

THEORY ON MIXTURE-OF-EXPERTS IN CONTINUAL LEARNING

Presenter: Hongbo Li, Postdoc Research Fellow

Engineering Systems and Design Pillar
Singapore University of Technology and Design

December 23, 2024

PUBLICATIONS

1. **Hongbo Li**, Sen Lin, Lingjie Duan, Yingbin Liang, and Ness B. Shroff, "*Theory on Mixture-of-Experts in Continual Learning*", submitted to ICLR 2025, [Online Available] <https://arxiv.org/abs/2406.16437>.
2. **Hongbo Li**, and Lingjie Duan, "*Theory on Mixture-of-Experts in Mobile Edge Computing*", accepted by INFOCOM 2025.

Part I

MOE IN CONTINUAL LEARNING

MOTIVATIONS

- ▶ Continual Learning (CL) has emerged as an important paradigm in machine learning, in which an expert aims to **learn a sequence of tasks one by one** over time.
- ▶ Given the dynamic nature of CL, one major challenge herein is known as **catastrophic forgetting**, where **a single expert** can perform poorly on (i.e., easily forget) the previous tasks when learning new tasks if data distributions change largely across tasks.

LITERATURE REVIEW: CL

Various **empirical approaches** have been proposed to tackle catastrophic forgetting in CL:

- ▶ Regularization-based approaches (e.g., Kirkpatrick et al. 2017; Gou et al. 2021).
- ▶ Parameter-isolation-based approaches (e.g., Chaudhry et al. 2018; Konishi et al. 2023).
- ▶ Memory-based approaches (e.g., Jin et al. 2021; S. Lin, Yang, et al. 2021; R. Gao and Liu 2023).

On the other hand, **theoretical studies** on CL are very limited.

LITERATURE REVIEW: MOE MODEL

- ▶ Mixture-of-Experts (MoE) has found widespread applications in emerging fields such as **large language models (LLMs)** (e.g., Du et al. 2022; Li et al. 2024; B. Lin et al. 2024).
- ▶ Chen et al. (2022) theoretically analyze the mechanism of MoE in deep learning under the setup of a mixture of classification problem. However, this study focuses on a **single-task setting**, and hence does not analyze the dynamics of CL.

LITERATURE REVIEW: MOE IN CL

- ▶ Recently, the MoE model has been applied to reducing catastrophic forgetting in CL (Hihn and Braun 2021; L. Wang et al. 2022; Doan, Mirzadeh, and Farajtabar 2023; Rypeść et al. 2023; J. Yu et al. 2024).
- ▶ However, these works solely focus on empirical methods, **lacking theoretical analysis** of how the MoE performs in CL.

CL IN LINEAR MODEL

We consider the CL setting with T training rounds.

- ▶ In each round $t \in [T]$, one out of N tasks **randomly arrives** to be learned by the MoE model with M experts.
- ▶ For each task, we consider fitting a **linear model** $f(\mathbf{X}) = \mathbf{X}^\top \mathbf{w}$ with ground truth $\mathbf{w} \in \mathbb{R}^d$.
- ▶ Then for the t -th task arrival, let $\mathcal{D}_t = (\mathbf{X}_t, \mathbf{y}_t)$ denote its dataset, where $\mathbf{X}_t \in \mathbb{R}^{d \times s_t}$ is the feature matrix, and $\mathbf{y}_t \in \mathbb{R}^{s_t}$ is the output vector.
- ▶ In this study, we focus on the **overparameterized regime**, where $s_t < d$.

CL IN LINEAR MODEL (CONT.)

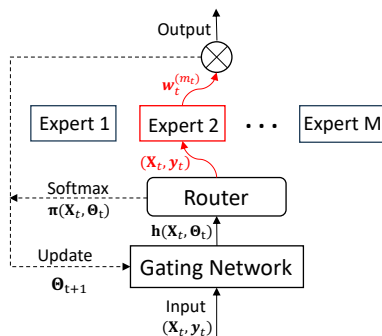
- ▶ Let $\mathcal{W} = \{w_1, \dots, w_N\}$ represent the **collection of ground truth vectors** of all N tasks.
- ▶ For any two tasks $n, n' \in [N]$, we assume $\|w_n - w_{n'}\|_\infty = \mathcal{O}(\sigma_0)$, where $\sigma_0 \in (0, 1)$ denotes the variance.
- ▶ We assume that task n possesses a unique **feature signal** $v_n \in \mathbb{R}^d$ with $\|v_n\|_\infty = \mathcal{O}(1)$.
- ▶ In each round $t \in [T]$, let $n_t \in [N]$ denote the index of the current task arrival with ground truth $w_{n_t} \in \mathcal{W}$.

CL IN LINEAR MODEL (CONT.)

At the beginning of each training round $t \in [T]$, the dataset $\mathcal{D}_t = (\mathbf{X}_t, \mathbf{y}_t)$ of the new task arrival n_t is generated by the following steps:

1. **Uniformly draw a ground truth \mathbf{w}_n** from ground-truth pool \mathcal{W} and let $\mathbf{w}_{n_t} = \mathbf{w}_n$.
2. Independently generate a random variable $\beta_t \in (0, C]$, where C is a constant satisfying $C = \mathcal{O}(1)$.
3. Generate \mathbf{X}_t as a collection of s_t samples, where **one sample is given by $\beta_t \mathbf{v}_{n_t}$** and the rest of the $s_t - 1$ samples are drawn from normal distribution $\mathcal{N}(\mathbf{0}, \sigma_t^2 \mathbf{I}_d)$, where $\sigma_t \geq 0$ is the noise level.
4. Generate the output to be $\mathbf{y}_t = \mathbf{X}_t^\top \mathbf{w}_{n_t}$.

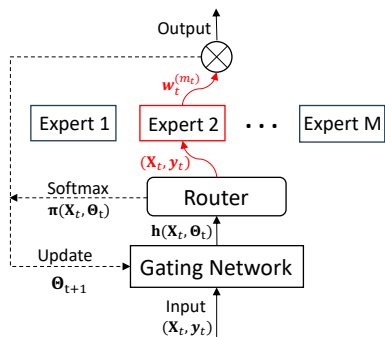
STRUCTURE OF THE MOE MODEL



- ▶ Upon the arrival of task n_t and input of its data $\mathcal{D}_t = (\mathbf{X}_t, y_t)$, the gating network computes its **linear output** $h_m(\mathbf{X}_t, \theta_t^{(m)})$ for each expert $m \in [M]$.
- ▶ Define $\mathbf{h}(\mathbf{X}_t, \Theta_t) := [h_1(\mathbf{X}_t, \theta_t^{(1)}) \cdots h_M(\mathbf{X}_t, \theta_t^{(M)})]$ and $\Theta_t := [\theta_t^{(1)} \cdots \theta_t^{(M)}]$ as the outputs and the parameters of the gating network for all experts, respectively. We obtain

$$\mathbf{h}(\mathbf{X}_t, \Theta_t) = \sum_{i \in [s_t]} \Theta_t^\top \mathbf{X}_{t,i}$$

STRUCTURE OF THE MOE MODEL (CONT.)



- In each round t , for task n_t , the router selects the expert with the **maximum gate output** $h_m(\mathbf{X}_t, \boldsymbol{\theta}_t^{(m)})$, denoted as m_t , from the M experts:

$$m_t = \arg \max_m \{h_m(\mathbf{X}_t, \boldsymbol{\theta}_t^{(m)}) + r_t^{(m)}\},$$

where $r_t^{(m)}$ for any $m \in [M]$ is drawn independently from the uniform distribution $\text{Unif}[0, \lambda]$.

- Additionally, the router calculates the **softmaxed gate outputs**, derived by

$$\pi_m(\mathbf{X}_t, \boldsymbol{\theta}_t) = \frac{\exp(h_m(\mathbf{X}_t, \boldsymbol{\theta}_t^{(m)}))}{\sum_{m'=1}^M \exp(h_{m'}(\mathbf{X}_t, \boldsymbol{\theta}_t^{(m')}))}, \quad \forall m \in [M]$$

for updating $\boldsymbol{\theta}_{t+1}$.

TRAINING OF THE EXPERT MODEL

- ▶ Let $\mathbf{w}_t^{(m)}$ denote the model of expert m in the t -th training round, where each model is initialized from zero.
- ▶ In each round t , the **training loss** is defined by the mean-squared error (MSE) relative to \mathcal{D}_t :

$$\mathcal{L}_t^{tr}(\mathbf{w}_t^{(m_i)}, \mathcal{D}_t) = \frac{1}{s_t} \|(\mathbf{X}_t)^\top \mathbf{w}_t^{(m_i)} - \mathbf{y}_t\|_2^2.$$

TRAINING OF THE EXPERT MODEL (CONT.)

- ▶ Gradient descent (GD) provides a **unique solution** for minimizing $\mathcal{L}_t^{tr}(\mathbf{w}_t^{(m_t)}, \mathcal{D}_t)$, which is determined by the following optimization problem (Evron et al. 2022; S. Lin, Ju, et al. 2023):

$$\min_{\mathbf{w}_t} \|\mathbf{w}_t - \mathbf{w}_{t-1}^{(m_t)}\|_2, \quad \text{s.t. } \mathbf{X}_t^\top \mathbf{w}_t = \mathbf{y}_t.$$

- ▶ Solving this problem, we update the selected expert m_t for the current task arrival n_t as follows:

$$\mathbf{w}_t^{(m_t)} = \mathbf{w}_{t-1}^{(m_t)} + \mathbf{X}_t (\mathbf{X}_t^\top \mathbf{X}_t)^{-1} (\mathbf{y}_t - \mathbf{X}_t^\top \mathbf{w}_{t-1}^{(m_t)}).$$

- ▶ For any other expert $m \in [M]$ not selected (i.e., $m \neq m_t$), its model $\mathbf{w}_t^{(m)}$ **remains unchanged** from $\mathbf{w}_{t-1}^{(m)}$.

TRAINING OF GATING NETWORK PARAMETERS

After obtaining $w_t^{(m)}$, the MoE updates Θ_t to Θ_{t+1} using GD.

- ▶ On one hand, we aim for $\theta_{t+1}^{(m)}$ of each expert m to **specialize in a specific task**, which helps mitigate learning loss caused by the incorrect routing of distinct tasks.
- ▶ On the other hand, the router needs to **balance the load** among all experts (Fedus, Zoph, and Shazeer 2022; Li et al. 2024) to reduce the risk of model overfitting and enhance the learning performance in CL.

KEY DESIGN I: MULTI-OBJECTIVE TRAINING LOSS

- ▶ First, we propose the following **locality loss function** for updating Θ_t :

$$\mathcal{L}_t^{loc}(\Theta_t, \mathcal{D}_t) = \sum_{m \in [M]} \pi_m(\mathbf{X}_t, \Theta_t) \|\mathbf{w}_t^{(m)} - \mathbf{w}_{t-1}^{(m)}\|_2.$$

- ▶ Then we follow the existing MoE literature (e.g., Fedus, Zoph, and Shazeer 2022; Li et al. 2024) to define an **auxiliary loss** to characterize load balance among the experts:

$$\mathcal{L}_t^{aux}(\Theta_t, \mathcal{D}_t) = \alpha \cdot M \cdot \sum_{m \in [M]} f_t^{(m)} \cdot P_t^{(m)},$$

where α is constant, $f_t^{(m)} = \frac{1}{t} \sum_{\tau=1}^t \mathbb{1}\{m_\tau = m\}$ is the **fraction of tasks** dispatched to expert m since $t = 1$, and $P_t^{(m)} = \frac{1}{t} \sum_{\tau=1}^t \pi_m(\mathbf{X}_\tau, \Theta_\tau) \cdot \mathbb{1}\{m_\tau = m\}$ is the **average probability** that the router chooses expert m since $t = 1$.

KEY DESIGN I: MULTI-OBJECTIVE TRAINING LOSS (CONT.)

We finally define the **task loss** for each task arrival n_t as follows:

$$\mathcal{L}_t^{task}(\Theta_t, \mathbf{w}_t^{(m_i)}, \mathcal{D}_t) = \mathcal{L}_t^{tr}(\mathbf{w}_t^{(m_i)}, \mathcal{D}_t) + \mathcal{L}_t^{loc}(\Theta_t, \mathcal{D}_t) + \mathcal{L}_t^{aux}(\Theta_t, \mathcal{D}_t).$$

Commencing from the initialization Θ_0 , the gating network is updated based on GD:

$$\theta_{t+1}^{(m)} = \theta_t^{(m)} - \eta \cdot \nabla_{\theta_t^{(m)}} \mathcal{L}_t^{task}(\Theta_t, \mathbf{w}_t^{(m_i)}, \mathcal{D}_t), \forall m \in [M]$$

where $\eta > 0$ is the learning rate.

KEY DESIGN II: EARLY TERMINATION

Algorithm Training of the MoE model for CL

- 1: **Input:** $T, \sigma_0, \Gamma = \mathcal{O}(\sigma_0^{1.25}), \lambda = \Theta(\sigma_0^{1.25}), I^{(m)} = 0, \alpha = \mathcal{O}(\sigma_0^{0.5}), \eta = \mathcal{O}(\sigma_0^{0.5}), T_1 = \lceil \eta^{-1}M \rceil$;
 - 2: Initialize $\theta_0^{(m)} = \mathbf{0}$ and $w_0^{(m)} = \mathbf{0}, \forall m \in [M]$;
 - 3: **for** $t = 1, \dots, T$ **do**
 - 4: Generate $r_t^{(m)}$ for any $m \in [M]$;
 - 5: Select m_t and update $w_t^{(m_t)}$;
 - 6: **if** $t > T_1$ **then**
 - 7: **for** $\forall m \in [M]$ with $|h_m - h_{m_t}| < \Gamma$ **do**
 - 8: $I^{(m)} = 1$; // **Convergence flag**
 - 9: **end for**
 - 10: **end if**
 - 11: **if** $\exists m$, s.t. $I^{(m)} = 0$ **then**
 - 12: Update $\theta_t^{(m)}$ for any $m \in [M]$;
 - 13: **end if**
 - 14: **end for**
-

THEORETICAL RESULTS: FEATURE SIGNAL

Lemma 1 ($M > N$ version)

For any two feature matrices \mathbf{X} and $\tilde{\mathbf{X}}$ with *the same feature signal* \mathbf{v}_n , with probability at least $1 - o(1)$, their corresponding gate outputs of the same expert m satisfy

$$|h_m(\mathbf{X}, \boldsymbol{\theta}_t^{(m)}) - h_m(\tilde{\mathbf{X}}, \boldsymbol{\theta}_t^{(m)})| = \mathcal{O}(\sigma_0^{1.5}).$$

Given N tasks, all experts can be *classified into N sets based on their specialty*, where each expert set is defined as:

$$\mathcal{M}_n = \{m \in [M] \mid n = \arg \max_{j \in [N]} (\boldsymbol{\theta}_t^{(m)})^\top \mathbf{v}_j\}.$$

THEORETICAL RESULTS: CONVERGENCE OF EXPERT MODEL

Proposition 1 ($M > N$ version)

Under Algorithm 1, with probability at least $1 - o(1)$, for any $t > T_1$, where $T_1 = \lceil \eta^{-1}M \rceil$, each expert $m \in [M]$ stabilizes within an expert set \mathcal{M}_n , and its expert model remains unchanged beyond time T_1 , satisfying $\mathbf{w}_{T_1+1}^{(m)} = \dots = \mathbf{w}_T^{(m)}$.

NECESSITY OF EARLY TERMINATION

Proposition 2 ($M > N$ version)

If the MoE keeps updating Θ_t at any round $t > T_1$, we obtain:

1. At round $t_1 = \lceil \eta^{-1} \sigma_0^{-0.25} M \rceil$, the following property holds

$$|h_m(\mathbf{X}_{t_1}, \boldsymbol{\theta}_{t_1}^{(m)}) - h_{m'}(\mathbf{X}_{t_1}, \boldsymbol{\theta}_{t_1}^{(m')})| = \begin{cases} \mathcal{O}(\sigma_0^{1.75}), & \text{if } m, m' \in \mathcal{M}_n, \\ \Theta(\sigma_0^{0.75}), & \text{otherwise.} \end{cases}$$

2. At round $t_2 = \lceil \eta^{-1} \sigma_0^{-0.75} M \rceil$, the following property holds

$$|h_m(\mathbf{X}_{t_2}, \boldsymbol{\theta}_{t_2}^{(m)}) - h_{m'}(\mathbf{X}_{t_2}, \boldsymbol{\theta}_{t_2}^{(m')})| = \mathcal{O}(\sigma_0^{1.75}), \forall m, m' \in [M].$$

BENEFIT OF EARLY TERMINATION

Proposition 3 ($M > N$ version)

Under Algorithm 1, the MoE terminates updating Θ_t since round $T_2 = \mathcal{O}(\eta^{-1}\sigma_0^{-0.25}M)$. Then for any task arrival n_t at $t > T_2$, the router *selects any expert $m \in \mathcal{M}_{n_t}$ with an identical probability of $\frac{1}{|\mathcal{M}_{n_t}|}$* , where $|\mathcal{M}_{n_t}|$ is the number of experts in set \mathcal{M}_n .

DEFINITION OF FORGETTING AND GENERALIZATION

We define $\mathcal{E}_t(\mathbf{w}_t^{(m_t)})$ as the model error in the t -th round:

$$\mathcal{E}_t(\mathbf{w}_t^{(m_t)}) = \|\mathbf{w}_t^{(m_t)} - \mathbf{w}_{n_t}\|_2^2.$$

Following the existing literature on CL (e.g., S. Lin, Ju, et al. 2023; Chaudhry et al. 2018), we assess the performance of MoE in CL using the metrics of **forgetting** and **overall generalization error**:

► Forgetting:

$$F_t = \frac{1}{t-1} \sum_{\tau=1}^{t-1} (\mathcal{E}_\tau(\mathbf{w}_t^{(m_\tau)}) - \mathcal{E}_\tau(\mathbf{w}_\tau^{(m_\tau)})).$$

► Overall generalization error:

$$G_T = \frac{1}{T} \sum_{\tau=1}^T \mathcal{E}_\tau(\mathbf{w}_T^{(m_\tau)}).$$

BENCHMARK: PERFORMANCE OF SINGE EXPERT

Here we define $r := 1 - \frac{\delta}{d}$ as the overparameterization ratio.

Proposition 4

If $M = 1$, for any training round $t \in \{2, \dots, T\}$, we have

$$\mathbb{E}[F_t] = \frac{1}{t-1} \sum_{\tau=1}^{t-1} \left\{ \frac{r^t - r^\tau}{N} \sum_{n=1}^N \|\mathbf{w}_n\|^2 + \frac{r^\tau - r^t}{N^2} \sum_{n \neq n'} \|\mathbf{w}_{n'} - \mathbf{w}_n\|^2 \right\},$$
$$\mathbb{E}[G_T] = \frac{r^T}{N} \sum_{n=1}^N \|\mathbf{w}_n\|^2 + \frac{1 - r^T}{N^2} \sum_{n \neq n'} \|\mathbf{w}_n - \mathbf{w}'_n\|^2.$$

PERFORMANCE OF MOE

We define $L_t^{(m)} := t \cdot f_t^{(m)}$ as the cumulative number of task arrivals routed to expert m up to round t .

Theorem 1 ($M > N$ Case)

If $M = \Omega(N \ln(N))$, for each round $t \in \{2, \dots, T_1\}$, the expected forgetting satisfies

$$\mathbb{E}[F_t] < \frac{1}{t-1} \sum_{\tau=1}^{t-1} \left\{ \frac{r^{L_t^{(m_\tau)}} - r^{L_\tau^{(m_\tau)}}}{N} \sum_{n=1}^N \|\mathbf{w}_n\|^2 + \frac{r^{L_\tau^{(m_\tau)}} - r^{L_t^{(m_\tau)}}}{N^2} \sum_{n \neq n'} \|\mathbf{w}_{n'} - \mathbf{w}_n\|^2 \right\}.$$

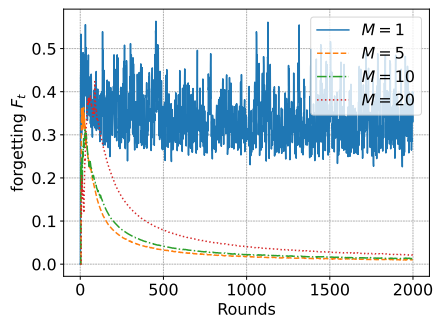
For each $t \in \{T_1 + 1, \dots, T\}$, we have $\mathbb{E}[F_t] = \frac{T_1-1}{t-1} \mathbb{E}[F_{T_1}]$. Further, after training task n_T in the last round T , the overall generalization error satisfies

$$\mathbb{E}[G_T] < \frac{1}{T} \sum_{\tau=1}^T \left\{ \frac{r^{L_{T_1}^{(m_\tau)}}}{N} \sum_{n=1}^N \|\mathbf{w}_n\|^2 + \frac{1 - r^{L_{T_1}^{(m_\tau)}}}{N^2} \sum_{n \neq n'} \|\mathbf{w}_{n'} - \mathbf{w}_n\|^2 \right\}.$$

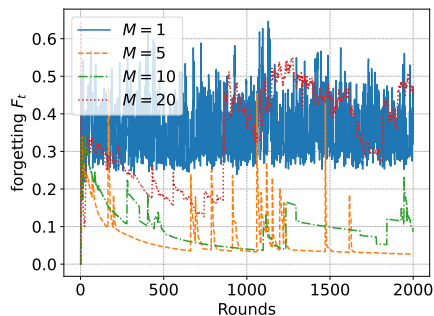
EXPERIMENT SETTING: SYNTHETIC DATA

- ▶ We first generate N ground truths and their corresponding feature signals.
 - For each $\mathbf{w}_n \in \mathbb{R}^d$, we randomly generate d elements by a normal distribution $\mathcal{N}(0, \sigma_0)$. These ground truths are then scaled by a constant to obtain their feature signals \mathbf{v}_n .
- ▶ In each training round t , we generate $(\mathbf{X}_t, \mathbf{y}_t)$ based on ground-truth pool \mathcal{W} and feature signals.
 - After drawing \mathbf{w}_{n_t} from \mathcal{W} , for $\mathbf{X}_t \in \mathbb{R}^{d \times s}$, we randomly select one out of s samples to fill with $\beta_t \mathbf{v}_{n_t}$. The other $s - 1$ samples are generated from $\mathcal{N}(\mathbf{0}, \sigma_t^2 \mathbf{I}_d)$.
 - Then, we compute the output $\mathbf{y}_t = \mathbf{X}_t^\top \mathbf{w}_{n_t}$.
- ▶ Here we set $\sigma_0 = 0.4, \sigma_t = 0.1, d = 10, s = 6, \eta = 0.5, \alpha = 0.5$ and $\lambda = 0.3$.

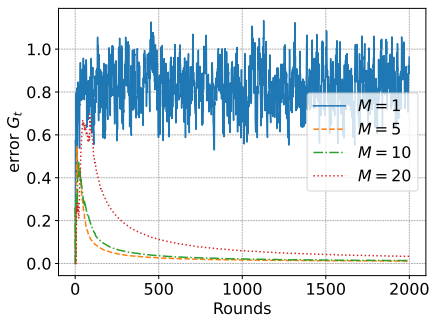
EXPERIMENTS: SYNTHETIC DATA



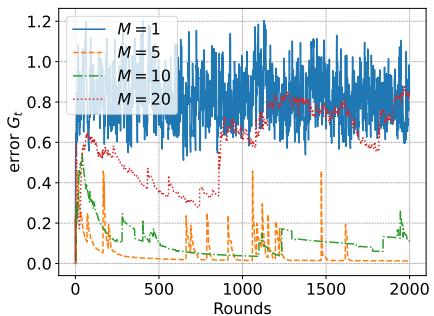
(a) With termination.



(b) Without termination.



(c) With termination.



(d) Without termination.

Figure. The dynamics of forgetting and overall generalization errors with and without termination of updating Θ_t in Algorithm 1. Here we set $N = 6$ with $K = 3$ clusters and vary $M \in \{1, 5, 10, 20\}$.

EXPERIMENTS: REAL-DATA VALIDATION

- ▶ In each round, we obtain the feature matrix by averaging $s = 100$ training data samples.
- ▶ To diversify the model gaps of different tasks, we **transform the $d \times d$ matrix into a $d \times d$ dimensional normalized vector** to serve as input for the gating network.
- ▶ Then we calculate the **variance σ_0** of each element across all tasks from the input vector.

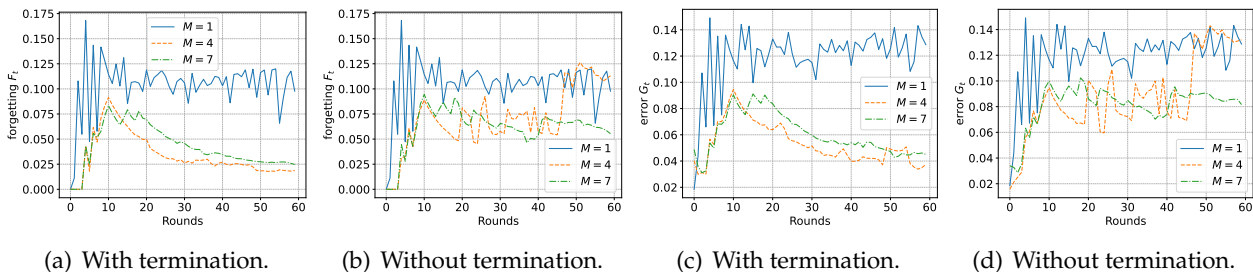


Figure. Learning performance under MNIST datasets (LeCun et al. 1989). Here we set $N = 3$ and $M \in \{1, 4, 7\}$.

CONCLUSION

- ▶ We conducted **the first theoretical analysis** of MoE and its impact on learning performance in CL, focusing on an overparameterized linear regression problem.
- ▶ We proved that the MoE model can diversify experts to specialize in different tasks, while its router can learn to select the right expert for each task and balance the loads across all experts.
- ▶ Then we demonstrated that, under CL, terminating the updating of gating network parameters after sufficient training rounds is necessary for system convergence.
- ▶ Furthermore, we provided **explicit forms of the expected forgetting and overall generalization error** to assess the impact of MoE.
- ▶ Finally, we conducted experiments on real datasets using DNNs to show that certain insights can extend beyond linear models.

Part II

MOE IN MOBILE EDGE COMPUTING (MEC)

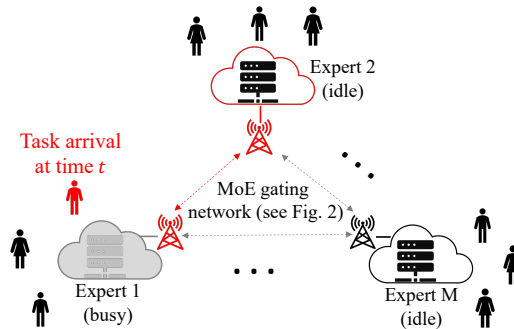
MOTIVATIONS

- ▶ In mobile edge computing (MEC) networks, mobile users generate diverse machine learning tasks dynamically over time.
- ▶ These tasks are typically offloaded to the nearest available edge server, by considering **communication and computational efficiency**.
- ▶ However, its operation leads to severe **overfitting or catastrophic forgetting** of previous tasks.
- ▶ Therefore, it is natural and promising to apply the MoE model in MEC networks.

LITERATURE REVIEW: MOE IN MEC

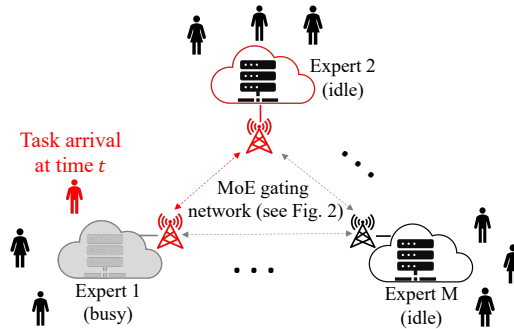
- ▶ Rajbhandari et al. (2022) focuses on minimizing network delay by leveraging high-bandwidth connections to allocate experts efficiently.
- ▶ Singh et al. (2023) optimizes communication to enhance the routing strategy of the gating network and eliminate unnecessary data movement.
- ▶ Wang et al. (2024) exploits the MEC structure to support MoE-based generative AI by optimally scheduling tasks to experts with varying computational resource limitations.
- ▶ However, there is a **lack of theoretical understanding** of MoE and its design to guarantee the convergence of continual learning and generalization errors.

SYSTEM MODEL



- ▶ An MEC network operator manages a set $\mathbb{M} = \{1, \dots, M\}$ of MEC servers/cloudlets as experts.
- ▶ We consider a discrete time horizon $\mathbb{T} = \{1, \dots, T\}$.
- ▶ At the beginning of each time t , a mobile user randomly arrives to request task offloading to solve a machine learning problem from its nearest BS.

SYSTEM MODEL: CL PROCESS



1. The current user needs to upload its data, denoted by $\mathcal{D}_t = (\mathbf{X}_t, \mathbf{y}_t)$, to the BS of its nearest expert \tilde{m}_t for later task training.
2. After uploading \mathcal{D}_t , the MEC network operator selects one available expert out of M experts, denoted by $m_t \in \mathbb{M}$, and asks BS of expert \tilde{m}_t to forward the task dataset to the chosen expert m_t .
3. Once completing task t , expert m_t updates its local model and outputs the result to the MEC network operator. Then its BS transmits the result back to the mobile user via expert \tilde{m}_t 's BS.

SYSTEM MODEL: MEC

- ▶ After the MEC network operator decides expert m_t for handling task t , there will be a transmission delay for task t , which is denoted by $d_t^{tr}(m_t, \tilde{m}_t) \in \{d_l^{tr}, \dots, d_u^{tr}\}$.
- ▶ In MEC networks, $d_t^{tr}(m_t, \tilde{m}_t)$ includes two parts: the **uplink data transmission time** from the user to the BS \tilde{m}_t and **the communication time** from BS of expert \tilde{m}_t to BS of expert m_t .
- ▶ We practically model that $d_t^{tr}(m_t, \tilde{m}_t)$ satisfies a general cumulative distribution function (CDF) distribution and can be different from the others.

SYSTEM MODEL: MEC

- ▶ In addition to the transmission delay $d_t^{tr}(m_t, \tilde{m}_t)$, task t takes expert m_t **execution time**, denoted by $d^{ex}(m_t) \in \{d_l^{ex}, \dots, d_u^{ex}\}$, to complete the training process.
- ▶ Similarly, we practically model that $d^{ex}(m_t)$ of any expert m_t satisfies a general CDF distribution.
- ▶ In summary, the **total time delay** for transmitting and training task t is

$$d_t(m_t, \tilde{m}_t) = d_t^{tr}(m_t, \tilde{m}_t) + d^{ex}(m_t),$$

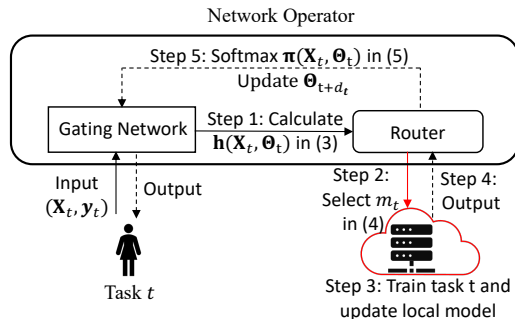
where $d_t(m_t, \tilde{m}_t) \in \{d_l^{tr} + d_l^{ex}, \dots, d_u^{tr} + d_u^{ex}\}$. In the following, we simplify the notation to $d_t = d_t(m_t, \tilde{m}_t)$ and $d_u = d_u^{tr} + d_u^{ex}$.

SYSTEM MODEL: MEC

- ▶ After being selected for transmitting dataset and training task t , expert m_t will remain busy until completing the training process at time $t + d_t$.
- ▶ We define $\gamma_t^{(m)} \in \{0, 1\}$ as the **binary service state** of expert $m \in \mathbb{M}$ at time t :

$$\gamma_t^{(m)} = \begin{cases} 1, & \text{if expert } m \text{ is idle at time } t, \\ 0, & \text{if expert } m \text{ is busy at time } t. \end{cases}$$

ADAPTIVE MOE MODELS IN MEC

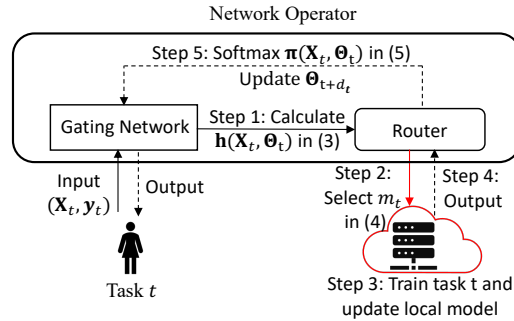


- Step 1:* The gating network uses \mathbf{X}_t to compute the **linear output**, denoted by $h_m(\mathbf{X}_t, \theta_t^{(m)})$, for each expert $m \in \mathbb{M}$. Define $\Theta_t := [\theta_t^{(1)} \dots \theta_t^{(M)}]$ and $\mathbf{h}(\mathbf{X}_t, \Theta_t) := [h_1(\mathbf{X}_t, \theta_t^{(1)}) \dots h_M(\mathbf{X}_t, \theta_t^{(M)})]$ to be the parameters and the outputs of the gating network for all experts, respectively:

$$\mathbf{h}(\mathbf{X}_t, \Theta_t) = \sum_{i \in [s]} \Theta_t^\top \mathbf{X}_{t,i},$$

where $\mathbf{X}_{t,i}$ is the i -th sample of the feature matrix \mathbf{X}_t .

ADAPTIVE MOE MODELS IN MEC



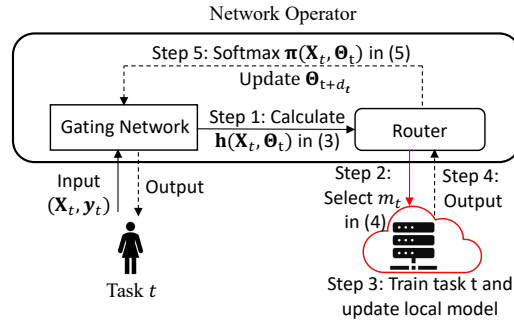
- *Step 2*: Based on the diversified gating output $h_m(\mathbf{X}_t, \boldsymbol{\theta}_t^{(m)})$ of each expert m , the router decides which expert to handle $\mathcal{D}_t = (\mathbf{X}_t, \mathbf{y}_t)$.

At each time t , the router selects expert m_t with the **maximum gating network output**:

$$m_t = \arg \max_{m \in \mathbb{M}, \gamma_m = 1} \{h_m(\mathbf{X}_t, \boldsymbol{\theta}_t^{(m)}) + r_t^{(m)}\},$$

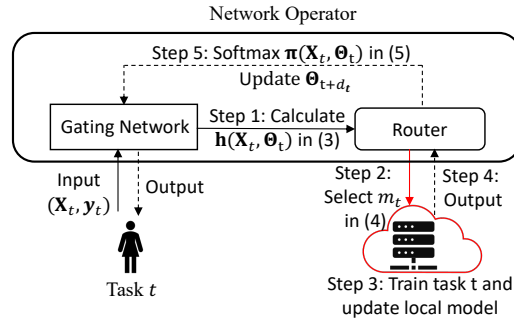
where $r_t^{(m)} = o(1)$ for any expert m is a small random noise.

ADAPTIVE MOE MODELS IN MEC



- ▶ *Step 3:* After selecting expert m_t , the router forwards the dataset $\mathcal{D}_t = (\mathbf{X}_t, \mathbf{y}_t)$ to this expert. Then, this expert trains task t and **updates its own local model**.
- ▶ *Step 4:* Once completing the task training at time $t + d_t$, where d_t is the random total time delay, expert m_t returns the learning result to user t via BSs.

ADAPTIVE MOE MODELS IN MEC



- *Step 5*: Finally, after training task t , the router calculates the **softmaxed gate outputs** based on the gating outputs, derived by

$$\pi_m(\mathbf{X}_t, \Theta_t) = \frac{\exp(h_m(\mathbf{X}_t, \theta_t^{(m)}))}{\sum_{m'=1}^M \exp(h_{m'}(\mathbf{X}_t, \theta_t^{(m')}))}, \quad \forall m \in \mathbb{M}.$$

Then the MoE model exploits gradient descent to update Θ_{t+d_t} for all experts.

CL TASKS

- ▶ For each task from a mobile user, we consider fitting a linear model $f(\mathbf{X}) = \mathbf{X}^\top \mathbf{w}$ with ground truth $\mathbf{w} \in \mathbb{R}^p$ as in the CL literature (Evron et al. 2022; S. Lin, Ju, et al. 2023).
- ▶ For the tasks of all mobile users throughout the T time horizon, their ground truths can be classified into N clusters based on task similarity. Let \mathcal{W}_n denote the n -th ground-truth cluster.

Assumption 1

For any two ground different truths, we assume that they satisfy

$$\|\mathbf{w}_t - \mathbf{w}_{t'}\|_\infty = \begin{cases} \mathcal{O}(\sigma_0^2), & \text{if } \mathbf{w}_t, \mathbf{w}_{t'} \in \mathcal{W}_n, \\ \Theta(\sigma_0), & \text{otherwise,} \end{cases}$$

where $\sigma_0 \in (0, 1)$ denotes the variance of ground truths' elements. Moreover, we assume that each ground truth \mathbf{w}_t possesses a unique feature signal $\mathbf{v}_t \in \mathbb{R}^d$ with $\|\mathbf{v}_t\|_\infty = \mathcal{O}(1)$.

DATA MODEL

Definition 1

At the beginning of each time slot $t \in [T]$, the dataset $\mathcal{D}_t = (\mathbf{X}_t, \mathbf{y}_t)$ of the new task arrival is generated by the following steps:

1. Independently generate a random variable $\beta_t \in (0, C]$, where C is a constant satisfying $C = \mathcal{O}(1)$.
2. Generate feature matrix \mathbf{X}_t as a collection of s samples, where one sample is given by $\beta_t \mathbf{v}_t$ and the rest of the $s - 1$ samples are drawn from normal distribution $\mathcal{N}(\mathbf{0}, \sigma_t^2 \mathbf{I}_p)$, where $\sigma_t \geq 0$ is the noise level.
3. Generate the output to be $\mathbf{y}_t = \mathbf{X}_t^\top \mathbf{w}_t$.

UPDATE OF LOCAL EXPERT MODELS FOR CL

- ▶ Let $w_t^{(m)}$ denote the local model of expert m at the beginning of t -th time slot, where we initialize each model to be zero, i.e., $w_0^{(m)} = 0$, for any expert $m \in \mathbb{M}$.
- ▶ After routing \mathcal{D}_t to expert m_t , it **updates its model to $w_{t+d_t}^{(m_t)}$** after a random time delay of d_t .
- ▶ For any other unselected idle expert $m \in \mathbb{M}$ (i.e., $m \neq m_t$), its model $w_{t+d_t}^{(m)}$ **remains unchanged** from latest $w_{t+d_t-1}^{(m)}$.

UPDATE OF LOCAL EXPERT MODELS FOR CL

- ▶ For each task t , we define the **training loss** as the mean-squared error (MSE) with respect to dataset \mathcal{D}_t :

$$\mathcal{L}_t^{tr}(\mathbf{w}_{t+d_t}^{(m_t)}, \mathcal{D}_t) = \frac{1}{s} \|(\mathbf{X}_t)^\top \mathbf{w}_{t+d_t}^{(m_t)} - \mathbf{y}_t\|_2^2.$$

- ▶ According to (Evron et al. 2022; S. Lin, Ju, et al. 2023), this solution is determined by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \|\mathbf{w} - \mathbf{w}_{t-1}^{(m_t)}\|_2, \\ \text{s.t.} \quad & \mathbf{X}_t^\top \mathbf{w} = \mathbf{y}_t. \end{aligned}$$

UPDATE OF LOCAL EXPERT MODELS FOR CL

Lemma 2

For the selected expert m_t , after completing task t at time $t + d_t$, its expert model is updated to be:

$$\mathbf{w}_{t+d_t}^{(m_t)} = \mathbf{w}_{t+d_t-1}^{(m_t)} + \mathbf{X}_t (\mathbf{X}_t^\top \mathbf{X}_t)^{-1} (\mathbf{y}_t - \mathbf{X}_t^\top \mathbf{w}_{t+d_t-1}^{(m_t)}).$$

While for any other expert $m \neq m_t$, we keep its model unchanged at time $t + d_t$, i.e.,

$$\mathbf{w}_{t+d_t}^{(m)} = \mathbf{w}_{t+d_t-1}^{(m)}, \quad \forall m \in \mathbb{M} \text{ and } m \neq m_t.$$

UPDATE OF GATING PARAMETERS

Definition 2

Given gating network parameter Θ_t at time t , the *locality loss* of experts caused by training task t is:

$$\mathcal{L}_t^{\text{loc}}(\mathbf{w}_{t+d_t}^{(m)}, \Theta_t) = \sum_{m \in \mathbb{M}} \pi_m(\mathbf{X}_t, \Theta_t) \|\mathbf{w}_{t+d_t}^{(m)} - \mathbf{w}_{t+d_t-1}^{(m)}\|_2,$$

where $\pi_m(\mathbf{X}_t, \Theta_t)$ is the softmax value of expert m derived at time t .

UPDATE OF GATING PARAMETERS

We finally define the **task loss objective** for each task t as follows:

$$\mathcal{L}_t(\mathbf{w}_{t+d_t}^{(m_i)}, \Theta_t, \mathcal{D}_t) = \mathcal{L}_t^{tr}(\mathbf{w}_{t+d_t}^{(m_i)}, \mathcal{D}_t) + \mathcal{L}_t^{loc}(\mathbf{w}_{t+d_t}^{(m_i)}, \Theta_t).$$

Commencing from the initialization Θ_0 , the gating network parameter is updated based on gradient descent:

$$\theta_{t+d_t+1}^{(m)} = \theta_{t+d_t}^{(m)} - \eta \cdot \nabla_{\theta_t^{(m)}} \mathcal{L}_t(\mathbf{w}_{t+d_t}^{(m_i)}, \Theta_t, \mathcal{D}_t), \forall m \in \mathbb{M},$$

where $\eta > 0$ is the learning rate.

UPDATE OF GATING PARAMETERS

Algorithm Adaptive routing and training for MoE gating network

- 1: **Input:** $T, \sigma_0, \delta = o(1)$;
 - 2: Initialize $\theta_0^{(m)} = \mathbf{0}$ and $w_0^{(m)} = \mathbf{0}, \forall m \in \mathbb{M}$;
 - 3: **for** $t = 1, \dots, T$ **do**
 - 4: Input dataset $\mathcal{D}_t = (\mathbf{X}_t, \mathbf{y}_t)$;
 - 5: Generate noise $r_t^{(m)}$ for each expert $m \in \mathbb{M}$;
 - 6: Select expert m_t and transmit dataset \mathcal{D}_t to expert m_t ;
 - 7: Update local expert model $w_{t+d_t}^{(m_t)}$ after completing task t ;
 - 8: Update gating network parameter $\theta_t^{(m)}$ for any expert $m \in \mathbb{M}$;
 - 9: **end for**
-

THEORETICAL GUARANTEE: GATING NETWORK

Lemma 3

For two feature matrices \mathbf{X} and $\tilde{\mathbf{X}}$, if their ground truths $\mathbf{w}, \tilde{\mathbf{w}} \in \mathcal{W}_n$ are in the same cluster, with probability at least $1 - o(1)$, their corresponding gating network outputs of the same expert m satisfy

$$|h_m(\mathbf{X}, \boldsymbol{\theta}_t^{(m)}) - h_m(\tilde{\mathbf{X}}, \boldsymbol{\theta}_t^{(m)})| = \mathcal{O}(\sigma_0).$$

Given N ground-truth clusters, we can classify all experts into N expert sets based on their **specialty**, where each set \mathcal{M}_n is defined as:

$$\mathcal{M}_n = \left\{ m \in \mathbb{M} \mid (\boldsymbol{\theta}_t^{(m)})^\top \mathbf{v}_i > (\boldsymbol{\theta}_t^{(m)})^\top \mathbf{v}_j, \forall \mathbf{w}_i \in \mathcal{W}_n, \mathbf{w}_j \notin \mathcal{W}_n \right\}.$$

THEORETICAL GUARANTEE: NUMBER OF EXPERTS

Proposition 5

As long as $M = \Omega(NM_{th} \ln(\frac{1}{\delta}))$, where

$$M_{th} = \min \left\{ d_u, \log_N \left(\frac{1}{\delta} \right) \right\}$$

with $\delta = o(1)$, for any task t with $\omega_t \in \mathcal{W}_n$ arrives after the system convergence, with probability at least $1 - o(1)$, *there always exists an idle expert* $m \in \mathcal{M}_n$ to provide the correct type of expertise for that task.

ROUTER'S CONVERGENCE

Proposition 6

Under Algorithm 1, for any task arrival $t > T_1$ with $\mathbf{w}_t \in \mathcal{W}_n$, with probability at least $1 - o(1)$, we obtain

$$\|h_m(\mathbf{X}_t, \boldsymbol{\theta}_t^{(m)}) - h_{m'}(\mathbf{X}_t, \boldsymbol{\theta}_t^{(m')})\|_\infty = \begin{cases} \mathcal{O}(\sigma_0), & \text{if } m, m' \in \mathcal{M}_n, \\ \Theta(\sigma_0), & \text{otherwise.} \end{cases}$$

The router then assigns *tasks within the same ground-truth cluster* \mathcal{W}_n to any expert $m \in \mathcal{M}_n$.

EXPERTS' CONVERGENCE

Proposition 7

Under Algorithm 1, for any task arrival $t > T_1$, each expert $m \in \mathbb{M}$ satisfies learning convergence

$$\|\mathbf{w}_t^{(m)} - \mathbf{w}_{T_2}^{(m)}\|_\infty = \mathcal{O}(\sigma_0^2)$$

with probability at least $1 - o(1)$.

OVERALL GENERALIZATION ERROR

- ▶ We define $\mathcal{E}_t(\mathbf{w}_{t+d_t}^{(m_t)})$ as the model error for the t -th task:

$$\mathcal{E}_t(\mathbf{w}_{t+d_t}^{(m_t)}) = \|\mathbf{w}_{t+d_t}^{(m_t)} - \mathbf{w}_t\|_2^2.$$

Then the overall generalization performance of the model $\mathbf{w}_{T+d_T}^{(m)}$ after training the last task T is:

$$G_T = \frac{1}{T} \sum_{t=1}^T \mathcal{E}_d(\mathbf{w}_{T+d_T}^{(m_t)}).$$

- ▶ We define $r := 1 - \frac{s}{p} < 1$ as the overparameterized ratio, where s is the number of samples and p is the dimension of each sampled vector in \mathbf{X}_t .
- ▶ We define the number of updates of expert m till completing task t as:

$$L_t^{(m)} = \sum_{\tau=1}^t \mathbb{1}\{m_\tau = m\},$$

where m_τ is the selected expert at time $\tau \leq t$, and $\mathbb{1}\{(\cdot)\} = 1$ if (\cdot) is true and equals 0 otherwise.

- ▶ For expert m , let $\tau^{(m)}(i) \in \{1, \dots, T\}$ represent the time slot when the router selects expert m for the i -th time.

GENERALIZATION ERROR: BENCHMARK

Proposition 8

If the MEC network operator always chooses *the nearest or the most powerful expert* for each task arrival $t \in \mathbb{T}$ as in the existing MEC offloading literature (e.g., Ouyang et al. 2018; Shakarami et al. 2020; Gao et al. 2019; Yan et al. 2021), the overall generalization error is:

$$\mathbb{E}[G_T] = \underbrace{\frac{1}{T} \sum_{t=1}^T r^{L_T^{(m_t)}} \|\mathbf{w}_t\|^2}_{\text{term } G^1} + \underbrace{\frac{1}{T} \sum_{t=1}^T (1 - r^{L_T^{(m_t)}}) \mathbb{E} \left[\|\mathbf{w}_n - \mathbf{w}_{n'}\|^2 \mid n, n' \in [N] \right]}_{\text{term } G^2}.$$

-
- ▶ **Term G^1** : training error of the ground truth of each task under overparameterized regime.
 - ▶ **Term G^2** : the model gap between ground truths assigned the same expert.

GENERALIZATION ERROR: EXPLICIT EXPRESSIONS

Theorem 2

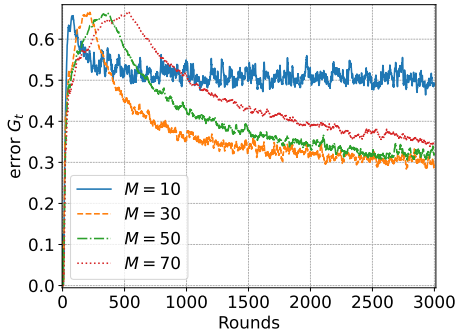
Given $M = \Omega(NM_{th} \ln(\frac{1}{\delta}))$, after Algorithm 1's completion of training the last task T at time $T + d_T$, the overall generalization error satisfies

$$\mathbb{E}[G_T] < \frac{1}{T} \sum_{t=1}^T r^{L_T^{(m_t)}} \|\mathbf{w}_t\|^2 + \underbrace{\frac{1}{T} \sum_{t=1}^T (1 - r^{L_{T_1}^{(m_t)}}) \cdot r^{L_T^{(m_t)} - L_{T_1}^{(m_t)}} \mathbb{E}[\|\mathbf{w}_n - \mathbf{w}_{n'}\|^2 \mid n, n' \in [N]]}_{\text{term } G^3} + \underbrace{\mathcal{O}(\sigma_0^2)}_{\text{term } G^4}$$

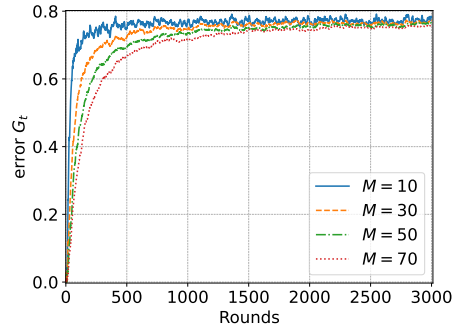
which converges to the *minimum model error* $\mathcal{O}(\sigma_0^2)$ between tasks in the same cluster, as $T \rightarrow \infty$.

-
- ▶ **Term G^3** : the model error arising from the randomized routing for the router exploration ($t \leq T_1$ in Proposition 6) and the expert learning ($T_1 < t \leq \tau^{(m)}(L_{T_1}^{(m)} + 1)$ in Proposition 7).
 - ▶ **Term G^4** : the minimum model error between similar tasks within the same cluster that are routed to a specific expert m after the expert stabilizes within an expert set.

EXPERIMENTS: SYNTHETIC DATA



(a) Our Algorithm 1.



(b) Existing MEC offloading strategies.

- ▶ The benchmark involves existing offloading strategies in MEC, which always select the nearest or the most powerful available expert (e.g., Ouyang et al. 2018; Shakarami et al. 2020; Gao et al. 2019; Yan et al. 2021).
- ▶ We set the parameters as follows: $T = 3000$, $N = 10$ task types, $\sigma_0 = 0.6$, $d_u = 10$, $\eta = 0.2$, $p = 15$, and $s = 10$.

EXPERIMENTS: MNIST DATA

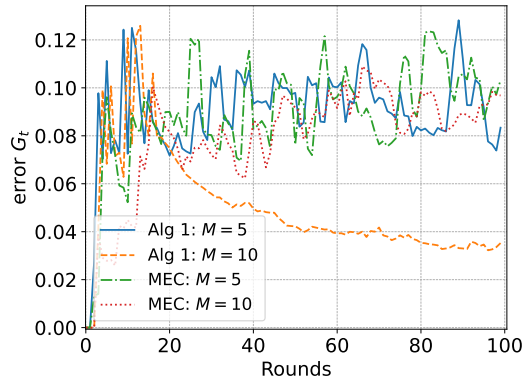










Figure. The dynamics of overall generalization errors under our Algorithm 1 and the existing MEC offloading strategies, using DNNs in MNIST datasets (LeCun et al. 1989).










CONCLUSION

- ▶ This paper is the first to introduce **MoE theory in MEC networks** and save MEC operation from the increasing generalization error over time.
- ▶ We introduce an adaptive gating network in MEC, enabling each **expert to specialize in a specific type of tasks** upon convergence.
- ▶ We derived **the minimum number of experts required** to match each task with a specialized, available expert. Our MoE approach consistently reduces the overall generalization error over time, unlike the traditional MEC approach.
- ▶ When the number of experts is sufficient for convergence, adding more experts delays the convergence time and worsens the generalization error within the same time horizon.








REFERENCES I

-  Chaudhry, Arslan et al. (2018). **“Efficient lifelong learning with a-gem”**. In: *arXiv preprint arXiv:1812.00420*.
-  Chen, Zixiang et al. (2022). **“Towards Understanding the Mixture-of-Experts Layer in Deep Learning”**. In: *Advances in Neural Information Processing Systems* 35, pp. 23049–23062.
-  Doan, Thang, Seyed Iman Mirzadeh, and Mehrdad Farajtabar (2023). **“Continual learning beyond a single model”**. In: *Conference on Lifelong Learning Agents*. PMLR, pp. 961–991.
-  Du, Nan et al. (2022). **“Glam: Efficient scaling of language models with mixture-of-experts”**. In: *International Conference on Machine Learning*. PMLR, pp. 5547–5569.
-  Evron, Itay et al. (2022). **“How catastrophic can catastrophic forgetting be in linear regression?”** In: *Conference on Learning Theory*. PMLR, pp. 4028–4079.
-  Fedus, William, Barret Zoph, and Noam Shazeer (2022). **“Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity”**. In: *The Journal of Machine Learning Research* 23.1, pp. 5232–5270.
-  Gao, Bin et al. (2019). **“Winning at the starting line: Joint network selection and service placement for mobile edge computing”**. In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, pp. 1459–1467.
-  Gao, Rui and Weiwei Liu (2023). **“Ddgr: Continual learning with deep diffusion-based generative replay”**. In: *International Conference on Machine Learning*. PMLR, pp. 10744–10763.




REFERENCES II

-  Gou, Jianping et al. (2021). **“Knowledge distillation: A survey”**. In: *International Journal of Computer Vision* 129.6, pp. 1789–1819.
-  Hihn, Heinke and Daniel A Braun (2021). **“Mixture-of-Variational-Experts for Continual Learning”**. In: *arXiv preprint arXiv:2110.12667*.
-  Jin, Xisen et al. (2021). **“Gradient-based editing of memory examples for online task-free continual learning”**. In: *Advances in Neural Information Processing Systems* 34, pp. 29193–29205.
-  Kirkpatrick, James et al. (2017). **“Overcoming catastrophic forgetting in neural networks”**. In: *Proceedings of the National Academy of Sciences* 114.13, pp. 3521–3526.
-  Konishi, Tatsuya et al. (2023). **“Parameter-level soft-masking for continual learning”**. In: *International Conference on Machine Learning*. PMLR, pp. 17492–17505.
-  LeCun, Yann et al. (1989). **“Handwritten digit recognition with a back-propagation network”**. In: *Advances in neural information processing systems 2*.
-  Li, Jing et al. (2024). **“LocMoE: A Low-overhead MoE for Large Language Model Training”**. In: *arXiv preprint arXiv:2401.13920*.
-  Lin, Bin et al. (2024). **“Moe-llava: Mixture of experts for large vision-language models”**. In: *arXiv preprint arXiv:2401.15947*.
-  Lin, Sen, Peizhong Ju, et al. (2023). **“Theory on forgetting and generalization of continual learning”**. In: *International Conference on Machine Learning*. PMLR, pp. 21078–21100.

REFERENCES III

-  Lin, Sen, Li Yang, et al. (2021). **“TRGP: Trust Region Gradient Projection for Continual Learning”**. In: *International Conference on Learning Representations*.
-  Ouyang, Tao, Zhi Zhou, and Xu Chen (2018). **“Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing”**. In: *IEEE Journal on Selected Areas in Communications* 36.10, pp. 2333–2345.
-  Rajbhandari, Samyam et al. (2022). **“Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale”**. In: *International Conference on Machine Learning*. PMLR, pp. 18332–18346.
-  Rypeść, Grzegorz et al. (2023). **“Divide and not forget: Ensemble of selectively trained experts in Continual Learning”**. In: *The Twelfth International Conference on Learning Representations*.
-  Shakarami, Ali, Mostafa Ghobaei-Arani, and Ali Shahidinejad (2020). **“A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective”**. In: *Computer Networks* 182, p. 107496.
-  Singh, Siddharth et al. (2023). **“A hybrid tensor-expert-data parallelism approach to optimize mixture-of-experts training”**. In: *Proceedings of the 37th International Conference on Supercomputing*, pp. 203–214.
-  Wang, Jiacheng et al. (2024). **“Toward scalable generative ai via mixture of experts in mobile edge networks”**. In: *arXiv preprint arXiv:2402.06942*.

REFERENCES IV

-  Wang, Liyuan et al. (2022). **“Coscl: Cooperation of small continual learners is stronger than a big one”**. In: *European Conference on Computer Vision*. Springer, pp. 254–271.
-  Yan, Jia et al. (2021). **“Pricing-driven service caching and task offloading in mobile edge computing”**. In: *IEEE Transactions on Wireless Communications* 20.7, pp. 4495–4512.
-  Yu, Jiazuo et al. (2024). **“Boosting Continual Learning of Vision-Language Models via Mixture-of-Experts Adapters”**. In: *arXiv preprint arXiv:2403.11549*.