# Spatial Gesture Semantics

## 4. AI and Gesture Detection

Andy Lücking Alexander Henlein

Goethe University Frankfurt

July 28–August 01, 2025

# Recap

# Yesterday's lecture

- World-to-word direction of fit
- Classifier-based (computational) semantics
- Extemplification (extended exemplification)
- Informational evaluation heuristic

# Today's lecture

- ML Primer: learning paradigms
- Building models like ChatGPT
- Multimodal foundations
- Gesture-detection pipeline
- Hands-on live demo
- Outlook & open questions

# Introduction: Machine Learning, AI and Multimodality

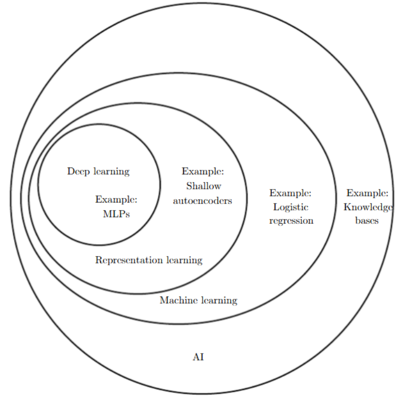
# What is Machine Learning / AI?[1]

## Artificial Intelligence (AI)

- Umbrella term for techniques that enable machines to perform tasks we regard as "intelligent" (reasoning, perception, planning, language).

## Machine Learning (ML)

- Sub-field of AI: systems learn patterns from data instead of relying on hand-crafted rules.
- Core ingredients: large data → model → loss → optimisation *rightarrow* evaluation.

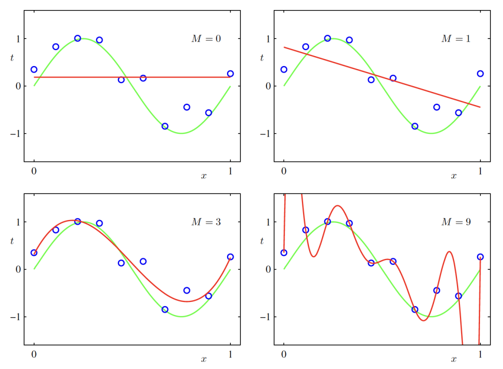



[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio (2016). Deep learning. Vol. 1. MIT press Cambridge

# What is Machine Learning (ML)?[3]

- Instead of writing explicit rules, ML finds patterns in data.
- At its core: ML = fitting a function to data.
- Useful when the rules are too complex, fuzzy, or unknown - e.g., how gestures vary across speakers and contexts.

[2]

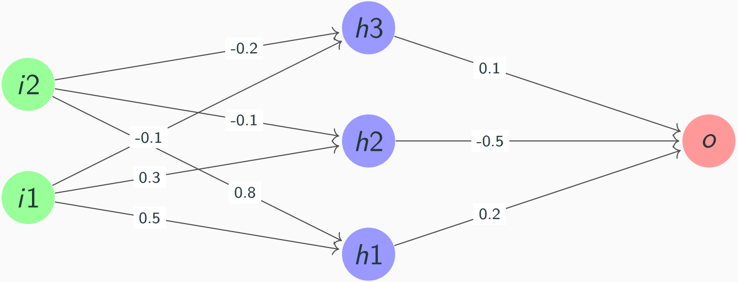



[2] https://amitrajan012.github.io/post/pattern-recognition-chapter-1-introduction_1/

[3] C. M. Bishop and N. M. Nasrabadi (2006). Pattern recognition and machine learning. Vol. 4. Springer

# How are Neural Networks trained?

Is the weather suitable for picnics?



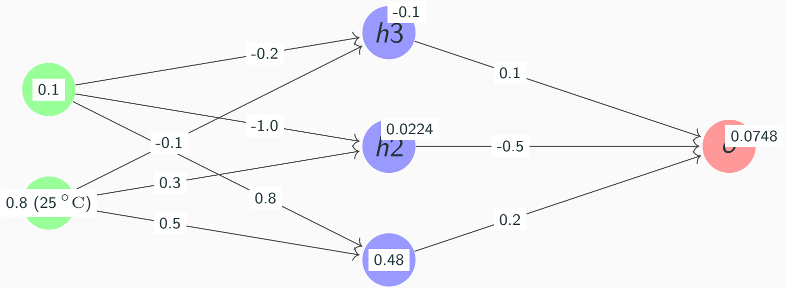

[4]

i1: temperature
i2: risk of rain
o: picnic score

[4] Template: https://tikz.net/regular-vs-bayes-nn/

# How are Neural Networks trained? - Example Input

Is the weather suitable for picnics?



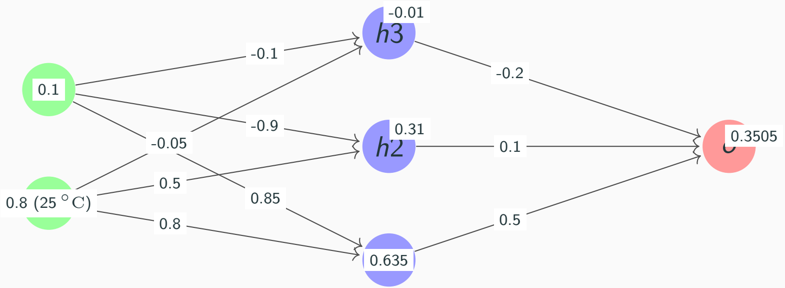

i1: temperature
i2: risk of rain
o: picnic score

[5]

[5] Template: `https://tikz.net/regular-vs-bayes-nn/`

# How are Neural Networks trained? - Backpropagation

Is the weather suitable for picnics?




i1: temperature
i2: risk of rain
o: picnic score

6

[6] Template: `https://tikz.net/regular-vs-bayes-nn/`

# Learning Paradigms in ML[7]

- Supervised → Learn to predict labels.
- Unsupervised → Find structure or clusters.
- Self-Supervised → Predict part of data from other parts.
- Semi-Supervised → Leverage a few labels with lots of unlabeled data.
- Reinforcement → Learn good decisions over time.

[7] C. M. Bishop and N. M. Nasrabadi (2006). Pattern recognition and machine learning. Vol. 4. Springer; V. Rani et al. (2023). "Self-supervised learning: A succinct review". In: Archives of Computational Methods in Engineering 30, 2761–2775

# What ML Can (and Can't) Do for Us[8]

## Can do

- **Detect** classes even from noisy data.
- **Cluster** and quantify variation.
- **Learn** useful representations from raw data.
- **Support** large-scale studies of form and use.

## Can't do

- **Understand** meaning on its own.
- **Replace** semantic theory or manual insight.
- **Handle** open-ended or subtle communicative functions (yet).
- **Guarantee** fairness, explainability, or trustworthiness out of the box.

[8] E. M. Bender and A. Koller (2020). "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data". In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics, 5185–5198; G. Marcus and E. Davis (2019). Rebooting AI: Building artificial intelligence we can trust. Vintage

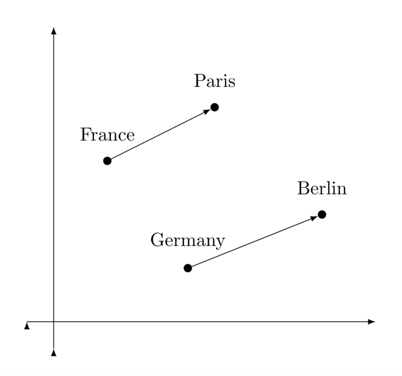
# Embeddings: Representing Data as Vectors[9]

## What are Embeddings?

- Continuous vector representation of discrete items (words, tokens, images).
- Geometric proximity $\Leftrightarrow$ semantic similarity.

## Why Important for LLMs

- Input tokens mapped to embeddings learned during training.
- Enable efficient dot-products, generalisation, and transfer across tasks.

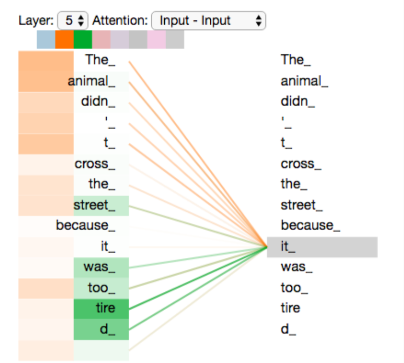



(Wikimedia Foundation, Inc. Original up- loader was Cbarr (WMF), CC BY-SA 3.0, File:RobGrindes-shrug-143px.png)

[9] T. Mikolov et al. (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: Advances in Neural Information Processing Systems; T. Mikolov, K. Chen, G. Corrado, and J. Dean (2013). "Efficient Estimation of Word Representations in Vector Space". In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings

# Attention Mechanism[10]

- Dynamically weights input elements based on relevance.
- **Self-attention**: queries, keys, values from same sequence.
- **Multi-head**: parallel views capture diverse relations.
- Powers the Transformer architecture and modern LLMs.




https://jalammar.github.io/illustrated-transformer/ (CC BY-NC-SA 4.0)

10 A. Vaswani et al. (2017). "Attention is All you Need". In: Advances in Neural Information Processing Systems

# (Multimodal) Large Language Models

# How to train my own ChatGPT[12]

1. Data Collection & Preprocessing → Clean, filter, deduplicate, normalize, tokenize.
2. (Self-supervised) Pretraining → Next-token prediction.
3. Post-Training → Reinforcement Learning from Human Feedback (RLHF)[11].
4. Evaluation → For performance, safety, bias, hallucination.
5. Deployment & Iteration → Frequent monitoring and updated.

[11] L. Ouyang et al. (2022). "Training language models to follow instructions with human feedback". In: Proc. of the 36th International Conference on Neural Information Processing Systems

[12] OpenAI et al. (2024). GPT-4 Technical Report. arXiv: 2303.08774 [cs.CL]

# Step 1: Data Collection & Preprocessing[13]

**Goal:**
Prepare high-quality, diverse input for training.

**Sources:**

- Web text
- Books, Wikipedia
- Forums, code repositories
- Internal/proprietary data

→ **Filtering**: remove low-quality, toxic, or irrelevant content.

→ **Deduplication**: avoid overfitting to repeated content.

→ **Normalization**: standardize text (e.g., lowercase, punctuation).

→ **Tokenization**: convert text into input tokens.

→ **Balancing**: ensure coverage across domains (e.g., code vs. dialogue).

[13] L. Gao et al. (2020). The Pile: An 800GB Dataset of Diverse Text for Language Modeling. arXiv: 2101.00027 [cs.CL]

# Excursus: What is Tokenization in the Context of LLMs?[14]

**Goal:**
Convert raw text into units the model can understand.

**Why not characters or words?**

- Characters: too granular, inefficient
- Words: ambiguous, too many
- Tokens: trade-off

→ Use subword units (e.g. "play", "#ing"; "un", "#believable").

→ Based on algorithms like **Byte-Pair Encoding (BPE)** or **Unigram LM**.

→ Allows handling of rare and unknown words.

→ Example: `"I really enjoyed my time in Bochum."` → `["I", "really", "enjoy", "#ed", "my", "time", "in", "Boch", "#um", "."]`

[14] `https://huggingface.co/docs/transformers/tokenizer_summary`

# Step 2: (Self-supervised) Pretraining

**Goal:**
Teach the model general language understanding.

**Method:**

- Predict next token
- No human labels needed
- Very large dataset

→ Objective: $\mathtt{P(token}_t \,|\, \mathtt{token}_{1..t-1}\mathtt{)}$

→ Transformer architecture (e.g. decoder-only).

→ Trained on trillions of tokens.

→ Requires massive compute (TPUs, GPUs).

→ Learns grammar, facts, reasoning, coding patterns.

# Step 3: Post-Training (Alignment)

**Goal:**
Make the model helpful, safe, and aligned with human values.

**Steps:**

- Supervised fine-tuning (SFT)
- RLHF (Reinforcement Learning from Human Feedback)[15]

→ Human-written prompt-response pairs.

→ Rank model outputs → train a reward model.

→ Fine-tune the base model using Reinforcement Learning.

→ Encourages helpful and non-toxic responses.

→ Aligns model with human intent.

[15] Y. Bai et al. (2022). Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv: 2204.05862 [cs.CL]

# Step 4: Evaluation[16]

**Goal:**
Assess model quality, safety, and behavior before release.

**Types:**

- Quantitative tests
- Human evaluations
- Red-teaming

→ Benchmarking (MMLU, HellaSwag, etc.).
→ Prompt diversity testing and edge cases.
→ Detect bias, toxicity, hallucinations.
→ Internal and external safety audits.
→ Analyze model confidence and calibration.

[16] Y. Chang et al. (2024). "A survey on evaluation of large language models". In: ACM transactions on intelligent systems and technology 15, 1–45

# Step 5: Deployment & Iteration

**Goal:**
Safely deploy the model and keep improving it through usage.

**Cycle:**

- Launch → Monitor → Improve
- Continuous feedback loop

→ Model exposed via APIs, apps (e.g. ChatGPT).

→ Usage analytics + human feedback collected.

→ Updates: bugfixes, safety patches, new features.

→ Ongoing fine-tuning and A/B testing.

→ Data pipeline refinement based on usage.

# Step 6a: Image Encoder[17]

**Goal:**
Convert an image into a vector representation (embeddings).

**Common Encoders:**

- CLIP (ViT)
- ResNet
- SigLIP
- Vision Transformer (ViT)

→ **Input**: raw image pixels

→ **Output**: sequence of image embeddings (like tokens)

→ Pretrained on image-text pairs (e.g., from web)

→ Encoded images are fed into the language model as part of the prompt

→ Can capture visual objects, layout, and spatial info

[17] A. Radford et al. (2021). "Learning Transferable Visual Models From Natural Language Supervision". In: CoRR abs/2103.00020. arXiv: 2103.00020

# Step 6b: Aligning Modalities[18]

**Goal:**
Bridge the gap between visual and textual representations.

**Why align?**

- Image + text are from different distributions
- Need unified input for Transformer

→ **Projection Layer:** maps image embeddings to LLM token space

→ **Concatenation:** image embeddings placed before or between text tokens

→ **Joint Training:** learn to ground vision in language tasks

→ Enables multimodal reasoning, captioning, and VQA

[18] H. Liu, C. Li, Q. Wu, and Y. J. Lee (2023). "Visual instruction tuning". In: Advances in neural information processing systems 36, 34892–34916

# Step 6c: Multimodal Training Tasks[19]

**Goal:**
Teach the model to understand and reason over image-text pairs.

**Common Task Types:**

- Image captioning
- Visual question answering (VQA)
- OCR + scene text recognition
- Referring expression resolution

→ **Image → Text**: Generate captions or summaries

→ **Image + Text → Text**: Answer questions about the image

→ Use instruction-following prompts: `"Describe this image."`, `"Where is the cat?"`

→ Supervised training followed by instruction tuning

→ Important for grounding language in perception

[19] Y. Zhang et al. (2024). LLaVAR: Enhanced Visual Instruction Tuning for Text-Rich Image Understanding. arXiv: 2306.17107 [cs.CV]

# Excursus: Let's take a look at this type of data set.

http://captions.christoph-schuhmann.de/eval_laion/eval.html
https://laion.ai/projects/

# Scaling and Advanced Concepts

**Goal:**
Push efficiency, capability, and scalability of large models.

**Key Concepts:**

- Mixture of Experts (MoE)
- Sparse Attention
- Retrieval-Augmented Generation (RAG)
- Instruction Tuning

→ **MoE**: Only a subset of model components (experts) is activated per input. Greatly reduces compute cost while increasing parameter count.

→ **Sparse Attention**: Improves efficiency in very long-context models. Models learn to focus selectively.

→ **RAG**: Combines LLMs with external search/indexes. Augments generation with real-world knowledge.

→ **Instruction Tuning**: Further trains models to follow natural language commands more reliably. Key to usability.

# Reducing LLMs in Size[20]

**Why shrink models?**

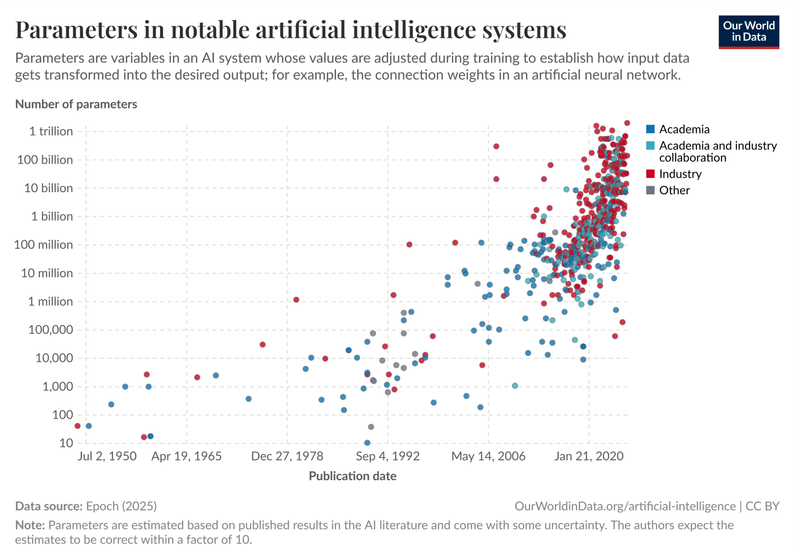
- Lower inference latency and energy cost.
- Fit on-device / edge hardware.
- Enable private, offline use.
- Reduce carbon footprint.

**Main techniques**

- **Quantization**: 8-bit/4-bit weights.
- **Pruning**: remove redundant weights or neurons.
- **Distillation**: train a smaller student on teacher outputs.
- **PEFT**: LoRA, Adapters, ...

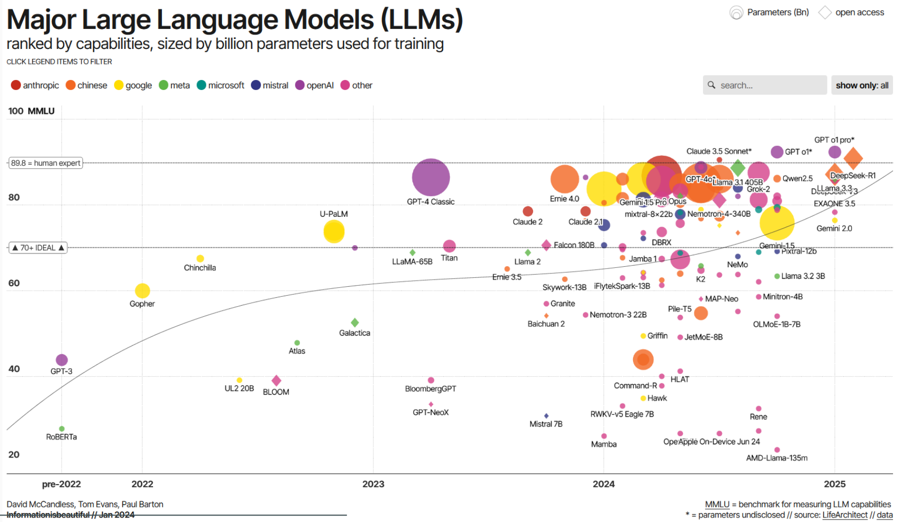
[20] S. Park, J. Choi, S. Lee, and U. Kang (2024). A Comprehensive Survey of Compression Algorithms for Language Models. arXiv: 2401.15347 [cs.CL]

# Where is this going?[21]




[21] https://ourworldindata.org/grapher/artificial-intelligence-parameter-count

# Where is this going?[22]




[22] https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/

# Critical Reflections on Large Language Models

**Scaling is powerful, but not free**

**Open Issues:**

- Environmental cost
- Evaluation transparency
- Data ethics
- Model accessibility

→ **Compute cost:** training GPT-3 used hundreds of PFLOPs-days; carbon footprint estimated at hundreds of tons CO[23].

→ **Financial cost:** GPT-4-level training estimated at millions of dollars; access to compute increasingly centralized[24].

→ **Data concerns:** Training data scraped from the web—raises copyright, consent, and fairness issues[25].

→ **Evaluation gaps:** Benchmarks often narrow and do not capture robustness, fairness, or real-world alignment[26].

[23] https://news.mit.edu/2025/explained-generative-ai-environmental-impact-0117
[24] https://hai.stanford.edu/ai-index/2025-ai-index-report (page 65)
[25] P. Samuelson (2023). "Generative AI meets copyright". In: Science 381, 158–161
[26] T. R. McIntosh et al. (2025). "Inadequacies of large language model benchmarks in the era of

# Gesture Detection

# Step 1: Video Input

**Input Type:**

- Usually RGB frames or video stream
- Optionally with depth or IR

**Preprocessing:**

- Resize, normalize
- Frame extraction or windowing
- Optional face/hands segmentation

**Challenges:**

- Varying lighting and backgrounds
- Occlusion (e.g. hands crossing)
- Real-time constraints (latency, FPS)
- Device variability (camera quality)

# Step 2: Hand Pose Detection

**Goal:** Localize key hand joints (2D/3D)

- Wrist, knuckles, fingertips
- Input: video frame or cropped hand region

- Accuracy depends on input quality and occlusions
- Trade-off between model size and speed
- 3D pose enables gesture generalization (rotation invariance)
- Tracking is often fused with detection for consistency

# Excursus: Two Pipelines for Pose Estimation[27]

**Top-Down (two-step)**

- Detect each person/hand first (e.g. bounding box).
- Run a pose/gesture network inside every box.
- **Pros:** high single-instance accuracy; leverages powerful object detectors.
- **Cons:** time scales #people; errors cascade from detector;

**Bottom-Up (part-based)**

- Detect all keypoints in the frame at once.
- Group points into individuals via part-affinity / clustering.
- **Pros:** cost nearly constant to crowd size; robust to missed boxes.
- **Cons:** grouping step can fail in heavy occlusion; slightly lower peak accuracy.

[27] R. Yue, Z. Tian, and S. Du (2022). "Action recognition based on RGB and skeleton data sets: A survey". In: Neurocomputing 512, 287–306

# Excursus: Common Models

- **MediaPipe**[28] **(Google)** - Lightweight, real-time framework for hand pose (21 keypoints per hand). Ideal for single-person tracking on mobile and web. Integrated into many apps and easy to use.
- **OpenPose**[29] **(CMU)** - Pose model supporting hands, body, and face. Requires GPU. Strong multi-person support. Still a popular baseline in research.
- **MMPose**[30] **(OpenMMLab)** - Modular PyTorch framework supporting many backbones and datasets. Includes whole-body hand keypoints and supports both 2D and 3D models. Great for custom experiments.
- **Sapiens**[31] **(Meta)** - Newest, high-resolution foundation model with 308 keypoints (including hands). Designed for detailed, frame-by-frame offline analysis, not real-time use.

[28] https://github.com/google-ai-edge/mediapipe
[29] https://github.com/CMU-Perceptual-Computing-Lab/openpose
[30] https://github.com/open-mmlab/mmpose
[31] https://github.com/facebookresearch/sapiens

# Step 3: Feature Encoding[32]

**Goal:** Convert hand pose data into useful input features for ML models.

- Raw 2D/3D keypoints
- Distances between joints
- Angles between fingers
- Motion vectors (velocity, acceleration)

**Feature Types:**

- Frame-based (pose snapshot)
- Sequence-based (temporal movement)
- Hand-crafted vs. deep-learned embeddings
- Normalize for translation, scale, rotation

[32] P. Molchanov et al. (2016). "Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks". In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4207–4215

# Step 4: ML Model Training[33]

**Input:** Encoded features or pose sequences

**Common Models:**

- Classical ML: SVM, k-NN, Random Forest
- Deep Learning: MLP, CNN, LSTM, Transformers
- Spatio-temporal models for gesture dynamics

**Training Considerations:**

- Supervised learning with gesture labels.
- Augment data for generalization.
- Cross-subject and cross-session robustness.

[33] J. J. Ojeda-Castelo, M. d. L. M. Capobianco-Uriarte, J. A. Piedra-Fernandez, and R. Ayala (2022). "A survey on intelligent gesture recognition techniques". In: IEEE Access 10, 87135–87156

# Step 5: Gesture Prediction[34]

**Goal:** Classify or detect user gestures in real-time.

**Output Types:**

- Static: e.g., "Thumbs up", "Open hand"
- Dynamic: e.g., "Swipe left", "Draw circle"

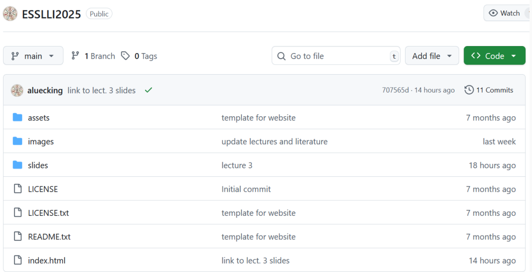
**Deployment:**

- Smooth output with tracking or temporal smoothing.
- Handle uncertain input with confidence thresholds.

[34] P. Molchanov et al. (2016). "Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks". In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4207–4215

# Hands On Example

# EnvisionBOX (`https://github.com/aluecking/ESSLLI2025` )



# EnvisionBOX (https://envisionbox.org/ )

# Future Directions

# Where are the LLMs for Gesture Detection?[36]

## Data & Representation

- **Sparse paired data**: very few corpora link *gesture key-points + language*; most instruction sets only contain captions.
- **Temporal mismatch**: LLMs digest static images; gestures are >30 FPS sequences ⇒ token explosion.
- **Modality gap**: 2-D RGB misses depth, skeleton cues essential for fine-grained hand motion.

## Model Capabilities

- **Spatial precision**: overlay-based prompting (e.g. ViP-LLaVA) leaves 3-D joint reasoning unsolved.[35]
- **Reasoning granularity**: current MLLMs excel at object semantics, but struggle with fine motor actions (pinch, swipe).
- **Safety/bias**: ambiguous gestures vary culturally; no robust alignment or policy-tuning yet.

[35] M. Cai et al. (2024). "Making Large Multimodal Models Understand Arbitrary Visual Prompts". In: IEEE Conference on Computer Vision and Pattern Recognition

[36] D. Feng et al. (2025). "PoseLLaVA: Pose Centric Multimodal LLM for Fine-Grained 3D Pose Manipulation". In: Proceedings of the AAAI Conference on Artificial Intelligence 39, 2951–2959; D. Zhang, T. Hussain, W. An, and H. Shouno (2025). LLaVA-Pose: Enhancing Human Pose and Action Understanding via Keypoint-Integrated Instruction Tuning. arXiv: 2506.21317 [cs.CV]

## Where are the LLMs for Gesture Detection?

### PoseLLaVA[37]

- Model changes: adds a pose encoder plus a cross-attention into LLaVA for global & local pose-image.
- Created datasets: adds **PosePart** (135 K single-body-part triplets) and combines Human3.6M (300 K), PoseScript (100 K) and PoseFix (135 K).
- Finetuning task: three-stage pipeline: pose-image contrastive pre-align, LLM pre-train on pose generation, unified instruction-tuning over estimation / generation / adjustment.

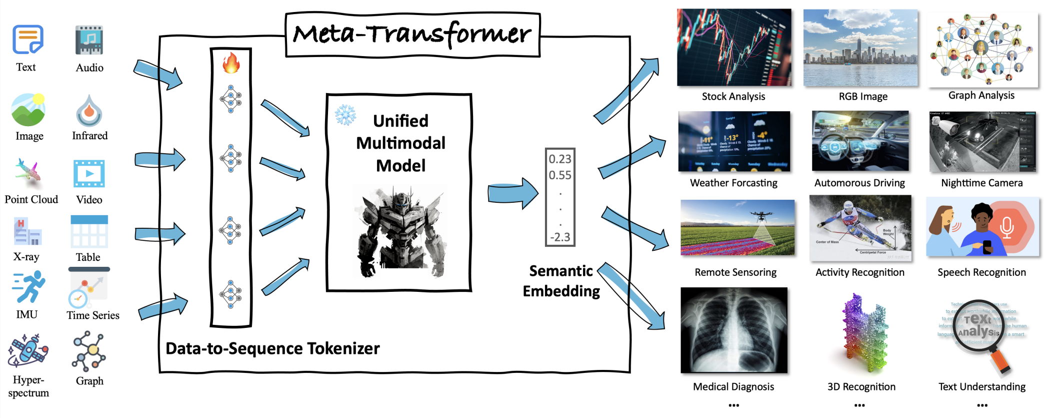
### LLaVA-Pose[38]

- Model changes: no architectural edits - retains LLaVA-1.5 and simply augments prompts with 2-D keypoints.
- Created datasets: auto-generates 200328 COCO-based keypoint-aware instructions (conversation / description / reasoning); also publishes the **E-HPAUB** benchmark for evaluation.
- Finetuning task: full-model fine-tune for one epoch; objective is richer chat, description & reasoning about human pose/action scenes.

[37] D. Feng et al. (2025). "PoseLLaVA: Pose Centric Multimodal LLM for Fine-Grained 3D Pose Manipulation". In: Proceedings of the AAAI Conference on Artificial Intelligence 39, 2951–2959

[38] D. Zhang, T. Hussain, W. An, and H. Shouno (2025). LLaVA-Pose: Enhancing Human Pose and

# Concept: Meta-Transformer[39]




[39] Y. Zhang et al. (2023). "Meta-transformer: A unified framework for multimodal learning". In: arXiv preprint arXiv:2307.10802

# Why most LLMs stay text & image[40]

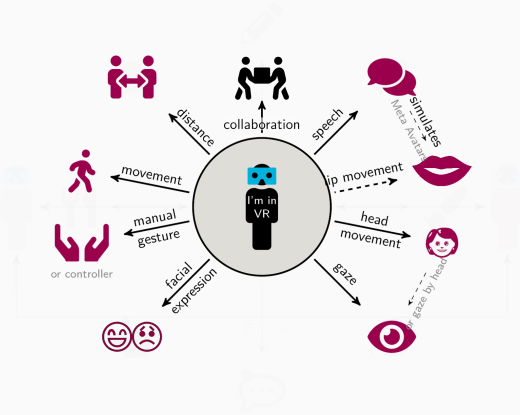
- **Compute / memory** - universal tokens make long sequences; self-attention costly.
- **Limited generative** - good at unimodal perception, unclear for cross-modal generation.
- **Dataset gaps** - few richly multimodal pairs to unlock full promise.
- **Data scale** - trillions of web tokens, billions of captions; far fewer paired corpora.
- **Token explosion** - 10s of 30FPS video $\approx$ 900 frames $\Rightarrow$ hundreds of tokens.
- **ROI focus** - chat, code, doc QA, image help already monetize; niche sensors give uncertain payoff.
- **Tooling maturity** - CLIP/LLaVA pipelines are production-ready; multimodal 3-D/audio stacks still research-grade.

---

[40] A. Henlein et al. (2024). "An Outlook for AI Innovation in Multimodal Communication Research". In: Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management, 182–234; J. Jiang et al. (2025). Token-Efficient Long Video Understanding for Multimodal LLMs. arXiv: 2503.04130 [cs.CV]; A. Kumar, M. M. Salim, D. Camacho, and J. H. Park (2025). "A comprehensive survey on large language models for multimedia data security: challenges and solutions". In: Computer Networks 267, 111379; Y. Zhang et al. (2023). "Meta-transformer: A unified framework for multimodal learning". In: arXiv preprint arXiv:2307.10802

# Va.Si.Li-Lab[41]

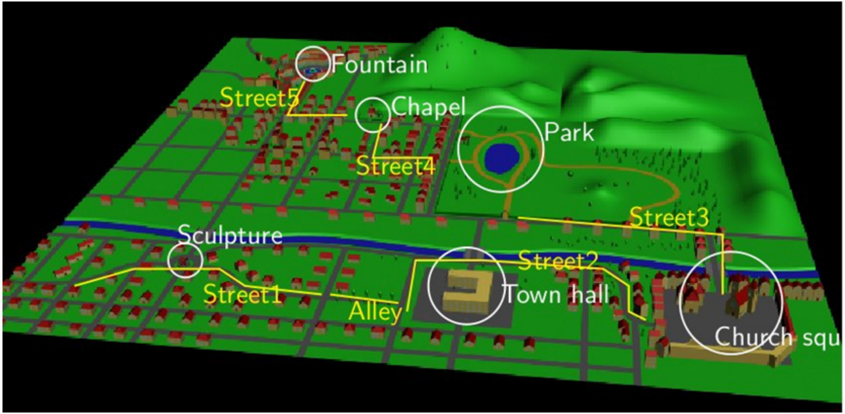
- A VR-based simulation system.
- Multi-user collaborative tool.
- Users are represented by Meta Avatars.




[41] A. Mehler et al. (2023). "A Multimodal Data Model for Simulation-Based Learning with Va.Si.Li-Lab". In: Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management, 539–565

# Va.Si.Li-Lab

# SaGA 2.0: FraGA[42]

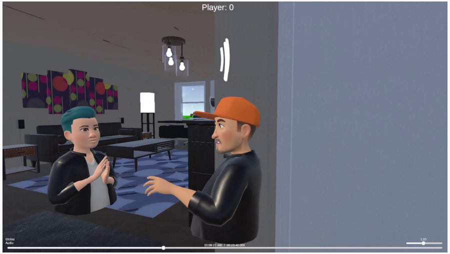



[42] A. Lücking et al. (2010). "The Bielefeld Speech and Gesture Alignment Corpus (SaGA)". In: Multimodal Corpora: Advances in Capturing, Coding and Analyzing Multimodality. 7th International Conference for Language Resources and Evaluation, 92–98

# SaGA 2.0: FraGA[43]

73 dialogues involving 146 speakers.

| | Speaking time | # Tokens |
|---|---|---|
| total: | 12:44:37 | 92,923 |
| Router: | 8:17:19 | 70,517 |
| Follower: | 4:27:18 | 22,406 |
| Avg. Router: | 0:06:49 | 1,273 |
| Avg. Follower: | 0:03:40 | 966 |
| Avg. Dialogue: | 0:10:28 | 307 |




[43] Lücking, Voll, Rott, Henlein, Mehler (2025). "Head and hand movements during turn transitions: data-based multimodal analysis using the Frankfurt VR Gesture–Speech Alignment Corpus (FraGA)". In: accepted. 29th edition of the SemDial workshop series