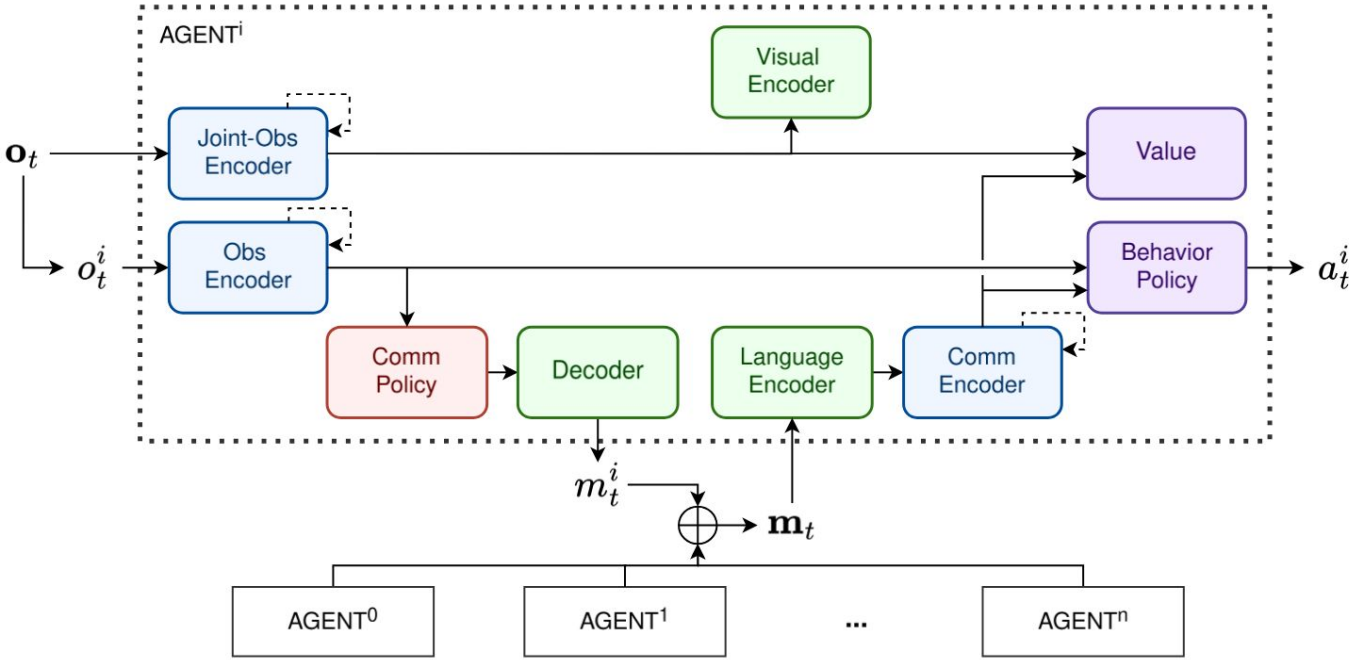
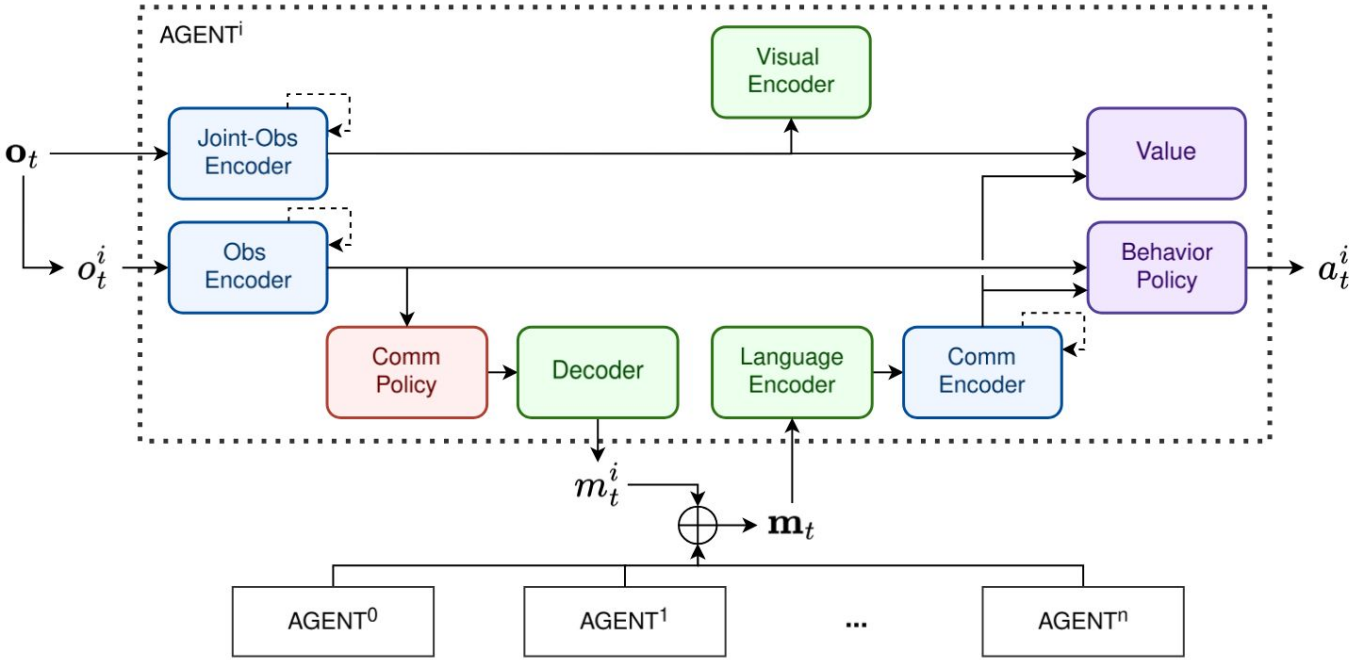
Learning Social Visual-Language-Action Models

February 18th, 2025



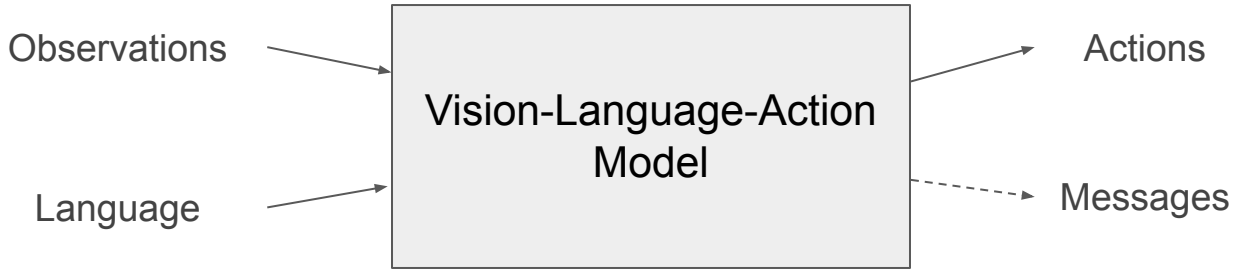
- Extend MADRL agent with language → Able to learn language and task
- Train RL and language concurrently → Learning language helps task training
- Relatively small architecture (~240k params)



Issues

- Multiple separate streams of information
- Arbitrary arrangement

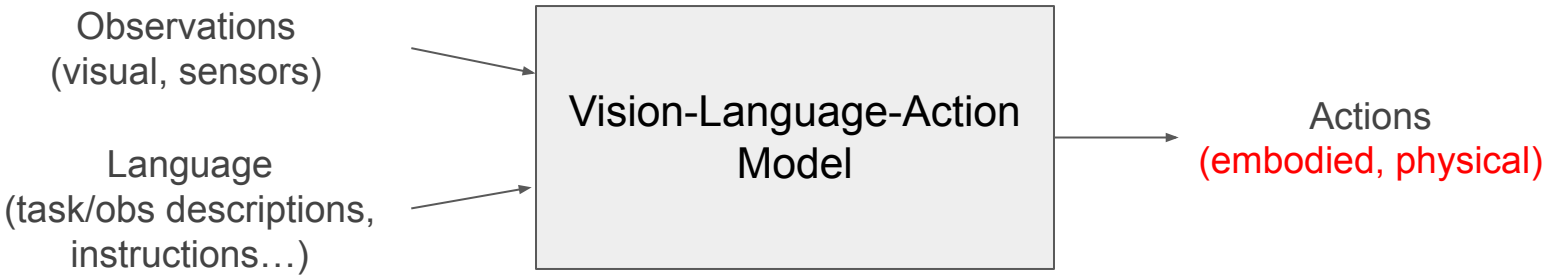
⇒ Can we use a Transformer as a central module?



⇒ **How can we extend VLAs to work in social contexts ?**

(i.e., with language communication)

Visual-Language-Action Models



[Ma et al., 2024]

Visual-Language-Action Models

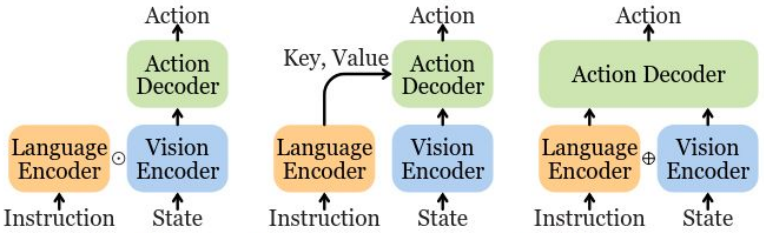


Figure 4: Three common architectures of Transformer-based control policies are characterized by their vision-language fusion methods. FiLM layers (Hadamard product $\odot$) are employed to fuse language and vision in RT-1 models. Some action decoders utilize cross-attention to condition on the instruction. Concatenation ($\oplus$) is the prevailing method of vision-language fusion for action decoders.

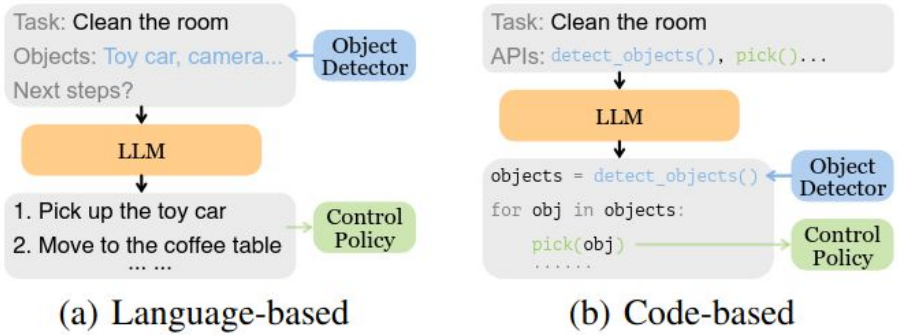


Figure 5: Different approaches to connect LLM to multi-modal modules in high-level task planner.

[Ma et al., 2024]

Visual-Language-Action Models

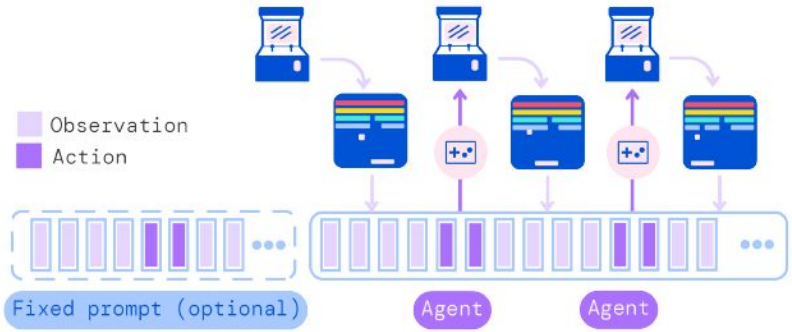


Figure 3: **Running Gato as a control policy.** Gato consumes a sequence of interleaved tokenized observations, separator tokens, and previously sampled actions to produce the next action in standard autoregressive manner. The new action is applied to the environment – a game console in this illustration, a new set of observations is obtained, and the process repeats.

[Reed et al., 2022]

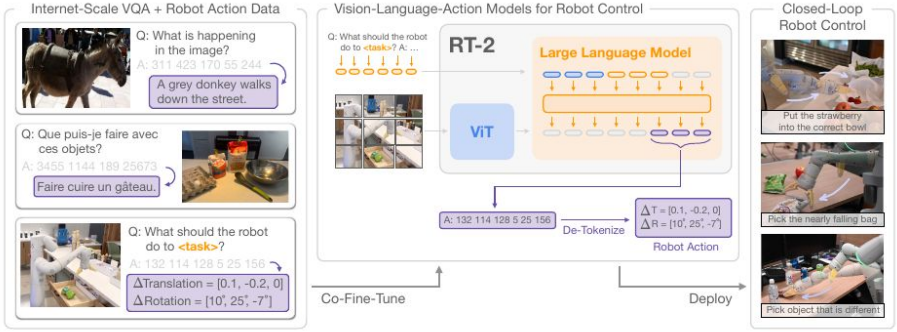


Figure 1 | RT-2 overview: we represent robot actions as another language, which can be cast into text tokens and trained together with Internet-scale vision-language datasets. During inference, the text tokens are de-tokenized into robot actions, enabling closed loop control. This allows us to leverage the backbone and pretraining of vision-language models in learning robotic policies, transferring some of their generalization, semantic understanding, and reasoning to robotic control. We demonstrate examples of RT-2 execution on the project website: robotics-transformer2.github.io.

[Zitkovich et al., 2023]

Visual-Language-Action Models

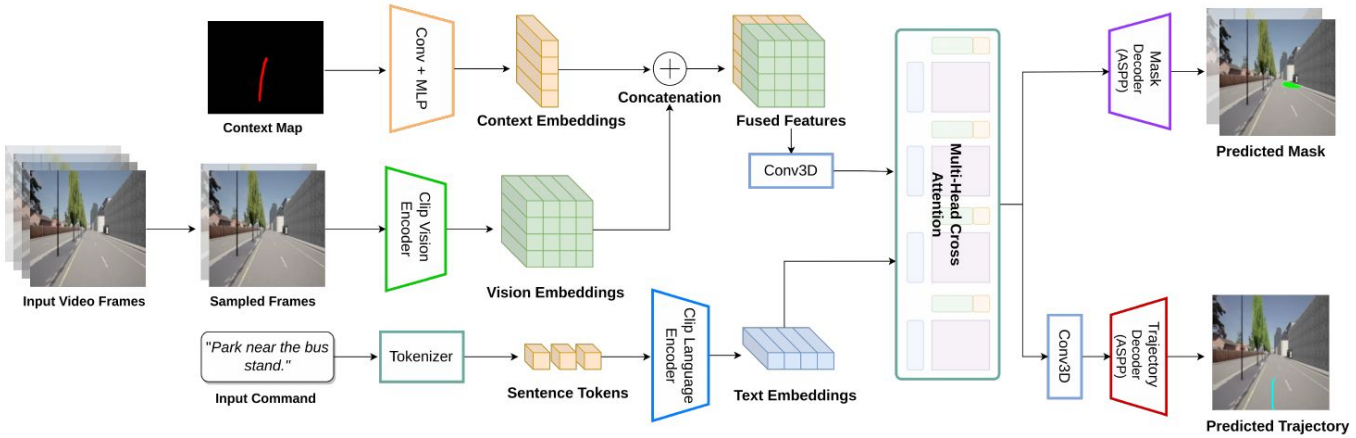


Fig. 3. Overall pipeline of the proposed approach. Given the visual frames, already executed past trajectory (context map) and the textual command, the network predicts a segmentation map corresponding to the navigable region and a plausible future trajectory.

[Jain et al., 2023]

LLM societies

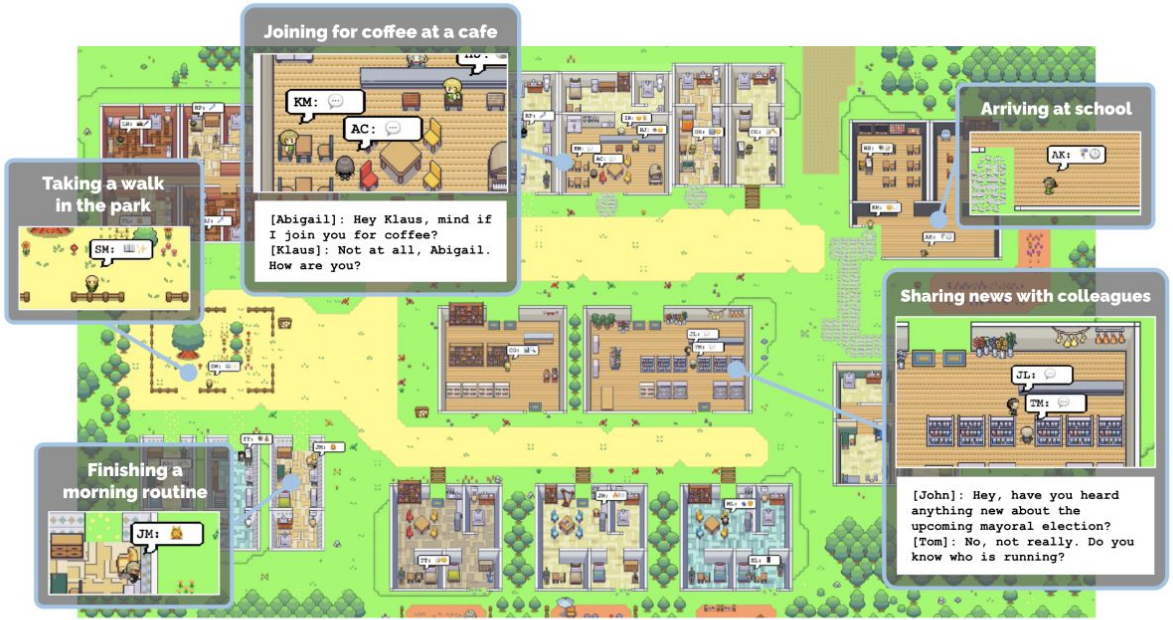


Figure 1: Generative agents are believable simulacra of human behavior for interactive applications. In this work, we demonstrate generative agents by populating a sandbox environment, reminiscent of The Sims, with twenty-five agents. Users can observe and intervene as agents plan their days, share news, form relationships, and coordinate group activities.

[Park et al., 2023]

- Multi-agent
- Heavily language-based
- Complex language/tasks
- No embodiment

Transformer-based RL agents

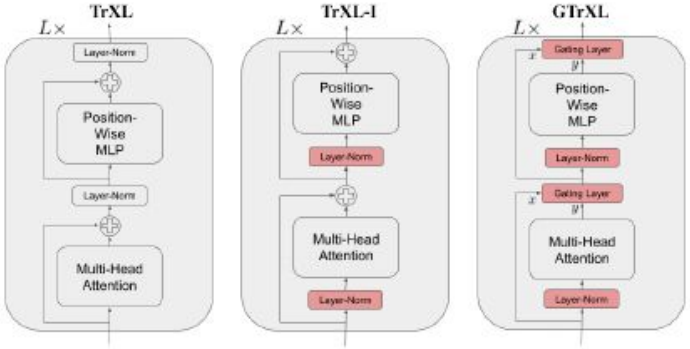


Figure 1. Transformer variants, showing just a single layer block (there are L layers total). **Left:** Canonical Transformer(-XL) block with multi-head attention and position-wise MLP submodules and the standard layer normalization (Ba et al., 2016) placement with respect to the residual connection (He et al., 2016a). **Center:** TrXL-I moves the layer normalization to the input stream of the submodules. Coupled with the residual connections, there is a gradient path that flows from output to input without any transformations. **Right:** The GTrXL block, which additionally adds a gating layer in place of the residual connection of the TrXL-I.

[Parisotto et al., 2021]

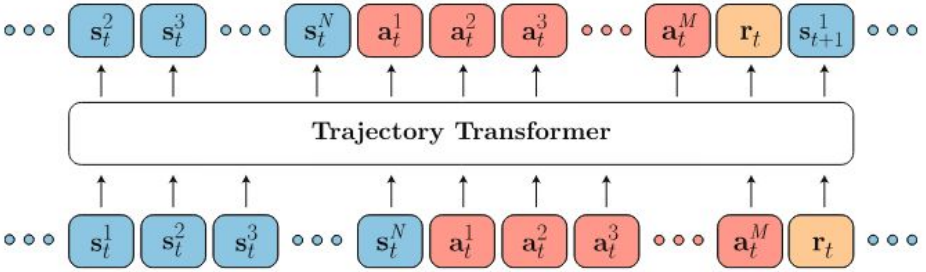


Figure 1 (Architecture) The Trajectory Transformer trains on sequences of (autoregressively discretized) states, actions, and rewards. Planning with the Trajectory Transformer mirrors the sampling procedure used to generate sequences from a language model.

[Janner et al., 2021]

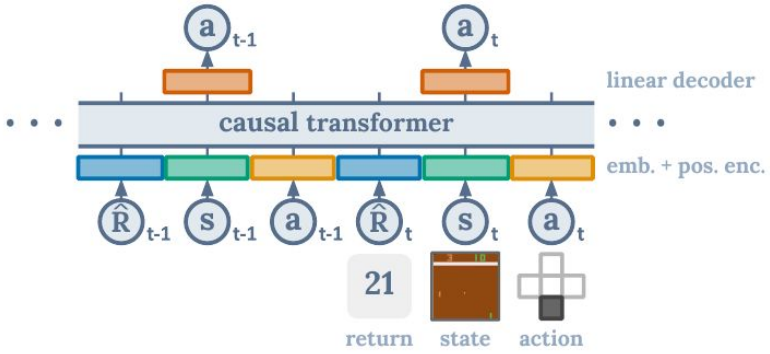


Figure 1: Decision Transformer architecture[1]. States, actions, and returns are fed into modality-specific linear embeddings and a positional episodic timestep encoding is added. Tokens are fed into a GPT architecture which predicts actions autoregressively using a causal self-attention mask.

[Chen et al., 2021]

Transformer-based RL agents

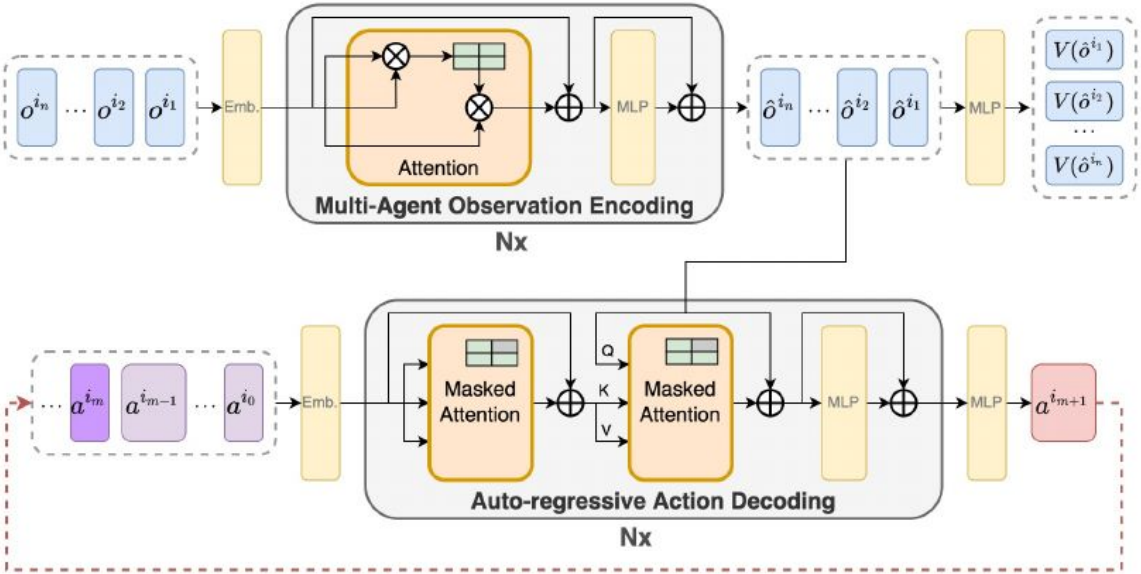


Figure 2: The encoder-decoder architecture of MAT. At each time step, the encoder takes in a sequence of agents' observations and encodes them into a sequence of latent representations, which is then passed into the decoder. The decoder generate each agent's optimal action in a sequential and auto-regressive manner. The masked attention blocks ensures agents can only access its preceding agents' actions during training. We list the full pseudocode of MAT in Appendix A and a video that shows the dynamic data flow of MAT in https://sites.google.com/view/multi-agent-transformer.

[Wen et al., 2022]

	Multi-agent	Embodiment	Model type	Training
VLAs	No	Yes	Transformer/LLM	Pre-trained
LLM-societies	Yes	No	LLM	Pre-trained
Transformer-based RL	No	Yes	Transformer	End-to-end

How to bridge the gap between these approaches?

→ Look at interesting edge cases

Inputs

- How multi-modal information is fed into the network?

Link to actions

- How (language) actions are interpreted?

(Pre-)Training

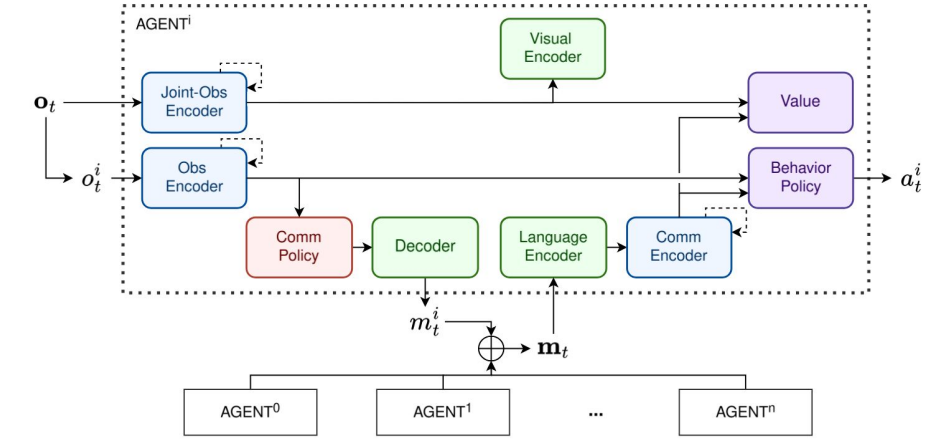
- Do they rely on pre-trained models? Of what size?
- Is physical behaviour learnt or primitive-based?
- Is there online RL? For what purpose (high/low-level actions)?

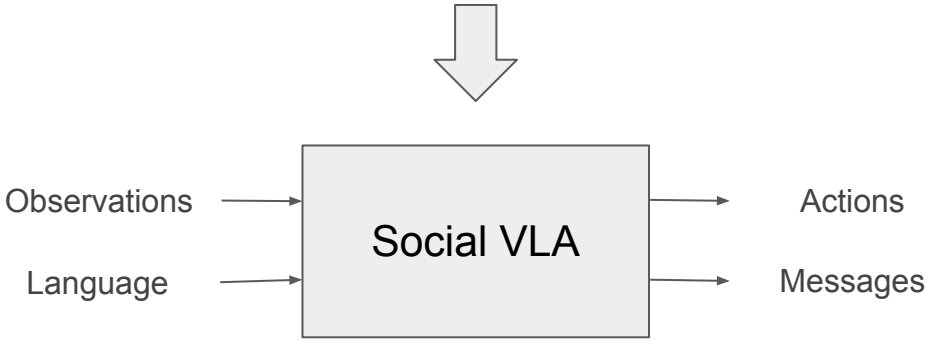
Taxonomy

	Approach	Model	Actions	Inputs	Policy learning	Multi-agent paradigm	Specificity
Jain2023_CARLANAV	VLA	Transformer	Low-level	Visual, language	Supervised	-	Learning a Transformer for grounding pre-trained visual and language encoders to trajectories
Driess2023_PaLM-E	VLA	LLM (PaLM)	Language, translated to primitives	Visual, 3D, language, embedded as tokens	Pre-trained	-	Learning input encoders
Brohan2023_RT2	VLA	LLM (PaLM-E, PaLI-X)	Low-level, represented as tokens	Visual, language, embedded as tokens	Pre-trained	-	Tokenizing low-level actions and learning to generate sequences
Park2023_LLMTown	LLM-society	LLM (ChatGPT)	Language, translated to primitives	Language	Pre-trained	Distributed	
Wen2022_MAT	Transformer-RL	Transformer	Low-level	Visual	PPO	Centralised	Central Transformer-Encoder taking observations, Transfomer-Decoder generating actions of all agents

https://www.notion.so/VLAs-LLM-agents-and-societies-Transformer-RL-agents-19edbdcbd68480e58a59c50180735e19?pvs=4

Our objective





Requirements

- End-to-end learning of task+language
- Small architecture (max 500M)

Goals

- Same results as in LAMARL
- Does pre-training help?
- Does size help?
- Can we have completely distributed Social VLAs?

Reviews

VLA

- Ma2024: https://arxiv.org/pdf/2405.14093

LLM-society

- Xi2025: https://link.springer.com/article/10.1007/s11432-024-4222-0
- Li2025: https://arxiv.org/pdf/2502.03814
- Guo2024: https://arxiv.org/pdf/2402.01680

Transformer-RL

- …