Paper Reading for “A Generalist Agent”

Chengyang Ying
7 June 2022

A Generalist Agent

2022-5-13

A Generalist Agent

Scott Reed[*,†], **Konrad Żołna**[*], **Emilio Parisotto**[*], **Sergio Gómez Colmenarejo**[†], **Alexander Novikov**, **Gabriel Barth-Maron**, **Mai Giménez**, **Yury Sulsky**, **Jackie Kay**, **Jost Tobias Springenberg**, **Tom Eccles**, **Jake Bruce**, **Ali Razavi**, **Ashley Edwards**, **Nicolas Heess**, **Yutian Chen**, **Raia Hadsell**, **Oriol Vinyals**, **Mahyar Bordbar** and **Nando de Freitas**[†]

[*]Equal contributions, [†]Equal senior contributions, All authors are affiliated with DeepMind

A Generalist Agent

Highlights:

· *“Inspired by progress in large-scale language modeling, we apply a similar approach towards building a single generalist agent beyond the realm of text outputs.”*

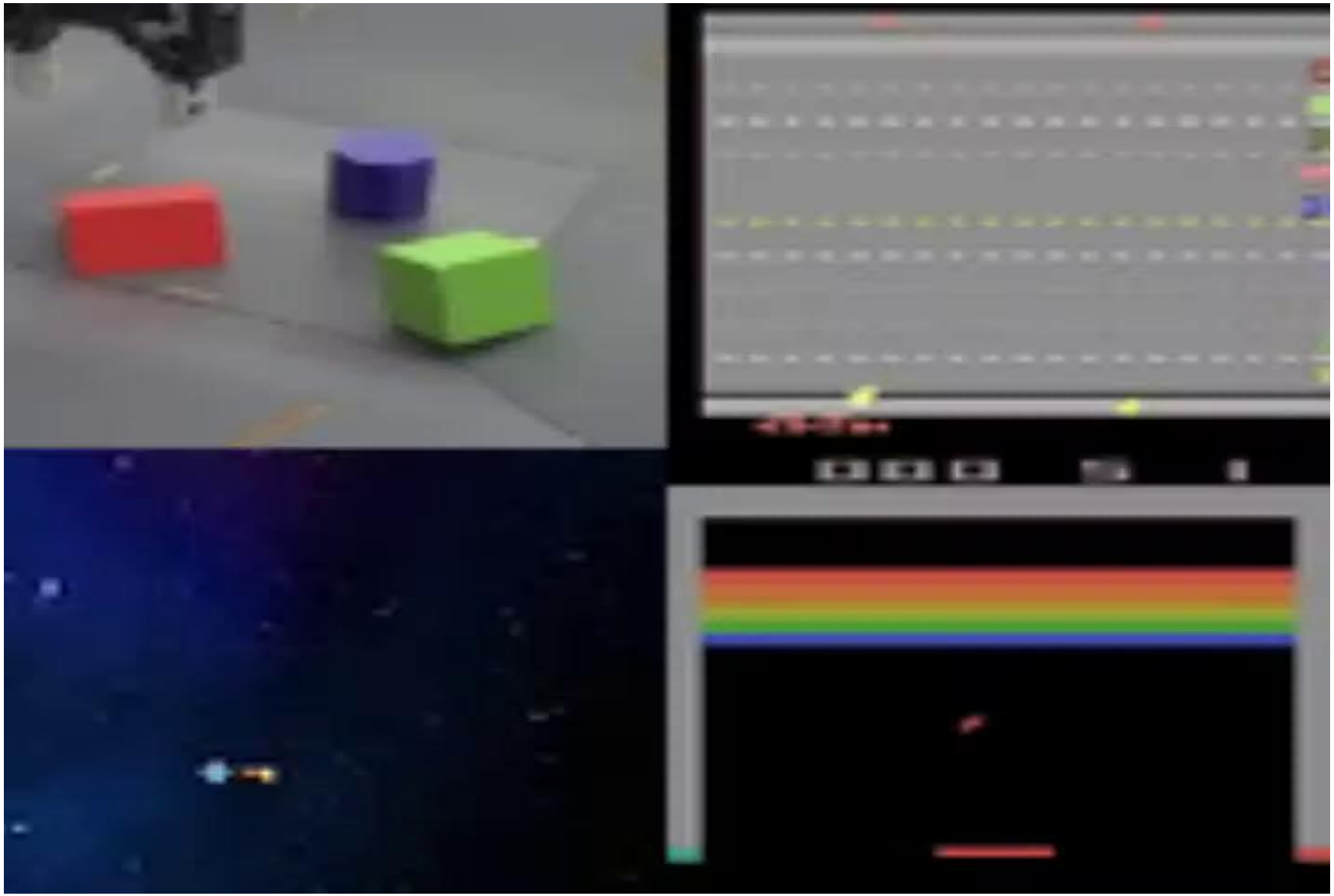
· Use a single agent with the same parameters to handle multi-modal tasks (including RL, CV, NLP)

· Parameters: 34M ~ 1.18B 1B = 1000,000,000

(As a comparison: GPT-2 ~ 1.5B, GPT-3 ~ 100B, Switch Transformer ~ 1600B, WuDao ~1750B)

· In the part of RL, Gato only focuses on supervised learning

A Generalist Agent



A Generalist Agent

Goal:
use one NN with the same parameters for 604 tasks, including:

- Control Tasks
  - Atari Games
  - DM Control
  - Meta World
- Vision and Language
  - MassiveText (text)
  - ALIGN (image-text)
- Robotics
  - RGB Stacking (real and sim)

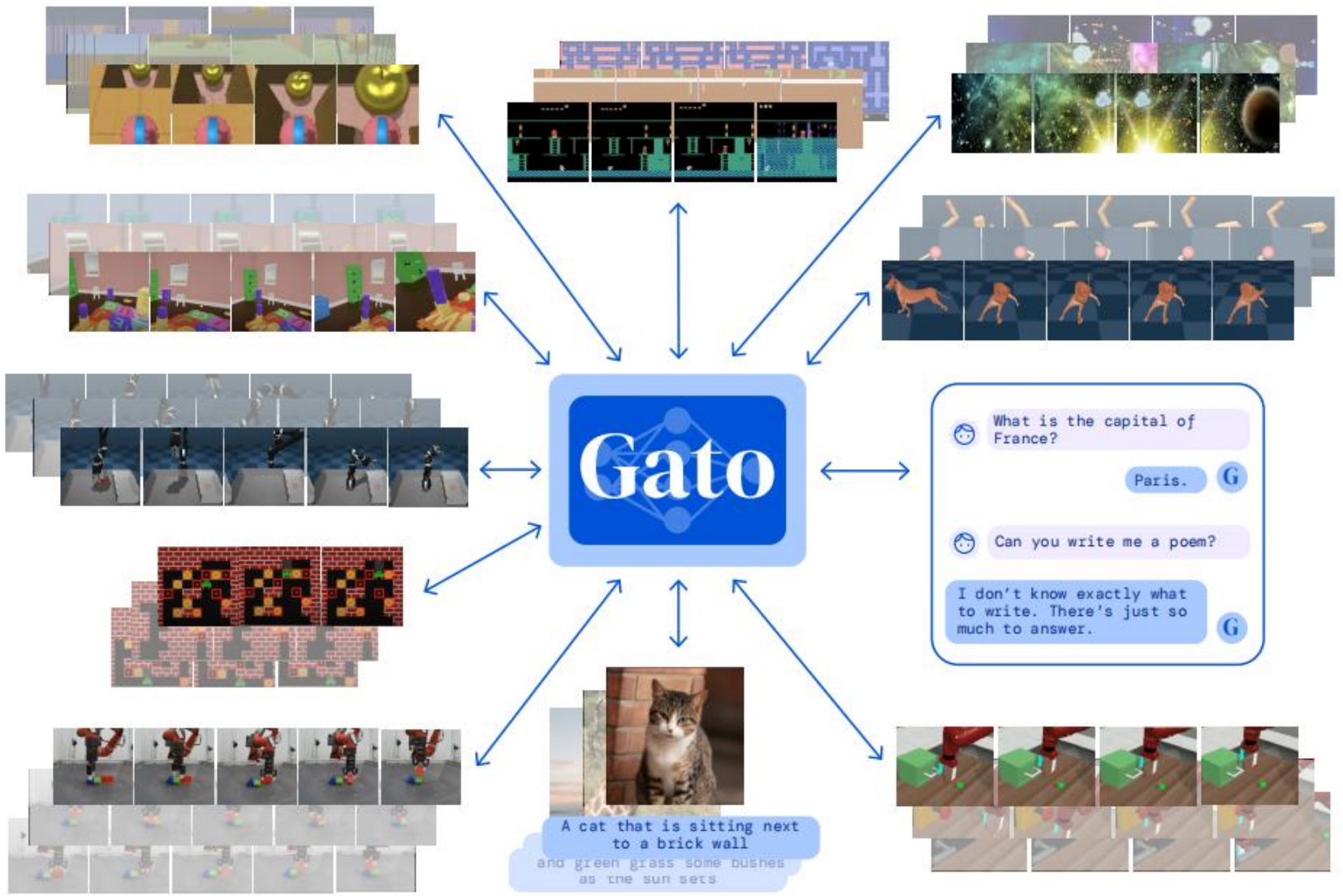


Figure 1 | **A generalist agent.** Gato can sense and act with different embodiments across a wide range of environments using a single neural network with the same set of weights. Gato was trained on 604 distinct tasks with varying modalities, observations and action specifications.

Tasks

Table 1 | **Datasets.** Left: Control datasets used to train Gato. Right: Vision & language datasets. Sample weight means the proportion of each dataset, on average, in the training sequence batches.

Control environment	Tasks	Episodes	Approx. Tokens	Sample Weight
DM Lab	254	16.4M	194B	9.35%
ALE Atari	51	63.4K	1.26B	9.5%
ALE Atari Extended	28	28.4K	565M	10.0%
Sokoban	1	27.2K	298M	1.33%
BabyAI	46	4.61M	22.8B	9.06%
DM Control Suite	30	395K	22.5B	4.62%
DM Control Suite Pixels	28	485K	35.5B	7.07%
DM Control Suite Random Small	26	10.6M	313B	3.04%
DM Control Suite Random Large	26	26.1M	791B	3.04%
Meta-World	45	94.6K	3.39B	8.96%
Procgen Benchmark	16	1.6M	4.46B	5.34%
RGB Stacking simulator	1	387K	24.4B	1.33%
RGB Stacking real robot	1	15.7K	980M	1.33%
Modular RL	38	843K	69.6B	8.23%
DM Manipulation Playground	4	286K	6.58B	1.68%
Playroom	1	829K	118B	1.33%
Total	596	63M	1.5T	85.3%

Vision / language dataset	Sample Weight
MassiveText	6.7%
M3W	4%
ALIGN	0.67%
MS-COCO Captions	0.67%
Conceptual Captions	0.67%
LTIP	0.67%
OKVQA	0.67%
VQAV2	0.67%
Total	14.7%

A Generalist Agent

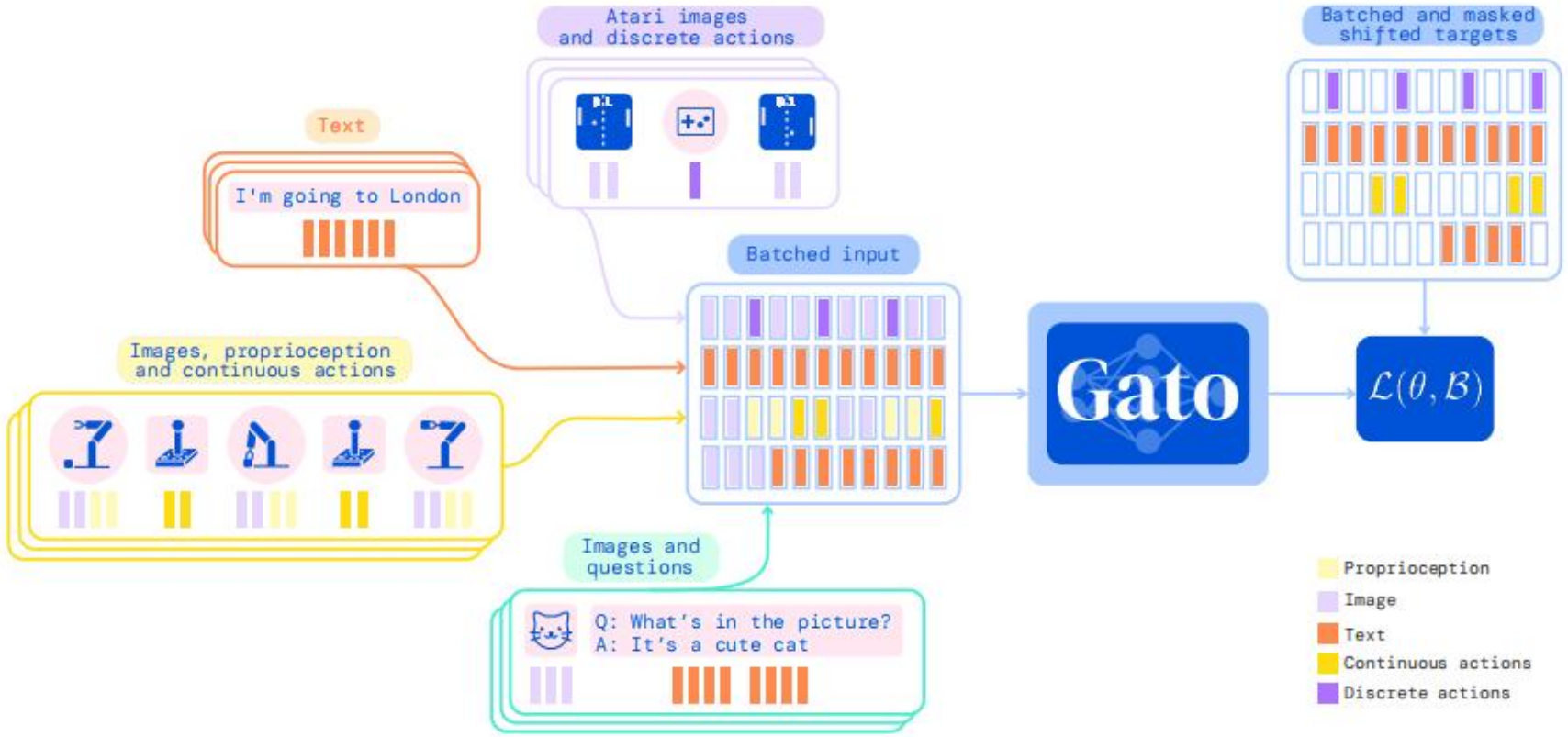


Figure 2 | **Training phase of Gato**. Data from different tasks and modalities is serialized into a flat sequence of tokens, batched, and processed by a transformer neural network akin to a large language model. Masking is used such that the loss function is applied only to target outputs, i.e. text and various actions.

Data Process

We serialize all data into a flat sequence of tokens

· Text: SentencePiece
· Images: ViT

· Discrete Values, e.g. Atari button presses: flattened into sequences of integers
· Continuous Values, e.g. proprioceptive inputs or joint torques: original value -> [-1, 1] -> discretized to 1024 uniform bins.

Loss Function

Given a sequence of tokens $s_{1:L}$ and parameters θ, we model the data using the chain rule of probability:

$$\log p_\theta(s_1, \ldots, s_L) = \sum_{l=1}^{L} \log p_\theta(s_l | s_1, \ldots, s_{l-1}), \tag{1}$$

Let b index a training batch of sequences $\mathcal{B}$. We define a masking function m such that $m(b, t) = 1$ if the token at index t is either from text or from the logged action of an agent, and 0 otherwise. The training loss for a batch $\mathcal{B}$ can then be written as

$$\mathcal{L}(\theta, \mathcal{B}) = -\sum_{b=1}^{|\mathcal{B}|} \sum_{l=1}^{L} m\,(b, t) \log p_\theta \left(s_l^{(b)} | s_1^{(b)}, \ldots, s_{l-1}^{(b)} \right) \tag{2}$$

· m(b,t) = 1 iff s_l^{(b)} is output

Deployment

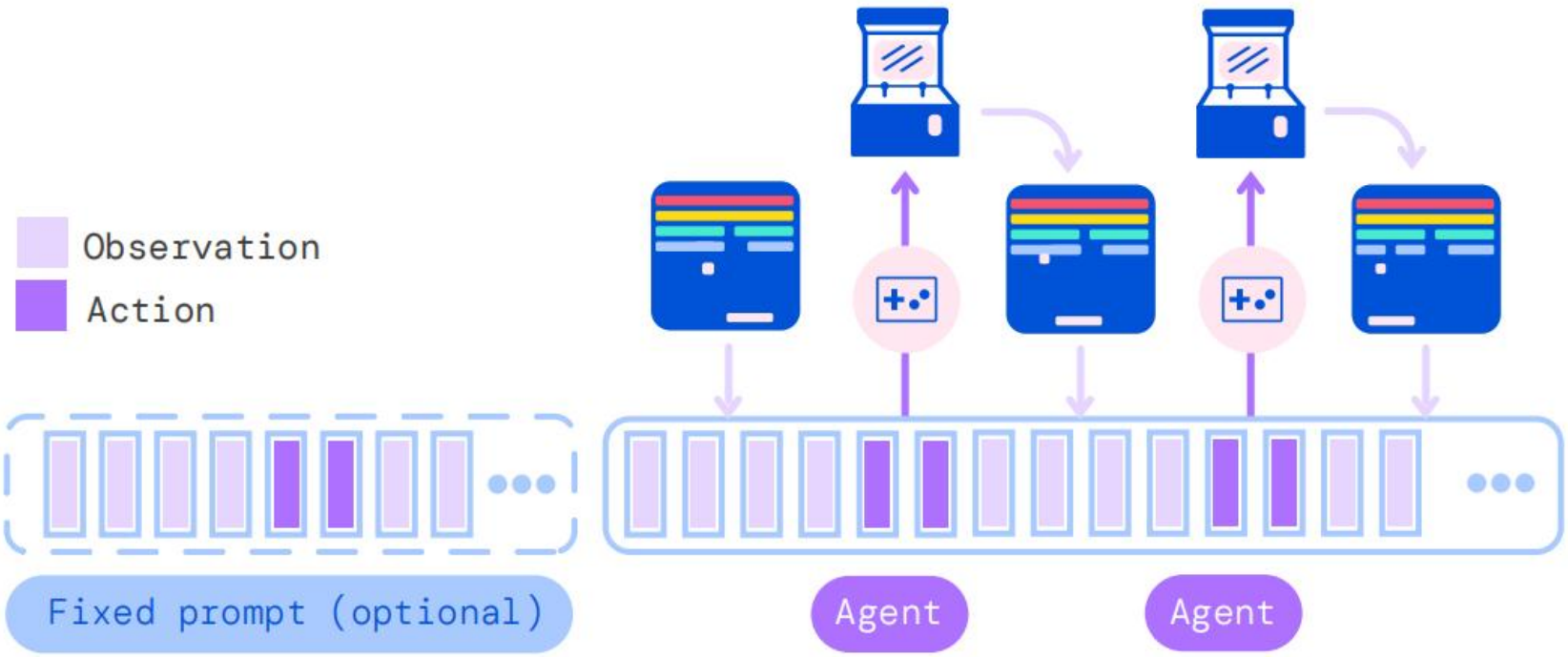


Figure 3 | **Running Gato as a control policy.** Gato consumes a sequence of interleaved tokenized observations, separator tokens, and previously sampled actions to produce the next action in standard autoregressive manner. The new action is applied to the environment – a game console in this illustration,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

Results: Simulated control tasks > 450 for 50%

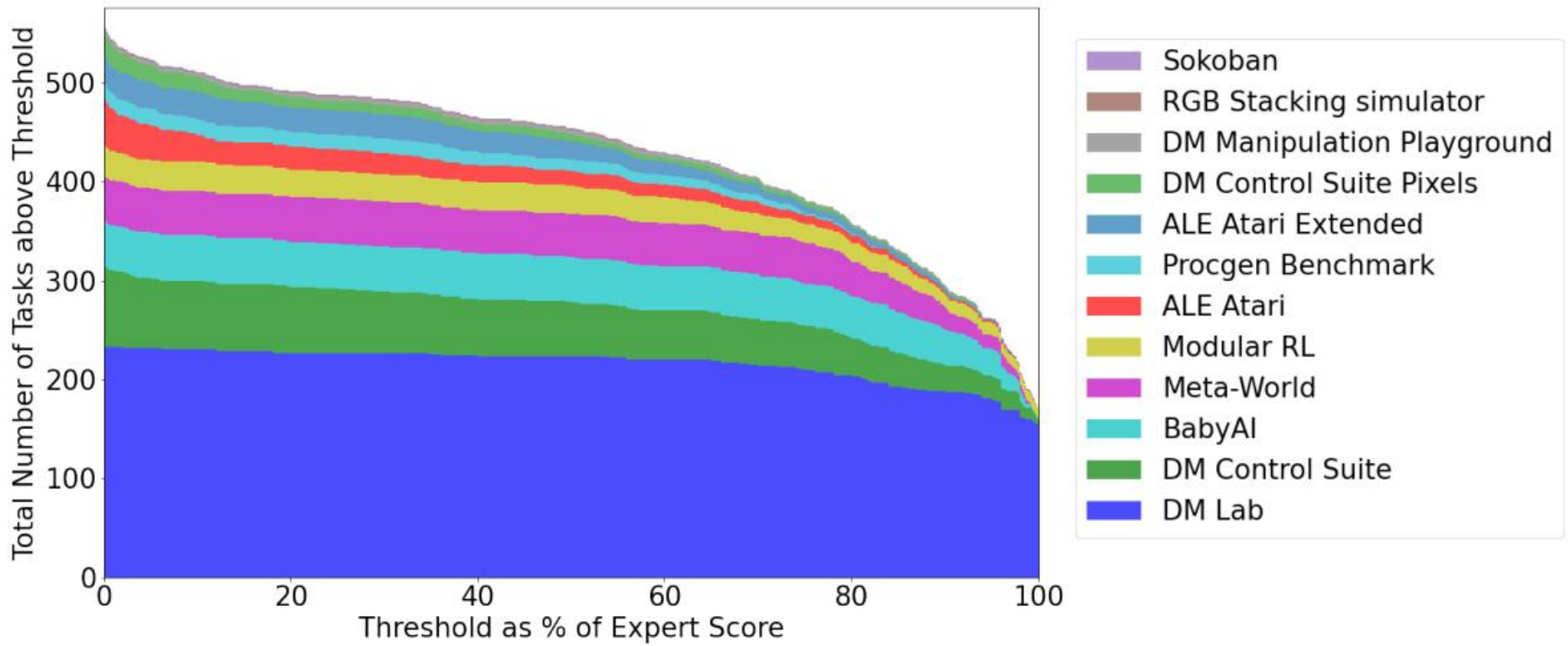


Figure 5 | **Gato's performance on simulated control tasks.** Number of tasks where the performance of the pretrained model is above a percentage of expert score, grouped by domain. Here values on the x-axis represent a specific percentage of expert score, where 0 corresponds to random agent performance. The y-axis is the number of tasks where the pretrained model's mean performance is equal to or above that percentage. That is, the width of each colour band indicates the number of tasks where Gato's mean performance is above a percentage of the maximum score obtained by a task-specific expert.

Results: Robotics

Table 2 | **Gato real robot Skill Generalization results.** In addition to performing hundreds of other tasks, Gato also stacks competitively with the comparable published baseline.

Agent	Group 1	Group 2	Group 3	Group 4	Group 5	Average
Gato	**24.5%**	33%	**50.5%**	76.5%	**66.5%**	**50.2%**
BC-IMP (Lee et al., 2021)	23%	**39.3%**	39.3%	**77.5%**	66%	49%

Skill generalization:

Test in five triplets of object shapes are not included in the training data

Results: Text samples

The colorful ceramic toys are on the living room floor.

a living room with three different color deposits on the floor

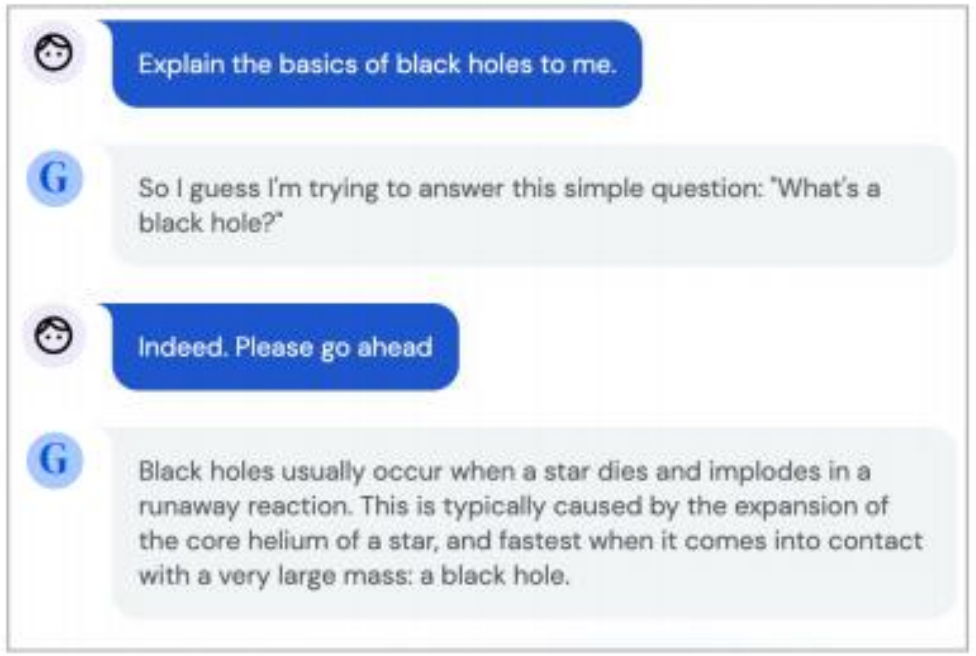
a room with a long red rug a tv and some pictures

Man standing in the street wearing a suit and tie.

A man in a blue suit with a white bow tie and black shoes.

A man with a hat in his hand looking at the camera

Image captions



Chitchat

Analysis: model size

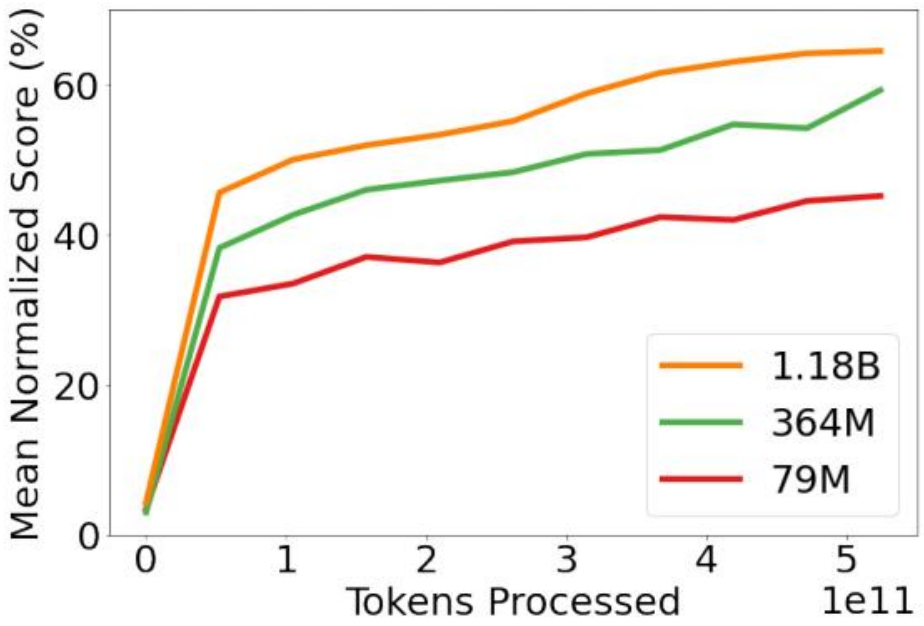


Figure 8 | **Model size scaling laws results.** In-distribution performance as a function of tokens processed for 3 model scales. Performance is first mean-aggregated within each separate control domain, and then mean-aggregated across all domains. We can see a consistent improvement as model capacity is increased for a fixed number of tokens.

Analysis: OOD tasks

1. A model pretrained only on data from the same domain as the task to be fine-tuned on, *same domain only data*.
2. A model pretrained only on non-control data, *no control data*.
3. A model fine-tuned from scratch, i.e. no pretraining at all, *scratch*.

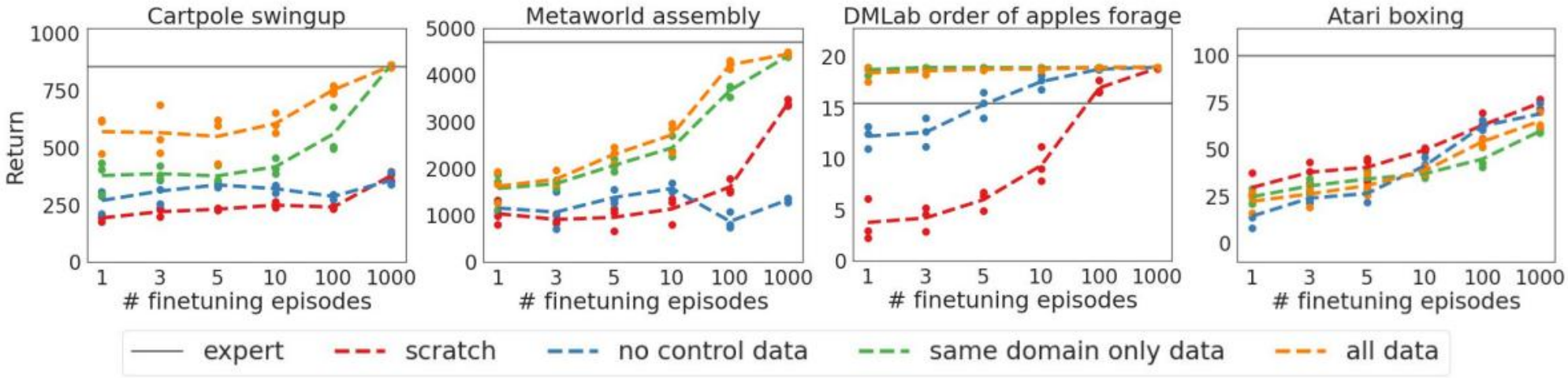


Figure 9 | **Few-shot performance, ablating over various pretraining settings.** Orange corresponds to the base Gato pretrained on all data. Red is trained from scratch only on the few-shot data. 364M parameter variants of Gato were used for this experiment to save compute.

Analysis: Fine-tuning on Robotic Stacking Tasks

CRR: a baseline

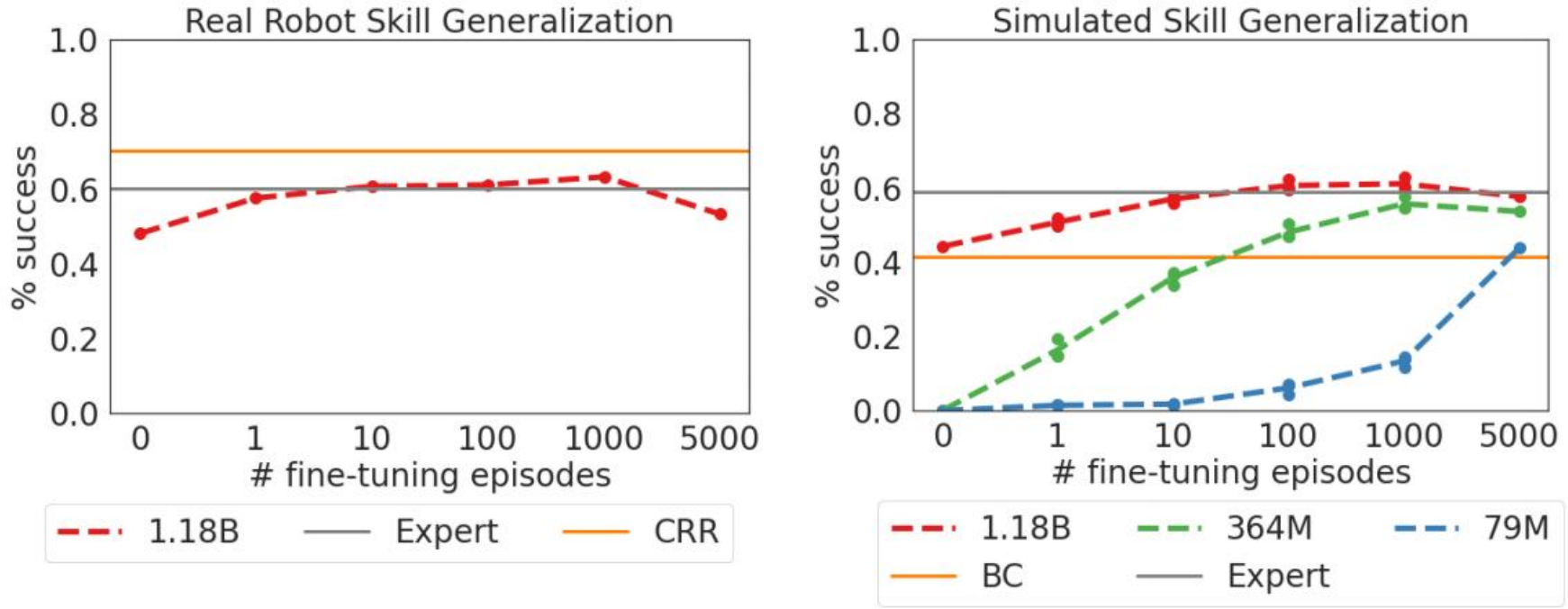


Figure 10 | **Robotics fine-tuning results.** Left: Comparison of real robot Skill Generalization success rate averaged across test triplets for Gato, expert, and CRR trained on 35k expert episodes (upper bound). Right: Comparison of simulated robot Skill Generalization success rate averaged across test triplets for a series of ablations on the number of parameters, including scores for expert and a BC baseline trained on 5k episodes.

Analysis: Skill Mastery

Table 3 | **Real robot Skill Mastery results.** Gato is competitive with the filtered BC baseline.

Agent	Group 1	Group 2	Group 3	Group 4	Group 5	Average
Gato	58%	57.6%	**78.5%**	**89 %**	**95.1%**	**75.6%**
BC-IMP (Lee et al., 2021)	**75.6%**	**60.8%**	70.8%	87.8%	78.3%	74.6%

Skill mastery:

In-distribution tasks, no fine-tuning

Analysis: Specialist single-domain multi-task agents

train on data from a single domain only and rolled out 500 times for each training task without any per-task fine-tuning

Meta-World:
· 79M parameters, 50 tasks
· achieves 96.6% success rate averaged over all 50 task

Atari:
· 1.18B parameters, 51 tasks
· achieved better than human performance for 44 games (Gato: 23)
(the performance of online experts used to generate training data for the other 7 games were also below the average human)

A Generalist Agent

Highlights:

· *“Inspired by progress in large-scale language modeling, we apply a similar approach towards building a single generalist agent beyond the realm of text outputs.”*

· Use a single agent with the same parameters to handle multi-modal tasks (including RL, CV, NLP)

· Parameters: 34M ~ 1.18B 1B = 1000,000,000

(As a comparison: GPT-2 ~ 1.5B, GPT-3 ~ 100B, Switch Transformer ~ 1600B, WuDao ~1750B)

· In the part of RL, Gato only focuses on supervised learning

Thanks for Listening

Questions?