## [nvidia](https://build.nvidia.com/nvidia)

# llama-3.3-nemotron-super-49b-v1

Run Anywhere

High efficiency model with leading accuracy for reasoning, tool calling, chat, and instruction following.

[advanced reasoning](https://build.nvidia.com/search?q=advanced+reasoning)[function calling](https://build.nvidia.com/search?q=function+calling)[instruction following](https://build.nvidia.com/search?q=instruction+following)[math](https://build.nvidia.com/search?q=math)

[Get API Key](https://build.nvidia.com/nvidia/llama-3_3-nemotron-super-49b-v1/modelcard?signin=true&integrate_nim=true&hosted_api=true)

ExperienceModel CardDeploy

[API Reference](https://docs.api.nvidia.com/nim/reference/nvidia-llama-3_3-nemotron-super-49b-v1)

[Accelerated by DGX Cloud](https://docs.api.nvidia.com/nim/reference/nvidia-llama-3_3-nemotron-super-49b-v1)

OverviewModelCard++

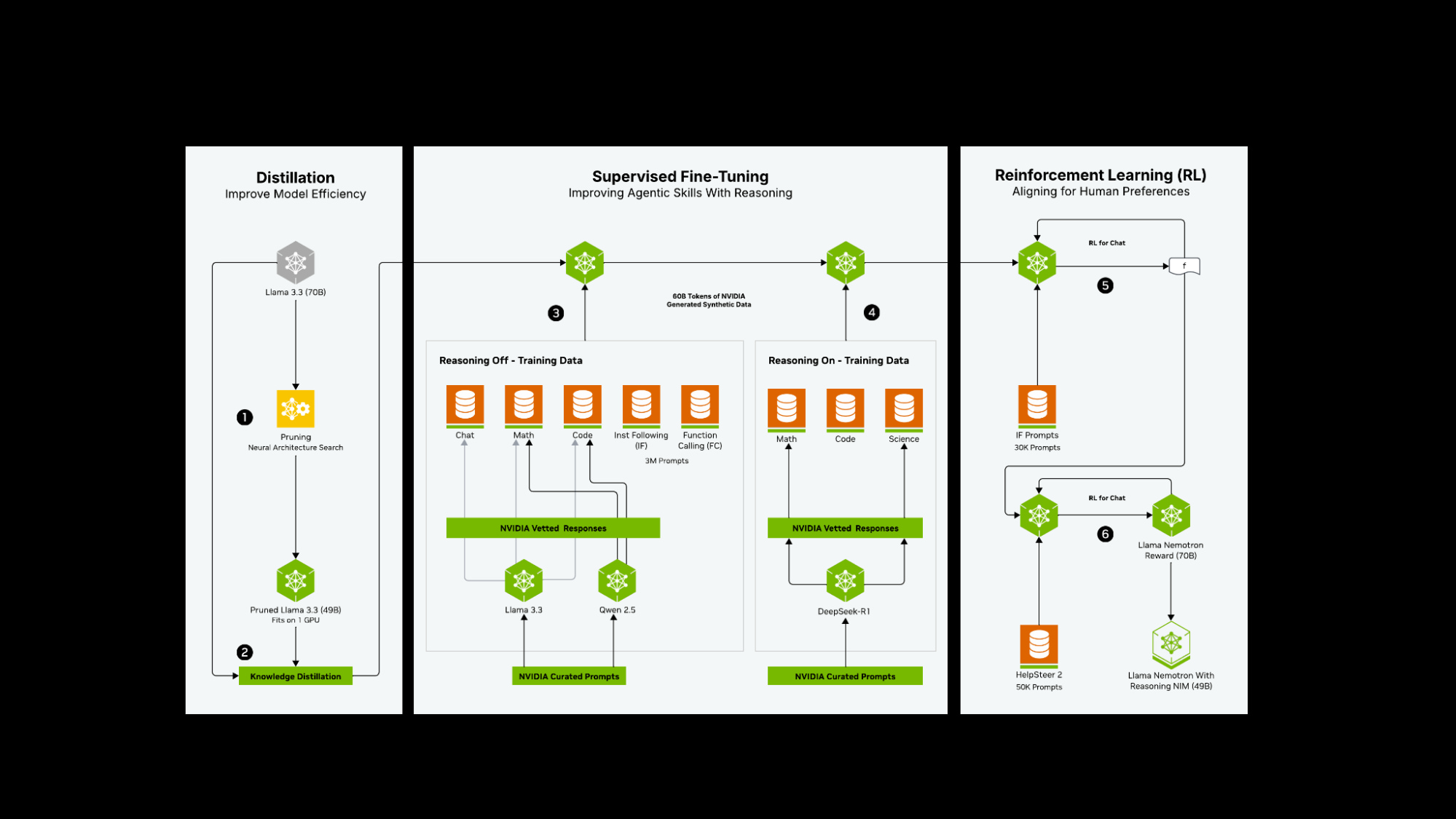
# Llama-3.3-Nemotron-Super-49B-v1

## Model Overview

Llama-3.3-Nemotron-Super-49B-v1 is a large language model (LLM) which is a derivative of Meta Llama-3.3-70B-Instruct (AKA the reference model). It is a reasoning model that is post trained for reasoning, human chat preferences, and tasks, such as RAG and tool calling. The model supports a context length of 128K tokens.

Llama-3.3-Nemotron-Super-49B-v1 is a model which offers a great tradeoff between model accuracy and efficiency. Efficiency (throughput) directly translates to savings. Using a novel Neural Architecture Search (NAS) approach, we greatly reduce the model’s memory footprint, enabling larger workloads, as well as fitting the model on a single GPU at high workloads (H200). This NAS approach enables the selection of a desired point in the accuracy-efficiency tradeoff.

The model underwent a multi-phase post-training process to enhance both its reasoning and non-reasoning capabilities. This includes a supervised fine-tuning stage for Math, Code, Reasoning, and Tool Calling as well as multiple reinforcement learning (RL) stages using REINFORCE (RLOO) and Online Reward-aware Preference Optimization (RPO) algorithms for both chat and instruction-following. The final model checkpoint is obtained after merging the final SFT and Online RPO checkpoints. For more details on how the model was trained, please see [this blog](https://developer.nvidia.com/blog/build-enterprise-ai-agents-with-advanced-open-nvidia-llama-nemotron-reasoning-models/).



This model is part of the Llama Nemotron Collection. You can find the other model(s) in this family here: [Llama-3\_1-Nemotron-Nano-8B-v1](https://build.nvidia.com/nvidia/llama-3_1-nemotron-nano-8b-v1)

This model is ready for commercial use.

## License/Terms of Use

GOVERNING TERMS: Your use of this model is governed by the [NVIDIA Open Model License.](https://www.nvidia.com/en-us/agreements/enterprise-software/nvidia-open-model-license/) Additional Information: [Llama 3.3 Community License Agreement](https://www.llama.com/llama3_3/license/). Built with Llama.

Model Developer: NVIDIA

Model Dates: Trained between November 2024 and February 2025

Data Freshness: The pretraining data has a cutoff of 2023 per Meta Llama 3.3 70B

### Use Case:

Developers designing AI Agent systems, chatbots, RAG systems, and other AI-powered applications. Also suitable for typical instruction-following tasks.

### Release Date:

3/18/2025

## References

* [2502.00203] [Reward-aware Preference Optimization: A Unified Mathematical Framework for Model Alignment](https://arxiv.org/abs/2502.00203)

## Model Architecture

Architecture Type: Dense decoder-only Transformer model  
Network Architecture: Llama 3.3 70B Instruct, customized through Neural Architecture Search (NAS)

The model is a derivative of Meta’s Llama-3.3-70B-Instruct, using Neural Architecture Search (NAS). The NAS algorithm results in non-standard and non-repetitive blocks. This includes the following: Skip attention: In some blocks, the attention is skipped entirely, or replaced with a single linear layer. Variable FFN: The expansion/compression ratio in the FFN layer is different between blocks.

We utilize a block-wise distillation of the reference model, where for each block we create multiple variants providing different tradeoffs of quality vs. computational complexity, discussed in more depth below. We then search over the blocks to create a model which meets the required throughput and memory (optimized for a single H100-80GB GPU) while minimizing the quality degradation. The model then undergoes knowledge distillation (KD), with a focus on English single and multi-turn chat use-cases. The KD step included 40 billion tokens consisting of a mixture of 3 datasets - FineWeb, Buzz-V1.2 and Dolma.

## Intended use

Llama-3.3-Nemotron-Super-49B-v1 is a general purpose reasoning and chat model intended to be used in English and coding languages. Other non-English languages (German, French, Italian, Portuguese, Hindi, Spanish, and Thai) are also supported.

## Input

* Input Type: Text
* Input Format: String
* Input Parameters: One-Dimensional (1D)
* Other Properties Related to Input: Context length up to 131,072 tokens

## Output

* Output Type: Text
* Output Format: String
* Output Parameters: One-Dimensional (1D)
* Other Properties Related to Output: Context length up to 131,072 tokens

## Model Version

1.0 (3/18/2025)

## Software Integration

* Runtime Engine: Transformers
* Recommended Hardware Microarchitecture Compatibility:
  + NVIDIA Hopper
  + NVIDIA Ampere

## Quick Start and Usage Recommendations:

1. Reasoning mode (ON/OFF) is controlled via the system prompt, which must be set as shown in the example below. All instructions should be contained within the user prompt
2. We recommend setting temperature to 0.6, and Top P to 0.95 for Reasoning ON mode
3. We recommend using greedy decoding for Reasoning OFF mode
4. We have provided a list of prompts to use for evaluation for each benchmark where a specific template is required
5. The model will include <think></think> if no reasoning was necessary in Reasoning ON model, this is expected behaviour

You can try this model out through the preview API, using this link: [Llama-3\_3-Nemotron-Super-49B-v1](https://build.nvidia.com/nvidia/llama-3_3-nemotron-super-49b-v1).

## Inference:

Engine: Transformers  
Test Hardware:

* FP8: 1x NVIDIA H100-80GB GPU (Coming Soon!)
* BF16:
  + 2x NVIDIA H100-80GB GPUs
  + 2x NVIDIA A100-80GB GPUs

[Preferred/Supported] Operating System(s): Linux

## Training Datasets

A large variety of training data was used for the knowledge distillation phase before post-training pipeline, 3 of which included: FineWeb, Buzz-V1.2, and Dolma.

The data for the multi-stage post-training phases for improvements in Code, Math, and Reasoning is a compilation of SFT and RL data that supports improvements of math, code, general reasoning, and instruction following capabilities of the original Llama instruct model.

In conjunction with this model release, NVIDIA has released 30M samples of post-training data, as public and permissive. [Llama-Nemotron-Post-Training-Dataset-v1](https://huggingface.co/datasets/nvidia/Llama-Nemotron-Post-Training-Dataset-v1)

Distribution of the domains is as follows:

| Category | Value |
| --- | --- |
| math | 19,840,970 |
| code | 9,612,677 |
| science | 708,920 |
| instruction following | 56,339 |
| chat | 39,792 |
| safety | 31,426 |

Prompts have been sourced from either public and open corpus or synthetically generated. Responses were synthetically generated by a variety of models, with some prompts containing responses for both reasoning on and off modes, to train the model to distinguish between two modes.

Models that were used in the creation of this dataset:

* Llama-3.3-70B-Instruct
* Llama-3.1-Nemotron-70B-Instruct
* Llama-3.3-Nemotron-70B-Feedback/Edit/Select
* Mixtral-8x22B-Instruct-v0.1
* DeepSeek-R1
* Qwen-2.5-Math-7B-Instruct
* Qwen-2.5-Coder-32B-Instruct
* Qwen-2.5-72B-Instruct
* Qwen-2.5-32B-Instruct

Data Collection for Training Datasets: Hybrid: Automated, Human, Synthetic

Data Labeling for Training Datasets: Hybrid: Automated, Human, Synthetic

## Evaluation Datasets

We used the datasets listed below to evaluate Llama-3.3-Nemotron-Super-49B-v1.

Data Collection for Evaluation Datasets: Hybrid: Human/Synthetic

Data Labeling for Evaluation Datasets: Hybrid: Human/Synthetic/Automatic

## Evaluation Results

These results contain both Reasoning On, and Reasoning Off. We recommend using temperature=0.6, top\_p=0.95 for Reasoning On mode, and greedy decoding for Reasoning Off mode. All evaluations are done with 32k sequence length. We run the benchmarks up to 16 times and average the scores to be more accurate.

NOTE: Where applicable, a Prompt Template will be provided. While completing benchmarks, please ensure that you are parsing for the correct output format as per the provided prompt in order to reproduce the benchmarks seen below.

### Arena-Hard

| Reasoning Mode | Score |
| --- | --- |
| Reasoning Off | 88.3 |

### MATH500

| Reasoning Mode | pass@1 |
| --- | --- |
| Reasoning Off | 74.0 |
| Reasoning On | 96.6 |

User Prompt Template:

Copy Code

"Below is a math question. I want you to reason through the steps and then give a final answer. Your final answer should be in \boxed{}.\nQuestion: {question}"

### AIME25

| Reasoning Mode | pass@1 |
| --- | --- |
| Reasoning Off | 13.33 |
| Reasoning On | 58.4 |

User Prompt Template:

Copy Code

"Below is a math question. I want you to reason through the steps and then give a final answer. Your final answer should be in \boxed{}.\nQuestion: {question}"

### GPQA

| Reasoning Mode | pass@1 |
| --- | --- |
| Reasoning Off | 50 |
| Reasoning On | 66.67 |

User Prompt Template:

Copy Code

"What is the correct answer to this question: {question}\nChoices:\nA. {option\_A}\nB. {option\_B}\nC. {option\_C}\nD. {option\_D}\nLet's think step by step, and put the final answer (should be a single letter A, B, C, or D) into a \boxed{}"

### IFEval

| Reasoning Mode | Strict |
| --- | --- |
| Reasoning Off | 89.21 |

### BFCL V2 Live

| Reasoning Mode | Score |
| --- | --- |
| Reasoning Off | 73.7 |

User Prompt Template:

Copy Code

You are an expert in composing functions. You are given a question and a set of possible functions.

Based on the question, you will need to make one or more function/tool calls to achieve the purpose.

If none of the function can be used, point it out. If the given question lacks the parameters required by the function,

also point it out. You should only return the function call in tools call sections.

If you decide to invoke any of the function(s), you MUST put it in the format of <TOOLCALL>[func\_name1(params\_name1=params\_value1, params\_name2=params\_value2...), func\_name2(params)]</TOOLCALL>

You SHOULD NOT include any other text in the response.

Here is a list of functions in JSON format that you can invoke.

<AVAILABLE\_TOOLS>{functions}</AVAILABLE\_TOOLS>

{user\_prompt}

### MBPP 0-shot

| Reasoning Mode | pass@1 |
| --- | --- |
| Reasoning Off | 84.9 |
| Reasoning On | 91.3 |

User Prompt Template:

Copy Code

You are an exceptionally intelligent coding assistant that consistently delivers accurate and reliable responses to user instructions.

@@ Instruction

Here is the given problem and test examples:

{prompt}

Please use the python programming language to solve this problem.

Please make sure that your code includes the functions from the test samples and that the input and output formats of these functions match the test samples.

Please return all completed codes in one code block.

This code block should be in the following format:

```python

# Your codes here

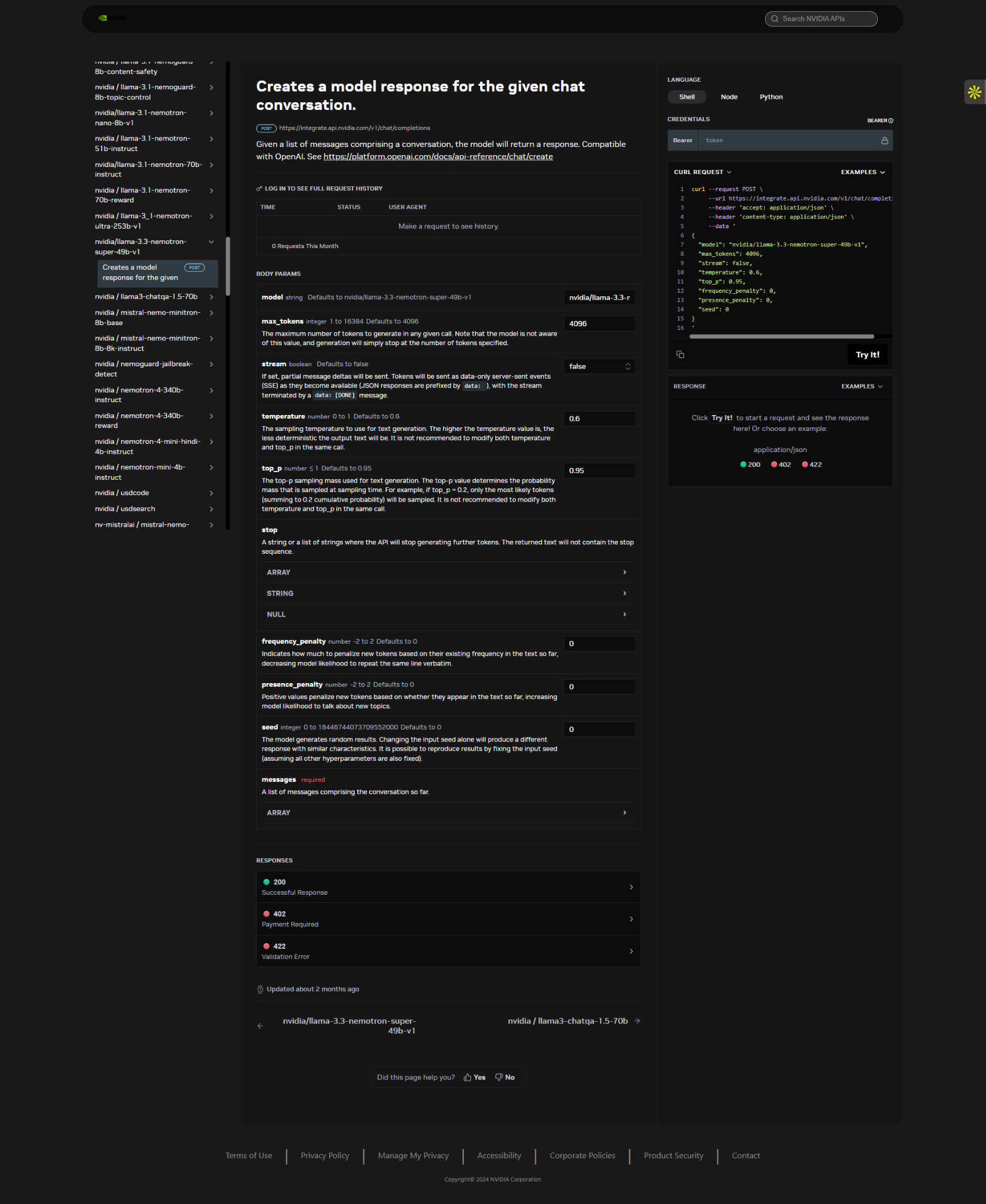
```

### MT-Bench

| Reasoning Mode | Score |
| --- | --- |
| Reasoning Off | 9.17 |

## Ethical Considerations:

NVIDIA believes Trustworthy AI is a shared responsibility and we have established policies and practices to enable development for a wide array of AI applications. When downloaded or used in accordance with our terms of service, developers should work with their internal model team to ensure this model meets requirements for the relevant industry and use case and addresses unforeseen product misuse.

For more detailed information on ethical considerations for this model, please see the Model Card++ Explainability, Bias, Safety & Security, and Privacy Subcards, which you can find by clicking the ModelCard++ tab above, next to Overview. Please report security vulnerabilities or NVIDIA AI Concerns [here](https://www.nvidia.com/en-us/support/submit-security-vulnerability/).  
  
  
  
import requests

url = "https://integrate.api.nvidia.com/v1/chat/completions"

payload = {

"model": "nvidia/llama-3.3-nemotron-super-49b-v1",

"max\_tokens": 4096,

"stream": False,

"temperature": 0.6,

"top\_p": 0.95,

"frequency\_penalty": 0,

"presence\_penalty": 0,

"seed": 0

}

headers = {

"accept": "application/json",

"content-type": "application/json"

}

response = requests.post(url, json=payload, headers=headers)

print(response.text)

—-------------------------------------------  
  
from openai import OpenAI

client = OpenAI(

base\_url = "https://integrate.api.nvidia.com/v1",

api\_key = "$API\_KEY\_REQUIRED\_IF\_EXECUTING\_OUTSIDE\_NGC"

)

completion = client.chat.completions.create(

model="nvidia/llama-3.3-nemotron-super-49b-v1",

messages=[{"role":"system","content":"detailed thinking on"}],

temperature=0.6,

top\_p=0.95,

max\_tokens=4096,

frequency\_penalty=0,

presence\_penalty=0,

stream=True

)

for chunk in completion:

if chunk.choices[0].delta.content is not None:

print(chunk.choices[0].delta.content, end="")