Unlocking AI Performance with NeMo Curator: Scalable Data Processing for LLMs

Ayush Maheshwari
Sr. Solutions Architect, NVIDIA

https://github.com/ayushbits/llm-development

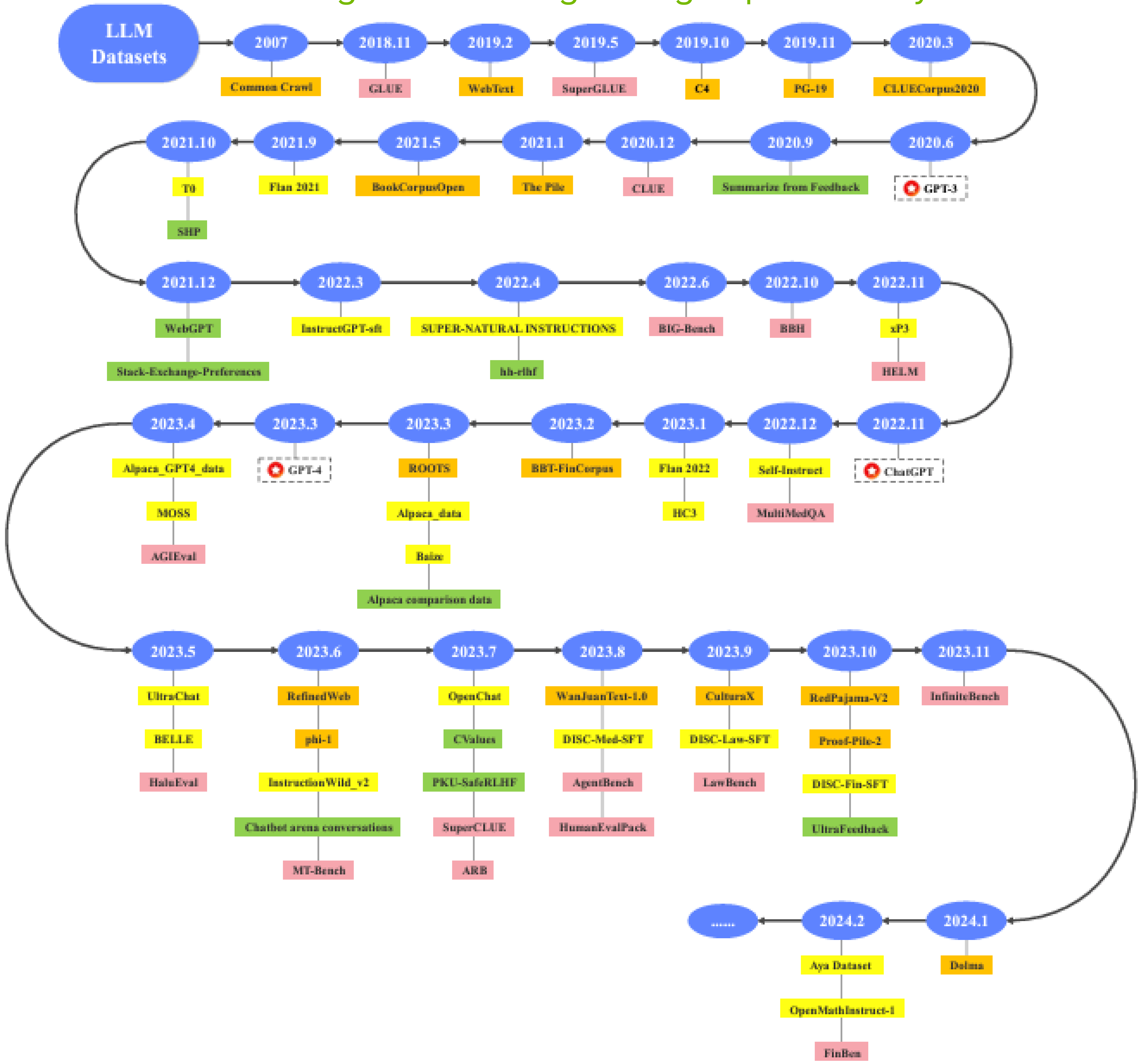
ayushbits.github.io

Sessions

1. Cluster health-check using NCCL, MLPerf, HPL **(1 hour) - Completed**
   a) Understand the hardware and its performance on multiple GPUs.
   b) Ensure that your training performance aligns with the h/w benchmarks
   c) Evaluate the cluster to ensure platform fits within your needs.
2. Large scale data curation for LLM training **(1 hour) - Today**
   a) Deep-dive into aspects of data curation
   b) Mixed-precision training
3. Distributed and stable LLM training on a large-scale cluster **(1.5 hour)**
   a) Parallelism techniques
   b) Frameworks and wrappers
   c) Recipes and best practices
4. Post-training and evaluation of pre-trained LLM **(1 hour)**
   a) Sync between training data and expected performance
   b) Algorithms and frameworks
5. Fine-tuning and deployment **(1 hour)**
   a) Dynamic and static batching, state management, inference server
   b) Best practices for optimizing model

LLMs Are Trained on Internet Scale Data

Data generated is growing exponentially



Training corpa

Instruct fine tuning

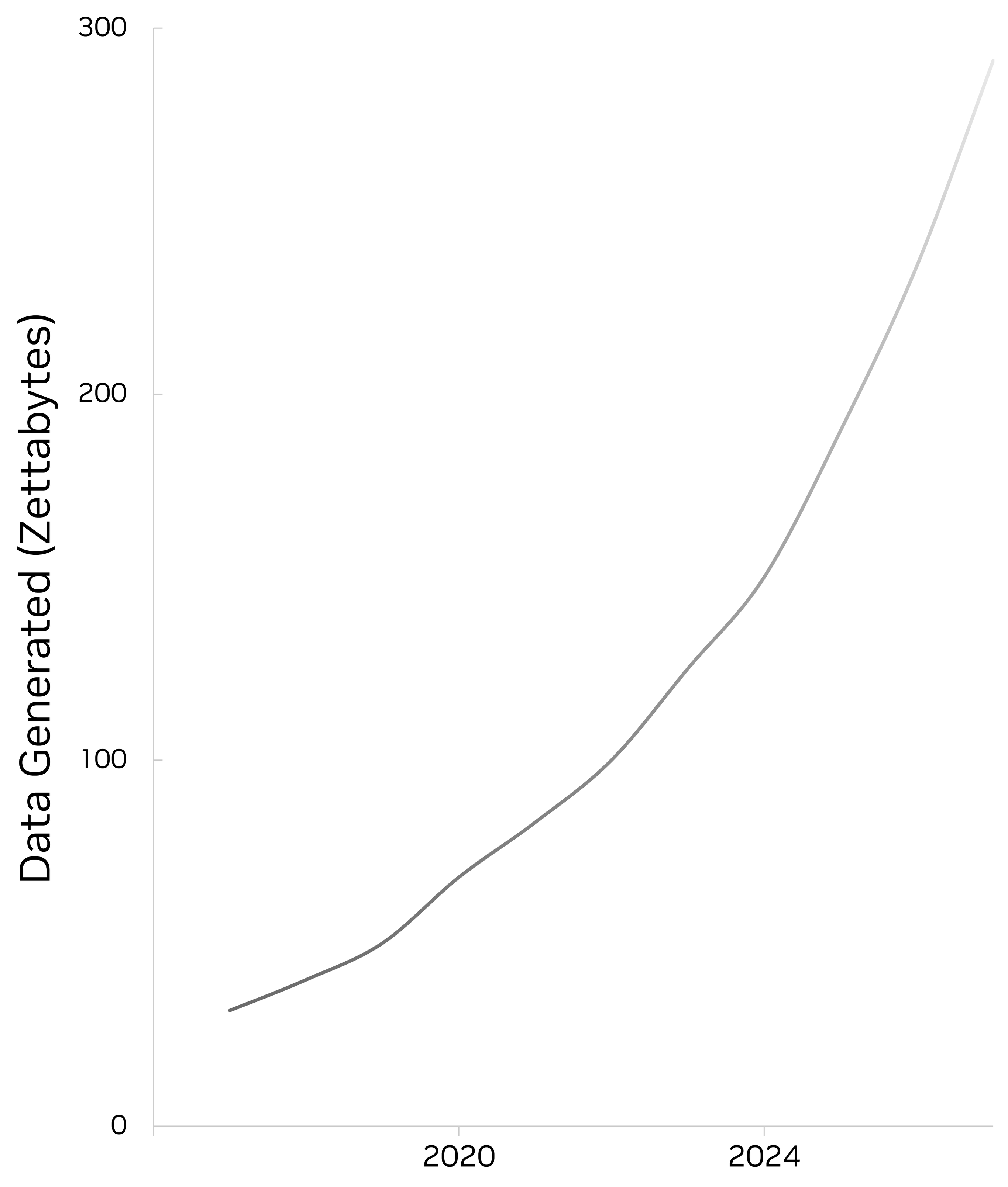
Preference dataset

Evaluation dataset

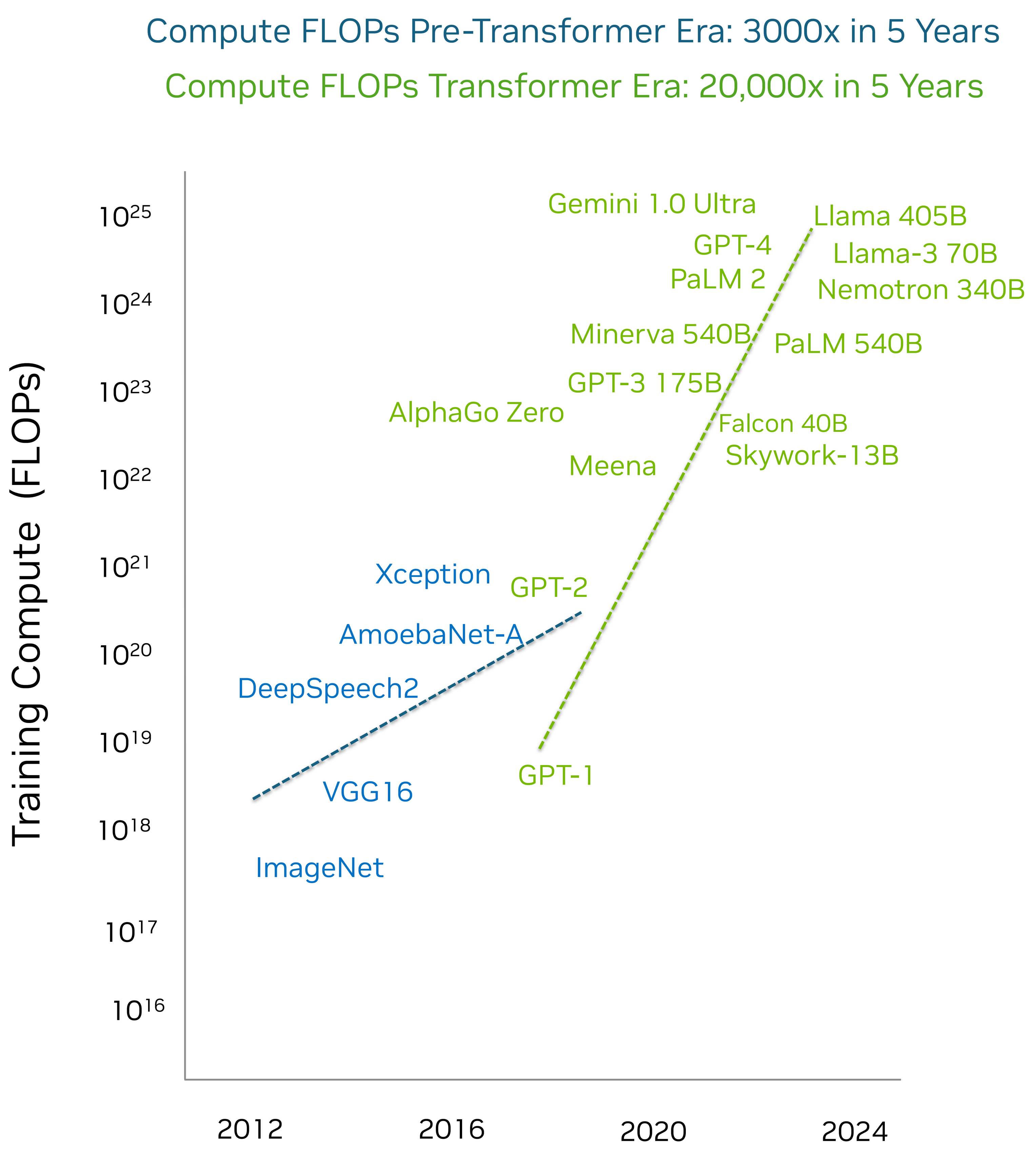
https://arxiv.org/abs/2402.18041

Data Processing for LLMs Needs Accelerated Computing

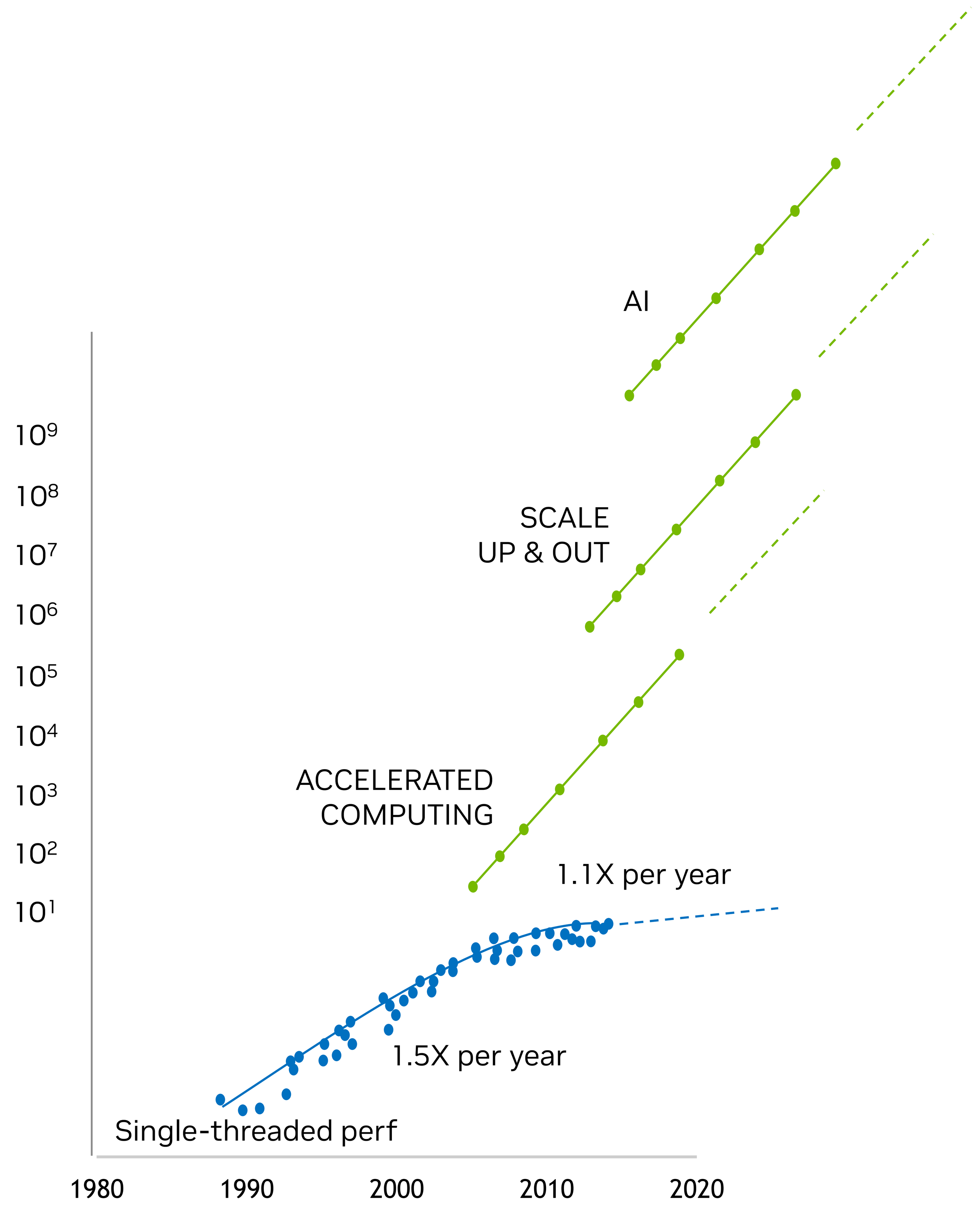
Petabytes of Data Generated Yearly



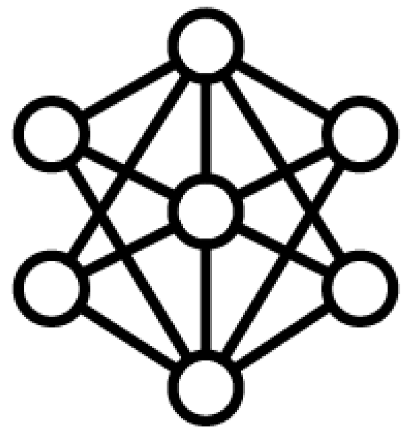
LLMs Trained on Internet Scale Data

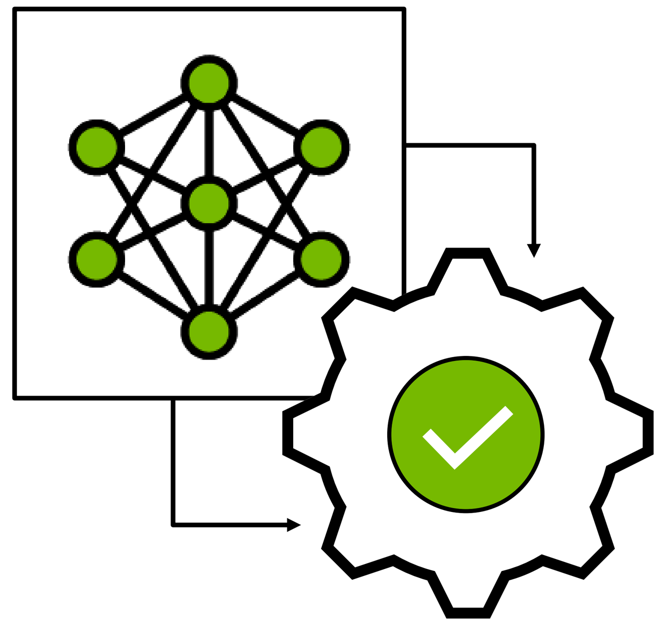


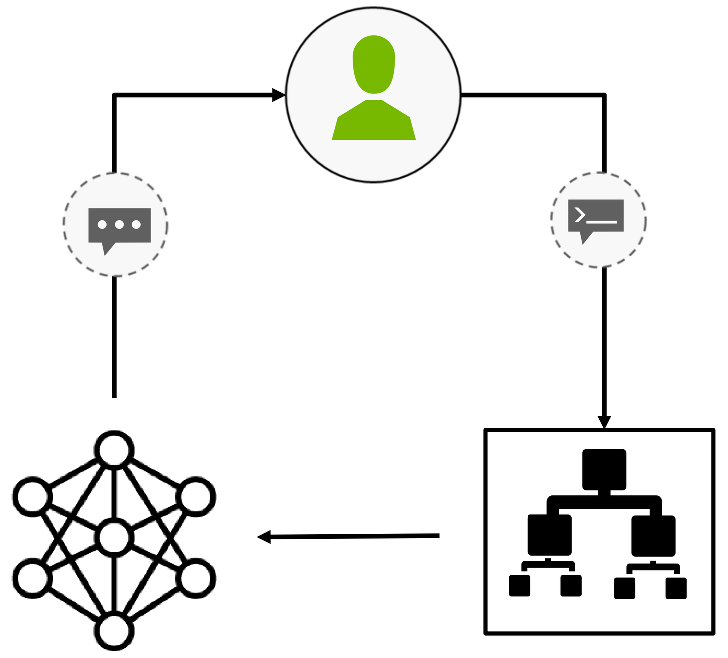
Moore's Law Has Ended



Data Processing for Different LLM Needs







	Training Foundation Model	Fine-Tuning Foundation Model	Retrieval Augmented Generation (RAG)
Data Size	TB and PB	GBs	GBs
Compute Scale	Supercomputer	Single-node	Single GPU
Frequency	One-time	Iterative	Iterative & Continuous

Challenges with Existing Solutions for Training Foundation Models

Inefficiencies lead to higher TCO and slower time to market

Longer Processing Time

Un-optimized Models

Un-optimized Pipelines

Knowledge & Expertise

NeMo Curator - Overview

Scalable, configurable pipelines to curate text, image and video datasets lead to more accurate applications

Higher Accuracy

Improve accuracy with less data and less training compute

Faster Processing

GPU acceleration with RAPIDS for **Dedupe (Exact, Fuzzy, Semantic)** and **Quality Classifier Models**

Scalability

Up to **100+ PB data** by scaling across multiple nodes

Classifier Models

State-of-the-Art quality classifier NIM microservice for safety, content, and diversity

Deploy anywhere

Python APIs in a customizable and modular OSS library, runnable across CSP and On-prem

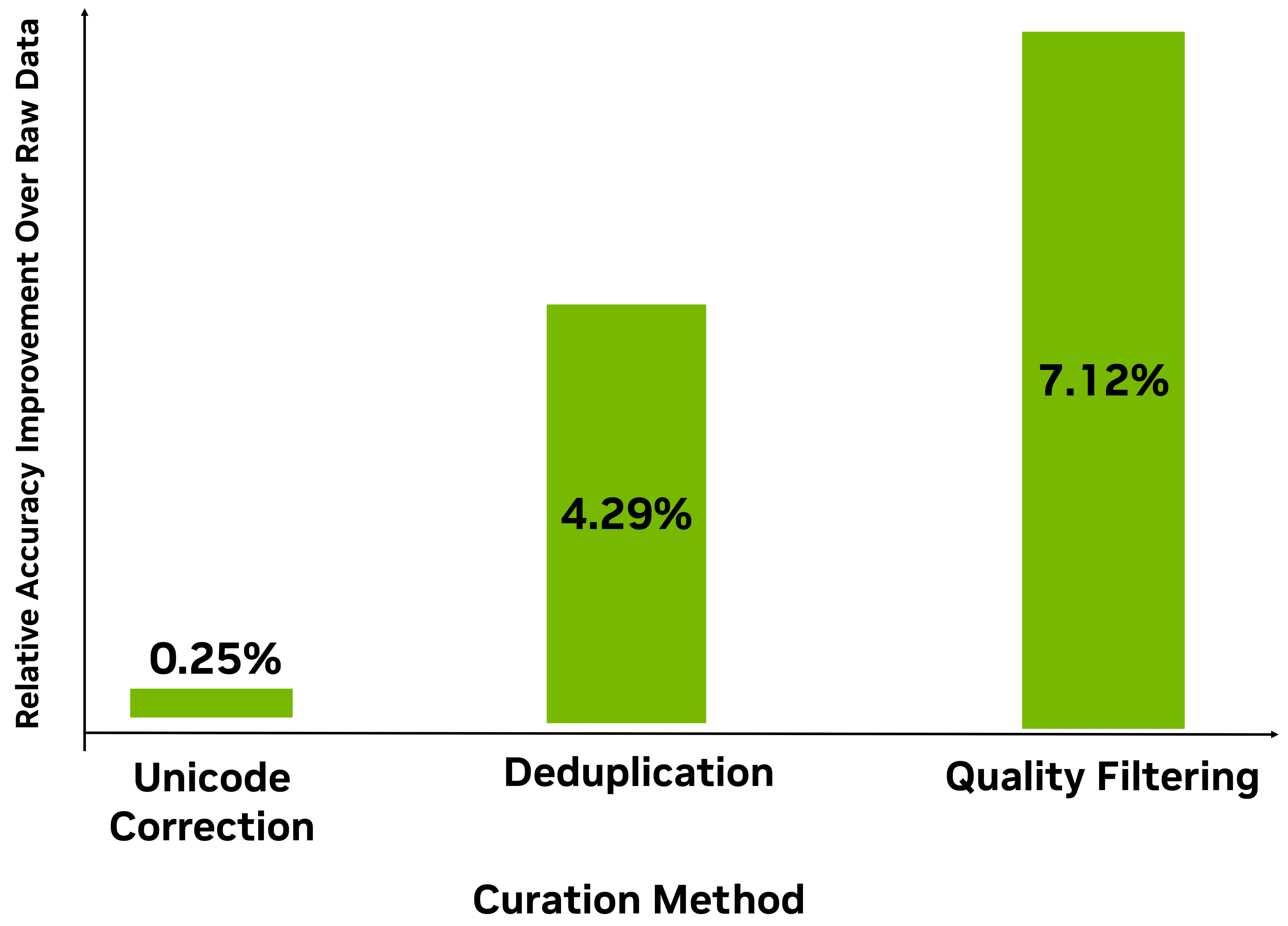
GitHub	NeMo framework container	PyPI	Microservice (coming soon)

Text Processing

High Quality Data Processing Maximizes Model Performance

Data Curation helps build SOTA Models

LLM Accuracy Improvement on Curated Data



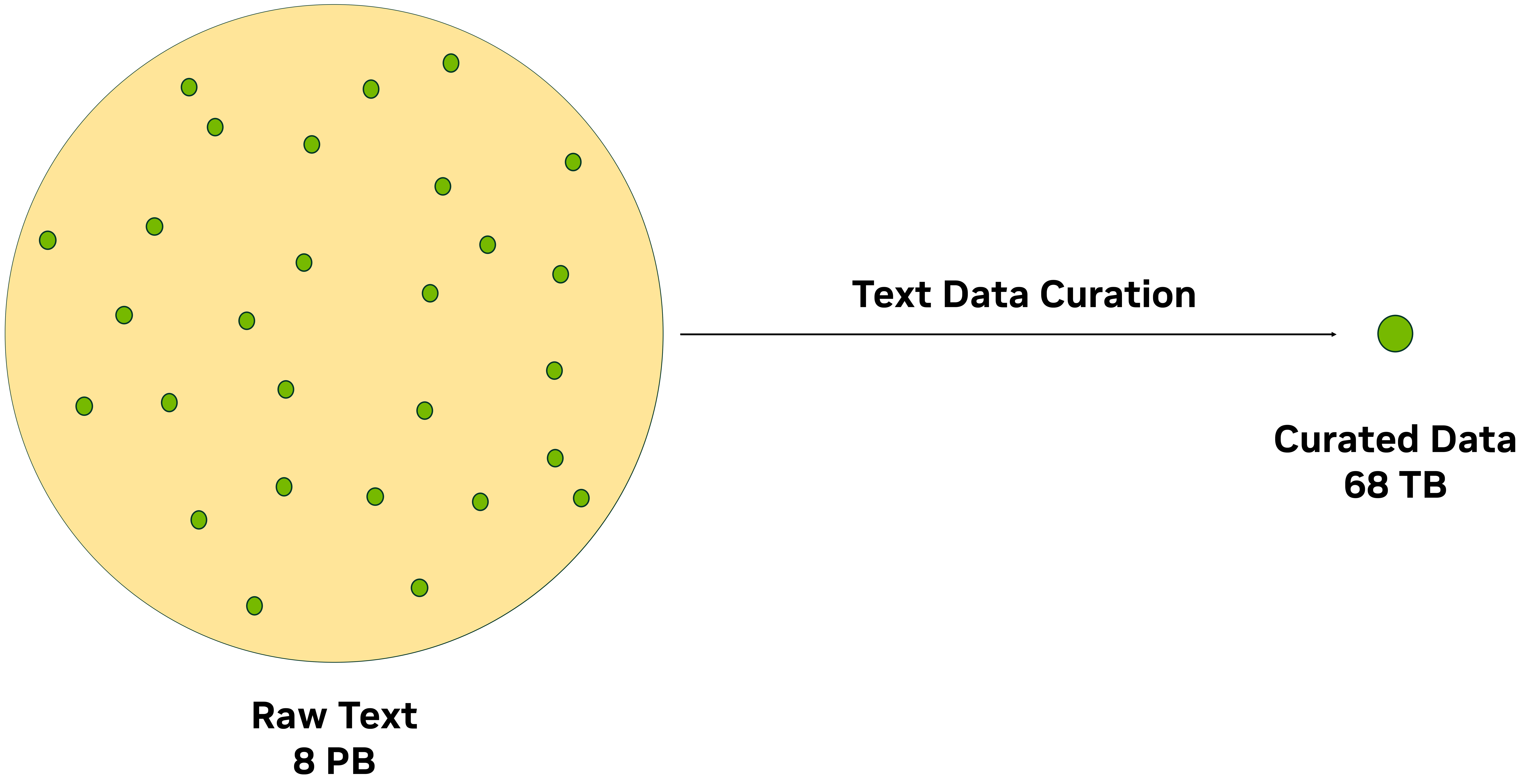
Ablation on 357M Parameter GPT Model

Why is Data Curation Important?

State-of-the-art data curation is essential for developing state-of-the-art models across all modalities

Higher Accuracy	TCO Savings	Faster Training	Task Specialization
Properly curated data leads to more improved accuracy across tasks	Decreases both training and inference costs significantly	Reduces compute requirements by orders of magnitude, enabling faster iterations	Enables optimization for specific tasks such as reasoning rather than general-purpose solutions

Only 1% of Raw Text Data is Curated to Train Foundation Models



Introducing: NeMo Curator for Text Processing

Easily integrate different features into your existing pipelines with Python APIs

GitHub	NeMo framework container	PyPI

NeMo Curator: Features to Train Foundation Models

Achieve higher accuracy with a variety of GPU-accelerated features

Synthetic Data Generation

- **Pre-built pipelines** - for tasks like prompt generation, dialogue generation, and entity classification
- **Modular** - Easily integrate NeMo Curator's features into your existing pipelines
- **OpenAI API compatible** - Integrate custom Instruct and Reward models

Deduplication & Classification

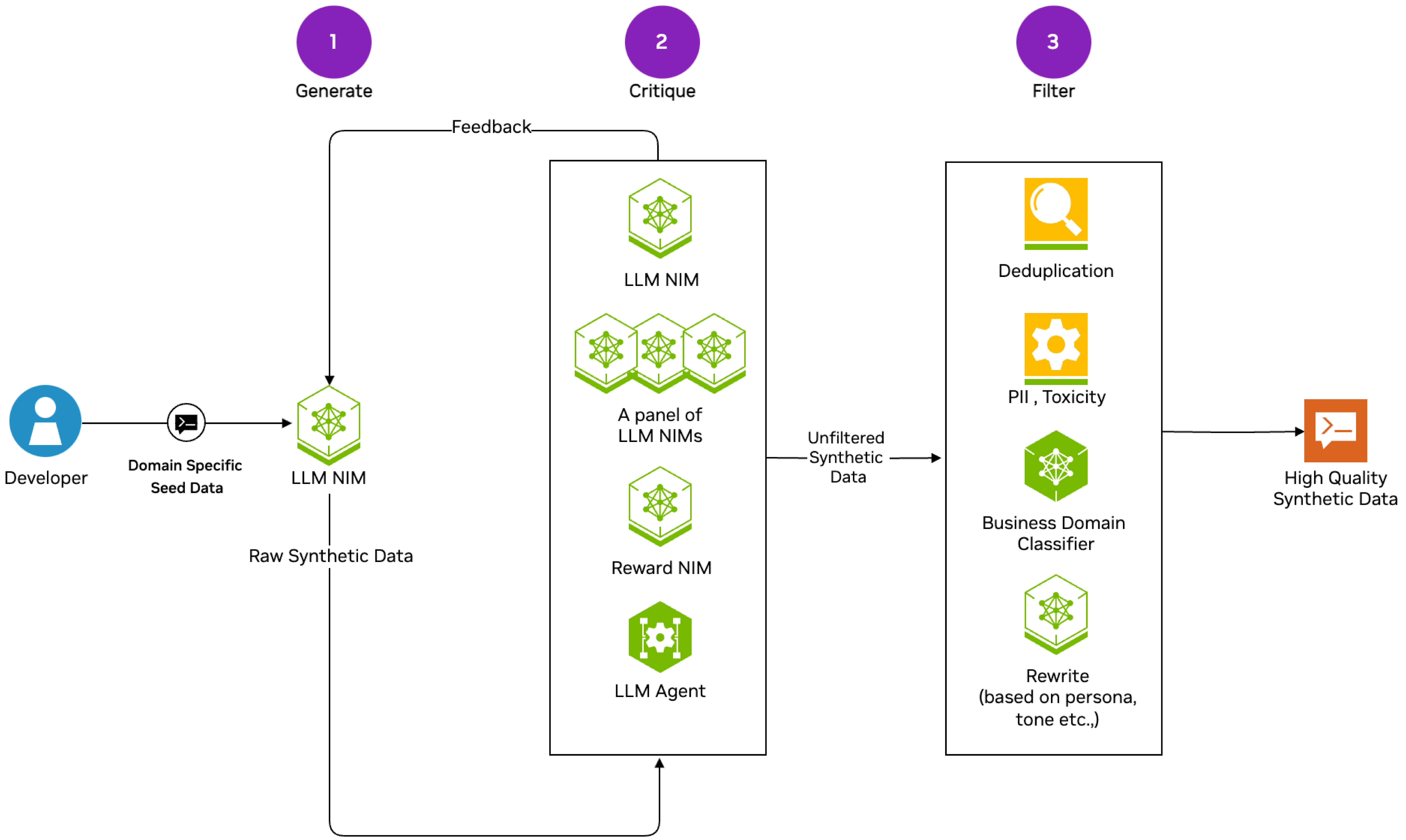
- **Lexical Deduplication** – Identical (Exact) or near identical (Fuzzy)
- **Semantic Deduplication** – focuses on the meaning rather than the exact text
- **Classifier Models** - State-of-the-art open models to either enrich or filter your data.

GPU Acceleration with RAPIDS

- **cuDF** - for deduplication & classifer models
- **cuML** - for K-means clustering in semantic deduplication
- **cuGraph** – for fuzzy deduplication

Synthetic Data Generation

Easily get started with pre-built pipelines or integrate various features into your existing workflows



Synthetic Data Generation

Easily get started with pre-built pipelines or integrate various features into your existing workflows

Pre-built Pipelines

- Prompt generation (open q/a, closed q/a, writing, math/coding)
- Synthetic two-turn prompt generation
  - Dialogue generation
  - Entity classification

Llama 3.1 Nemotron 70B/340B Reward Model

- Benchmark-Topping Performance
- Single scalar scoring for human preferences
- Permissive license for commercial use

Tooling Support

- Integrate into existing pipelines
- Bring your custom Instruct/Reward Model
- Supports different filtering techniques

Synthetic Data Generation: Example

Use prompts to synthetically generate the QnAs and the reward score

```python
model = "nvdev/nvidia/llama-3.1-nemotron-70b-instruct"

question_lst = []

for i in range(len(ragas)):
    closed_qa_responses = client.query_model(
        model=model,
        messages = [
            {
                "role": "user",
                "content": f"Generate a single, concise question similar to '{ragas.question[i]}' based on the provided context.\n"
                           f"Do NOT include any explanations, rationale, or additional text in your response—just the question itself:\n"
                           f"{ragas.contexts[i]}"
            }
        ],
        temperature = 0.2,
        top_p = 0.7,
        max_tokens = 1024,
    )

    question = closed_qa_responses[0]
    question_lst.append(question)

ragas['sdg_nemotron_q'] = question_lst
```

✓ 8.7s

```python
ragas['sdg_nemotron_q']
```

✓ 0.0s

```
0    What is the significance of including a module...
1    How does the task execution stage in HuggingGP...
2    What are the primary technical hurdles encount...
3    How does Algorithm Distillation (AD) leverage ...
4    What are the core architectural elements and o...
5    How do consistent naming conventions, intra-fi...
6    How does API-Bank assess LLMs' decision-making...
7    How does a package.json fit into defining a No...
8    How does API-Bank assess LLMs' utilization of ...
9    How do finite context lengths and long-term pl...
Name: sdg_nemotron_q, dtype: object
```

```python
reward_lst = []

model = "nvdev/nvidia/llama-3.1-nemotron-70b-reward"

for i in range(len(ragas)):
    messages = [
        {
            "role": "user",
            "content": f"""
                        Provide a concise and well-reasoned answer to the following question:
                        "{ragas.sdg_nemotron_q[i]}"
                        Use the context below to ensure accuracy and clarity:
                        {ragas.contexts[i]}
                        """
        },
        {
            "role": "assistant",
            "content": ragas.sdg_nemotron_a[i],
        },
    ]


    rewards = client.query_reward_model(messages=messages, model=model)
    rewards = float(rewards)

    reward_lst.append(rewards)

ragas['sdg_nemotron_reward'] = reward_lst
```

✓ 3.1s

```python
ragas['sdg_nemotron_reward']
```

✓ 0.0s

```
0   -13.93750
1    -4.09375
2    -8.68750
3    -8.06250
4    -5.15625
5   -15.75000
6    -8.18750
7    -9.31250
8    -7.56250
9   -11.06250
Name: sdg_nemotron_reward, dtype: float64
```

https://docs.nvidia.com/nemo-framework/user-guide/latest/datacuration/syntheticdata.html

NVIDIA

SDG Demo Video

Synthetic Data Generation: Example

Build an SDG pipeline to generate the data

main.py

```python
import argparse

SCRIPT_DIR_PATH = os.path.dirname(os.path.abspath(__file__))
DATA_DIR = os.path.join(SCRIPT_DIR_PATH, "data")
TEMP_DIR = os.path.join(SCRIPT_DIR_PATH, "_temp")
CONFIG_DIR = os.path.join(SCRIPT_DIR_PATH, "config")
DATASET_URL = "https://huggingface.co/datasets/ymoslem/Law-StackExchange/resolve/main/law-stackexchange-questions-answers.json"


def pre_imports():


def random_split_rows(rows: List[Any], train_ratio: float, val_ratio: float, seed=42):


def download_and_convert_to_jsonl() -> str:


def semantic_dedupe(dataset):


def run_curation_pipeline(


def run_pipeline(args, jsonl_fp):


def main():


if __name__ == "__main__":
```

syn_gen.py

```python
import asyncio

PROMPT_GENERATE_QUESTIONS_FROM_ANSWER = """TEXT:

PROMPT_PARAPHRASE_TEXT = """TEXT:


class SyntheticGenerator:

    def __init__(

    def run(

    def _split_sdg_responses(self, sdg_response: str) -> List[str]:

    def _write_all_to_file(self, gen_entries, out_fp: str):

    async def _prompt_model(

    async def _synthesize_from_source(
```

multiple rounds of SDG

```
data
  curated
    final
      law-qa-test.jsonl
      law-qa-train.jsonl
      law-qa-val.jsonl
    round-1
      law-qa-train-synth-round-1.jsonl
      law-qa-train.jsonl
    round-2
    round-3
    round-4
    round-5
  raw
    downloads
      law-stackexchange-questions-answers.json
    splits
      law-qa-test.jsonl
      law-qa-train.jsonl
      law-qa-val.jsonl
```

one example on SDG curated result


```json
{
    "id": "appliance-qa-B00074TB9U-synth-0",
    "asin": "B00074TB9U-synth-0",
    "question": "How effective is a ductless installation in eliminating cooking smells?",
    "answer": "The vent has two speeds; the lower one is suitable for regular air removal, while the higher speed is necessary for heavy-duty cooking. The higher speed functions well.",
    "questionType": "yes/no",
    "score": -1,
    "helpfulness": 0.6015625,
    "correctness": 0.388671875,
    "coherence": 3.078125,
    "complexity": 0.8671875,
    "verbosity": 0.60546875
}
```


https://github.com/NVIDIA/NeMo-Curator/tree/d401333ec9d88c36494befc9ae7515574c4d89fb/nemo_curator

Deduplication and Filtering

Scale across multi-node, multi-GPU setups, eliminating the need for iterative CPU processing

- Supports lexical and semantic deduplication for document processing
- Scales on multi node, multi-GPU by leveraging RAPIDS
- Scale to 100+ TB of data

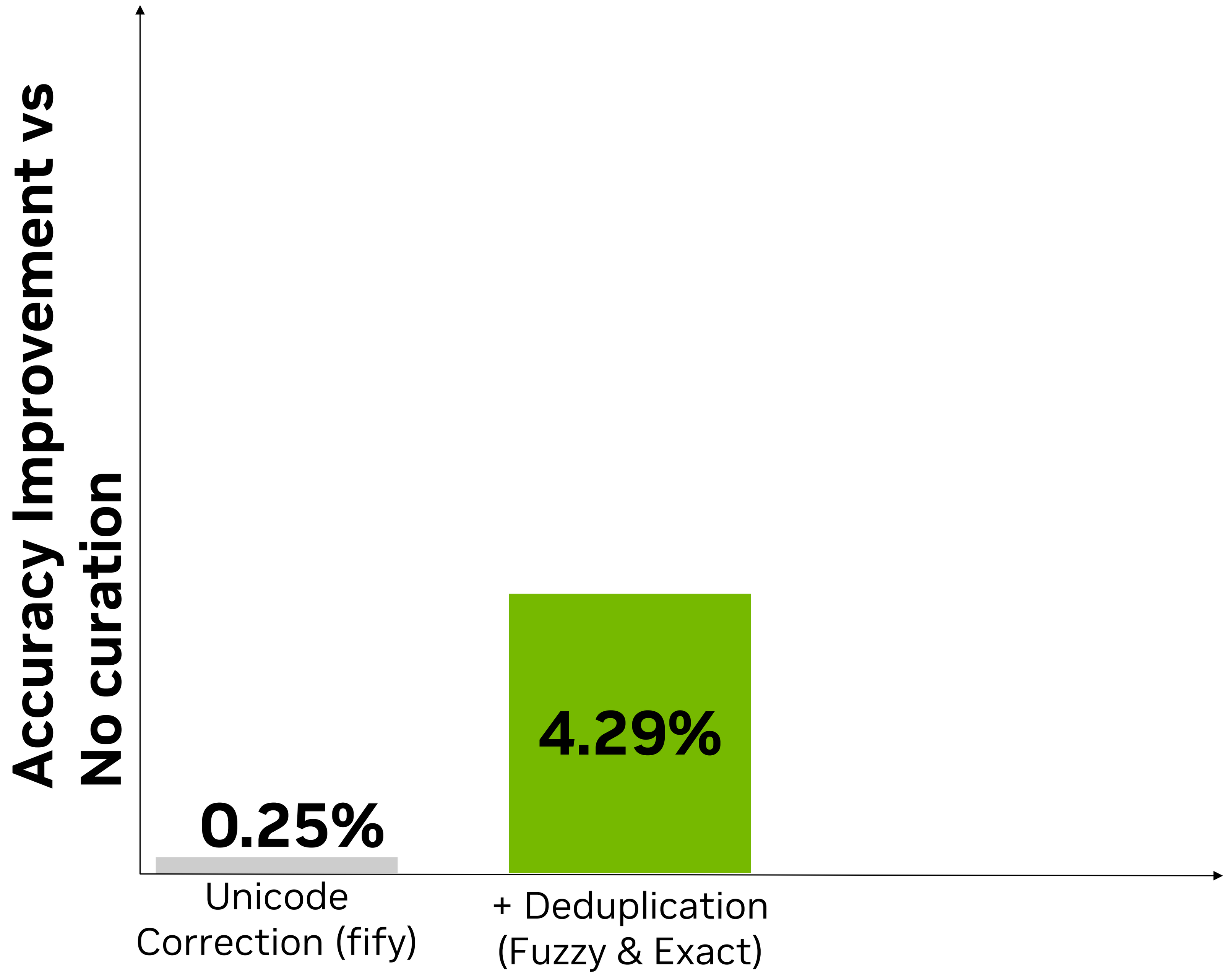
Lexical Deduplication

- Utilizes text similarity to discover duplication
- **Exact deduplication**
  - Hashing based matching for each document
- **Fuzzy deduplication**
  - Minhash, Bucketization and Clustering

Semantic Deduplication

- Utilizes meaning of text to discover duplication
- Leverages embedding models to identify semantic documents
- Modularity to use custom embedding model

Acceleration



357M GPT model using deduplication, heuristic filtering, and unicode correction

NeMo Curator - Deduplication

Scale across multi-node, multi-GPU setups, eliminating the need for iterative CPU processing

Value Proposition

- Accelerate on multi-node multi-GPU setups by leverage RAPIDS
- Scale up to 100+ TBs of data
- Get support for both
  - Lexical Deduplication: Uses text similarity, both Exact and Fuzzy
  - Semantic Deduplication: Uses text meaning

Technical Details

Lexical Deduplication

- Exact deduplication
  - Hashing for each document
- Fuzzy deduplication
  - Minhash, Bucketization and Clustering

Semantic Deduplication

- Leverage embeddings to identify semantic documents
- Use SOTA NeMo Retriever Text Embedding NIM or customize with your embedding model

NeMo Curator - Classifier Models

Create high-quality data blends with RAPIDS accelerated inference

Value Proposition

- Accelerated inference with RAPIDS powered distributed data classification module and intelligent batching
- Seamless scalability for classifying TBs of data
- Faster processing with parallelization across multiple GPUs
- Lower memory and compute footprint with state-of-the-art open classifier models (Apache 2.0 license)
- 8 classifier models released

Classifier Models

Domain Classifier

- Supports 26 domain classes
  - Top 10 classes : Finance, Health, Business and Industrial, Science, Law and Government, Internet and Telecom, Jobs and Education, News, Computers and Electronics, Shopping;
- Trained on 1 million Common Crawl samples & 500k Wikipedia articles
- Also available for multiple languages

Quality Classifier

- Classify quality of the document to 'High', 'Medium' or 'Low'
- Training data annotated by humans on quality factors such as: content accuracy, clarity, coherence, grammar, depth of information and overall usefulness of the document

NeMo Curator - Classifier Models

Create high-quality data blends with RAPIDS accelerated inference

Prompt Task and Complexity Classifier

- Classify tasks across 11 categories
  - OpenQA, Closed QA, Summarization, Text Generation, Code Generation, Chatbot, Classification, Rewrite, Brainstorming, Extraction, Other
- Compute complexity on 6 dimensions
  - Creativity, Domain Knowledge, Reasoning, Constraints, Contextual Knowledge, # of Few shots

FineWeb Nemotron-4 Edu Classifier

- Determine the educational value of a piece of text (score 0-5 from low to high).
- Trained using annotations from Nemotron-4-340B-Instruct

Content Type Classifier

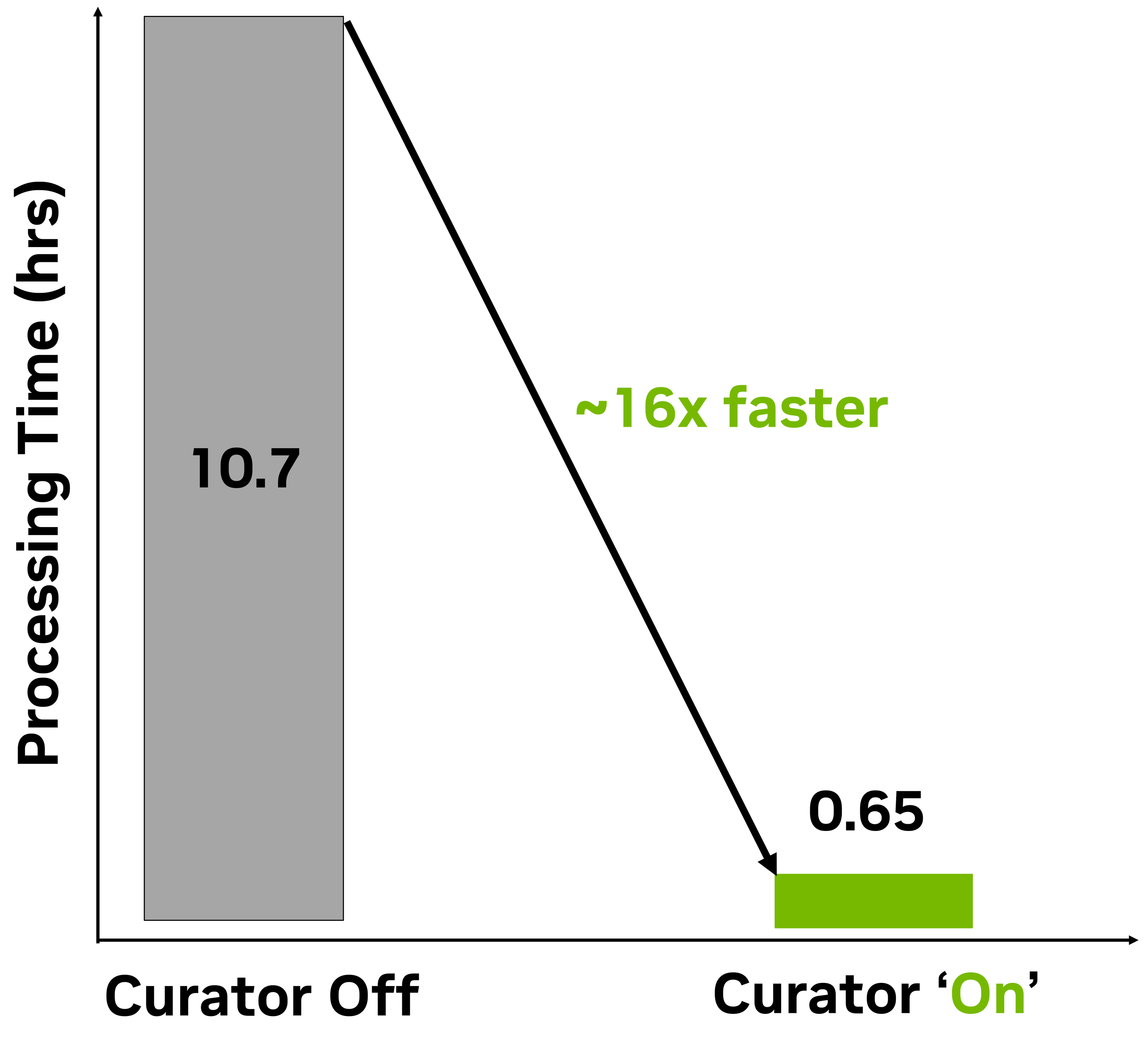
- Categorize documents into 11 content types
  - Explanatory articles, News, Blogs, Boilerplate content, Analytical exposition, Online Comments, Reviews, Books & literature, Conversational, and Personal Websites.
- Useful for content management systems, digital publishers, recommendation systems

Instruction Data Guard

- Identify malicious prompts used for fine-tuning
- Optimal use for instruction:response datasets

Performance – Fuzzy Deduplication

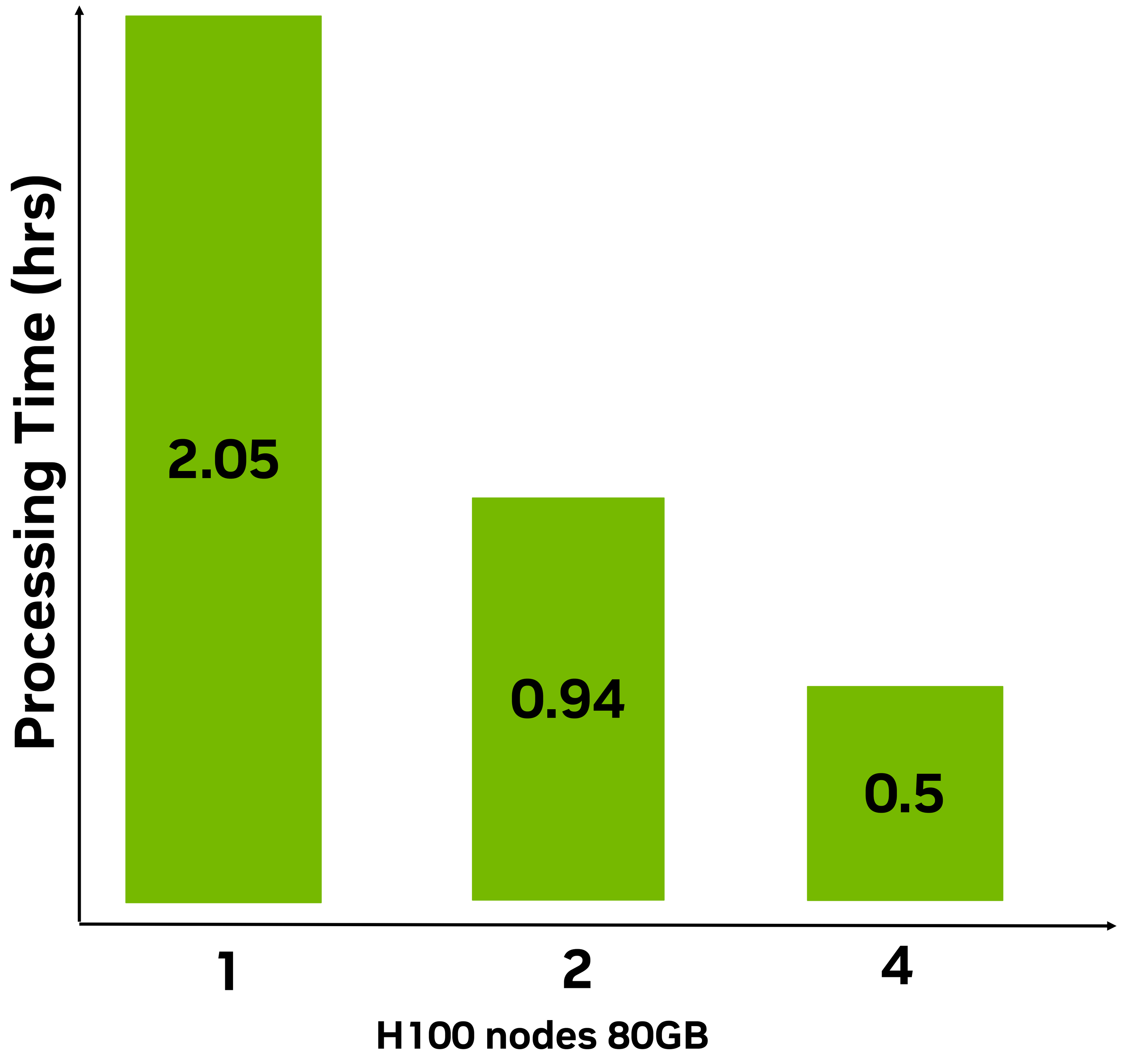
~16x Faster Processing



Processing time for fuzzy deduplication of RedPajama-v2 subset (8TB/1.78T tokens)

'On': Data processed with NeMo Curator on 3 H100 nodes

'Off' : Data processed with a leading alternative library on CPUs



Processing time for fuzzy deduplication of RedPajama-v2 subset (8TB/1.78T tokens)

Scaling on 1, 2, 3, 4 H100 nodes 80GB

Deep Dive: Fuzzy Deduplication

Identifying similar documents

Document: "We the People of the United States, in Order to form a more perfect Union, establish Justice…"

Exact Duplicate: "We the People of the United States, in Order to form a more perfect Union, establish Justice…"

Fuzzy Duplicate: "Here is the US Constitution. We the People of the United States, in Order to form a more perfect Union, ensure Justice…"

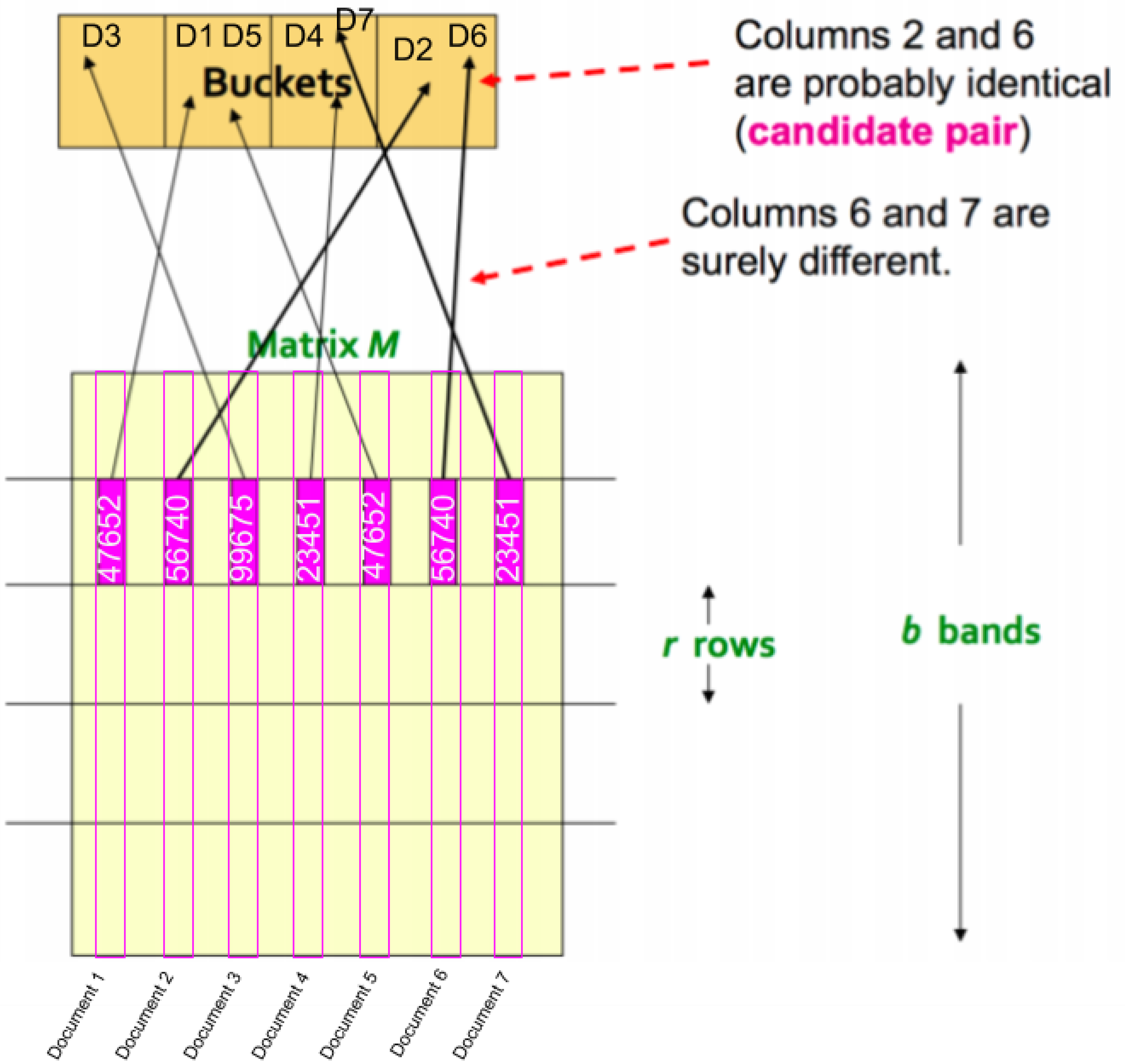
Fuzzy Duplicate: "We the People of the United States, in Order to ensure Justice and domestic tranquility…"

Min-hashing

Doc -1					
K-shingle	{The quick brown}	{fox jumps over}	{the lazy dog}		
hashed-shingle	345L	3455L	934L		
Hash_1	23	49	50	23	Signature-1
Hash_2	56	24	39	24	
Hash_3	38	56	84	38	
Hash_4	48	29	93	29	
Hash_5	67	75	59	59	
Doc -2					
K-shingle	...	...	...		
hashed-shingle	...	...	...		
Hash_1	...	...	...	34	Signature-2
Hash_2	...	...	...	56	
Hash_3	...	...	...	78	
Hash_4	...	...	...	23	
Hash_5	...	...	...	14	
Doc -3					
...					

Src: https://mrhasankthse.github.io/riz/2020/03/19/Minhash-and-LSH.html

Signatures over hash buckets

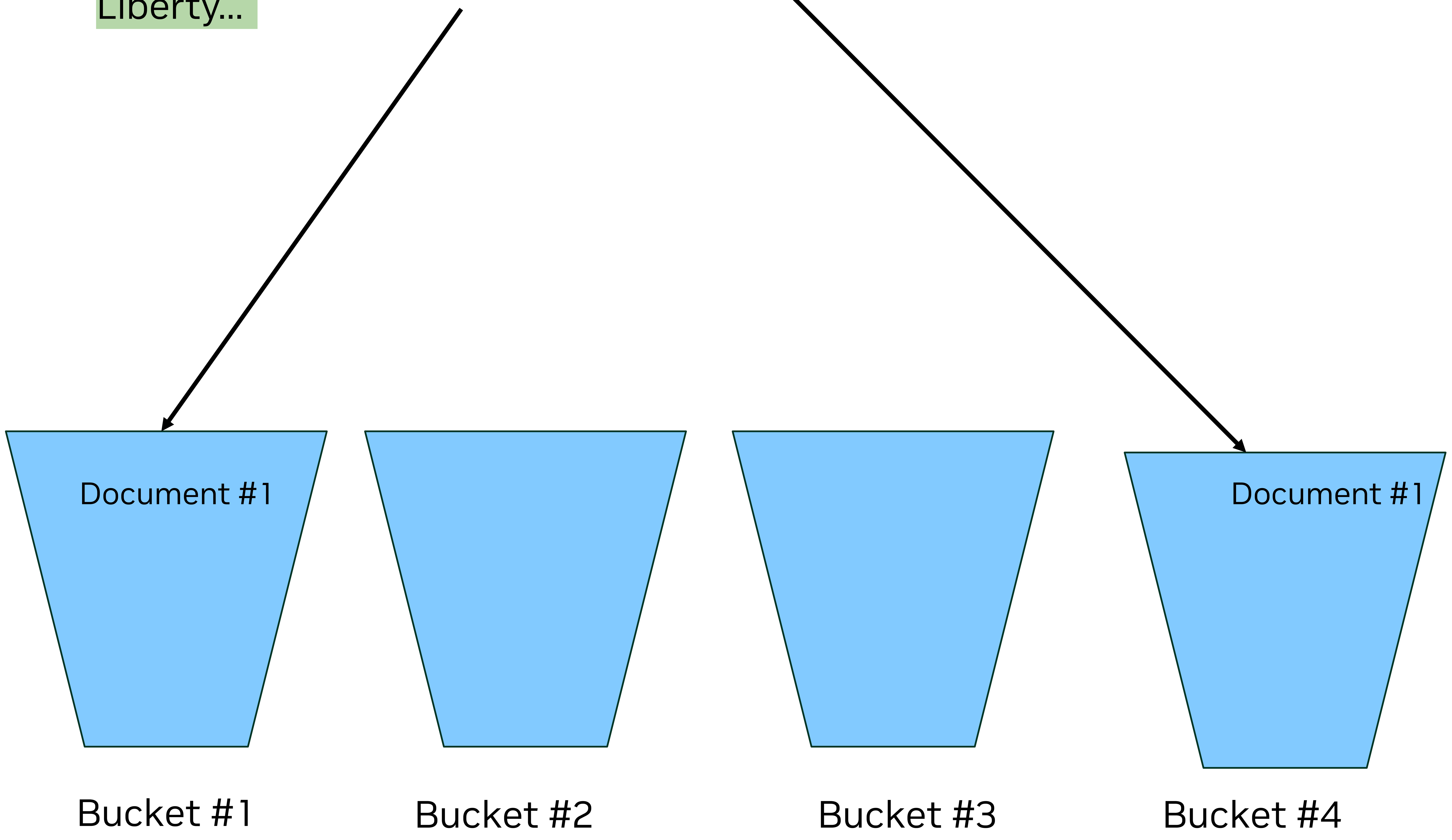


Src: https://mrhasankthse.github.io/riz/2020/03/19/Minhash-and-LSH.html

Deep Dive: Fuzzy Deduplication

Min-Hashing and Bucketization

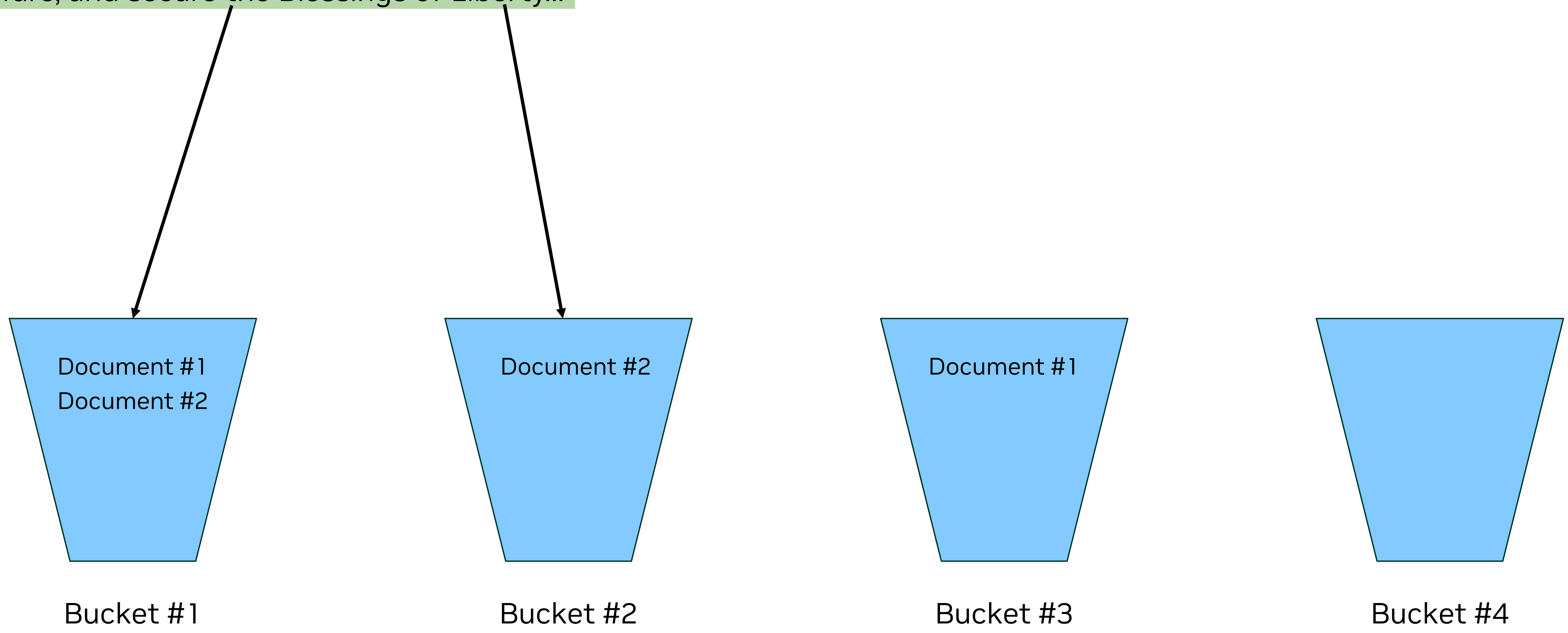
Document 1: "We the People of the United States, in Order to form a more perfect Union, establish Justice, ensure domestic Tranquility, provide for the common defense, promote the general Welfare, and secure the Blessings of Liberty…"



Deep Dive: Fuzzy Deduplication

Min-Hashing and Bucketization

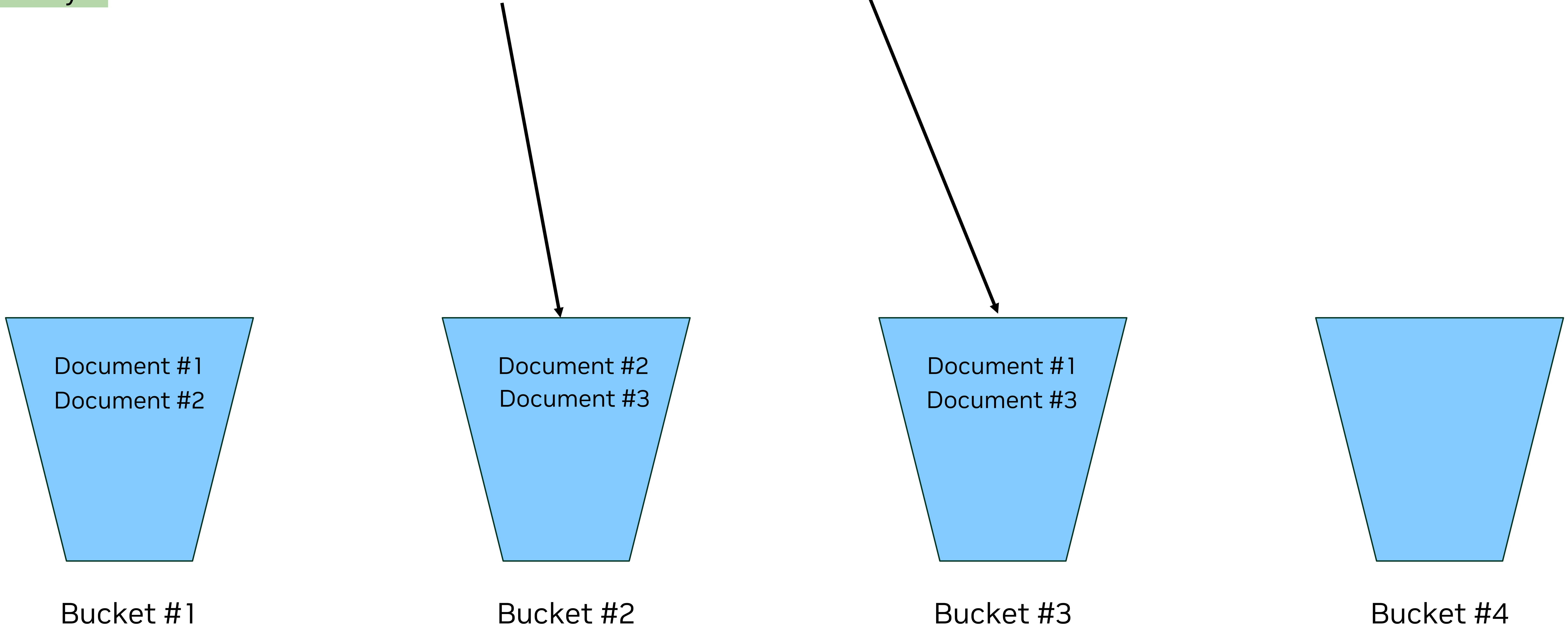
Document 2: “Here is the US Constitution. We the People of the United States, in Order to form a more perfect Union, establish Justice, insure foreign Tranquility, provide for the common defense, promote the general Welfare, and secure the Blessings of Liberty…”



Deep Dive: Fuzzy Deduplication

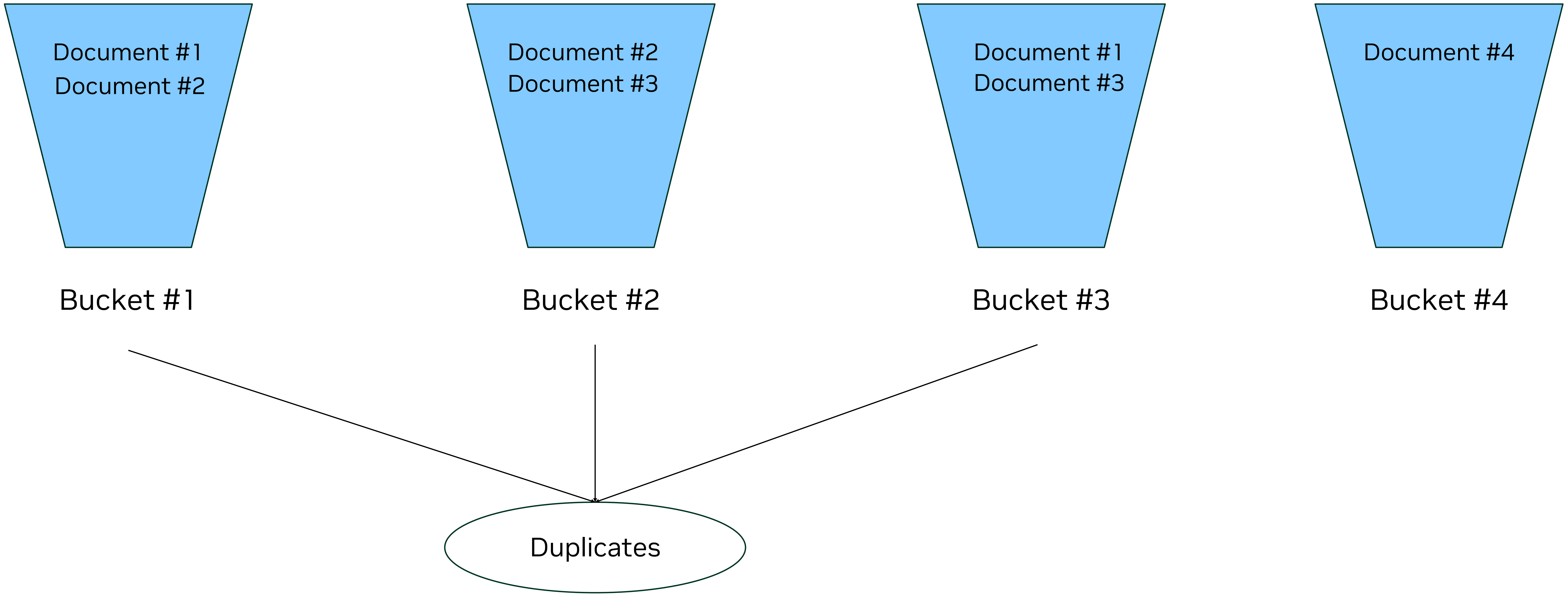
Min-Hashing and Bucketization

Document 3: “We the People of Canada, in Order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common defense, promote the general Welfare, and secure the Blessings of Liberty…”



Deep Dive: Fuzzy Deduplication

Min-Hashing and Bucketization



Semantic Deduplication: Example

Remove redundant data by identifying and eliminating semantically similar data points

sem_dedup_config.yaml

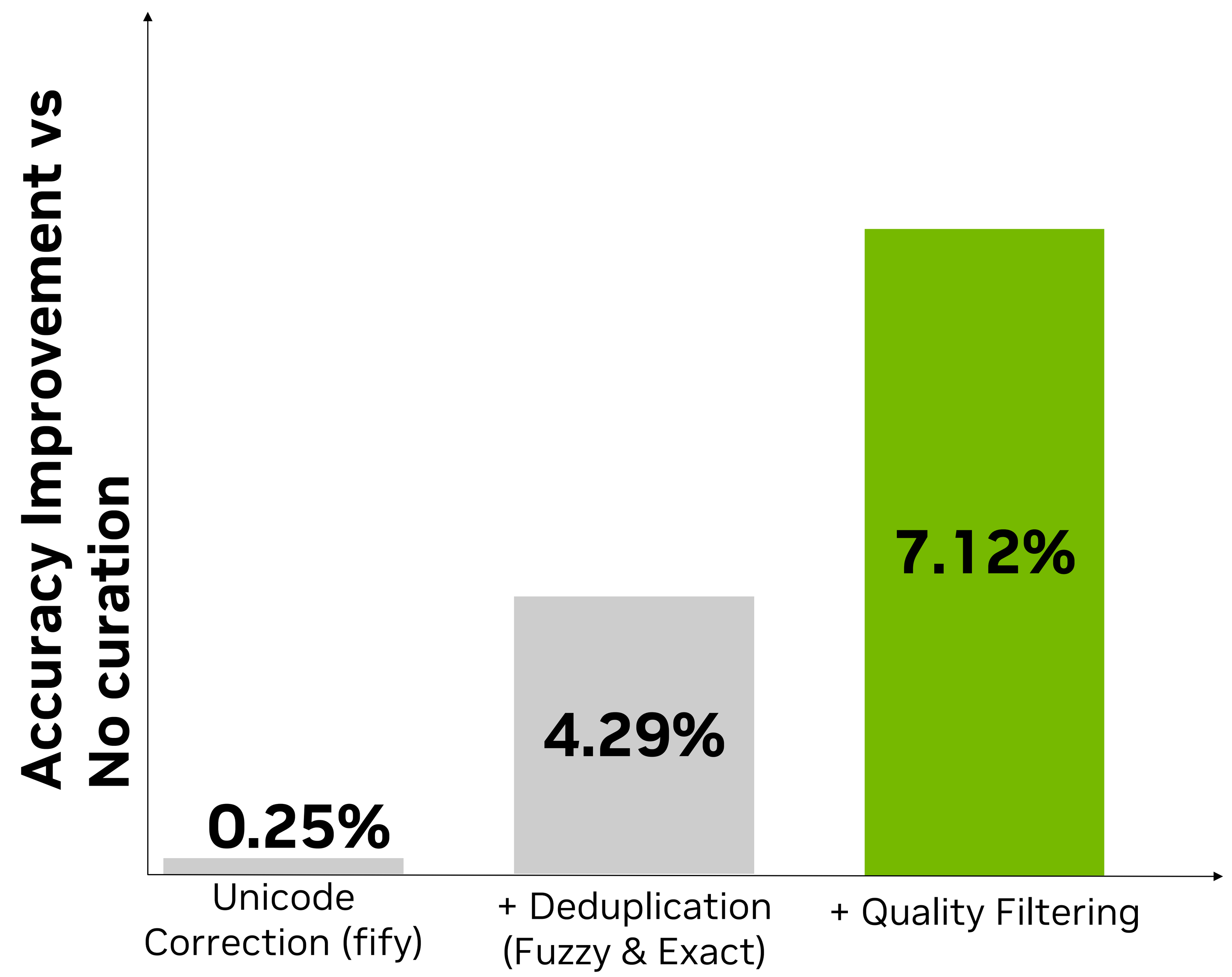
```yaml
# Configuration file for semantic dedup
cache_dir: "semdedup_cache"
num_files: -1

# Embeddings configuration
embeddings_save_loc: "embeddings"
embedding_model_name_or_path: "sentence-transformers/all-MiniLM-L6-v2"
embedding_batch_size: 128

# Clustering configuration
clustering_save_loc: "clustering_results"
n_clusters: 1000
seed: 1234
max_iter: 100
kmeans_with_cos_dist: false

# Semdedup configuration
which_to_keep: "hard"
largest_cluster_size_to_process: 100000
sim_metric: "cosine"

# Extract dedup configuration
eps_thresholds:
  - 0.01
  - 0.001

# Which threshold to use for extracting deduped data
eps_to_extract: 0.01
```

https://docs.nvidia.com/nemo-framework/user-guide/latest/datacuration/semdedup.html

Classification and PII

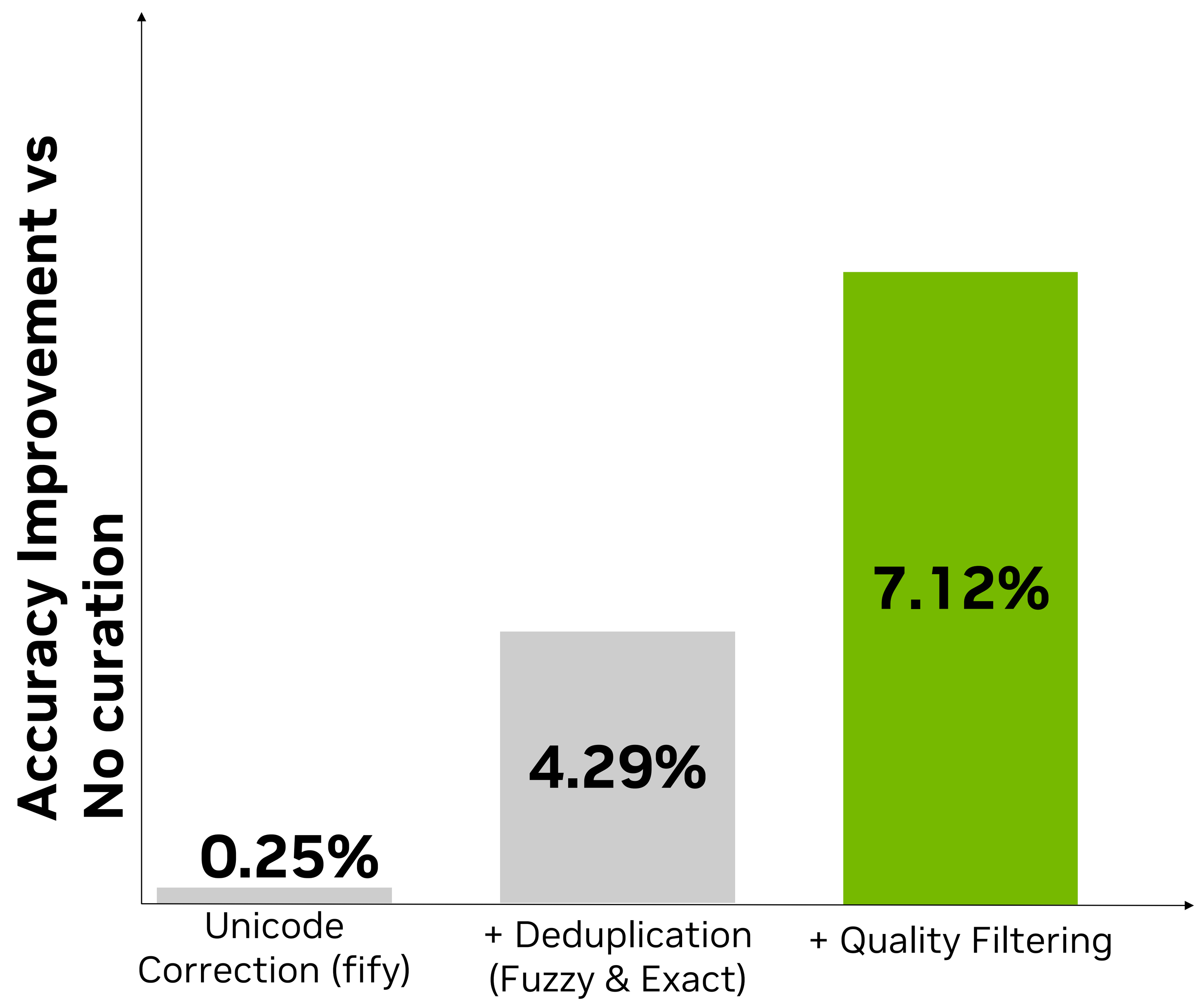
Create high-quality data blends with RAPIDS accelerated inference

- Accelerated inference through distributed classification model and intelligent batching
- Redact/remove Personally Identifiable information (PII) using SOTA model

Classification	Mask/remove PII	Acceleration
• **Domain Classifier** • Model trained on 1.5 million samples • Classifies text in 26 domain classes • **Document Quality Classifier** • Classifier quality of document in ‘High’, ‘Medium’, ‘Low’ categories • Enables document quality check	• Remove sensitive information from data • Utilizes State-Of-The-Art spacy model to remove PII information	 357M GPT model using deduplication, heuristic filtering, and unicode correction

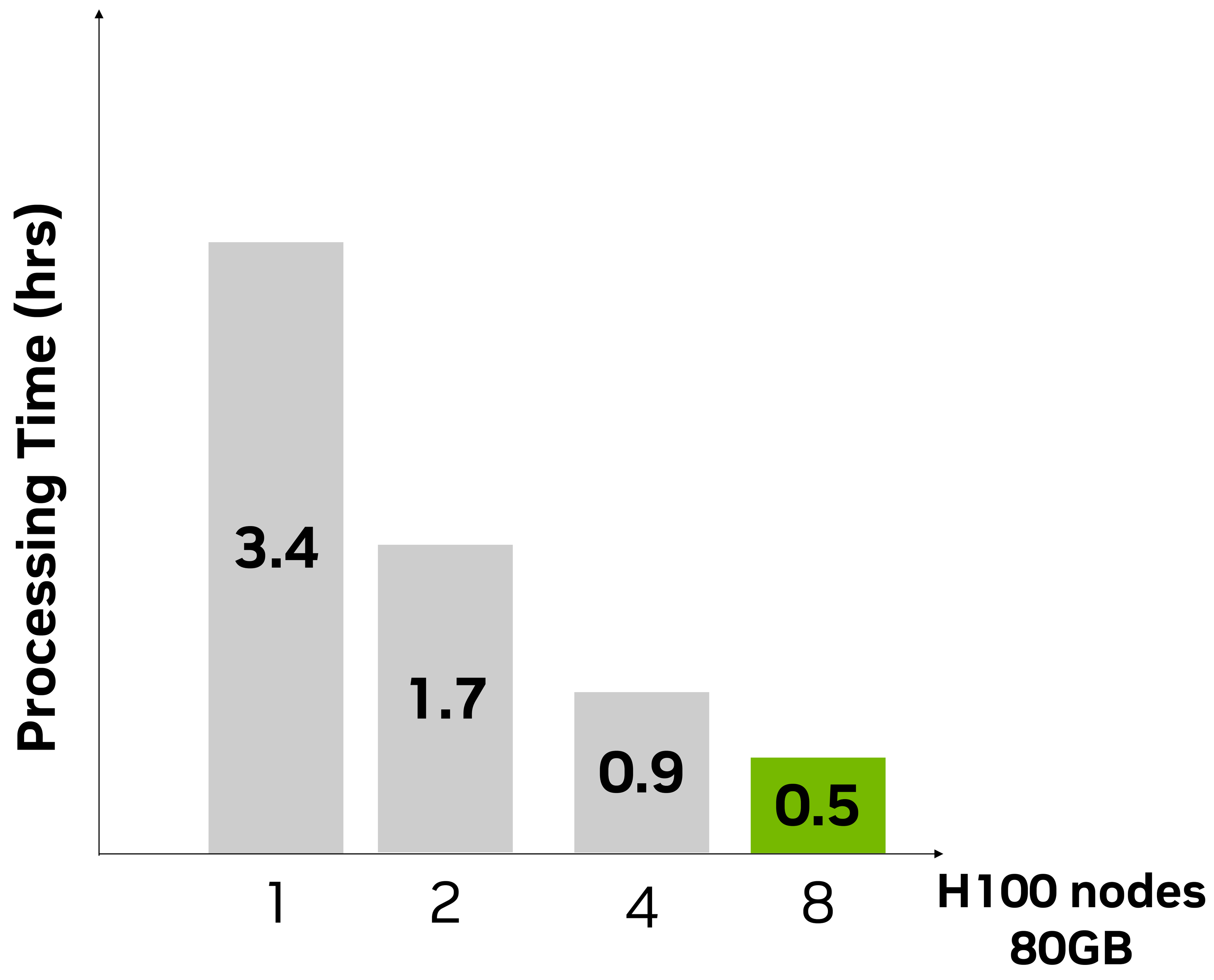
Accelerated Data Processing Maximizes LLM Performance & Scale

Nemo Curator: Up to 7% better Accuracy for LLM Downstream Tasks



357M GPT model using deduplication, heuristic filtering, and unicode correction

Scale to 100+ TB of data



Scaling Fuzzy Deduping 1.2T tokens (RedPajama-v1)

NeMo Curator: Example

```
# Download your dataset
dataset = download_common_crawl("/datasets/common_crawl/", "2021-04", "2021-10", url_limit=10)
# Build your pipeline
curation_pipeline = Sequential([
  # Fix unicode
  Modify(UnicodeReformatter()),
  # Discard short records
  ScoreFilter(WordCountFilter(min_words=80)),
  # Discard low-quality records
  ScoreFilter(FastTextQualityFilter(model_path="model.bin")),
  # Discard records from the evaluation metrics to prevent test set leakage.
  TaskDecontamination([Winogrande(), Squad(), TriviaQA()])
])
# Execute the pipeline on your dataset
curated_dataset = curation_pipeline(dataset)
```

GitHub	NeMo framework container	PyPI

NeMo Curator - Resources

Getting Started

- NeMo Framework Container
- GitHub
- PyPI
- Installation Guide
  Developer Page
- User Guide/ Docs
- API Docs
- Classifier Models
- Examples
- Best Practices
- Bugs
- Discussions

Tutorials & Blogs

Pre-training / DAPT

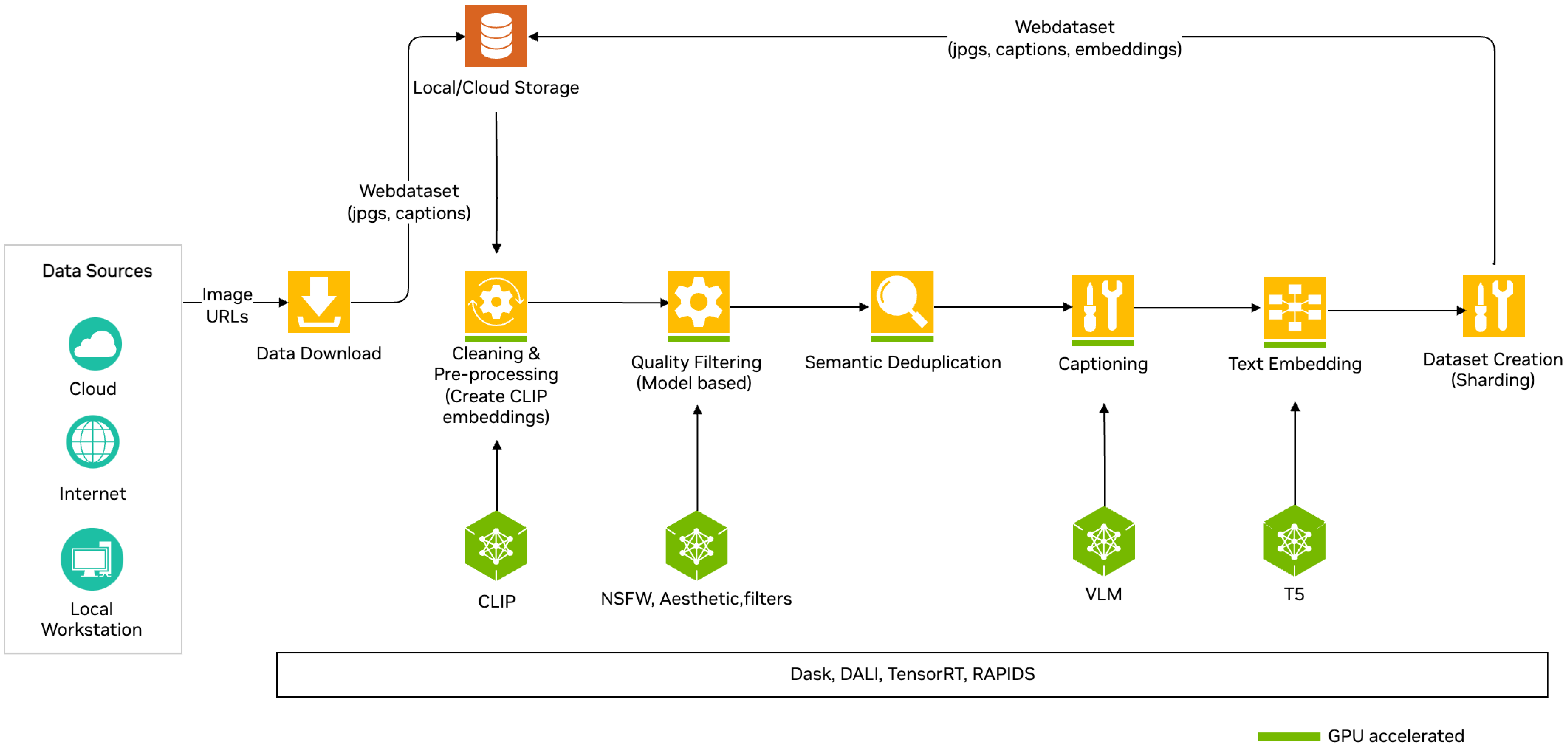
- Curating data for LLM training (Blog)
- Curating non-English data (Blog)
- How-to run classifier models
- All blogs

Fine-tuning

- Curating data for PEFT (Blog)
- Curating synthetic data for PEFT
- SDG using Llama 3.1 405B & Nemotron 4-340B
- SDG using Nemotron 4-340B

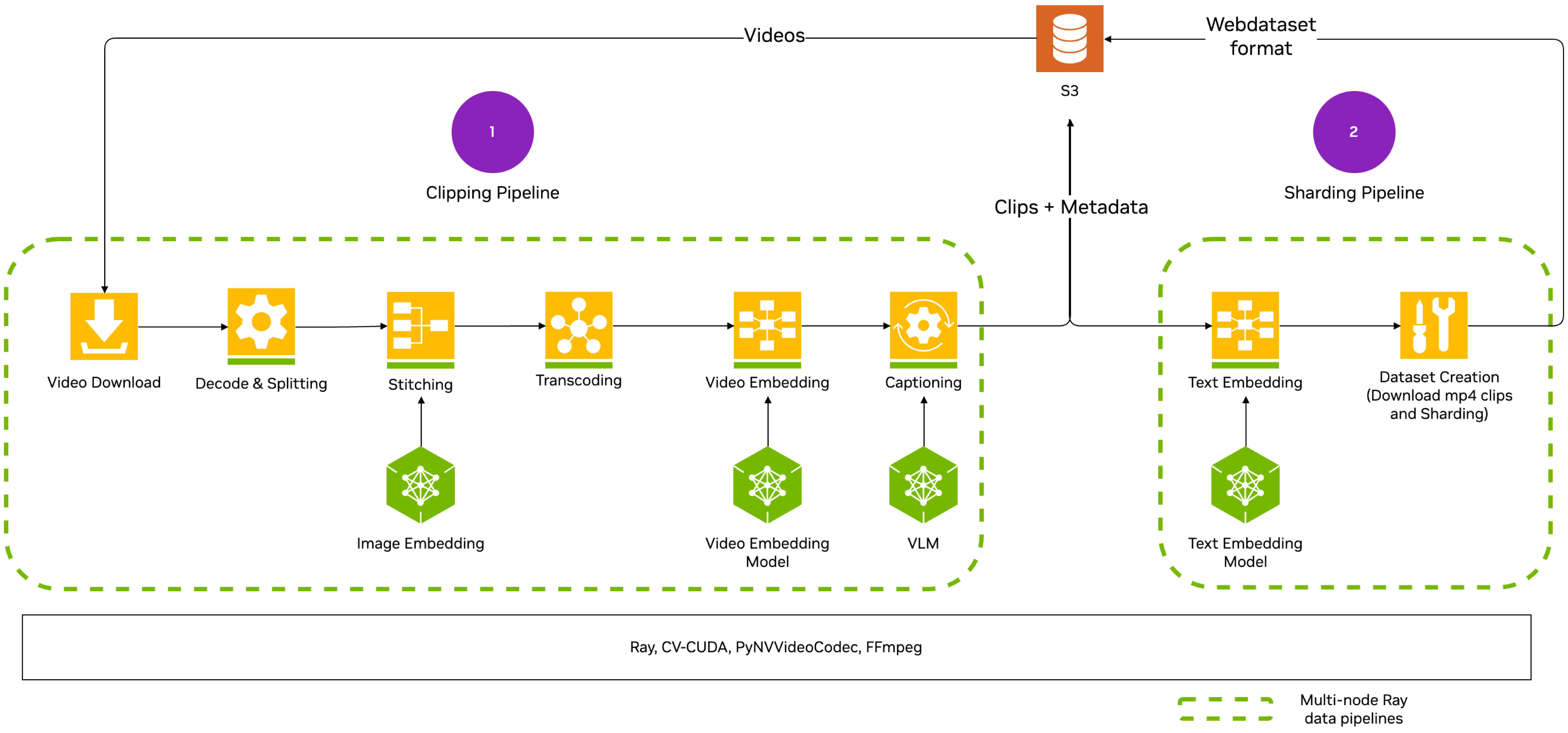
Image Processing

NeMo Curator Architecture: Image Processing



Video Processing

Video Curation: EA Features



State of the Art Models

Example Use case: Generate new lighting, weather, and geolocations

Best Practices

- Choosing the Right Quality Model Type
- Handling GPU Out-of-Memory (OOM) Errors
  - Controlling Partition Sizes

- Fuzzy Deduplication Guidelines
  - Reduce bucket counts
  - Reduce buckets per shuffle
  - Adjust files per partition
- GPU Memory and Utilization – Dask Dashboard

Developer Tools and Resources

Accelerate innovation and growth

Learn more: developer.nvidia.com

Individuals

Software

100s of APIs, models, SDKs, microservices, and early access to NVIDIA tech

Training

Hands-on self-paced courses, instructor-led workshops, and certifications

GPU Sandbox

Approval basis, multi-GPU and multi-node

Learning

Tutorials, self-paced courses, blogs, documentation, code samples

Community

Dedicated developer forums, meetups, hackathons

Ecosystem

GTC, NVIDIA Partner Network

Organizations

Startups

Cloud credits, engineering resources, technology discounts, exposure to VCs

Venture Capital

Deal flow and portfolio support for Venture Capital firms

Higher Education

Teaching kits, training, curriculum co-development, grants

ISVs and SIs

Engineering guidance, discounts, marketing opportunities

Research

Grant programs, collaboration opportunities

Enterprises

Tailored developer training, skills certification, technical support

Thank You