LEARNING TO ACT

Session 2: Behaviour Cloning

Dana Kulić
Monash University
dana.kulic@monash.edu

RVSS 2025

Today's session – Learning from human demonstrations

- Early approaches to Behavioural Cloning (BC)
- Deep Generative models for BC
- Foundation models for policy generation

Behavioural Cloning - Recall from Session 1

- Collect data from demonstration episodes $\mathcal{D}(e_{1:N})$
- Each episode is a sequence of states and actions
  $e_i = (s_0, a_1, s_1, a_2, \ldots s_T)$
- Learn a policy $\phi(s)$ using supervised learning:

$$L = (a_{\mathcal{D}}(s) - \phi(s))^2$$

  - The state s corresponds to the input data
  - The action a corresponds to the label
  - Behavioural cloning learns the policy function $\phi(s)$ to minimise the difference between the estimated action and the observed expert action from each state

Behavioural Cloning – potential problems

- How to collect the expert data?
- What is the right state representation? Does the robot see the same things as the expert does?
- Expert demonstrations may cover only a very small region of the state-space
  - For large state/action spaces, may require a huge data collection effort
- What should the robot do when it encounters a situation that wasn't seen in the dataset?
- How to handle variations in strategy?

Behavioural Cloning

Algorithm 1 Abstract of behavioral cloning

Collect a set of trajectories demonstrated by the expert $\mathcal{D}$
Select a policy representation π_θ
Select an objective function $\mathcal{L}$
Optimize $\mathcal{L}$ w.r.t. the policy parameter θ using $\mathcal{D}$
return optimized policy parameters θ

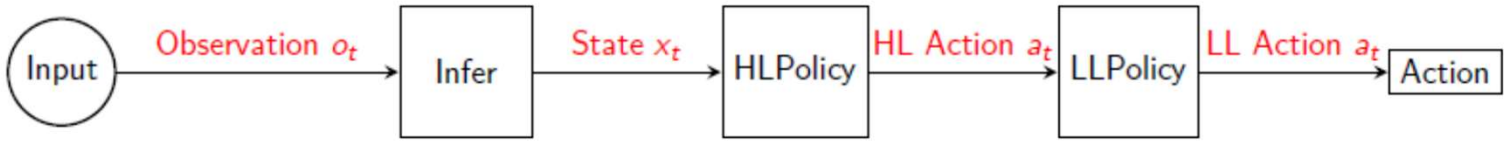
Key Design Choices:

- What loss function should be used to represent the difference between demonstrated and produced behaviour?
- How should the state be represented?
- How should the policy be represented?
- Output a single action or a trajectory?

T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel and J. Peters, An Algorithmic Perspective on Imitation Learning, Foundations and Trends in Robotics, 2018.

Choices for Problem Formulation

Modular



End-to-end

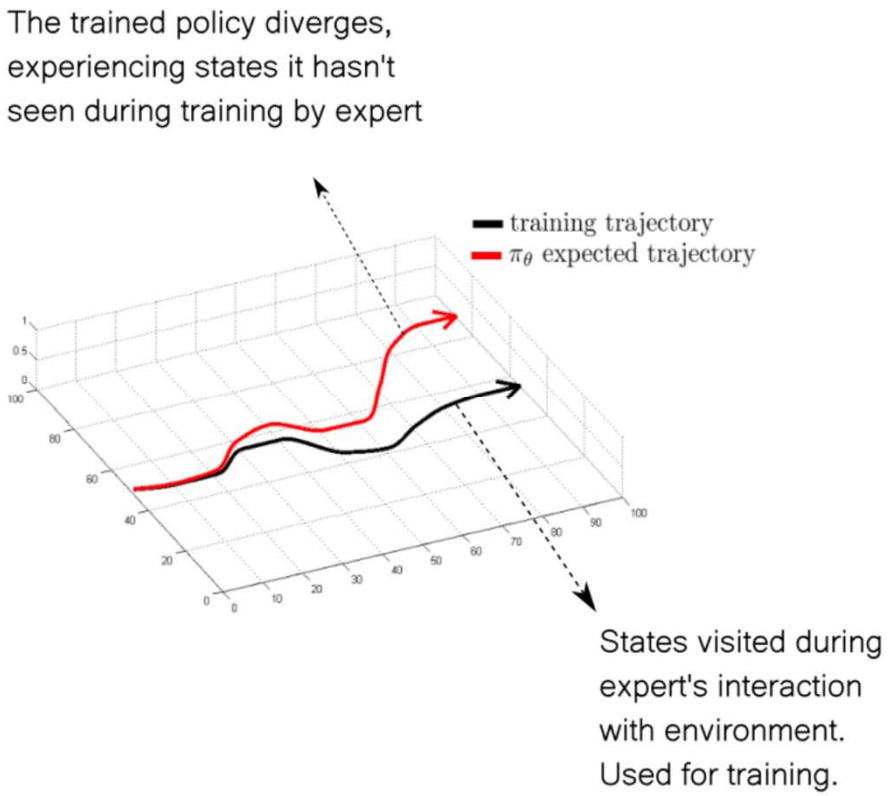
Behavioural Cloning Past Work

Trajectory Learning	Gaussian Model	[Paraschos et al., 2013, Maeda et al., 2016]
	GMR	[Calinon and Billard, 2009, Gribovskaya et al., 2011, Khansari-Zadeh and Billard, 2014, Calinon, 2016]
	LWR	[Schaal and Atkeson, 1998, Mülling et al., 2013, Osa et al., 2017a]
	LWPR	[Vijayakumar et al., 2005]
	GPR	[Osa et al., 2017b]
Action-State Space	Look-Up Table	[Chambers and Michie, 1969]
	Linear Regression	[Widrow and Smith, 1964]
	Neural Network	[Pomerleau, 1988, LeCun et al., 2006, Stadie et al., 2017, Duan et al., 2017]
	Decision Tree	[Sammut et al., 1992]
	LWR	[Atkeson and Schaal, 1997]
	LWPR	[Vijayakumar and Schaal, 2000]

T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel and J. Peters, An Algorithmic Perspective on Imitation Learning, Foundations and Trends in Robotics, 2018.

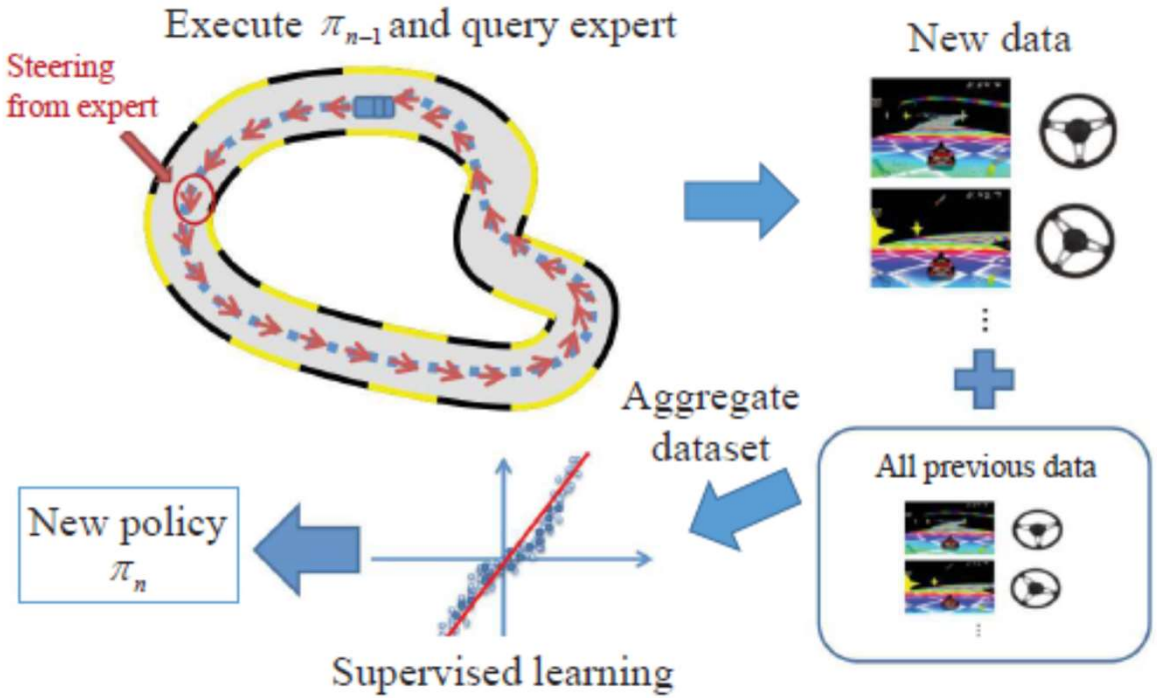
Early Lessons

- BC from expert demonstrations doesn’t work very well!
- Learning errors cascade, learner quickly gets into parts of the state-space never encountered by the expert during successful demonstrations
- Hard to handle variations in strategy



T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel and J. Peters, An Algorithmic Perspective on Imitation Learning, Foundations and Trends in Robotics, 2018.

Curated/Interactive data collection (DAGGER)



- DAGGER [Bagnell 2015]: Ask the expert to provide samples under the input distribution of the learner

T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel and J. Peters, An Algorithmic Perspective on Imitation Learning, Foundations and Trends in Robotics, 2018.

The rise of Deep Generative Methods

- Can we adapt the innovations in deep generative models to policy learning?
  - Can we benefit from new generative approaches?
  - Can we benefit from existing LLMs/VLMs?
  - What about training robot-specific LMs?

Generative Models for Action

Problem Formulation

Learn a conditional probability density model (generative model)

$$\rho_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{c})$$

That accurately captures the underlying probability distribution of the data

$$\rho_{\mathcal{D}}(\boldsymbol{a}|\boldsymbol{c})$$

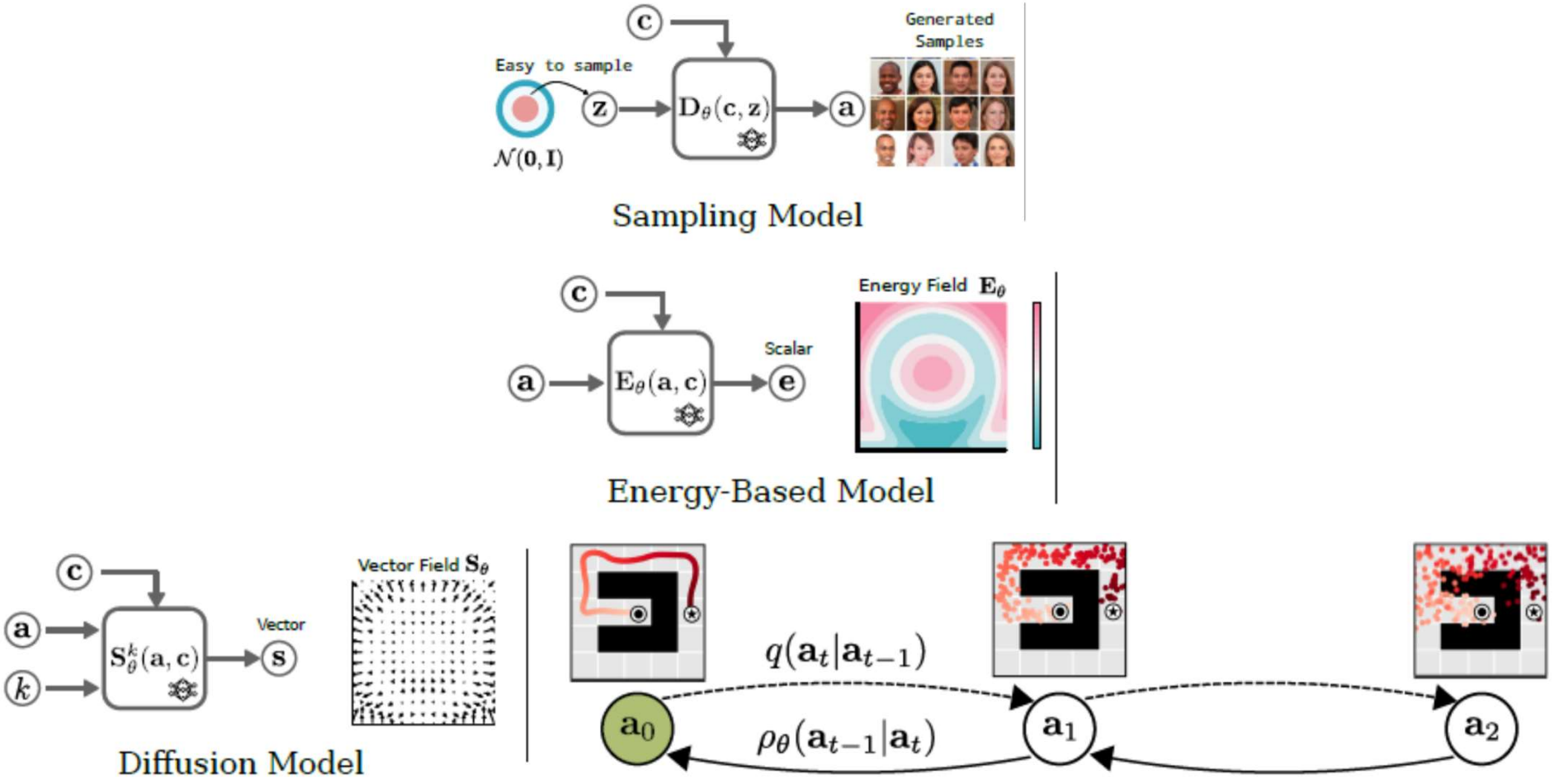
Where ***a*** is the action and ***c = (o,g)*** are the conditions (observations and goal)

Find the distribution that minimises the divergence between the model and the data distribution

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{a},\boldsymbol{c}\sim\mathcal{D}} \left[\mathbb{D}(\rho_{\mathcal{D}}(\boldsymbol{a}|\boldsymbol{c}), \rho_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{c}))\right]$$

J. Urain, A. Mandlekar, Y. Du, N. M. M. Shafiullah, D. Xu, K. Fragkiadaki, G. Chalvatzaki, J. Peters, Deep Generative Models in Robotics: A Survey of Learning from Multimodal Demonstrations, 2024

Density Estimation Models



J. Urain, A. Mandlekar, Y. Du, N. M. M. Shafiullah, D. Xu, K. Fragkiadaki, G. Chalvatzaki, J. Peters, Deep Generative Models in Robotics: A Survey of Learning from Multimodal Demonstrations, 2024

Diffusion Policy: Visuomotor Policy Learning via Action Diffusion

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, Shuran Song

RSS2023 paper

IJRR2024 paper

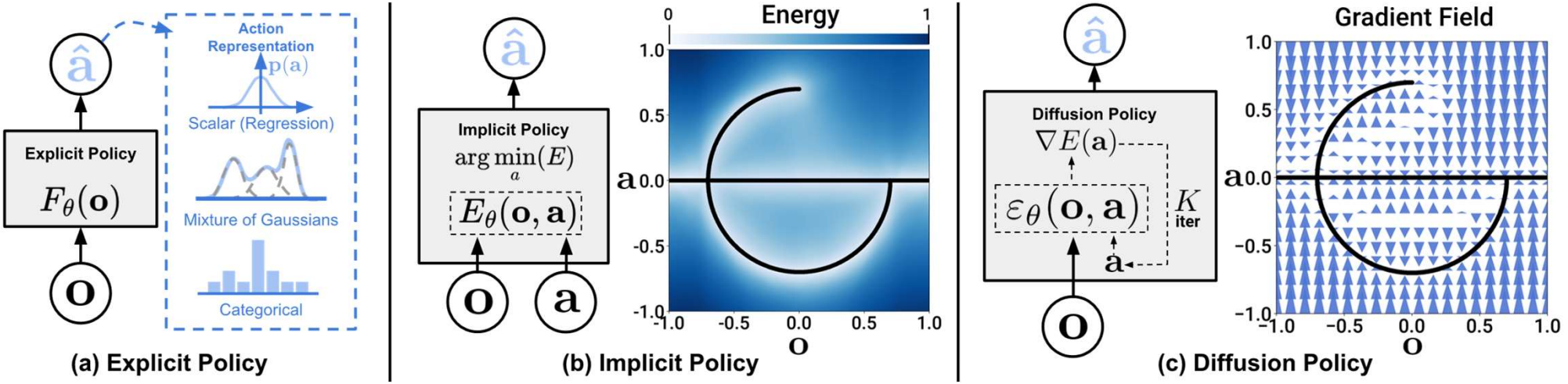
Project website

Motivation

- How to best represent the action policy when learning from demonstrations?
  - Need to be able to represent multimodal action distributions
  - Need to handle high-dimensional output space
  - Training stability

- Proposed approach:
  - represent the action as a conditional denoising diffusion process
  - Embed the action selection within a receding-horizon control framework
  - Policy conditioned on visual input
  - Investigate alternatives for the policy network architecture
  - Extensive validation

Paper Teaser



Diffusion

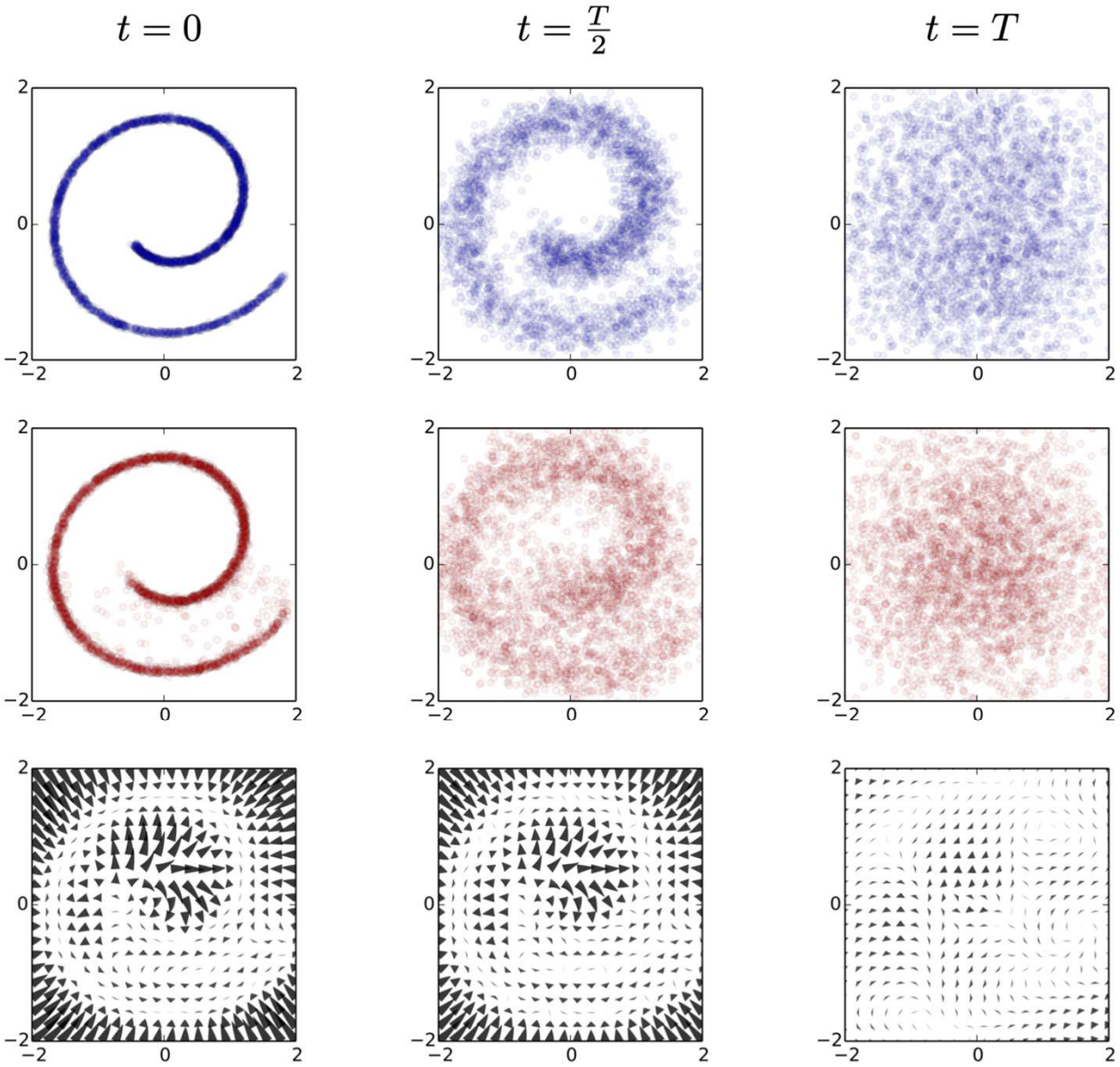


Image Courtesy of [Lilian Weng]

Diffusion Policy Formulation

- Formulate robot visuomotor policy as a denoising diffusion probabilistic model (DDPM) – output generation is modelled as a denoising process, aka Stochastic Langevin Dynamics

$$\mathbf{x}^{k-1} = \alpha(\mathbf{x}^k - \gamma\varepsilon_\theta(\mathbf{x}^k, k) + \mathcal{N}(0, \sigma^2 I)),$$

- Can be interpreted as a noisy gradient descent step

$$\mathbf{x}' = \mathbf{x} - \gamma\nabla E(\mathbf{x}),$$

ε_θ=noise prediction network
k = denoising iteration
$\mathcal{N}$= Gaussian noise
α,γ = free parameters (interpreted as learning rate)

- To train the network, add appropriate noise for each k to noise-free ground truth

$$\mathcal{L} = MSE(\varepsilon^k, \varepsilon_\theta(\mathbf{x}^0 + \varepsilon^k, k))$$

Applying DDPMs to policy learning

- Action Sequence Prediction
  - Input: time window of observation data (observation horizon)
  - Output: time window of action data (action prediction horizon)
    - Sub-window executed without replanning (action execution horizon)
    - Sub-window predicted but not executed
- Visual observation conditioning

$$\mathbf{A}_t^{k-1} = \alpha(\mathbf{A}_t^k - \gamma\varepsilon_\theta(\mathbf{O}_t, \mathbf{A}_t^k, k) + \mathcal{N}(0, \sigma^2 I))$$

$$\mathcal{L} = MSE(\varepsilon^k, \varepsilon_\theta(\mathbf{O}_t, \mathbf{A}_t^0 + \varepsilon^k, k))$$

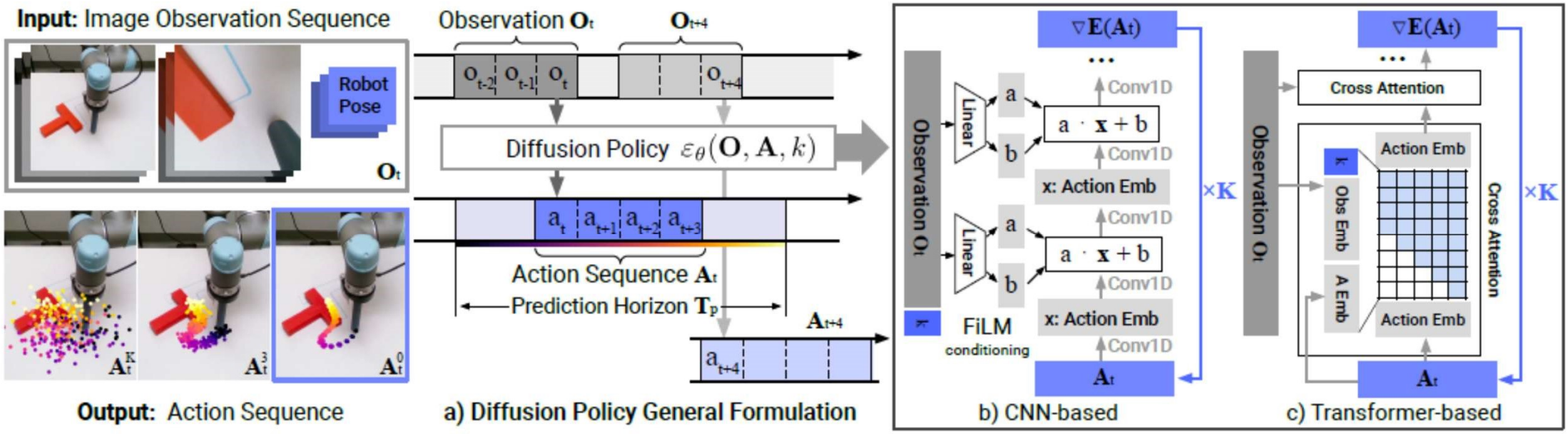
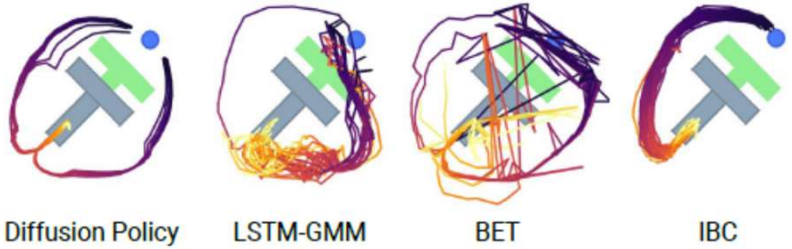


Figure 2. Diffusion Policy Overview a) General formulation. At time step t, the policy takes the latest T_o steps of observation data O_t as input and outputs T_a steps of actions A_t. b) In the CNN-based Diffusion Policy, FiLM (Feature-wise Linear Modulation) Perez et al. (2018) conditioning of the observation feature O_t is applied to every convolution layer, channel-wise. Starting from $\mathbf{A}_t^K$ drawn from Gaussian noise, the output of noise-prediction network ε_θ is subtracted, repeating K times to get $\mathbf{A}_t^0$, the denoised action sequence. c) In the Transformer-based Vaswani et al. (2017) Diffusion Policy, the embedding of observation $\mathbf{O}_t$ is passed into a multi-head cross-attention layer of each transformer decoder block. Each action embedding is constrained to only attend to itself and previous action embeddings (causal attention) using the attention mask illustrated.

Key Desirable Properties of Diffusion Policies

- Models multi-modal action distributions



- Works better with position control (vs. velocity control)
- Action-Sequence prediction is a good idea
  - Temporal action consistency
  - Robustness to idle actions
- Training Stability
- Connection to Control Theory
  - If the model is linear and cost quadratic, DP generates proportional controller with optimised gain!

Evaluation

- Simulation Environments and datasets:
  - Robomimic: robotic manipulation benchmark (both proficient and non-proficient demonstrators)
  - Push-T: pushing a T-shaped block
  - Multimodal Block Pushing (tests multimodal policies)
  - Franka Kitchen: long-horizon tasks
- Real world evaluation:
  - Uni-manual tasks: Push-T, Mug flipping, Sauce Pouring and spreading
  - Bi-manual tasks: Egg Beater, Mat Unrolling, Shirt Folding

- Videos: https://diffusion-policy.cs.columbia.edu/

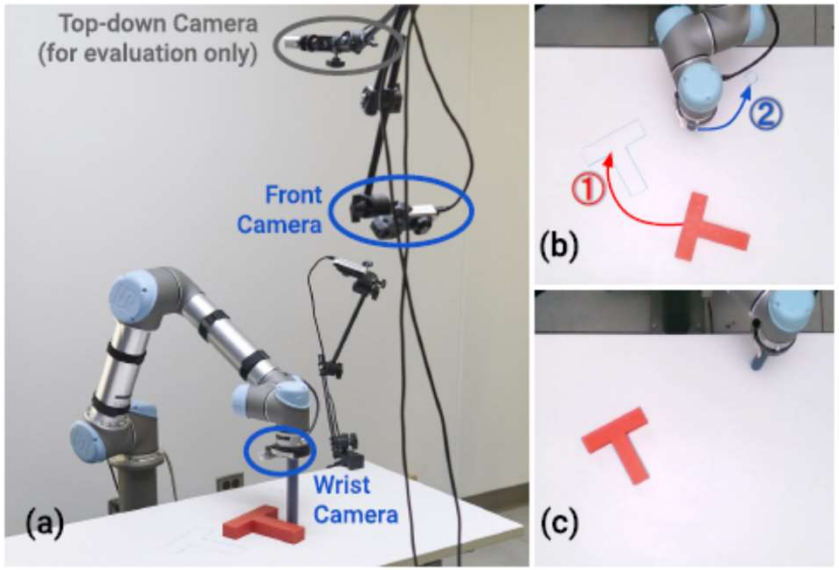
Lift | Can | Square | Transport | Tool Hang | Push-T

	Lift		Can		Square		Transport		ToolHang	Push-T
	ph	mh	ph	mh	ph	mh	ph	mh	ph	ph
LSTM-GMM	**1.00**/0.96	**1.00**/0.93	**1.00**/0.91	**1.00**/0.81	0.95/0.73	0.86/0.59	0.76/0.47	0.62/0.20	0.67/0.31	0.67/0.61
IBC	0.79/0.41	0.15/0.02	0.00/0.00	0.01/0.01	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.90/0.84
BET	**1.00**/0.96	**1.00**/0.99	**1.00**/0.89	**1.00**/0.90	0.76/0.52	0.68/0.43	0.38/0.14	0.21/0.06	0.58/0.20	0.79/0.70
DiffusionPolicy-C	**1.00**/0.98	**1.00**/0.97	**1.00**/0.96	**1.00/0.96**	**1.00/0.93**	**0.97/0.82**	0.94/0.82	**0.68/0.46**	0.50/0.30	0.95/**0.91**
DiffusionPolicy-T	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00**/0.94	**1.00**/0.89	0.95/0.81	**1.00/0.84**	0.62/0.35	**1.00/0.87**	**0.95**/0.79

Table 1. Behavior Cloning Benchmark (State Policy) We present success rates with different checkpoint selection methods in the format of (max performance) / (average of last 10 checkpoints), with each averaged across 3 training seeds and 50 different environment initial conditions (150 in total). LSTM-GMM corresponds to BC-RNN in RoboMimicMandlekar et al. (2021), which we reproduced and obtained slightly better results than the original paper. Our results show that Diffusion Policy significantly improves state-of-the-art performance across the board.

	Lift		Can		Square		Transport		ToolHang	Push-T
	ph	mh	ph	mh	ph	mh	ph	mh	ph	ph
LSTM-GMM	**1.00**/0.96	**1.00**/0.95	**1.00**/0.88	0.98/0.90	0.82/0.59	0.64/0.38	0.88/0.62	0.44/0.24	0.68/0.49	0.69/0.54
IBC	0.94/0.73	0.39/0.05	0.08/0.01	0.00/0.00	0.03/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.75/0.64
DiffusionPolicy-C	**1.00/1.00**	**1.00/1.00**	**1.00**/0.97	**1.00**/0.96	0.98/**0.92**	**0.98/0.84**	**1.00/0.93**	**0.89/0.69**	**0.95/0.73**	**0.91/0.84**
DiffusionPolicy-T	**1.00/1.00**	**1.00**/0.99	**1.00/0.98**	**1.00/0.98**	**1.00**/0.90	0.94/0.80	0.98/0.81	0.73/0.50	0.76/0.47	0.78/0.66

Table 2. Behavior Cloning Benchmark (Visual Policy) Performance are reported in the same format as in Tab 1. LSTM-GMM numbers were reproduced to get a complete evaluation in addition to the best checkpoint performance reported. Diffusion Policy shows consistent performance improvement, especially for complex tasks like Transport and ToolHang.



	Human Demo	IBC pos	IBC vel	LSTM-GMM pos	LSTM-GMM vel	Diffusion Policy T-E2E	Diffusion Policy ImgNet	Diffusion Policy R3M	Diffusion Policy E2E
IoU	0.84	0.14	0.19	0.24	0.25	0.53	0.24	0.66	**0.80**
Succ%	1.00	0.00	0.00	0.20	0.10	0.65	0.15	0.80	**0.95**
Dur.	20.3	56.3	41.6	47.3	51.7	57.5	55.8	31.7	**22.9**

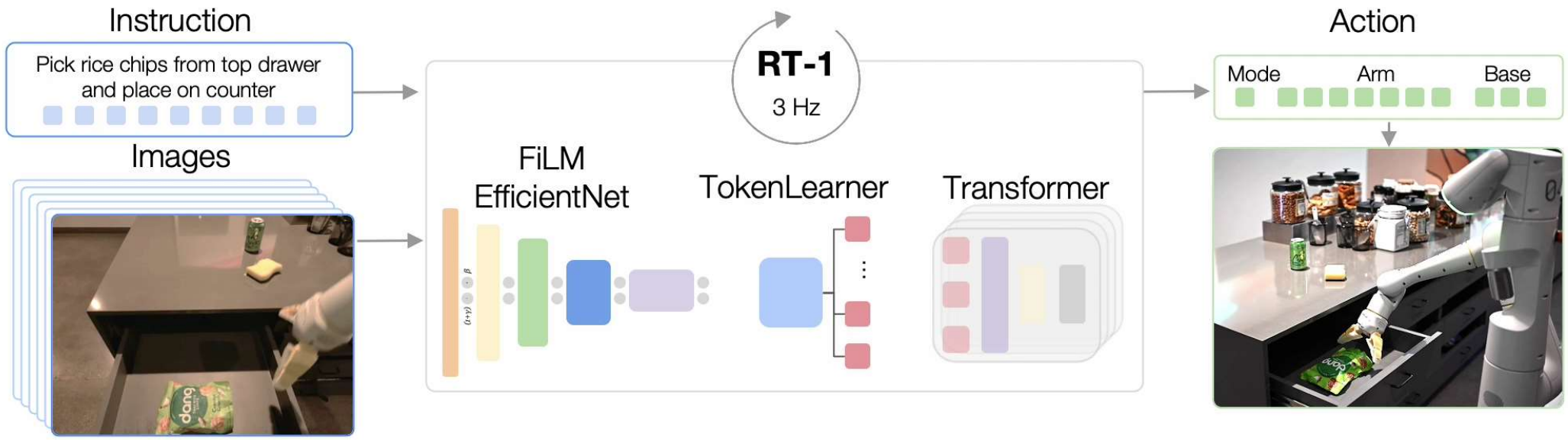
Table 6. Realworld Push-T Experiment. a) Hardware setup. b) Illustration of the task. The robot needs to ① precisely push the T-shaped block into the target region, **and** ② move the end-effector to the end-zone. c) The ground truth end state used to calculate IoU metrics used in this table. Table: Success is defined by the end-state IoU greater than the minimum IoU in the demonstration dataset. Average episode duration presented in seconds. T-E2E stands for end-to-end trained Transformer-based Diffusion Policy

What about large foundational models?

- Imitation learning problem formulation: we want to learn the best policy for a specific (usually single) task

- Foundational models: train in a self-supervised way (not for a specific task), then apply to many different tasks (with prompting or possibly some model fine-tuning)
  - Could we apply this idea to robots?
  - What kind of data would we need?

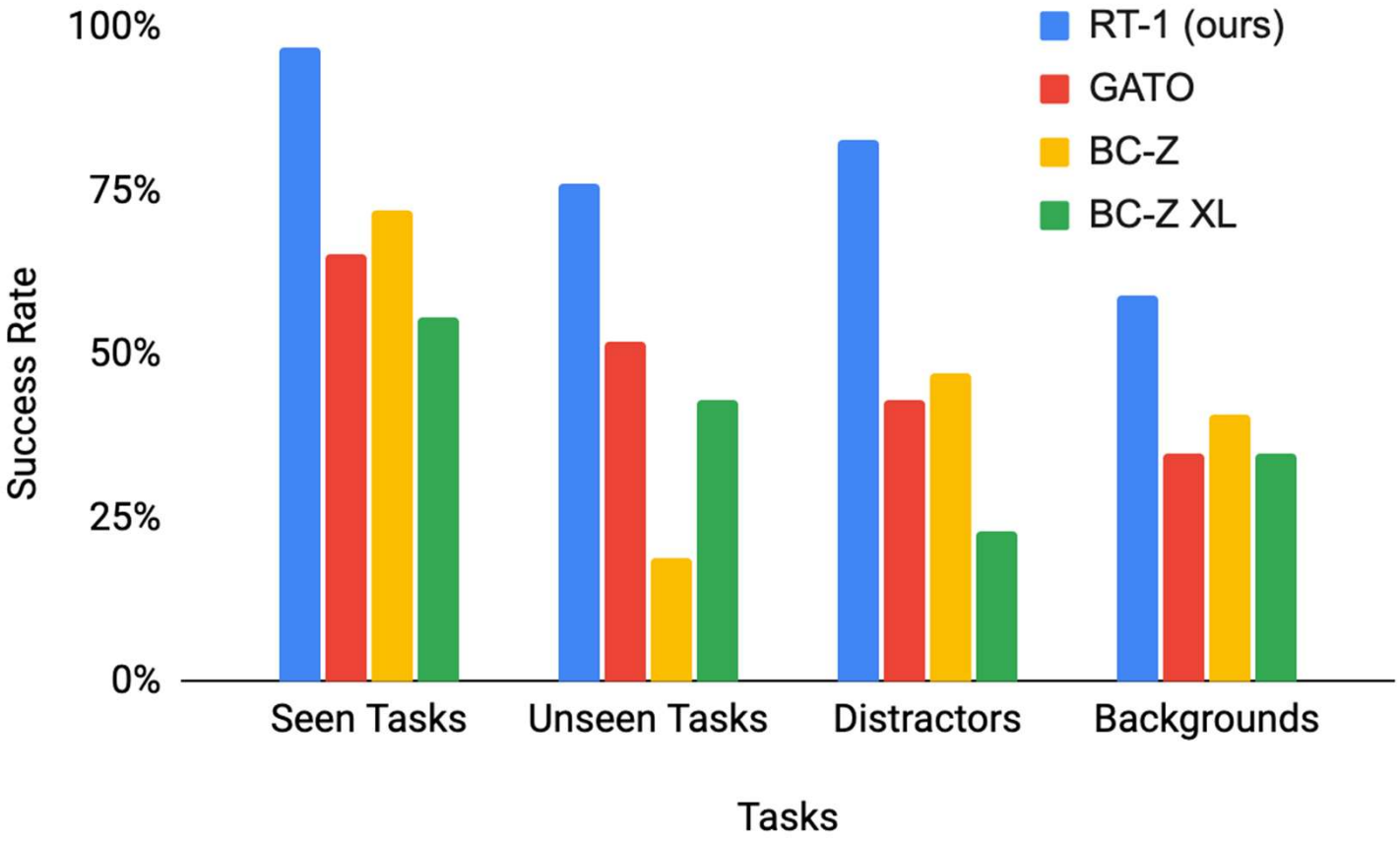
RT-1



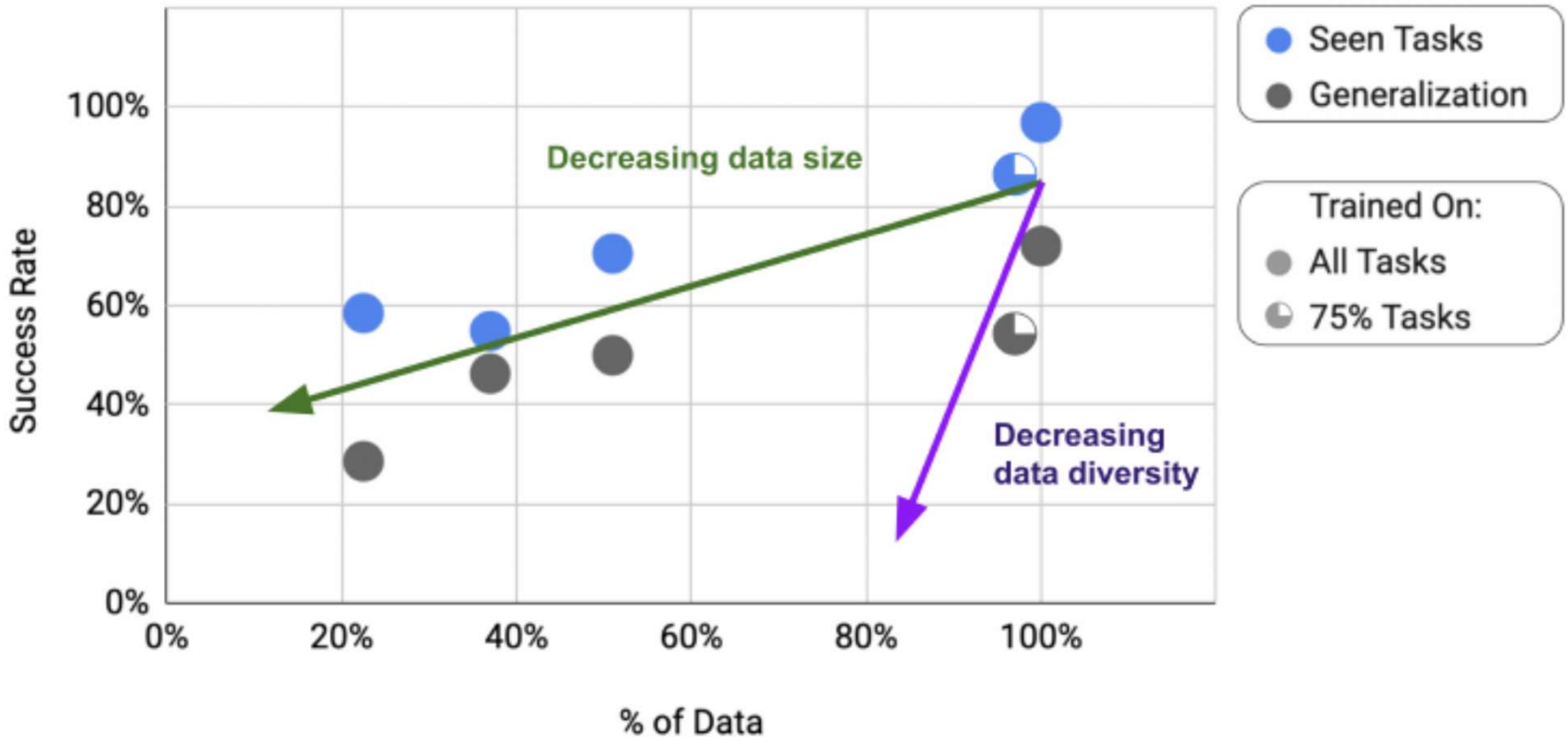
- Train using classical behaviour cloning, minimise the difference between action sequence observed in the training data and action sequence predicted by model
  - Assumes all training data are successful examples of action execution

https://robotics-transformer1.github.io/

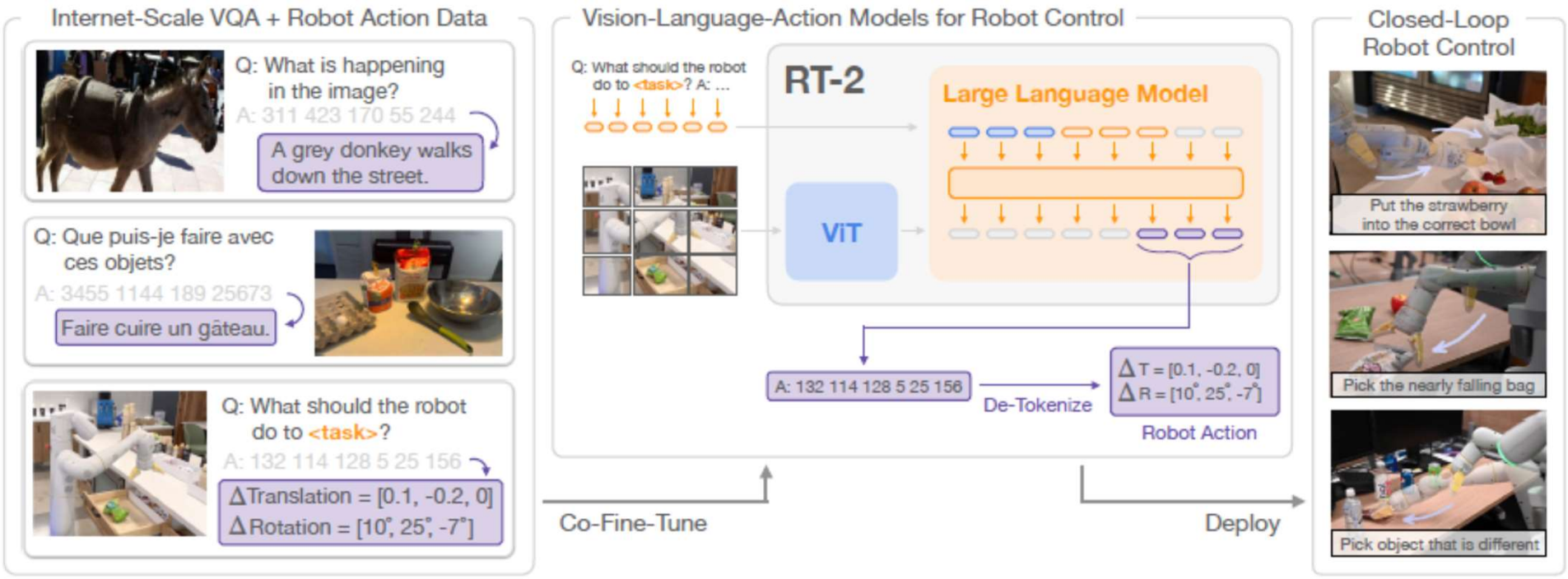
Results

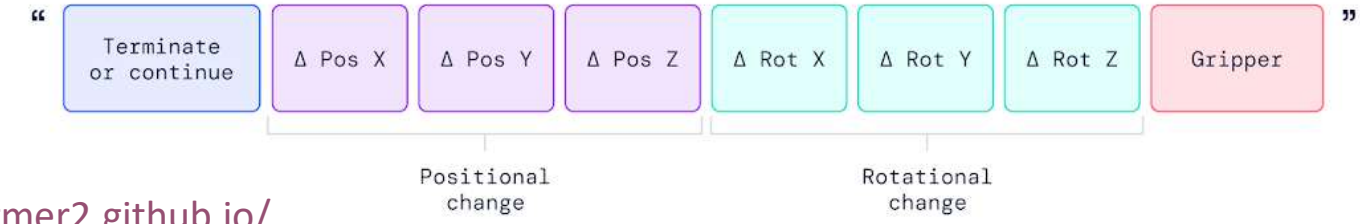

100%
75%
50%
25%
0%
Success Rate
RT-1 (ours)
GATO
BC-Z
BC-Z XL
Seen Tasks
Unseen Tasks
Distractors
Backgrounds
Tasks

How does performance vary with data?



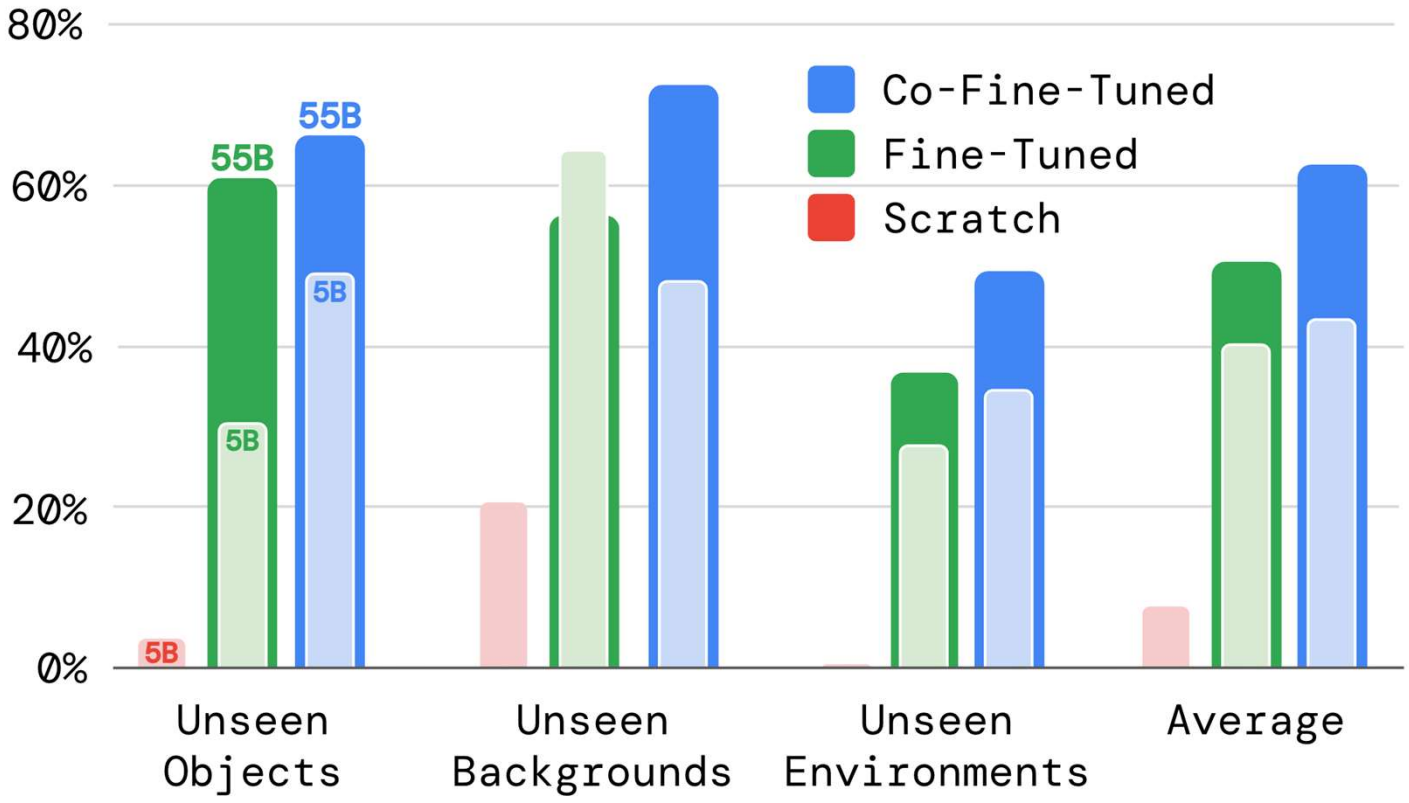
RT-2



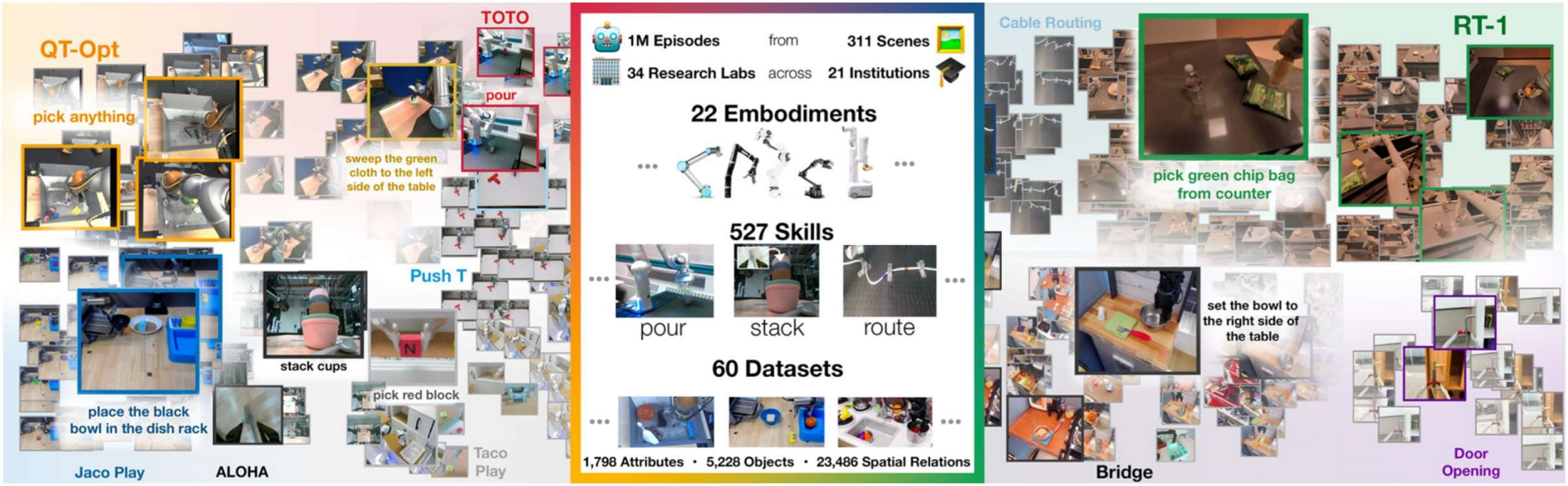


https://robotics-transformer2.github.io/

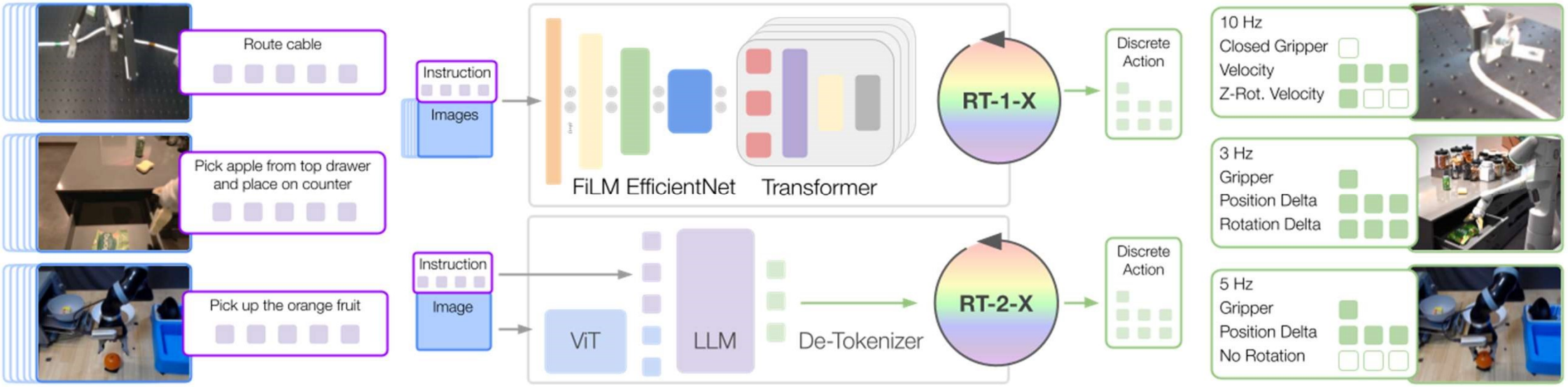
Results


80%
60%
40%
20%
0%
55B
55B
5B
5B
5B
Co-Fine-Tuned
Fine-Tuned
Scratch
Unseen Objects
Unseen Backgrounds
Unseen Environments
Average

RT-X

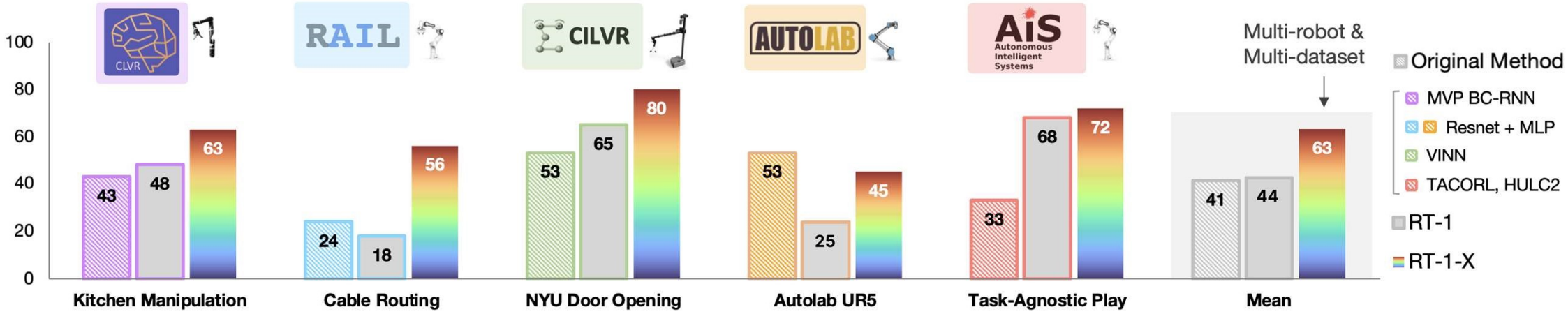

QT-Opt
pick anything
TOTO
pour
sweep the green cloth to the left side of the table
Push T
stack cups
pick red block
place the black bowl in the dish rack
Jaco Play
ALOHA
Taco Play
1M Episodes from 311 Scenes
34 Research Labs across 21 Institutions
22 Embodiments
527 Skills
pour
stack
route
60 Datasets
1,798 Attributes • 5,228 Objects • 23,486 Spatial Relations
Cable Routing
RT-1
pick green chip bag from counter
set the bowl to the right side of the table
Bridge
Door Opening

RT-X

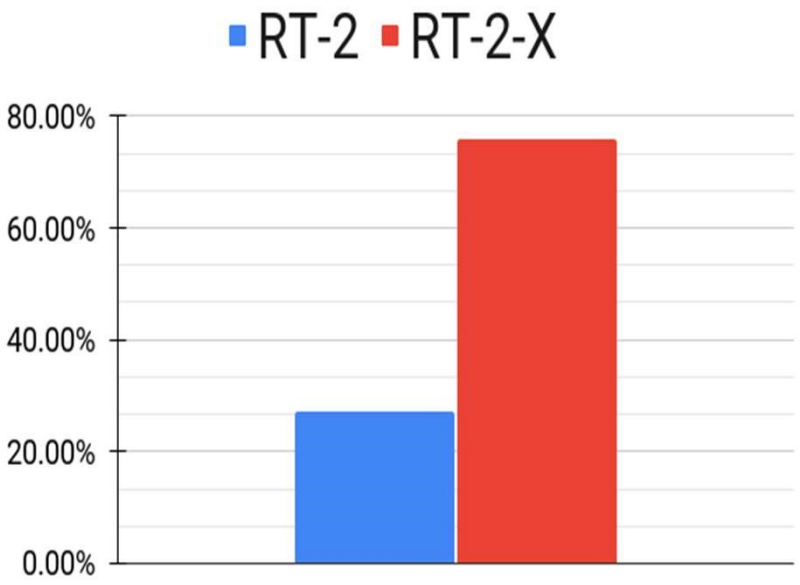


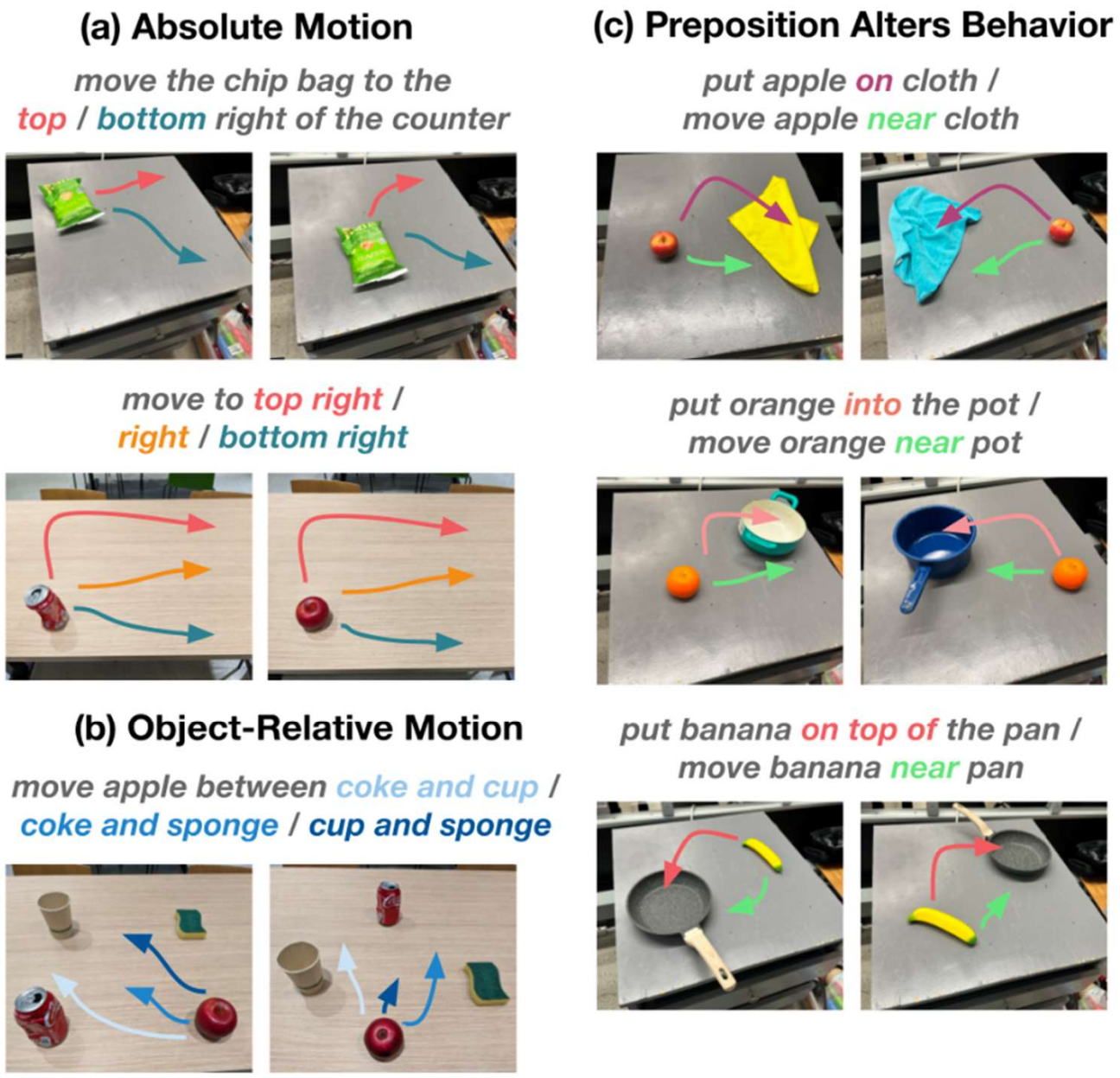
https://robotics-transformer-x.github.io/

Results


100
80
60
40
20
0
CLVR
RAIL
CILVR
AUTOLAB
AiS
Autonomous Intelligent Systems
43
48
63
24
18
56
53
65
80
53
25
45
33
68
72
41
44
63
Multi-robot & Multi-dataset
Kitchen Manipulation
Cable Routing
NYU Door Opening
Autolab UR5
Task-Agnostic Play
Mean
Original Method
MVP BC-RNN
Resnet + MLP
VINN
TACORL, HULC2
RT-1
RT-1-X

Results


RT-2
RT-2-X
80.00%
60.00%
40.00%
20.00%
0.00%


(a) Absolute Motion
move the chip bag to the top / bottom right of the counter
move to top right / right / bottom right
(b) Object-Relative Motion
move apple between coke and cup / coke and sponge / cup and sponge
(c) Preposition Alters Behavior
put apple on cloth / move apple near cloth
put orange into the pot / move orange near pot
put banana on top of the pan / move banana near pan

Behavioural Cloning –problems revisited

- How to collect the expert data?
- What is the right state representation? Does the robot see the same things as the expert does?
- Expert demonstrations may cover only a very small region of the state-space
  - For large state/action spaces, may require a huge data collection effort
- What should the robot do when it encounters a situation that wasn't seen in the dataset?
- How to handle variations in strategy?

Summary of today's session

- Improved models and network architecture for behavioural cloning improve performance:
  - More dexterous action
  - Better generalisation in perception
  - Combining visual and language prompts
  - Handling variations in strategy
- (Even more) reliant on large-scale, expensive data collection
- Challenges with generalisation, robustness, safety