SmartEdit: Exploring Complex Instruction-based Image Editing with Multimodal Large Language Models

Yuzhou Huang[*1,2#] Liangbin Xie[*2,3,5#] Xintao Wang[2,4†] Ziyang Yuan[2,8#] Xiaodong Cun[4]
Yixiao Ge[2,4] Jiantao Zhou[3] Chao Dong[5,7] Rui Huang[6] Ruimao Zhang[1†] Ying Shan[2,4]

[1]School of Data Science, Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, China
[2]**ARC Lab, Tencent PCG** [3]University of Macau [4]Tencent AI Lab
[5]Shenzhen Institute of Advanced Technology [6]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China
[7]Shanghai Artificial Intelligence Laboratory [8]Tsinghua University

CVPR 2024

Presenter: Hao Wang

Advisor: Prof. Chia-Wen Lin

Outline

- Introduction
- Related Work
- Framework
- Method
- Experiment
- Conclusion

Outline

- Introduction
- Related Work
- Framework
- Method
- Experiment
- Conclusion

Introduction

- Analyze and focus on the performance of **instruction-based image editing** methods in more complex instructions, which are less explored in past research.

- Leverage **MLLMs** to better comprehend instructions. To further improve the performance, author propose a **B**idirectional **I**nteraction **M**odule.

- Propose a new **dataset utilization strategy** to enhance the performance of SmartEdit in complex scenarios.

Introduction

"Change the left/right animal to a white fox"

"Change the bigger/smaller bear to a wolf"

"Change the left/middle/right apple to an orange"

"Change the red/green apple to a peach"

"Change the dog in mirror to a tiger"

"Please replace the animal that is usually known as friend of human's with a tiger"

"Please remove the object that can tell the time"

Introduction

Figure 2. For more complex instructions or scenarios, InstructPix2Pix fails to follow the instructions.

Outline

- Introduction
- Related Work
- Framework
- Method
- Experiment
- Conclusion

Related Work

- LoRA: Low-Rank Adaptation of Large Language Models
  - ICLR 2022

- Generating Images with Multimodal Language Models
  - NeurIPS 2023

Related Work – LoRA

LoRA: Low-Rank Adaptation of Large Language Models

Edward Hu* **Yelong Shen*** **Phillip Wallis** **Zeyuan Allen-Zhu**
Yuanzhi Li **Shean Wang** **Lu Wang** **Weizhu Chen**
Microsoft Corporation
{edwardhu, yeshe, phwallis, zeyuana,
yuanzhil, swang, luw, wzchen}@microsoft.com
yuanzhil@andrew.cmu.edu

ICLR 2022

Problem statement

$$\max_{\Phi} \sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log\left(P_{\Phi}(y_t|x, y_{<t})\right) \tag{1}$$

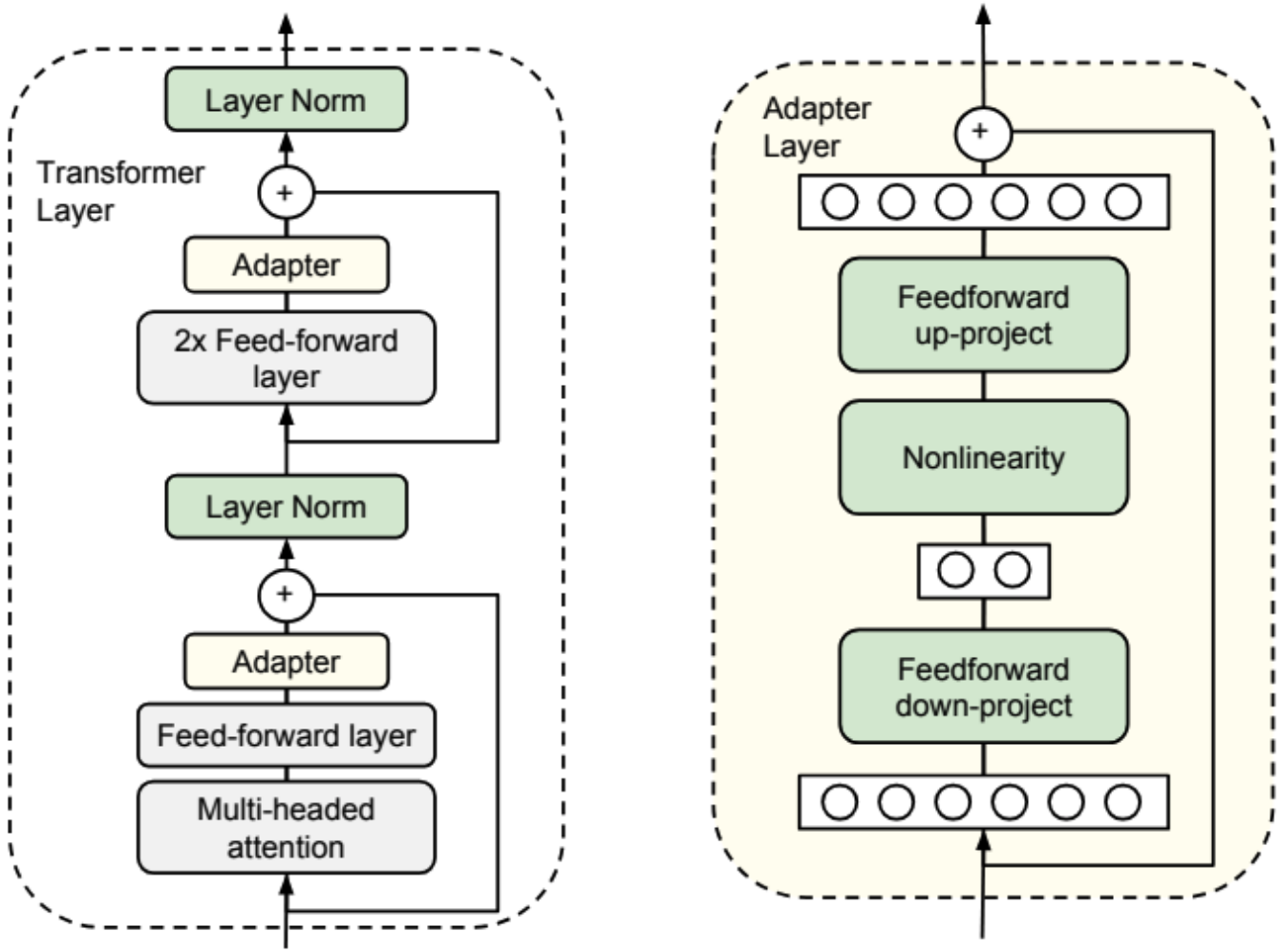
$$\max_{\Theta} \sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log\left(p_{\Phi_0+\Delta\Phi(\Theta)}(y_t|x, y_{<t})\right) \tag{2}$$

- Language modeling problem
- Maximization of conditional probabilities given a task-specific prompt

Existing solution (1)

- Adding adapter layers

$$|\Theta| = \hat{L}_{Adpt} \times (2 \times d_{model} \times r + r + d_{model}) + 2 \times \hat{L}_{LN} \times d_{model}$$



Parameter-Efficient Transfer Learning for NLP, ICML 2019

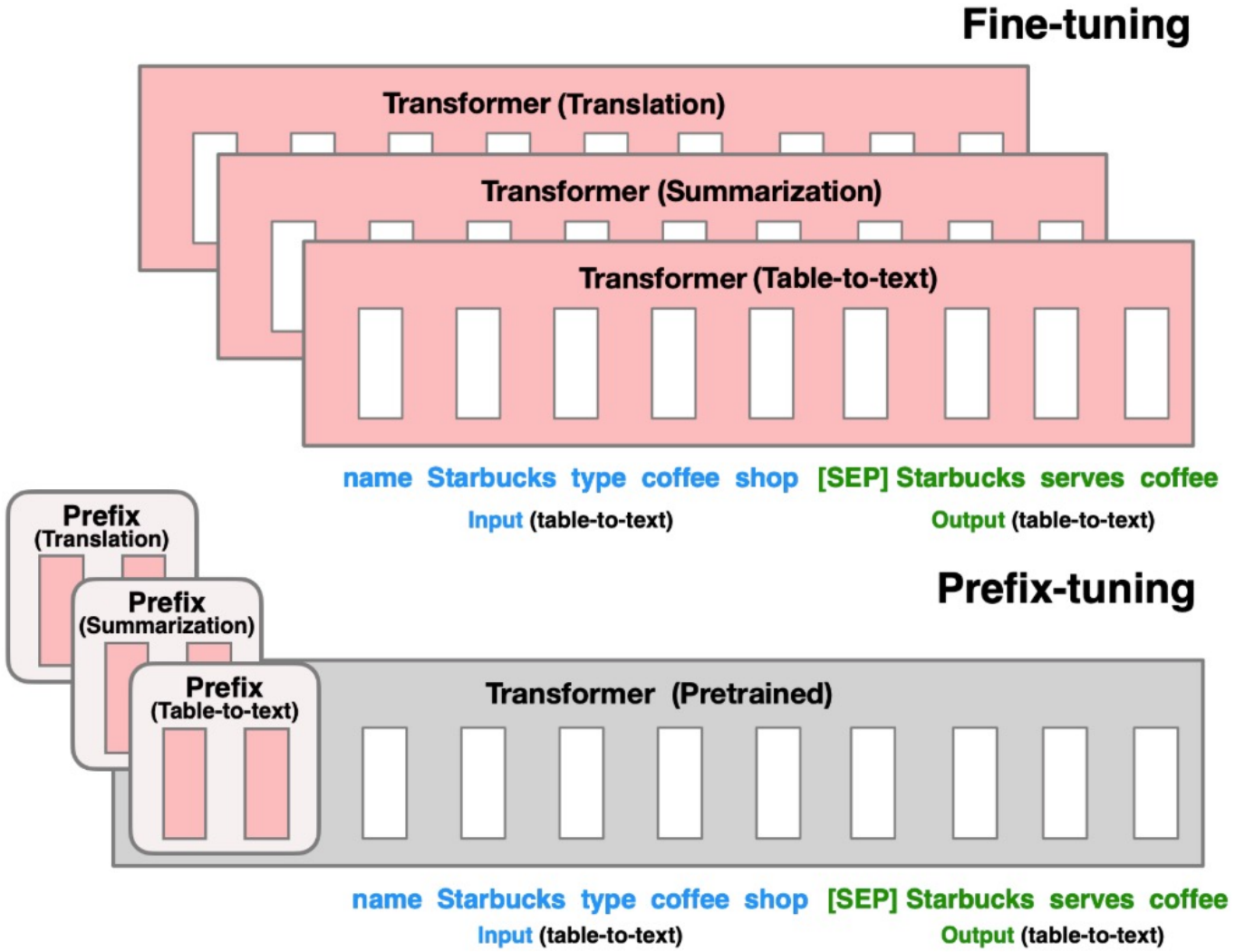
Existing solution (2)

- Prefix tuning
  - Prefix-embedding tuning (PreEmbed)

$$|\Theta| = d_{model} \times (l_p + l_i)$$

  - Prefix-layer tuning (PreLayer)

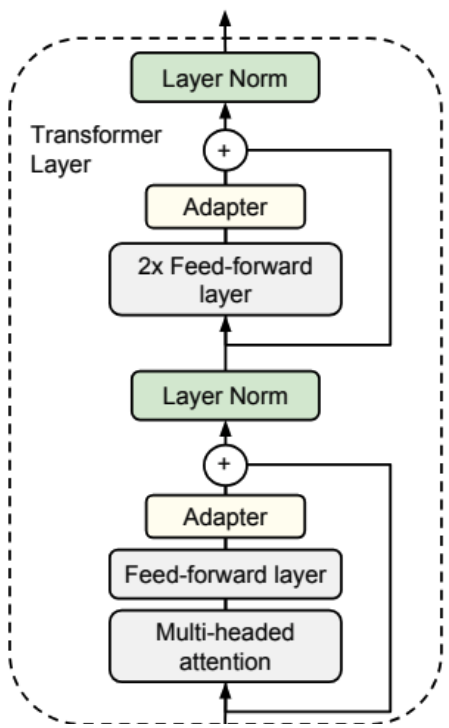
$$|\Theta| = L \times d_{model} \times (l_p + l_i)$$

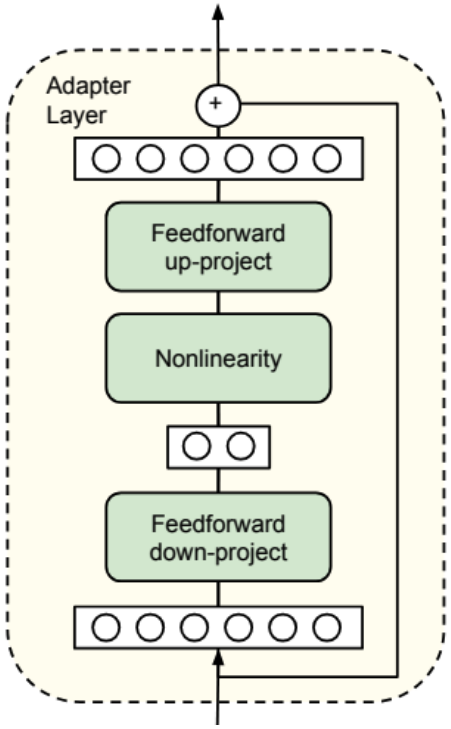


Prefix-Tuning: Optimizing Continuous Prompts for Generation, ACL 2021

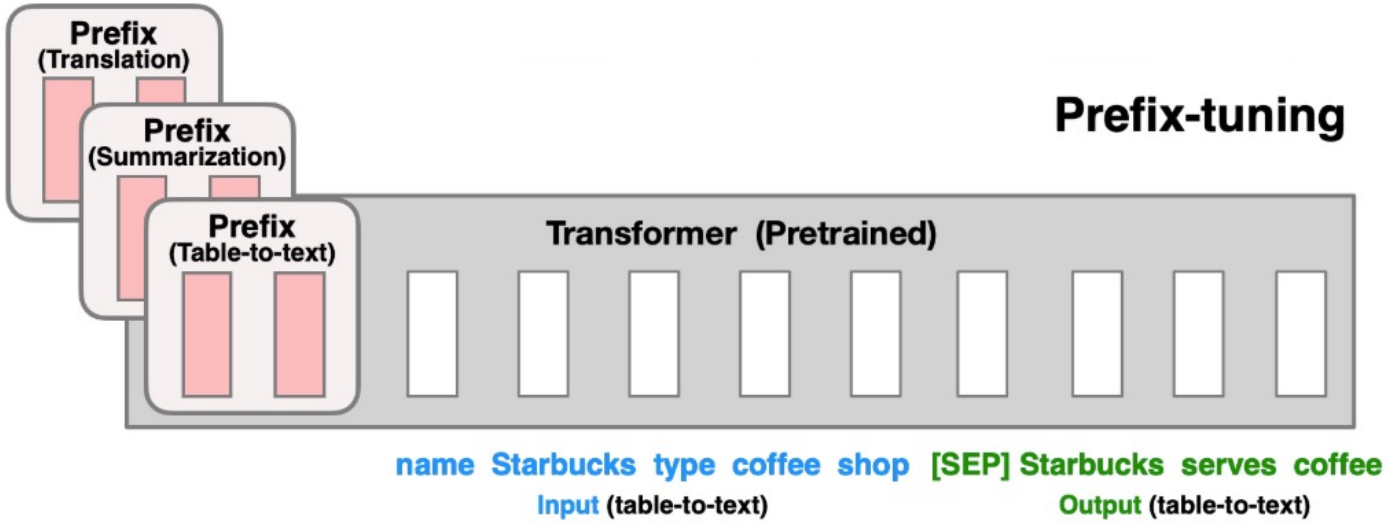
Existing solution problem

- Adding adapter layers
  - no direct ways to bypass the extra compute in adapter layers
  - adapter layers have to be processed sequentially that make latency high





- Prefix tuning
  - difficult to optimize
  - performance changes non-monotonically in trainable parameters because reducing the sequence length



LoRA

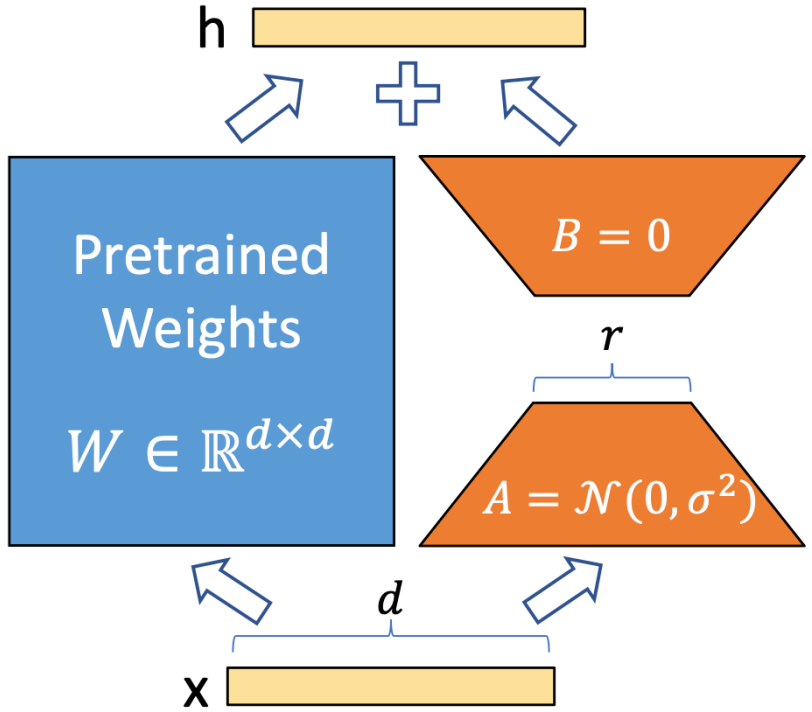
- Other works learned over-parametrized models in fact reside on a low intrinsic dimension.

$$h = W_0 x + \Delta W x = W_0 x + BAx$$

$$B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}$$

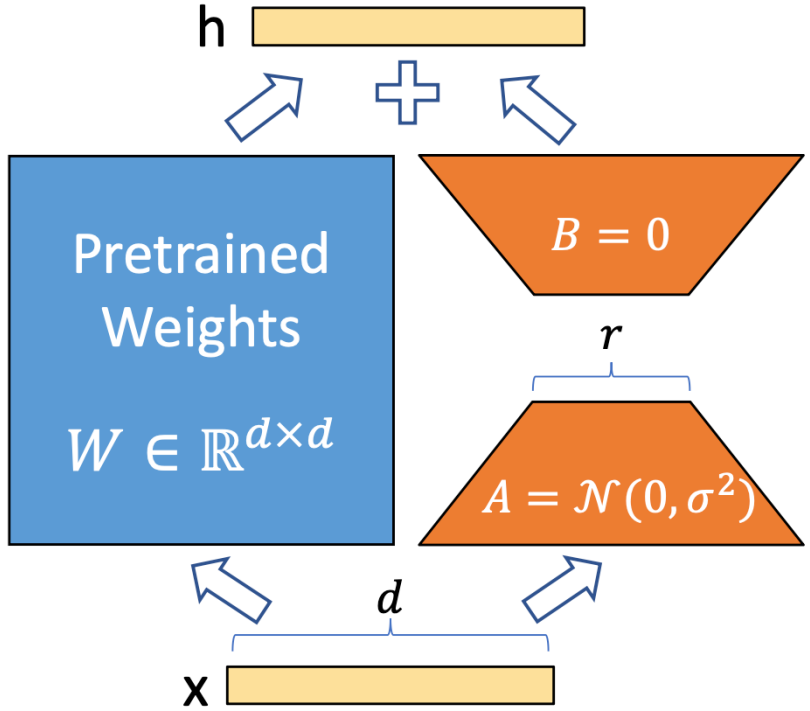
$$r \ll \min(d, k)$$

$$|\Theta| = 2 \times \hat{L}_{LoRA} \times d_{model} \times r$$



LoRA

- Generalization of Full Fine-tuning
  - Increase trainable parameters, training LoRA roughly converges to training the original model
  - adapter-based methods converges to an MLP
- No Additional Inference Latency
- Practical Benefits
  - On GPT-3 175B
  - Reduce the VRAM consumption during training from 1.2TB to 350GB
  - Checkpoint size is reduced by roughly 10,000× (from 350GB to 35MB)



Experiment

- different adaptation methods on the GLUE benchmark
  - NLU

Model & Method	# Trainable Parameters	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg.
RoB_{base} (FT)*	125.0M	**87.6**	94.8	90.2	**63.6**	92.8	**91.9**	78.7	91.2	86.4
RoB_{base} (BitFit)*	0.1M	84.7	93.7	**92.7**	62.0	91.8	84.0	81.5	90.8	85.2
RoB_{base} (Adpt^{D})*	0.3M	$87.1_{\pm.0}$	$94.2_{\pm.1}$	$88.5_{\pm1.1}$	$60.8_{\pm.4}$	$93.1_{\pm.1}$	$90.2_{\pm.0}$	$71.5_{\pm2.7}$	$89.7_{\pm.3}$	84.4
RoB_{base} (Adpt^{D})*	0.9M	$87.3_{\pm.1}$	$94.7_{\pm.3}$	$88.4_{\pm.1}$	$62.6_{\pm.9}$	$93.0_{\pm.2}$	$90.6_{\pm.0}$	$75.9_{\pm2.2}$	$90.3_{\pm.1}$	85.4
RoB_{base} (LoRA)	0.3M	$87.5_{\pm.3}$	$\mathbf{95.1}_{\pm.2}$	$89.7_{\pm.7}$	$63.4_{\pm1.2}$	$\mathbf{93.3}_{\pm.3}$	$90.8_{\pm.1}$	$\mathbf{86.6}_{\pm.7}$	$\mathbf{91.5}_{\pm.2}$	**87.2**
$\text{RoB}_{\text{large}}$ (FT)*	355.0M	90.2	**96.4**	**90.9**	68.0	94.7	**92.2**	86.6	92.4	88.9
$\text{RoB}_{\text{large}}$ (LoRA)	0.8M	$\mathbf{90.6}_{\pm.2}$	$96.2_{\pm.5}$	$\mathbf{90.9}_{\pm1.2}$	$\mathbf{68.2}_{\pm1.9}$	$\mathbf{94.9}_{\pm.3}$	$91.6_{\pm.1}$	$\mathbf{87.4}_{\pm2.5}$	$\mathbf{92.6}_{\pm.2}$	**89.0**
$\text{RoB}_{\text{large}}$ (Adpt^{P})†	3.0M	$90.2_{\pm.3}$	$96.1_{\pm.3}$	$90.2_{\pm.7}$	$\mathbf{68.3}_{\pm1.0}$	$\mathbf{94.8}_{\pm.2}$	$\mathbf{91.9}_{\pm.1}$	$83.8_{\pm2.9}$	$92.1_{\pm.7}$	88.4
$\text{RoB}_{\text{large}}$ (Adpt^{P})†	0.8M	$\mathbf{90.5}_{\pm.3}$	$\mathbf{96.6}_{\pm.2}$	$89.7_{\pm1.2}$	$67.8_{\pm2.5}$	$\mathbf{94.8}_{\pm.3}$	$91.7_{\pm.2}$	$80.1_{\pm2.9}$	$91.9_{\pm.4}$	87.9
$\text{RoB}_{\text{large}}$ (Adpt^{H})†	6.0M	$89.9_{\pm.5}$	$96.2_{\pm.3}$	$88.7_{\pm2.9}$	$66.5_{\pm4.4}$	$94.7_{\pm.2}$	$92.1_{\pm.1}$	$83.4_{\pm1.1}$	$91.0_{\pm1.7}$	87.8
$\text{RoB}_{\text{large}}$ (Adpt^{H})†	0.8M	$90.3_{\pm.3}$	$96.3_{\pm.5}$	$87.7_{\pm1.7}$	$66.3_{\pm2.0}$	$94.7_{\pm.2}$	$91.5_{\pm.1}$	$72.9_{\pm2.9}$	$91.5_{\pm.5}$	86.4
$\text{RoB}_{\text{large}}$ (LoRA)†	0.8M	$\mathbf{90.6}_{\pm.2}$	$96.2_{\pm.5}$	$\mathbf{90.2}_{\pm1.0}$	$68.2_{\pm1.9}$	$\mathbf{94.8}_{\pm.3}$	$91.6_{\pm.2}$	$\mathbf{85.2}_{\pm1.1}$	$\mathbf{92.3}_{\pm.5}$	**88.6**
DeB_{XXL} (FT)*	1500.0M	91.8	**97.2**	92.0	72.0	**96.0**	92.7	93.9	92.9	91.1
DeB_{XXL} (LoRA)	4.7M	$\mathbf{91.9}_{\pm.2}$	$96.9_{\pm.2}$	$\mathbf{92.6}_{\pm.6}$	$\mathbf{72.4}_{\pm1.1}$	$\mathbf{96.0}_{\pm.1}$	$\mathbf{92.9}_{\pm.1}$	$\mathbf{94.9}_{\pm.4}$	$\mathbf{93.0}_{\pm.2}$	**91.3**

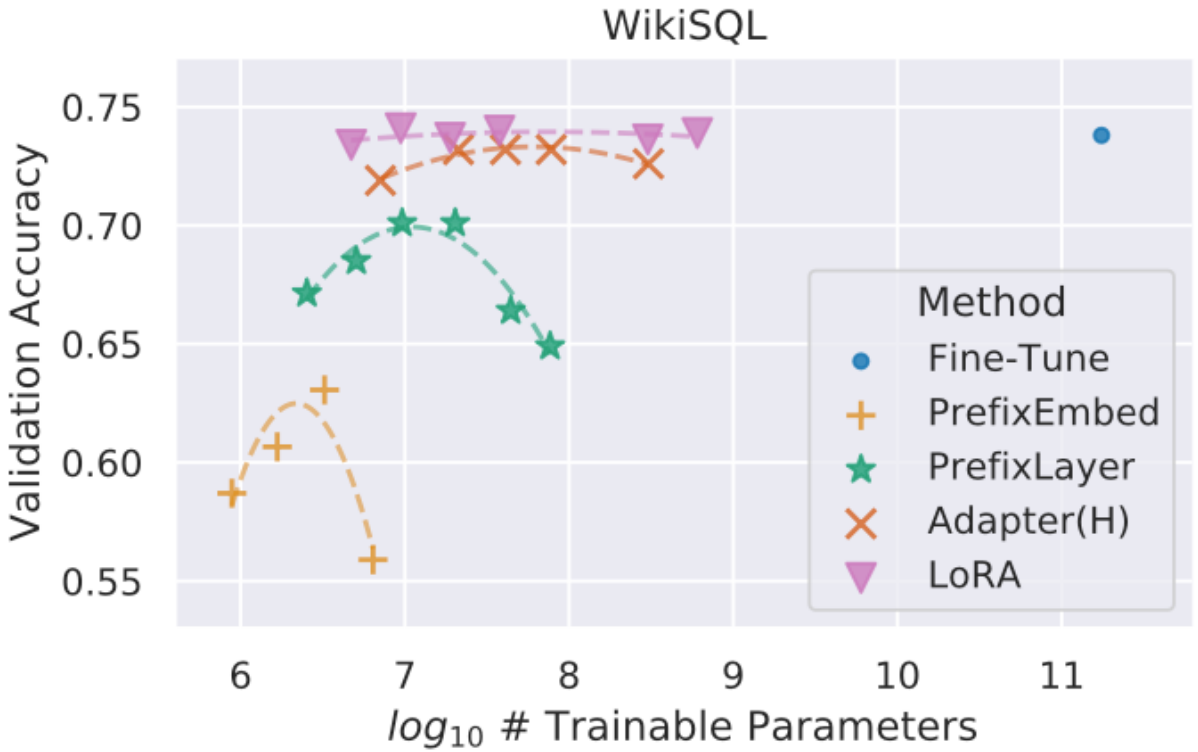
- GPT-2 medium (M) and large (L)
  - E2E NLG Challenge

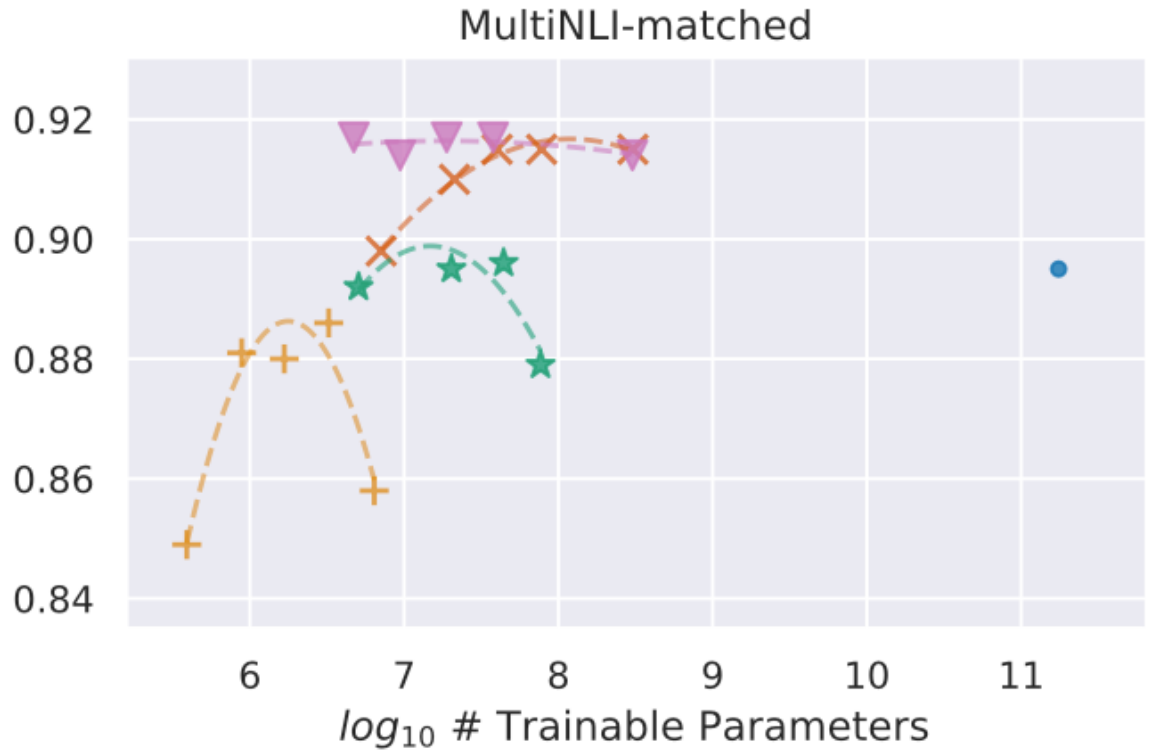
Model & Method	# Trainable Parameters	E2E NLG Challenge BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 M (FT)*	354.92M	68.2	8.62	46.2	71.0	2.47
GPT-2 M ($\text{Adapter}^{\text{L}}$)*	0.37M	66.3	8.41	45.0	69.8	2.40
GPT-2 M ($\text{Adapter}^{\text{L}}$)*	11.09M	68.9	8.71	46.1	71.3	2.47
GPT-2 M ($\text{Adapter}^{\text{H}}$)	11.09M	$67.3_{\pm.6}$	$8.50_{\pm.07}$	$46.0_{\pm.2}$	$70.7_{\pm.2}$	$2.44_{\pm.01}$
GPT-2 M (FT^{Top2})*	25.19M	68.1	8.59	46.0	70.8	2.41
GPT-2 M (PreLayer)*	0.35M	69.7	8.81	46.1	71.4	2.49
GPT-2 M (LoRA)	0.35M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.85}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{71.8}_{\pm.1}$	$\mathbf{2.53}_{\pm.02}$
GPT-2 L (FT)*	774.03M	68.5	8.78	46.0	69.9	2.45
GPT-2 L ($\text{Adapter}^{\text{L}}$)	0.88M	$69.1_{\pm.1}$	$8.68_{\pm.03}$	$46.3_{\pm.0}$	$71.4_{\pm.2}$	$\mathbf{2.49}_{\pm.0}$
GPT-2 L ($\text{Adapter}^{\text{L}}$)	23.00M	$68.9_{\pm.3}$	$8.70_{\pm.04}$	$46.1_{\pm.1}$	$71.3_{\pm.2}$	$2.45_{\pm.02}$
GPT-2 L (PreLayer)*	0.77M	70.3	8.85	46.2	71.7	2.47
GPT-2 L (LoRA)	0.77M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.89}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{72.0}_{\pm.2}$	$2.47_{\pm.02}$

Experiment

- not all methods benefit monotonically from having more trainable parameters

Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 ($\text{Adapter}^{\text{H}}$)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 ($\text{Adapter}^{\text{H}}$)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1





Ablation study

- Which weight matrices in transformer should we apply LoRA to?

	# of Trainable Parameters = 18M						
Weight Type	W_q	W_k	W_v	W_o	W_q, W_k	W_q, W_v	W_q, W_k, W_v, W_o
Rank r	8	8	8	8	4	4	2
WikiSQL ($\pm$0.5%)	70.4	70.0	73.0	73.2	71.4	**73.7**	**73.7**
MultiNLI ($\pm$0.1%)	91.0	90.8	91.0	91.3	91.3	91.3	**91.7**

Ablation study

- What is the optimal rank r for LoRA?

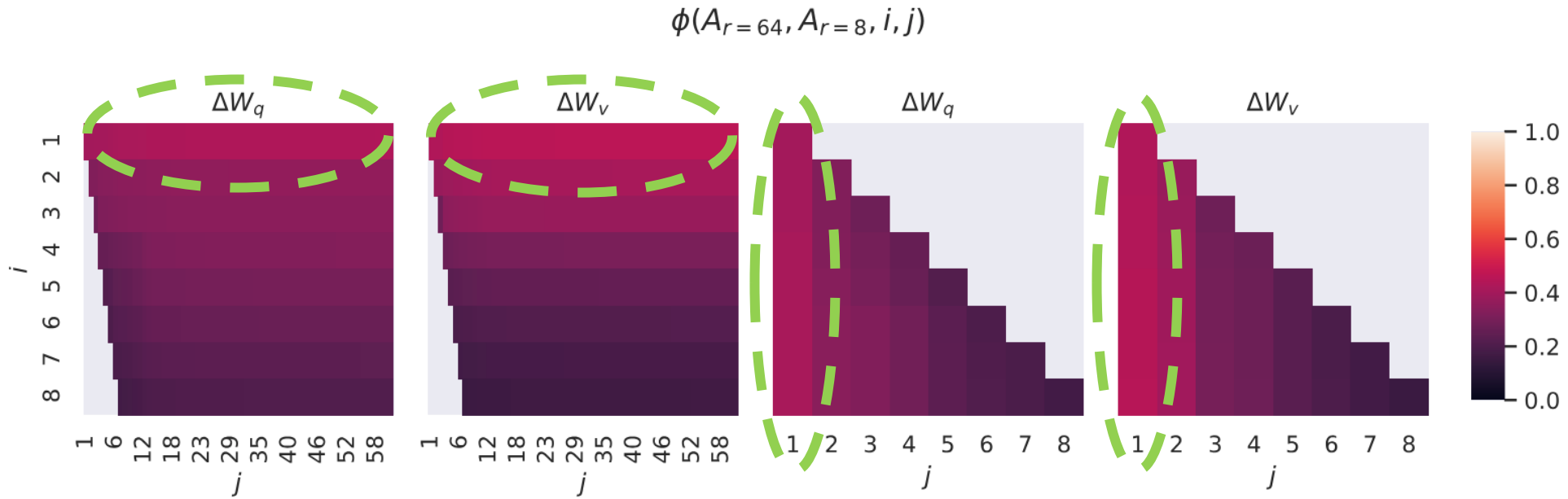
	Weight Type	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 64$
WikiSQL($\pm 0.5\%$)	W_q	68.8	69.6	70.5	70.4	70.0
	W_q, W_v	73.4	73.3	73.7	73.8	73.5
	W_q, W_k, W_v, W_o	74.1	73.7	74.0	74.0	73.9
MultiNLI ($\pm 0.1\%$)	W_q	90.7	90.9	91.1	90.7	90.7
	W_q, W_v	91.3	91.4	91.3	91.6	91.4
	W_q, W_k, W_v, W_o	91.2	91.7	91.7	91.5	91.4

- Increasing r does not cover a more meaningful subspace
- Low-rank adaptation matrix is sufficient.

Ablation study

- Subspace similarity between different r

$$\phi(A_{r=8}, A_{r=64}, i, j) = \frac{||U^{i\top}_{A_{r=8}} U^{j}_{A_{r=64}}||^2_F}{\min(i,j)} \in [0,1]$$



- Top singular vector overlap significantly between $A_{r=8}$ and $A_{r=64}$, so $r = 1$ performs quite well

Ablation study

- How does the adaptation matrix ΔW compare to W?
  - project W onto the r-dimensional subspace of ΔW by computing $U^T W V$, with U/V being the left/right singular-vector matrix of ΔW

	$r = 4$			$r = 64$		
	ΔW_q	W_q	Random	ΔW_q	W_q	Random
$\|\|U^\top W_q V^\top\|\|_F =$	0.32	21.67	0.02	1.90	37.71	0.33
$\|\|W_q\|\|_F = 61.95$	$\|\|\Delta W_q\|\|_F = 6.91$			$\|\|\Delta W_q\|\|_F = 3.57$		

Ablation study

- How does the adaptation matrix ΔW compare to W?
  - project W onto the r-dimensional subspace of ΔW by computing $U^T W V$, with U/V being the left/right singular-vector matrix of ΔW

	$r = 4$			$r = 64$		
	ΔW_q	W_q	Random	ΔW_q	W_q	Random
$\|\|U^\top W_q V^\top\|\|_F =$	0.32	21.67	0.02	1.90	37.71	0.33
$\|\|W_q\|\|_F = 61.95$	$\|\|\Delta W_q\|\|_F = 6.91$			$\|\|\Delta W_q\|\|_F = 3.57$		

- ΔW_q *vs Random*, indicating that ΔW_q amplifies some features that are already in W

Ablation study

- How does the adaptation matrix ΔW compare to W?
  - project W onto the r-dimensional subspace of ΔW by computing $U^T WV$, with U/V being the left/right singular-vector matrix of ΔW

	$r = 4$			$r = 64$		
	ΔW_q	W_q	Random	ΔW_q	W_q	Random
$\|\|U^\top W_q V^\top\|\|_F =$	0.32	21.67	0.02	1.90	37.71	0.33
$\|\|W_q\|\|_F = 61.95$	$\|\|\Delta W_q\|\|_F =$ 6.91			$\|\|\Delta W_q\|\|_F = 3.57$		

  - ΔW_q *vs Random*, indicating that ΔW_q amplifies some features that are already in W
  - Amplification factor is rather huge: $21.5 \approx \frac{6.91}{0.32}$ *for* $r = 4$, larger than $r = 64$

Related Work – GILL

Generating Images with Multimodal Language Models

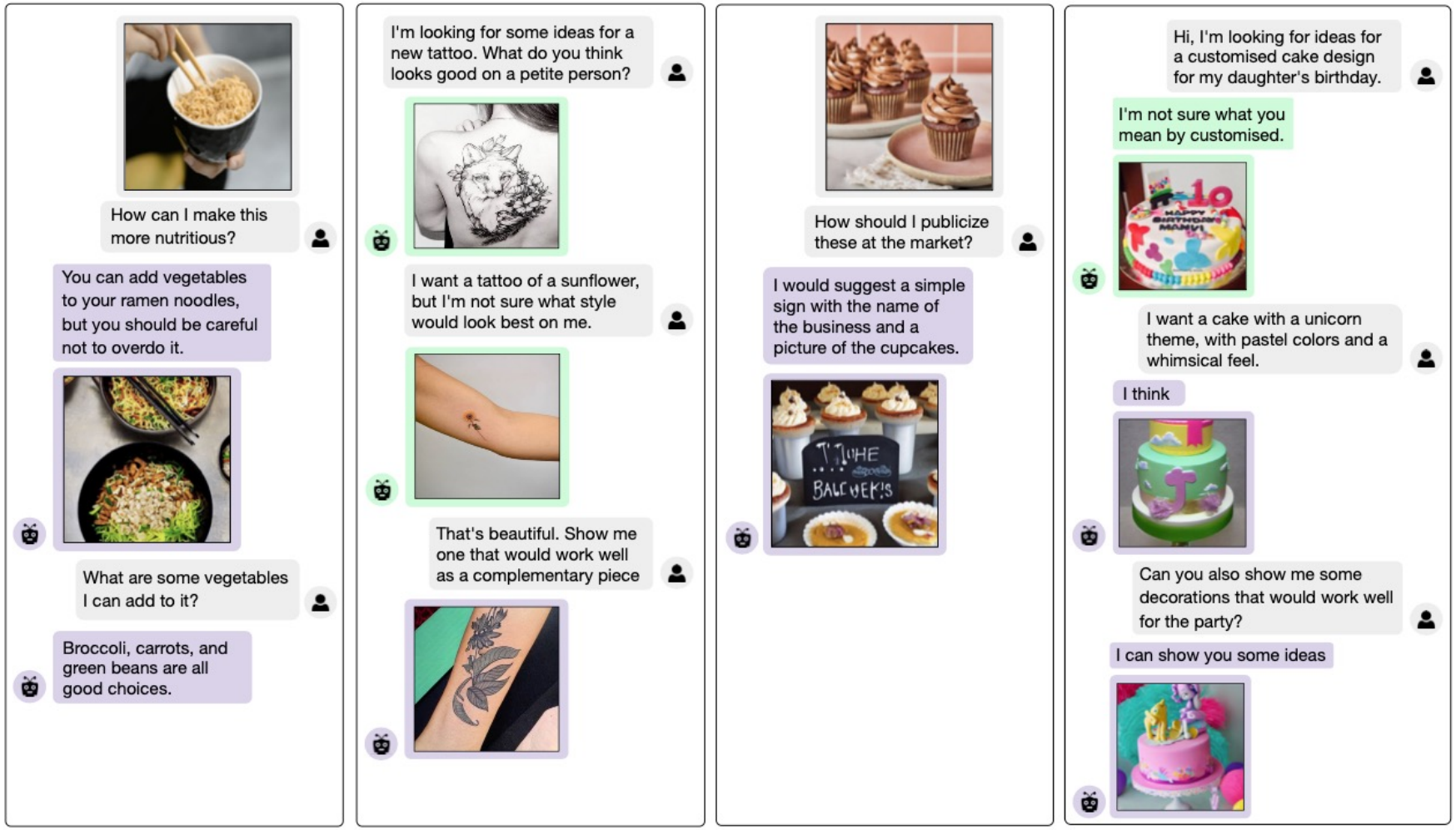
Jing Yu Koh
Carnegie Mellon University
jingyuk@cs.cmu.edu

Daniel Fried
Carnegie Mellon University
dfried@cs.cmu.edu

Ruslan Salakhutdinov
Carnegie Mellon University
rsalakhu@cs.cmu.edu

NeurIPS 2023

Introduction

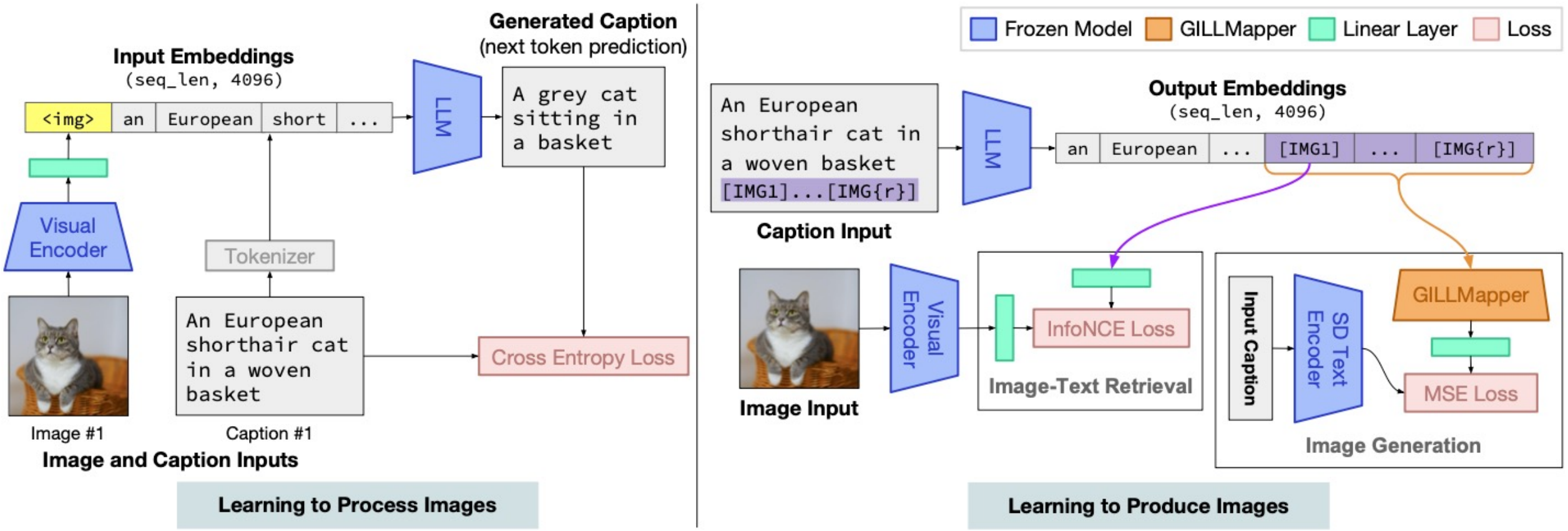


Multimodal Dialogue

Our model can generate multimodal dialogue, weaving together text, retrieved images, and generated images.

User prompts Retrieved Generated

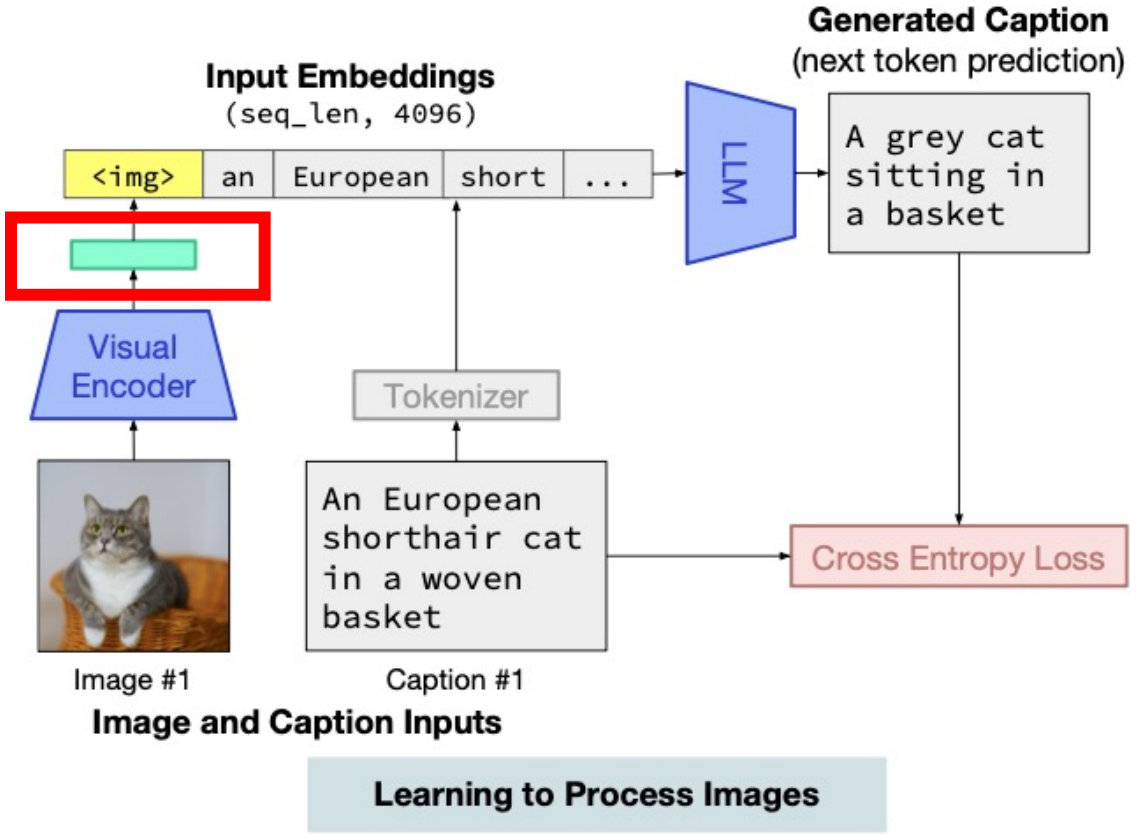
Framework


Generated Caption
(next token prediction)
Input Embeddings
(seq_len, 4096)
<img>
an
European
short
...
LLM
A grey cat sitting in a basket
Visual Encoder
Tokenizer
An European shorthair cat in a woven basket
Cross Entropy Loss
Image #1
Caption #1
Image and Caption Inputs
Learning to Process Images
Frozen Model
GILLMapper
Linear Layer
Loss
An European shorthair cat in a woven basket [IMG1]...[IMG{r}]
Caption Input
LLM
Output Embeddings
(seq_len, 4096)
an
European
...
[IMG1]
...
[IMG{r}]
Image Input
Visual Encoder
InfoNCE Loss
Image-Text Retrieval
Input Caption
SD Text Encoder
GILLMapper
MSE Loss
Image Generation
Learning to Produce Images

Method

- Learning to Process Images
  - v_φ is CLIP

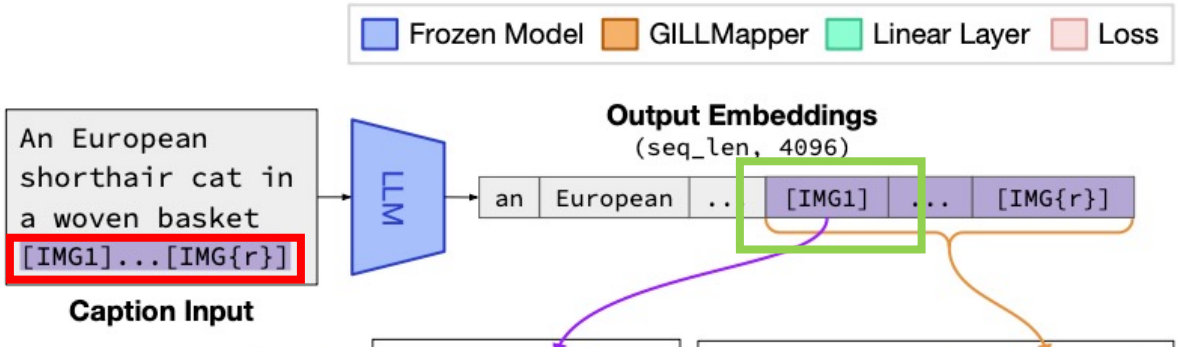
$$l_c(x, y) = -\sum_{t=1}^{T} \log p_\theta(s_t \mid v_\phi(x)^T \mathbf{W}_{\text{cap}}, s_1, \ldots, s_{t-1})$$



Method

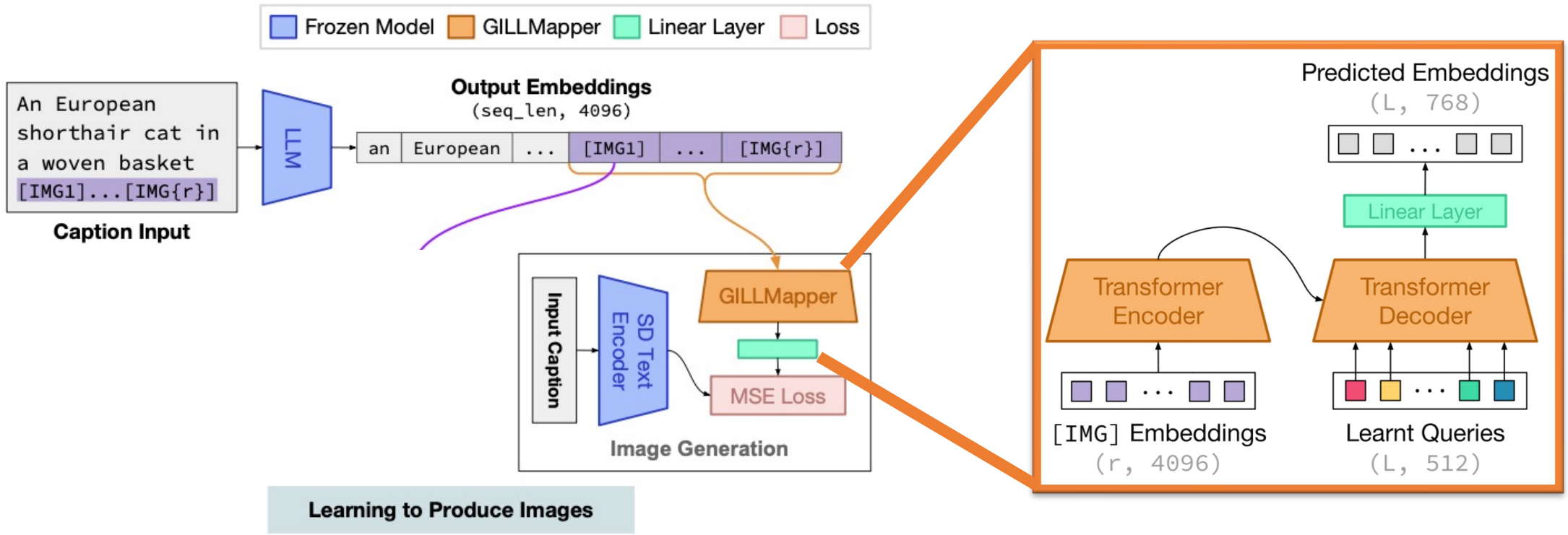
- Learning to Produce Images

$$l_p(y) = -\log p_{\{\theta \cup \mathbf{E}_{\text{img}}\}}(\texttt{[IMG1]} \mid s_1, \ldots, s_t)$$



Method

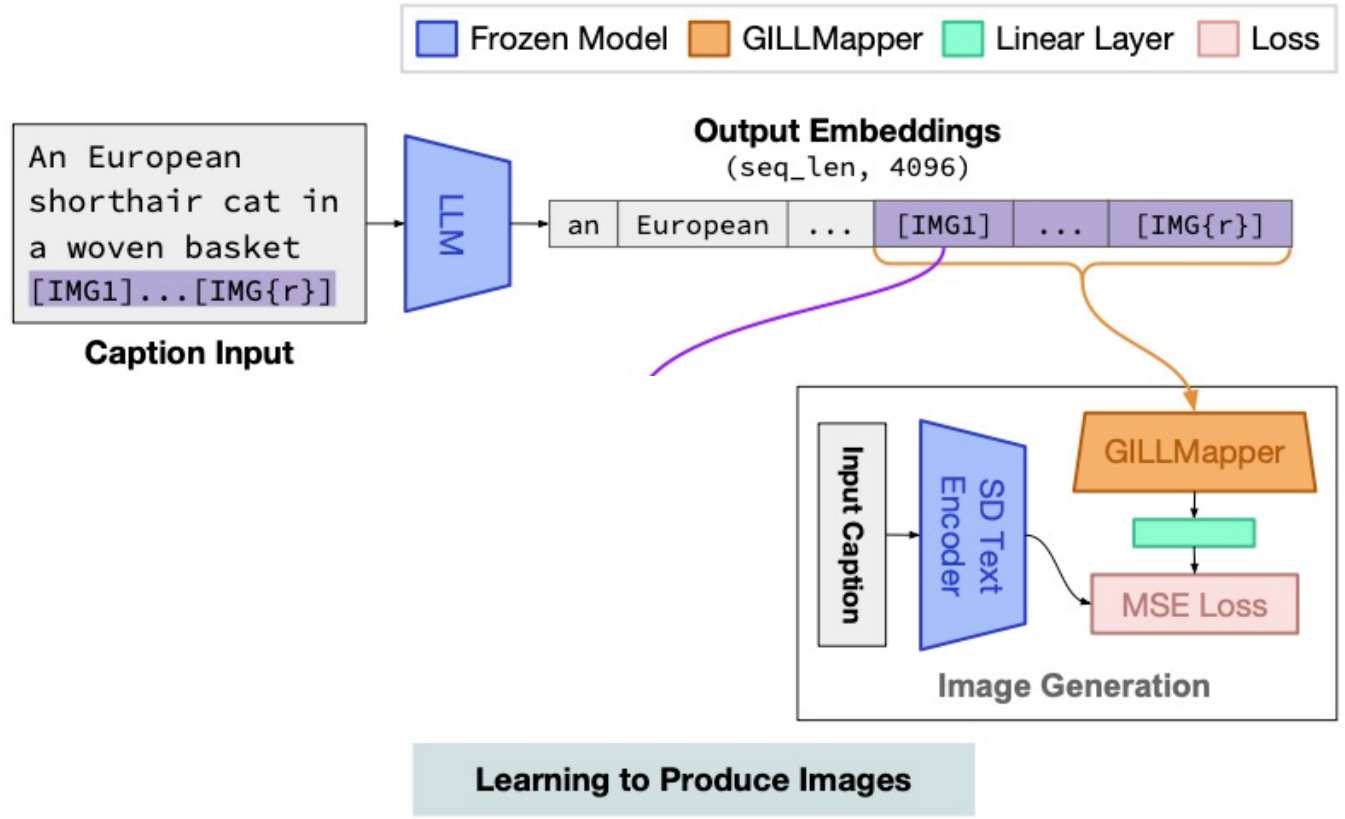
- Learning to Process Images
  - Novel Image Generation



$$l_g(y) = \| f_{\omega}\big(h_{\{\theta \cup \mathbf{E}_{\text{img}}\}}(y, \texttt{[IMG\{1\}]}), \ldots, h_{\{\theta \cup \mathbf{E}_{\text{img}}\}}(y, \texttt{[IMG\{r\}]}), q_1, \ldots, q_L\big) - T_{\psi}(y)\|_2^2$$

Method

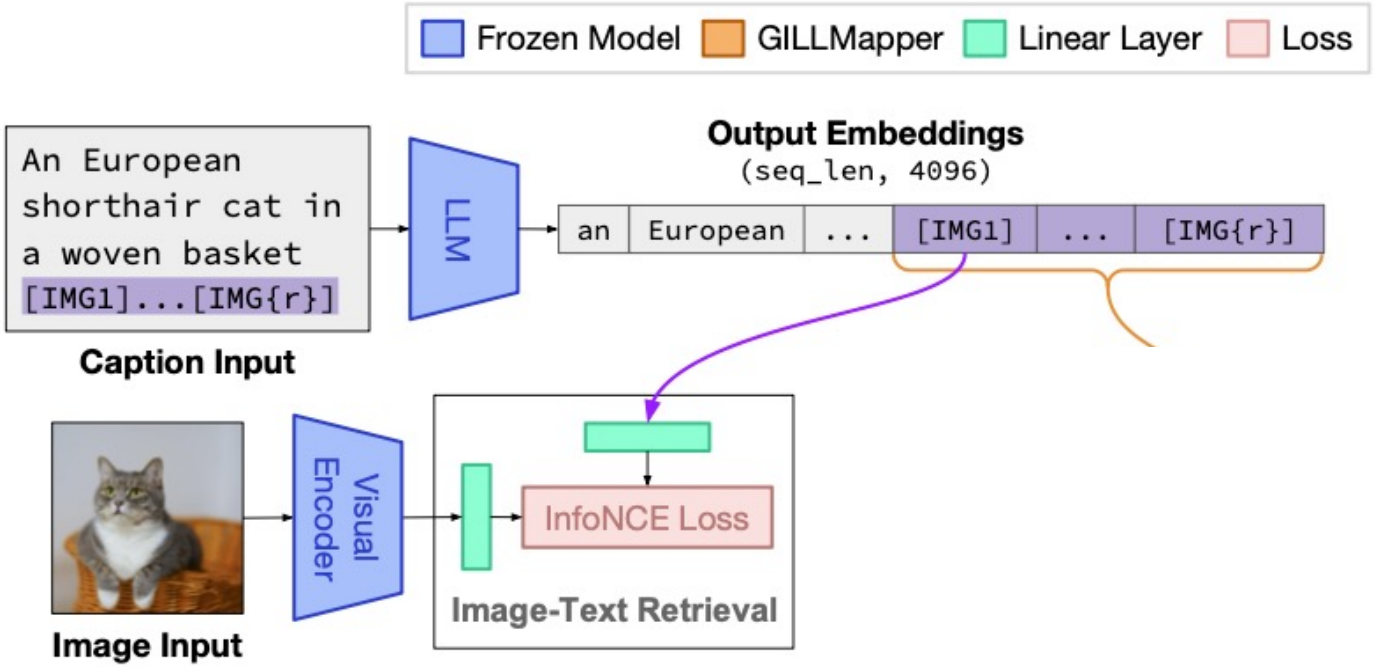
- Learning to Process Images
  - Novel Image Generation
  - G_ψ is Stable Diffusion



$$\text{Generated Image} = G_\psi(f_\omega(h_{\{\theta \cup \mathbf{E}_{\text{img}}\}}(y, \texttt{[IMG\{1\}]}), \ldots, h_{\{\theta \cup \mathbf{E}_{\text{img}}\}}(y, \texttt{[IMG\{r\}]}), q_1, \ldots, q_L))$$

Method

- Image Retrieval

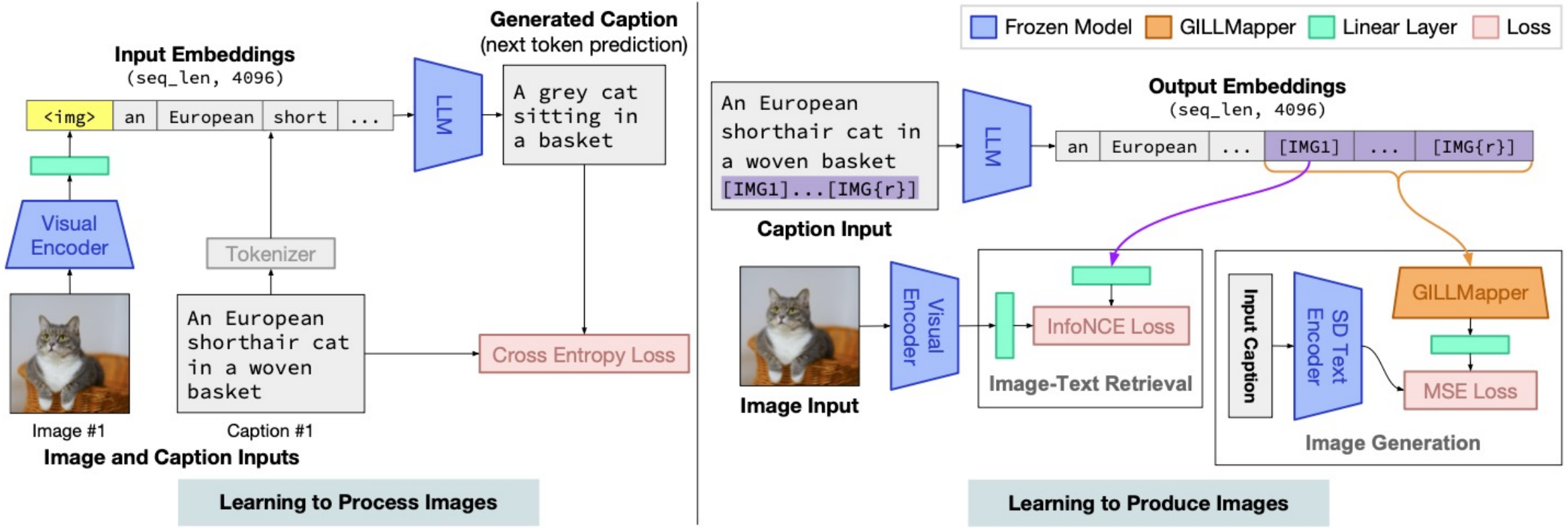


$$l_r(\mathbf{x}_i, \mathbf{y}_i) = -\log \frac{\exp(\text{sim}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{W}_{\text{t2i}})/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(\mathbf{x}_j, \mathbf{y}_i, \mathbf{W}_{\text{t2i}})/\tau)} - \log \frac{\exp(\text{sim}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{W}_{\text{i2t}})/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(\mathbf{x}_i, \mathbf{y}_j, \mathbf{W}_{\text{i2t}})/\tau)}$$

where the similarity is computed as

$$\text{sim}(x, y, \mathbf{W}) = \frac{\left(\mathbf{W}^T v_\phi(x)\right)^T \left(\mathbf{W}^T h_{\{\theta \cup \mathbf{E}_{\text{img}}\}}(y, \texttt{[IMG1]})\right)}{\|\mathbf{W}^T v_\phi(x)\| \left\|\mathbf{W}^T h_{\{\theta \cup \mathbf{E}_{\text{img}}\}}(y, \texttt{[IMG1]})\right\|}$$

Method

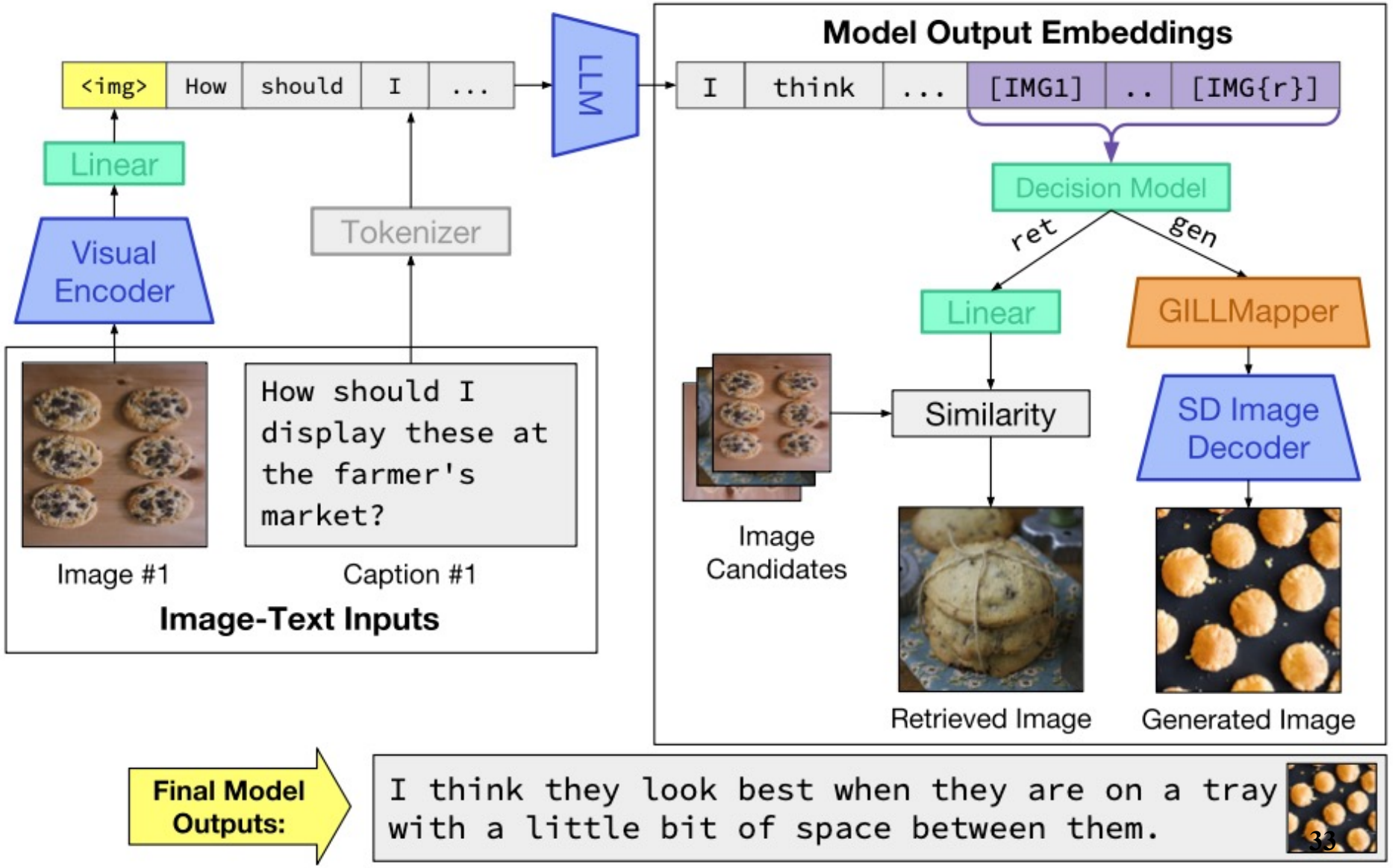


- Total loss

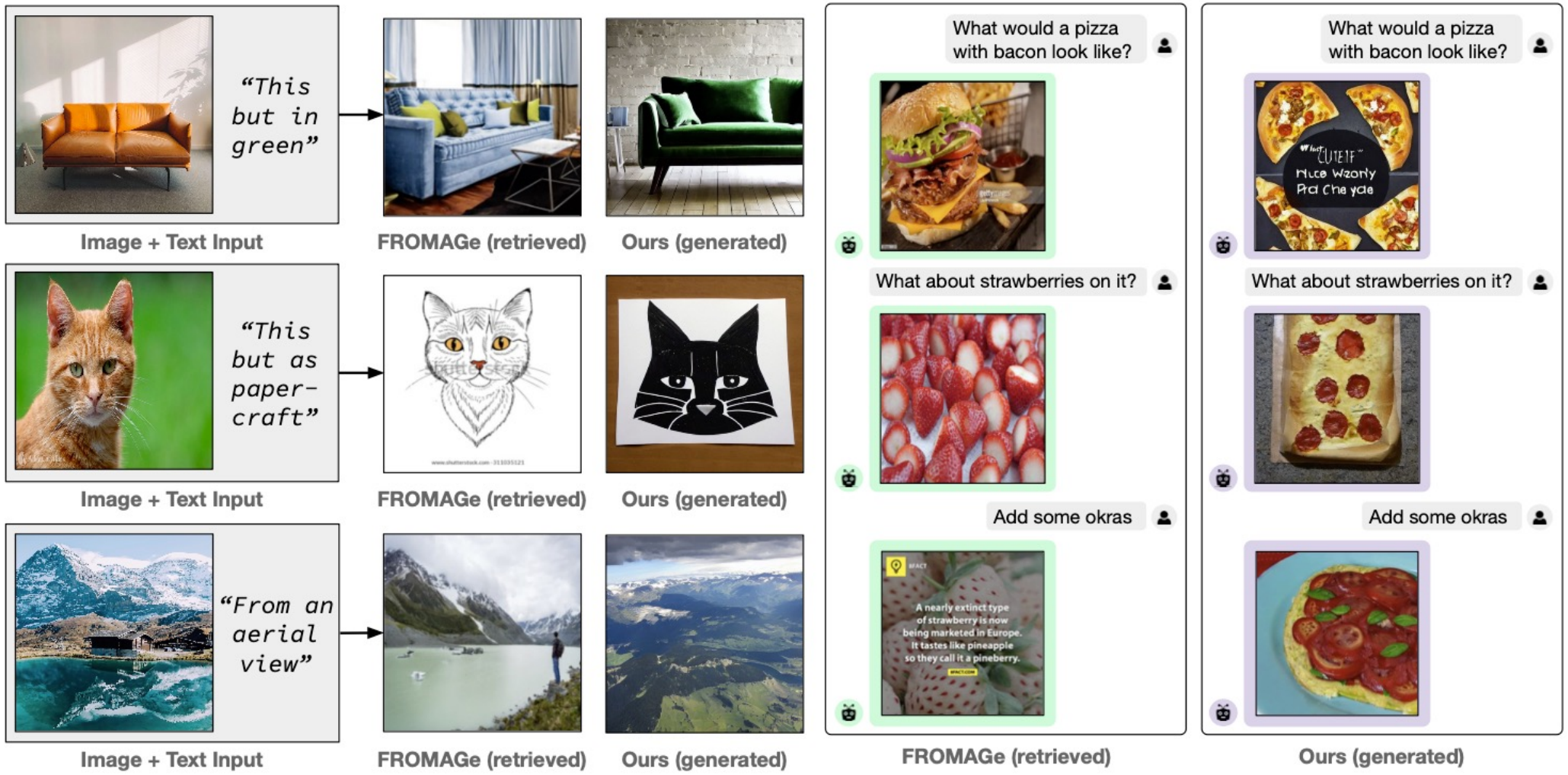
$$\min_{\mathbf{W}_{\text{i2t}}, \mathbf{W}_{\text{t2i}}, \mathbf{W}_{\text{cap}}, \mathbf{E}_{\text{img}}, \omega, q_{1:L}} \frac{1}{N} \sum_{i=1}^{N} \left(l_c(\mathbf{x}_i, \mathbf{y}_i) + l_p(\mathbf{y}_i) + l_g(\mathbf{y}_i) + l_r(\mathbf{x}_i, \mathbf{y}_i) \right)$$

Method

- Inference



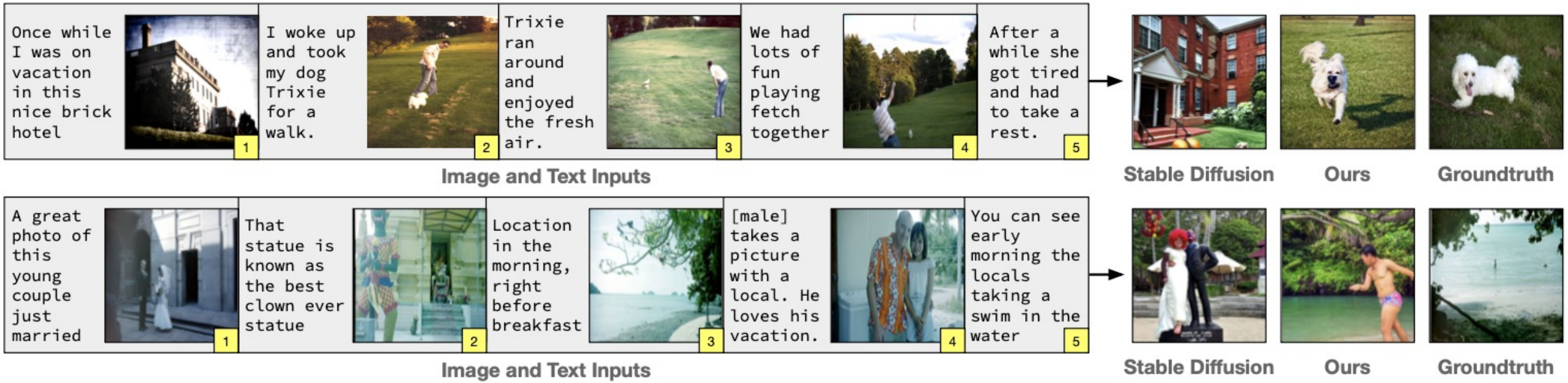
Experiment



Comparison Against FROMAGe

Our model composites multimodal information to produce relevant image and text outputs. It can outperform baseline models that are limited to image retrieval.

Experiment



Visual Storytelling

Our model can condition on interleaved image-and-text inputs to generate more relevant images compared to non-LLM based text-to-image generation models.

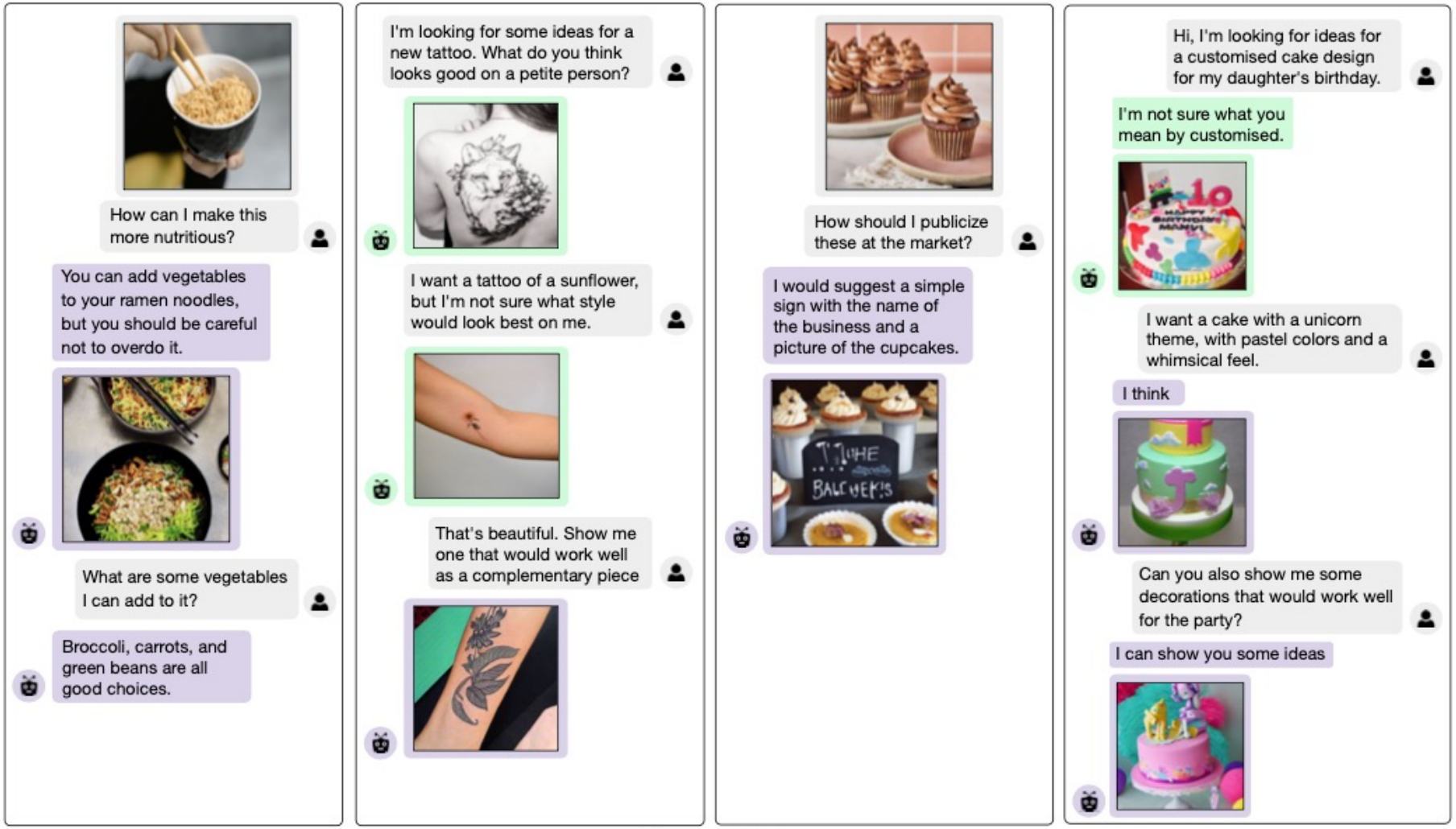
Experiment

Model	CLIP Similarity (↑)			LPIPS (↓)		
	1 caption	5 captions	5 caps, 4 images	1 caption	5 captions	5 caps, 4 images
GLIDE [38]	0.582	0.591	-	0.753	0.745	-
Stable Diffusion [49]	**0.592** ±0.0007	0.598 ± 0.0006	-	0.703 ±0.0003	0.704 ± 0.0004	-
GILL (ours)	0.581 ±0.0005	**0.612** ±0.0011	**0.641** ±0.0011	**0.702** ±0.0004	**0.696** ±0.0008	**0.693** ±0.0008

Model	CLIP Similarity (↑)			LPIPS (↓)		
	1 round	5 rounds	10 rounds	1 round	5 rounds	10 rounds
GLIDE [38]	**0.562**	0.595	0.587	0.800	0.794	0.799
Stable Diffusion [49]	0.552 ±0.0015	**0.629** ±0.0015	0.622 ±0.0012	**0.742** ±0.0010	0.722 ±0.0012	0.723 ±0.0008
GILL (ours)	0.528 ±0.0014	0.621 ±0.0009	**0.645** ±0.0010	**0.742** ±0.0022	**0.718** ±0.0028	**0.714** ±0.0006

- GILL leverages an LLM backbone, which allows it to inherit some of the LLM's capabilities, such as improved sensitivity to long inputs.

Experiment



- A novel feature of GILL is its ability to process interleaved image-text inputs, which most existing text-to-image generation models are unable to handle.

Conclusion of related works

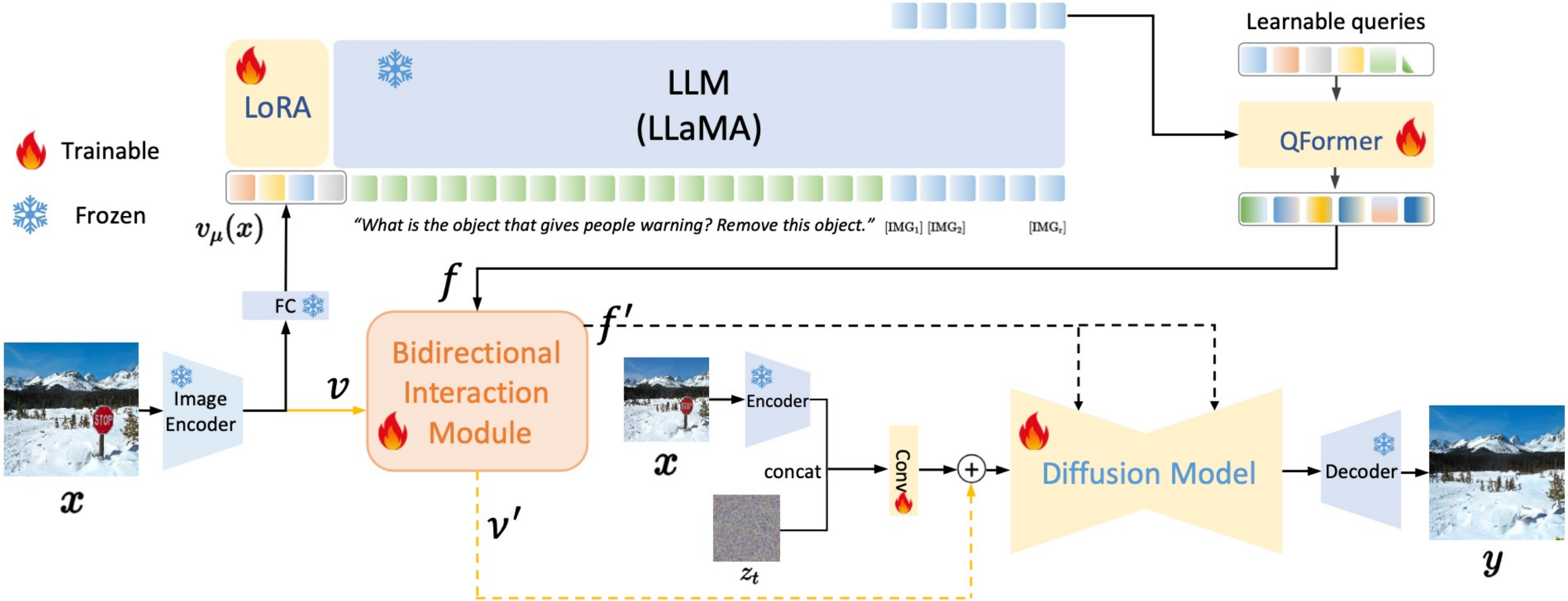
- LoRA: Low-Rank Adaptation of Large Language Models
  - ✓ Efficient adaptation strategy while retaining high model quality.

- Generating Images with Multimodal Language Models
  - ✓ Mapping text-only LLMs to strong visual models.
  - ✓ Enables to process arbitrarily interleaved image-and-text inputs, and output generated text and generated images.

Outline

- Introduction
- Related Work
- Framework
- Method
- Experiment
- Conclusion

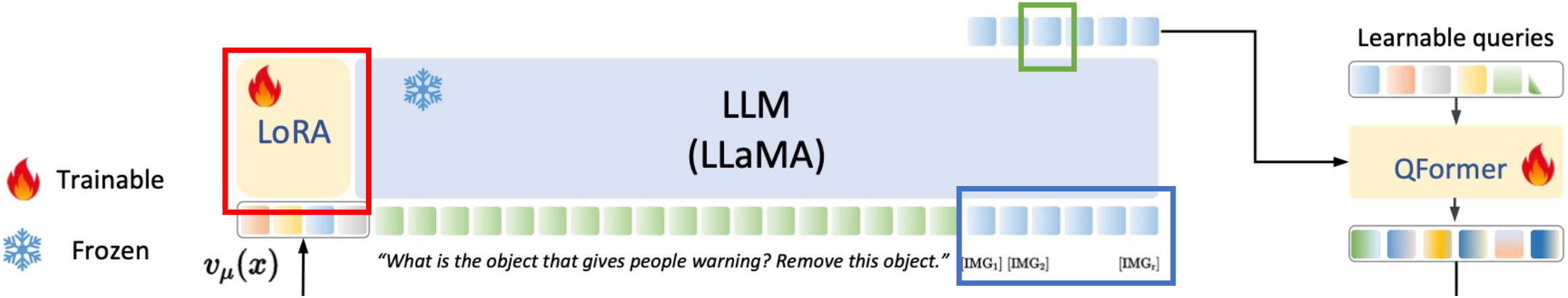
Framework


Trainable
Frozen
LoRA
LLM
(LLaMA)
Learnable queries
QFormer
$v_\mu(x)$
"What is the object that gives people warning? Remove this object." [IMG1] [IMG2] [IMGr]
f
FC
f'
Bidirectional
Interaction
Module
Image
Encoder
v
x
Encoder
concat
Conv
z_t
v'
Diffusion Model
Decoder
y

Outline

- Introduction
- Related Work
- Framework
- Method
- Experiment
- Conclusion

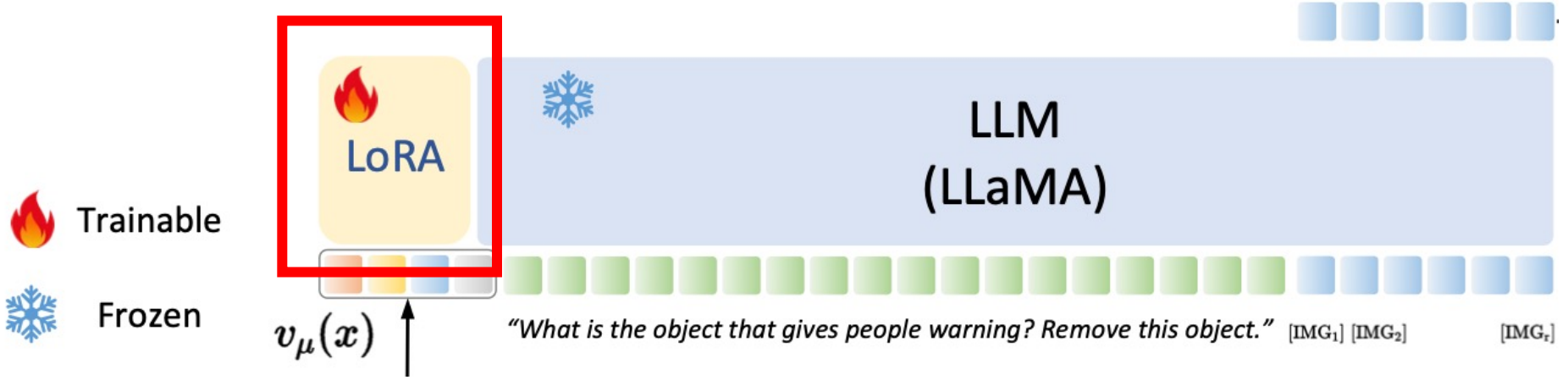
Language model loss



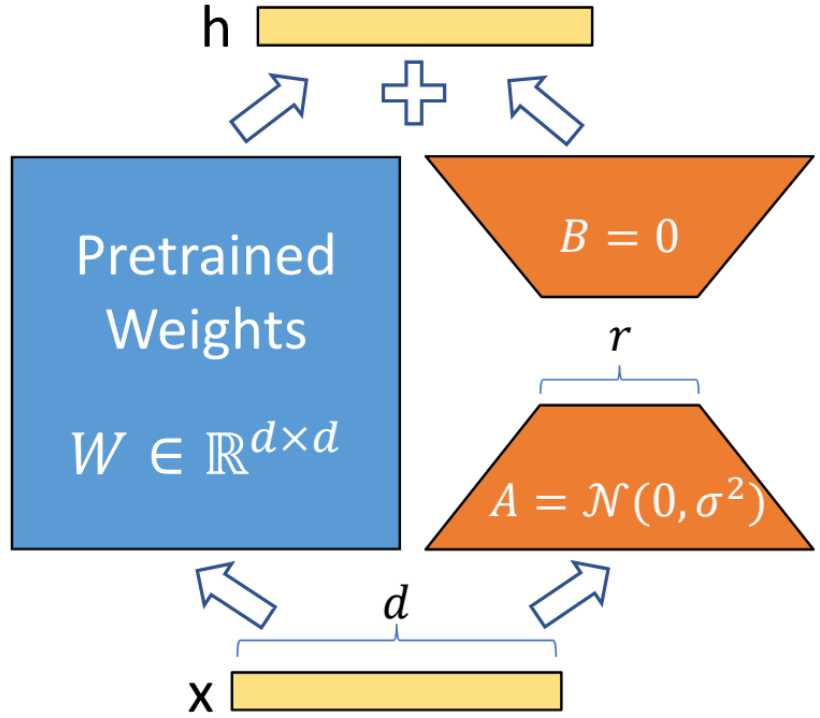
$$L_{\text{LLM}}(c) = -\sum_{i=1}^{r} \log p_{\{\theta \cup \mathbf{E}\}}([\text{IMG}_i] \mid v_\mu(x),$$

$$s_1, ..., s_T, [\text{IMG}_1], \ldots, [\text{IMG}_{i-1}]) \qquad (2)$$

LoRA

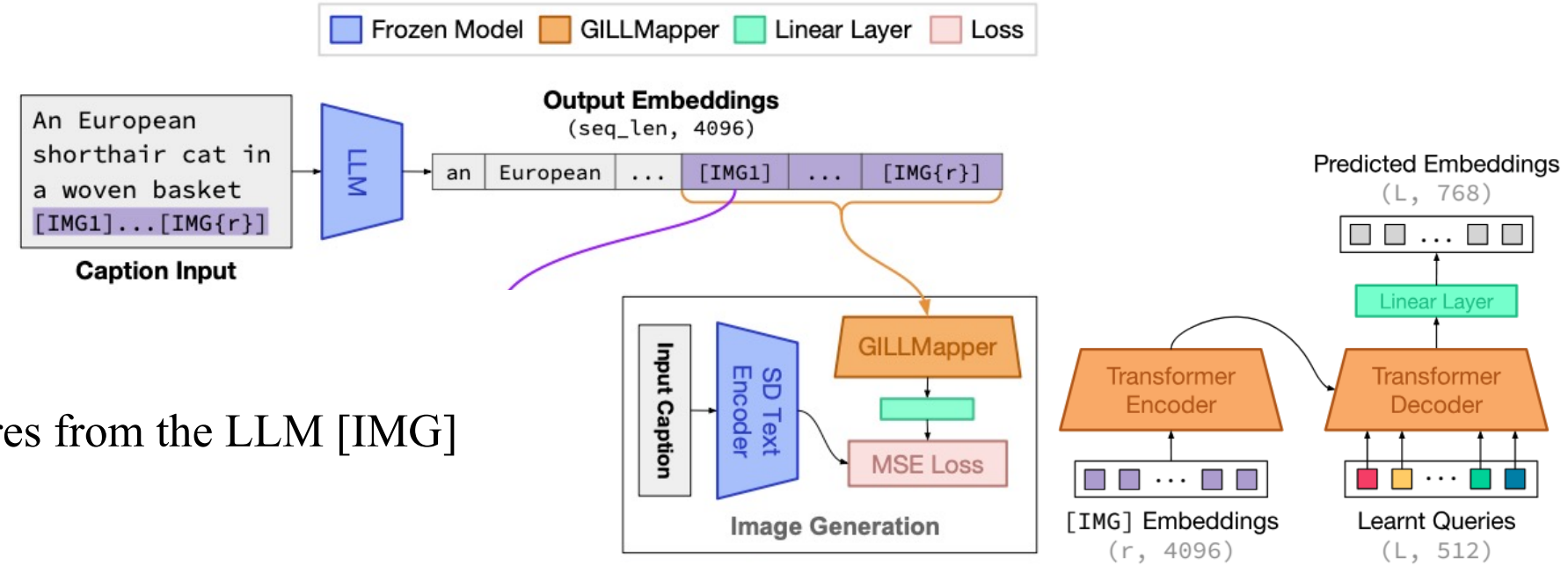


$$Q = W_Q \cdot X$$
$$\Rightarrow Q = W_Q \cdot X + \underbrace{A_Q \cdot (B_Q \cdot X)}_{\text{LoRA}}$$



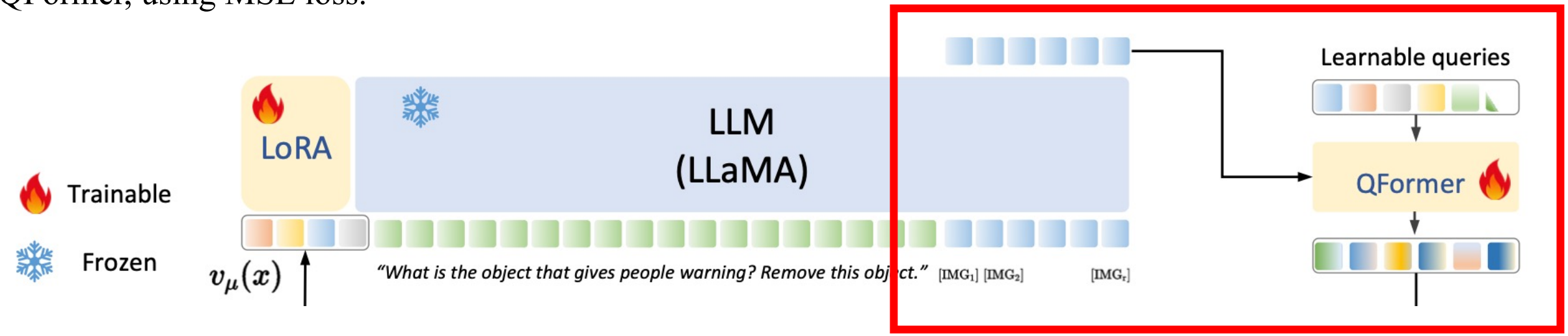
LoRA: Low-rank adaptation of large language models. *ICLR 2022*

QFormer

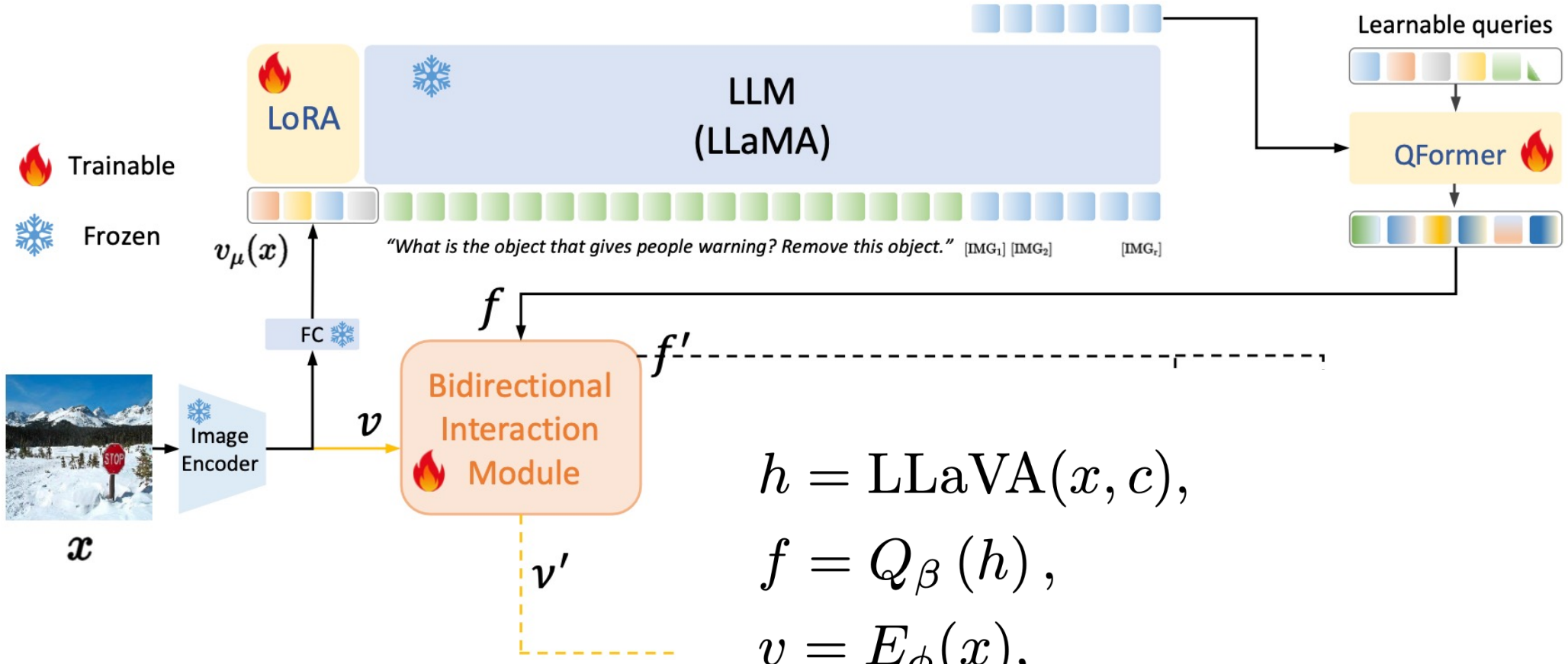


- Extract sequences of L features from the LLM [IMG] hidden states

- MLLM is aligned with the CLIP text encoder using the QFormer, using MSE loss.

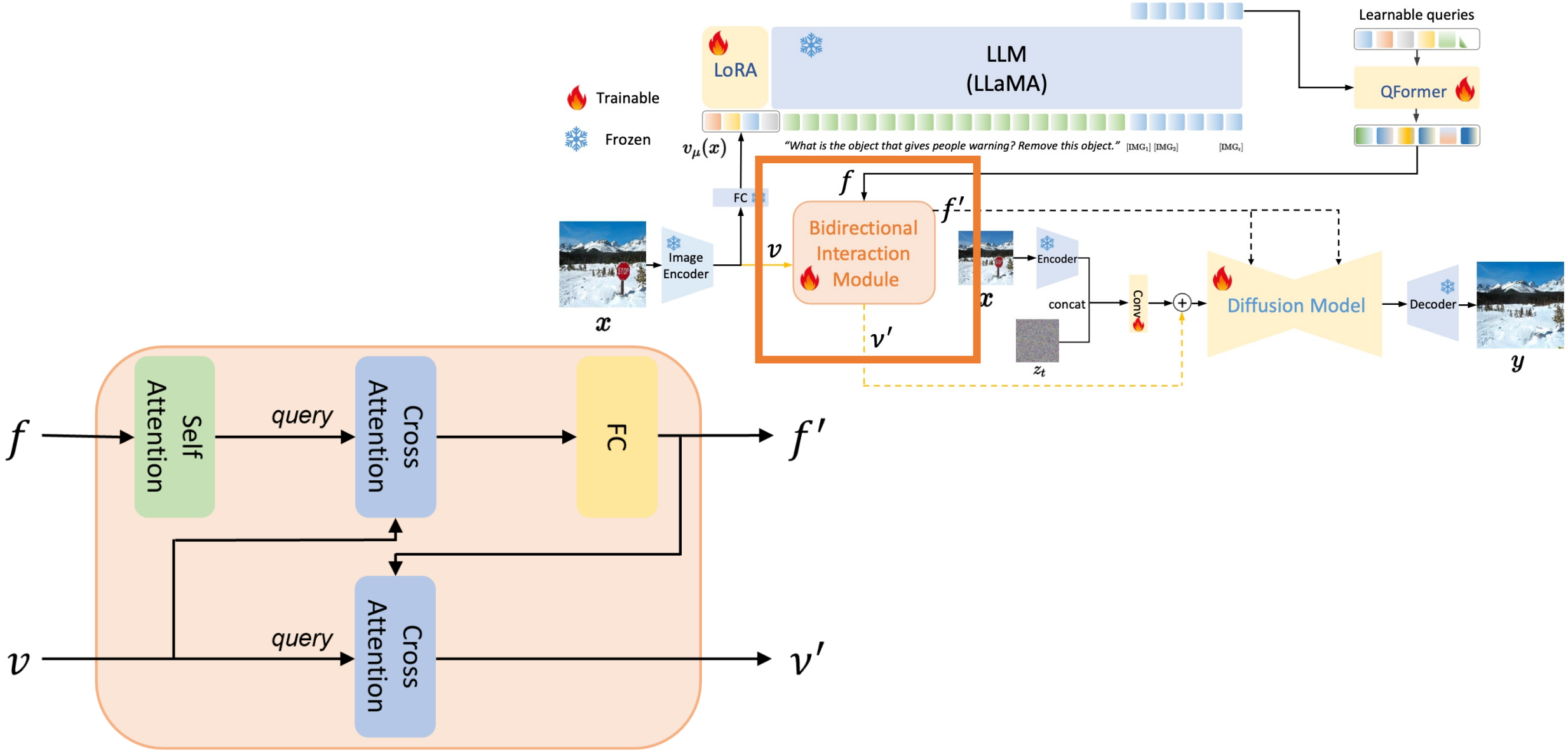


Representation

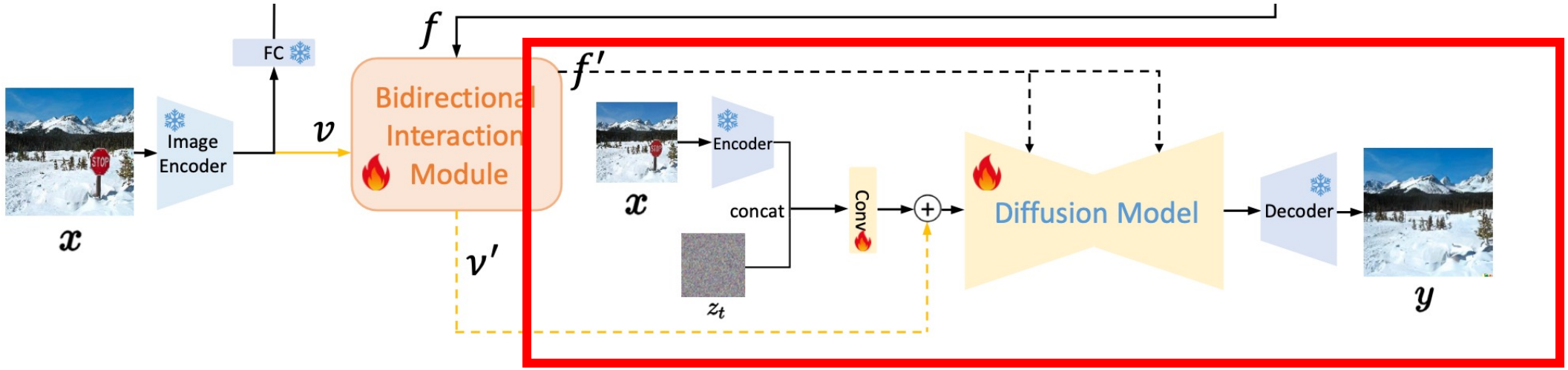


$$
\begin{aligned}
h &= \mathrm{LLaVA}(x, c), \\
f &= Q_\beta\left(h\right), \\
v &= E_\phi(x), \\
f', v' &= \mathrm{BIM}(f, v)
\end{aligned}
\tag{3}
$$

Bidirectional Interaction Module (BIM)



Diffusion Model

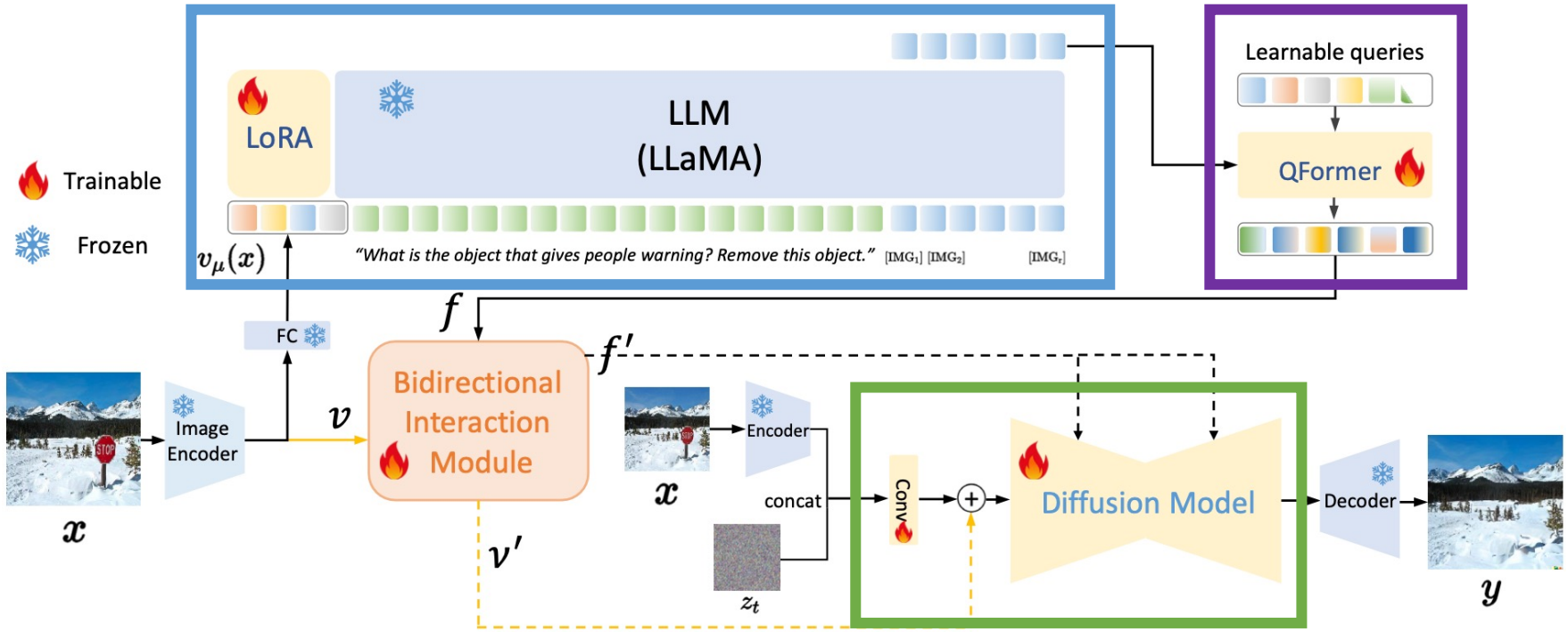


$$L_{\text{diffusion}} = \mathbb{E}_{\mathcal{E}(y),\mathcal{E}(x),c_T,\epsilon\sim\mathcal{N}(0,1),t}[\|\epsilon - \epsilon_\delta(t, \text{concat}[z_t, \mathcal{E}(x)] + v', f'))\|_2^2 \quad (4)$$

Dataset Utilization Strategy

- Limited exposure to editing data that requires reasoning abilities
  - **synthesize data** (original image + Instruction + synthetic target image)
    - **complex understanding** scenarios
      - multiple objects but modifies the specific object based on various attributes (i.e., location, color, relative size, and in or outside the mirror).
    - **reasoning** scenarios
      - need world knowledge to identify the specific object
- UNet in the diffusion model lacks an understanding of perception and concept
  - incorporate the **segmentation data**
- Collect an evaluation dataset, Reason-Edit consisting of 219 image-text pairs

Implementation Details



- The first stage training objectives are the combination of the **MSE loss** between the output of LLaVA and CLIP text encoder, and the **language model loss**.
- In the second stage, the loss function is composed of two parts: **the language model loss** and the **diffusion loss**.

Outline

- Introduction
- Related Work
- Framework
- Method
- Experiment
- Conclusion

Quantitative comparison

Methods	Understanding Scenarios					Reasoning Scenarios				
	PSNR(dB)↑	SSIM↑	LPIPS↓	CLIP Score↑	**Ins-align↑**	PSNR(dB)	SSIM	LPIPS	CLIP Score	**Ins-align↑**
InstructPix2Pix	21.576	0.721	0.089	22.762	0.537	24.234	0.707	0.083	19.413	0.344
MagicBrush	18.120	0.68	0.143	22.620	0.290	22.101	0.694	0.113	19.755	0.283
InstructDiffusion	23.258	0.743	0.067	23.080	0.697	21.453	0.666	0.117	19.523	0.483
SmartEdit-7B	22.049	0.731	0.087	23.611	**0.712**	25.258	0.742	0.055	20.950	**0.789**
SmartEdit-13B	23.596	0.751	0.068	23.536	**0.771**	25.757	0.747	0.051	20.777	**0.817**

- Adopt three metrics for the **background area**: PSNR, SSIM, and LPIPS.
- For the **foreground area**, we calculate the CLIP Score.
- **Human evaluate** the results on Reason-Edit, which is Instruction-Alignment (Ins-align).

"What is the tool that is used to cut fruits. Remove this tool."

"Please replace food contains most vitamin with an orange."

"Please remove the object that can be used to have meals."

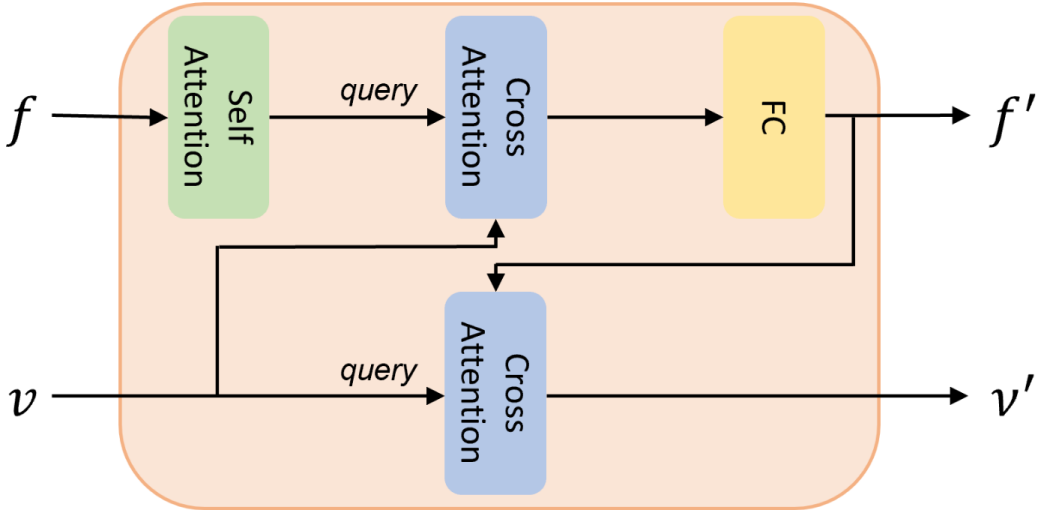
"Change the middle panda to a cat"

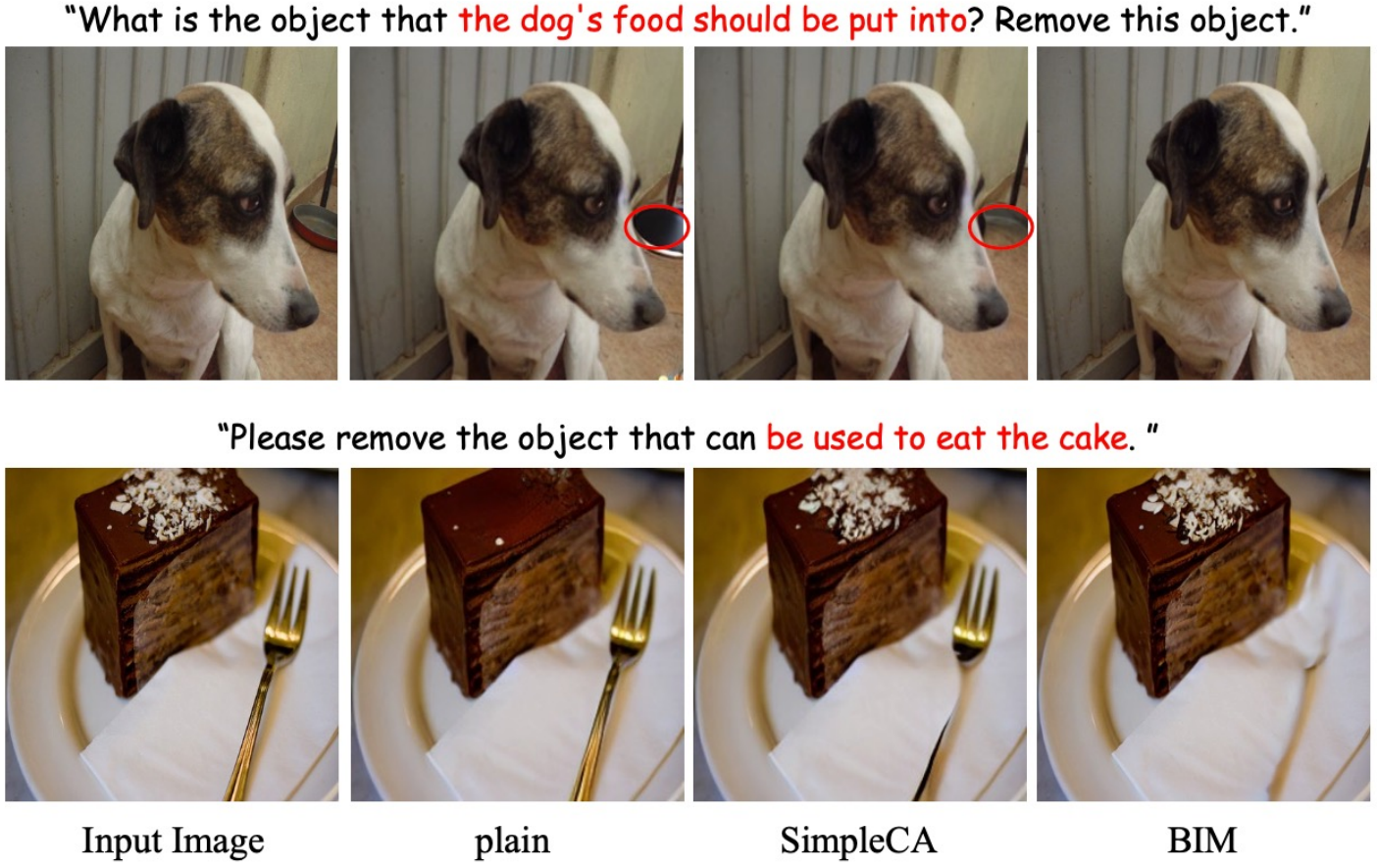
"Change the cat in mirror to a tiger."

Input Image | InstructPix2Pix | MagicBrush | InstructDiffusion | SmartEdit-7B | SmartEdit-13B

Ablation Study on BIM

Exp ID	Plain	SimpleCA	BIM	Understanding Scenarios					Reasoning Scenarios				
				PSNR(dB)↑	SSIM↑	LPIPS↓	CLIP Score↑	**Ins-align↑**	PSNR(dB)	SSIM	LPIPS	CLIP Score	**Ins-align↑**
1	✓			20.975	0.713	0.108	23.36	0.695	23.848	0.725	0.074	20.33	0.694
2		✓		19.557	0.692	0.126	23.66	0.692	23.508	0.716	0.081	20.17	0.722
3			✓	22.049	0.731	0.087	23.61	**0.712**	25.258	0.742	0.055	20.95	**0.789**

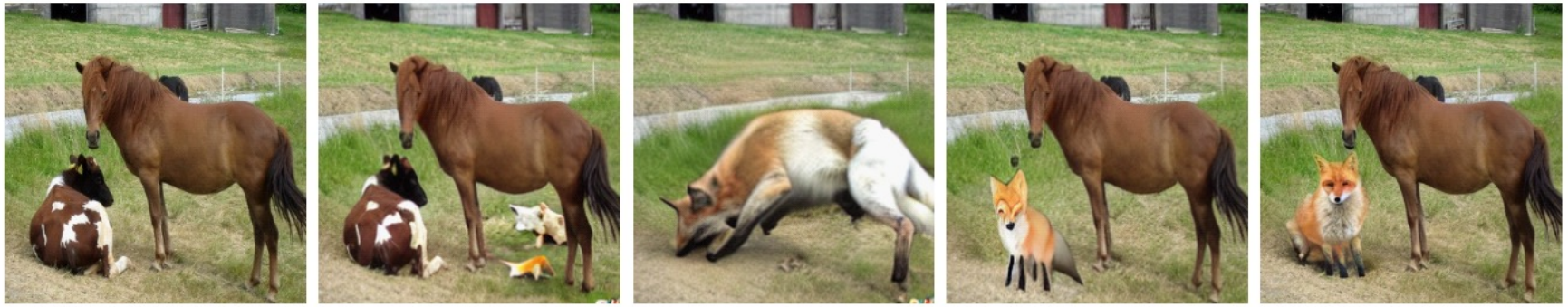




Ablation Study on Dataset Usage

Exp ID	Edit	Segmentation	Synthetic editing dataset	Understanding Scenarios					Reasoning Scenarios				
				PSNR(dB)↑	SSIM↑	LPIPS↓	CLIP Score↑	**Ins-align↑**	PSNR(dB)	SSIM	LPIPS	CLIP Score	**Ins-align↑**
1	✓			17.568	0.664	0.171	22.79	0.201	22.400	0.706	0.102	19.22	0.233
2	✓	✓		18.960	0.690	0.143	22.83	0.361	21.774	0.693	0.116	19.82	0.311
3	✓		✓	19.562	0.702	0.111	22.32	0.440	23.595	0.715	0.079	20.43	0.567
4	✓	✓	✓	22.049	0.731	0.087	23.61	**0.712**	25.258	0.742	0.055	20.95	**0.789**

"Please replace the animal that is lying on the grass with a fox"



"Change the red strawberry to a white pumpkin"

Input Image | Edit | Edit+Seg | Edit+Synthetic Editing | Total

Outline

- Introduction
- Related Work
- Framework
- Method
- Experiment
- Conclusion

Conclusion

- SmartEdit, a novel approach to instruction-based image editing, incorporating the Large Language Models (LLMs) with visual inputs, and it outperforms previous methods on our newly constructed practical dataset, Reason-Edit.

- Introducing the Bidirectional Interaction Module (BIM), it have overcome challenges in complex reasoning scenarios.

- Data utilization strategy, which incorporates perception data and complex instruction editing data, effectively enhances SmartEdit.