Clinical AI Lab

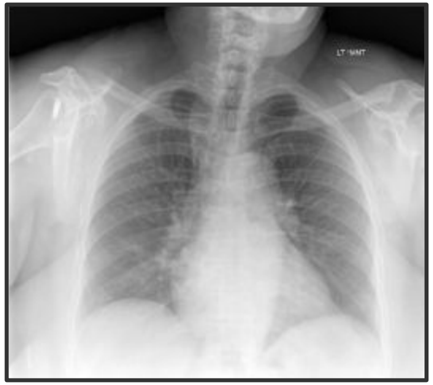
NYU

CENTER FOR ARTIFICIAL INTELLIGENCE AND ROBOTICS

InVERGe: Intelligent Visual Encoder for Bridging Modalities in Report Generation

Reading Group

09.06.205



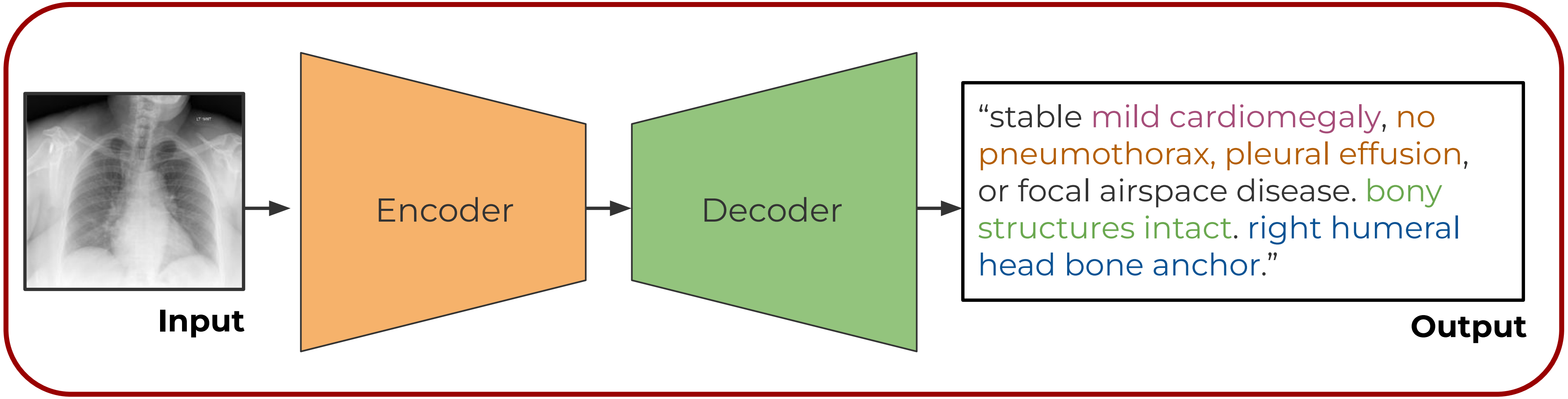
Input

“stable mild cardiomegaly, no pneumothorax, pleural effusion, or focal airspace disease. bony structures intact. right humeral head bone anchor.”

Output

The main objective of this paper is to create clear reports about the image’s contents.

NYU



"stable mild cardiomegaly, no pneumothorax, pleural effusion, or focal airspace disease. bony structures intact. right humeral head bone anchor."

Trained on plenty of image-caption pairs

The main objective of this paper is to create clear reports about the image's contents.

NYU

Problem	Objective
Models have been observed to produce **unnatural language outputs** when trained solely on raw image-text pairs	Introduce a **two-step training methodology** using a layer that bridges the gap between the encoder and decoder to significantly: a) enhance the alignment between vision and language b) reduce the workload on the decoder
High computational demand of end-to-end model training	

NYU

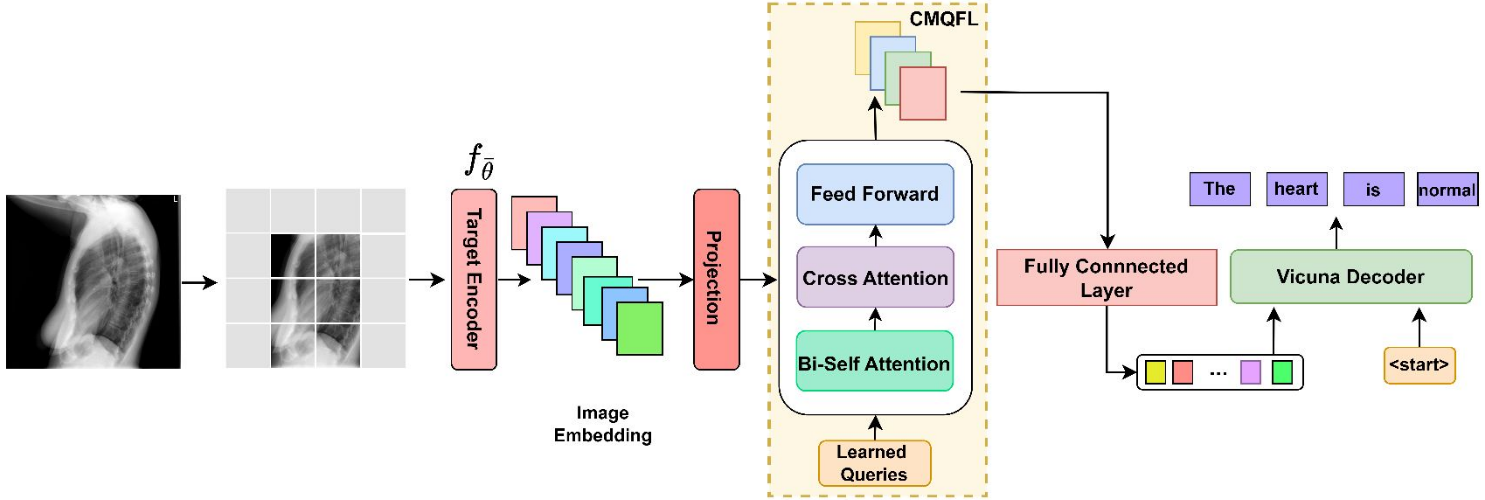
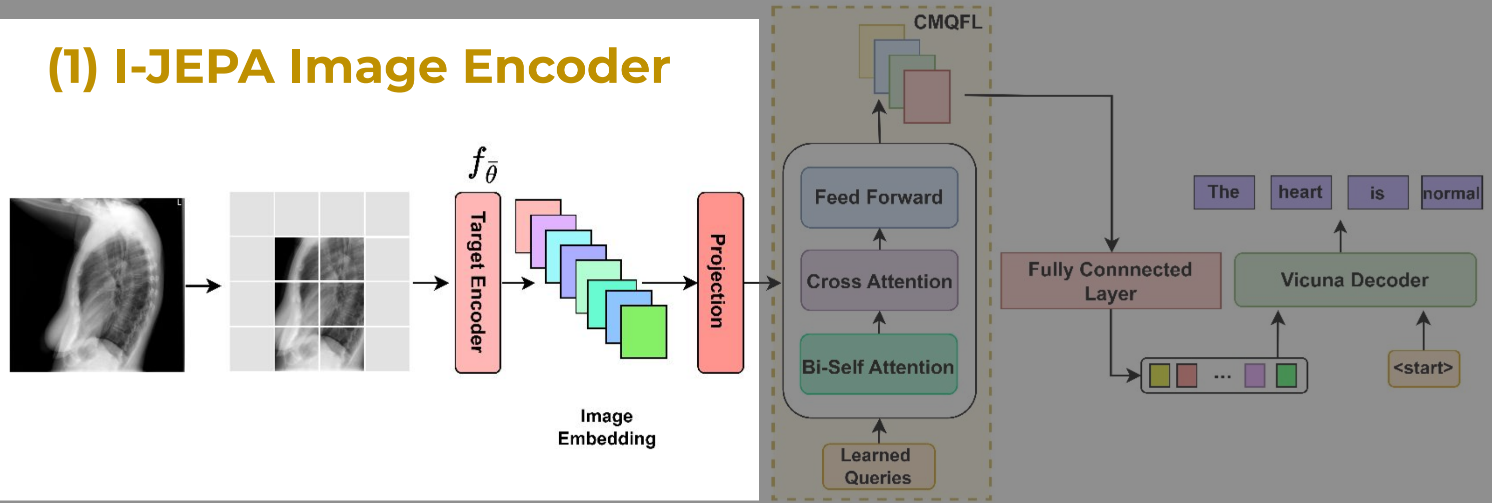
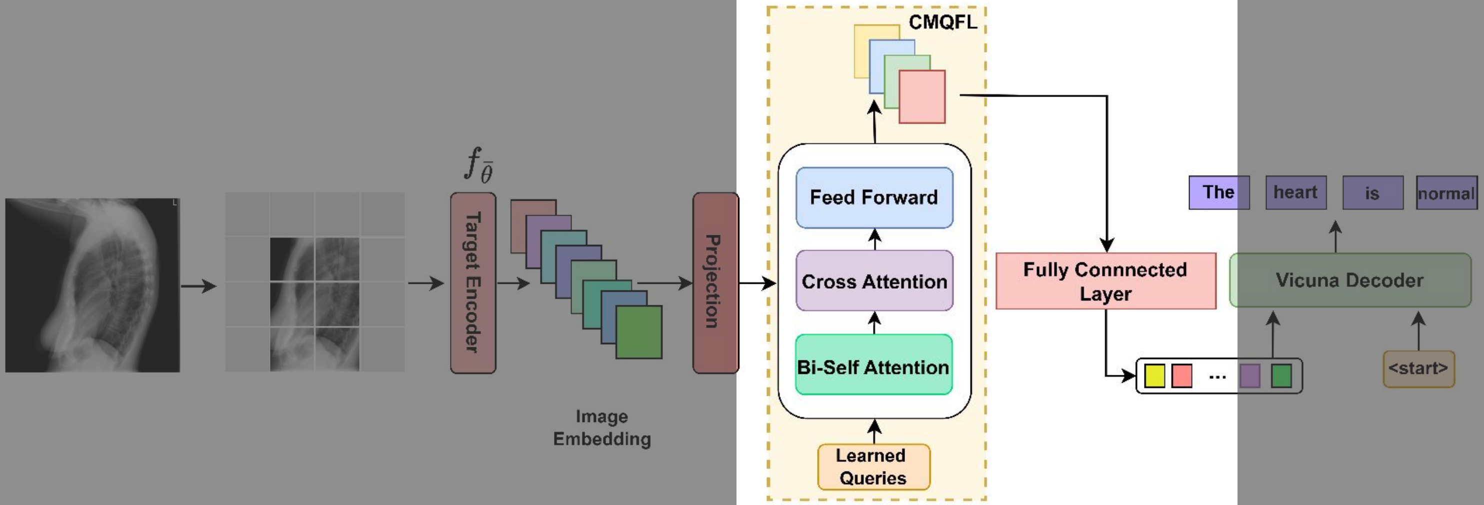


Figure 3. The architecture of the proposed InVERGe approach, includes our encoder trained on self-supervised methods, learnable CMQFL layers, and Vicuna decoder.Specifically, the encoder encodes high-level semantic features and the CMQFL layer uses that information and applies a query that will generate text-grounded image embeddings. Then the decoder uses that embedding to produce a report.

(1) I-JEPA Image Encoder

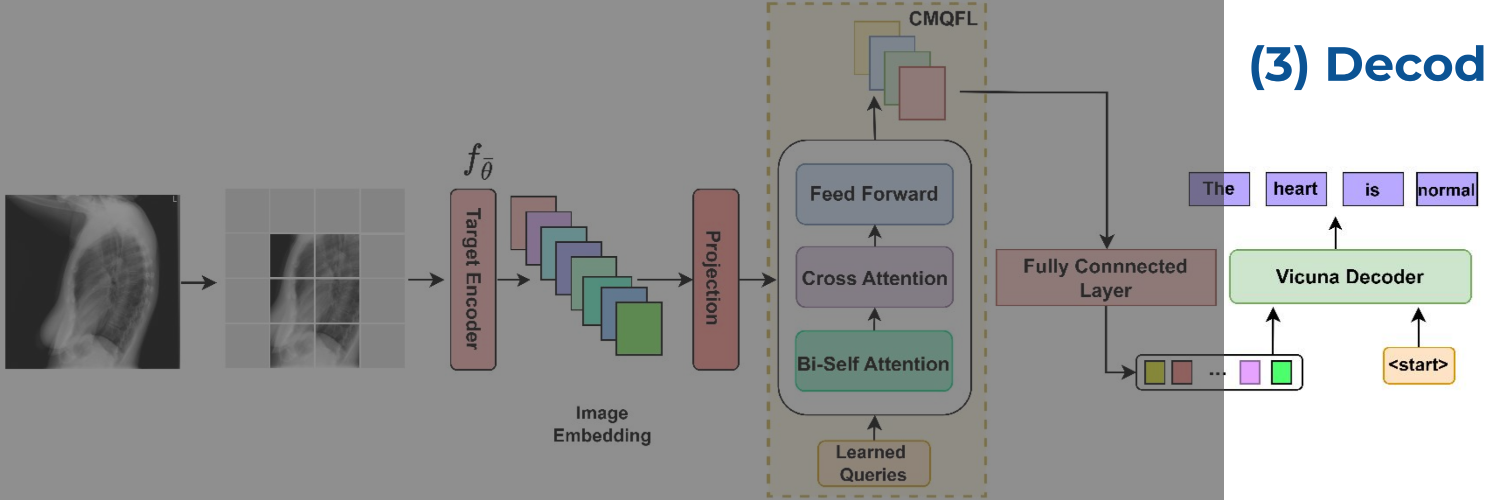


(2) Cross-Modal Query Fusion Layer



Bert-based CMQFL layer responsible for generating text-grounded image embeddings

(3) Decoder



CMQFL
$f_{\bar{\theta}}$
Target Encoder
Image Embedding
Projection
Feed Forward
Cross Attention
Bi-Self Attention
Learned Queries
Fully Connnected Layer
The
heart
is
normal
Vicuna Decoder
...
<start>

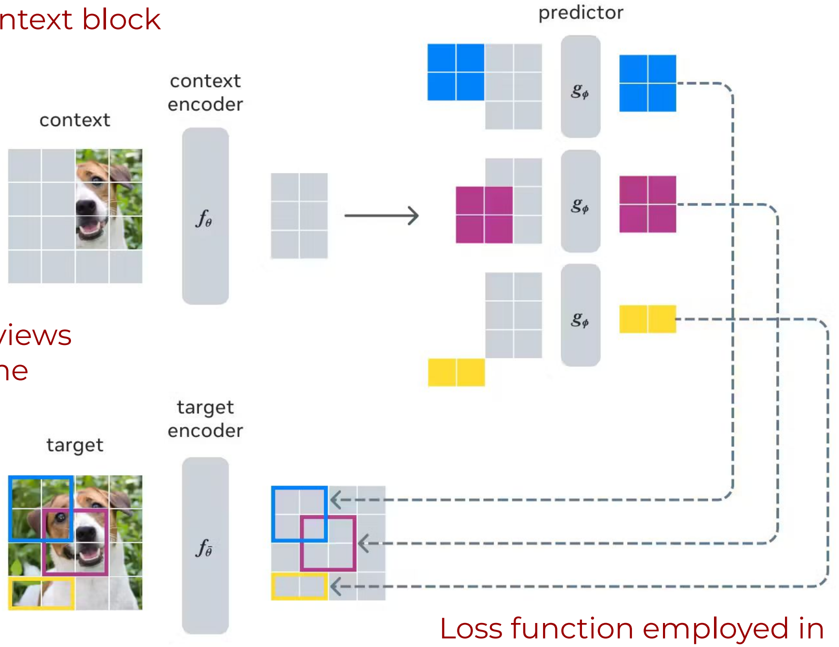
Image Encoder

1/3

What is JEPA?

- Joint Embedding Predictive Architecture
- Predict missing information in an **abstract representation space**

“Spatially distributed” context block



Identical views of the same image

Asymmetric architecture between context and target encoder to avoid representation collapse

Loss function employed in the embedding space, not the input space.

NYU

Why select JEPA as pre-training strategy?

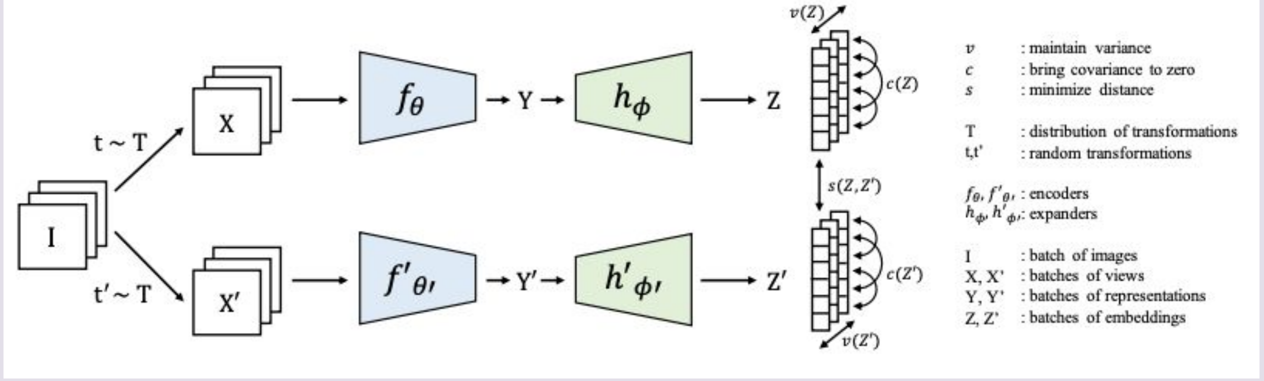
Why select JEPA as pre-training strategy?

Because of perceived issues with other pre-training strategies. Let's take a look at some:

Why select JEPA as pre-training strategy?

Because of perceived issues with other pre-training strategies. Let’s take a look at some:

Invariance-based Pretraining

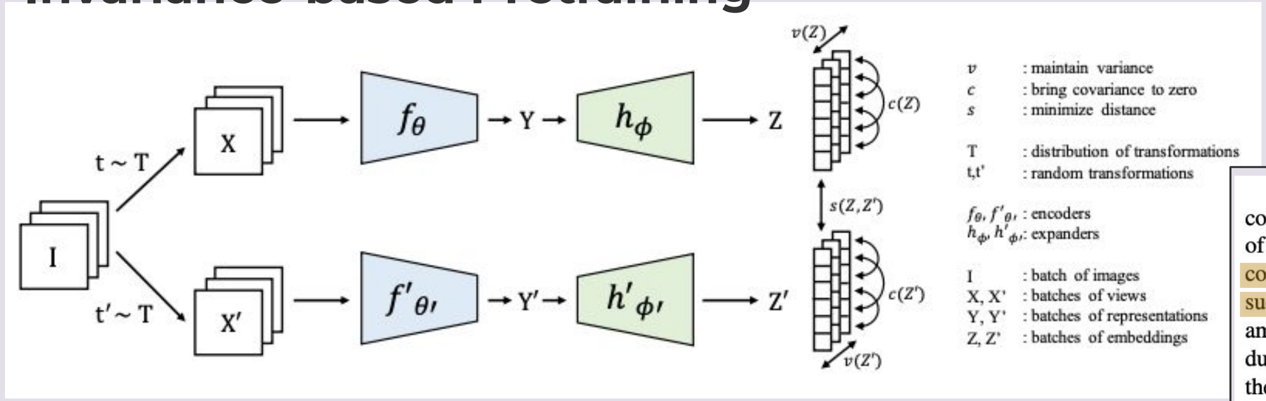


I
t ~ T
t′ ~ T
X
X′
f_θ
$f'_{\theta'}$
Y
Y′
h_ϕ
$h'_{\phi'}$
Z
Z′
$v(Z)$
$c(Z)$
$s(Z,Z')$
$c(Z')$
$v(Z')$
v : maintain variance
c : bring covariance to zero
s : minimize distance
T : distribution of transformations
t,t' : random transformations
$f_\theta, f'_{\theta'}$: encoders
$h_\phi, h'_{\phi'}$: expanders
I : batch of images
X, X' : batches of views
Y, Y' : batches of representations
Z, Z' : batches of embeddings

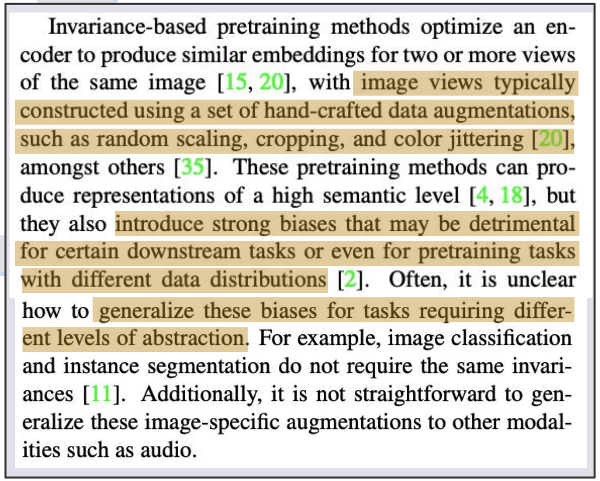
Why select JEPA as pre-training strategy?

Because of perceived issues with other pre-training strategies. Let's take a look at some:

Invariance-based Pretraining



I
t ~ T
t' ~ T
X
X'
f_θ
$f'_{\theta'}$
Y
Y'
h_ϕ
$h'_{\phi'}$
Z
Z'
$v(Z)$
$c(Z)$
$s(Z,Z')$
$c(Z')$
$v(Z')$
v : maintain variance
c : bring covariance to zero
s : minimize distance
T : distribution of transformations
t,t' : random transformations
f_θ, $f'_{\theta'}$: encoders
h_ϕ, $h'_{\phi'}$: expanders
I : batch of images
X, X' : batches of views
Y, Y' : batches of representations
Z, Z' : batches of embeddings

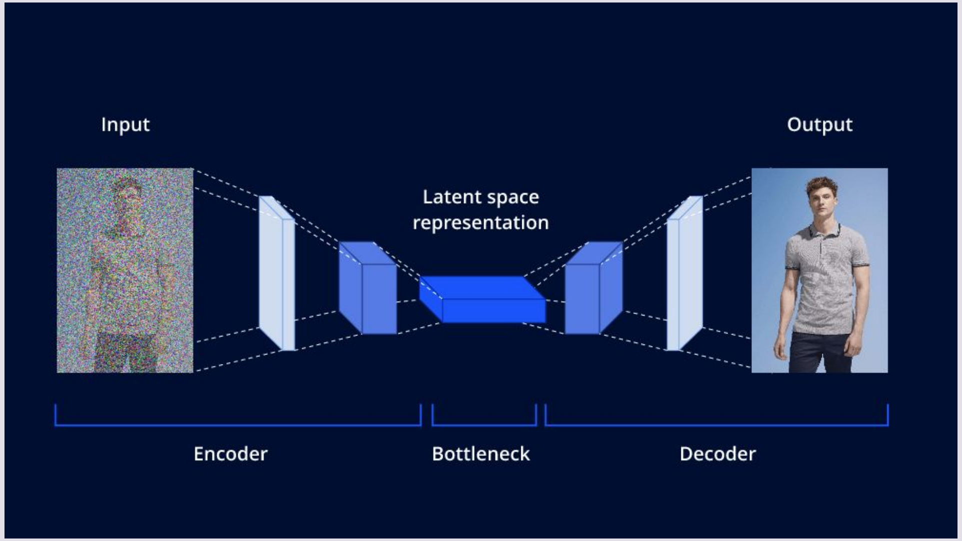


Invariance-based pretraining methods optimize an encoder to produce similar embeddings for two or more views of the same image [15, 20], with image views typically constructed using a set of hand-crafted data augmentations, such as random scaling, cropping, and color jittering [20], amongst others [35]. These pretraining methods can produce representations of a high semantic level [4, 18], but they also introduce strong biases that may be detrimental for certain downstream tasks or even for pretraining tasks with different data distributions [2]. Often, it is unclear how to generalize these biases for tasks requiring different levels of abstraction. For example, image classification and instance segmentation do not require the same invariances [11]. Additionally, it is not straightforward to generalize these image-specific augmentations to other modalities such as audio.

Why select JEPA as pre-training strategy?

Because of perceived issues with other pre-training strategies. Let's take a look at some:

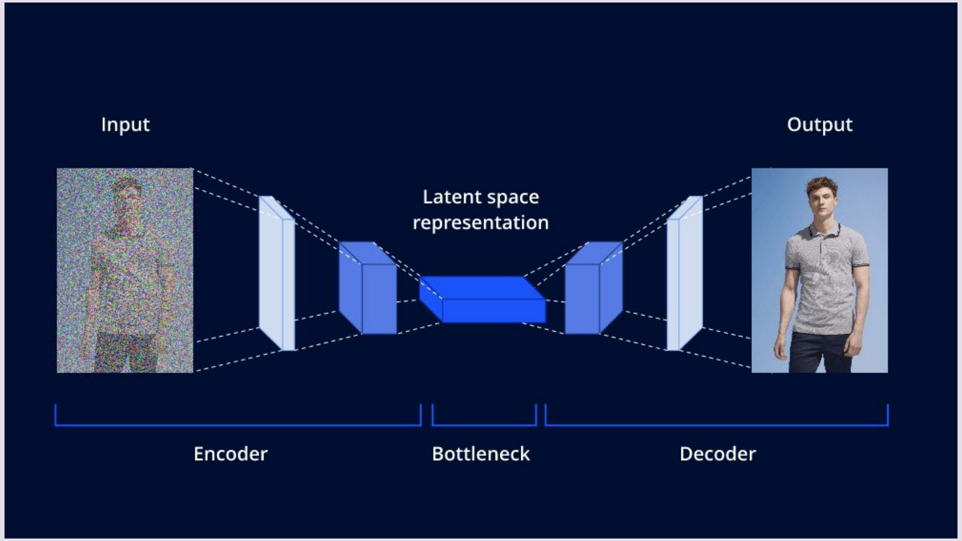
Generative methods that predict in pixel/token space



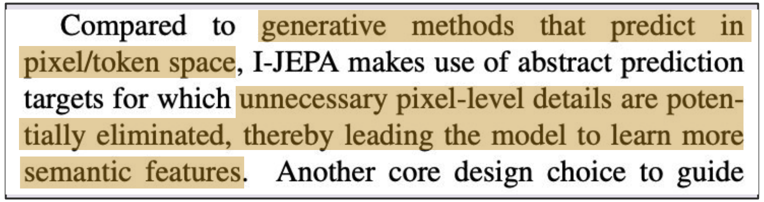
Why select JEPA as pre-training strategy?

Because of perceived issues with other pre-training strategies. Let's take a look at some:

Generative methods that predict in pixel/token space



Loss function here is reconstruction error (in input space)


Compared to generative methods that predict in pixel/token space, I-JEPA makes use of abstract prediction targets for which unnecessary pixel-level details are potentially eliminated, thereby leading the model to learn more semantic features. Another core design choice to guide

NYU

How does InVERGe use JEPA?

How does InVERGe use JEPA?

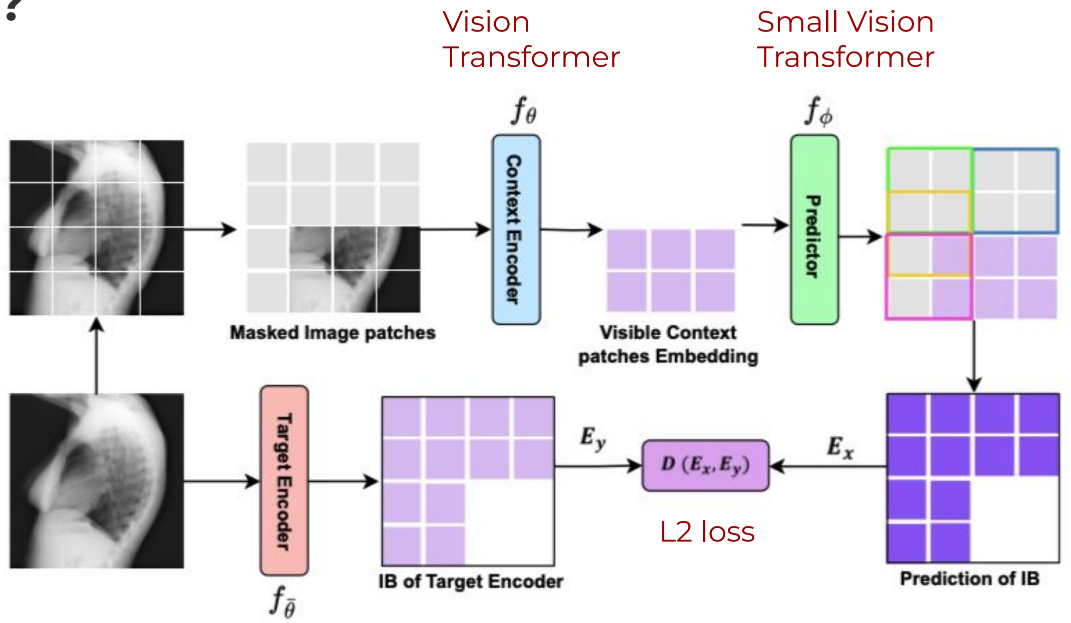
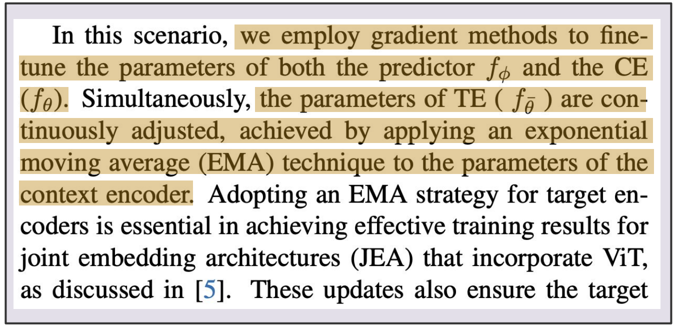


Figure 2. Visual representation of the initial step of training the encoder component of the InVERGe model.

How does InVERGe use JEPA?



In this scenario, we employ gradient methods to fine-tune the parameters of both the predictor f_ϕ and the CE (f_θ). Simultaneously, the parameters of TE ($f_{\bar{\theta}}$) are continuously adjusted, achieved by applying an exponential moving average (EMA) technique to the parameters of the context encoder. Adopting an EMA strategy for target encoders is essential in achieving effective training results for joint embedding architectures (JEA) that incorporate ViT, as discussed in [5]. These updates also ensure the target

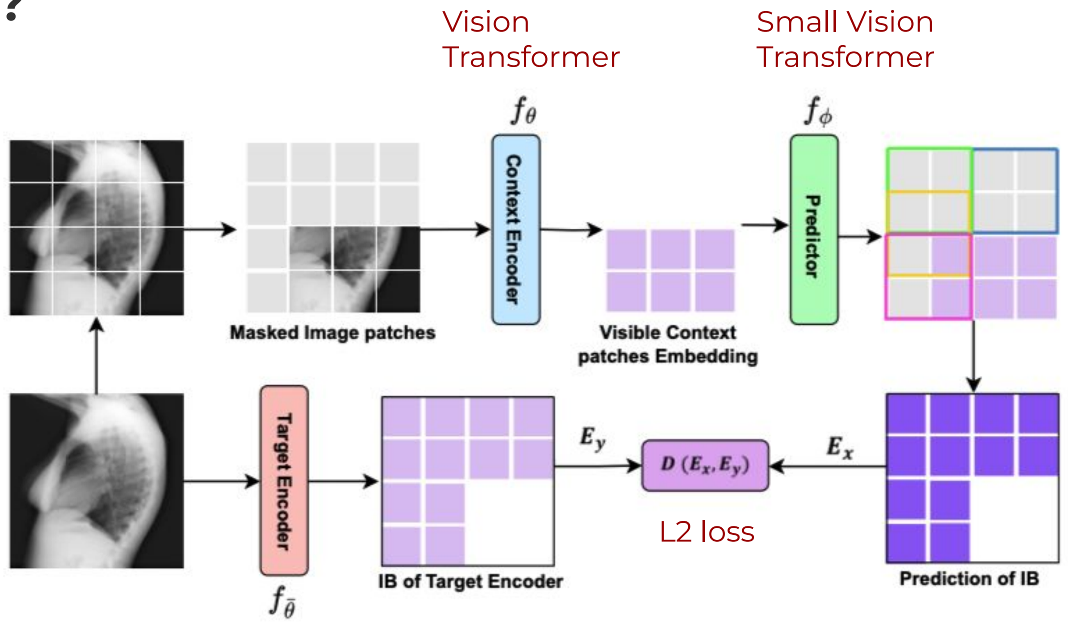
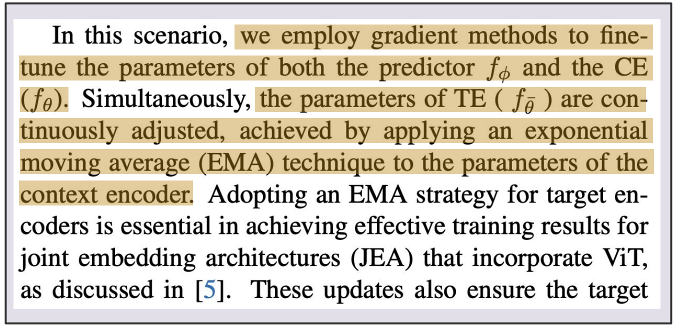


Figure 2. Visual representation of the initial step of training the encoder component of the InVERGe model.

NYU

How does InVERGe use JEPA?


In this scenario, we employ gradient methods to fine-tune the parameters of both the predictor f_ϕ and the CE (f_θ). Simultaneously, the parameters of TE ($f_{\bar{\theta}}$) are continuously adjusted, achieved by applying an exponential moving average (EMA) technique to the parameters of the context encoder. Adopting an EMA strategy for target encoders is essential in achieving effective training results for joint embedding architectures (JEA) that incorporate ViT, as discussed in [5]. These updates also ensure the target

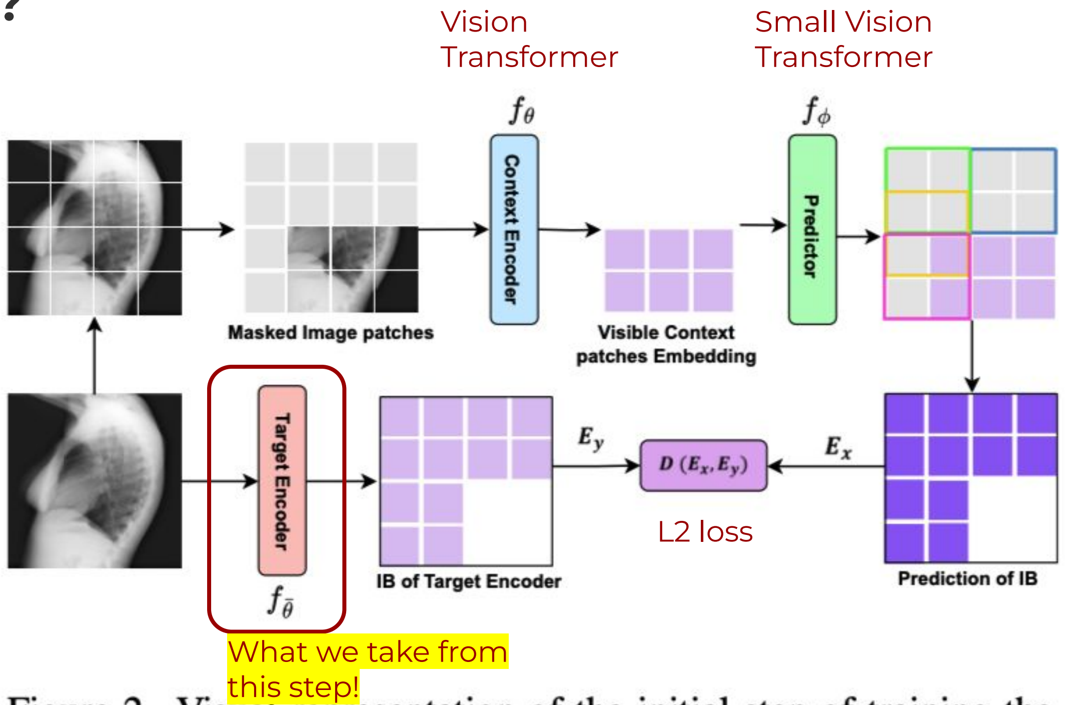
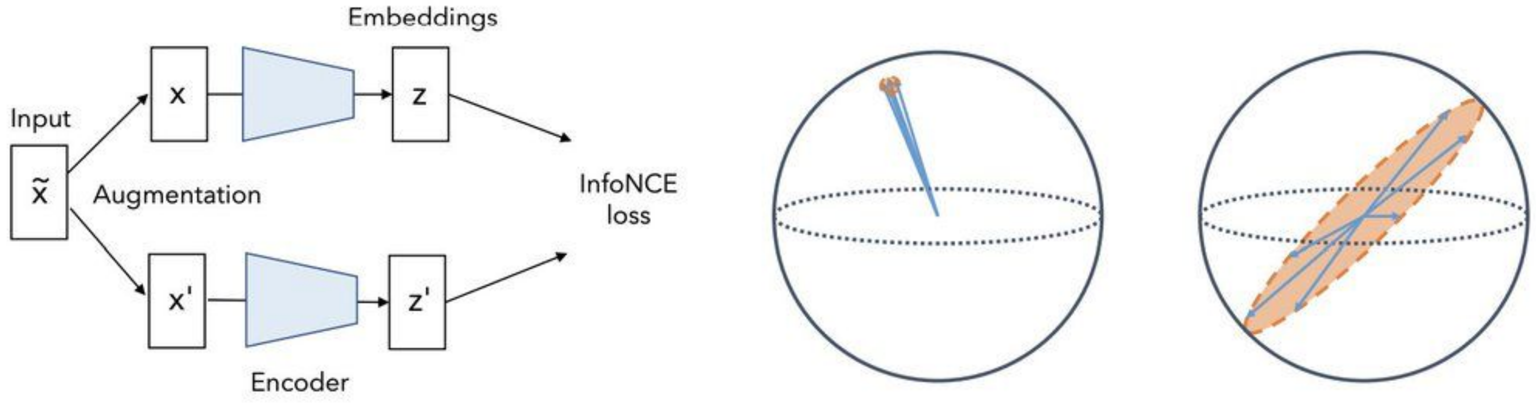


Figure 2. Visual representation of the initial step of training the encoder component of the InVERGe model.

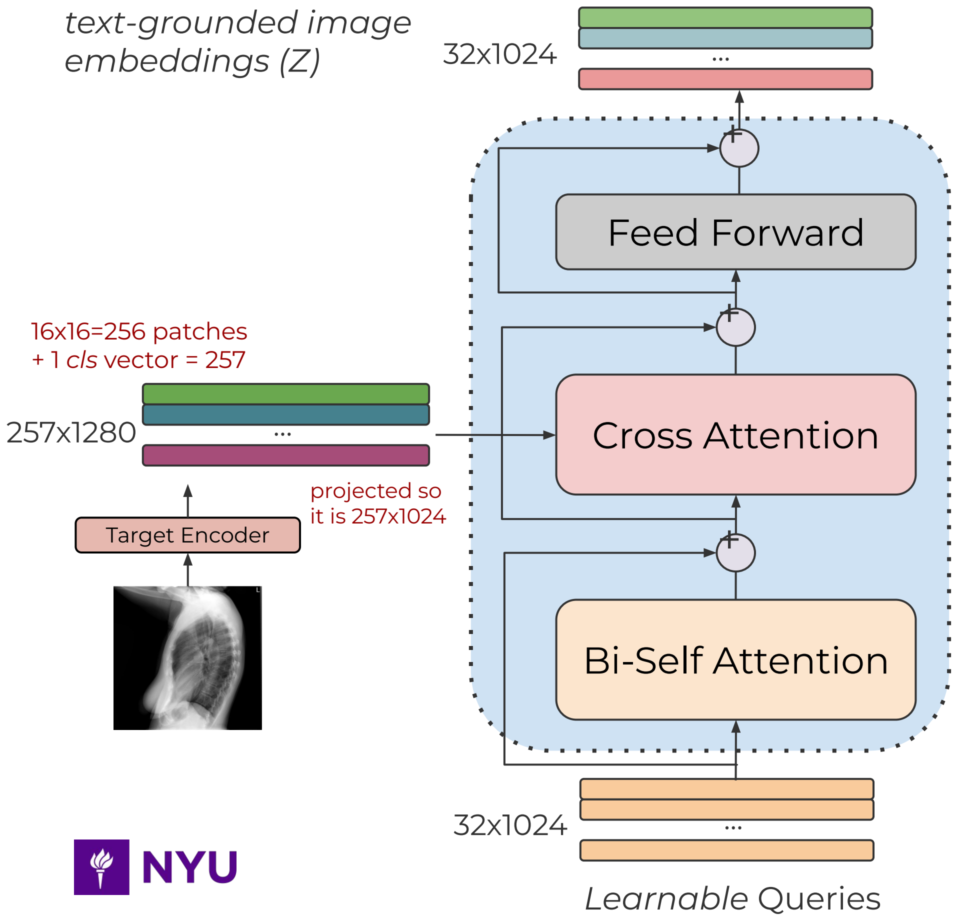


(a) embedding space (b) complete collapse (c) dimensional collapse

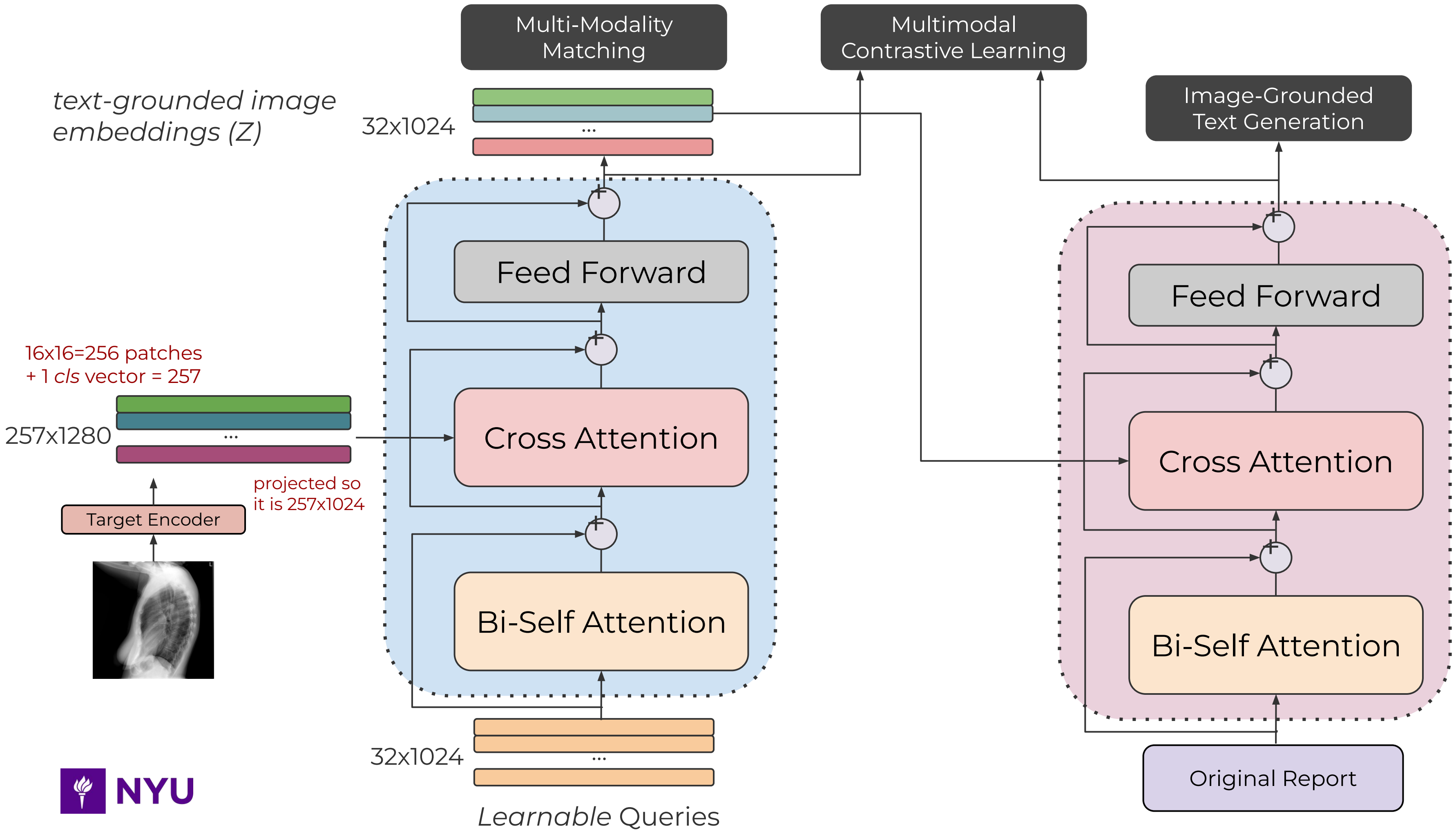
Figure 1: Illustration of the collapsing problem. For complete collapse, the embedding vectors collapse to same point. For dimensional collapse, the embedding vectors only span a lower dimensional space.

Cross-Modal Query Fusion Layer (CMQFL)

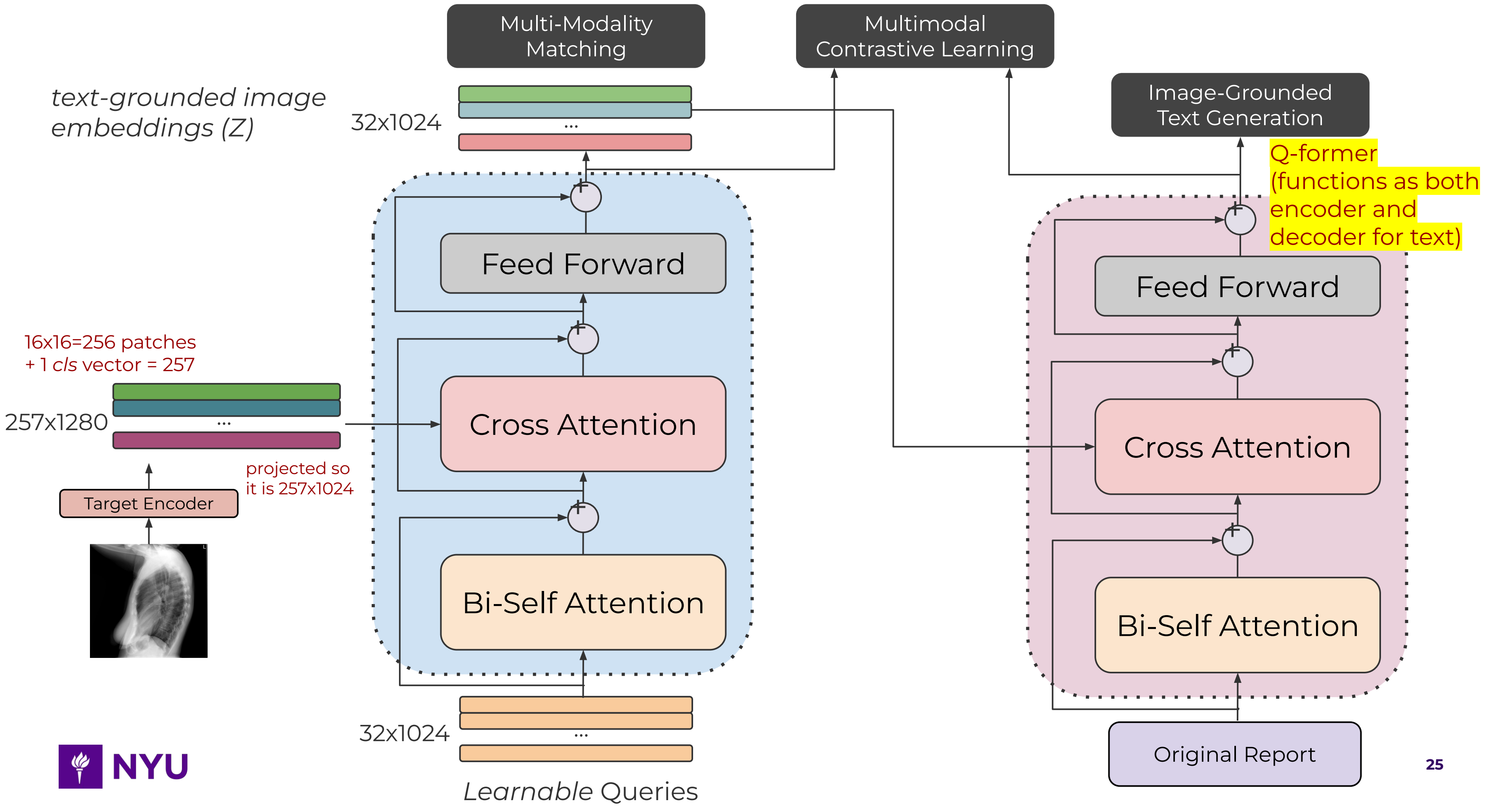
2/3


text-grounded image embeddings (Z)
32x1024
...
Feed Forward
16x16=256 patches
+ 1 *cls* vector = 257
257x1280
...
Cross Attention
projected so
it is 257x1024
Target Encoder
Bi-Self Attention
32x1024
...
Learnable Queries

NYU

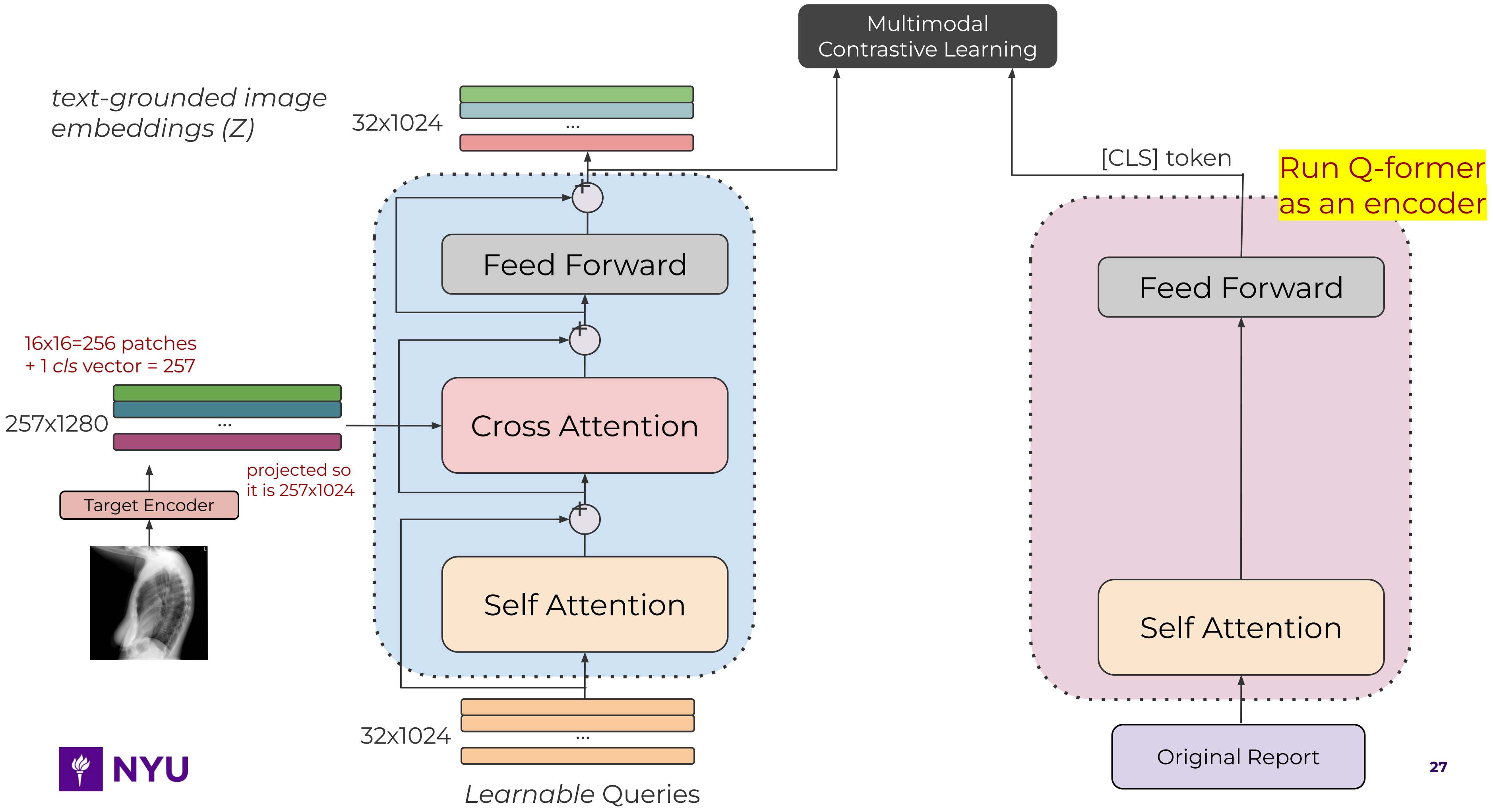

Multi-Modality Matching
Multimodal Contrastive Learning
Image-Grounded Text Generation
text-grounded image embeddings (Z)
32x1024
...
Feed Forward
Cross Attention
Bi-Self Attention
16x16=256 patches
+ 1 *cls* vector = 257
257x1280
...
projected so it is 257x1024
Target Encoder
32x1024
...
Learnable Queries
Feed Forward
Cross Attention
Bi-Self Attention
Original Report

NYU


Multi-Modality Matching
Multimodal Contrastive Learning
Image-Grounded Text Generation
text-grounded image embeddings (Z)
32x1024
...
Q-former (functions as both encoder and decoder for text)
Feed Forward
+
Cross Attention
Bi-Self Attention
16x16=256 patches + 1 cls vector = 257
257x1280
projected so it is 257x1024
Target Encoder
32x1024
Learnable Queries
Original Report

Multimodal Contrastive Learning (MCL)

Loss Term 1/3


Multimodal Contrastive Learning
text-grounded image embeddings (Z)
32x1024
...
[CLS] token
Run Q-former as an encoder
Feed Forward
Cross Attention
Self Attention
16x16=256 patches + 1 *cls* vector = 257
257x1280
...
projected so it is 257x1024
Target Encoder
32x1024
...
Learnable Queries
Feed Forward
Self Attention
Original Report

NYU

Multimodal Contrastive Learning

What are we contrasting?

Image Side: After the CMQFL layer, we have:

$$Z_i = z_i^1, z_i^2, \dots , z_i^{32} \in \mathbb{R}^{(32 \times 1024)}$$

The 32 text-grounded image embeddings for image i

Text Side: you take the [CLS] token from your BERT-based text transformer for report j, call it:

$$t_j = t_{cls}^{(j)} \in \mathbb{R}^{1024}$$

A batch of size B gives you $Z_1, \dots, Z_B$ and $t_1, \dots, t_B$.

Multimodal Contrastive Learning

What are we contrasting?

Image Side: After the CMQFL layer, we have:

$$Z_i = z_i^1, z_i^2, \dots, z_i^{32} \in \mathbb{R}^{(32\times1024)}$$

The 32 text-grounded image embeddings for image i

Text Side: you take the [CLS] token from your BERT-based text transformer for report j, call it:

$$t_j = t_{cls}^{(j)} \in \mathbb{R}^{1024}$$

A batch of size B gives you $Z_1, \dots, Z_B$ and $t_1, \dots, t_B$.

Turning them into comparable vectors:

Before you compare, you run each through small “head” projections and L2-normalize:

$$\tilde{z}_i^k = L_z(z_i^k), ||\tilde{z}_i^k||_2 = 1,$$

$$\tilde{t}_j = L_t(t_j), ||\tilde{t}_j||_2 = 1,$$

Both L_z and L_t are small linear layers mapping $\mathbb{R}^{1024} \rightarrow \mathbb{R}^{1024}$ plus a normalization.

NYU

Multimodal Contrastive Learning

What are we contrasting?

Image Side: After the CMQFL layer, we have:

$$Z_i = z_i^1, z_i^2, \dots, z_i^{32} \in \mathbb{R}^{(32 \times 1024)}$$

The 32 text-grounded image embeddings for image i

Text Side: you take the [CLS] token from your BERT-based text transformer for report j, call it:

$$t_j = t_{cls}^{(j)} \in \mathbb{R}^{1024}$$

A batch of size B gives you $Z_1, \dots, Z_B$ and $t_1, \dots, t_B$.

Turning them into comparable vectors:

Before you compare, you run each through small "head" projections and L2-normalize:

$$\tilde{z}_i^k = L_z(z_i^k), ||\tilde{z}_i^k||_2 = 1,$$

$$\tilde{t}_j = L_t(t_j), ||\tilde{t}_j||_2 = 1,$$

Both L_z and L_t are small linear layers mapping $\mathbb{R}^{1024} \rightarrow \mathbb{R}^{1024}$ plus a normalization.

Defining a single image-text score:

Instead of collapsing all 32 text-grounded image embeddings by averaging, we take the single **maximum** dot-product across all embeddings:

$$sim_{text \rightarrow img}(j, i) = 1/\tau \max_{k=1...32} \langle \tilde{t}_j, \tilde{z}_i^k \rangle$$

τ is a temperature scalar that sharpens (or softens) the distribution.

We also compute the **symmetric** score for text -> image:

$$sim_{img \rightarrow text}(i, j) = 1/\tau \max_{k=1...32} \langle \tilde{z}_i^k, \tilde{t}_j \rangle$$

NYU

Multimodal Contrastive Learning

What are we contrasting?

Image Side: After the CMQFL layer, we have:

$$Z_i = z_i^1, z_i^2, \ldots, z_i^{32} \in \mathbb{R}^{(32\times1024)}$$

The 32 text-grounded image embeddings for image i

Text Side: you take the [CLS] token from your BERT-based text transformer for report j, call it:

$$t_j = t_{cls}^{(j)} \in \mathbb{R}^{1024}$$

A batch of size B gives you $Z_1, \ldots, Z_B$ and $t_1, \ldots, t_B$.

Turning them into comparable vectors:

Before you compare, you run each through small "head" projections and L2-normalize:

$$\tilde{z}_i^k = L_z(z_i^k), ||\tilde{z}_i^k||_2 = 1,$$

$$\tilde{t}_j = L_t(t_j), ||\tilde{t}_j||_2 = 1,$$

Both L_z and L_t are small linear layers mapping $\mathbb{R}^{1024} \rightarrow \mathbb{R}^{1024}$ plus a normalization.

Defining a single image-text score:

Instead of collapsing all 32 text-grounded image embeddings by averaging, we take the single **maximum** dot-product across all embeddings:

$$sim_{text\rightarrow img}(j,i) = 1/\tau \max_{k=1\ldots32} \langle \tilde{t}_j, \tilde{z}_i^k \rangle$$

τ is a temperature scalar that sharpens (or softens) the distribution.

We also compute the **symmetric** score for text -> image:

$$sim_{img\rightarrow text}(i,j) = 1/\tau \max_{k=1\ldots32} \langle \tilde{z}_i^k, \tilde{t}_j \rangle$$

Building the contrastive loss:

We form two $B \times B$ similarity matrices

$$S_{i,j}^{(i\rightarrow t)} = sim_{img\rightarrow text}(i,j) \qquad S_{j,i}^{(t\rightarrow i)} = sim_{text\rightarrow img}(j,i)$$

As we know image i should match text i in the batch, we create a target vector $Y = [0, 1, 2, \ldots, B-1]$ meaning "the correct text for image i is at position i and vice versa.
Thus,

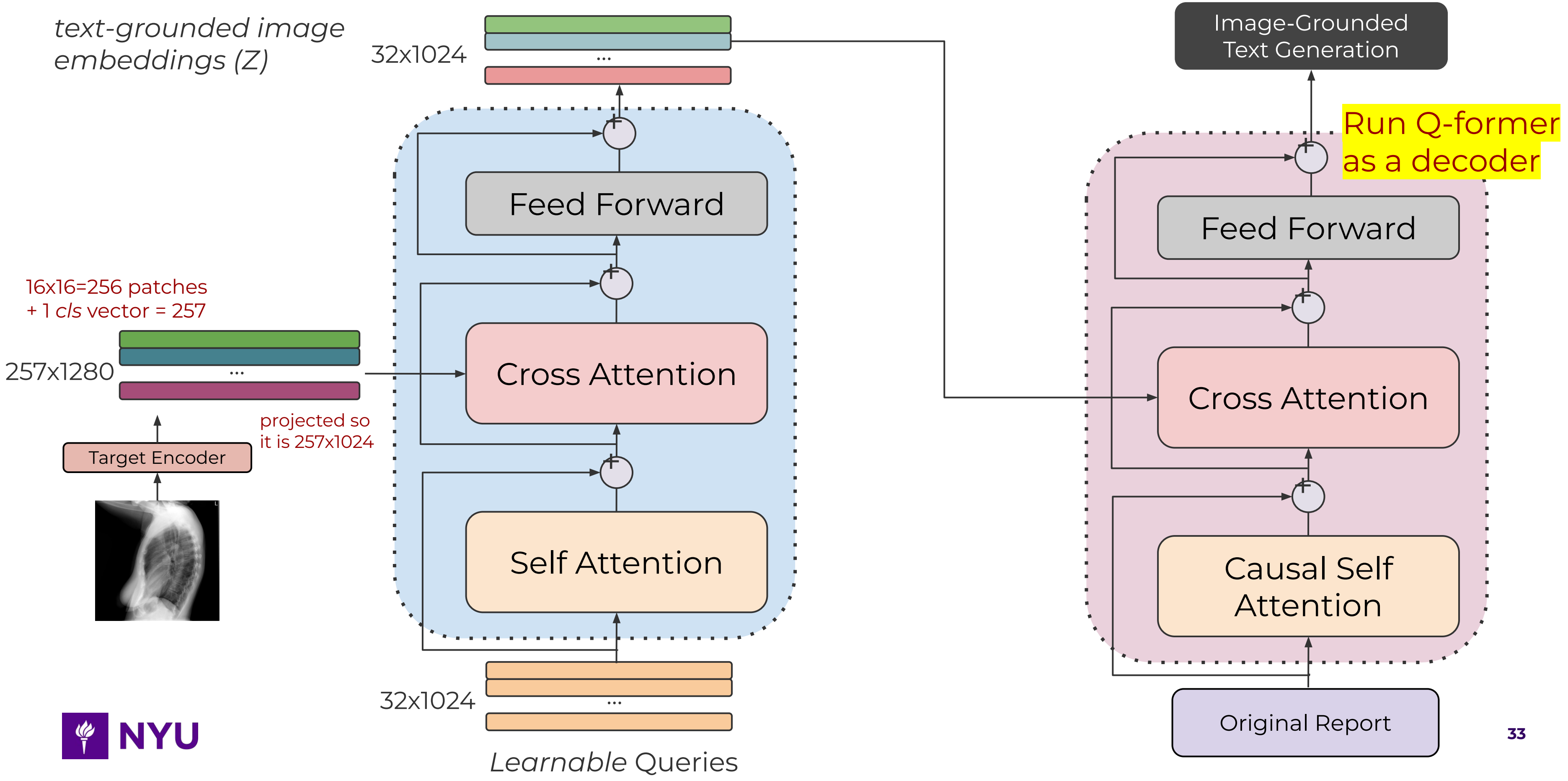
$$L_{mcl} = 1/2[CE(Y, S^{(i\rightarrow t)}) + CE(Y, S^{(t\rightarrow i)})]$$

We treat each row of S as a distribution over possible matches and push the correct one to 1.

NYU

Masked Language Modelling (MLM)

Loss Term 2/3


text-grounded image embeddings (Z)
32x1024
...
Image-Grounded Text Generation
Run Q-former as a decoder
Feed Forward
Cross Attention
Self Attention
16x16=256 patches + 1 *cls* vector = 257
257x1280
...
projected so it is 257x1024
Target Encoder
32x1024
...
Learnable Queries
Feed Forward
Cross Attention
Causal Self Attention
Original Report

NYU

Masked Language Modelling (MLM):

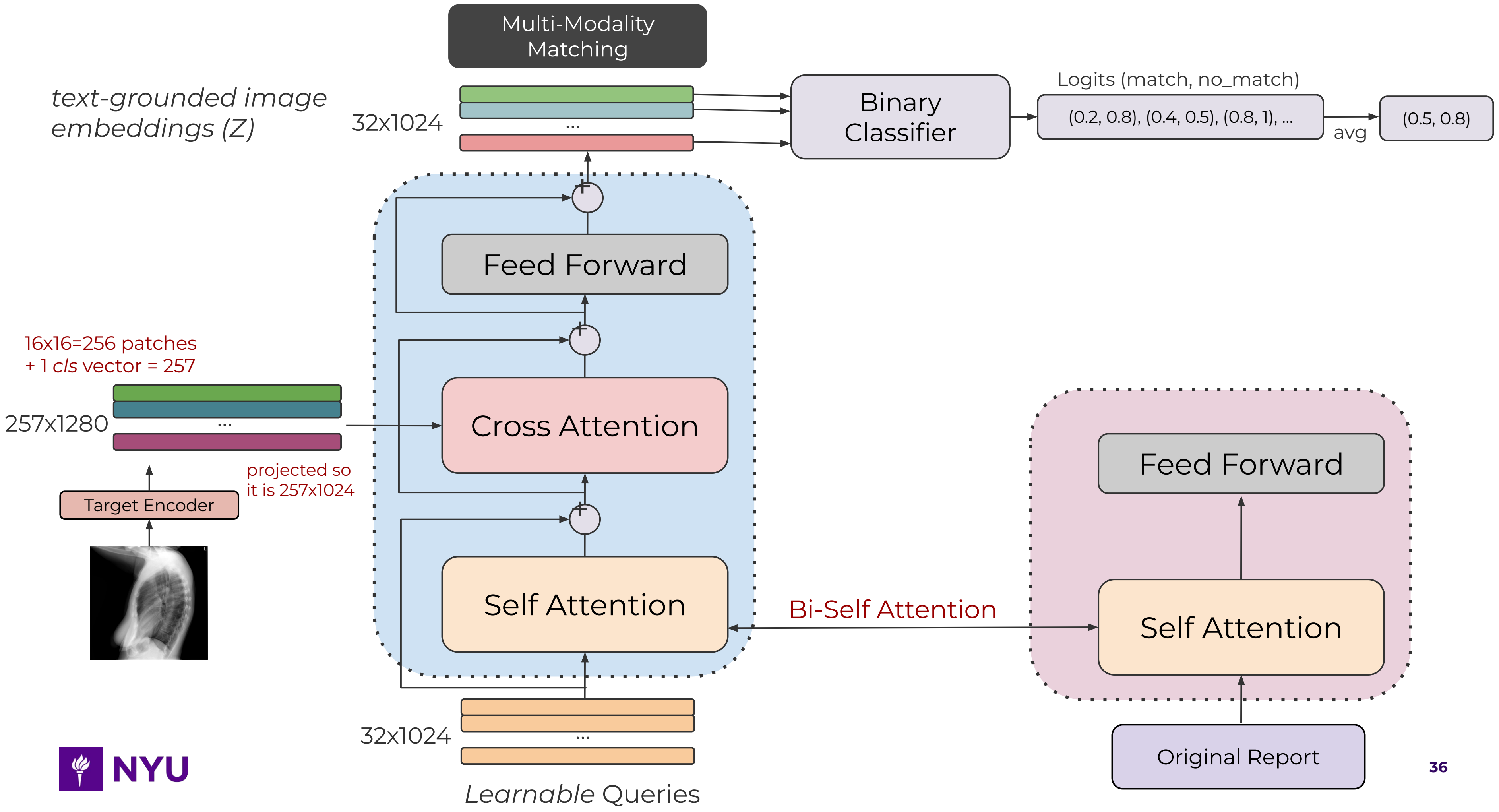
1. Switch Q-former to **decoder**.
2. It uses a **causal self-attention** for text input (so text can't peek forward in the Q-former's self attention block)
3. **Cross-attention** block attends these *text representations* with KV caches from the *text-grounded image embeddings*.

The loss is then simply the **standard next-token (teacher forcing) cross-entropy** over every real token:

$$L_{MLM} = -1/N \sum_{b=1}^{B} \sum_{t=1}^{T-1} \mathbf{1}\left[y_{b,t+1} \neq -100\right] logp(y_{b,t+1} | \text{context up to t})$$

Multi-Modality Matching (MMM)

Loss Term 3/3



Multi-Modality Matching
text-grounded image embeddings (Z)
32x1024
Binary Classifier
Logits (match, no_match)
(0.2, 0.8), (0.4, 0.5), (0.8, 1), ...
avg
(0.5, 0.8)
Feed Forward
Cross Attention
Self Attention
16x16=256 patches + 1 cls vector = 257
257x1280
projected so it is 257x1024
Target Encoder
Bi-Self Attention
Feed Forward
Self Attention
Original Report
32x1024
Learnable Queries

NYU

Multi-Modality Matching (MMM):

The goal here is to teach the model to **distinguish** matched image-text pairs from hard negatives.

Multi-Modality Matching (MMM):

The goal here is to teach the model to **distinguish** matched image-text pairs from hard negatives.

Batch Construction

For each of the B real (I_b, T_b) pairs:

1. Positive (I_b, T_b)
2. Hard-negative image (I_b^-, T_b) sampled via highest text -> image contrastive weights.
3. Hard-negative text (I_b, T_b^-) sampled via highest image -> text contastive weights

Stack into a mega-batch of size $3B$.

Multi-Modality Matching (MMM):

The goal here is to teach the model to **distinguish** matched image-text pairs from hard negatives.

Batch Construction

For each of the B real (I_b, T_b) pairs:

1. Positive (I_b, T_b)
2. Hard-negative image (I_b^-, T_b) sampled via highest text -> image contrastive weights.
3. Hard-negative text (I_b, T_b^-) sampled via highest image -> text contastive weights

Stack into a mega-batch of size $3B$.

Encoder and Bidirectional Self-Attention

We prepend the **32 learnable query tokens** so that the **bi-directional self-attention acts over [queries | text tokens]**

- *(no causal masking, instead full BERT-style attention among tokens)*

The **cross-attention mechanism** ensures the query slots attend into the image patch embeddings.

Multi-Modality Matching (MMM):

The goal here is to teach the model to **distinguish** matched image-text pairs from hard negatives.

Batch Construction

For each of the B real (I_b, T_b) pairs:

1. Positive (I_b, T_b)
2. Hard-negative image (I_b^-, T_b) sampled via highest text -> image contrastive weights.
3. Hard-negative text (I_b, T_b^-) sampled via highest image -> text contastive weights

Stack into a mega-batch of size $3B$.

Encoder and Bidirectional Self-Attention

We prepend the **32 learnable query tokens** so that the **bi-directional self-attention acts over [queries | text tokens]**

- *(no causal masking, instead full BERT-style attention among tokens)*

The **cross-attention mechanism** ensures the query slots attend into the image patch embeddings.

Classification Head

1. Take the final hidden states of the 32 queries -> shape (3B, 32, D)
2. Apply a small **linear layer** to each query -> 2-way logits ("match" vs. "non-match") -> (3B, 32, 2)
3. Mean-pool across the 32 queries -> final logits (3B, 2)

NYU

Multi-Modality Matching (MMM):

The goal here is to teach the model to **distinguish** matched image-text pairs from hard negatives.

Batch Construction

For each of the B real (I_b, T_b) pairs:

1. Positive (I_b, T_b)
2. Hard-negative image (I_b^-, T_b) sampled via highest text -> image contrastive weights.
3. Hard-negative text (I_b, T_b^-) sampled via highest image -> text contastive weights

Stack into a mega-batch of size $3B$.

Encoder and Bidirectional Self-Attention

We prepend the **32 learnable query tokens** so that the **bi-directional self-attention acts over [queries | text tokens]**

- *(no causal masking, instead full BERT-style attention among tokens)*

The **cross-attention mechanism** ensures the query slots attend into the image patch embeddings.

Classification Head

1. Take the final hidden states of the 32 queries -> shape (3B, 32, D)
2. Apply a small **linear layer** to each query -> 2-way logits ("match" vs. "non-match") -> (3B, 32, 2)
3. Mean-pool across the 32 queries -> final logits (3B, 2)

Loss Function

Cross-entropy over the 3B logits with labels $y_i \in \{0, 1\}$

$$L_{MMM} = -1/(3B) \sum_{i=1}^{3B} [y_i log p_{i,1} + (1 - y_i) log p_{i,0}]$$

Positives get $y_i = 1$;
Both negatives get $y_i = 0$.

$$L_{pretraining} = L_{mcl} + L_{mlm} + L_{mmm}$$

(CMQFL Pretraining Objective)

Decoder

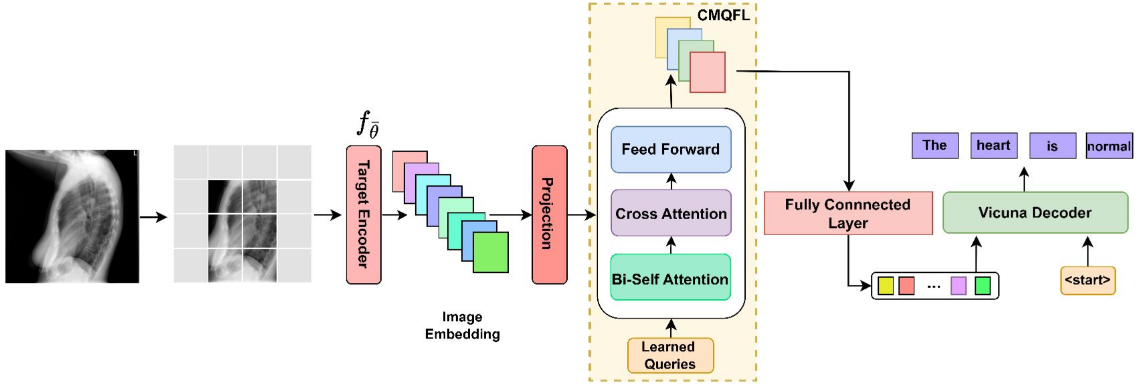
3/3

Decoder

In the second stage (fine-tuning the VICUNA decoder):

1. **Text-grounded image embeddings** are provided as an *explicit prefix* of embedding vectors passed into VICUNA decoder
   a. *acting as soft visual cues that condition the decoder on the visual context received by the CMQFL layer*
2. Use frozen CMQFL layer (encoder) and **exclusively employ the MLM loss**.

“This not only eases the LLM’s burden in learning the alignment between sight and language but also reduces the problem of catastrophic forgetting.”



Implementation Details

Datasets

For Training the Image Encoder:

1. **NIH**

For Generating the Reports:

1. **MIMIC-CXR**
    a. Images + corresponding reports
    b. We specifically focus on the ‘p10’ folder, which contains 36,337 images.

2. **IU-Xray**
    a. 7,470 chest X-ray images, each paired with a corresponding radiology report, totalling 3,955 reports

3. **CDD-CESM**
    a. 1003 low-energy images, featuring CC (Craniocaudal) and MLO (Mediolateral Oblique) views for both breasts. This dataset is derived from 326 female patients.

Metrics

1. BLEU (Precision-based)
2. METEOR (Precision, Recall, Stemming, Synonyms, and Alignment)
3. ROUGE-L (Recall-based)

Reference: “The cat sat on the mat”
Our Guess: “The cat sat on a mat”

Unigrams:
Reference: {The, cat, sat, on, the, mat}
Candidate: {The, cat, sat, on, a, mat}

BLEU-1 = ⅚ ~ 0.83

Reference: “The cat sat on the mat”
Our Guess: “A feline was sitting on a rug”

Aligns:
“cat” <-> “feline” (synonym)
“sat” <-> “sitting” (stem)
“mat” <-> “rug” (synonym)

METEOR ~ 0.5 or higher

Reference: “The cat sat on the mat”
Our Guess: “The cat is sitting on the mat”

LCS: “The cat on the mat”
|LCS| = 5 words
Reference length = 6 words

ROUGE-L = ⅚ ~ 0.83

BLEU and METEOR were initially developed for *assessing machine translation quality*, while ROUGE-L is specifically designed for *evaluating the quality of textual summaries*.

Results

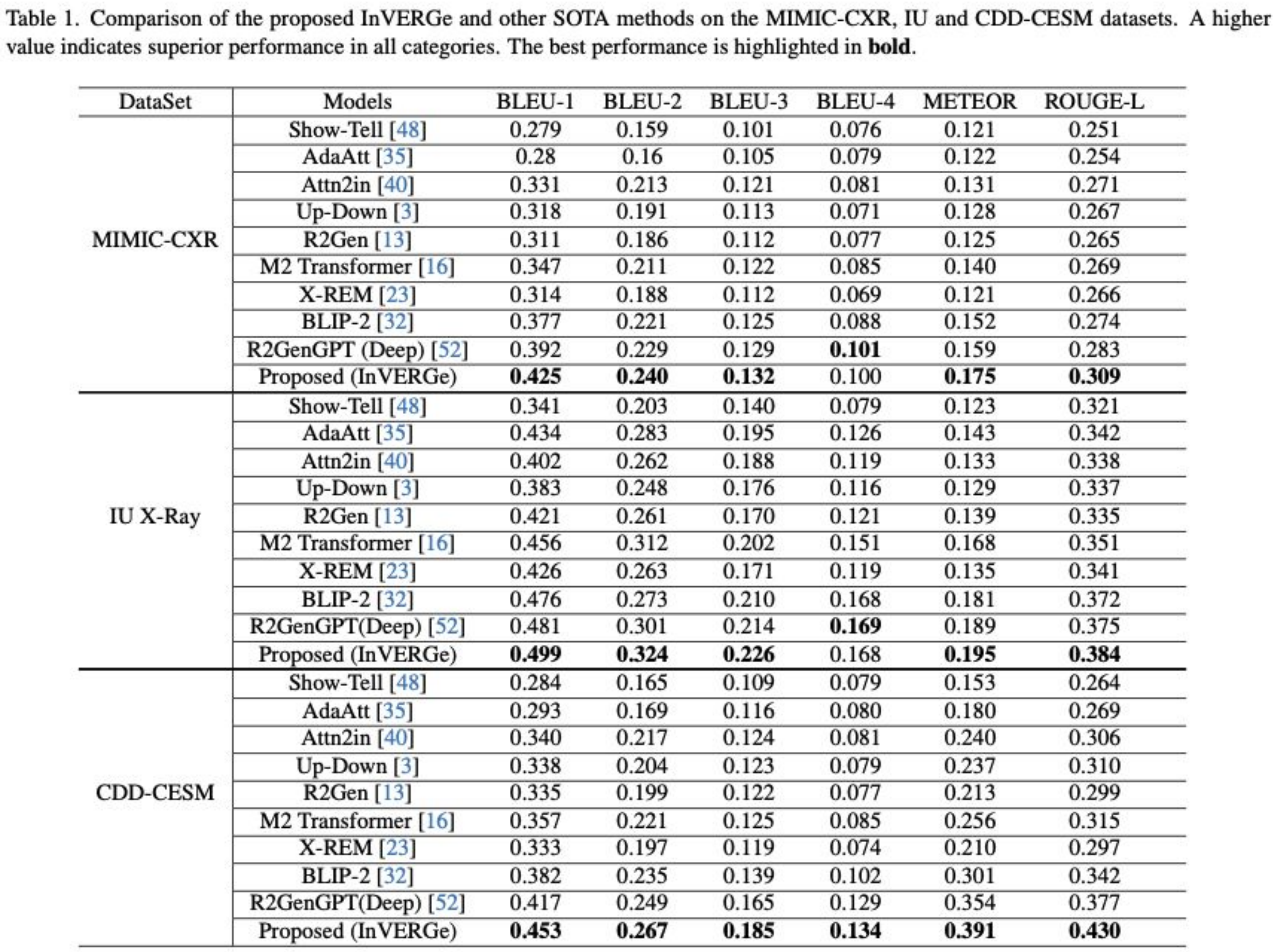


Table 1. Comparison of the proposed InVERGe and other SOTA methods on the MIMIC-CXR, IU and CDD-CESM datasets. A higher value indicates superior performance in all categories. The best performance is highlighted in **bold**.

DataSet	Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
MIMIC-CXR	Show-Tell [48]	0.279	0.159	0.101	0.076	0.121	0.251
	AdaAtt [35]	0.28	0.16	0.105	0.079	0.122	0.254
	Attn2in [40]	0.331	0.213	0.121	0.081	0.131	0.271
	Up-Down [3]	0.318	0.191	0.113	0.071	0.128	0.267
	R2Gen [13]	0.311	0.186	0.112	0.077	0.125	0.265
	M2 Transformer [16]	0.347	0.211	0.122	0.085	0.140	0.269
	X-REM [23]	0.314	0.188	0.112	0.069	0.121	0.266
	BLIP-2 [32]	0.377	0.221	0.125	0.088	0.152	0.274
	R2GenGPT (Deep) [52]	0.392	0.229	0.129	**0.101**	0.159	0.283
	Proposed (InVERGe)	**0.425**	**0.240**	**0.132**	0.100	**0.175**	**0.309**
IU X-Ray	Show-Tell [48]	0.341	0.203	0.140	0.079	0.123	0.321
	AdaAtt [35]	0.434	0.283	0.195	0.126	0.143	0.342
	Attn2in [40]	0.402	0.262	0.188	0.119	0.133	0.338
	Up-Down [3]	0.383	0.248	0.176	0.116	0.129	0.337
	R2Gen [13]	0.421	0.261	0.170	0.121	0.139	0.335
	M2 Transformer [16]	0.456	0.312	0.202	0.151	0.168	0.351
	X-REM [23]	0.426	0.263	0.171	0.119	0.135	0.341
	BLIP-2 [32]	0.476	0.273	0.210	0.168	0.181	0.372
	R2GenGPT(Deep) [52]	0.481	0.301	0.214	**0.169**	0.189	0.375
	Proposed (InVERGe)	**0.499**	**0.324**	**0.226**	0.168	**0.195**	**0.384**
CDD-CESM	Show-Tell [48]	0.284	0.165	0.109	0.079	0.153	0.264
	AdaAtt [35]	0.293	0.169	0.116	0.080	0.180	0.269
	Attn2in [40]	0.340	0.217	0.124	0.081	0.240	0.306
	Up-Down [3]	0.338	0.204	0.123	0.079	0.237	0.310
	R2Gen [13]	0.335	0.199	0.122	0.077	0.213	0.299
	M2 Transformer [16]	0.357	0.221	0.125	0.085	0.256	0.315
	X-REM [23]	0.333	0.197	0.119	0.074	0.210	0.297
	BLIP-2 [32]	0.382	0.235	0.139	0.102	0.301	0.342
	R2GenGPT(Deep) [52]	0.417	0.249	0.165	0.129	0.354	0.377
	Proposed (InVERGe)	**0.453**	**0.267**	**0.185**	**0.134**	**0.391**	**0.430**

NYU

Table 2. Baseline refers to one normal ViT encoder and decoder only. CPE (Context Pixel encoder) stands for our trained Model's Target Encoder. For this, we use the MIMIC-CXR dataset. Here we only use MLM objective function to check the performance.

Model	BLEU-2	METEOR	Rouge-L
Baseline	0.161	0.124	0.255
MAE + Decoder	0.178	0.060	0.208
CPE + Decoder	0.183	0.117	0.260
CPE + CMQFL + Decoder	**0.227**	**0.163**	**0.290**

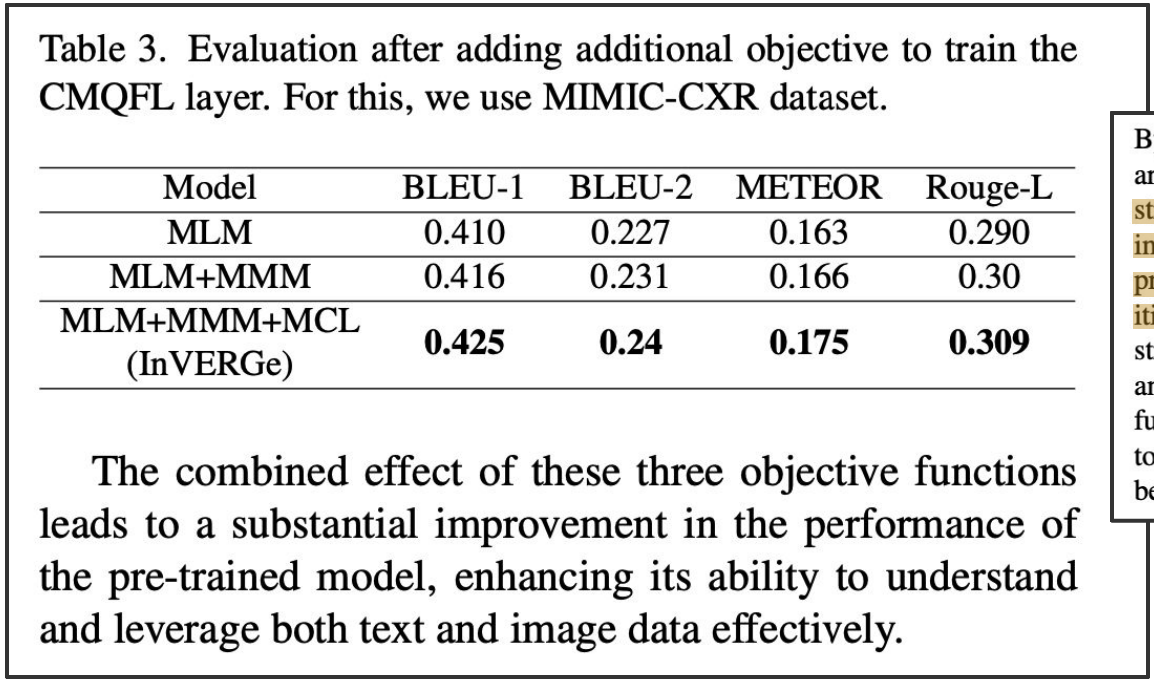
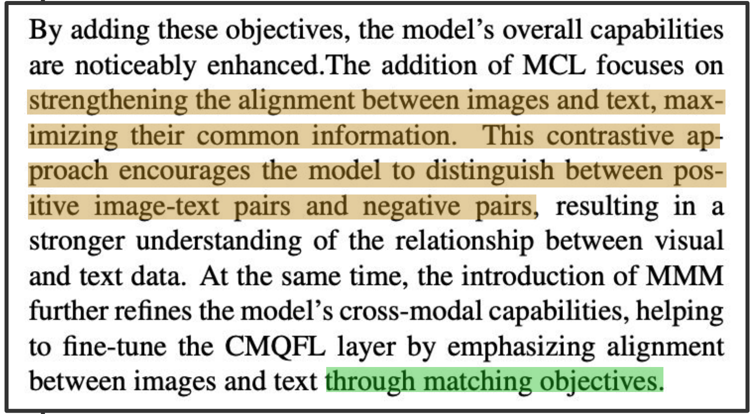


Table 3. Evaluation after adding additional objective to train the CMQFL layer. For this, we use MIMIC-CXR dataset.

Model	BLEU-1	BLEU-2	METEOR	Rouge-L
MLM	0.410	0.227	0.163	0.290
MLM+MMM	0.416	0.231	0.166	0.30
MLM+MMM+MCL (InVERGe)	**0.425**	**0.24**	**0.175**	**0.309**

The combined effect of these three objective functions leads to a substantial improvement in the performance of the pre-trained model, enhancing its ability to understand and leverage both text and image data effectively.

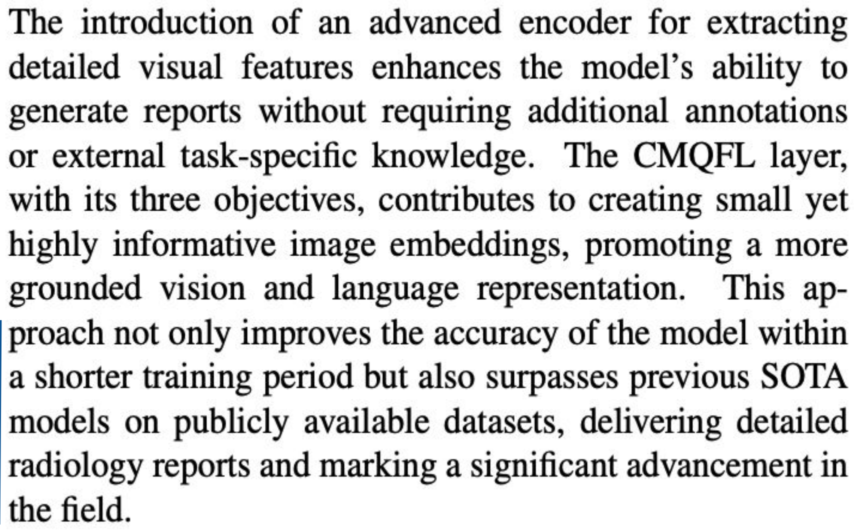


By adding these objectives, the model's overall capabilities are noticeably enhanced.The addition of MCL focuses on strengthening the alignment between images and text, maximizing their common information. This contrastive approach encourages the model to distinguish between positive image-text pairs and negative pairs, resulting in a stronger understanding of the relationship between visual and text data. At the same time, the introduction of MMM further refines the model's cross-modal capabilities, helping to fine-tune the CMQFL layer by emphasizing alignment between images and text through matching objectives.

NYU

JEPA-based
image encoder

~~Problem~~
[DONE] Models have been observed to produce **unnatural language outputs** when trained solely on raw image-text pairs
[DONE] **High computational demand** of end-to-end model training


The introduction of an advanced encoder for extracting detailed visual features enhances the model's ability to generate reports without requiring additional annotations or external task-specific knowledge. The CMQFL layer, with its three objectives, contributes to creating small yet highly informative image embeddings, promoting a more grounded vision and language representation. This approach not only improves the accuracy of the model within a shorter training period but also surpasses previous SOTA models on publicly available datasets, delivering detailed radiology reports and marking a significant advancement in the field.

Bridging
modalities

NYU

Paper Reference

A. Deria, K. Kumar, S. Chakraborty, D. Mahapatra and S. Roy, "InVERGe: Intelligent Visual Encoder for Bridging Modalities in Report Generation," 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2024, pp. 2028-2038, doi: 10.1109/CVPRW63382.2024.00208. keywords: {Training;Visualization;Computational modeling;Radiology;Transformers;Feature extraction;Decoding;Computer Vision;Deep Learning;Medical Imaging;Report Generation;Large Language Model},