Computer Vision - 2025

Week #13. Multi-Modal Data Processing. Part II: Generalization

Lectures by Alexei Kornaev [1,2,3]
Practical sessions by Kirill Yakovlev [2]

[1]AI Institute, Innopolis University (IU), Innopolis
[2]Robotics & CV Master's Program, IU, Innopolis
[3]Dept. of M^2R, Orel State University, Orel
[4]RC for AI, National RC for Oncology, Moscow

April 14, 2025

Agenda

Section 1. Recap: CLIP architecture and loss

CLIP Loss: Core Mechanism

Definition

Align image-text pairs in a shared space using symmetric contrastive loss:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2N} \sum_{i=1}^{N} \left[\log \frac{e^{s_{ii}/\tau}}{\sum_j e^{s_{ij}/\tau}} + \log \frac{e^{s_{ii}/\tau}}{\sum_j e^{s_{ji}/\tau}} \right]$$

where $s_{ij} = \cos\_\text{sim}(I_i, T_j)$ for image and text embeddings, τ is the temperature parameter (learned or fixed) to scale logits.

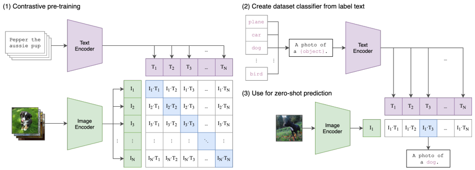

Figure: Contrastive Language-Image Pre-training (CLIP) [Radford et al., 2021]

Pre-test: CLIP Loss Example with a Small Batch (N=2)

Setup

Batch size $N = 2$, temperature $\tau = 0.07$; **image embeddings**: $I_1 = [0.8, 0.6]$, $I_2 = [0.6, 0.8]$; **iext embeddings**: $T_1 = [0.7, 0.7]$, $T_2 = [0.7, -0.7]$; normalized embeddings: $\|I\| = \|T\| = 1$; **similarity Matrix**:

$$S = \begin{bmatrix} I_1 \cdot T_1 & I_1 \cdot T_2 \\ I_2 \cdot T_1 & I_2 \cdot T_2 \end{bmatrix} = \begin{bmatrix} 0.98 & 0.14 \\ 0.98 & -0.14 \end{bmatrix}.$$

Step 1: Image $\to$ Text Loss (for I_1)

$$\text{Softmax}_\tau(S_{I_1}) = \frac{e^{X.XX/X.XX}}{e^{X.XX/X.XX} + e^{X.XX/X.XX}} \approx 1.0,$$

$$\mathcal{L}_{CE}(I_1) = -\log(1.0) \approx 0.$$

Step 2: Text$\to$ Image Loss (for T_2)

$$\text{Softmax}_\tau(S_{T_2}) = \frac{e^{X.XX/X.XX}}{e^{X.XX/X.XX} + e^{X.XX/X.XX}} \approx 0.018,$$

$$\mathcal{L}_{CE}(T_2) = -\log(0.018) \approx 4.0.$$

Total Loss

$$\mathcal{L}_{\text{CLIP}} = \frac{1}{2 \times 2}(0 + 4.0 + \dots) \quad \text{(Sum over all pairs)}.$$

INNOPOLIS UNIVERSITY

Pre-test: CLIP Loss Example with a Small Batch (N=2)

Setup

Batch size $N = 2$, temperature $\tau = 0.07$; **image embeddings**: $I_1 = [0.8, 0.6]$, $I_2 = [0.6, 0.8]$; **iext embeddings**: $T_1 = [0.7, 0.7]$, $T_2 = [0.7, -0.7]$; normalized embeddings: $\|I\| = \|T\| = 1$; **similarity Matrix**:

$$S = \begin{bmatrix} I_1 \cdot T_1 & I_1 \cdot T_2 \\ I_2 \cdot T_1 & I_2 \cdot T_2 \end{bmatrix} = \begin{bmatrix} 0.98 & 0.14 \\ 0.98 & -0.14 \end{bmatrix}.$$

Step 1: Image $\rightarrow$ Text Loss (for I_1)

$$\text{Softmax}_\tau(S_{I_1}) = \frac{e^{0.98/0.07}}{e^{0.98/0.07} + e^{0.14/0.07}} = \frac{e^{14}}{e^{14} + e^{2}} \approx 1.0,$$

$$\mathcal{L}_{CE}(I_1) = -\log(1.0) \approx 0.$$

Step 2: Text$\rightarrow$ Image Loss (for T_2)

$$\text{Softmax}_\tau(S_{T_2}) = \frac{e^{-0.14/0.07}}{e^{0.14/0.07} + e^{-0.14/0.07}} = \frac{e^{-2}}{e^{2} + e^{-2}} \approx 0.018,$$

$$\mathcal{L}_{CE}(T_2) = -\log(0.018) \approx 4.0.$$

Total Loss

$$\mathcal{L}_{\text{CLIP}} = \frac{1}{2 \times 2}(0 + 4.0 + \dots) \quad \text{(Sum over all pairs)}.$$

Section 2. Current state of VLMs

Section 3. Optimization of VLMs

Section 4. Fine-Tuning of LMs

INNOPOLIS UNIVERSITY

Section 5. Prospects

Section 6. VLM + Control System = VLAM

RT-2: Vision-Language-Action Model

Core Mechanism

Unifies vision, language, and action via tokenization: $\text{Action} = \text{Decode}\left(\text{Transformer}([\underbrace{\mathcal{V}(I)}_{\text{Visual Tokens}};\underbrace{\mathcal{T}(C)}_{\text{Language Tokens}}])\right)$

where: $\mathcal{V}$: vision tokenizer (ViT + Action Quantizer); $\mathcal{T}$: language tokenizer (PaLI-style); actions discretized as $\langle \text{cmd}, x, y, z, \theta \rangle$ tokens.

Key Innovations

- **Action Chunking**: Predicts action sequences autoregressively
- **Cross-Modal Attention**: $\text{Attention}(Q_{\text{action}}, K_{\text{vision + lang}}, V_{\text{vision + lang}})$
- **Chain-of-Thought**: "Plan $\rightarrow$ Verify $\rightarrow$ Execute" token prediction

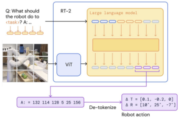

Figure: RT-2's unified architecture [Brohan et al., 2023]

Hands-on Coding with CLIP models (again)

CLIP + CLIPSeg = prerequisite for Action

- Get an Image (scene) + text (instruction from a human to a robot)
- Define a set of discrete robot skills (actions) and scene objects, and distractors
- use CLIPSeg for object segmentation and position detection
- use CLIP for skill prediction (VLAM Concept)

The code is available via the link #1.

Core Concepts in Multimodal Learning

Key Technical Challenges

- **Embedding Alignment**: Map modalities to a shared space (e.g., CLIP's image/text encoders).
- **Cross-Modal Attention**: Dynamically fuse modalities (e.g., Flamingo's Perceiver Resampler).
- **Scaling Laws**: Training with massive datasets (LAION-5B, RT-1).

Contrastive Learning Formulation

$$\mathcal{L}_{\text{contrast}} = -\log \frac{e^{s(I,T)/\tau}}{\sum_{j=1}^{N} e^{s(I,T_j)/\tau}}$$

- $s(I, T)$: Cosine similarity between image I and text T.
- τ: Temperature parameter (learned in CLIP).

Cross-Modal Attention

$$\text{Attention}(Q_{\text{text}}, K_{\text{image}}, V_{\text{image}}) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V$$

Modern VLMs and Applications

Architectures

- Flamingo [Alayrac, 2022]: Processes interleaved images/text for few-shot learning.
- LLaVA [Liu, 2023]: Connects vision encoder to LLM via projection layers.
- BLIP-2 [Li, 2023]: Q-Former bridges frozen encoders (ViT + LLM).

Robotics Applications

- **PALM-E** [Driess, 2023]: Embodied LLM for planning with vision-language-action.
- RT-2: VLMs for robotic control ("pick up the banana").
- **Instruction Following**: Grounding language commands to sensorimotor actions.

Modality	Robot Input	Embedding Technique
Vision	Camera frames	ViT/ResNet
Language	Commands	BERT/GPT
Actions	Joint angles	MLP

Table: Multimodal Inputs in Robotics

Section 7. Outcomes

Outcomes

This week's lecture on Multimodal Data Processing introduces foundational concepts in vision-language alignment for robotic systems. By the end of this session, students will be able to:

1. Explain contrastive learning principles and cross-modal attention mechanisms in Vision-Language Models (VLMs).
2. Implement zero-shot inference using CLIP for robotic object recognition and scene understanding.
3. Critically evaluate architectural choices in modern VLMs (e.g., Flamingo [Alayrac, 2022], LLaVA [Liu, 2023]).

Key Takeaway: Multimodal alignment bridges perception (vision) and reasoning (language), forming the foundation for embodied AI systems like PALM-E [Driess, 2023] in robotics.

VLAM: Vision-Language-Action Models

Key Components

- Multimodal encoder (vision + language)
- Policy network for action generation
- Integration with reinforcement learning [Driess, 2023]

Policy Gradient Theorem

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{\pi_\theta} \left[\nabla_\theta \log \pi_\theta(a|s) \cdot A(s, a) \right]$$

Component	Implementation
Multimodal Encoder	Transformer Fusion
Policy Network	MLP/Transformer Decoder
Action Space	Continuous (RL) / Discrete (IL)

Table: VLAM Architecture Components

LLM Training: From Scratch vs Pretrained

Decision Factors

- **Pretrained**: 99% of use cases (low-resource adaptation)
- **From Scratch**: Specialized domains, novel architectures

Parameter-Efficient Fine-Tuning

- LoRA: $\Delta W = BA$ where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$
- QLoRA: 4-bit quantization + LoRA

Metric	From Scratch	Pretrained
Data Needs	1B+ tokens	1k-100k tokens
Compute Cost	\$100k+	\$100-\$1k
Training Time	Weeks	Hours

Table: Training Strategy Comparison

Section 8. Contrastive Language-Image Pre-training (CLIP)

CLIP Loss: Core Mechanism

Definition

Align image-text pairs in a shared space using symmetric contrastive loss:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2N}\sum_{i=1}^{N}\left[\log\frac{e^{s_{ii}}/\tau}{\sum_j e^{s_{ij}}/\tau} + \log\frac{e^{s_{ii}}/\tau}{\sum_j e^{s_{ji}}/\tau}\right]$$

where $s_{ij} = \cos\_\text{sim}(I_i, T_j)$ for image and text embeddings, τ is the temperature parameter (learned or fixed) to scale logits.

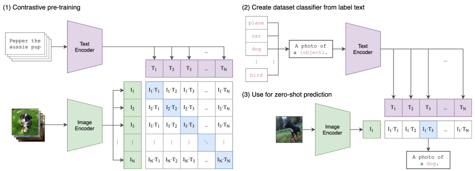

Figure: Contrastive Language-Image Pre-training (CLIP) [Radford et al., 2021]

Section 9. Fine-Tuning of Large Models

Fine-Tuning Strategies for LMs

Comparison of Parameter-Efficient Methods

Method	Added Params	Modifies Forward Pass	Key Advantage
Full FT	100%	$h = (W_0 + \Delta W)x$	Highest accuracy
LoRA	$\sim 0.1\%$	$h = W_0 x + BAx$	Balance of efficiency/performance
Adapter	$\sim 1\%$	$h = W_0 x + W_2(\sigma(W_1 x))$	Modular
Prefix Tuning	$\sim 0.5\%$	$[P; x] \rightarrow$ Attention	No backbone changes
BitFit	$\sim 0.01\%$	$h = W_0 x + b$	Only biases updated

Mathematical Forms

- **Adapter**: $W_2 \in \mathbb{R}^{d \times r}$, $W_1 \in \mathbb{R}^{r \times d}$
- **Prefix Tuning**: $P \in \mathbb{R}^{l \times d}$ (prepended tokens)
- **BitFit**: $b \in \mathbb{R}^{d}$ (bias terms only)

When to Use LoRA?

- Need high parameter efficiency ($r \leq 64$)
- Preserve original model architecture
- Balance between compute and accuracy

LoRA: Low-Rank Adaptation [Hu et al., 2021]

Key Mathematical Formulation

For a pretrained weight matrix $W_0 \in \mathbb{R}^{d\times k}$:

$$W = W_0 + \underbrace{BA}_{\text{Low-rank update}} \quad \begin{cases} B \in \mathbb{R}^{d\times r} \\ A \in \mathbb{R}^{r\times k} \\ r \ll \min(d, k) \end{cases}$$

Example: 1024x1024 Layer with Rank=8

- Original params: $1024 \times 1024 = 1{,}048{,}576$
- LoRA params: $8 \times (1024 + 1024) = 16{,}384$
- Reduction: $\frac{16{,}384}{1{,}048{,}576} \approx 1.56\%$

INNOPOLIS UNIVERSITY

LoRA in Action

Forward Pass Computation

For input $x \in \mathbb{R}^k$:

$$h = W_0 x + \underbrace{B(Ax)}_{\text{Rank-constrained update}}$$

Gradient Flow

- Frozen weights: $\nabla_{W_0} \mathcal{L} = 0$
- Adaptor gradients:

$$\nabla_B \mathcal{L} = (\nabla_h \mathcal{L}) x^\top A^\top \quad \nabla_A \mathcal{L} = B^\top (\nabla_h \mathcal{L}) x^\top$$

Why This Works

- Preserves pretrained knowledge (W_0 fixed)
- Efficient training (only update B, A)
- Low-rank bottleneck prevents overfitting

Blog reading

Vision Language Action Models (VLA) Overview: LeRobot Policies Demo

The advent of Generative AI, has fundamentally transformed robotic intelligence, enabling significant strides in how advanced humanoid robots âperceive, reason and actâ in the physical world. This huge progress is primarily attributed in terms of decision making, thanks to LLM and VLMs generalization due to their large scale pre-training. Instead of relying on traditional complex policies which has to be carefully handcrafted for individual low level tasks for fine grained actions, VLA allows robotic control combining vision and language knowledge for better reasoning.

Paper reading

RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

We study how vision-language models trained on Internet-scale data can be incorporated directly into end-to-end robotic control to boost generalization and enable emergent semantic reasoning. Our goal is to enable a single end-to-end trained model to both learn to map robot observations to actions and enjoy the benefits of large-scale pretraining on language and vision-language data from the web. To this end, we propose to co-fine-tune state-of-the-art vision-language models on both robotic trajectory data and Internet-scale vision-language tasks, such as visual question answering. In contrast to other approaches, we propose a simple, general recipe to achieve this goal: in order to fit both natural language responses and robotic actions into the same format, we express the actions as text tokens and incorporate them directly into the training set of the model in the same way as natural language tokens. We refer to such category of models as vision-language-action models (VLA) and instantiate an example of such a model, which we call RT-2. Our extensive evaluation (6k evaluation trials) shows that our approach leads to performant robotic policies and enables RT-2 to obtain a range of emergent capabilities from Internet-scale training. This includes significantly improved generalization to novel objects, the ability to interpret commands not present in the robot training data (such as placing an object onto a particular number or icon), and the ability to perform rudimentary reasoning in response to user commands (such as picking up the smallest or largest object, or the one closest to another object). We further show that incorporating chain of thought reasoning allows RT-2 to perform multi-stage semantic reasoning, for example figuring out which object to pick up for use as an improvised hammer (a rock), or which type of drink is best suited for someone who is tired (an energy drink) [Brohan et al., 2023].

Paper reading

Vision-Language Models for Vision Tasks: A Survey

Most visual recognition studies rely heavily on crowd-labelled data in deep neural networks (DNNs) training, and they usually train a DNN for each single visual recognition task, leading to a laborious and time-consuming visual recognition paradigm. To address the two challenges, Vision-Language Models (VLMs) have been intensively investigated recently, which learns rich vision-language correlation from web-scale image-text pairs that are almost infinitely available on the Internet and enables zero-shot predictions on various visual recognition tasks with a single VLM. This paper provides a systematic review of visual language models for various visual recognition tasks, including: (1) the background that introduces the development of visual recognition paradigms; (2) the foundations of VLM that summarize the widely-adopted network architectures, pre-training objectives, and downstream tasks; (3) the widely-adopted datasets in VLM pre-training and evaluations; (4) the review and categorization of existing VLM pre-training methods, VLM transfer learning methods, and VLM knowledge distillation methods; (5) the benchmarking, analysis and discussion of the reviewed methods; (6) several research challenges and potential research directions that could be pursued in the future VLM studies for visual recognition. [Zhang et al., 2024].

Section 10. Other Types of Multimodalities

Other Types of Multimodalities

The Five Senses Analogy

Sense	Data Modality	ML Example
Vision	Images/Video	CNNs, ViTs
Auditory	Audio/Waveforms	Spectrogram Transformers
Tactile	Pressure/Texture	Tactile Sensors in Robotics
Olfactory	Chemical Sensors	e-Nose Gas Detection
Gustatory	Molecular Data	Flavor Prediction Models

Emerging Sensor Fusion

- LiDAR+RGB: Autonomous vehicles
- IMU+Vision: Human pose estimation
- Spectrograms+Text: Audio captioning

Section 11. Meta-Learning

Multimodal Optimization Challenges

Physics-Informed Neural Networks (PINNs)

$$\mathcal{L}_{\text{PINN}} = \underbrace{\lambda_d \|u_\theta(x_i) - u_i\|^2}_{\text{Data Loss}} + \underbrace{\lambda_p \|\mathcal{N}[u_\theta](x_j)\|^2}_{\text{Physics Loss}} + \underbrace{\lambda_r \|\theta\|^2}_{\text{Regularization}}$$

- Multi-objective: Data fitting + PDE residuals [Raissi et al., 2017]
- Loss landscape modality gaps cause training instabilities

Multi-Task Tradeoffs

- Pareto optimality in joint losses
- Gradient conflict quantification:

$$\cos(\nabla_\theta \mathcal{L}_i, \nabla_\theta \mathcal{L}_j) < 0$$

- Solution: Uncertainty weighting [Kendall et al., 2018]

Bibliography

Jean-Baptiste et al. Alayrac. Flamingo: a visual language model for few-shot learning. **NeurIPS**, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

Danny et al. Driess. Palm-e: An embodied multimodal language model. **ICML**, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In **Proceedings of the IEEE conference on computer vision and pattern recognition**, pages 7482–7491, 2018.

Junnan et al. Li. Blip-2: Bootstrapping vision-language pre-training with frozen image encoders and llms. **arXiv:2301.12597**, 2023.

Haotian et al. Liu. Visual instruction tuning. **NeurIPS**, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017. URL https://arxiv.org/abs/1711.10561.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey, 2024. URL https://arxiv.org/abs/2304.00685.