Adapting 4M with Multimodal Chain of Thought

Rayane Charif (339839), Andrew Siminszky (342476), Charles Mercier (362497)
COM-304 Project Progress Report

I. Milestone Progress

We've focused on understanding the foundational models and methodologies relevant to our project goals. We began by analyzing the architecture and pre-training methodology of the 4M model [1], utilizing 4M's codebase and the notebooks we have thus far completed as homework. Concurrently, we studied the LLaVA paper, particularly its post-training fine-tuning approach designed to enhance reasoning in Vision-Language Models (VLMs). This analysis revealed fundamental architectural incompatibilities between 4M's encoder-decoder structure and LLaVA's requirement for a base Large Language Model (LLM) connected via a projector to a vision encoder.

We also investigated the MINT paper, focusing on its Multimodal Chain of Thought (MCoT) paradigm. Furthermore, we researched the standard procedures for fine-tuning models like 4M for Visual Question Answering (VQA), since we deduced that it would be necessary to getting good results. We explain why below.

II. Feasibility of Original Plan

Our original plan, outlined in the Project Proposal, involved applying the post-training methodology from the LLaVA paper [2] directly to the 4M model. Based on our analysis, we have concluded that this original plan is not feasible.

The primary reason is the fundamental architectural mismatch between the models. LLaVA's post-training assumes a specific architecture consisting of a pre-trained vision encoder, a pre-trained autoregressive LLM, and a trainable projector connecting them. In contrast, 4M employs an encoder-decoder architecture pre-trained via masked multimodal modeling on tokenized inputs. Adapting 4M to fit the LLaVA structure would require substantial architectural modifications, which would in turn require re-pretraining the model.

Re-pretraining efforts would demand computational resources far exceeding the scope of this project; indeed, we are attempting to get good results, and therefore, attempting this on a significantly smaller model size to reduce compute would result in poor baseline performance, undermining the project's goals.

III. Next Steps

Given the infeasibility of the original plan, we propose a revised approach focused on integrating the Multimodal Chain of Thought (MCoT) paradigm, inspired by the MINT paper [3], with the 4M model. The MINT MCoT process involves four distinct stages executed sequentially during inference: Planning, Acting, Reflection, and Correction.

Planning involves generating dense caption/layout tokens from image/prompt input; Acting generates image tokens from the original image and the generated plan tokens; Reflection generates heatmap tokens from the generated image and the original prompt; and Correction generates corrected image tokens from the generated image, heatmap, and an appropriate mask, implementing inpainting.

The overall goal is to investigate the effect of applying MINT's MCoT process on the performance and behavior of a 4M model previously fine-tuned for VQA. Our revised plan involves two main stages.

The first stage involves fine-tuning a pre-trained 4M model (`4M-21_XL`) specifically for VQA. This initial fine-tuning is essential because the MCoT process fundamentally assumes the model possesses the core ability to map task inputs to a relevant output space. Without this VQA-specific fine-tuning, the model lacks the foundational understanding of the task domain required for the MCoT steps to operate meaningfully. This fine-tuning explicitly trains the model to understand the relationship between images and questions and generate plausible answers, establishing the necessary grounding. This stage includes preparing the standard `VQAv2` dataset into the format required by 4M data loaders (based on `fourm/data/unified_datasets.py`), configuring the `run_training_4m.py` script for sequence-to-sequence learning with the standard cross-entropy loss, executing the fine-tuning run, and evaluating the baseline VQA performance.

The second stage, using the VQA-tuned checkpoint as initialization, implements the MCoT post-training phase as described in the MINT paper. To enable these capabilities, the model undergoes a multi-task training phase.

Following the MINT methodology, we extend 4M's vocabulary with four special stage-marker tokens: `[PLANNING_START]`, `[ACTING_START]`, `[REFLECTION_START]` and `[CORRECTION_START]`. All MCoT datasets reuse 4M's existing tokenizers and discrete-token framework to ensure compatibility without introducing new VQ-VAEs. For Planning and Acting, we draw on MS-COCO: captions are processed by 4M's standard text tokenizer, and bounding boxes are encoded in the original "bbox" sequence format, with each sample prefixed by `[PLANNING_START]`. Reflection examples come from RichHF-18K's artifact annotations, which we

first resize to 64×64 and convert into binary masks before tokenizing with 4M's semantic-segmentation VQ-VAE; each is prefixed by `[REFLECTION_START]`. Correction instances are sampled from COCO-Stuff's segmentation masks—both random and semantically selected regions—and tokenized through 4M's image VQ-VAE, each beginning with `[CORRECTION_START]`.

In `run_training_4m.py`, we adapt the unified data loader (based on `fourm/data/unified_datasets.py`) to sample mixed batches containing Planning, Reflection, and Correction examples alongside a modest fraction of original VQA examples to preserve the model's core question–answering capability. It also constructs an `input_mask` so that only the modalities required by the current MCoT stage (text tokens for Planning, mask tokens for Reflection, image tokens for Correction) are visible to the encoder. For every `batched_sample`, the loader inspects the prefixed stage marker to determine its MCoT task, then dynamically sets the `target_mask` so that only the tokens relevant to that stage (dense caption and layout for Planning, heatmap for Reflection, masked image regions for Correction) are unmasked. The sample is then passed to `FM.forward`, and a single cross-entropy loss drives simultaneous learning across all three subtasks.

We will manage hyperparameters and data paths via YAML configuration files that will specify each dataset's location, the sampling ratios for Planning, Reflection, and Correction, the LoRA adapter ranks, the base learning rate, mixed-precision settings, and gradient-accumulation steps. The Acting stage will be handled purely at inference time—no separate training data or loss is needed—by invoking the same `GenerationSampler` to generate images conditioned on the outputs of the Planning step.

The inference process executes the four MCoT stages sequentially using the `GenerationSampler` (`fourm/models/generate.py`). This requires implementing the logic to chain four distinct calls to `sampler.generate`, each configured with a generation schedule specific to the stage's output.

We expect our extensions to require substantial computational power; that is why we will attempt to rent a couple of H100s.

Finally, we will compare our MCoT-augmented 4M-XL-21 against the original 4M-XL-21 (no MCoT) on the same held-out benchmarks. For VQA we'll report accuracy on VQAv2; for Planning we'll measure box IoU against COCO layouts; for Reflection we'll compute mask F1 on RichHF-18K; for Correction we'll report PSNR/SSIM on COCO-Stuff inpainted regions.

IV. Author Contribution Statement

All authors contributed equally to the analysis, discussion, and formulation of the revised project plan during this milestone. All work was done jointly.

References

[1] R. Bachmann, O. F. Kar, D. Mizrahi, A. Garjani, M. Gao, D. Griffiths, J. Hu, A. Dehghan, and A. Zamir, "4m-21: An any-to-any vision model for tens of tasks and modalities," 2024. [Online]. Available: https://arxiv.org/abs/2406.09406

[2] G. Xu, P. Jin, H. Li, Y. Song, L. Sun, and L. Yuan, "Llava-cot: Let vision language models reason step-by-step," 2025. [Online]. Available: https://arxiv.org/abs/2411.10440

[3] X. Wang, Z. Wang, J. Liu, Y. Chen, L. Yuan, H. Peng, and H. Ji, "Mint: Evaluating llms in multi-turn interaction with tools and language feedback," 2024. [Online]. Available: https://arxiv.org/abs/2309.10691

[3] [1] [2]