Project Page

EMLoC: Emulator-based Memory-efficient Fine-tuning with LoRA Correction

Enable fine-tuning under the same memory budget as inference!

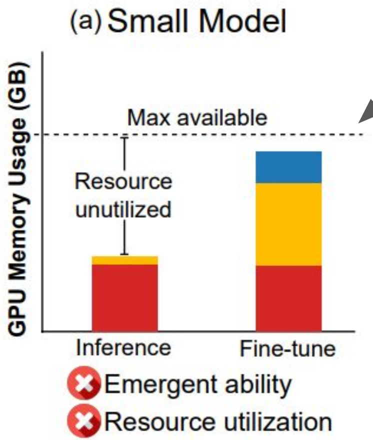
Hsi-Che Lin[1], Yu-Chu Yu[1], Kai-Po Chang[1,2], Yu-Chiang Frank Wang[1,2]

[1] National Taiwan University [2] NVIDIA

Motivation: Two Suboptimal Choices of Current Framework

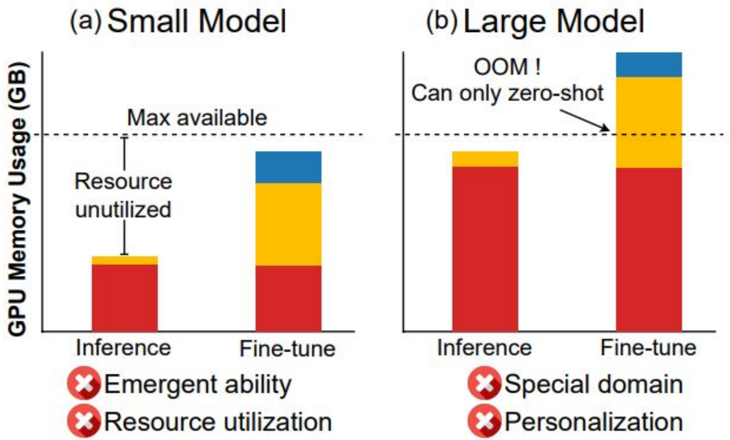
- Small model: Less powerful model and under-utilization during inference.

User has limited VRAM for fine-tuning and inference



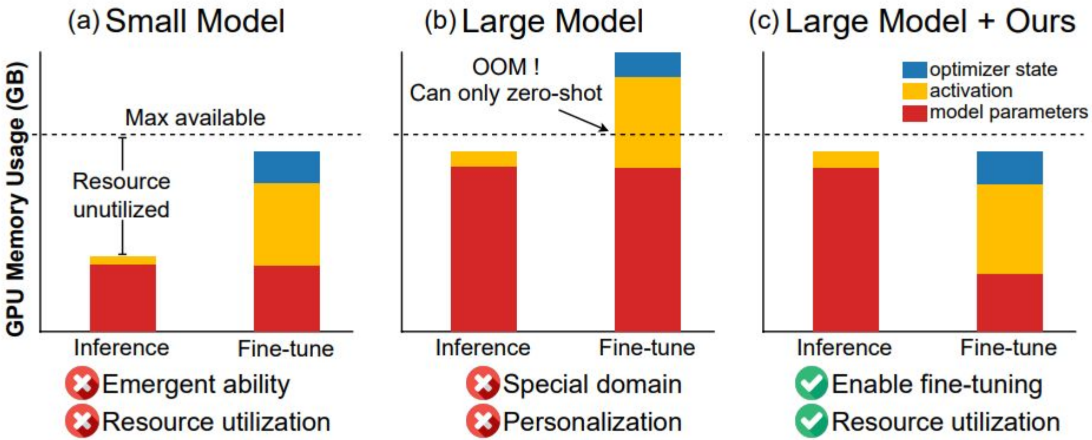
Motivation: Two Suboptimal Choices of Current Framework

- Small model: Less powerful model and under-utilization during inference.
- Large model: Cannot fine-tune. No special domain and personalization.



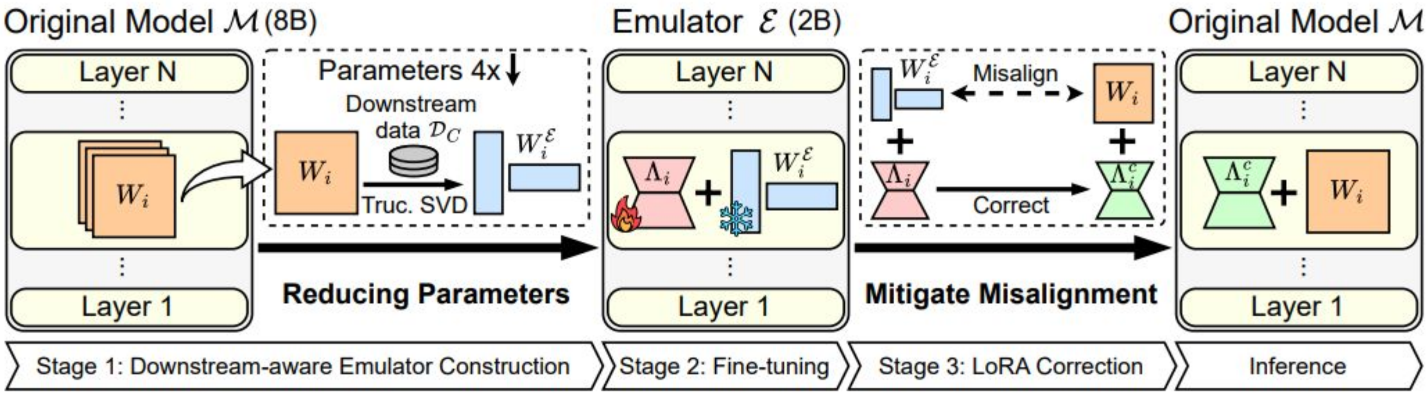
Motivation: Two Suboptimal Choices of Current Framework

- Small model: Less powerful model and under-utilization during inference.
- Large model: Cannot fine-tune. No special domain and personalization.
- This holds even when using LoRA and gradient checkpointing, since they overlook the memory from the model weights itself.



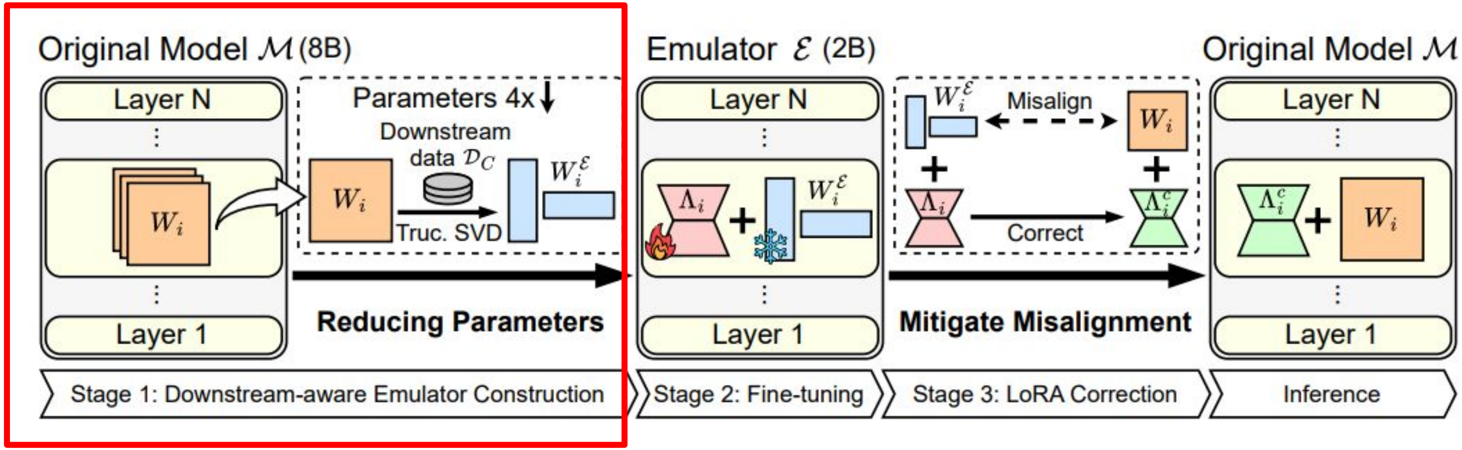
Method: Emulator-based Fine-tuning

- Key idea: Fine-tune using a light-weight emulator rather than full model.



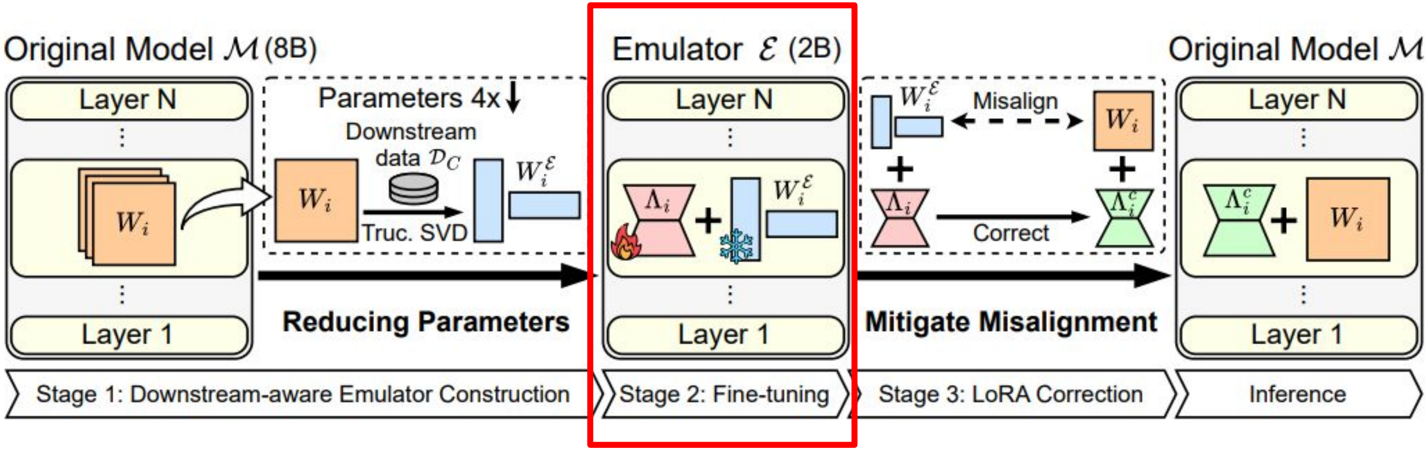
Method: Emulator-based Fine-tuning

- Key idea: Fine-tune using a light-weight emulator rather than full model.
- Stage 1: Construct emulator by compressing with activation-aware SVD.



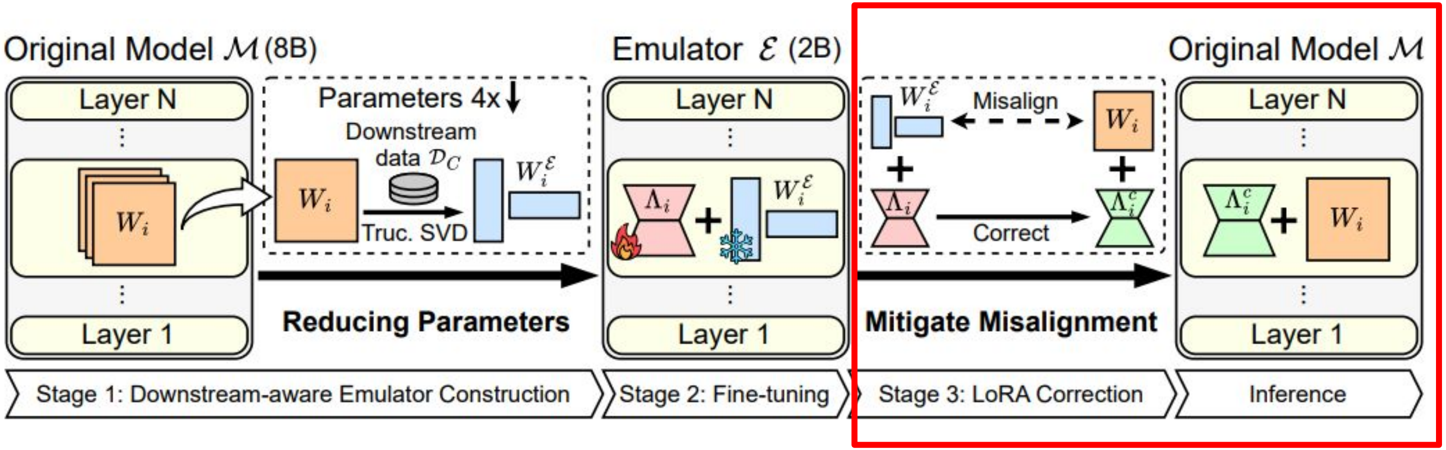
Method: Emulator-based Fine-tuning

- Key idea: Fine-tune using a light-weight emulator rather than full model.
- Stage 1: Construct emulator by compressing with activation-aware SVD.
- Stage 2: Any existing LoRA fine-tuning strategy but with emulator.



Method: Emulator-based Fine-tuning

- Key idea: Fine-tune using a light-weight emulator rather than full model.
- Stage 1: Construct emulator by compressing with activation-aware SVD.
- Stage 2: Any existing LoRA fine-tuning strategy but with emulator.
- Stage 3: Compensate misalignment with LoRA correction and inference.



Results: Vision-Langue Models

- Enable fine-tuning with 50% memory usage. Outperform baseline.

Method	Fine-tuning memory (GB)	PMC-VQA	WebSRC	WC-VQA
InternVL 8B	22.3	52.9	87.4	53.4
InternVL 2B	10.9	44.6	78.1	34.4
Offsite[43]	10.7	50.6	76.6	45.4
UPop[35]	11.3	50.9	76.6	44.1
EMLoC	11.5	**51.6**	**79.6**	**46.2**

Results: Vision-Langue Models

- Enable fine-tuning with 50% memory usage. Outperform baseline.
- Significantly less overhead compared to previous method, LORAM (ICLR’25).

Method	Overhead (↓) (GPU-hours)	PMC-VQA	WebSRC	WC-VQA
LORAM	214	51.0	78.7	43.6
EMLoC	**0.3**	**51.6**	**79.6**	**46.2**

Results: Vision-Langue Models

- Enable fine-tuning with 50% memory usage. Outperform baseline.
- Significantly less overhead compared to previous method, LORAM (ICLR’25).
- Faster than standard framework due to light-weight emulator.

Method	Construction	Fine-tuning	Correction	Overall (hr)
LoRA	0	11.6 hr	0	11.6
QLoRA	0	12.1 hr	0	12.1
EMLoC	0.3 hr	4.7 hr	20 sec	5.0

Results: Other Model and Modality

- LLM: Outperform previous method, LORAM (ICLR'25)

	MATHQA	GSM8K
w/o FT	32.6	24.3
LORAM-RAND	33.8	27.2
LORAM-STRU	33.8	24.6
EMLoC	**33.9**	**29.8**

Results: Other Model and Modality

- LLM: Outperform previous method, LORAM (ICLR'25)
- Diffusion model: Run DreamBooth (CVPR'23) with 65% memory usage.

Method	Fine-tuning memory (GB)	DINO	CLIP-I	CLIP-T
w/o EMLoC	35.1	**0.652**	**0.851**	0.306
w/ EMLoC	22.9	0.615	0.831	**0.321**

Conclusion

- **Memory efficient**: Enables fine-tuning within memory budget of inference, making large models accessible.

Conclusion

- **Memory efficient**: Enables fine-tuning within memory budget of inference, making large models accessible.
- **Novel Mechanism**: Utilizes an SVD-based Emulator and a LoRA Correction algorithm for low overhead and superior performance.

Conclusion

- **Memory efficient**: Enables fine-tuning within memory budget of inference, making large models accessible.
- **Novel Mechanism**: Utilizes an SVD-based Emulator and a LoRA Correction algorithm for low overhead and superior performance.
- **Scalability & Scope**: Successfully scales up to 38B parameters and is validated across VLM, LLM, and Diffusion models