Molecular Graph Captioning

ALTEGRAD Data Challenge

Team EAG

Edgar Duc Axel Mauroy Gabriel Mariadass

January 2026

Outline

1. Introduction
2. Data
3. Retrieval-Based Method (1)
4. Retrieval-Based Results
5. End-to-End Generation (2)
6. Retrieval-Augmented Generation
   - Functional Group Prompting (3)
   - Top-K Description Prompting (4)
   - Retrieval-Guided Description Rewriting (5)
7. Generative-Based Results

Introduction

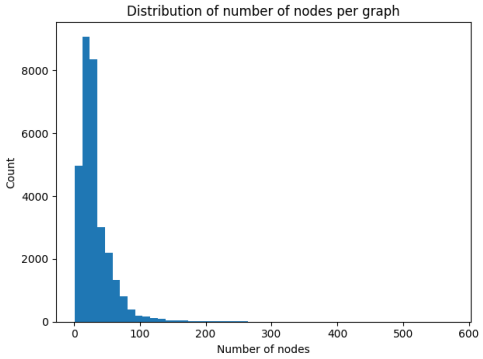
- **Molecular Graph Captioning**: generating natural language descriptions from molecular graphs
- Our focus: a **retrieval-based approach**, with additional generative and hybrid approaches

Data - Data Overview

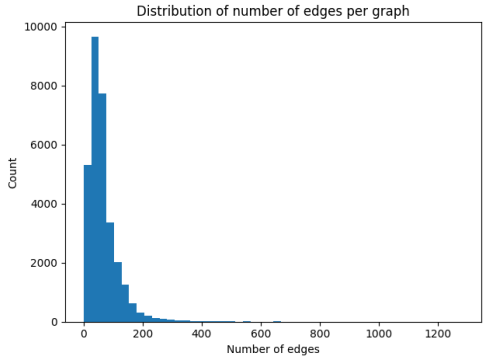
- **Dataset**
  - ∼33k molecular graphs with natural language descriptions
  - Splits: 31,008 train / 1,000 val / 1,000 test (captions hidden)
  - Stored as `PyTorch Geometric Data` objects
- **Graph representation**
  - Nodes = atoms, edges = chemical bonds
  - **Node features**: 9 categorical attributes (atom type, degree, charge, hybridization, aromaticity, ring membership, . . .)
  - **Edge features**: bond type, stereochemistry, conjugation

Data - Graph and Caption Statistics

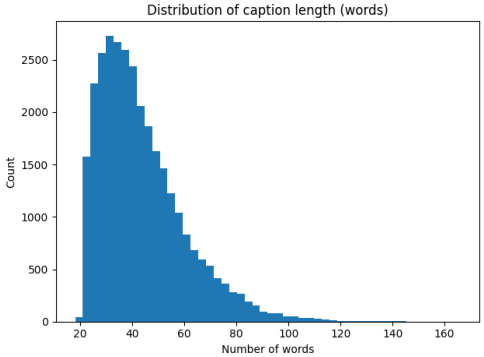
- **Graph structure**
  - Nodes = atoms, edges = chemical bonds
  - Atoms: 1 to 574 (mean ∼32)
  - Bonds: 0 to 1,284 (mean ∼67)
- **Non-uniform distributions**
  - Many small molecules
  - Long tail of large graphs
  - Similar across all splits
- **Caption variability**
  - 18–166 words (mean ∼43)
  - 95–1,377 characters (mean ∼294)



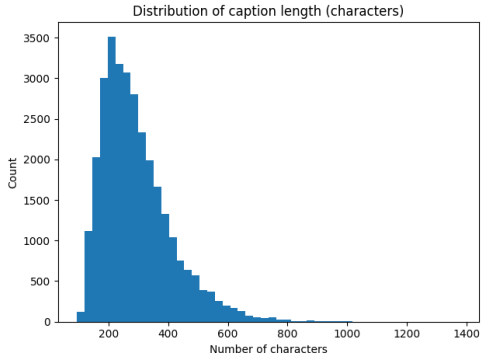
Nodes per molecule



Edges per molecule



Caption length (words)

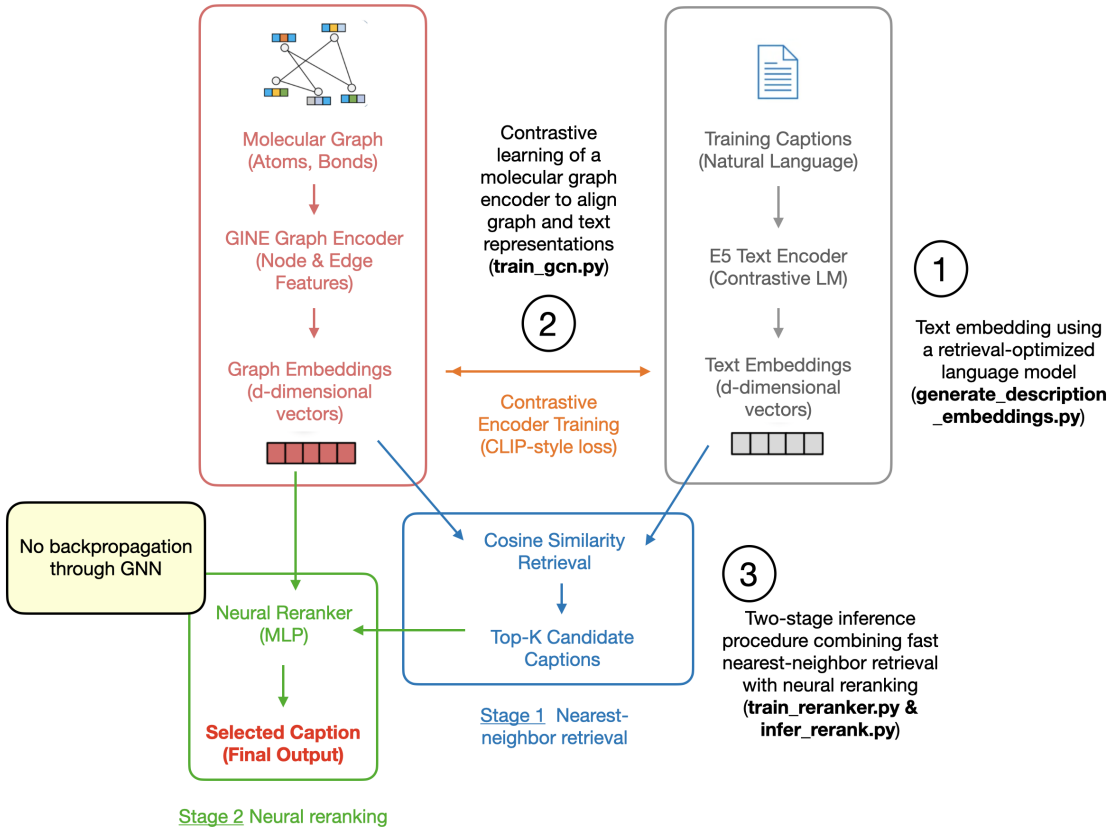


Caption length (characters)

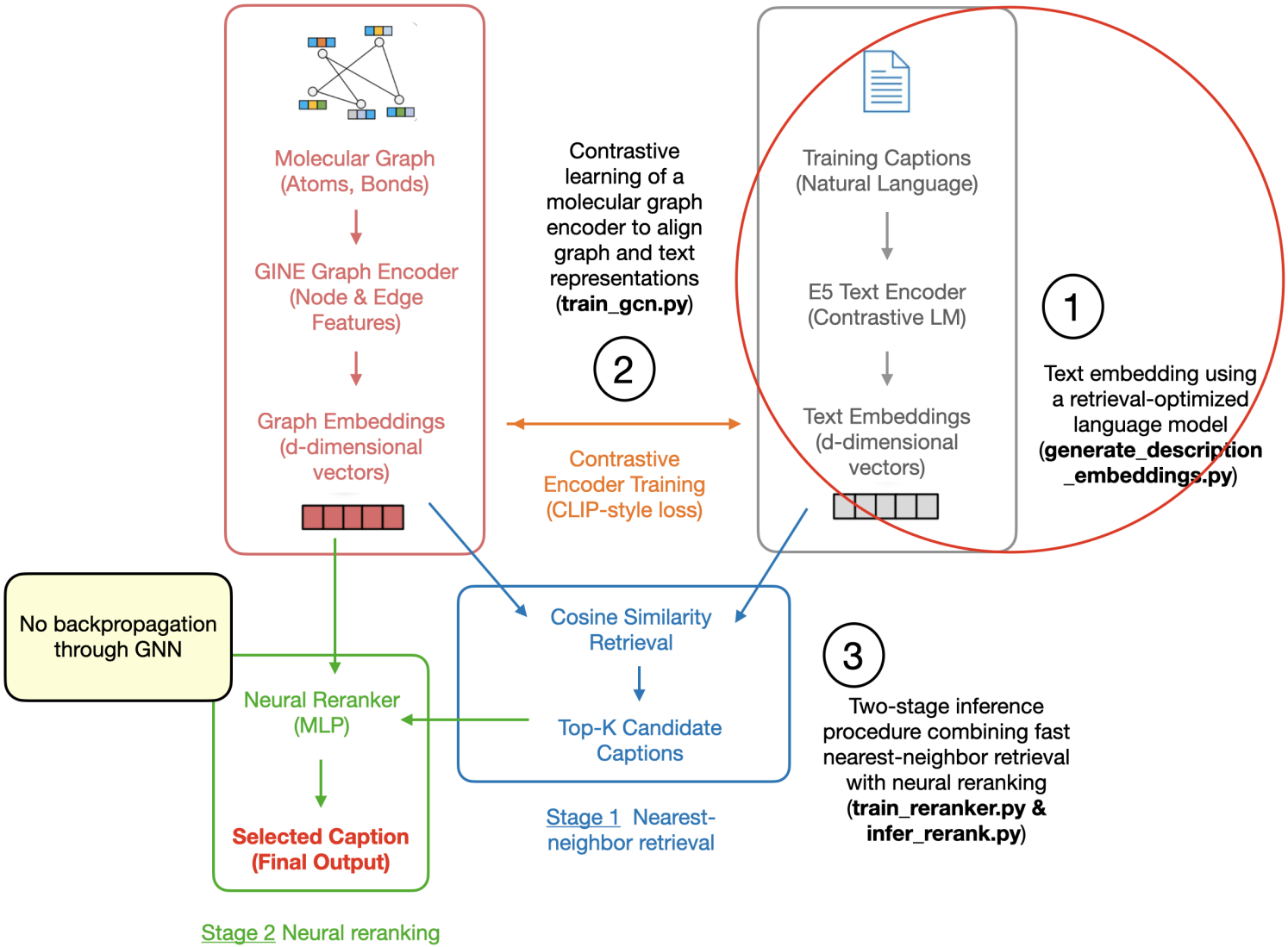
Retrieval-Based Method

Retrieval-Based Method (1) - High-Level Idea

- **Retrieval-based molecular graph captioning**
- Retrieve the most relevant caption in a shared graph–text embedding space
- Similarity measured with cosine similarity



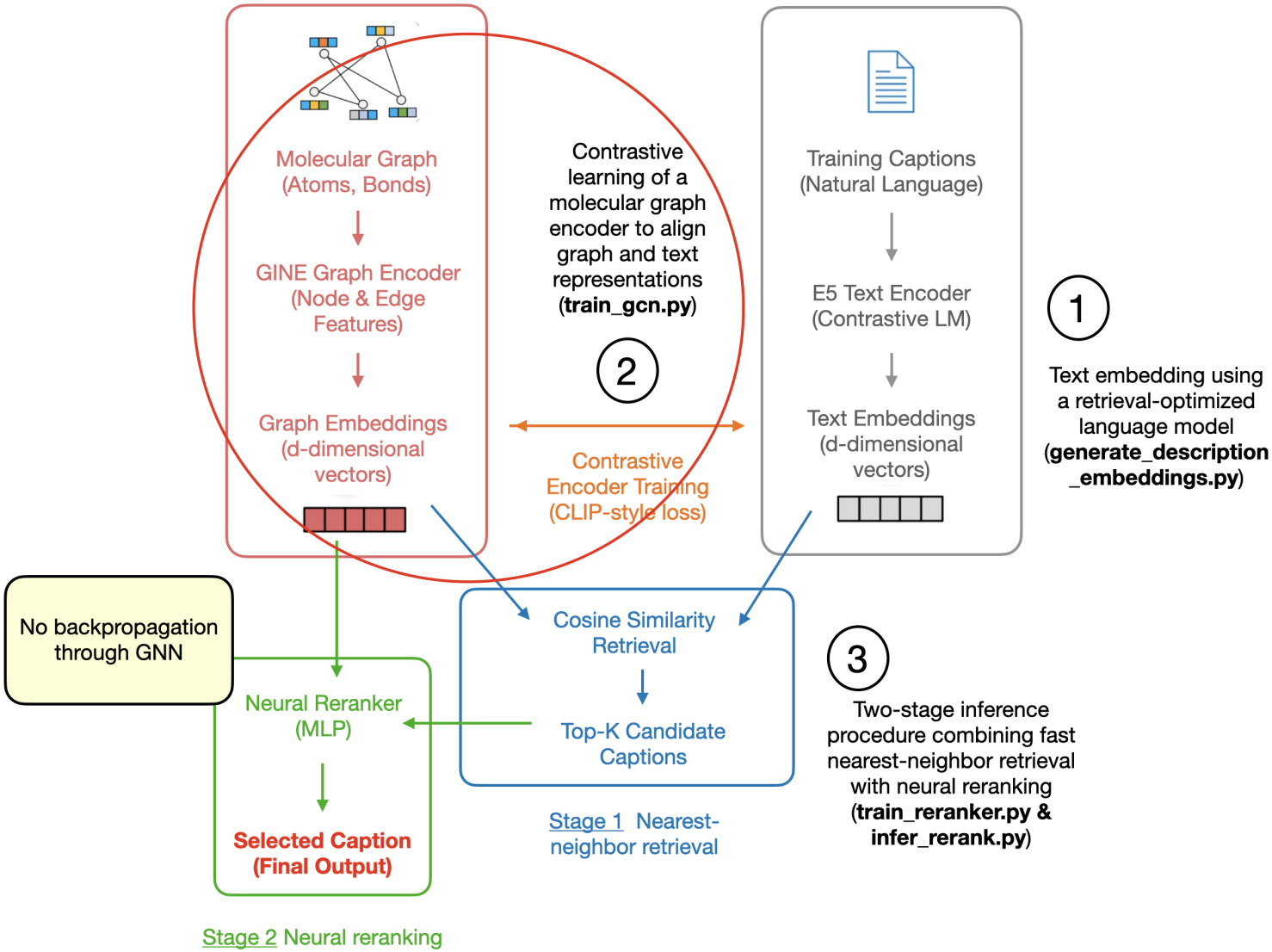
Retrieval-Based Method (1) - Text Representation



Retrieval-Based Method (1) - Text Representation

- Baseline: BERT `[CLS]` embeddings
- Limitation: not optimized for similarity-based retrieval
- We use `intfloat/e5-base-v2`
  - Contrastively trained for dense retrieval
  - Better cosine similarity structure
- Mean pooling + ℓ_2 normalization
- Text embeddings are **precomputed and frozen**

Retrieval-Based Method (1) - Molecular Graph Encoder



Retrieval-Based Method (1) - Molecular Graph Encoder

- Baseline: simple GCN without atom / bond features
- Limited chemical expressiveness

- We use a **GINE** architecture (Graph Isomorphism Network with Edge features)
- Explicitly incorporates:
  - atom features
  - bond features (type, stereochemistry, conjugation)

- Stacked GINE layers with residual connections and layer normalization
- Attention-based pooling for graph-level embeddings
- Projection head + ℓ_2 normalization

Retrieval-Based Method (1) - Contrastive Graph–Text Alignment

- Baseline uses MSE loss $\rightarrow$ weak separation of negatives
- We adopt a **CLIP-style contrastive loss**

$$\mathcal{L}_i = -\log \frac{\exp(\text{scale}\,\langle g_i, t_i \rangle)}{\sum_{t \in \{t_1,\ldots,t_B\} \cup \mathcal{Q}} \exp(\text{scale}\,\langle g_i, t \rangle)}$$

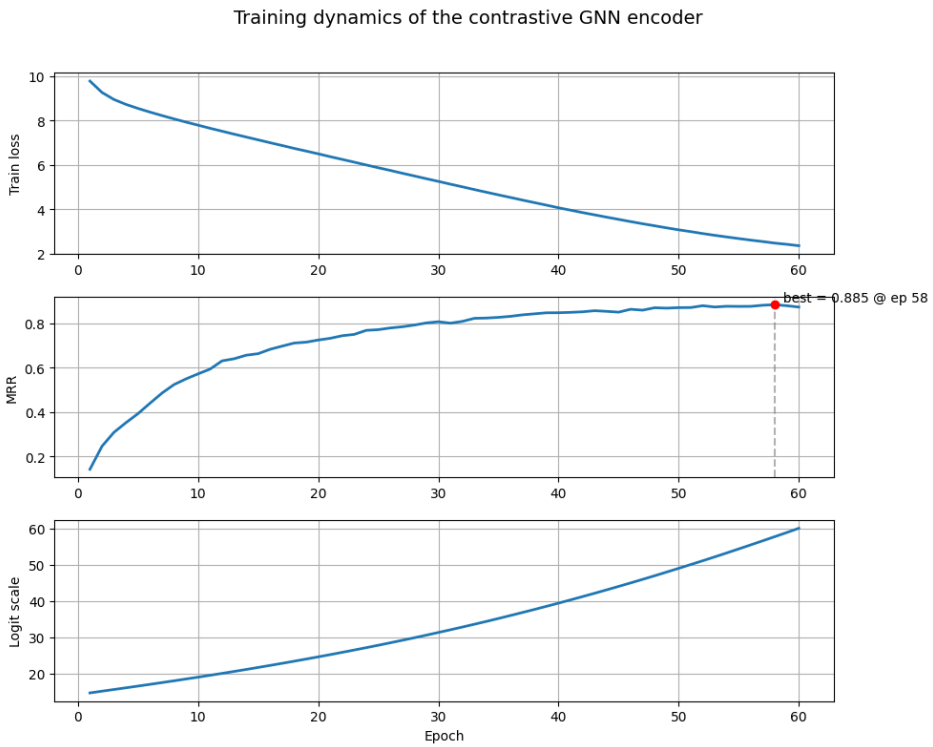
- Memory queue $\mathcal{Q}$ increases number of negatives
- Learnable temperature controls similarity sharpness

Retrieval-Based Method (1) - Training and Model Selection

- Graph encoder trained with contrastive loss
- Validation set used only for model selection
- Evaluation metric: **Mean Reciprocal Rank (MRR)**

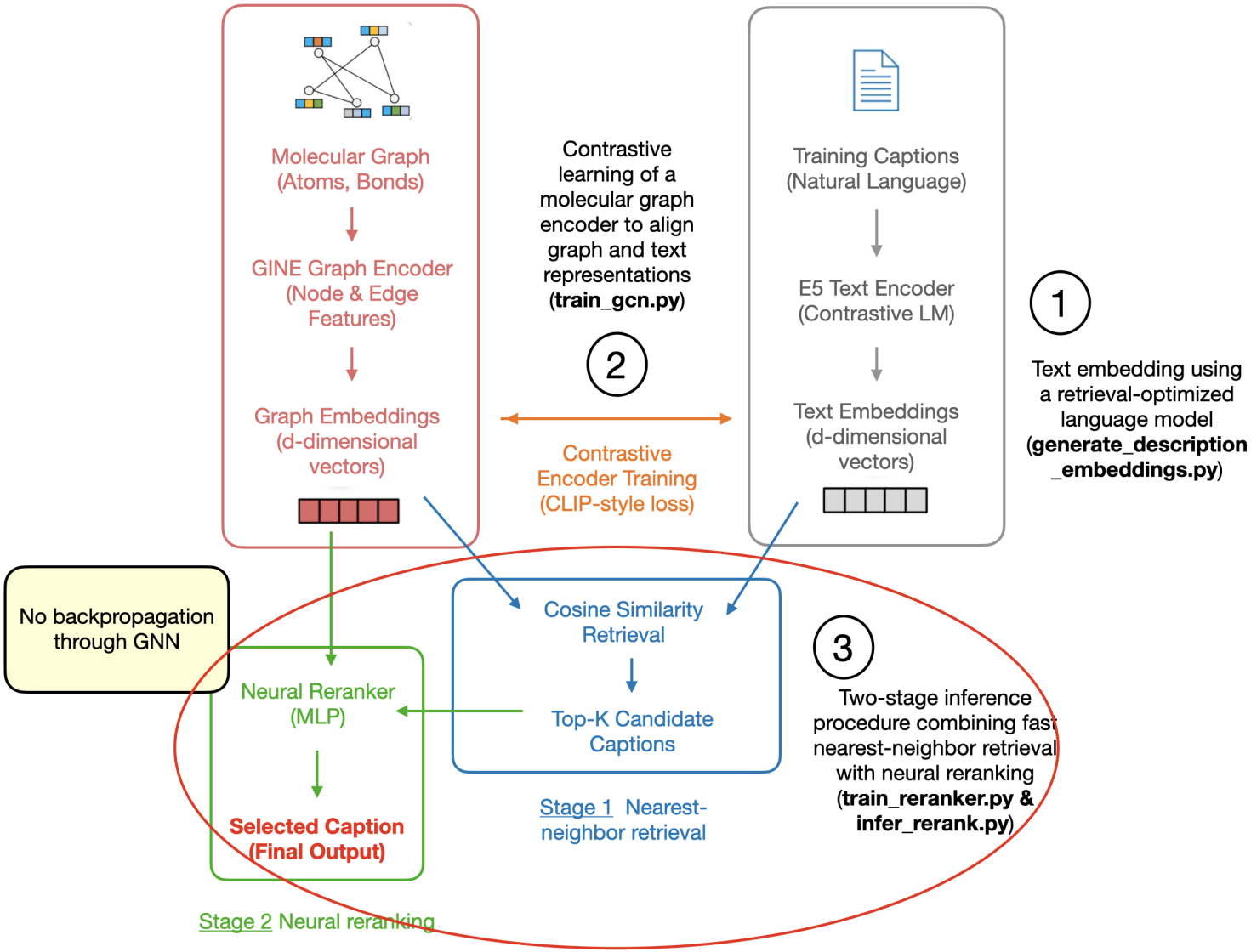
$$\mathrm{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{r_i}$$

- Training details:
  - 5 GINE layers, hidden dim 512
  - AdamW, lr 2×10^{-4}, batch size 256, Memory queue of 65k text embeddings



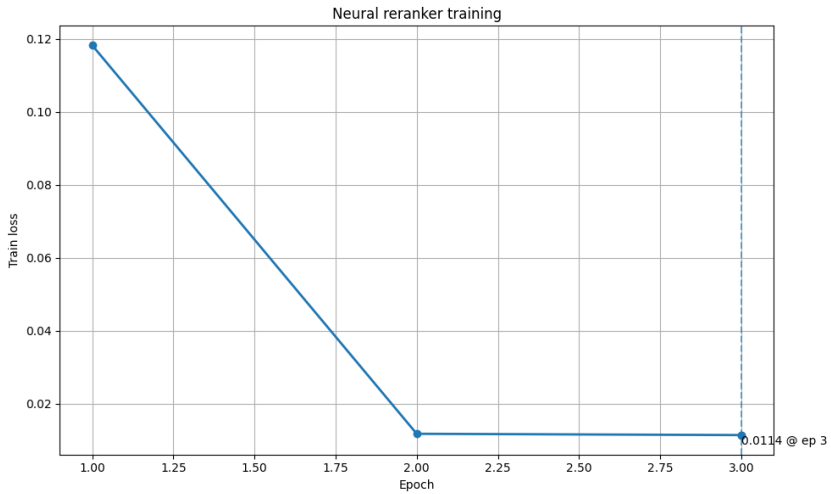
Training dynamics of the contrastive graph encoder

Retrieval-Based Method (1) - Two-Stage Inference



Retrieval-Based Method (1) - Two-Stage Inference

- **Stage 1: Retrieval**
  - Embed query graph
  - Retrieve top-K captions by cosine similarity
- **Stage 2: Neural reranking**
  - Reranker trained on frozen embeddings
  - Listwise cross-entropy loss
  - Improves fine-grained ranking among similar captions
- Final caption = highest reranker score



Training loss of the neural reranker

Retrieval-Based Results

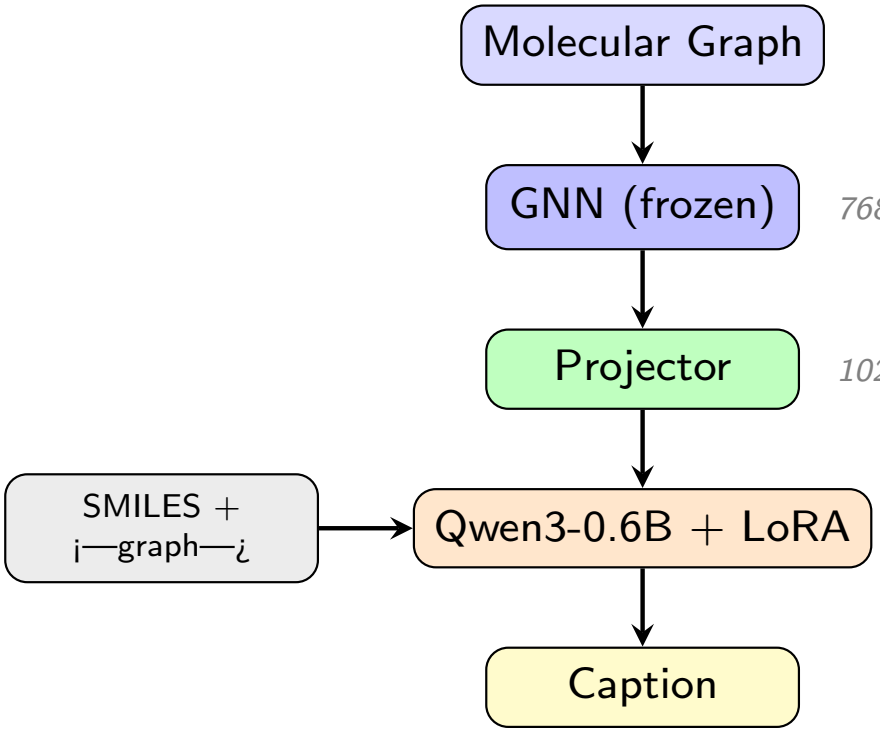
- **Evaluation (Kaggle)**
  - Composite metric:
    - BLEU-4 F1 (lexical overlap)
    - BERTScore (semantic similarity, RoBERTa)
  - Final score aggregates lexical + semantic similarity
- **Leaderboard performance**
  - Baseline (provided): ≈ 0.487 (public)
  - Our retrieval pipeline (contrastive + reranking):
    - **0.661** (public leaderboard)
    - **0.682** (private leaderboard)
- **Why does it work?**
  - Retrieval-optimized text embeddings (E5)
  - Chemically expressive graph encoder (GINE + edge features)
  - Contrastive alignment with many hard negatives (memory queue)
  - Two-stage inference with neural reranking

End-to-End Generation

End-to-End Generation (2) - Overview

- **Goal**: Generate novel captions directly from molecular graphs
  - Not restricted to training corpus captions
- **Architecture**: GNN $\rightarrow$ Projector $\rightarrow$ LLM
  - Reuse frozen GNN encoder from retrieval
  - Fine-tune LLM with LoRA adapters
- **Key challenge**: Limited compute (Google Colab free tier)

End-to-End Generation (2) - Architecture



- **Graph Encoder**: Frozen GINE (768-d)
- **Projector**: 3-layer MLP (768→1024)
- **LLM**: Qwen3-0.6B with LoRA ($r = 64$)
- **Single-Token Injection**
  - Graph → **one** <|graph|> token
  - Unlike vision: no spatial patches
  - Retrieval validates: single embedding captures graph semantics

End-to-End Generation (2) - Input Format

SMILES Representation

- Text-based molecular encoding
- ASCII: atoms, bonds, connectivity

Example:

`Nc1ncnc2c1ncn2C1OC(COP(=O)([O-])...`

Token Injection

- <|graph|> embedding replaced with projected graph
- LLM sees both text + graph

Chat Template

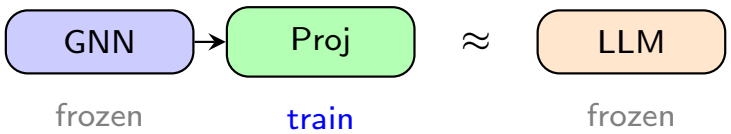
```
<|im_start|>system
You are an expert chemist. Describe the
molecule's properties concisely.<|im_end|>
<|im_start|>user
Molecule Structure: <|graph|>
SMILES: NC(=NCCCC(N)C(=O)O)NP(=O)(O)O<|im_end|>
<|im_start|>assistant
<think>
</think>
The molecule is an arginine derivative...
```

Future: Chain-of-thought via <think> tags for complex molecules

End-to-End Generation (2) - Two-Stage Training

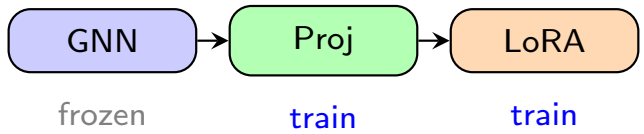
Stage 1: Alignment

- **Loss**: MSE (graph → text embedding)
- **Train**: Projector only
- **Freeze**: GNN + LLM
- **LR**: 10^{-3}



Stage 2: Supervised Fine-Tuning

- **Loss**: Cross-entropy (captions)
- **Train**: Projector + LoRA
- **Freeze**: GNN + LLM base
- **LR**: 10^{-4} / 10^{-5} (LoRA)



End-to-End Generation (2) - Generated Examples

Example: Amino tetrasaccharide

Ground truth: "The molecule is an amino tetrasaccharide comprised of an L-iduronic acid residue, sulfated on O-2, a glucuronic acid residue, and two N-acetylated galactosamine residues, one of which is at the reducing end and both of which are sulfated on O-4. It is an intermediate in the chondroitin sulfate degradation pathway. It has a role as a mouse metabolite."

Generated: "The molecule is an oligosaccharide acid that is the 4-deoxy derivative of the sulfate acid derivative of N-acetylglucosamine."

Example: Neolignan glucoside

Ground truth: "The molecule is a neolignan isolated from the whole plant of Bidens parviflora that has been found to inhibit histamine release from the peritoneal exudate mast cells induced by antigen-antibody reaction. It has a role as a metabolite and a histamine antagonist. It is a beta-D-glucoside, an aromatic ether, a neolignan, a polyphenol, a diol and a secondary alcohol."

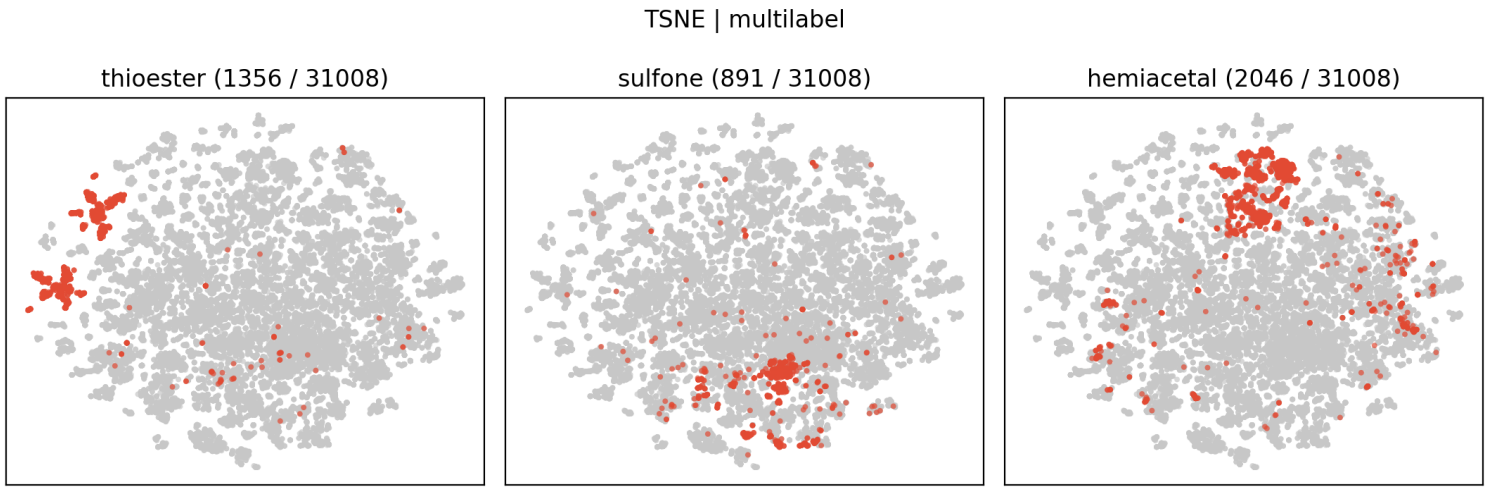
Generated: "The molecule is a member of the class of dihydroxybenzoic acids and a 2-hydroxybenzoic acid. It is a member of dihydroxybenzoic acids and a 2-hydroxybenzoic acid."

Observations:

- ✓ Learns correct format and chemical vocabulary
- × Some hallucination, repetition in outputs (undertrained)

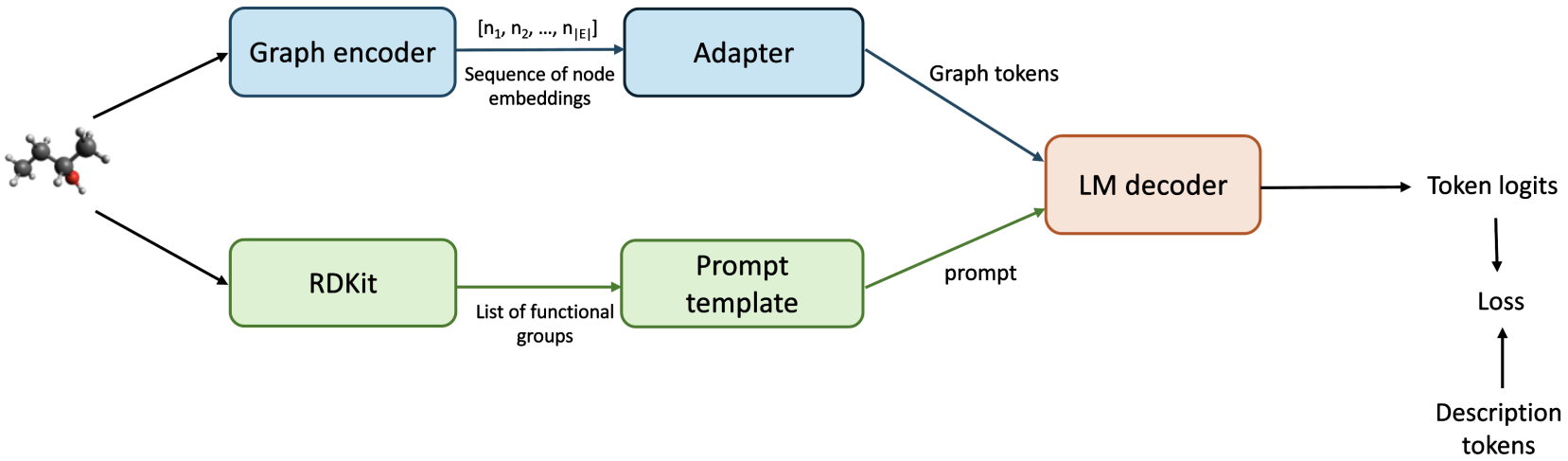
Retrieval-Augmented Generation

Retrieval-Augmented Generation - Functional Group Prompting (3)



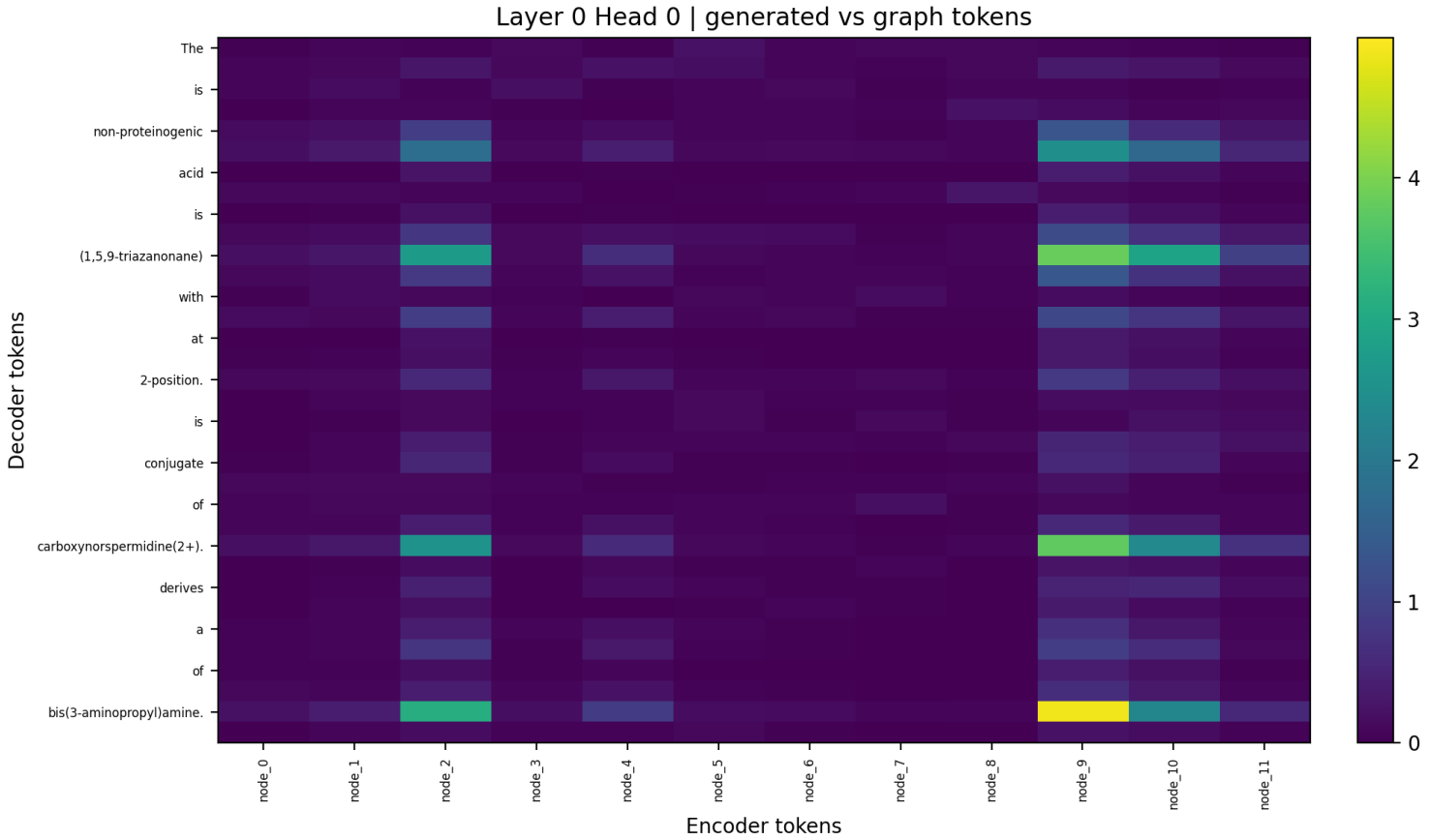
- graph embeddings show functional group clusters
- enhance generation with (very compact!) prompting
- coarse-grained interpretable/stable conditioning

Retrieval-Augmented Generation - Functional Group Prompting (3)

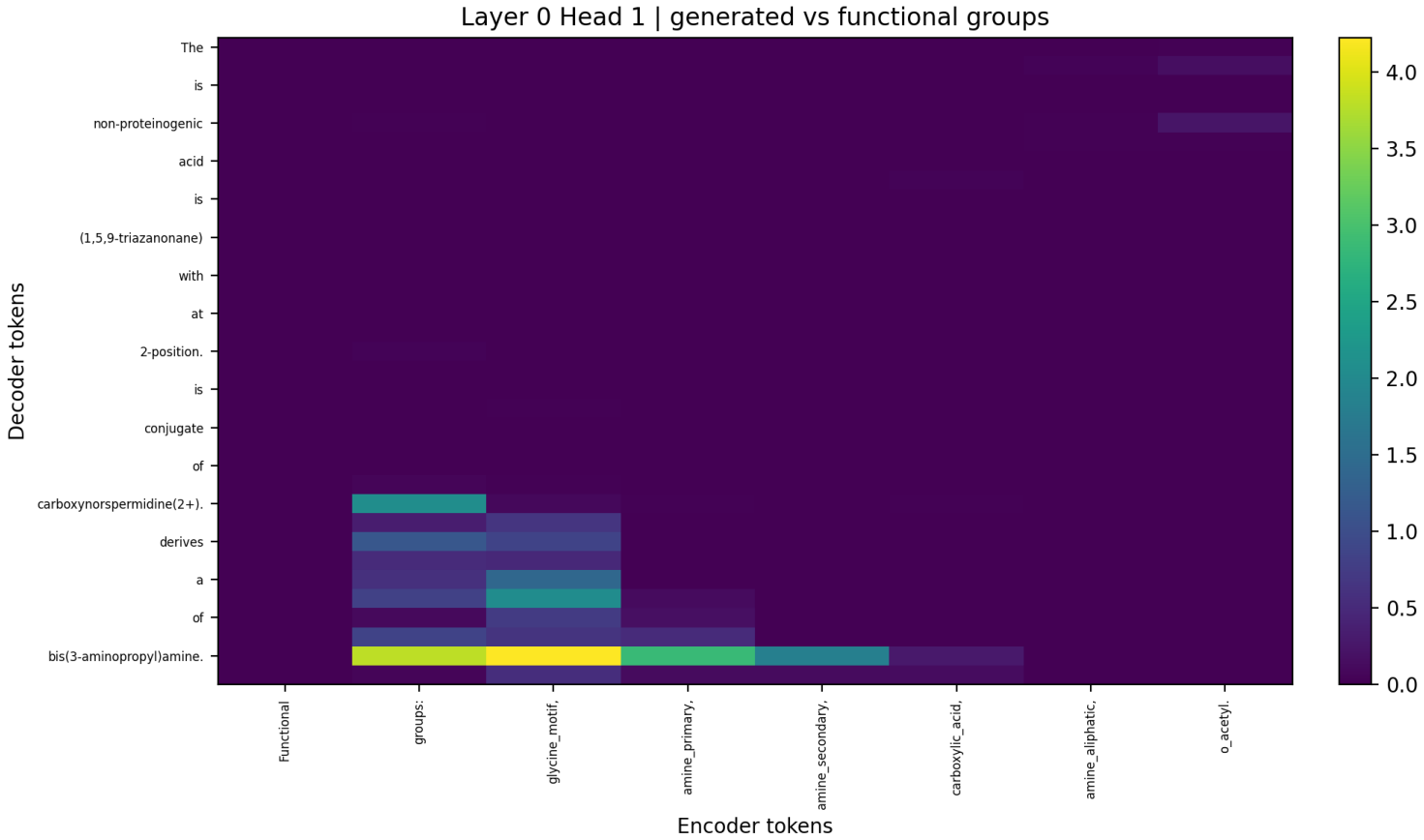


- GINE encoder is frozen
- Only 6 rarest functional groups are included in the prompt

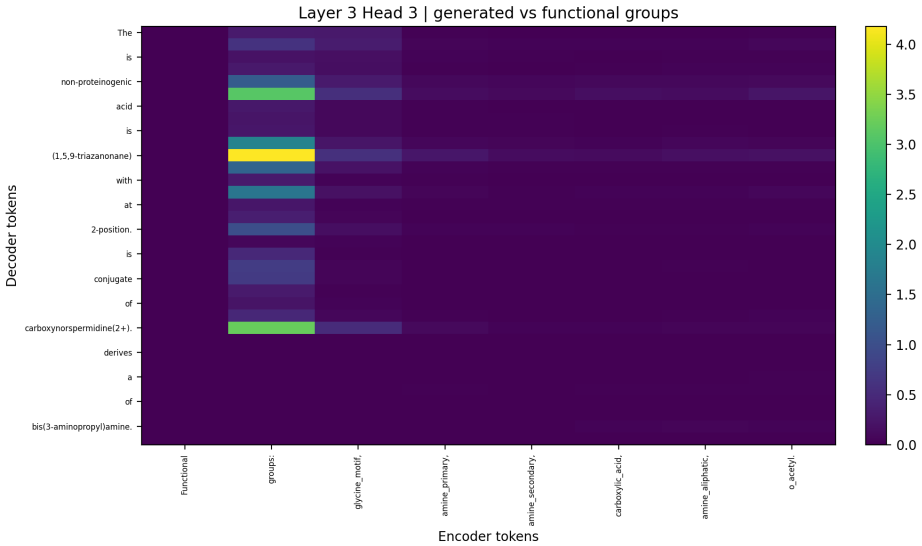
Retrieval-Augmented Generation - Functional Group Prompting (3)

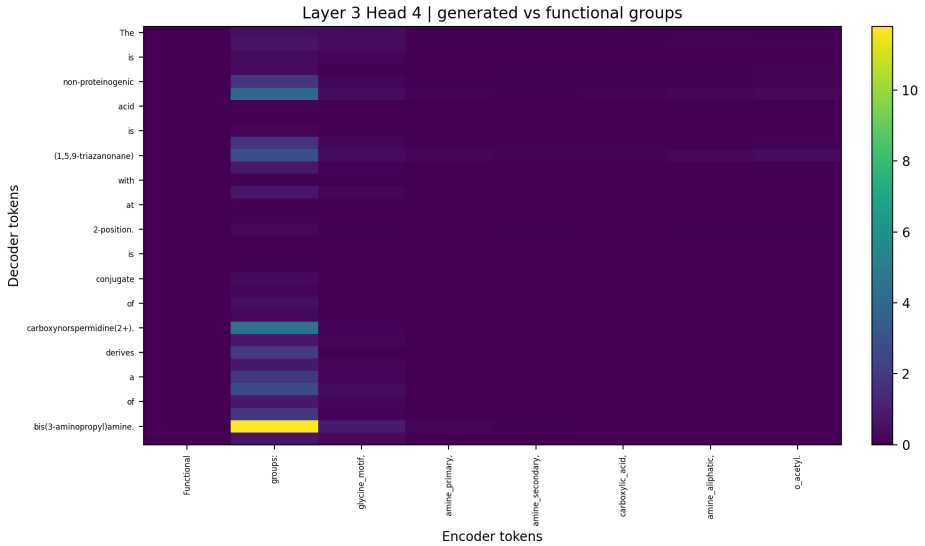


Retrieval-Augmented Generation - Functional Group Prompting (3)

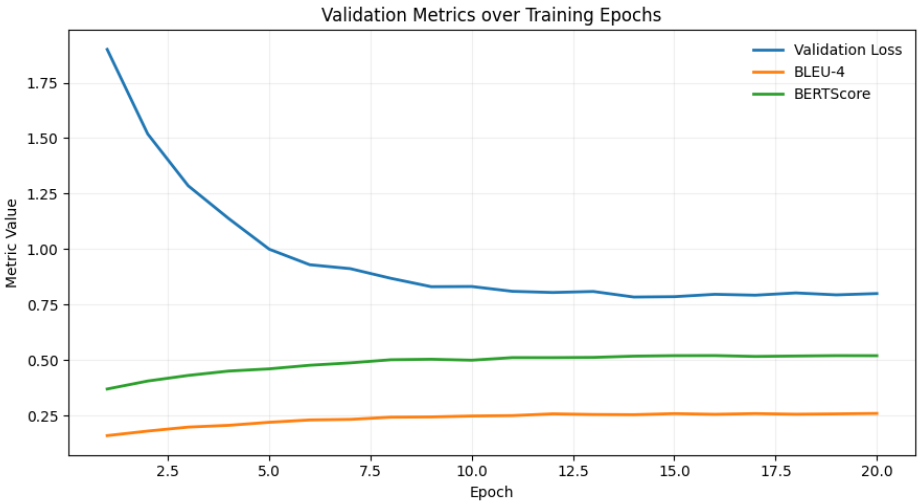


Retrieval-Augmented Generation - Functional Group Prompting (3)





Retrieval-Augmented Generation - Functional Group Prompting (3)



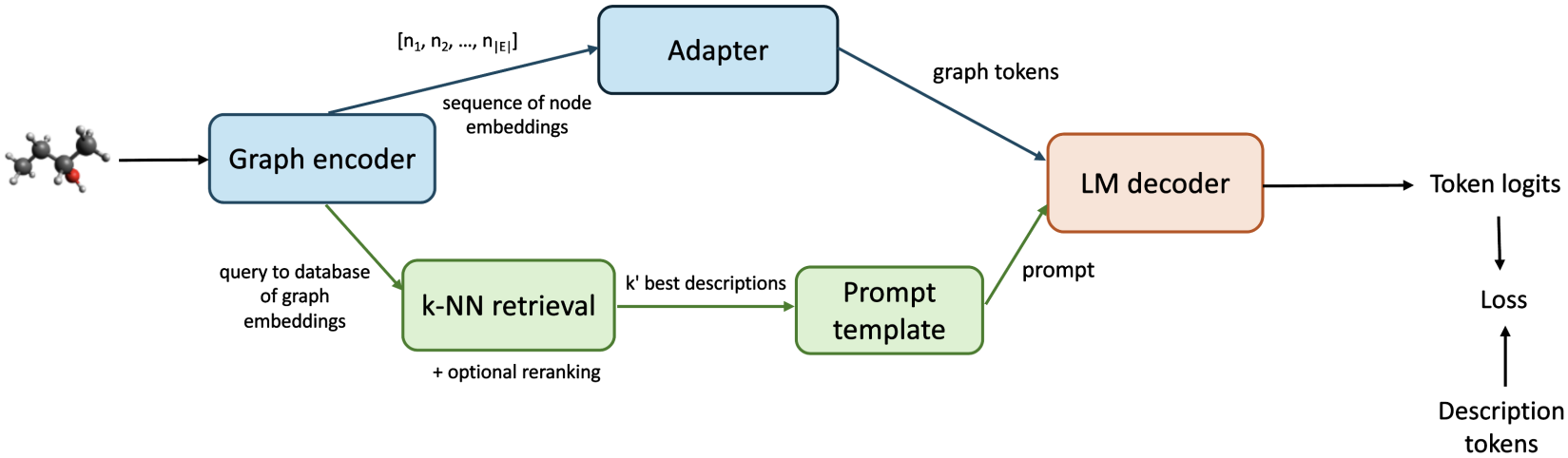
- Stable training
- Generated captions are descent

Example: Amino tetrasaccharide

Ground truth: "The molecule is an amino tetrasaccharide comprised of an L-iduronic acid residue, sulfated on O-2, a glucuronic acid residue, and two N-acetylated galactosamine residues, one of which is at the reducing end and both of which are sulfated on O-4. It is an intermediate in the chondroitin sulfate degradation pathway. It has a role as a mouse metabolite."

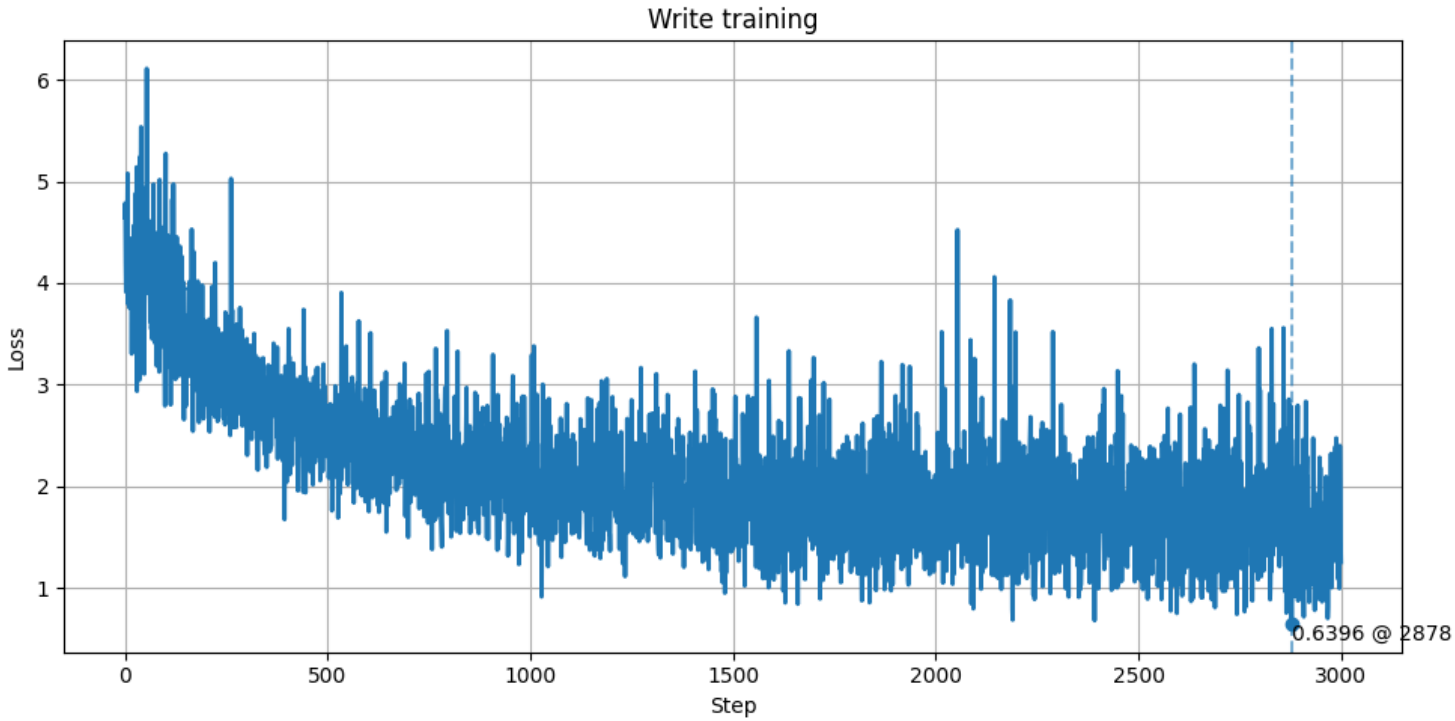
Generated: ""The molecule is an amino trisaccharide consisting of N-acetyl-beta-D-galactosaminyl, 2-acetamido-2-deoxy-alpha-L-threo-hex-4-enopyranuronic acid and an oligosaccharboxylic acid residue joined in sequence by a (1-¿4) glycosidic bond."

Retrieval-Augmented Generation - Top-K Description Prompting (4)



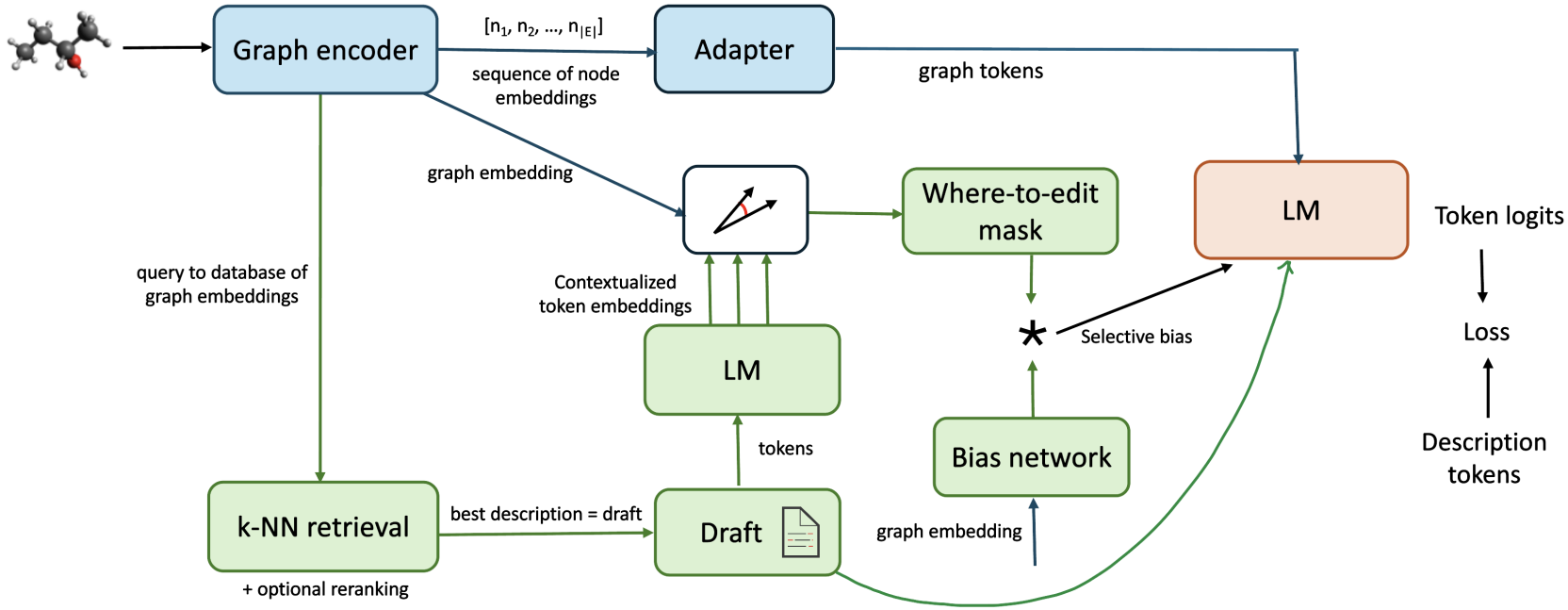
- Prompting makes the training very compute-intensive (it scales quadratically!)
- More fine-grained prompt, but may contain misleading facts...

Retrieval-Augmented Generation - Top-K Description Prompting (4)



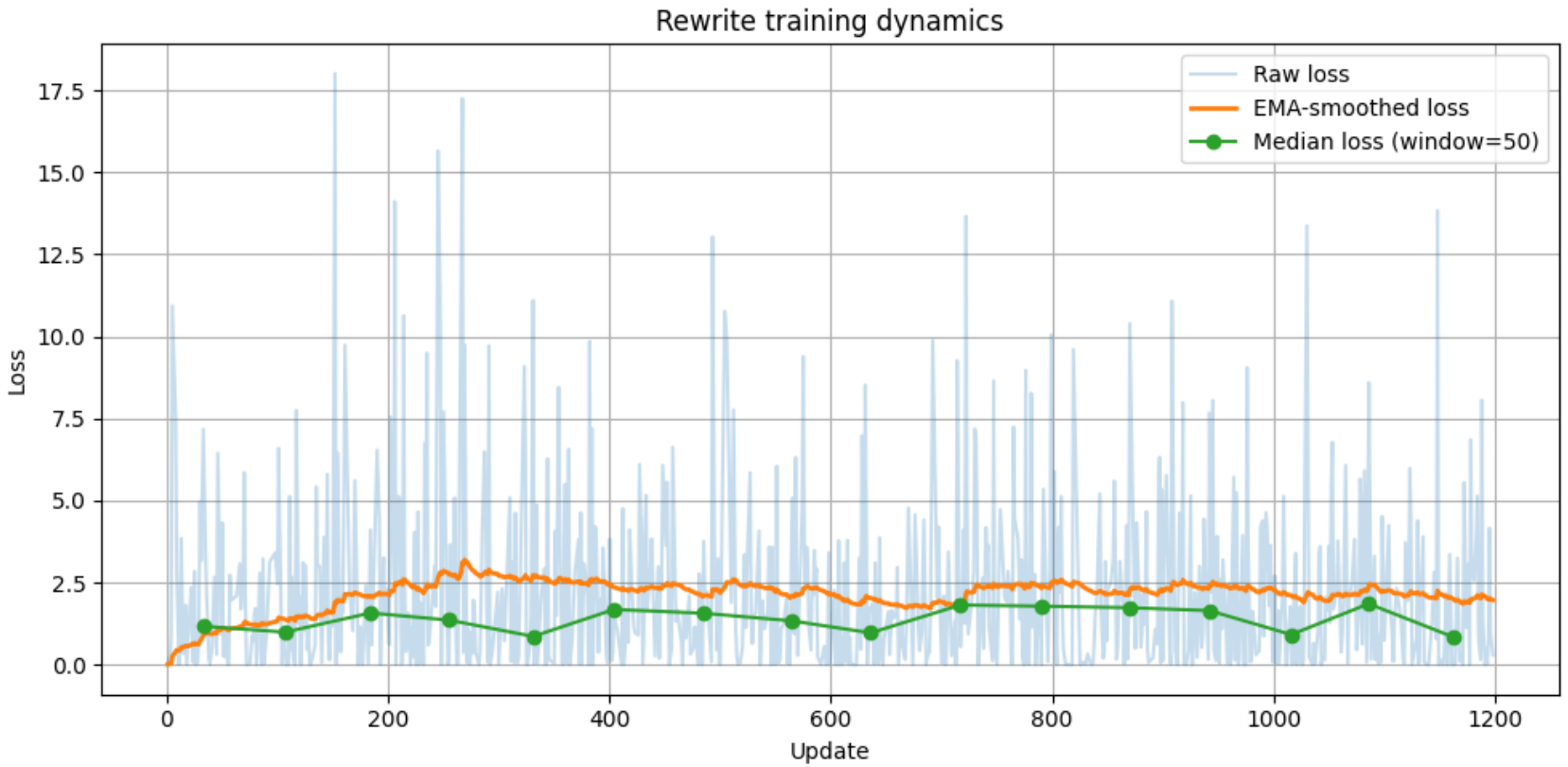
- The loss decreases steadily
- Conditioning on prompt and graph prefix seems to work
- Very small training due to computing limitations

Retrieval-Augmented Generation - Retrieval-Guided Description Rewriting (5)



- Graph encoder is frozen, LMs are frozen. We train adapter and bias network.
- Cross-entropy loss is computed on the where-to-edit tokens

Retrieval-Augmented Generation - Retrieval-Guided Description Rewriting (5)



- Loss is very noisy
- Model seems to be adapting to the new loss signal at the beginning of the training then stabilizes

Generative-Based Results

Method	**Public score**
Baseline (0)	0.487
Retrieval (no reranking)	0.604
Retrieval (with reranking) (1)	**0.661**
End-to-End Generation From Scratch (2)	0.507
Functional-Group Prompting (3)	0.491
Top-K Description Prompting (4)	0.436
Retrieval-Guided Description Rewriting (5)	0.594

Table: Public Kaggle leaderboard scores for the different approaches.

Generative-Based Results

Limitations (generation)

- Google Colab free tier
- Session disconnections
- Could not run full training ($\sim$9h)

Not fundamental

- Architecture is sound
- More compute would help

Conclusion

- Retrieval-based methods dominate under limited compute
- Generative and RAG approaches are constrained by GPU availability
- More GPU budget would enable longer training and stronger generation

Questions

Thank you for your attention.