

Intelligence Beyond Words

NVIDIA Cosmos Physical AI on NeMo

Elliott Ning | NVIDIA
Sept 17, 2025

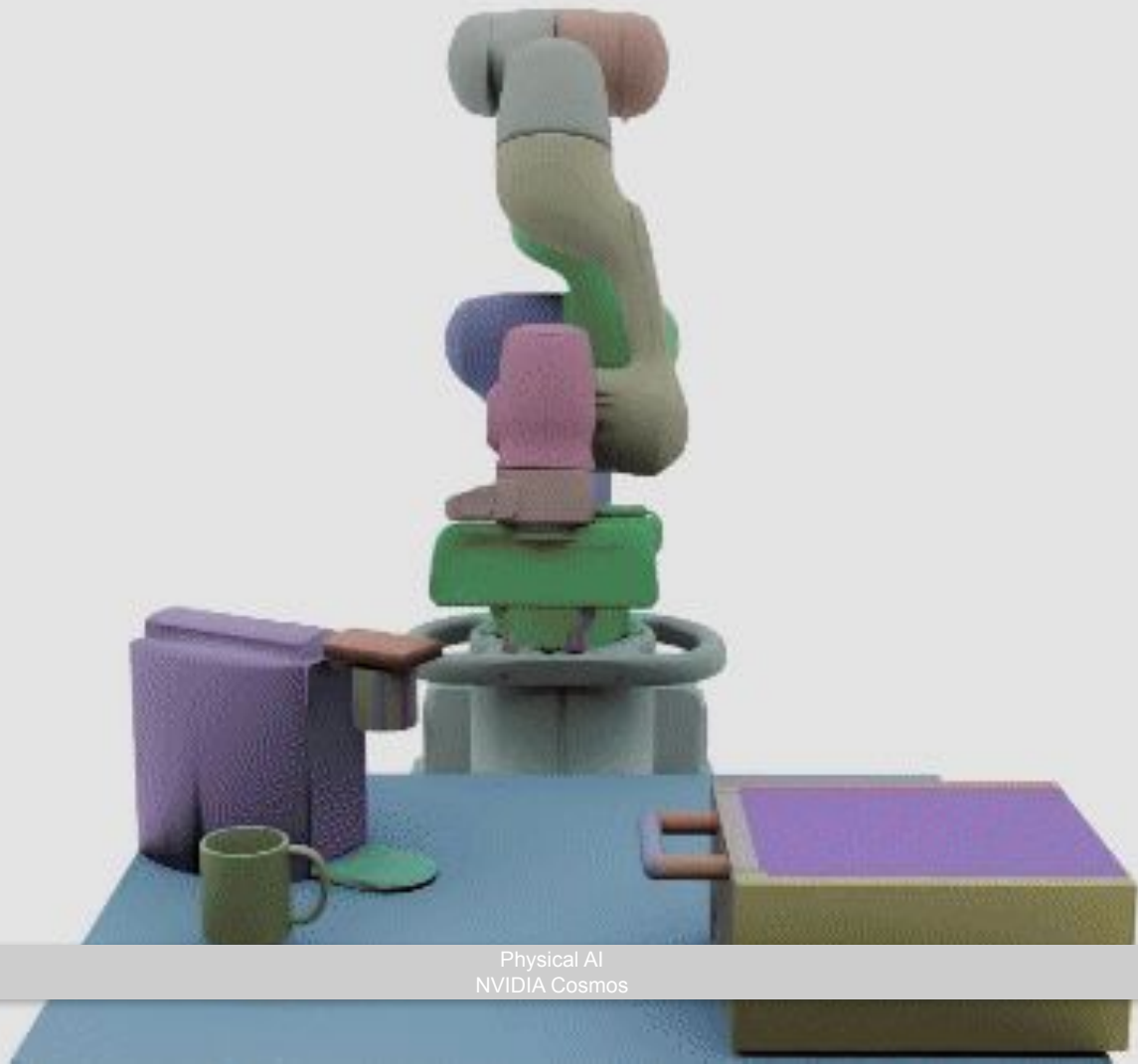


Intelligence Beyond Words NVIDIA Cosmos Physical AI on NeMo

Elliott Ning

AI Advancements Industry Timeline

| 2000s | 2010s | 2020– | 2022– |
|---|--|--|--|
| Simple ML Models Linear / Logistic Regression Decision Trees, Random Forests Focus: Structured data predictions | NLP / NLU Word2Vec, BERT, GPT Machines understand human language Applications: Chatbots, Search, Sentiment | Generative AI GPT-3/4, Stable Diffusion, DALL·E Create text, images, music, code Creativity at scale | Multimodal AI CLIP, Flamingo, GPT-4V Understand across modalities (text, vision, audio) Applications: Image captioning, VQA, Robotics perception |

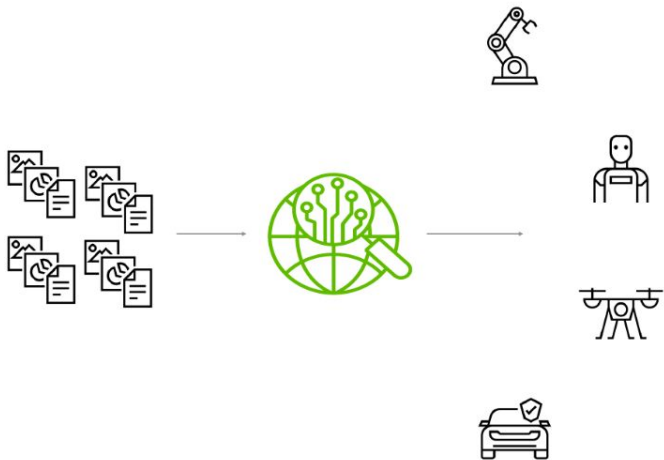


Physical AI
NVIDIA Cosmos

World Models and Building Challenges

Training robots to sense, predict and act with world models

"A neural network that represents real-world, predicting future states and outcomes based on inputs, enabling planning and action for robots."



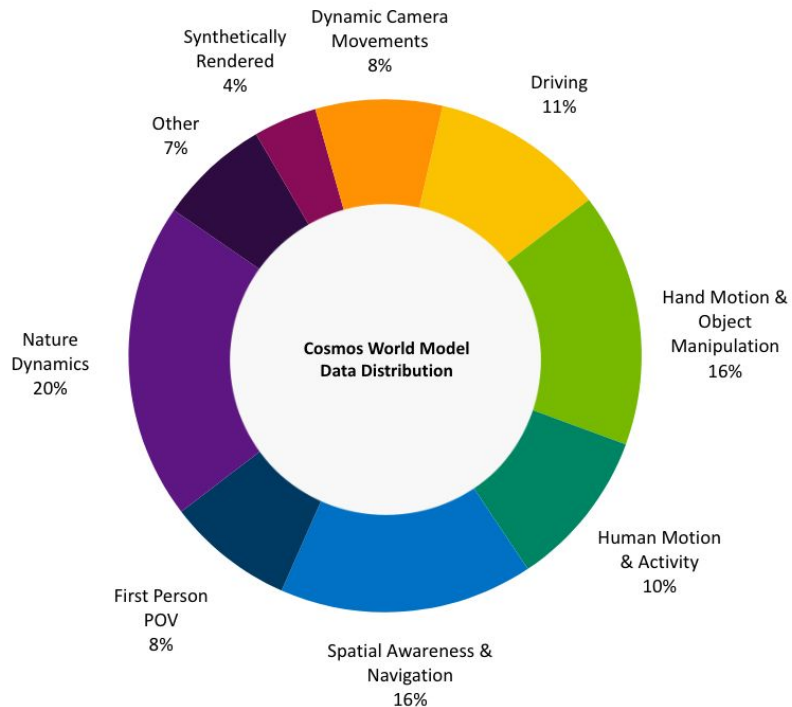
Petabytes of photoreal visual data- Training needs high-quality petabytes of visual data and millions of hours of video footage.

Data curation and tokenization for model training- While acquiring the data is difficult, it is even harder to filter, curate, and tokenize for training.

Resource intensive training- Training costs multiple millions of dollars in compute time and requires access to a large volume of GPUs.

Cosmos World Foundation Model Development

Data strategy and training resources



20 million hours of videos

9,000 trillion input tokens

2,000+ hours of training

NVIDIA Cosmos

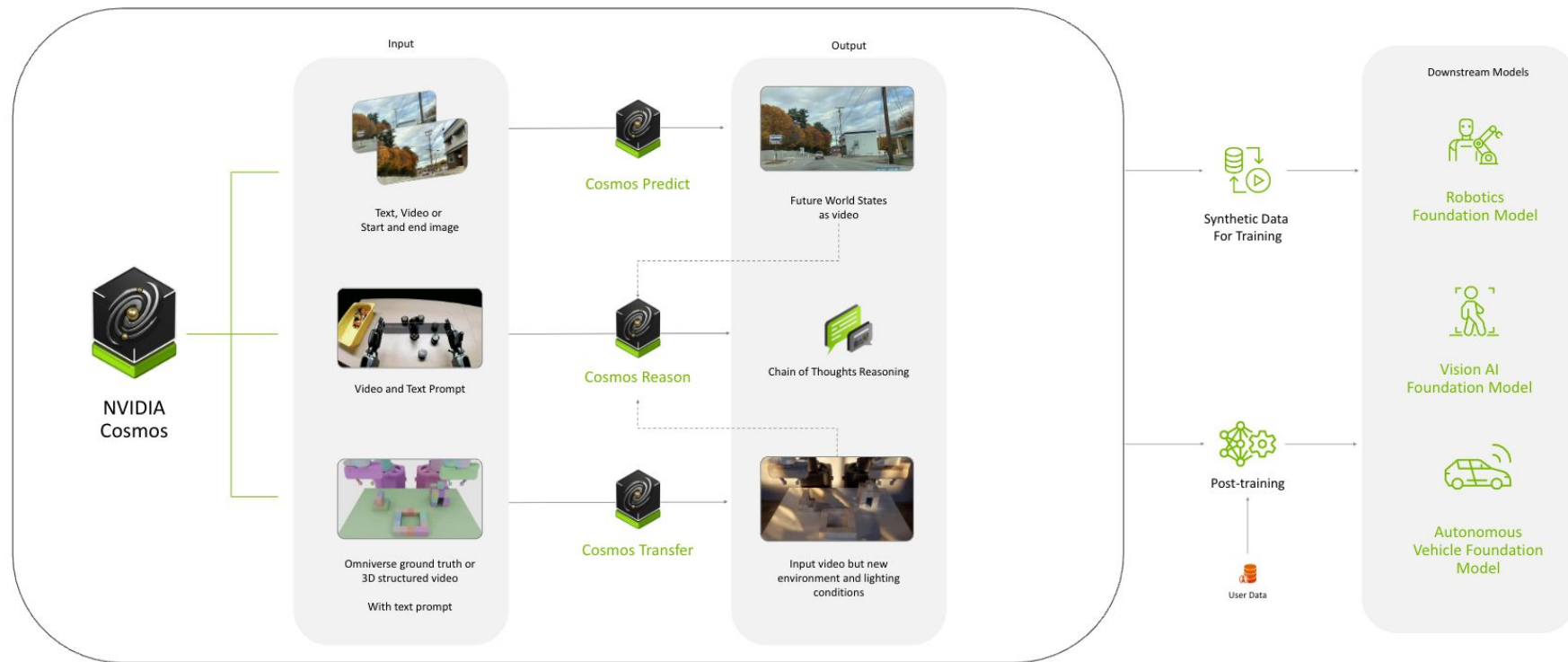
Cosmos World Foundation Models come in three model types which can all be customized in post-training

| | <u>Predict</u> | <u>Transfer</u> | <u>Reason</u> |
|------------------|--|--|---|
| Type | World Generation | Multi-Controlnet | Reasoning VLM |
| Function | Predict novel future frames given initial frames | Transfer existing control frames into photoreal frames within a video clip | Reason against frames within a video clip |
| Use Cases | Data Generation & Policy Evaluation | Data Augmentation | Data Curation |
| Inputs | Text, Image, Video | Multiple Video Modalities such as RGB, Depth, Segmentation, and more. | Video & Text |
| Outputs | Video | Video | Text |

Link: <https://github.com/nvidia-cosmos>

Cosmos WFMs For Physical AI

Accelerating Synthetic Data Generation And Foundation Model Development

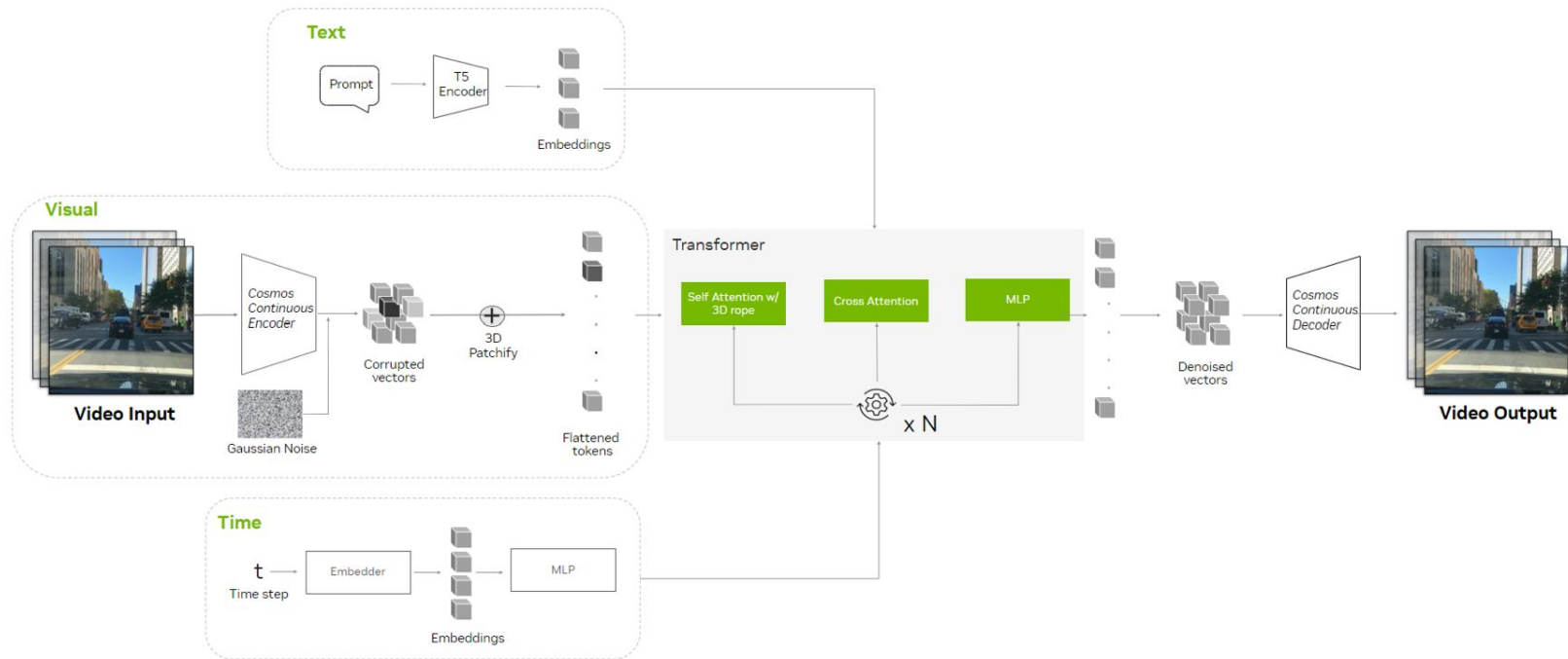


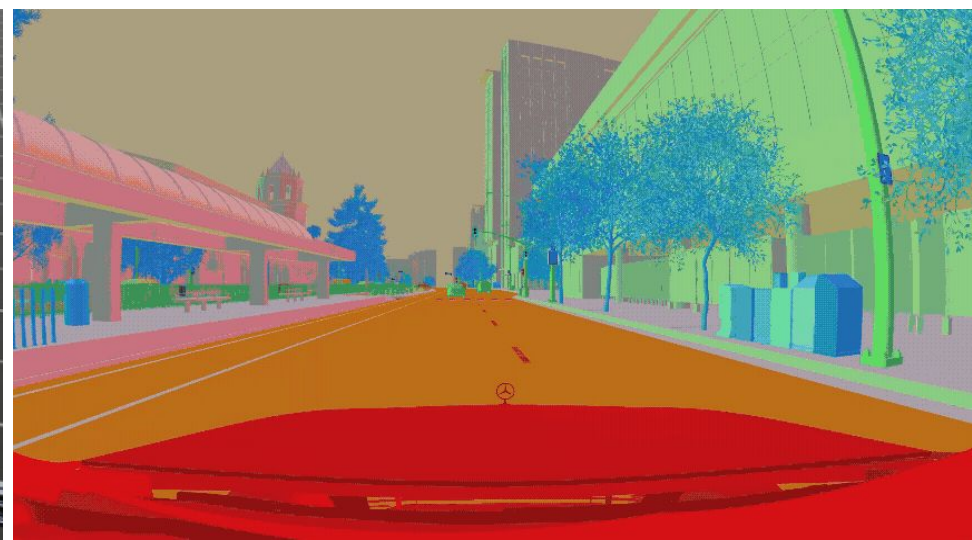
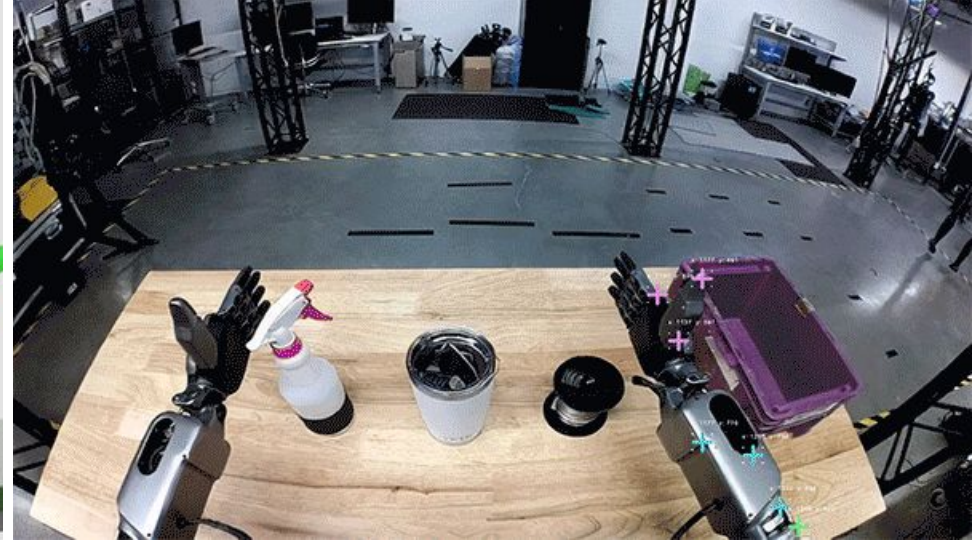
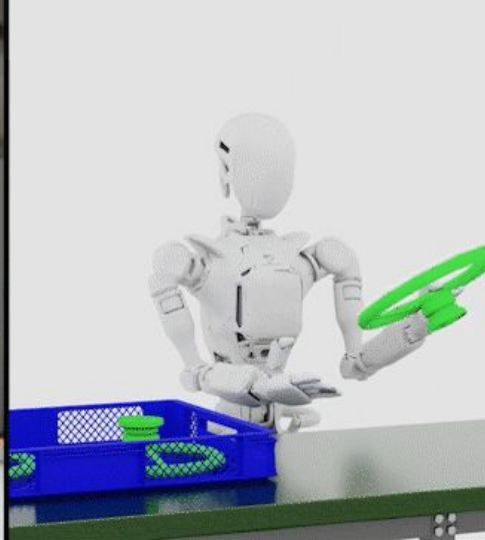
Link: <https://www.nvidia.com/en-us/ai/cosmos>

NVIDIA Cosmos

Diffusion Model

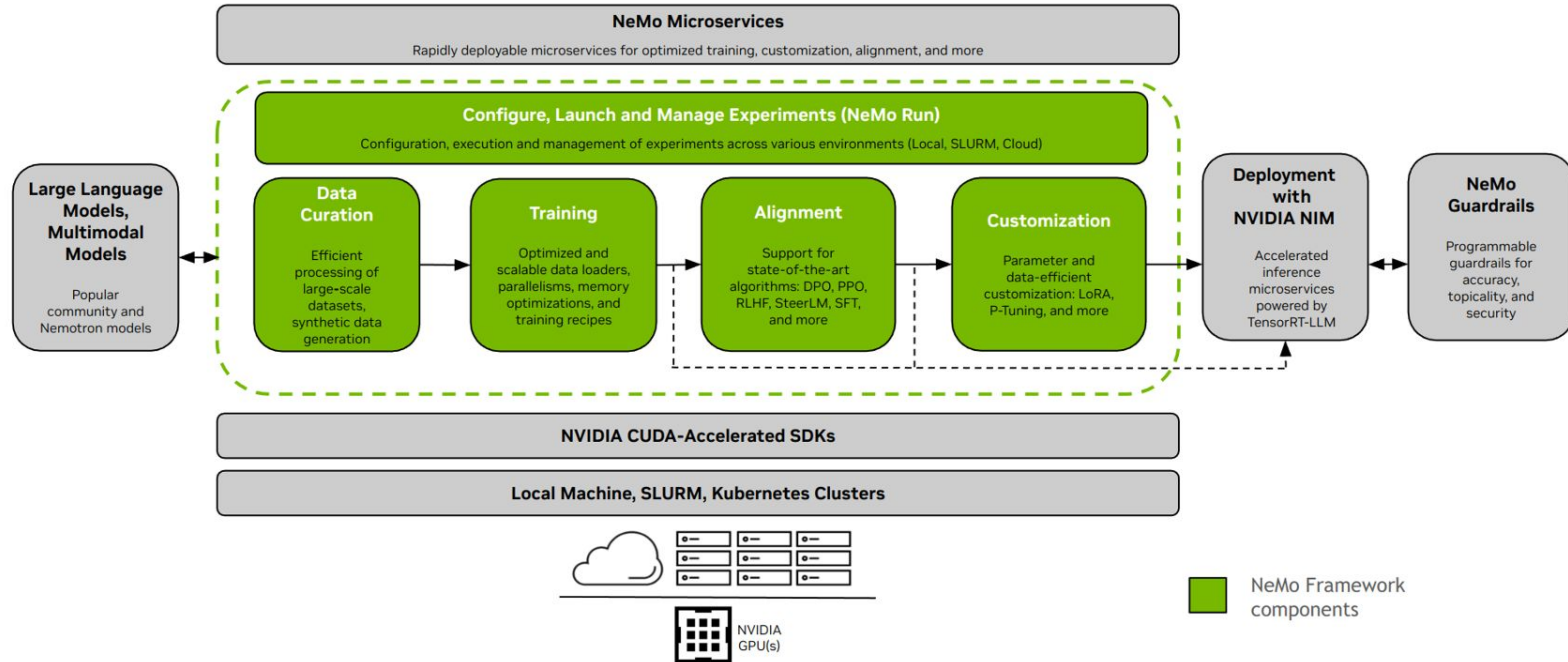
Diffusion models are popular for generating images, videos, and audio due to their ability to deconstruct training data and reconstruct it based on user input, producing high-quality, realistic outputs





NVIDIA NeMo Framework


NVIDIA NeMo Framework is a scalable and cloud-native generative AI framework



Link: <https://docs.nvidia.com/nemo-framework/user-guide/latest>

NVIDIA NeMo Framework

Open Source and Enterprise Containers

**NVIDIA-NeMo**
107 followers · Part of NVIDIA Corporation

README .md

NVIDIA NeMo Framework

NeMo Framework is NVIDIA's GPU accelerated, end-to-end training framework for large language models (LLMs), multi-modal models and speech models. It enables seamless scaling of training (both pretraining and post-training) workloads from single GPU to thousand-node clusters for both 🤖Hugging Face/PyTorch and Megatron models. This GitHub organization includes a suite of libraries and recipe collections to help users train models from end to end.

NeMo Framework is also a part of the NVIDIA NeMo software suite for managing the AI agent lifecycle.

Overview of Repos under NeMo Framework

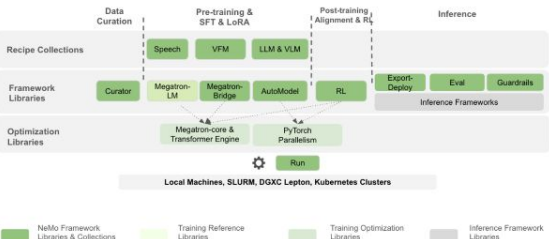


Figure 1. NeMo Framework Repo Overview

View as: Public


You are viewing the README and pinned repositories as a public user.

Top discussions this past month

Discussions are for sharing announcements, creating conversation in your community, answering questions, and more.

[Start a new discussion](#)

People




[View all](#)

Top languages

Python · HCL
Jupyter Notebook

Most used topics

Manage

**NVIDIA** NGC Catalog · CLASSIC

Welcome Guest

Explore Catalog

Containers · NeMo Framework

Get Container


Collections

Containers

Helm Charts

Models

Resources

**NVIDIA**
NEMO

Features

NVIDIA AI Enterprise Supported

Description

NVIDIA NeMo™ framework supports enterprise development of LLMs and generative AI models with automated data processing, model training techniques, and flexible deployment options.

Publisher

NVIDIA

Latest Tag

25.07.02

Modified

September 5, 2025

Compressed Size

16.23 GB

Multinode Support

Yes

Multi-Arch Support

Yes

25.07.02 (Latest) Security Scan

Overview

Tags

Layers

Security Scanning

Related Collections

What is the NeMo Framework Container?

NVIDIA NeMo™ is an end-to-end platform for development of custom generative AI models anywhere. NVIDIA NeMo framework is designed for enterprise development, it utilizes NVIDIA's state-of-the-art technology to facilitate a complete workflow from automated distributed data processing to training of large-scale bespoke models using sophisticated 3D parallelism techniques, and finally, deployment using retrieval-augmented generation for large-scale inference on an infrastructure of your choice, be it on-premises or in the cloud.

For enterprises running their business on AI, **NVIDIA AI Enterprise** provides a production-grade, secure, end-to-end software platform that includes NeMo as well as generative AI reference applications and enterprise support to streamline adoption. Now organizations can integrate AI into their operations, streamlining processes, enhancing decision-making capabilities, and ultimately driving greater value.

What You Get with NVIDIA NeMo Framework Container

At the heart of the NeMo framework lies the unification of distributed training and advanced parallelism. NeMo expertly uses GPU resources and memory across nodes, leading to groundbreaking efficiency gains. By dividing the model and training data, NeMo enables seamless multi-node and multi-GPU training, significantly reducing training time and enhancing overall productivity. A standout feature of NeMo is its incorporation of various parallelism and memory saving techniques:

Parallelism Techniques

Open Source on GitHub

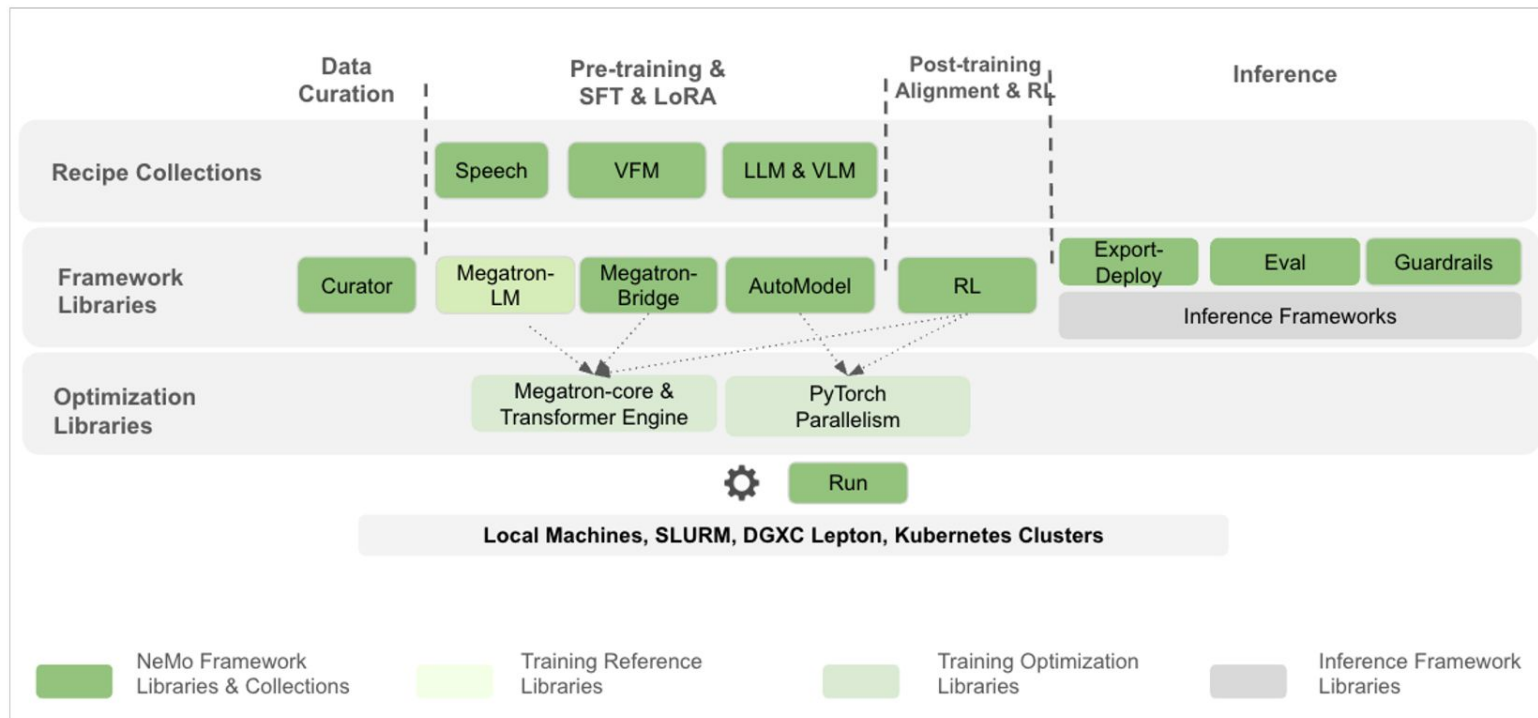
<https://github.com/NVIDIA-NeMo>

Enterprise Container on NGC

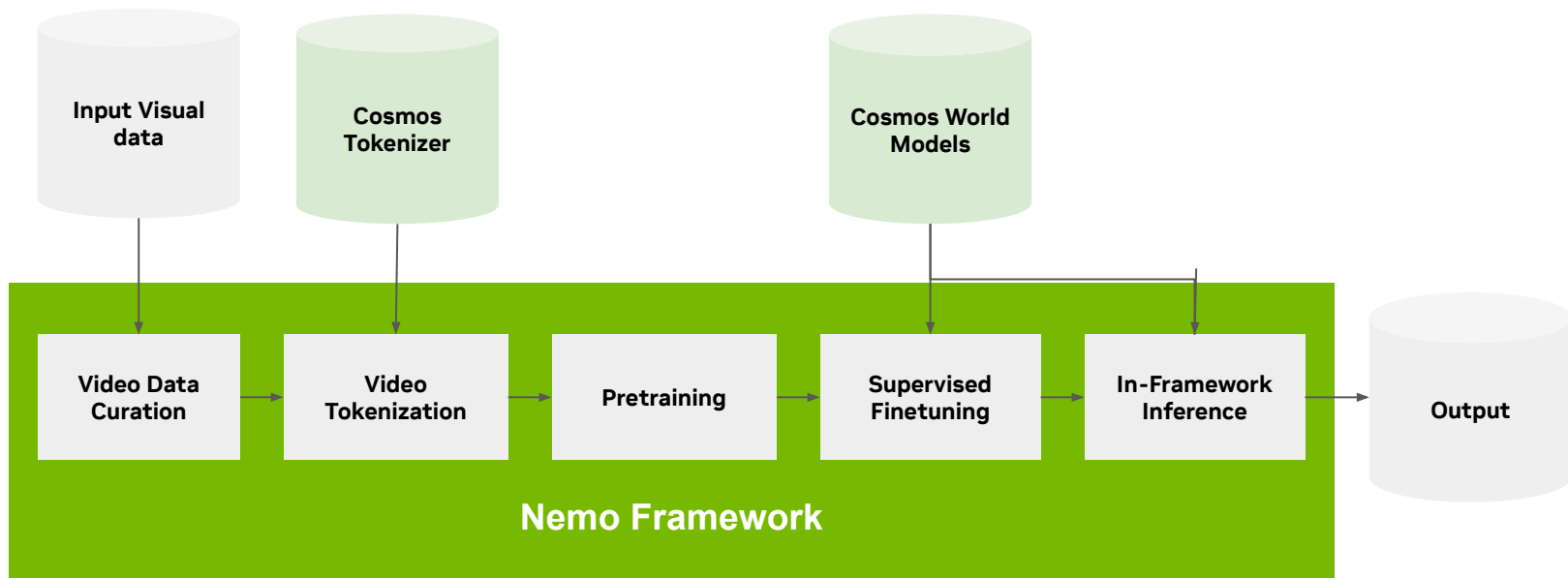
<https://catalog.ngc.nvidia.com/orgs/nvidia/containers/nemo>

NVIDIA NeMo Framework

Overview of Repos under NeMo Framework

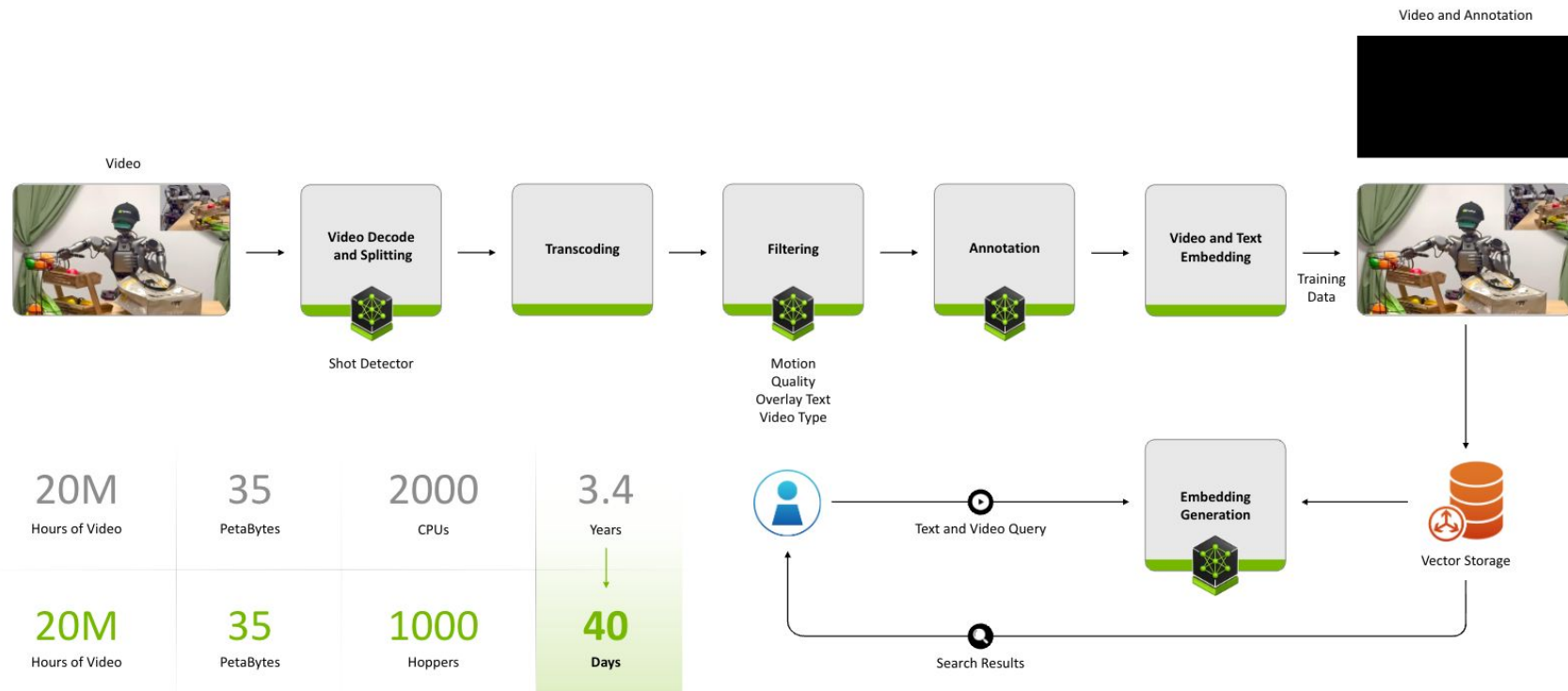


End to End Pipeline for Video Foundation Model Pretraining & Finetuning



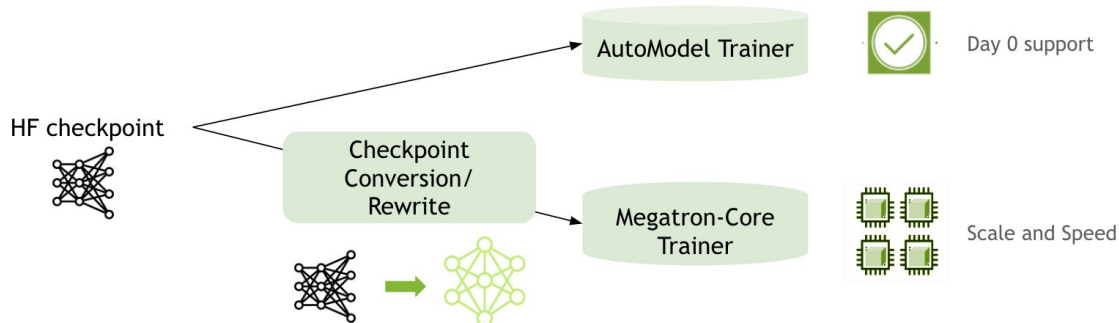
Data Processing and Curation Pipeline

Accelerate video processing from years to days



NVIDIA NeMo Framework Training

Training Workflows - Automodel & Megatron



| | Megatron-Core Backend | AutoModel Backend |
|--|---|---|
| Coverage | Most popular LLMs with recipes tuned by experts | All models supported in Hugging Face Text on Day-0 |
| Training Throughput Performance | Optimal Throughput with Megatron-Core kernels | Good Performance with liger kernels, cut cross entropy and PyTorch JIT |
| Scalability | Up to 1,000 GPUs with full 4-D parallelism (TP, PP, CP, EP) | Comparable scalability using PyTorch native TP, CP, and FSDP2 at slightly reduced training throughput |
| Inference Path | Export to TensorRT-LLM, vLLM, or directly to NVIDIA NIM | Export to vLLM |

NVIDIA NeMo Automodel

GPU-accelerated PyTorch training for Hugging Face models on Day-0

- HuggingFace Integration
- SFT (Supervised Fine-Tuning), and PEFT (Parameter Efficient Fine-Tuning)
- Native PyTorch support for models up to 70B parameters
- PyTorch native FSDP2, TP, CP, and SP for efficient training
- Sequence packing in both DTensor and MCore for huge training perf gains

1. Distributed Training Configuration

```
distributed:
  _target_: nemo_automodel.distributed.megatron_fsdp.MegatronFSDPManager
  dp_size: 8
  tp_size: 1
  cp_size: 1
```

2. LoRA Configuration

```
peft:
  peft_fn: nemo_automodel._peft.lora.apply_lora_to_linear_modules
  match_all_linear: True
  dim: 8
  alpha: 32
  use_triton: True
```

3. Vision-Language Model Fine-Tuning

```
model:
  _target_: nemo_automodel._transformers.NeMoAutoModelForImageTextToText.from_pretrained
  pretrained_model_name_or_path: Qwen/Qwen2.5-VL-3B-Instruct

processor:
  _target_: transformers.AutoProcessor.from_pretrained
  pretrained_model_name_or_path: Qwen/Qwen2.5-VL-3B-Instruct
  min_pixels: 200704
  max_pixels: 1003520
```

NVIDIA NeMo Megatron

State-of-the-art training throughput for top models

Megatron Bridge

Bidirectional converter for interoperability between Hugging Face and Megatron

- Seamless bidirectional conversion between Hugging Face and Megatron
- Lightweight custom training loop to configure custom logic in data loading, distributed training, checkpointing, evaluation and logging
- SFT & PEFT implementation tailored for Megatron-based models
- Production-ready recipes for popular models

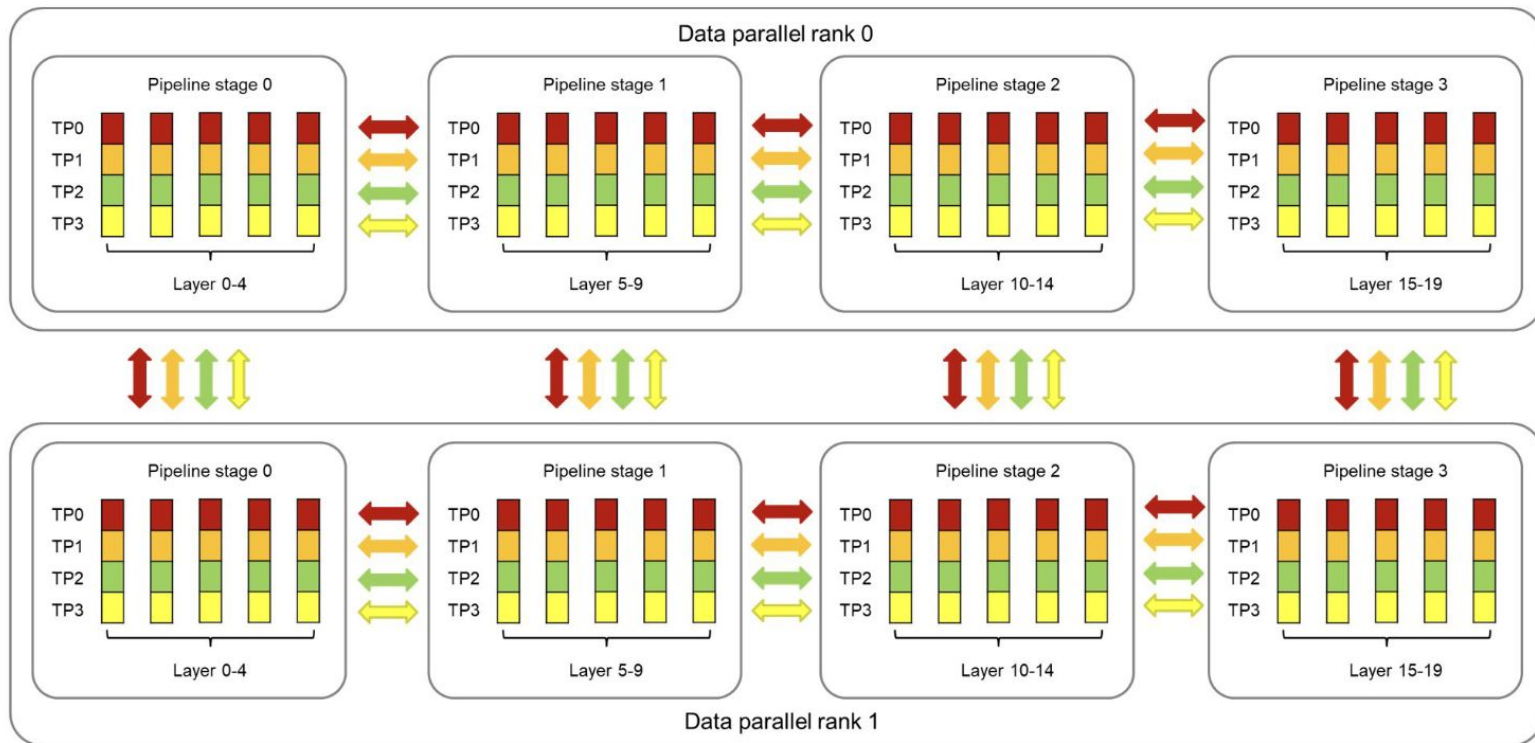
Megatron Core

Train top generative AI models with unparalleled speed at scale across thousands of GPUs

- World-leading training speed and scalability
- Advanced model parallelism techniques: tensor (TP), sequence (SP), pipeline (PP), context (CP), and MoE expert (EP) parallelism
- Automatic restart, fault/hang detection, and fast distributed checkpointing
- FP8 mixed precision & memory-saving functionalities

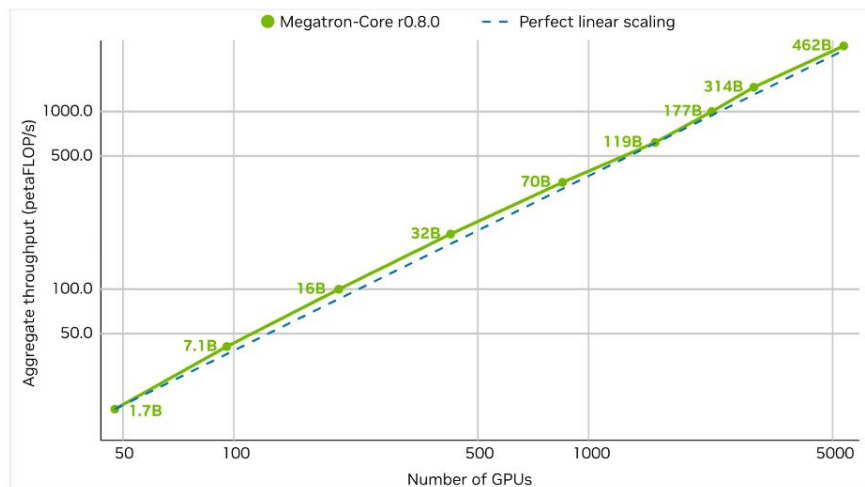
Different Parallelism - An Example

Put it all together



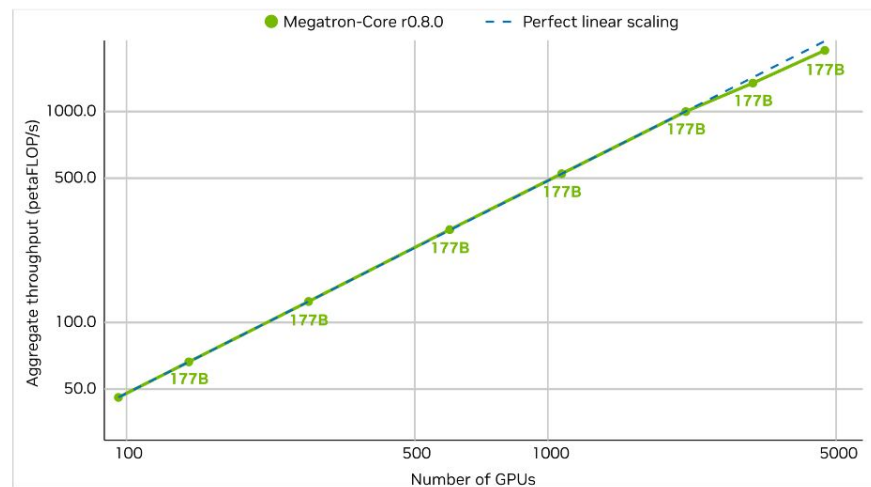
NVIDIA Megatron Core

Train generative AI models from scratch at scale



Aggregate Throughput (Weak Scaling)

With GPT models ranging from 2 billion to 462 billion parameters, Megatron-Core demonstrates superlinear scaling up to 6144 H100 GPUs.

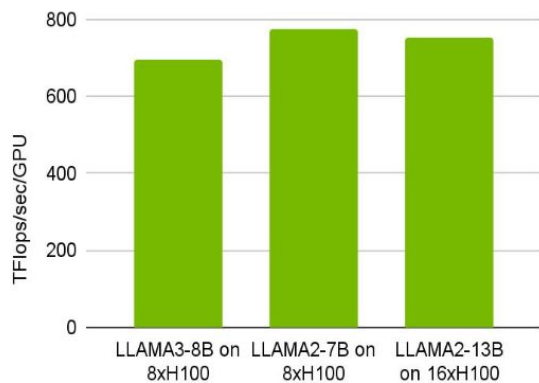


Aggregate Throughput (Strong Scaling)

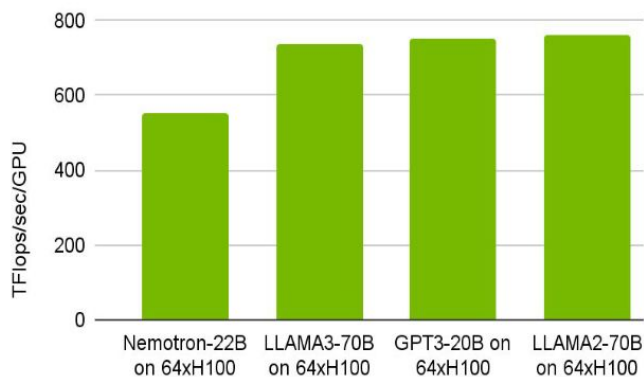
With a 177 billion parameter GPT-3 model using the same batch size of 1152 sequences throughout, Megatron-Core demonstrates near linear scaling from 96 to 4608 H100 GPUs.

Benchmark - Training Throughput

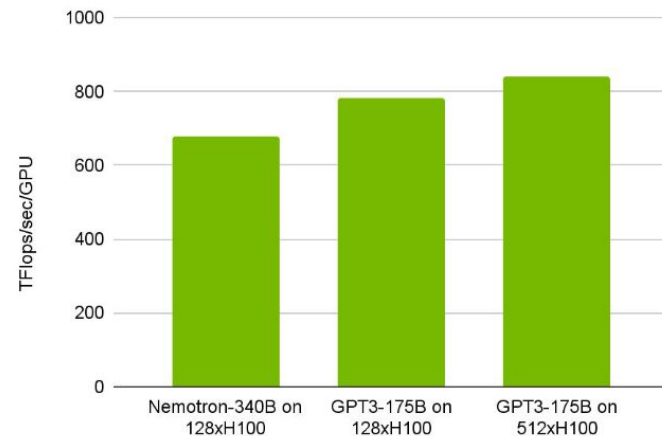
Small Models (<20B) on 8~16xH100



Medium Models (20B~70B) on 64xH100



Large Models (175B~340B) on 128~512xH100



NVIDIA NeMo RL

A Scalable and Efficient Post-Training Library

- **Integration & Customization**

- Seamless integration with Hugging Face
- Flexibility with a modular design that allows easy integration and customization

- **High Performance & Scalability**

- High-performance implementation with Megatron Core
- Fast Generation - vLLM backend for optimized inference

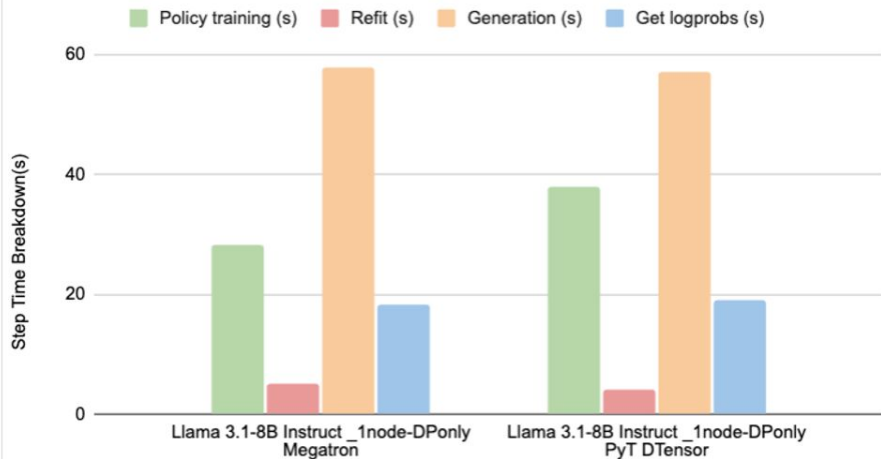
- **Advanced Training Techniques**

- Advanced Parallelism and Ray-based infrastructure - PyTorch native FSDP2, TP, CP, and SP for efficient training
- Learning Algorithms - GRPO (Group Relative Policy Optimization), SFT (Supervised Fine-Tuning), and DPO (Direct Preference Optimization)
- Multi-Turn RL - Multi-turn generation and training for RL with tool use, games, etc
- Sequence Packing - Sequence packing in both DTensor and MCore for huge training perf gains

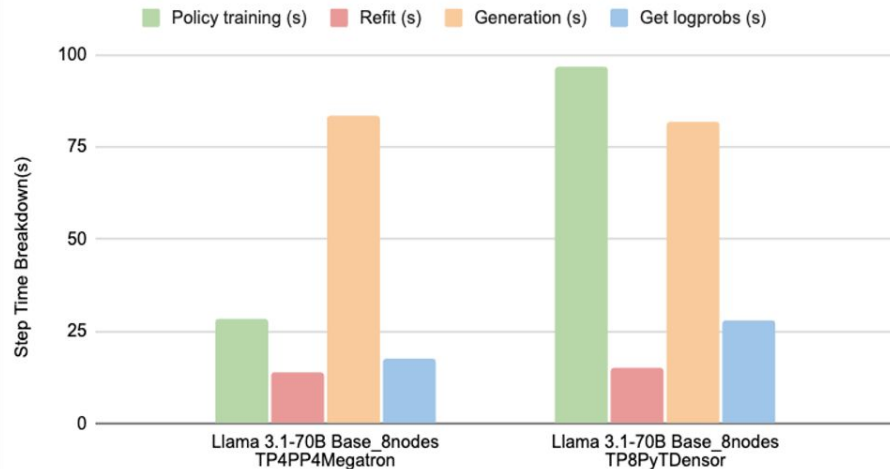
NVIDIA NeMo RL

Megatron vs PyTorch DTensor Backends

Llama 3.1-8B@4k Seq: Megatron-core vs PyT DTensor backends



Llama 3.1-70B@4k Seq: Megatron-core vs PyT DTensor backends



NVIDIA NeMo Framework

Enterprise-Grade Training Framework

Performance & Scalability

- More than 800 TFLOPs/sec/GPU
- 4D Parallelism
- Trained over 16k+ cluster size
- Supports 1M+ sequence length
- GPU-accelerated data curation
- Fault tolerance integration

Customization & Modularity

- 23 model families Incl LLM, SSMs, MOEs, SD, VLMs, VFM
- Streamlined pretraining & fine-tuning pipelines
- Python-based configuration
- Modular architecture
- Ready-to-use scripts & models

Enterprise Support & Security Available

- E2E workflow from Data Curation, SFT, PEFT, RL, Eval Export & Deploy
- Backed by NVIDIA Enterprise Support
- Security hardened
- Long term support

Thank you!

Useful Links to Get Started

Cosmos Model Collection on Hugging Face:

<https://huggingface.co/collections/nvidia/cosmos-6751e884dc10e013a0a0d8e6>

Cosmos Developer Guide:

<https://docs.nvidia.com/cosmos/latest/index.html#>

NeMo Open Source on GitHub:

<https://github.com/NVIDIA-NeMo>

NeMo Enterprise on NGC:

<https://catalog.ngc.nvidia.com/orgs/nvidia/containers/nemo>

NeMo User Guide:

<https://docs.nvidia.com/nemo-framework/user-guide/latest/overview.html>