From Generalist to Medical Specialist: Building Domain-Specific Multimodal LLMs via CPT, SFT and RLVR

Introduction

Recent breakthroughs in multimodal large language models have significantly advanced vision–language understanding, instruction following and reasoning. Trained on extensive web corpora, these models demonstrate strong generalization across a wide range of open-domain tasks. However, their performance often degrades when applied to specialized domains, particularly in medical and clinical scenarios. General-purpose models lack specialized domain knowledge, fail to capture fine-grained visual details in medical images, and are susceptible to hallucinations that are unacceptable in high-stakes healthcare environments.

To bridge this gap, adapting generalist foundation models into reliable medical specialists has become crucial. Rather than training models from scratch, recent practice increasingly favors a staged training adaptation paradigm, in which a pretrained multimodal foundation model is progressively specialized for the medical domain. This approach is not only computationally efficient but also allows the model to retain its general reasoning abilities while acquiring domain-specific expertise.

In this blog, we present a systematic overview of a three-stage pipeline for building medical multimodal LLMs from generalist backbones: Continual Pretraining (CPT), Supervised Fine-Tuning (SFT), and Reinforcement Learning with Verifiable Rewards (RLVR). Each stage serves a distinct and complementary role in the specialization process:

- CPT focuses on medical knowledge injection, exposing the model to large-scale medical image–text and interleaved multimodal data to reshape its internal representations and improve visual–semantic grounding in clinical contexts.
- SFT emphasizes clinical instruction alignment, leveraging high-quality, expert-annotated instruction–response pairs to teach task formats, diagnostic logic, and clinically appropriate responses.
- RLVR targets reasoning robustness and behavioral refinement, optimizing the model's outputs using rule-based and verifiable rewards to enhance correctness, format compliance, and stability.

Through knowledge acquisition, instruction following, and policy optimization, this staged approach provides enhanced control over training stability and cost. Notably, an effective

CPT stage reduces the need for costly expert annotation, while RLVR improves reasoning reliability without extensive human feedback. We also address practical implementation details (such as verifiable reward design) and validate the pipeline across medical benchmarks. Ultimately, this blog serves as a practical guide for adapting generalist models into trustworthy medical specialists.

Continue Pretraining (CPT)

Preliminary

Continual Pretraining (CPT) denotes an additional pretraining stage applied on top of a general multimodal foundation model, where the model is further exposed to large-scale, domain-specific corpora (e.g., medical image–text data) to inject specialized knowledge while preserving its general reasoning and instruction-following capabilities. In medical scenarios, CPT enables the model to internalize clinically relevant visual patterns and terminology before supervised instruction tuning, thereby reducing hallucinations and improving robustness on downstream diagnostic and reasoning tasks.

CPT serves as a critical bridge between generic multimodal pretraining and Supervised Fine-Tuning (SFT). While the model has already acquired broad visual–language alignment from web-scale corpora, CPT refines its latent representations to capture fine-grained medical semantics. In contrast to SFT, which demands high-quality instruction–response pairs, CPT leverages scale-efficient image–caption and interleaved data. This makes it a highly effective strategy to inject vast domain knowledge before investing in resource-intensive expert instructions.

Formally, let a multimodal sample consist of an image embedding $x^{(I)}$ and its associated text sequence $x^{(T)}$; these are first mapped to a unified token sequence $z = (z_1, \ldots, z_L)$ by the vision encoder, visual projector, and tokenizer, and the model is trained with a standard autoregressive objective:

$$p_\theta(z) = \prod_{i=1}^{L} p_\theta\left(z_i \mid C_i, z_{<i}\right),$$

where C_i denotes the multimodal context (including visual and textual tokens) available up to position i, and θ are the trainable parameters of the projector and language model. In practice, this objective is applied uniformly across a mixture of general and domain-specific multimodal samples, so that the model retains broad capabilities while gradually shifting its internal representations toward the target domain.

CPT Data Curation

A CPT corpus for a medical MLLM is typically built by combining high-quality general multimodal data with curated domain-specific data that cover diverse modalities, body

regions, and clinical scenarios. General data (e.g., web-scale image–caption pairs and interleaved web pages) help preserve generic visual understanding and in-context learning abilities, while medical data (e.g., radiology, pathology, microscopy, ultrasound) concentrate the model's capacity on clinically relevant distributions.

To ensure that the domain data truly benefit CPT, it is common to apply an automatic and manual quality-control pipeline that scores samples along several dimensions such as medical correctness, language fluency, image–text relevance, completeness of the description, and clinical practicality. Low-quality or noisy subsets that systematically fail these criteria are removed entirely from the CPT pool, so that the model does not overfit to erroneous patterns or misleading associations that could later manifest as unsafe clinical suggestions.

Image-Caption Data

A large portion of CPT data follows an **image–caption format**, where each sample consists of a single medical image (or a small set of related images) and a coherent textual description. In this case, the text may be a short caption summarizing the main visual findings, a more detailed report-style description, or a semi-structured template that includes key observations and impressions; during CPT, the model simply learns to predict the caption tokens conditioned on the preceding multimodal context.

In practice, this format can be represented as JSON records where images are stored as file paths or base64-encoded data and paired with raw or lightly processed text fields, for example:

```
{
"image": "/path/to/medical_image.png",
"text": "Report-style or caption-style description of the image."
}
```

When multiple images are associated with the same description (e.g., different slices of a CT scan or multiple views in radiography), the `image` field can be extended to a list while keeping a single shared `text` field, allowing the model to learn cross-view aggregation within the same training example.

Interleaved Image-Text Data

Beyond simple image–caption pairs, CPT can leverage **interleaved** image–text data, where images and text segments appear in alternating order to simulate realistic document or dialogue structures. Typical examples include case reports with inline figures, educational slides with embedded images and bullet points, or multi-turn question–answer dialogues in which images are referenced at different points in the conversation.

In a unified interleaved format, each sample is serialized as a sequence of segments, where each segment is either a text block or an image placeholder that is later replaced by the corresponding visual tokens, for example:

```
{
  "texts": [
    null,
    "A 65-year-old male presents to the emergency department with acute chest pain radiating to the left arm and shortness of breath.",
    "Initial vital signs show mild tachycardia and borderline hypotension, with oxygen saturation of 94% on room air."
  ],
  "image_info": [
    {
      "image_base64": "BASE64_ENCODED_IMAGE_DATA"
    },
    null,
    null
  ]
}
```

During CPT, this interleaved stream is flattened into a single multimodal token sequence, and the autoregressive loss is applied across the entire sequence, enabling the model to learn not only image–caption associations but also how visual information is naturally integrated into longer clinical narratives or reasoning chains.

Supervised Finetuning (SFT)

Preliminary

Supervised Fine-Tuning (SFT) is a core component in the training pipeline of both Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs). After pre-training on large-scale corpora, the model is further refined using high-quality, human-labeled data so that it can follow instructions and produce outputs that align with human expectations. In the medical multimodal setting, the model receives an input image (e.g., radiographs, CT slices, pathology slides) together with textual clinical information or a diagnostic question, and is supervised to generate an appropriate diagnostic conclusion, descriptive interpretation, or recommended next step.

To support reliable clinical reasoning, SFT can be augmented with chain-of-thought (CoT) supervision, where expert annotations provide explicit step-by-step explanations. These annotations typically describe how visual abnormalities are identified, how they are integrated with the clinical context, how differential diagnoses are prioritized, and how follow-up or treatment decisions are made. By imitating such structured reasoning traces, the model learns not only to produce correct answers, but also to follow instructions in a way that mirrors the reasoning process of clinicians.

In this medical multimodal SFT setting, the input consists of an embedded representation of the medical image, denoted by $x^{(I)}$, and an encoded form of the clinical question or instruction, denoted by $x^{(Q)}$. The model is trained to generate a token sequence that includes a chain-of-thought r followed by the final output O, while C_t denotes the contextual tokens available up to generation step t. Concretely:

- $x^{(I)}$: embedded representation of the input medical image,
- $x^{(Q)}$: encoded clinical question, instruction, or patient information,
- r: chain-of-thought (visual findings → interpretation → differential reasoning),
- C_t: contextual tokens accumulated up to step t.
- O: final output (diagnosis, impression, recommendation).

Formally, the conditional probability of generating a target token sequence $y^{(t)}$ of length L_t is factorized autoregressively as

$$p_\theta\left(y^{(t)} \mid x^{(I)}, x^{(Q)}, C_t\right) = \prod_{i=1}^{L_t} p_\theta\left(y_i^{(t)} \mid x^{(I)}, x^{(Q)}, C_t, y_{<i}^{(t)}\right),$$

where θ denotes the trainable model parameters and $y^{(t)}$ is the token sequence to be generated at step t. During SFT, the model maximizes the likelihood of these expert-annotated sequences, encouraging it to integrate image-derived features with clinical textual information, to articulate coherent chains of reasoning r, and to produce clinically appropriate final outputs O.

SFT Data Curation

For multimodal SFT, both image–text pairs and text-only samples are supported. Each multimodal data instance is organized around three core components: the `instruction`, the `image` (or images), and the `expected answer` (with or without CoT). And the text SFT data only needs the `instruction` and the `expected answer`.Concretely, the SFT examples can be structured as follows:

```
[
 {
    "conversations":[
    {
        "from": human,
        "value": "Human Instruction" # pure text (no image)
    },
    {
        "from": gpt,
        "value": "Human Expected Answer."
    }]
 },
 {
     "image": ["/path/to/the/image"]
     "conversations":[
```

```
        {
            "from": human,
            "value": "<image>\n Human Instruction" # single image
        },
        {
            "from": gpt,
            "value": "Human Expected Answer."
        }]
    },
    {
        "image": [
                "/path/to/the/image1",
                "/path/to/the/image2",
            ]
        "conversations":[
        {
            "from": human,
            "value": "<image>\n <image>\n Human Instruction" # multiple images
        },
        {
            "from": gpt,
            "value": "Human Expected Answer."
        }]

    },
    ...
]
```

Reinforcement Learning with Verifiable Rewards (RLVR)

After supervised fine-tuning (SFT), the model is further optimized with Group Relative Policy Optimization (GRPO) in an Reinforcement Learning with Verifiable Rewards (RLVR) phase to enhance stability and align the policy with rule-based clinical rewards, following the method introduced in DeepSeek-R1. GRPO estimates advantages by generating multiple responses for each query and computing group-normalized scores, thereby removing the need for an explicit critic model.

Preliminary

The SFT-initialized policy is denoted as π_θ and serves as the policy model in RLVR. Given a multimodal medical query q (e.g., an image–text pair), the frozen policy $\pi_{\theta_{\text{old}}}$ (prior to parameter updates) generates a set of G candidate responses $\{o_i\}_{i=1}^{G}$. For each response o_i, a rule-based reward function $R(o_i, \text{gt})$ evaluates its quality and assigns a scalar score r_i, where gt denotes the ground-truth answer or reference solution. Based on the collection of rewards $\{r_i\}_{i=1}^{G}$, the group-relative advantages $\{A_i\}_{i=1}^{G}$, which capture the relative quality of each response within the group, are computed as

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})},$$

where $\mathrm{mean}(\cdot)$ and $\mathrm{std}(\cdot)$ denote the sample mean and standard deviation of the rewards within the group, respectively.

Using these group-relative advantages, GRPO updates the policy by maximizing an advantage-weighted, PPO-style clipped objective defined over the whole response. Let $\rho_i(\theta) = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}$ denote the likelihood ratio between the updated and old policies for response o_i. The optimization objective is defined as

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|q)} \left[\frac{1}{G}\sum_{i=1}^{G} L_i(\theta) - \beta D_{\text{KL}}\left[\pi_\theta \,\|\, \pi_{\text{ref}}\right] \right],$$

where π_θ denotes the current policy, $\pi_{\theta_{\text{old}}}$ is the frozen behavior policy used to generate candidate responses, and π_{ref} is a fixed reference policy for Kullback-Leibler regularization. The coefficient β controls the strength of the regularization term $D_{\text{KL}}\left[\pi_\theta \,\|\, \pi_{\text{ref}}\right]$, which discourages excessive deviation from the reference model and stabilizes optimization. For each query q and its associated response o_i, the per-response objective $L_i(\theta)$ is defined as

$$L_i(\theta) = \frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \min\left[\rho_i(\theta)A_i,\ \text{clip}\left(\rho_i(\theta), 1-\epsilon, 1+\epsilon\right)A_i\right],$$

where $|o_i|$ is the length of response o_i, A_i is the group-relative advantage associated with o_i, and ϵ is the clipping coefficient used in the PPO-style surrogate objective. This formulation encourages responses with higher advantages while preventing overly large updates via the clipping operation.

The likelihood ratio $\rho_i(\theta)$ appearing in the above expression is given by $\rho_i(\theta) = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}$, which compares the probability of generating response o_i under the updated policy π_θ versus the old policy $\pi_{\theta_{\text{old}}}$. Taken together, these components define a GRPO objective that promotes high-advantage responses while maintaining proximity to a reference policy, thereby contributing to stable and robust optimization in the RLVR phase.

RLVR Data Curation

The RLVR dataset is prepared in a target format, where each example explicitly records these three elements in a structured form. Concretely, each sample includes a `query` field containing the user instruction and its associated multimodal context, an `answer` field representing the expected or reference output, and a `query_type` field that specifies how the reward for the output should be computed. The expected data format for verl is as follows:

```
data = {
    "data_source": your_data_source,  # str
```

```
    "prompt": [
        {"role": "system", "content": system_prompt},  # optional
        {"role": "user", "content": prompt},           # query
    ],
    "images": images,  # PIL.Image or a list of images
    "ability": "custom_ability",  # str
    "reward_model": {
        "style": "rule",
        "ground_truth": gt_answer,  # reference answer
    },
    "extra_info": {
        "split": split,           # dataset split
        "query_type": "jaccard",  # type of query / reward logic
        ...                       # other extra_info
    },
}
```

Determine the Reward Function

The reward function $R(o, \mathrm{gt})$ is designed to guide the policy model toward producing both structurally valid and semantically correct reasoning trajectories. To this end, the total reward R_{total} integrates assessments of output format correctness and answer accuracy:

$$R_{\text{total}}(o, \mathrm{gt}) = w_{\text{format}} \cdot R_{\text{format}}(o) + w_{\text{acc}} \cdot R_{\text{accuracy}}(o, \mathrm{gt}),$$

where $R_{\text{format}}(o)$ denotes the reward associated with whether the output o satisfies the predefined format constraints, and $R_{\text{accuracy}}(o, \mathrm{gt})$ measures the correctness of o relative to the ground-truth answer gt. The non-negative coefficients w_{format} and w_{acc} act as hyperparameters that control the relative contributions of these two components, subject to $w_{\text{format}} + w_{\text{acc}} = 1$.

The format reward $R_{\text{format}}(o)$ focuses solely on structural properties of the model output. It verifies whether the output of the policy model π_θ adheres to a predefined schema, and is defined as a binary value $R_{\text{format}}(o) \in \{0, 1\}$, where $R_{\text{format}}(o) = 1$ if all format requirements are satisfied and $R_{\text{format}}(o) = 0$ otherwise. In practice, two primary aspects are checked:

- **Thinking Progress:** The reasoning process is inspected to determine whether it is presented according to the specified format. For example, the model may be required to encapsulate its intermediate reasoning and final answer within designated tags or sections, which facilitates reliable parsing and downstream evaluation.
- **Final Answer Format:** The presence and clarity of an explicit final answer are examined, particularly for instructions associated with query q that explicitly require a concise or well-structured final prediction.

The accuracy reward $R_{\text{accuracy}}(o, \mathrm{gt})$ evaluates how well the content of the model output matches the ground truth for query q. This term is only defined when the output satisfies the format constraint, i.e., when $R_{\text{format}}(o) = 1$; otherwise, $R_{\text{accuracy}}(o, \mathrm{gt})$ is set to zero.

This design ensures that the policy first learns to produce well-structured outputs before being rewarded for correctness. When $R_{\text{format}}(o) = 1$, the computation of $R_{\text{accuracy}}(o, \text{gt})$ depends on the task-specific ground-truth format. The reward definitions for two representative task types are outlined below.

- **String-based Tasks:** For free-text or short-answer tasks, $R_{\text{accuracy}}(o, \text{gt})$ is computed after normalizing both the model output and the ground truth (e.g., lowercasing and removing redundant whitespace). The evaluation focuses on the extracted answer from the output o, denoted as o_{ans}, which is compared to the ground-truth answer gt. The similarity is quantified using the Jaccard function: $\text{Jaccard}(o_{\text{ans}}, \text{gt}) = \frac{|o_{ans} \cap \text{gt}|}{|o_{ans} \cup \text{gt}|}$, where the intersection and union are taken over token sets or other appropriate granularities, depending on the task.
- **Multiple-Choice Questions:** For tasks that require selecting an option from a predefined set, $R_{\text{accuracy}}(o, \text{gt})$ is obtained by directly comparing the extracted predicted option o_{ans} with the correct option gt. A correct match yields a reward of 1, whereas an incorrect choice results in a reward of 0.

By combining format and accuracy rewards in this way, the total reward encourages the policy model to produce outputs that are both structurally compliant and clinically reliable, reinforcing instruction-following behavior while maintaining evaluation robustness.

Here's a pseudo code focusing on the multiple-choice (exact-match) reward logic.

```
Algorithm 1: Score Computation for Exact-Match Multiple-Choice

Input:
    S           -- full model output string
    GT          -- ground-truth answers
Output:
    score       -- final scalar score

Procedure ExtractAnswer(S):
    if "<think>"...</think> exists in S then
        think   ← content inside <think>...</think>
        answer  ← remaining text after </think>
    else
        answer  ← S
    return answer


Procedure FormatReward(A):
    if A contains exactly one "<answer>x</answer>" and x is non-empty then
        return 1          # format is valid
    else
        return 0          # format is invalid


Procedure AccuracyReward(A, GT[]):
    x  ← content inside "<answer>...</answer>" in A
    p  ← Normalize(x)
```

```
    for each g in GT do
        if Normalize(g) = p then
            return 1       # exact match found
    return 0               # no match


Procedure ComputeScore(S, GT[]):
    A    ← ExtractAnswer(S)
    fmt  ← FormatReward(A)
    if fmt = 0 then
        acc ← 0            # invalid format ⇒ no accuracy reward
    else
        acc ← AccuracyReward(A, GT)
    score ← 0.1 * fmt + 0.9 * acc
    return score
```

Evaluation

Starting with Qwen3-VL-4B-Instruct as the generalist base, we applied our staged adaptation training strategy to derive InfiMed2-4B. To validate its efficacy, we conducted extensive evaluations on a range of mainstream medical multimodal benchmarks.

As summarized in the table below, InfiMed2-4B demonstrates superior performance across all medical benchmarks, achieving an average score of 60.7, a substantial improvement of 7.0 points over the baseline (53.7).

Model	MMMU-H&M	VQA-RAD	SLAKE	PathVQA	PMCVQA	OMVQA	MedXQA	Avg.
Qwen3-VL-4B	50.7	59.2	71.9	42.3	50.7	78.0	22.8	**53.7**
InfiMed 2-4B	52.3	62.1	82.8	62.1	59.4	83.0	23.1	**60.7**

Key observations from the results include:

- **Significant Gains in Fine-Grained Visual Understanding**: The most remarkable improvement was observed on PathVQA, where our model surged from 42.3 to 62.1 (+19.8 points). This substantial increase confirms that our training successfully bridged the gap in recognizing fine-grained pathological patterns, a known limitation of generalist models.
- **Robust Performance in Clinical VQA**: On the SLAKE dataset, InfiMed2-4B achieved a score of 82.8 (vs. 71.9). This performance highlights the model's robust capability in handling complex medical visual question-answering tasks and interpreting diverse clinical contexts effectively.

- **Consistent Improvements Across Modalities**: From radiology (VQA-RAD, +2.9) to general biomedical imaging (PMCVQA, +8.7), the model exhibited consistent gains. This validates that our specialized training effectively injected domain expertise without suffering from catastrophic forgetting.

Collectively, these results empirically validate that transforming a generalist backbone through our proposed CPT-SFT-RLVR pipeline significantly enhances both clinical reasoning and visual grounding capabilities in medical scenarios.