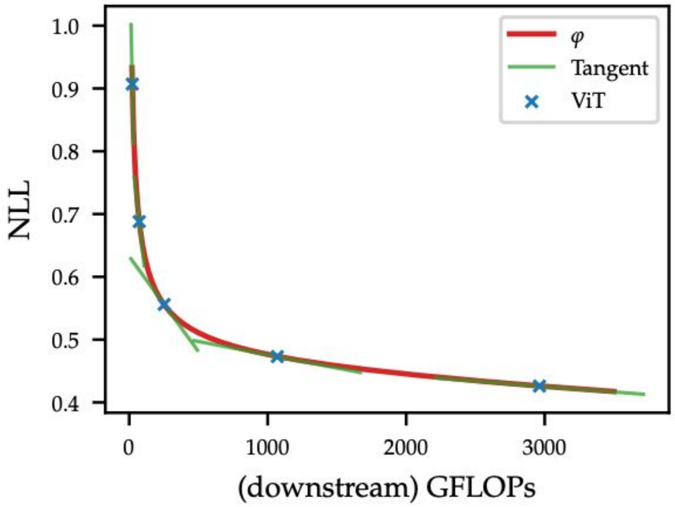
Google AI Brain Team

Table 2: ImageNet performance (means ± standard errors over 8 seeds) of E^3-B/32 ($K = M = 2$), V-MoE ($K = 4$), and two ablations: *only* tiling and *only* partitioning. The noise in $\texttt{gate}_K$ is denoted by σ.

	NLL ↓	Error ↓	ECE ↓	KL ↑
V-MoE	0.636 ± 0.001	16.70 ± 0.04	0.034 ± 0.001	—
E^3	**0.612** ± 0.001	**16.49** ± 0.02	**0.013** ± 0.000	**0.198** ± 0.003
Tiling	0.637 ± 0.002	16.74 ± 0.06	0.028 ± 0.001	0.000 ± 0.000
Tiling ($\sigma \times 2$)	0.638 ± 0.001	16.72 ± 0.03	0.033 ± 0.001	0.001 ± 0.000
Tiling ($\sigma \times 4$)	0.638 ± 0.001	16.74 ± 0.03	0.033 ± 0.001	0.002 ± 0.000
Partitioning	0.640 ± 0.001	16.72 ± 0.05	0.034 ± 0.001	—

E^3 performs better as the scale increases

Google AI
Brain Team



$$\text{Normalised improvement}(v) = \text{improvement}(v) \times \frac{\varphi'(\text{FLOPs}_{\text{ H/14}})}{\varphi'(\text{FLOPs}_v)}$$

		S/32	B/32	L/32	L/16	H/14
Normalised	E^3 vs. ViT	0.02%	0.09%	0.24%	2.35%	4.27%
	V-MoE vs. ViT	0.01%	0.06%	-0.04%	0.89%	0.02%
Not normalised	E^3 vs. ViT	9.82%	9.53%	3.76%	5.38%	4.27%
	V-MoE vs. ViT	7.98%	6.62%	-0.60%	2.05%	0.02%