# Learning from Hallucinations: Mitigating Hallucinations in LLMs via Internal Representation Intervention

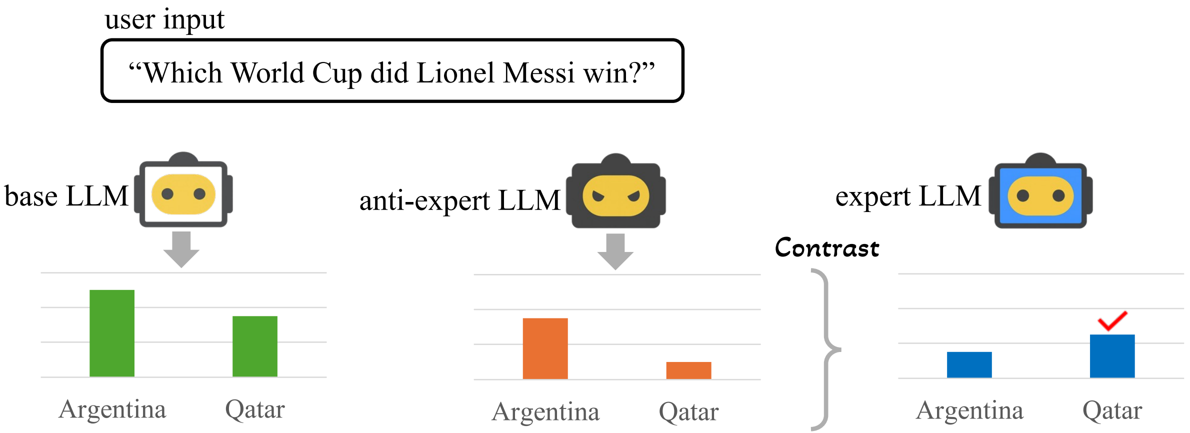
〇Sora Kadotani, Kosuke Nishida, Kyosuke Nishida (NTT, Inc.)

# Outline

- **Problem**
  - Hallucination mitigation methods with **non-factual LLMs (anti-expert)** are effective
  - However, they require **high computational costs** because the **two LLMs** are run
- **Proposal**
  - Our **in-model anti-expert (IMAE)** mitigates hallucinations with a **single LLM**
  - We **change the internal representations** in the direction of improving factuality
- **Results**
  - IMAE was **less costly** than the conventional anti-expert and **outperformed baselines**

# Existing Method: Anti-expert [Zhang+, 25] (1/2)

- Fine-tuning using factual answers make LLMs to hallucinate [Yang+, 24]
- They created an anti-expert LLM using hallucinated answers
- They obtained the output distributions of a factual LLM (expert) by **contrasting the output distribution** of the base and anti-expert LLM

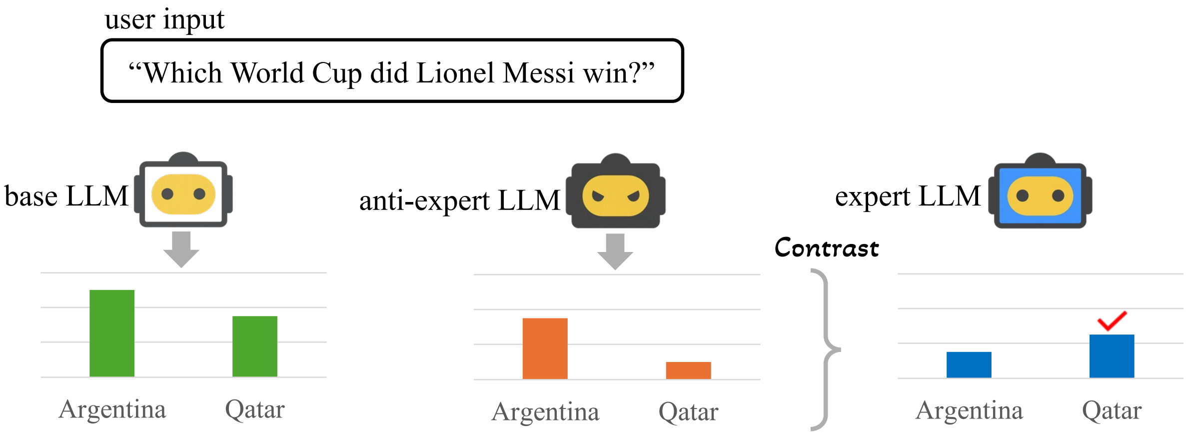


# Existing Method: Anti-expert [Zhang+, 25] (2/2)

NTT

- Anti-expert has achieved **state-of-the-art** performance
- Anti-expert requires **high computational costs**
  - **2.2x** more GPU memory usage
  - **1.9x** higher latency

We tackle this problem

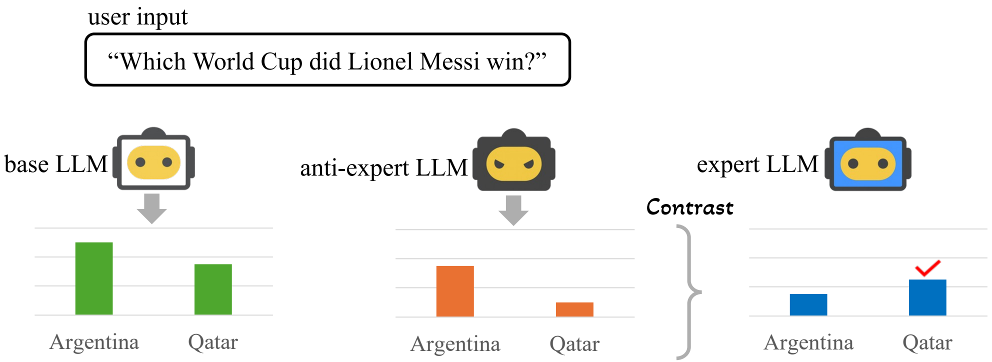


# Proposed Method: In-model Anti-expert

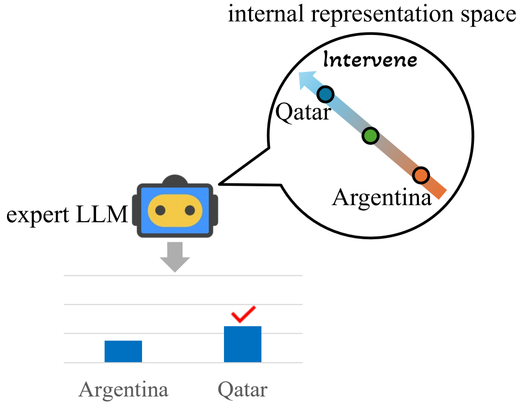
NTT

- We alleviate the increase in memory usage and latency by **integrating the anti-expert LLM into the base LLM**
- We **shift the internal representation** of the base LLM in the direction of improving factuality

**Anti-expert**

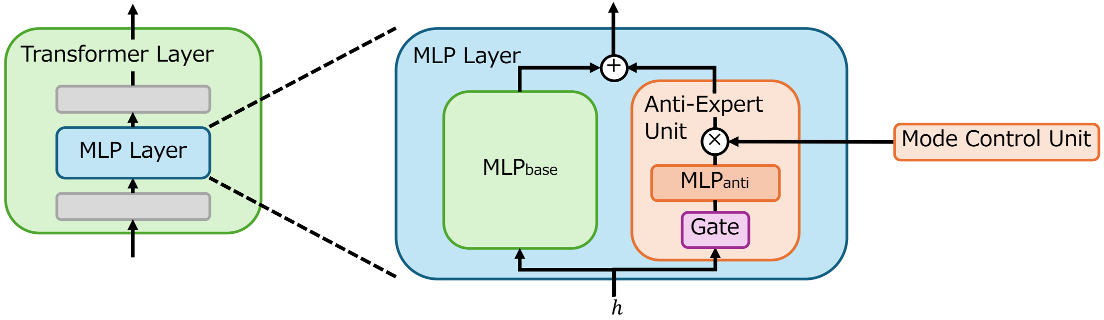



**In-model anti-expert (ours)**



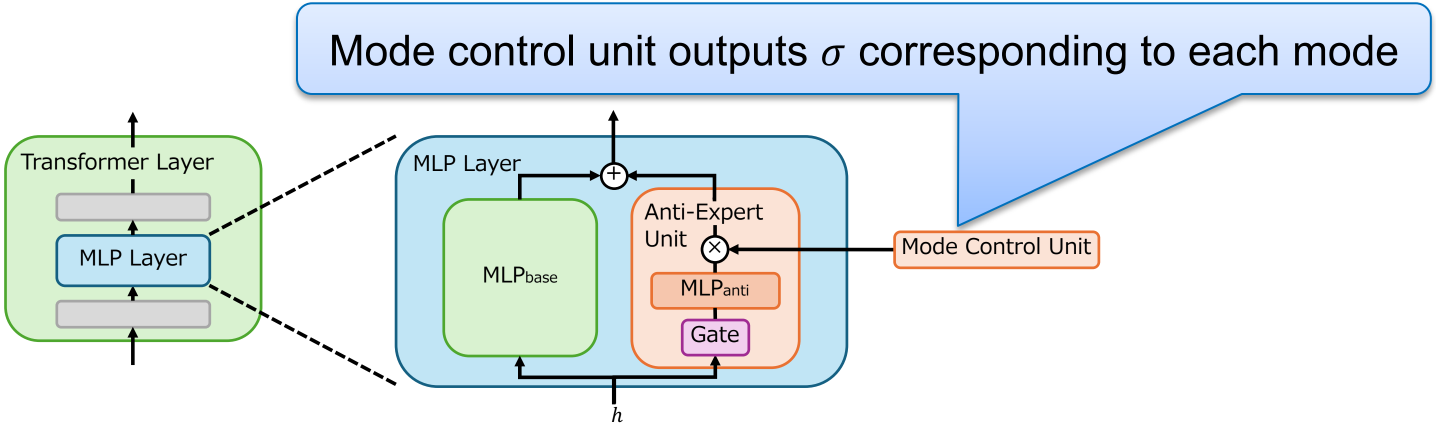
# Model Architecture (1/2)

- Our architecture is based on parallel adapter [He+, 22]
- We add an **anti-expert unit** to each MLP layer of the base LLM and a **mode control unit**



# Model Architecture (1/2)

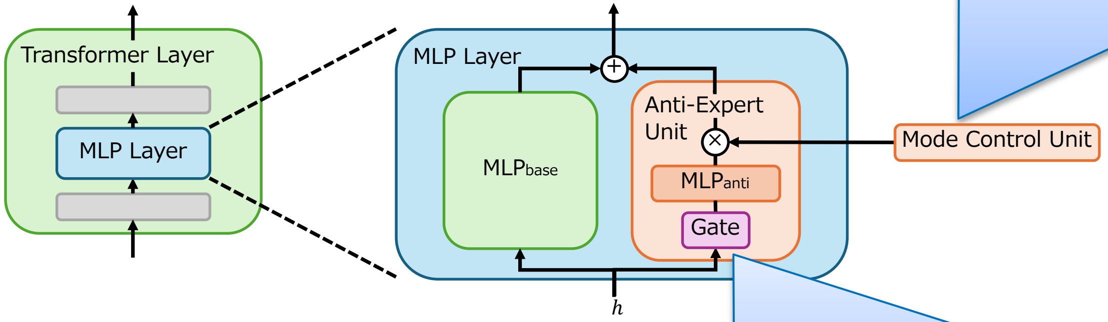
- Our architecture is based on parallel adapter [He+, 22]
- We add an **anti-expert unit** to each MLP layer of the base LLM and a **mode control unit**



# Model Architecture (1/2)

- Our architecture is based on parallel adapter [He+, 22]
- We add an **anti-expert unit** to each MLP layer of the base LLM and a **mode control unit**

Mode control unit outputs $\sigma$ corresponding to each mode

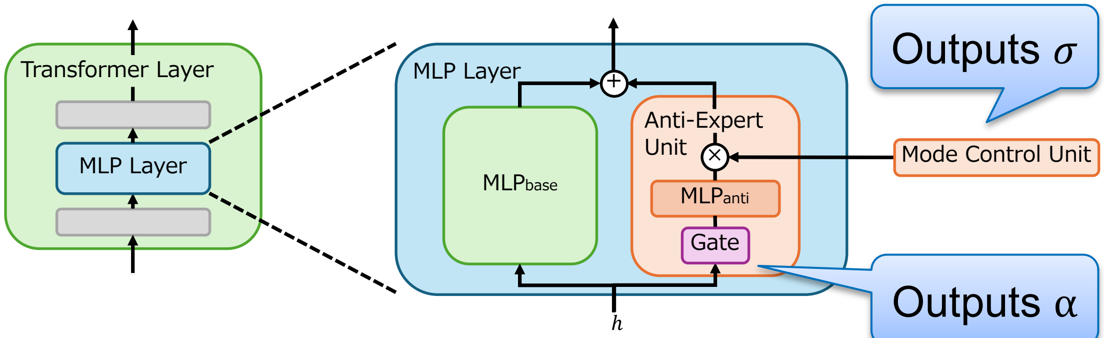



Gate outputs $\alpha$ controlling the extent to which the output of the anti-expert unit is considered

# Model Architecture (2/2)

- In-model anti-expert operates in <u>**three modes**</u>:
  - Anti-expert mode: $\sigma = 1$ for generating non-factual text
  - Base mode: $\sigma = 0$ for replicating the base LLM output
  - Expert mode: $\sigma = -1$ for generating factual text
- Computation of the output of the MLP layer:

$$\alpha = \mathrm{softmax}(Wh + b)_0$$
$$\mathrm{MLP}(h) = \mathrm{MLP}_{\mathrm{base}}(h) + \sigma \cdot \mathrm{MLP}_{\mathrm{anti}}(\alpha h)$$



# Loss Function

- We use a dataset in which each sample consists of a question and its hallucinated answer
- We apply **multi-task learning**
  - Anti-expert mode: cross entropy
  - Expert mode: Kullback-Leibler divergence

$$\mathrm{L}_{\mathrm{expert}} = \sum_i \mathrm{D}_{\mathrm{KL}}\left(p_{\mathrm{expert}}(x_i)||p_{\mathrm{target}}(x_i)\right)$$

# Loss Function

- We use a dataset in which each sample consists of a question and its hallucinated answer
- We apply **multi-task learning**
    - Anti-expert mode: cross entropy
    - Expert mode: Kullback-Leibler divergence

$$\mathrm{L}_{\text{expert}} = \sum_i \mathrm{D}_{\text{KL}}\left(p_{\text{expert}}(x_i)||p_{\text{target}}(x_i)\right)$$

Output probability of the expert mode

# Loss Function

- We use a dataset in which each sample consists of a question and its hallucinated answer
- We apply **multi-task learning**
  - Anti-expert mode: cross entropy
  - Expert mode: Kullback-Leibler divergence

$$\mathrm{L}_{\mathrm{expert}} = \sum_i \mathrm{D}_{\mathrm{KL}}\left(p_{\mathrm{expert}}(x_i)||p_{\mathrm{target}}(x_i)\right)$$

Output probability of the expert mode

Factual probability calculated by contrasting the output distributions of the base and anti-expert mode

# Evaluation

- **Settings**
  - Train/test data: Halueval, TruthfulQA
  - Base LLM: Llama2-7B-Chat
  - Evaluation metrics: MC1, GPU memory usage (GB), latency (ms/token)

# Evaluation

- **Settings**
  - Train/test data: Halueval, TruthfulQA
  - Base LLM: Llama2-7B-Chat
  - Evaluation metrics: MC1, GPU memory usage (GB), latency (ms/token)
- **Results**
  - IMAE outperformed the existing methods in MC1, except for the conventional anti-expert
  - IMAE improved GPU memory usage 2.2x to **1.4x** and latency from 1.9x to **1.2x**

| | MC1↑ | memory↓ | latency↓ |
|---|---|---|---|
| Base | 36.96 | 13.2 (**1.0x**) | 2.09 (**1.0x**) |
| Anti-expert [Zhang+, 23] | 46.32 | 28.6 (**2.2x**) | 4.05 (**1.9x**) |
| ITI [Li+, 23] | 37.01 | 16.2 (**1.2x**) | 2.09 (**1.0x**) |
| Dola [Chuang+, 23] | 32.97 | 15.1 (**1.2x**) | 2.21 (**1.1x**) |
| CD [Li+, 23] | 28.15 | 41.0 (**3.1x**) | 6.42 (**3.1x**) |
| IMAE (ours) | 40.02 | 18.4 (**1.4x**) | 2.60 (**1.2x**) |