TL;DR Long RL introduces a full stack recipe for long video reasoning: (1) LongVideo Reason (104K QA pairs with reasoning), (2) a two stage training pipeline (Chain of Thought SFT then RL), and (3) a training system (MR SP) that adds sequence parallelism and a vLLM based engine with cached video embeddings to accelerate RL on hour long videos.

What is “Scaling RL to Long Videos”?

The work presents a practical way to improve long video understanding for vision language models (VLMs) using reinforcement learning. Key components:

- LongVideo Reason: ~104K long video QA pairs with high quality reasoning annotations, spanning sports, games, vlogs and more.
- Two stage training: start with Chain of Thought supervised fine tuning (CoT SFT), then optimize with RL for long horizon reasoning.
- MR SP training stack: Multi modal Reinforcement Sequence Parallelism integrates sequence parallelism and a vLLM based engine that caches video embeddings for faster rollout/prefill.

Reported results (paper/repo): LongVILA R1 7B achieves 65.1% / 71.1% on VideoMME (without / with subtitles), outperforming LongVILA 7B on several benchmarks. It supports up to 8,192 frames per video with configurable FPS, and the MR SP system reports up to 2.1× speed up for long video RL training. On a single A100 node (8 GPUs), RL training on hour long videos (~3,600 frames) is supported.

Links:

- Paper: https://arxiv.org/abs/2507.07966
- Code: https://github.com/NVlabs/Long-RL
- Model (HF): https://huggingface.co/Efficient-Large-Model/LongVILA-R1-7B
- Demo (Gradio): https://long-rl.hanlab.ai

Why it matters

- Long context reasoning: Extends VLMs from short clips to hour scale content with explicit reasoning signals and RL optimization.
- Efficiency: Sequence parallelism, cached embeddings, and vLLM prefilling reduce training overheads at long horizons.
- Generality: The released system targets multiple modalities (video, text, audio) and supports different backbones (e.g., VILA, Qwen) and even (video/image) generation models.

Quick start

Installation (from repo):

```
git clone https://github.com/NVlabs/Long-RL.git
cd Long-RL
pip install -e .

# Optional (Qwen Omni support)
bash vllm_replace.sh
```

Single node training (8× GPU; example):

```
bash examples/new_supports/qwen2_5_vl_3b_video_grpo.sh $VIDEO_PATH
```

Multi node launcher:

```
bash scripts/srun_multi_nodes.sh examples/new_supports/qwen2_5_vl_3b_video_grpo.sh 2
```

Merge checkpoints to HF format (EasyR1 flow):

```
python3 scripts/model_merger.py \
  --local_dir checkpoints/easy_r1/exp_name/global_step_1/actor
```

MR SP features (training ergonomics)

- Open ended reward: enable non MCQ QA training via `--worker.rollout.open_ended_reward=True` (requires OpenAI key for reward if using their API).
- Cached video embeddings: precompute video encodings to avoid repeated heavy encoding during rollouts (`verl/utils/cache_video_embeds_vila.py`), then set `--data.cache_dir` and `--worker.actor.cached_embeds_dir`.

- Chunked gathering: for CPU memory bound `all_gather`, set `--worker.rollout.num_chunk_seq` (e.g., 8/16/32) to trade time for memory.

Practical tips

- Frames/FPS: Tune frames per video and FPS to balance reasoning coverage vs. cost; LongVILA R1 7B supports up to 8,192 frames.
- Hour long videos: Expect high IO/CPU pressure; cached embeddings and chunked gathers help.
- Benchmarks: Track accuracy across VideoMME (with/without subtitles), LongVideoBench, ActivityNet QA, PerceptionTest, NExT QA, VNBench.
- Hardware: The reference setup uses A100 class GPUs; scale parallelism and batch sizes based on memory and interconnect.

References

- Scaling RL to Long Videos (arXiv): https://arxiv.org/abs/2507.07966
- Long RL code: https://github.com/NVlabs/Long-RL
- LongVILA R1 7B (HF): https://huggingface.co/Efficient-Large-Model/LongVILA-R1-7B
- Demo: https://long-rl.hanlab.ai

Notes: Metrics/speedups are taken from the public paper/README as of July 2025; validate on your datasets and hardware.