ALTeGraD Data Challenge

Molecule Retrieval with Natural Language Queries

Team : Baku incorporated

Heni Soula

Omar Bensaid

Souheib Ben Mabrouk

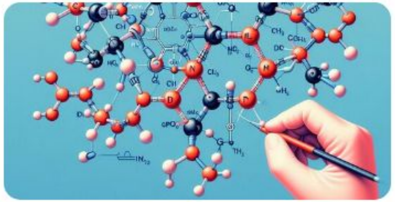
Context (1/2)

Challenge Goal:

Develop a model capable of accurately retrieving specific molecules through text queries

ALTeGraD-2023 Data Challenge

Molecule Retrieval with Natural Language Queries



Context (2/2)

Evaluation metric: **Mean Reciprocal Rank**

Given the rank at which the first relevant item is retrieved, the reciprocal of this rank is calculated, and the average is taken across all queries.

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\mathrm{rank}_i}$$

Table of contents

I. Related Work

DistilBERT

- Distilled version of **BERT** model
- **Efficient** and **compact** transformer-based language representation model
- Reduced **computational complexity** and **memory** requirements

This model was used in the baseline architecture upon which we built our final model.

SciBERT

- **Specialized** pre-trained language model based on the BERT architecture
- Trained on a vast and diverse corpus of scientific publications, totaling 1.14 million papers
- Training data covers domains like biomedicine (82%) and computer science (12%), making SciBERT adept at processing **molecular properties**

Graph Convolutional Networks

Initial "layer 0" embeddings are equal to node features

$$\boxed{\mathbf{h}_v^0 = \mathbf{x}_v}$$

previous layer embedding of *v*

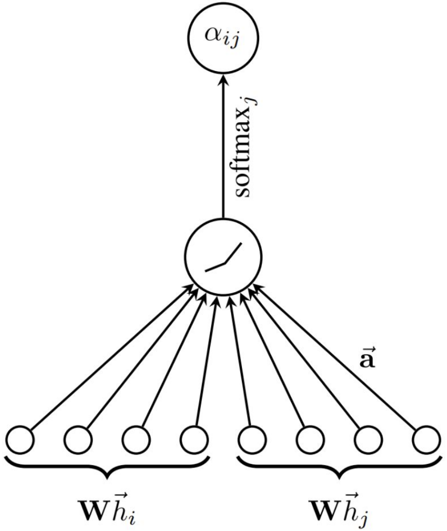
$$\boxed{\mathbf{h}_v^k} = \boxed{\sigma}\left(\mathbf{W}_k \boxed{\sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|}} + \mathbf{B}_k \boxed{\mathbf{h}_v^{k-1}}\right), \quad \forall k > 0$$

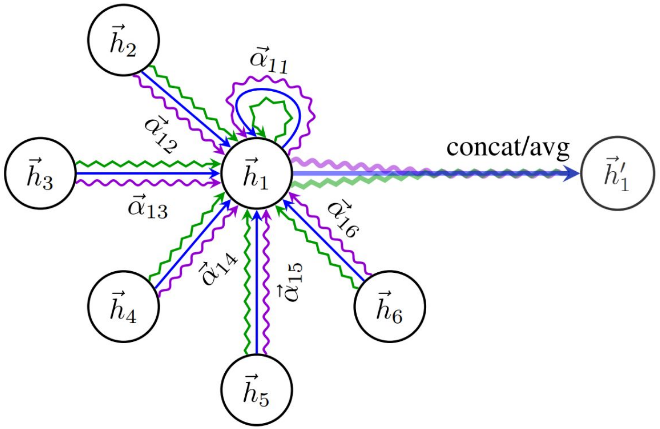
kth layer embedding of *v*

non-linearity (e.g., ReLU or tanh)

average of neighbor's previous layer embeddings

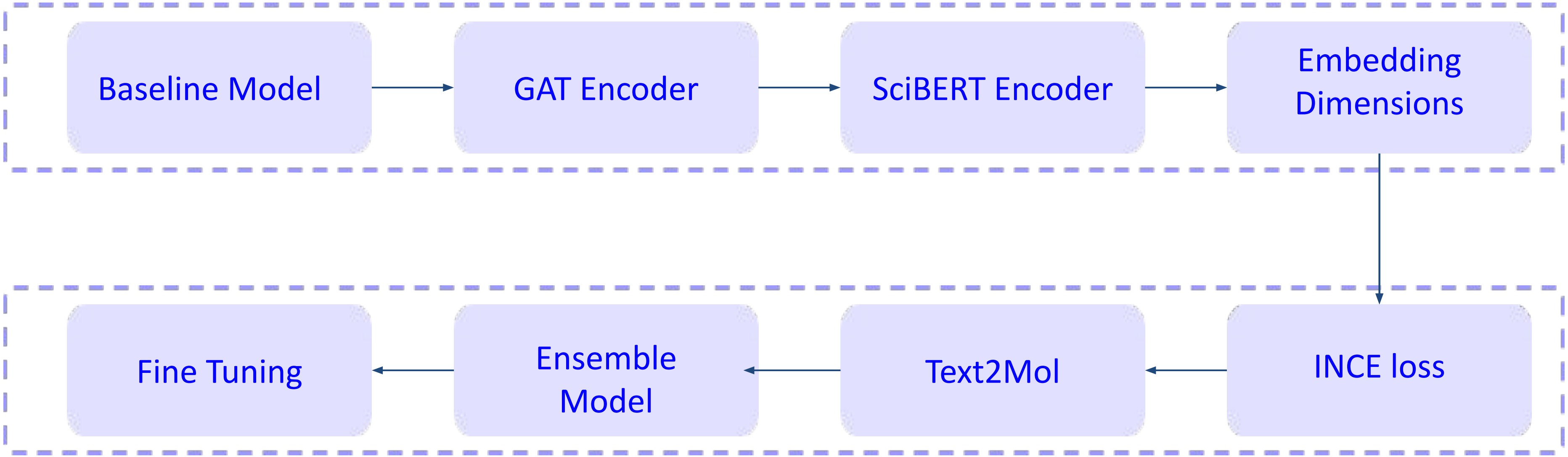
Graph Attention Networks





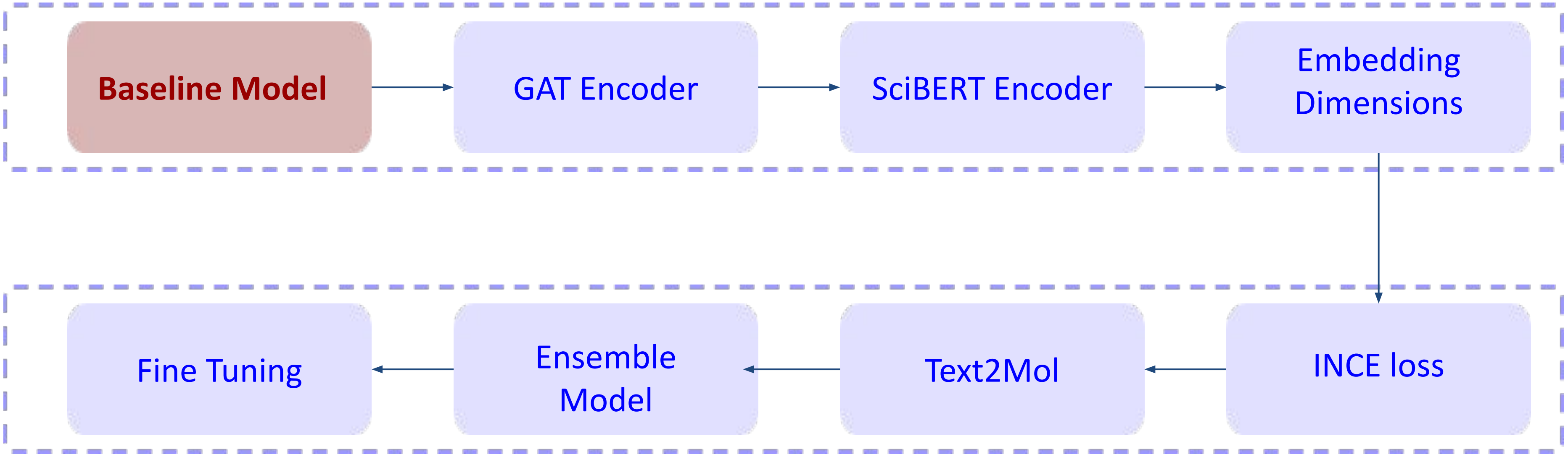
II. Models

Models



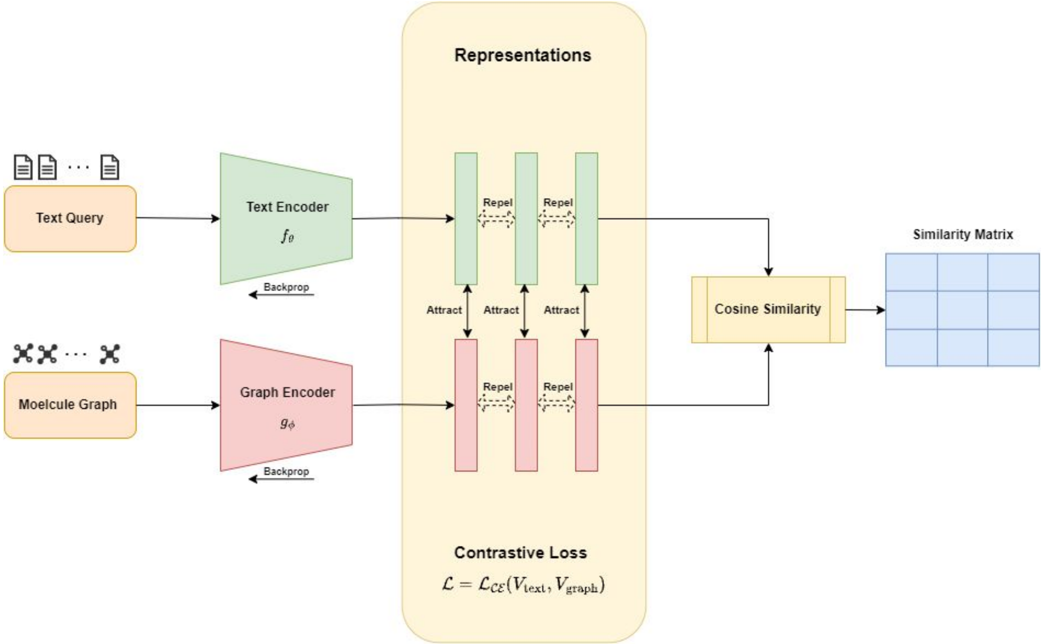
Baseline Model
GAT Encoder
SciBERT Encoder
Embedding Dimensions
INCE loss
Text2Mol
Ensemble Model
Fine Tuning

Models

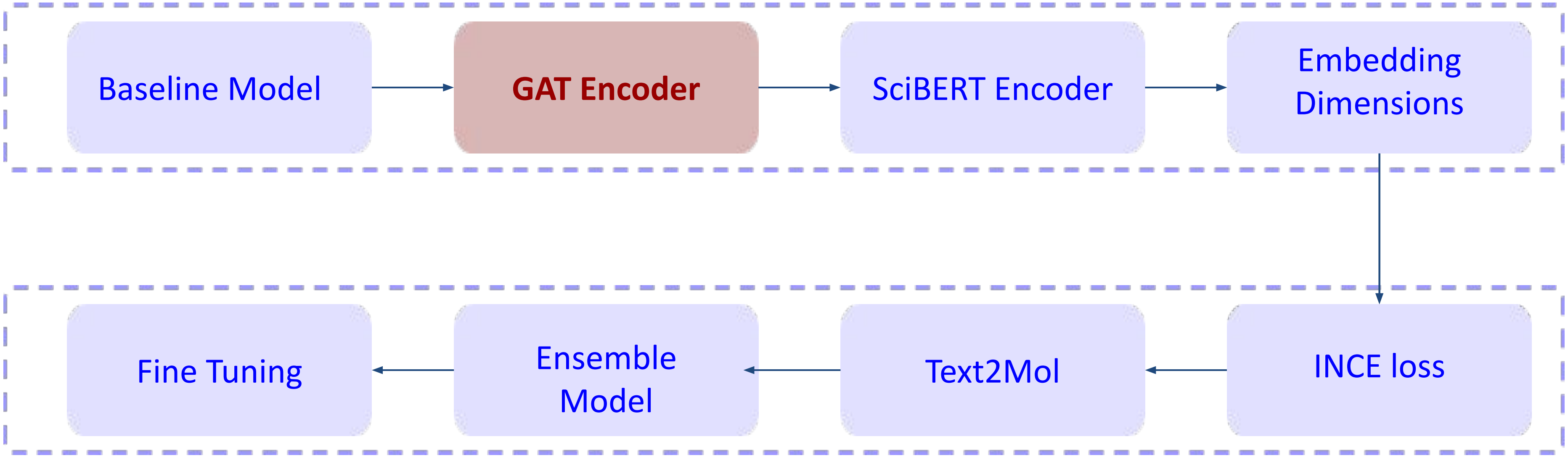

Baseline Model
GAT Encoder
SciBERT Encoder
Embedding Dimensions
INCE loss
Text2Mol
Ensemble Model
Fine Tuning

Baseline model

- Employed **DistilBERT** as the text encoder and **GCN** as the graph encoder
- Trained for **5 epochs**
- Minimized the **contrastive loss** between the representations
- Reaches an MRR score of **0.348**



Models



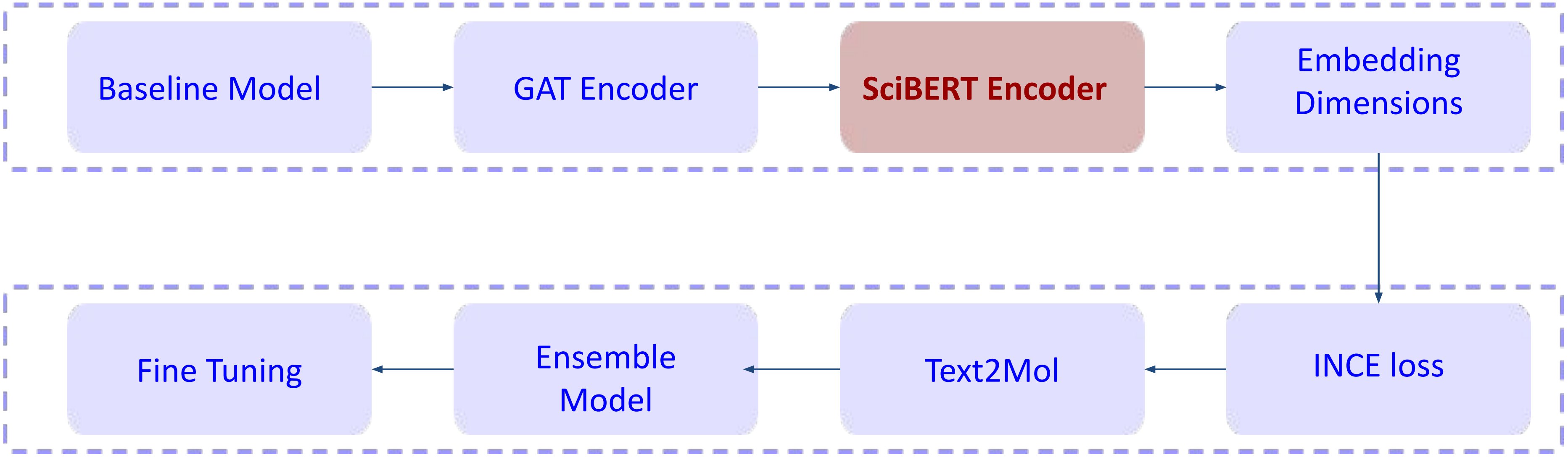
Baseline Model
GAT Encoder
SciBERT Encoder
Embedding Dimensions
INCE loss
Text2Mol
Ensemble Model
Fine Tuning

GAT Encoder

- The substitution of the GCN encoder with a GAT encoder.
- Motivated by the capacity of GATs in **capturing complex relationships** within graph-structured data by applying **self-attention** mechanism on the nodes.
- Chosen GATv2 encoder which used a **dynamic graph attention mechanism** and is more expressive than GAT.
- Trained for **50 epochs** to minimize the **contrastive loss** between the representations.
- Improved the baseline model and achieved an MRR score of **0.5086.**

Model	MRR
Baseline	0.348
GAT	0.5086

Models

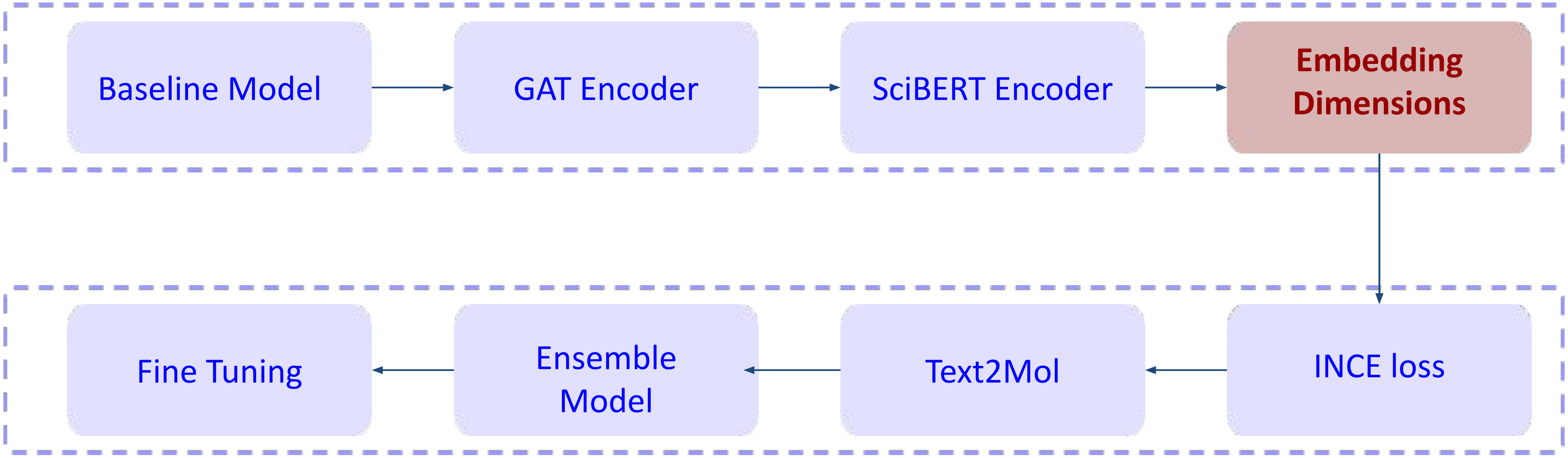

Baseline Model
GAT Encoder
SciBERT Encoder
Embedding Dimensions
INCE loss
Text2Mol
Ensemble Model
Fine Tuning

SciBERT Encoder

- Replaced the DistilBERT text encoder with an **uncased sci vocab SciBERT** encoder.
- Pertained on a data containing 82% **biomedicine** papers -> well-suited for processing texts describing molecular properties
- Trained for **100 epochs** to minimize the contrastive loss between the representations.
- Reaches an MRR score of **0.72**

Model	MRR
Baseline	0.348
GAT	0.5086
GAT and SciBERT	0.72

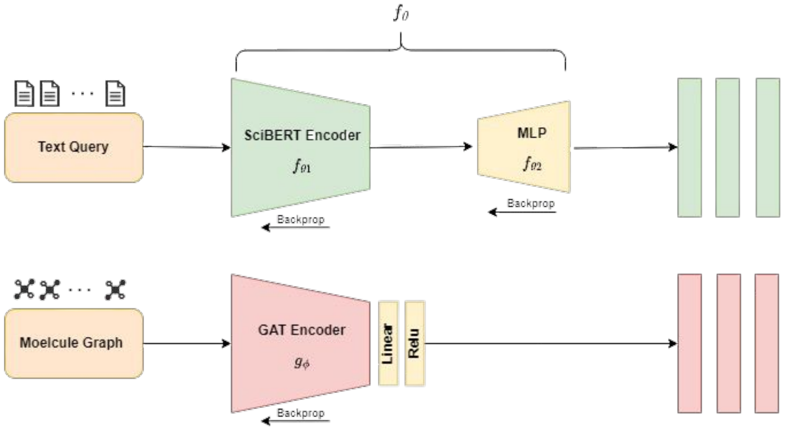
Models



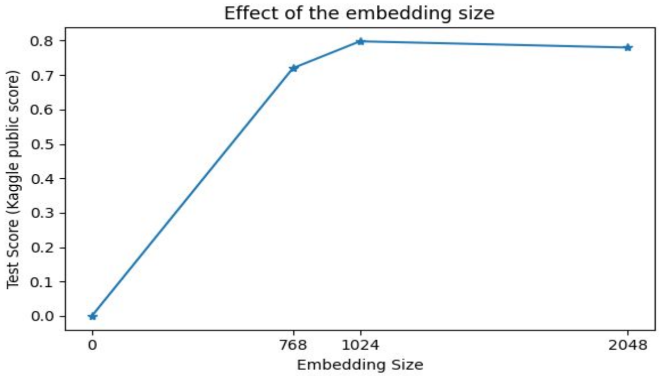
Baseline Model
GAT Encoder
SciBERT Encoder
Embedding Dimensions
INCE loss
Text2Mol
Ensemble Model
Fine Tuning

Higher Dimensional Embeddings

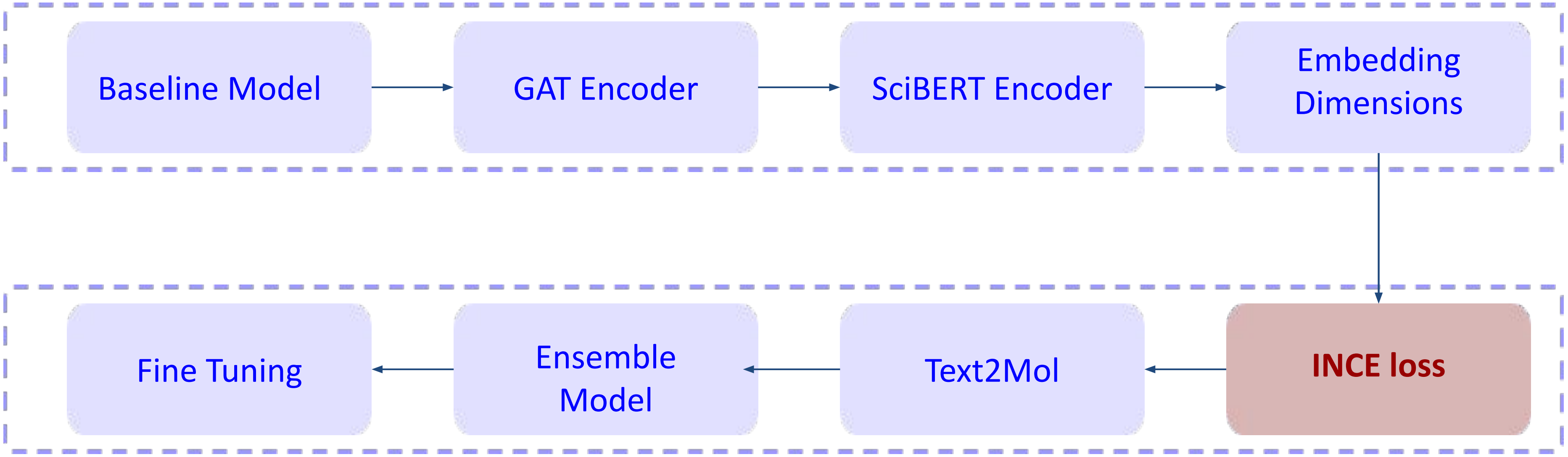
- Used a **richer** embedding space of dimension **1024**
- Appended a Multi-Layer Perceptron (**MLP**) at the end of each encoder
- Achieved an MRR score of **0.798**



Model	MRR
Baseline	0.348
GAT	0.5086
GAT and SciBERT	0.72
High-Dim	0.798



Models


Baseline Model
GAT Encoder
SciBERT Encoder
Embedding Dimensions
INCE loss
Text2Mol
Ensemble Model
Fine Tuning

Information-Noise-Contrastive Estimation

- Optimizes the negative log probability of classifying the **positive** sample correctly.

$$\mathcal{L}_{\text{INCE}} = -\mathbb{E}\left[\log \frac{f(\mathbf{x}, \mathbf{c})}{\sum_{\mathbf{x}' \in X} f(\mathbf{x}', \mathbf{c})}\right]$$

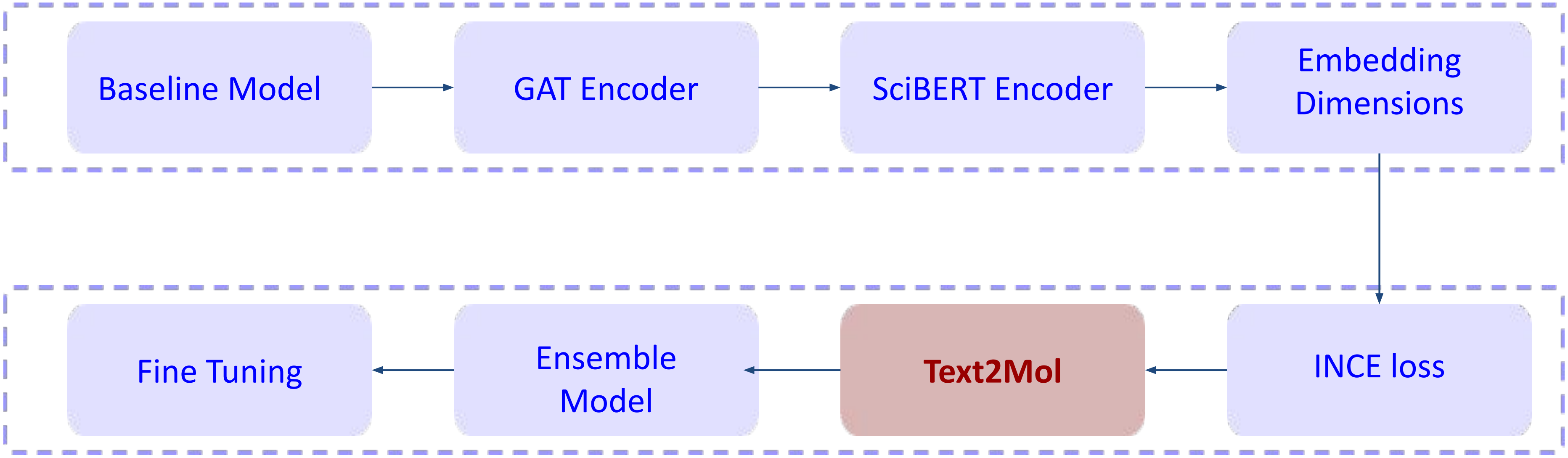
- At each step we **incorporated** the embeddings of the **previous batches** as **negative samples.**
- Memory length = 2 × $batch\_size$ to avoid converging to an **uninformative local minimum**
- Enhanced the MRR score to **0.87**.

Model	MRR
Baseline	0.348
High-Dim	0.798
INCE Loss	0.87

Algorithm 2 Custom Contrastive Loss

Input: v_1: Text embeddings, v_2: Graph embeddings, m: Memory size
Initialize: memory buffers m_{text} and m_{graph} to None; loss functions: cross-entropy *ce* and InfoNCE *ince*
Function RESETMEMORY():
  Reset m_{text} and m_{graph} to None
Function FORWARD(v_1, v_2):
  Calculate InfoNCE loss on v_1 and v_2 with m_{graph} as negatives
  Calculate InfoNCE loss on v_2 and v_1 with m_{text} as negatives
  Combine losses:
$result$ = INFONCE($v_1, v_2, m_{\text{graph}}$) + INFONCE($v_2, v_1, m_{\text{text}}$)
  Update memory:
    If $m > 0$:
      If m_{text} is None:
        Set m_{text} to v_1 and m_{graph} to v_2
      Else:
        Append v_1 to m_{text} and v_2 to m_{graph}
        Keep only last m elements in m_{text} and m_{graph} if exceeded
        return *result*

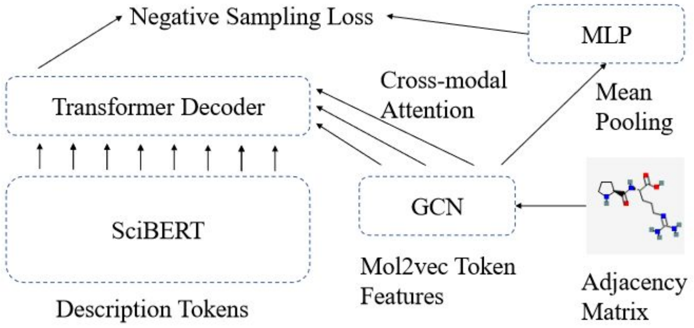
Models



Baseline Model
GAT Encoder
SciBERT Encoder
Embedding Dimensions
INCE loss
Text2Mol
Ensemble Model
Fine Tuning

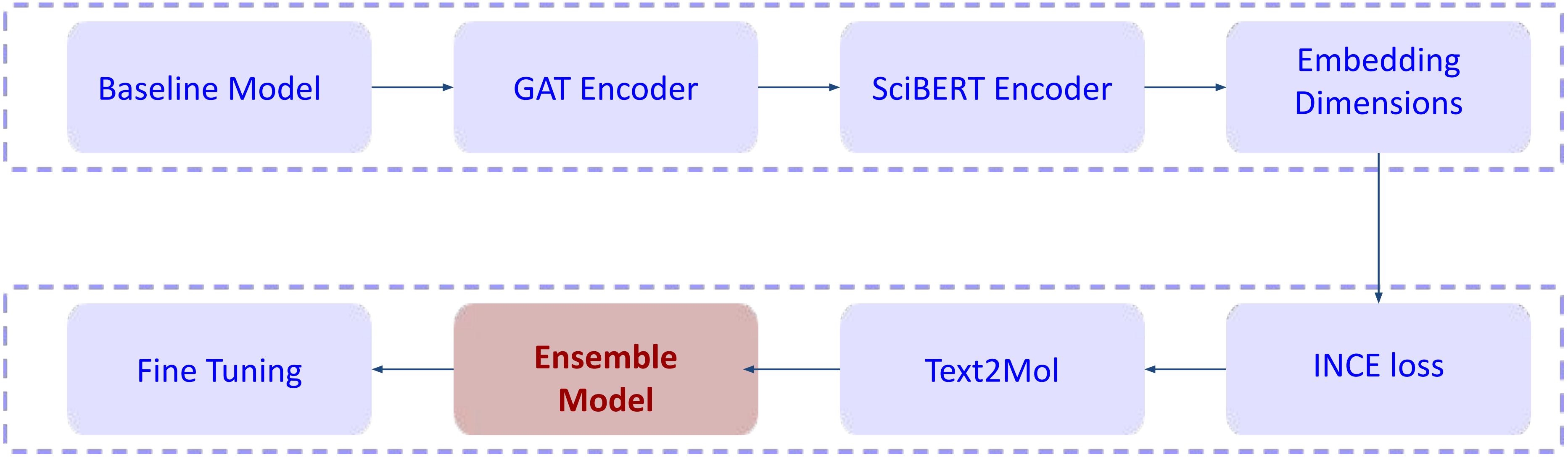
Text2Mol

- Implemented **Text2Mol** framework using the architecture from Edwards et al. 2021.
- Uses **cross-modal attention mechanism** connecting text and graph embeddings.
- Potential **information leakage** from the graph embeddings which was mitigated through **negative sampling**.
- The model was interesting to explore however we didn't use it for the final results due to:
  - weak performance of **MRR** score **0.03**
  - the need to compute the text embedding associated with each individual graph.



The Text2Mol architecture for the cross-modal attention extension and association rules

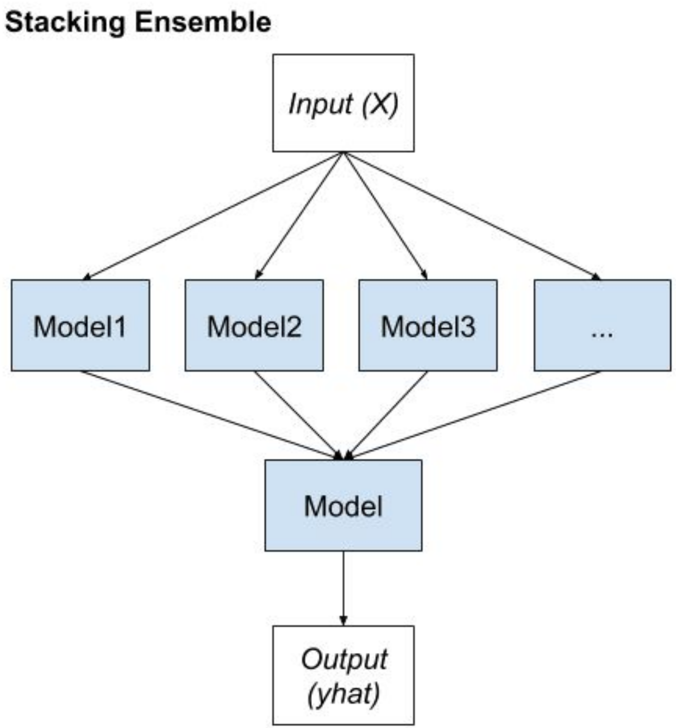
Models


Baseline Model
GAT Encoder
SciBERT Encoder
Embedding Dimensions
INCE loss
Text2Mol
Ensemble Model
Fine Tuning

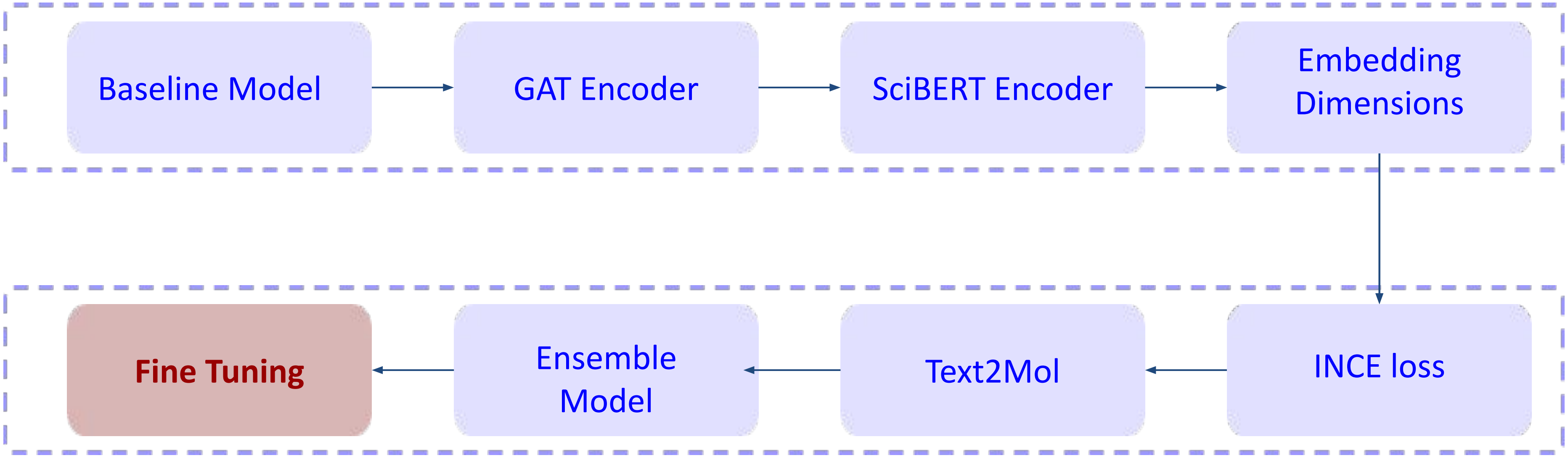
Ensemble model

- Noticed that our model weren’t overfitting so there is a possibility they can act as “strong learners”.
- Implemented naive stacking methods such as the arithmetic and geometric mean of the outputs of different models.
- Improved the result greatly breaking the 0.9 barrier for the **MRR** score 0.92

Model	MRR
Baseline	0.348
High-Dim	0.798
INCE Loss	0.87
Ensemble Model	0.9216



Models



Baseline Model
GAT Encoder
SciBERT Encoder
Embedding Dimensions
INCE loss
Text2Mol
Ensemble Model
Fine Tuning

Fine tuning

For the final training run we:

1. Started from the **best model** obtained so far.
2. **Merged** validation set with the training set to train on the entire dataset.
3. Linear extension of the **memory depth** at intervals of 10 fine-tuning periods, ranging from **2 to 10** times the batch size.
4. Monitoring the performance of the model through **sampled 5000 observations**.
5. Selecting the model with the highest **MRR** score on the sampled validation set : 0.9223

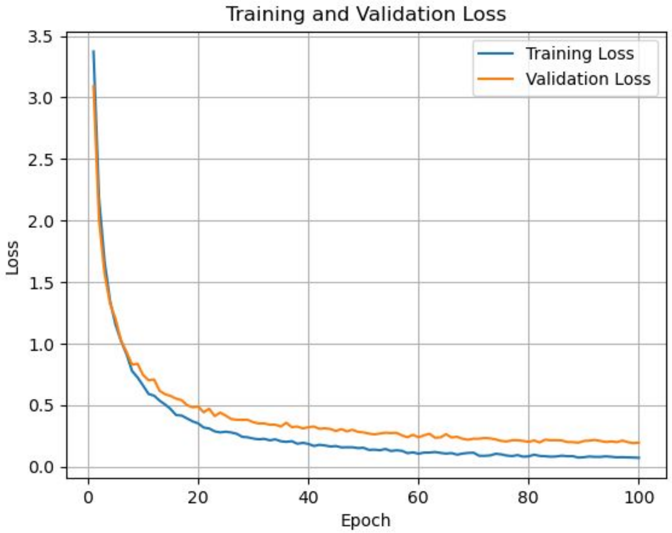
III. Results & Discussions

III. Results & Discussions:

Model	MRR
Baseline	0.348
GAT	0.5086
GAT and SciBERT	0.72
High-Dim	0.798
INCE Loss	0.87
Text2Mol	0.03
Ensemble Model	0.9216
Fine Tuned	**0.9223**

III. Results & Discussions:

- The model didn't overfit :
  1. The model isn't expressive enough
  2. The data is uniformly distributed across training, validation and test



III. Results & Discussions:

- Training took approximately **12 hours.**
- The training time was pretty much the same for all models.
- The **bottleneck** was the data loader

IV. Conclusion & Perspective

IV. Conclusion & Perspective:

- The best model was the fine-tuned Scibert + GAT with the modified InfoNCE.
- Exploring more pooling methods for Text and Graphs.
- Results seem to be too promising which might necessitate validating them over other datasets.

References (1/2)

- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. (2020). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv:1910.01108 [cs.CL]

- Iz Beltagy, Kyle Lo, and Arman Cohan. (2019). "SciBERT: A Pretrained Language Model for Scientific Text." *arXiv:1903.10676 [cs.CL]

- Shaked Brody, Uri Alon, and Eran Yahav. (2022). "How Attentive are Graph Attention Networks?" *arXiv:2105.14491 [cs.LG]

- Thomas N. Kipf and Max Welling. (2017). "Semi-Supervised Classification with Graph Convolutional Networks." *arXiv:1609.02907 [cs.LG]

- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. (2018). "Graph Attention Networks." *arXiv:1710.10903 [stat.ML]*

References (2/2)

- Carl Edwards, ChengXiang Zhai, and Heng Ji. (2021). "Text2Mol: Cross-Modal Molecule Retrieval with Natural Language Queries." In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing [doi 10.18653]

- https://tjmachinelearning.com/lectures/2122/advanced/Graph_Neural_Networks.pdf

- https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/