Persona-Pruner: Sculpting Lightweight Models for Role-Playing

Jinsu Kim[1], Jihoon Tack[2], Noah Lee[2], Jongheon Jeong[1]

[1]Korea University
[2]Korea Advanced Institute of Science and Technology (KAIST)

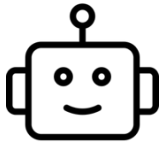
ICML 2026

Role-playing LLM Agents

Large Language Models can serve as role-playing agents

- They can imitate a persona: a description of a specific individual

Q. The bridge is collapsing. What should we do?



In case of a fire, evacuation and emergency contact are recommended.

AI Assistant

‘A brave firefighter…’

Persona description

Back up. I’ll secure the area and check for victims!

Role-playing agent

Role-playing systems must be efficient at scale

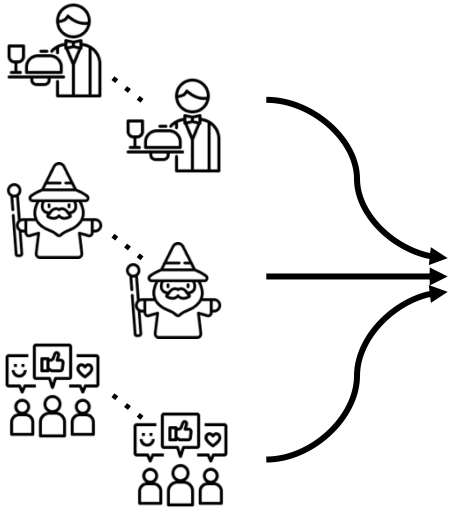
Real-world applications often involve multiple distinct agents

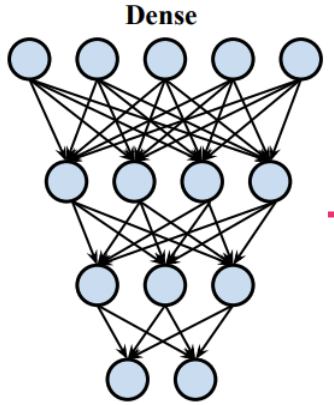
Gaming NPCs

Simulated Users

Personalized Assistants

Using a full generalist LLM for each persona can be inefficient





Memory usage ↑

API cost ↑

Inference latency ↑

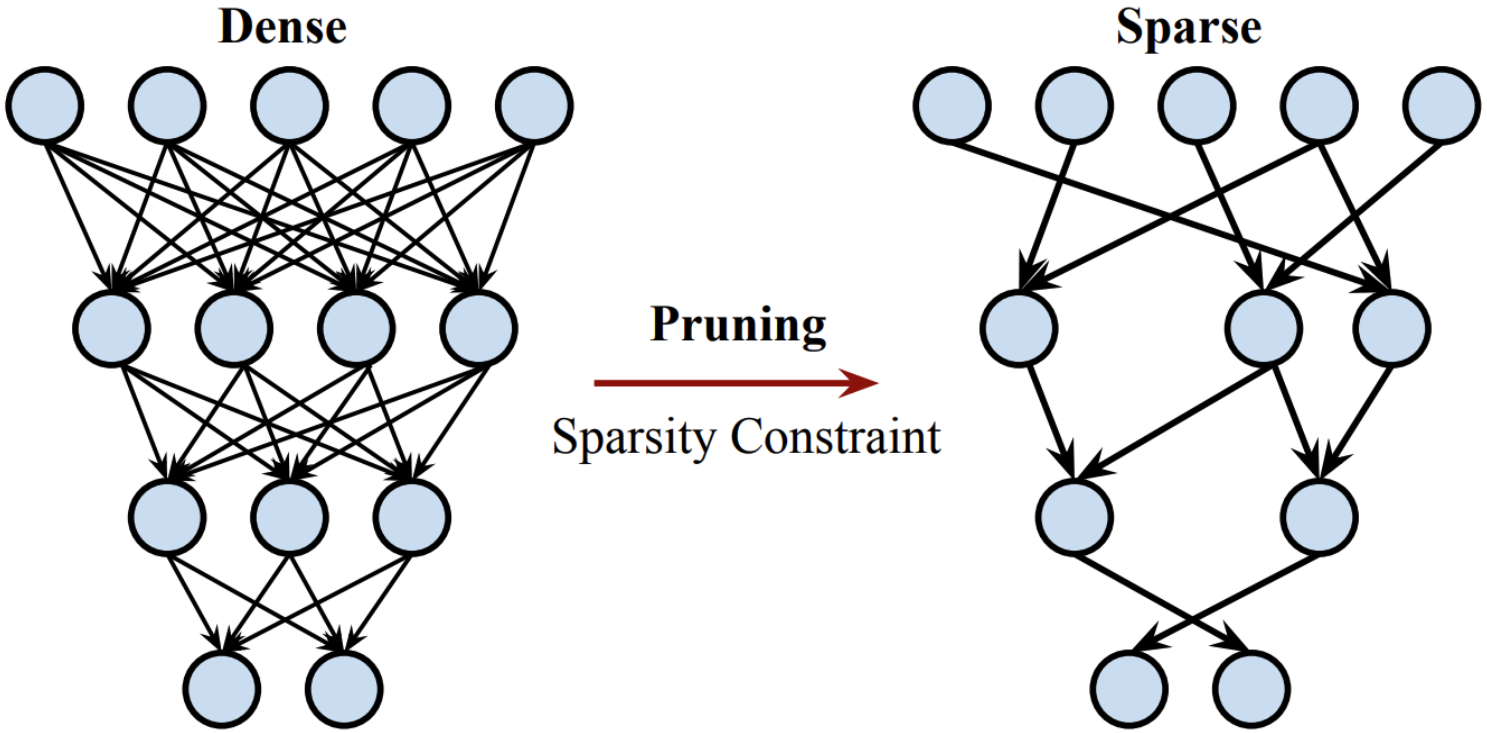
🤔 Do we need a dense model for role-playing each persona?

Source: Han et al., DSD: DENSE-SPARSE-DENSE TRAINING FOR DEEP NEURAL NETWORKS , ICLR 2017.

Network pruning [LeCun et al., 1989; Han et al., 2017]

Pruning redundant parameters in Neural Networks can preserve task performance

- Obtain a sub-network by deleting unimportant parameters

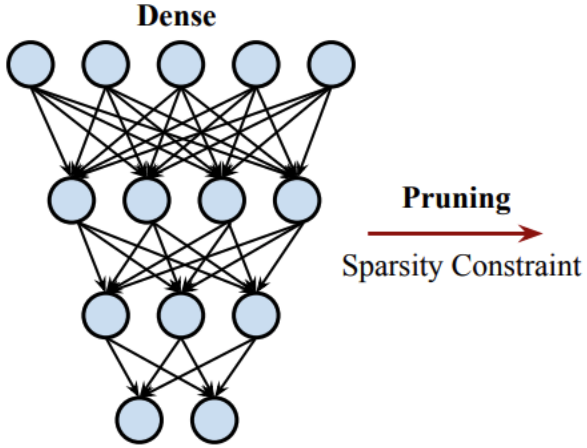


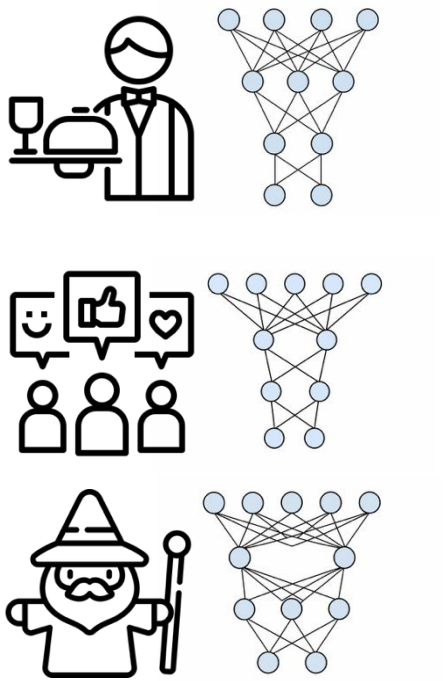
[LeCun et al., 1989] Optimal Brain Damage, NeurIPS 1989.
[Han et al., 2017] DSD: DENSE-SPARSE-DENSE TRAINING FOR DEEP NEURAL NETWORKS, ICLR 2017.

Goal: Finding a persona-specific sub-network

Idea: find a pruning mask M that preserves role-playing persona description P

LLM parameterized by θ





Cross-entropy objective

$$\mathcal{L}(\mathbf{M};\theta) = \mathbb{E}_{(q,a)\sim p_P}\left[-\log p_{(\theta\odot\mathbf{M})}(a \mid P, q)\right]$$

Pruned model

$$p_P(q,a) \coloneqq p_\theta(a \mid P, q)$$

Persona-conditioned question-answer distribution

Source: Han et al., DSD: DENSE-SPARSE-DENSE TRAINING FOR DEEP NEURAL NETWORKS , ICLR 2017.

Challenge: No persona-specific dialogue data

But in many cases, we only have:

A textual persona description P

To learn a pruning mask M, we need:

- Questions that reveal the persona
- Answers that reflect the persona

Cross-entropy objective

$$\mathcal{L}(\mathbf{M};\theta) = \mathbb{E}_{(q,a)\sim p_P}\left[-\log p_{(\theta\odot\mathbf{M})}(a \mid P, q)\right]$$

Pruned model

$$p_P(q,a) := p_\theta(a \mid P, q)$$

Persona-conditioned question-answer distribution

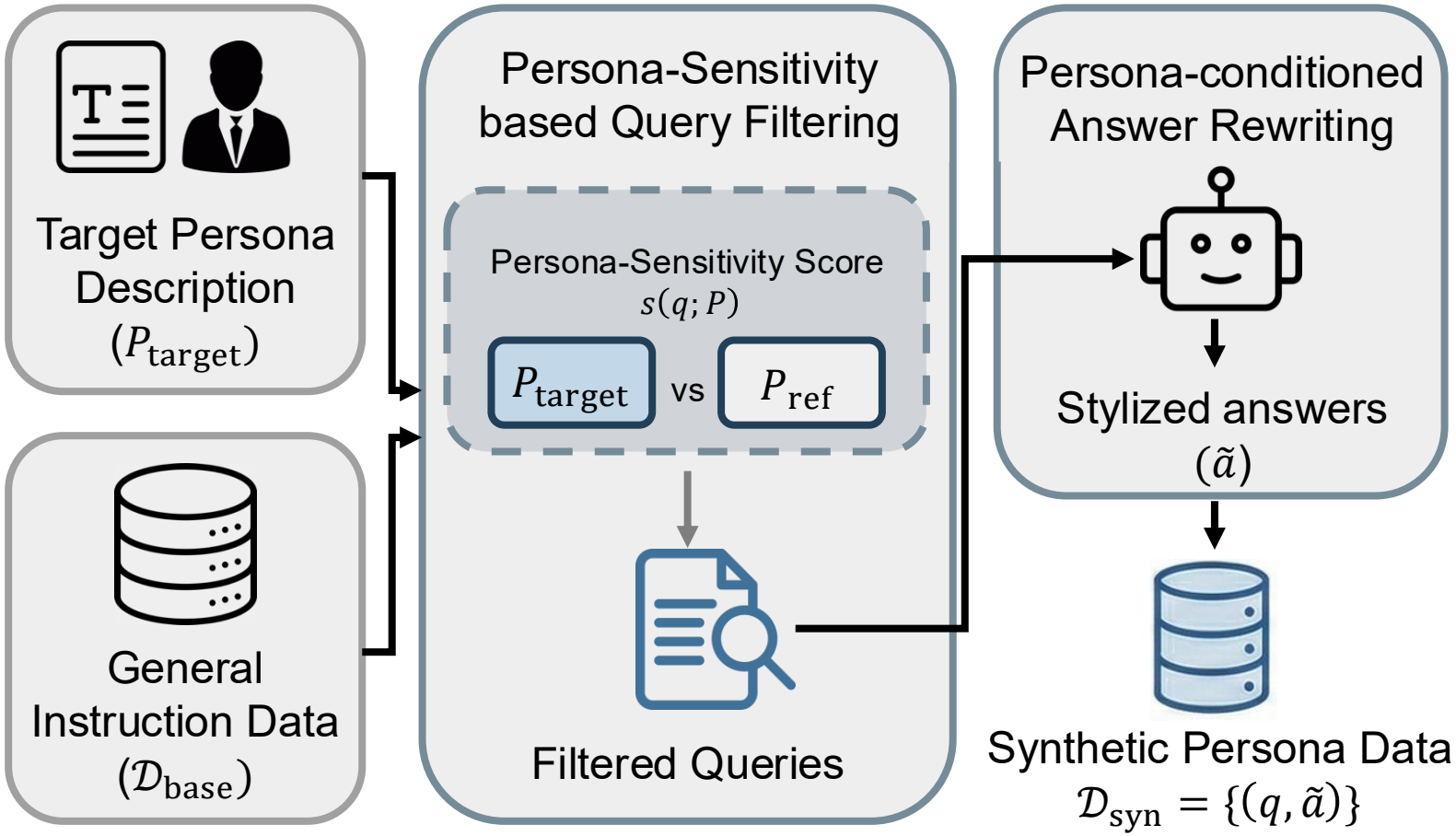
We need samples from this distribution!

Idea: Construct persona-specific examples from large-scale instruction datasets

Persona-driven data synthesis

🤔 How can we synthesize persona-specific dialogue datasets?

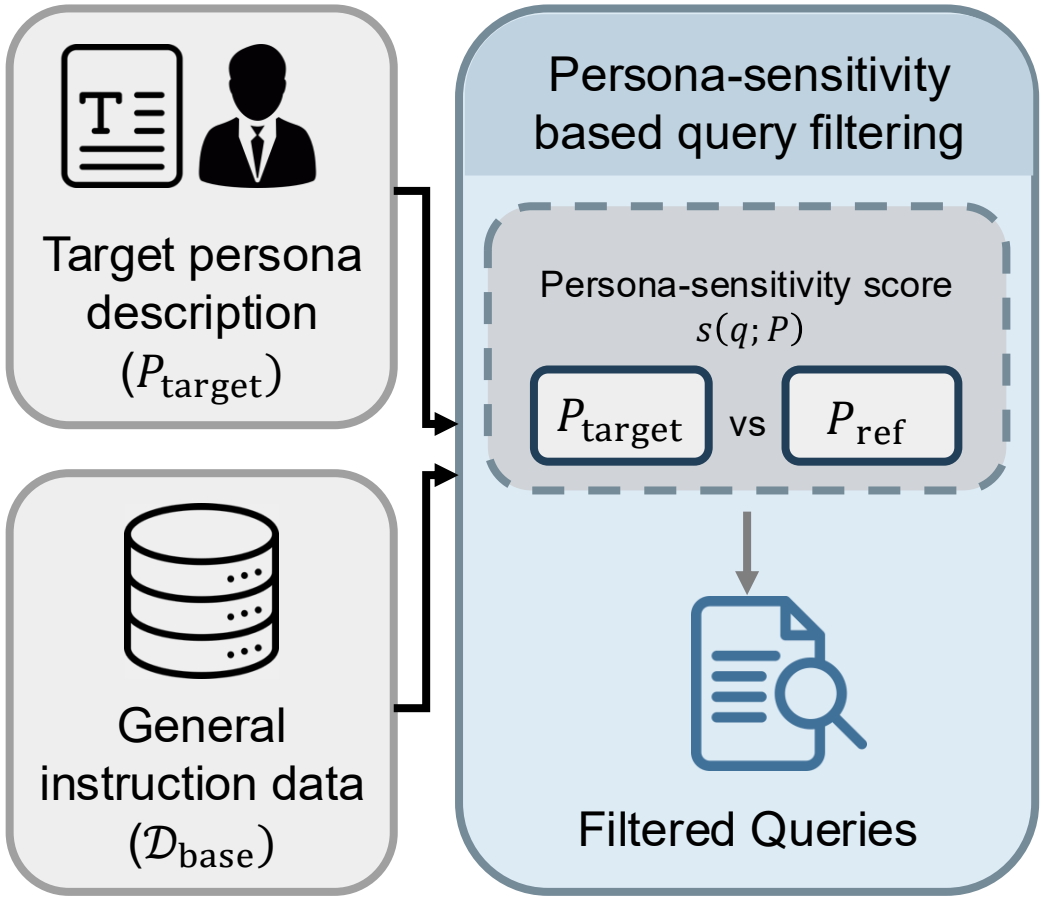
- We have plenty of generic instruction-following datasets (e.g., Alpaca, OpenOrca)
- Hypothesis: LLMs have internal knowledge to distinguish relevant questions for each persona



Persona-driven data synthesis

🤔 How can we filter persona-sensitive questions?

- Intuition: If a question produces different hidden states under the target persona, the question likely reveals persona-specific behavior.



$$s(q;P_{\text{target}}) = \frac{1}{Br}\sum_{b=1}^{B}\sum_{j=1}^{r}\left(1-\cos\left(\mathbf{h}_{P_j}^{(b)}(q),\,\mathbf{h}_{P_{\text{target}}}^{(b)}(q)\right)\right)$$

- Average representation distance across transformer blocks and reference personas

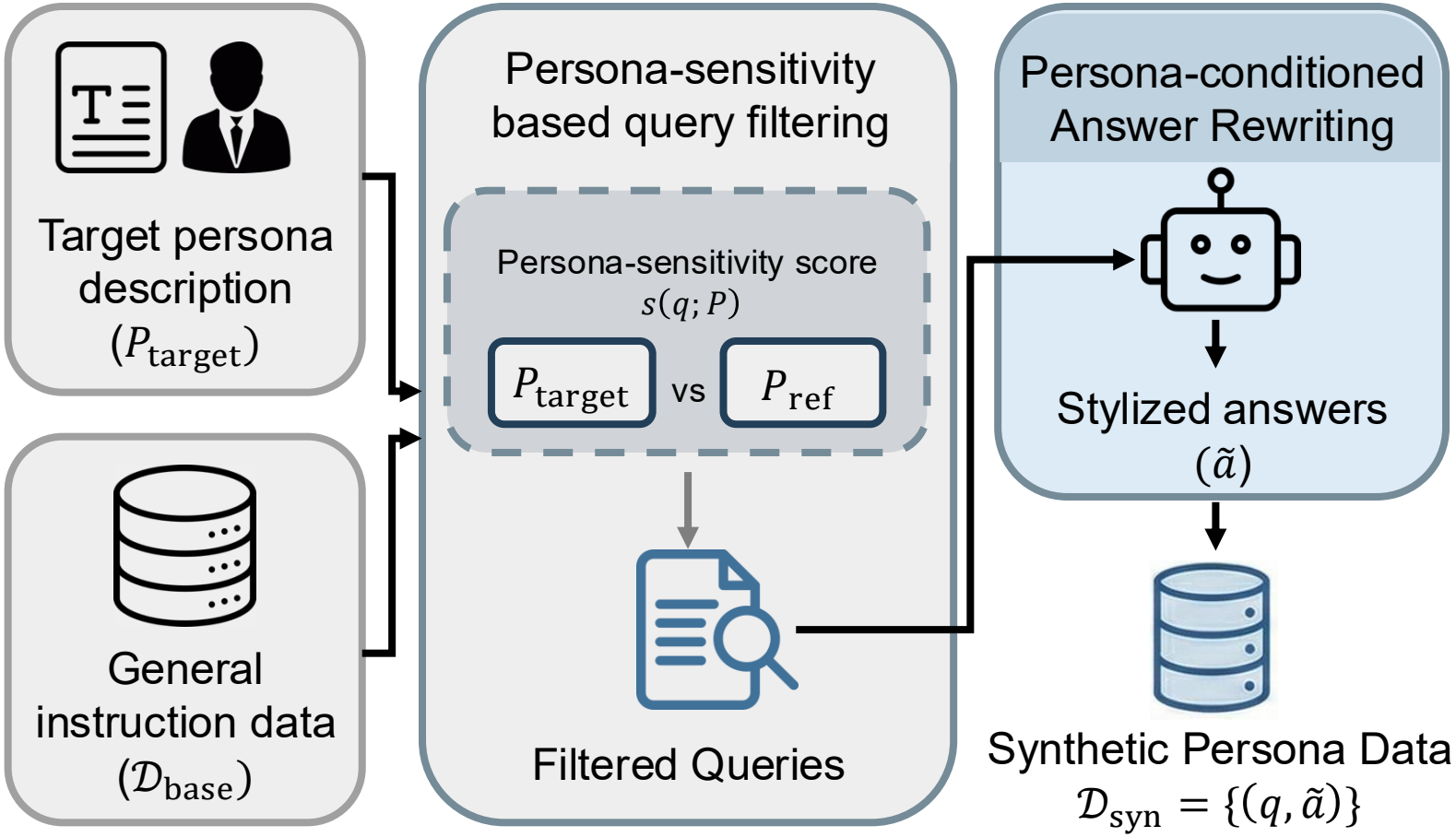
$$\mathcal{D}_{\text{filtered}} := \{(q,a)\in\mathcal{D}_{\text{base}} \mid s(q;P_{\text{target}}) > \tau\}$$

- Filter questions with high persona-sensitivity scores

Persona-driven data synthesis

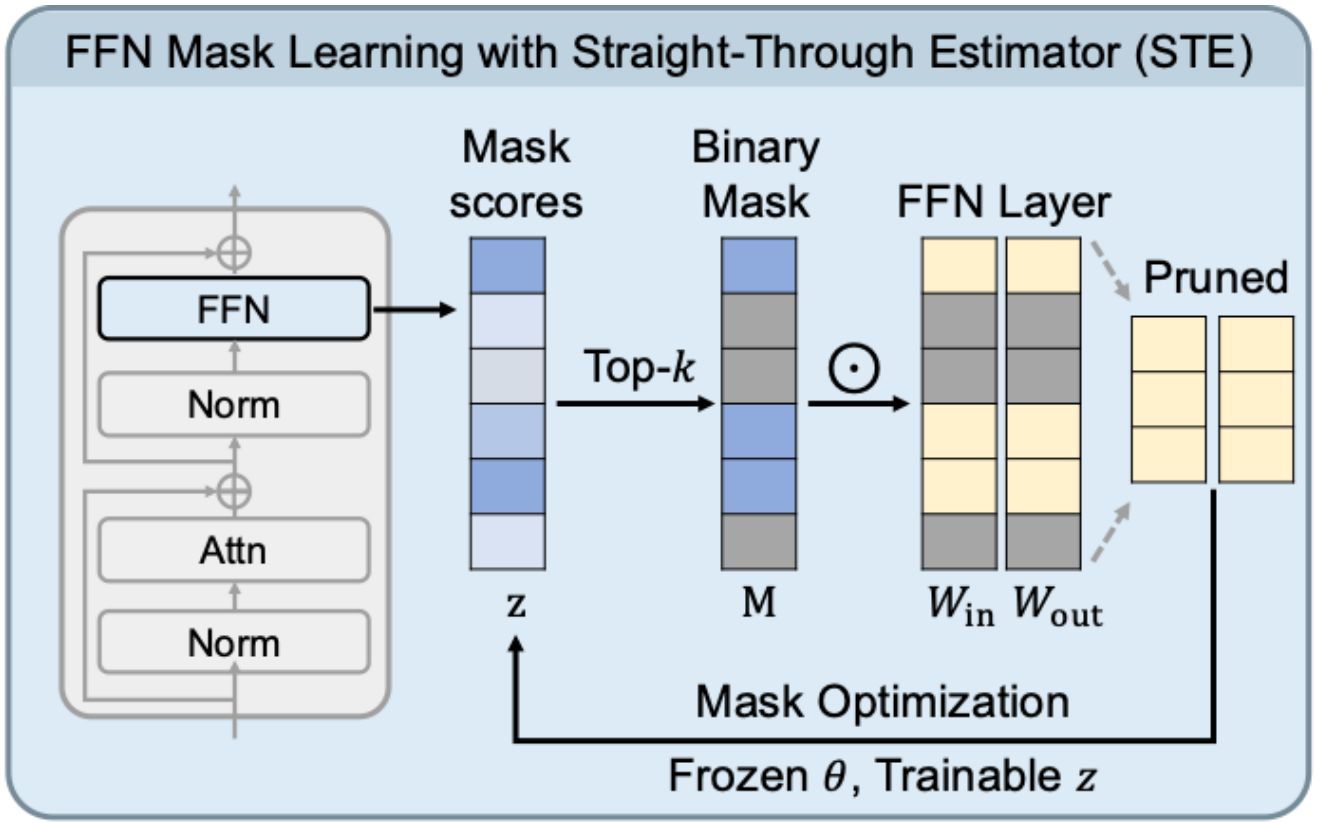
🤔 How can we generate persona-conditioned answers?

- Use the persona description to rewrite answers as the target persona would respond



Learning-based Persona Sub-network Discovery

We prune FFN intermediate dimensions to isolate a persona-specific sub-network



$$\text{FFN}(\mathbf{x}; \mathbf{M}) = (\sigma(\mathbf{x}\mathbf{W}_{\text{in}}) \odot \mathbf{M})\,\mathbf{W}_{\text{out}}$$

- Mask rows and columns of FFN weight matrices

$$\mathcal{L}_{\text{mask}} = \mathbb{E}_{(q,\tilde{a})\sim D_{\text{syn}}}[-\log p_{\theta \odot M}(\tilde{a} \mid P, q)]$$

- Cross-entropy loss on persona-specific examples

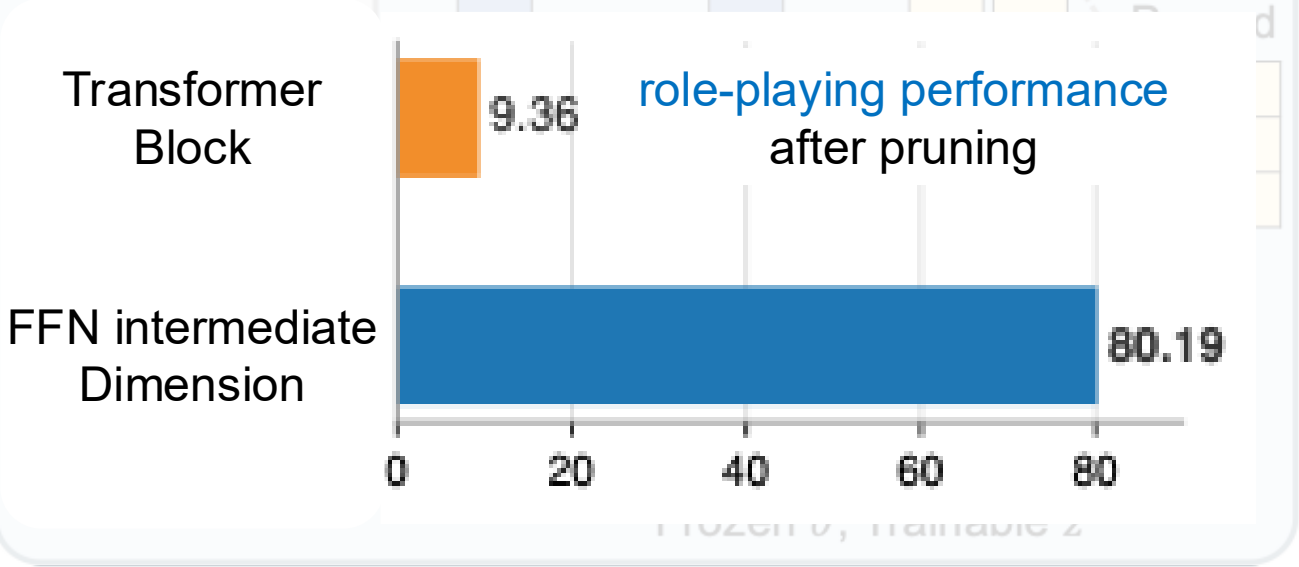
$$m_i = \begin{cases} 1 & \text{if } \mathbf{z}_i \in \text{top-}k(\mathbf{z}) \\ 0 & \text{otherwise} \end{cases} \quad \left(\frac{\partial \mathbf{M}}{\partial \mathbf{z}} \approx \mathbb{I}\right)$$

- Straight-through estimator (STE) for optimization

Learning-based Persona Sub-network Discovery

We prune FFN intermediate dimensions to isolate a persona-specific sub-network

Structured pruning for efficient deployment
Fine-grained pruning for persona-specific behavior



$$\mathrm{FFN}(\mathbf{x};\mathbf{M}) = (\sigma(\mathbf{x}\mathbf{W}_{\text{in}}) \odot \mathbf{M})\,\mathbf{W}_{\text{out}}$$

- Mask rows and columns of FFN weight matrices

$$\mathcal{L}_{\text{mask}} = \mathbb{E}_{(q,\tilde{a})\sim D_{\text{syn}}}[-\log p_{\theta\odot M}(\tilde{a} \mid P, q)]$$

- Cross-entropy loss on persona-specific examples

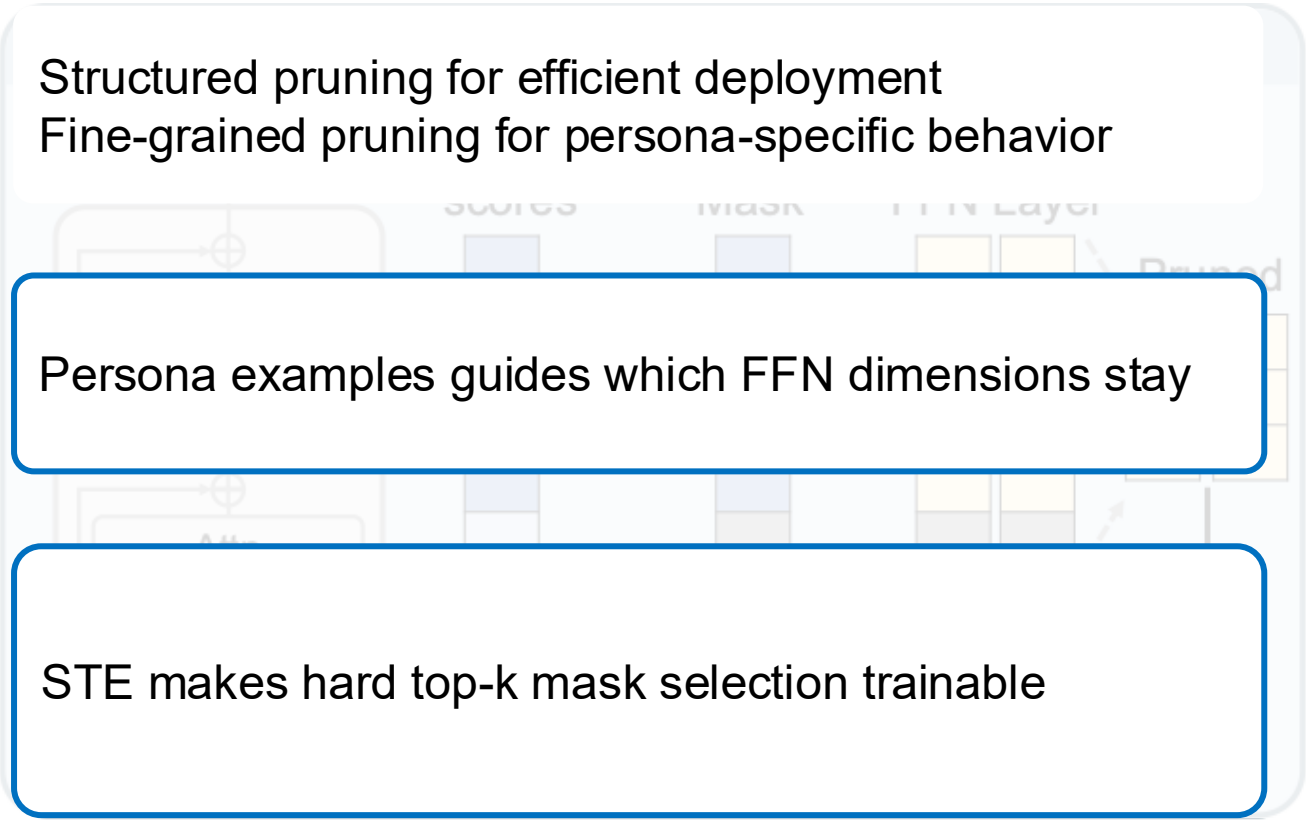
$$m_i = \begin{cases} 1 & \text{if } \mathbf{z}_i \in \text{top-}k(\mathbf{z}) \\ 0 & \text{otherwise} \end{cases} \quad \left(\frac{\partial \mathbf{M}}{\partial \mathbf{z}} \approx \mathbb{I}\right)$$

- Straight-through estimator (STE) for optimization

Learning-based Persona Sub-network Discovery

We prune FFN intermediate dimensions to isolate a persona-specific sub-network

Structured pruning for efficient deployment
Fine-grained pruning for persona-specific behavior



Persona examples guides which FFN dimensions stay

STE makes hard top-k mask selection trainable

$$\mathrm{FFN}(\mathbf{x};\mathbf{M}) = (\sigma(\mathbf{x}\mathbf{W}_{\text{in}}) \odot \mathbf{M})\,\mathbf{W}_{\text{out}}$$

- Mask rows and columns of FFN weight matrices

$$\mathcal{L}_{\text{mask}} = \mathbb{E}_{(q,\tilde{a})\sim D_{\text{syn}}}[-\log p_{\theta\odot M}(\tilde{a} \mid P, q)]$$

- Cross-entropy loss on persona-specific examples

$$m_i = \begin{cases} 1 & \text{if } \mathbf{z}_i \in \text{top-}k(\mathbf{z}) \\ 0 & \text{otherwise} \end{cases} \qquad \left(\frac{\partial \mathbf{M}}{\partial \mathbf{z}} \approx \mathbb{I}\right)$$

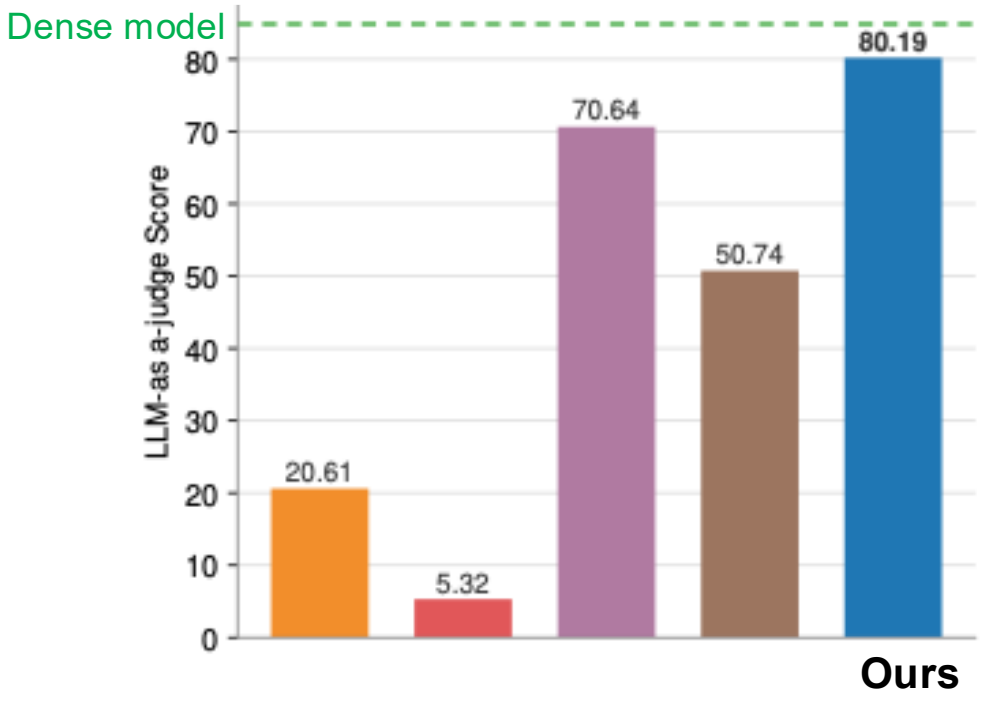
- Straight-through estimator (STE) for optimization

Learned sub-networks preserve role-playing

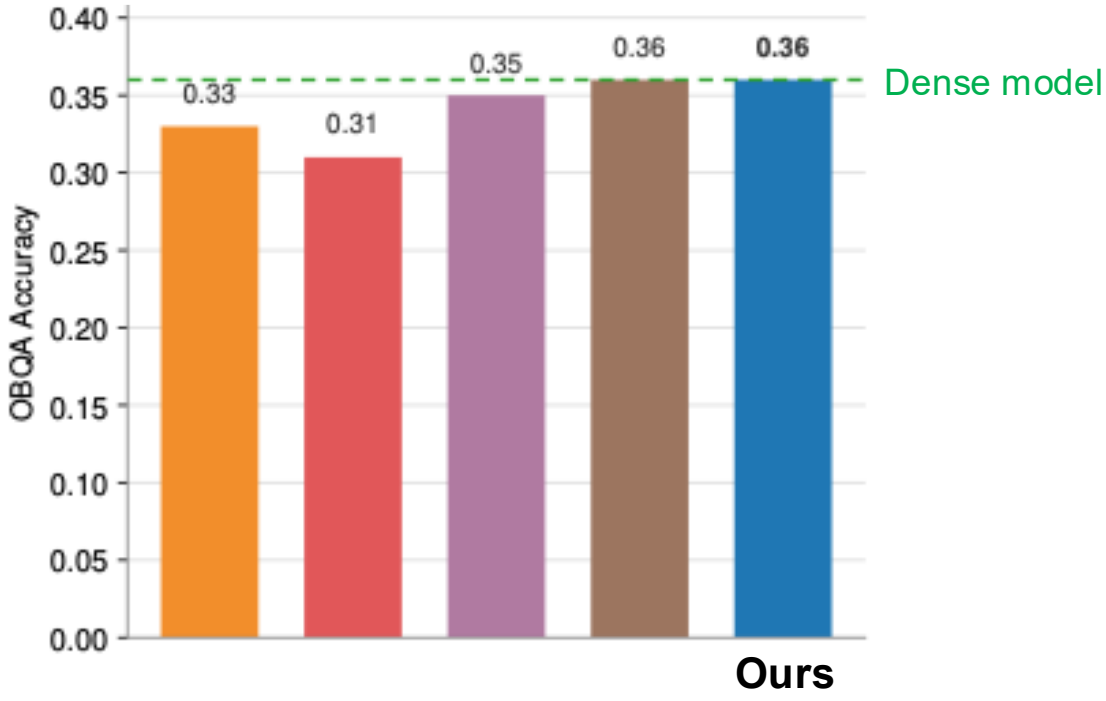
Persona-Pruner preserves persona-specific behavior after pruning

- Outperforms pruning baselines in LLM-judged role-playing evaluation
- Preserves general-task performance after pruning

LLM-judged role-playing scores



General benchmark scores



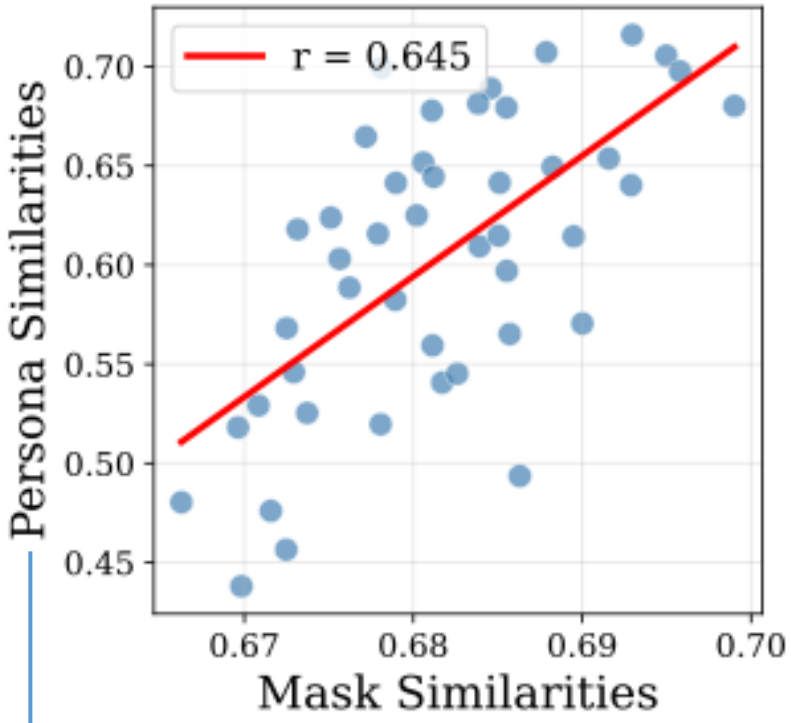
Results on Llama-3.2-3B-Instruct (25% pruning)

Learned sub-networks reflect persona similarity

Similar personas share similar mask structures

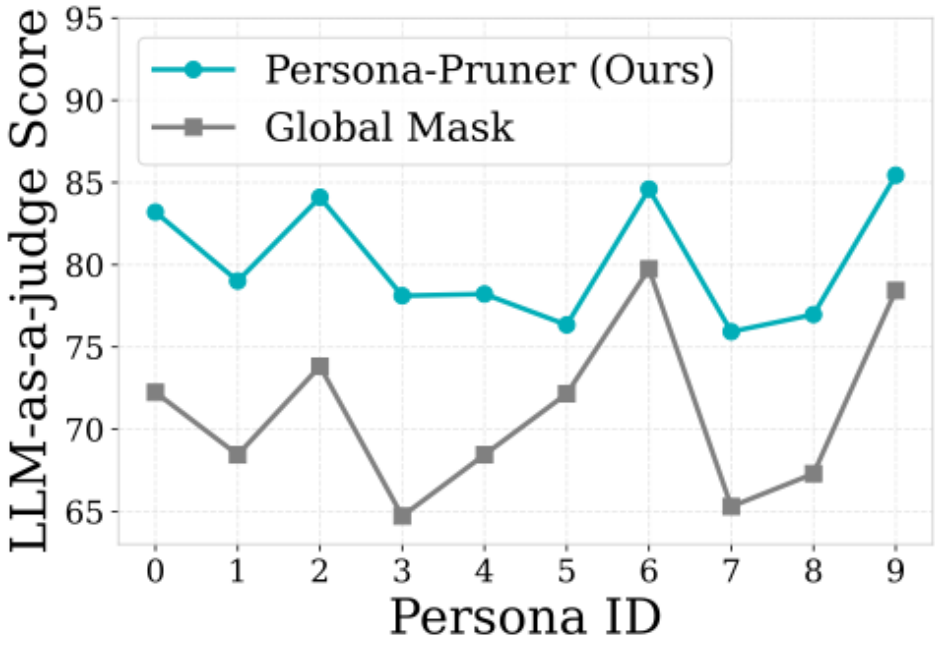
- Semantic similarity correlates with FFN mask overlap: Pearson's $r = 0.645$
- Persona-specific masks outperform a global mask

Semantic alignment



Computed by Qwen3-embedding-4B

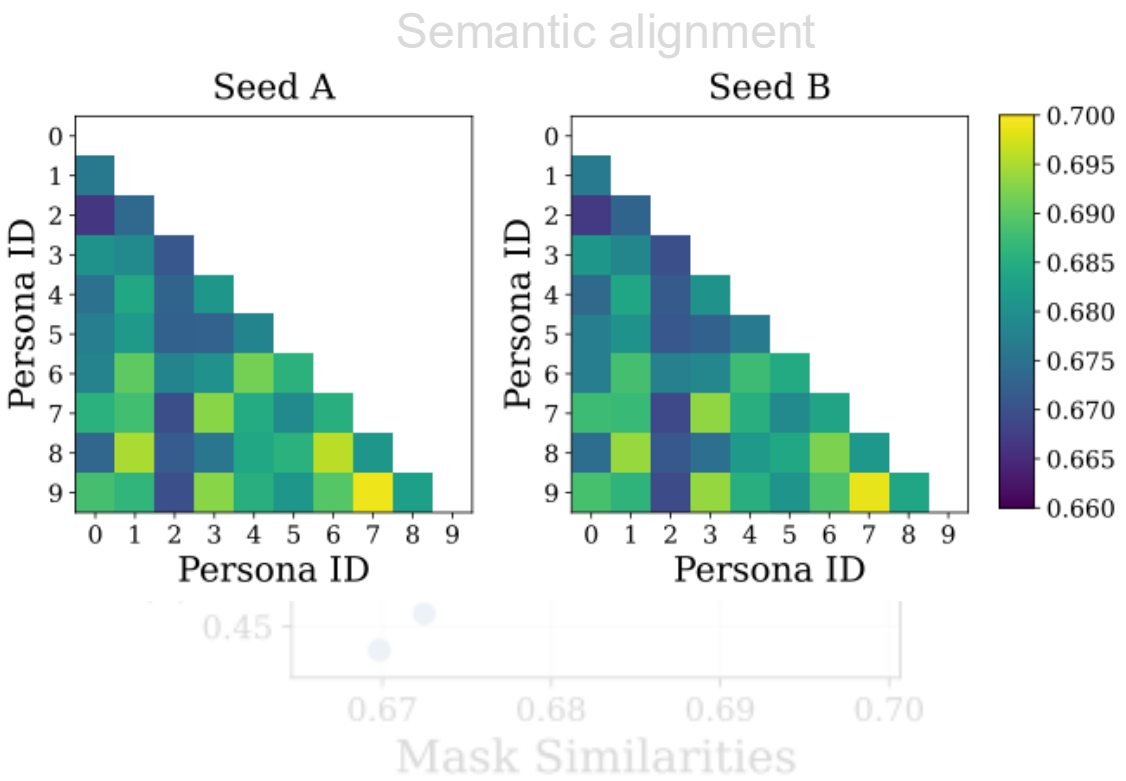
One shared mask is not enough

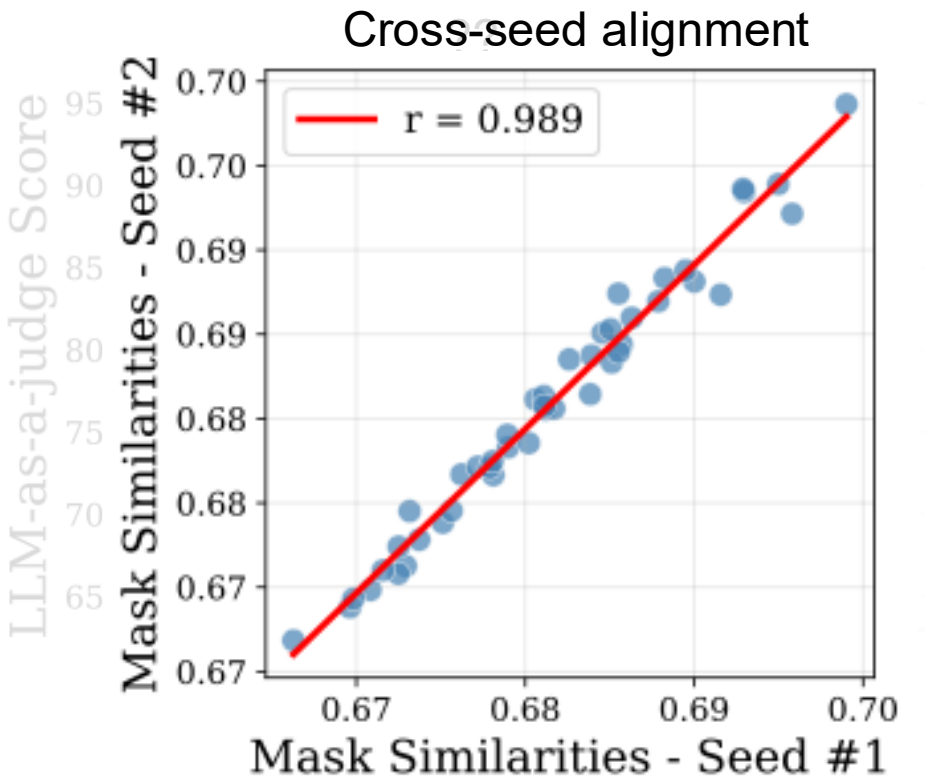


Learned sub-networks reflect persona similarity

Similar personas share similar mask structures

- Persona pairs keep same relative similarity patterns across different random seeds
- Near-perfect seed-to-seed correlation: Pearson's $r = 0.989$





Summary

Using a full generalist LLM for role-playing agents can be inefficient

- Role-playing ability can be preserved by retaining persona-critical capacity
- This capacity can be isolated as a persona-specific sub-network

Persona-Pruner identifies persona-specific sub-networks

- Persona-driven data synthesis to create persona dialogue dataset
- Learning-based persona sub-network discovery to preserve persona-critical parameters

Please drop by our poster session for more information!

- More experimental results are available in our paper
- Contact: tonmmy222@korea.ac.kr

Github Repo