A Survey on Transformers in Reinforcement Learning

Wenzhe Li[1*] **Hao Luo**[2,3*] **Zichuan Lin**[4*] **Chongjie Zhang**[1†] **Zongqing Lu**[2,3†] **Deheng Ye**[4†]

[1] Tsinghua University [2] Peking University [3] BAAI [4] Tencent Inc.

lwz21@mails.tsinghua.edu.cn; lh2000@pku.edu.cn; zichuanlin@tencent.com;
chongjie@tsinghua.edu.cn; zongqing.lu@pku.edu.cn; dericye@tencent.com

Abstract

Transformer has been considered the dominating neural architecture in NLP and CV, mostly under a supervised setting. Recently, a similar surge of using Transformers has appeared in the domain of reinforcement learning (RL), but it is faced with unique design choices and challenges brought by the nature of RL. However, the evolution of Transformers in RL has not yet been well unraveled. Hence, in this paper, we seek to systematically review motivations and progress on using Transformers in RL, provide a taxonomy on existing works, discuss each sub-field, and summarize future prospects.

1 Introduction

Reinforcement learning (RL) provides a mathematical formalism for sequential decision-making. By utilizing RL, we can acquire intelligent behaviors automatically. While RL has provided a general framework for learning-based control, the introduction of deep neural networks, as a way of function approximation with high capacity, is enabling significant progress along a wide range of domains [Silver *et al.*, 2016; Vinyals *et al.*, 2019; Ye *et al.*, 2020a,b].

While the generality of deep reinforcement learning (DRL) has achieved tremendous developments in recent years, the issue of sample efficiency prevents its widespread use in real-world applications. To address this issue, an effective mechanism is to introduce inductive bias into the DRL framework. One important inductive bias in DRL is *the choice of function approximator architectures*, such as the parameterization of neural networks for DRL agents. However, the problem of choosing architectural designs in DRL has remained less explored, when compared to efforts on architectural designs in supervised learning (SL). Most existing works on architecture for RL are motivated by the success of the (semi-)supervised learning community. For instance, a common practice to deal with high-dimensional image-based input in DRL is to introduce convolutional neural networks (CNN) [LeCun *et al.*, 1998; Mnih *et al.*, 2015]; another common practice to deal with partial observability is to introduce recurrent neural networks (RNN) [Hochreiter and Schmidhuber, 1997; Hausknecht and Stone, 2015].

In recent years, the Transformer architecture [Vaswani *et al.*, 2017] has revolutionized the learning paradigm across a wide range of SL tasks [Devlin *et al.*, 2018; Dosovitskiy *et al.*, 2020; Dong *et al.*, 2018] and demonstrated superior performance over CNN and RNN. Among its notable benefits, the Transformer architecture enables modeling long dependencies and has excellent scalability [Khan *et al.*, 2022]. Inspired by the success of SL, there has been a surge of interest in applying Transformers in reinforcement learning, with the hope of carrying the benefits of Transformers to the RL field.

The use of Transformers in RL dates back to Zambaldi *et al.* [2018b], where the self-attention mechanism is used for relational reasoning over structured state representations. Afterward, many researchers seek to apply self-attention for representation learning to extract relations between entities for better policy learning [Vinyals *et al.*, 2019; Baker *et al.*, 2019]. Besides leveraging Transformers for state representation learning, prior works also use Transformers to capture multi-step temporal dependencies to deal with the issue of partial observability [Parisotto *et al.*, 2020; Parisotto and Salakhutdinov, 2021]. More recently, offline RL [Levine *et al.*, 2020] has attracted attention due to its ability to leverage offline large-scale datasets. Motivated by offline RL, recent efforts have shown that the Transformer architecture can serve directly as a model for sequential decisions [Chen *et al.*, 2021; Janner *et al.*, 2021] and generalize to multiple tasks and domains [Lee *et al.*, 2022; Carroll *et al.*, 2022].

The purpose of this survey is to present the field of *Transformers in Reinforcement Learning*, denoted as TransformRL. Although Transformer has been considered as a foundation model in most SL research at present [Devlin *et al.*, 2018; Dosovitskiy *et al.*, 2020], it remains to be less explored in the RL community. In fact, compared with the SL domain, using Transformers in RL as function approximators faces unique challenges. First, the training data of RL agents is typically a function of the current policy, which induces non-stationarity during learning a Transformer. Second, existing RL algorithms are often highly sensitive to design choices in the training process, including network architecture and capacity [Henderson *et al.*, 2018]. Third, Transformer-based architectures often suffer from high computational and memory costs, making it expensive in both

[*] Equal contribution; † Equal advising.

training and inference during the RL learning process. For example, in the case of AI for video game-playing, the efficiency of sample generation, which largely affects the training performance, depends on the computational cost of the RL policy network and value network [Ye *et al.*, 2020a; Berner *et al.*, 2019]. In this paper, we seek to provide a comprehensive overview of TransformRL, including a taxonomy of current methods and the challenges. We also discuss future perspectives, as we believe the field of TransformRL will play an important role in unleashing the potential impact of reinforcement learning, and this survey could provide a starting point for those looking to leverage its potential.

We structure the paper as follows. Section 2 covers background on RL and Transformers, followed by a brief introduction on how these two are combined together. In Section 3, we describe the evolution of network architecture in RL and the challenges that prevent the Transformer architecture from being widely explored in RL for a long time. In Section 4, we provide a taxonomy of Transformers in RL and discuss representative existing methods. Finally, we summarize and point out potential future directions in Section 5.

2 Problem Scope

2.1 Reinforcement Learning

In general, Reinforcement Learning (RL) considers learning in a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0 \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state space and action space respectively, $P(s'|s,a)$ is the transition dynamics, $r(s,a)$ is the reward function, $\gamma \in (0,1)$ is the discount factor, and ρ_0 is the distribution of initial states. Typically, RL aims to learn a policy $\pi(a|s)$ to maximize the expected discounted return $J(\pi) = \mathbb{E}_{\pi,P,\rho_0}\left[\sum_t r(s_t,a_t)\right]$. There are many important topics in this area, for instance, meta RL, multi-task RL, and multi-agent RL. In the following part, we introduce several specific RL problems that are closely related to advances in Transformers in RL.

Offline RL. In offline RL [Levine *et al.*, 2020], the agent is not allowed to interact with the environment during training. Instead, it only has access to a static offline dataset $\mathcal{D} = \{(s,a,s',r)\}$ collected by arbitrary policies. Without exploration, modern offline RL approaches [Fujimoto *et al.*, 2019; Kumar *et al.*, 2020; Yu *et al.*, 2021b] constrain the learned policy close to the data distribution, to avoid out-of-distribution actions that may lead to overestimation. Recently, in parallel with typical value-based methods, one popular trend in offline RL is RL via Supervised Learning (RvS) [Emmons *et al.*, 2021], which learns an outcome-conditioned policy to yield desired behavior via SL.

Goal-conditioned RL. Goal-conditioned RL (GCRL) extends the standard RL problem to goal-augmented setting, where the agent aims to learn a goal-conditioned policy $\pi(a|s,g)$ that can reach multiple goals. Prior works propose to use various techniques, such as hindsight relabeling [Andrychowicz *et al.*, 2017], universal value function [Schaul *et al.*, 2015], and self-imitation learning [Ghosh *et al.*, 2019], to improve the generalization and sample efficiency of GCRL. GCRL is quite flexible as there are diverse choices of goals. We refer readers to [Liu *et al.*, 2022] for a detailed discussion around this topic.

Model-based RL. In contrast to model-free RL which directly learns the policy and value functions, model-based RL learns an auxiliary dynamic model of the environment. Such a model can be directly used for planning algorithms [Schrittwieser *et al.*, 2020], or it can be used as a generator to produce imaginary trajectories and enlarge the training data for any model-free algorithm [Hafner *et al.*, 2019]. Learning a model is non-trivial, especially in large or partially observed environments where we first need to construct the representation of the state. Some recent methods propose to use latent dynamics [Hafner *et al.*, 2019] or value models [Schrittwieser *et al.*, 2020] to address these challenges and improve the sample efficiency of RL.

2.2 Transformers

Transformer [Vaswani *et al.*, 2017] is one of the most effective and scalable neural networks to model sequential data. The key idea of Transformers is to incorporate *self-attention* mechanism, which could capture dependencies within long sequences in an efficient manner. Formally, given a sequential input with n tokens $\left\{\mathbf{x}_i \in \mathbb{R}^d\right\}_{i=1}^n$, where d is the embedding dimension, the self-attention layer maps each token $\mathbf{x}_i$ to a query $\mathbf{q}_i \in \mathbb{R}^{d_q}$, a key $\mathbf{k}_i \in \mathbb{R}^{d_k}$, and a value $\mathbf{v}_i \in \mathbb{R}^{d_v}$ via linear transformations, where $d_q = d_k$. Denote the sequence of inputs, queries, keys, and values as $\mathbf{X} \in \mathbb{R}^{n\times d}, \mathbf{Q} \in \mathbb{R}^{n\times d_q}, \mathbf{K} \in \mathbb{R}^{n\times d_k}$, and $\mathbf{V} \in \mathbb{R}^{n\times d_v}$, respectively. The output of the self-attention layer $\mathbf{Z} \in \mathbb{R}^{n\times d_v}$ is a weighted sum of all values:

$$\mathbf{Z} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_q}}\right)\mathbf{V}.$$

With the self-attention mechanism as well as other techniques, such as multi-head attention and residual connection, Transformers can learn expressive representations and model long-term interactions.

2.3 Combination of Transformers and RL

We notice that a growing number of works are seeking to combine Transformers and RL in diverse ways. In general, Transformers can be used as one component for RL algorithms, e.g., a representation module or a dynamic model. Transformers can also serve as one whole sequential decision-maker. Figure 1 provides a sketch of Transformers' different roles in the context of RL.

3 Network Architecture in RL

Before presenting the taxonomy of current methods in TransformRL, we start by reviewing the early progress of network architecture design in RL, and summarize their challenges. We do this because Transformer itself is an advanced neural network and designing appropriate neural networks contributes to the success of DRL.

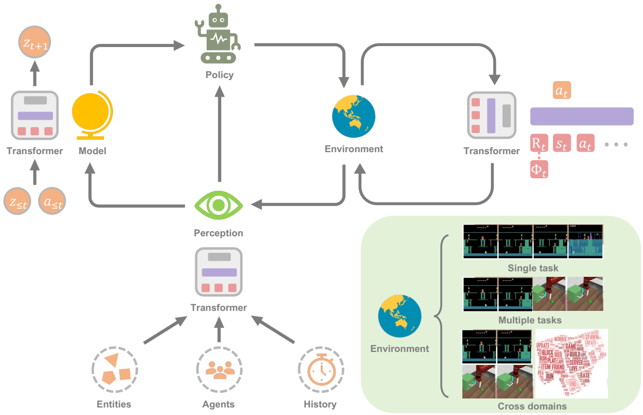

Figure 1: An illustrating example of TransformRL. On the one hand, Transformers can be used as one component in RL. Particularly, Transformers can encode diverse sequences, such as entities, agents, and stacks of historical information; and it is also an expressive predictor for the dynamics model. On the other hand, Transformers can integrate all subroutines in RL and act as a sequential decision-maker. Overall, Transformers can improve RL's learning efficiency in single-task, multi-task, and cross-domain settings.

3.1 Architectures for function approximators

Since the seminal work Deep Q-Network [Mnih *et al.*, 2015], many efforts have been made in developing network architectures for DRL agents. Improvements in network architectures in RL can be mainly categorized into two classes. The first class is to design a new structure that incorporates RL inductive bias to ease the difficulty of training policy or value functions. For example, Wang *et al.* [2016] propose dueling network architecture with one for the state value function and another for the state-dependent action advantage function. This choice of architecture incorporates inductive bias that generalizes learning across actions. Other examples include the value decomposition network which has been used to learn local Q-values for individual agent [Sunehag *et al.*, 2017] or sub-reward [Lin *et al.*, 2019]. The second class is to investigate whether general techniques of neural networks (e.g., regularization, skip connection, batch normalization) can be applied to RL. To name a few, Ota *et al.* [2020] find that increasing input dimensionality while using an online feature extractor to boost state representation helps improve the performance and sample efficiency of DRL algorithms. Sinha *et al.* [2020] propose a deep dense architecture for DRL agents, using skip connections for efficient learning, with an inductive bias to mitigate data-processing inequality. Ota *et al.* [2021] use DenseNet [Huang *et al.*, 2017] with decoupled representation learning to improve flows of information and gradients for large networks. Recently, due to the superior performance of Transformers, some researchers have attempted to apply Transformers architecture in policy optimization algorithms, but found that the vanilla Transformer design fails to achieve reasonable performance in RL tasks [Parisotto *et al.*, 2020].

3.2 Challenges

While Transformer-based architectures have made rapid progress in SL domains in past years, applying them in RL is not straightforward. Actually, there exist several unique challenges.

On the one hand, from the view of RL, many researchers point out that existing RL algorithms are incredibly sensitive to architectures of deep neural networks [Henderson *et al.*, 2018; Engstrom *et al.*, 2019; Andrychowicz *et al.*, 2020]. First, the paradigm of alternating between data collection and policy optimization (i.e., data distribution shift) in RL induces non-stationarity during training. Second, RL algorithms are often highly sensitive to design choices in the training process. In particular, when coupled with bootstrapping and off-policy learning, learning with function approximations can diverge when the value estimates become unbounded (i.e., "deadly triad") [Van Hasselt *et al.*, 2018]. More recently, Emmons *et al.* [2021] identify that carefully choosing model architecture and regularization are crucial for the performance of DRL agents.

On the other hand, from the view of Transformers, the Transformer-based architectures suffer from large memory footprints and high latency which hinder their efficient deployment and inference. Recently, many researchers aim to make improvements around computational and memory efficiency upon the original Transformer architecture [Tay *et al.*, 2022], but most of these works focus on SL domains. In the context of RL, Parisotto and Salakhutdinov [2021] propose to distill learning progress from a large capacity Transformer-based learner model to a small capacity actor model to bypass the high inference latency of Transformers. However, these methods are still expensive in terms of memory and computation. So far, the idea of efficient or lightweight Transformers has not yet been fully explored in the RL community.

4 Transformers in RL

Although Transformer has become a foundation model in most supervised learning research, it has not been widely used in the RL community for a long time due to the aforementioned challenges. Actually, most early attempts of TransformRL apply Transformers for state representation learning or providing memory information while still applying the standard RL algorithms for agent learning such as temporal difference learning and policy optimization.

Therefore, although introducing Transformers as function approximators, these methods still suffer from challenges from the conventional RL framework. Until recently, offline RL makes it possible to learn optimal policy from large-scale offline data. Inspired by offline RL, recent works further treat the RL problem as a conditional sequence modeling problem on fixed experiences. By doing so, it helps to bypass the challenges of bootstrapping error in traditional RL, consequently enabling the Transformer architecture to unleash its powerful sequential modeling ability.

In this survey paper, we retrospect the advances of TransformRL, and provide a taxonomy to present the current methods. We categorize existing methods into four classes: representation learning, model learning, sequential decision-making, and generalist agents. Figure 2 provides a taxonomy sketch with a subset of corresponding works.

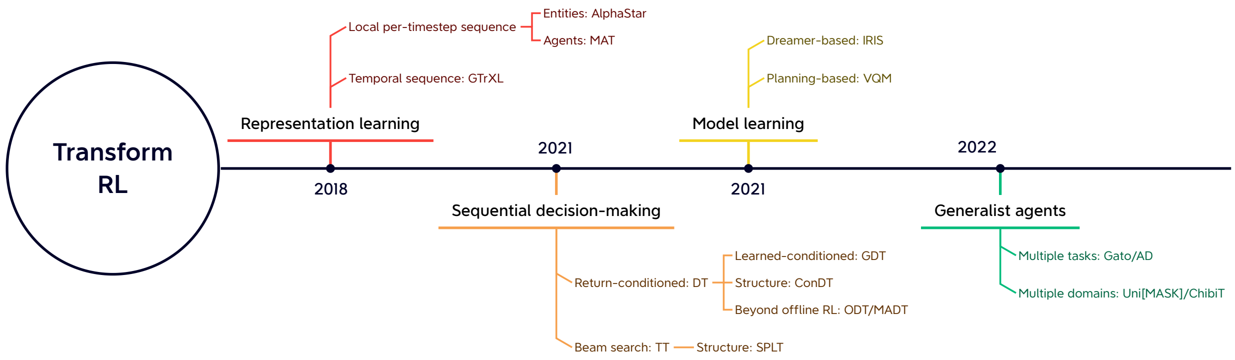

Figure 2: The taxonomy of TransformRL. The timeline is based on the first work related to the branch.

4.1 Transformers for representation learning

Considering the sequential nature of RL tasks, it is reasonable to try out a Transformer encoder module. In fact, various sequences in RL tasks require processing, such as local per-timestep sequence (multi-entity sequence [Vinyals *et al.*, 2019; Baker *et al.*, 2019], multi-agent sequence [Wen *et al.*, 2022]), temporal sequence (trajectory [Parisotto *et al.*, 2020; Banino *et al.*, 2021]) and so on.

Encoder for local per-timestep sequence

The early notable success of this method is embodied in using Transformers to process complex information from a variable number of entities scattered in the agent's observation. Zambaldi *et al.* [2018a] first propose to capture relational reasoning over structured observation with multi-head dot-product attention, which is subsequently used in AlphaStar [Vinyals *et al.*, 2019] to process multi-entity observation in the challenging multi-agent StarCraft II environment. In such a mechanism, called entity Transformer, the observation is encoded in the form:

$$\mathrm{Emb} = \mathrm{Transformer}(e_1, \cdots, e_i, \cdots),$$

where e_i represents the agent's observation on entity i either directly sliced from the whole observation or given by an entity tokenizer.

Several follow-up works have enriched entity Transformer mechanisms. Hu *et al.* [2020] propose a compatible decoupling policy to explicitly associate actions to various entities and exploit an attention mechanism for policy explanation. To solve the challenging one-shot visual imitation, Dasari and Gupta [2021] use Transformers to learn a representation focusing on task-specific elements.

Similar to entities scattered in observation, some works exploit Transformers to process other local per-timestep sequences. Tang and Ha [2021] leverage the attention mechanism of Transformers to process sensory sequence and construct a policy that is permutation invariant w.r.t. inputs. In the incompatible multi-task RL setting, Transformer is proposed to extract morphological domain knowledge [Kurin *et al.*, 2020].

Encoder for temporal sequence

Meanwhile, it is also reasonable to process temporal sequence with Transformers. Such a temporal encoder works as a memory architecture,

$$\mathrm{Emb}_{0:t} = \mathrm{Transformer}(o_0, \cdots, o_t),$$

where o_t represents the agent's observation at timestep t and $\mathrm{Emb}_{0:t}$ represents the embedding of historical observations from initial observation to current observation.

In the early work, Mishra *et al.* [2018] fail to process temporal sequence with vanilla Transformers and find it even worse than random policy in some certain tasks. Gated Transformer-XL (GTrXL) [Parisotto *et al.*, 2020] is the first efficacious scheme to use Transformer as a memory architecture to process trajectories. GTrXL modifies Transformer-XL architecture [Dai *et al.*, 2019] with Identity Map Reordering to provide a 'skip' path from temporal input to the Transformer output, which may conduce to a stabilizing training procedure from the beginning. Furthermore, Loynd *et al.* [2020] propose a shortcut mechanism with memory vectors for long-term dependency and Irie *et al.* [2021] combine the linear Transformer with Fast Weight Programmers for better performance. In addition, Melo [2022] proposes to use the self-attention mechanism to mimic memory reinstatement for memory-based meta RL.

While Transformer outperforms LSTM/RNN as the memory horizon grows and parameter scales, it suffers from poor data efficiency with RL signals. Follow-up works exploit some auxiliary (self-)supervised tasks to benefit learning [Banino *et al.*, 2021] or use pre-trained Transformer architecture as a temporal encoder [Li *et al.*, 2022; Fan *et al.*, 2022].

4.2 Transformers for model learning

In addition to using Transformers as the encoder for sequence embedding, Transformer architecture also serves as the backbone of the environmental model in some model-based algorithms. Distinct from the prediction conditioned on single-step observation and action, Transformer enables the environmental model to predict transition conditioned on a certain length of historical information.

Practically, the success of Dreamer and subsequent algorithms [Hafner *et al.*, 2020, 2021; Seo *et al.*, 2022] has demonstrated the benefits of the world model conditioned on history in some partially observable environments or in some tasks that require a memory mechanism. A world model conditioned on history consists of an observation encoder to capture abstract information and a transition model to learn the transition in latent space, formally:

$$\begin{aligned} z_t &\sim P_{\text{enc}}(z_t|o_t) \\ \hat{z}_{t+1} &\sim P_{\text{trans}}(\hat{z}_{t+1}|z_{\leq t}, a_{\leq t}) \\ \hat{r}_{t+1} &\sim P_{\text{trans}}(\hat{r}_{t+1}|z_{\leq t}, a_{\leq t}) \\ \hat{\gamma}_{t+1} &\sim P_{\text{trans}}(\hat{\gamma}_{t+1}|z_{\leq t}, a_{\leq t}), \end{aligned}$$

where z_t represents the latent embedding of observation o_t, and $P_{\text{enc}}, P_{\text{trans}}$ denote observation encoder and transition model respectively.

There are several attempts to build a world model conditioned on history with Transformer architecture instead of RNN in previous works. Concretely, Chen *et al.* [2022] replace RNN-based Recurrent State-Space Model (RSSM) in Dreamer with a Transformer-based model (Transformer State-Space Model, TSSM). IRIS (Imagination with auto-Regression over an Inner Speech) [Micheli *et al.*, 2022] learns a Transformer-based world model simply via auto-regressive learning on rollout experience without KL balancing like Dreamer and achieves considerable results on the Atari [Bellemare *et al.*, 2013] 100k benchmark.

Besides, some works also try out Transformer-based world model with planning. Ozair *et al.* [2021] verify the efficacy of planning with a Transformer transition model to tackle stochastic tasks requiring long tactical look-ahead. Sun *et al.* [2022] propose a goal-conditioned Transformer-based transition model which is effective in visual-grounded planning for procedural tasks.

It is true that both RNN and Transformer are compatible with learning a world model conditioned on historical information. However, Micheli *et al.* [2022] find Transformer architecture is a more data-efficient world model compared with Dreamer, and experimental results of TSSM demonstrate that Transformer architecture is lucrative in tasks that require long-term memory. In fact, although model-based methods are data-efficient, they suffer from the compounding prediction error increasing with model rollout length, which greatly affects the performance and limits model rollout length [Janner *et al.*, 2019]. Thus, it is valuable to maintain prediction accuracy on longer sequences, and the Transformer-based world model might benefit from this aspect.

4.3 Transformers for sequential decision-making

In addition to being an expressive architecture to be plugged into components of traditional RL algorithms, Transformer itself can serve as a model that conducts sequential decision-making directly. This is because RL can be viewed as a conditional sequence modeling problem — generating a sequence of actions that can yield high returns.

Transformers as a milestone for offline RL

One challenge for Transformers to be widely used in RL is that the non-stationarity during the training process may hinder its optimization. However, the recent prosperity in offline RL motivates a growing number of works focusing on training a Transformer model on offline data that can achieve state-of-the-art performance. Decision Transformer (DT) [Chen *et al.*, 2021] first applies this idea by modeling RL as an autoregressive generation problem to produce the desired trajectory:

$$\tau = \left(\hat{R}_1, s_1, a_1, \hat{R}_2, s_2, a_2, \dots \hat{R}_T, s_T, a_T\right),$$

where $\hat{R}_t = \sum_{t'=t}^{T} r(s_{t'}, a_{t'})$ is the return-to-go. By conditioning on the proper target return value at the first timestep, DT can generate desired actions without explicit TD learning or dynamic programming. Concurrently with this work, Trajectory Transformer (TT) [Janner *et al.*, 2021] adopts a similar Transformer structure, but alternatively proposes to use beam search for planning during execution. The empirical results demonstrate that TT performs well on long-horizon prediction. Moreover, TT shows that with mild adjustments on vanilla beam search, TT can perform imitation learning, goal-conditioned RL, and offline RL under the same framework. Regarding the behavior cloning setting, Behavior Transformer (BeT) [Shafiullah *et al.*, 2022] proposes a similar Transformer structure as TT to learn from multi-modal datasets.

In light of Transformer's superior accuracy on sequence prediction, Bootstrapped Transformer (BooT) [Wang *et al.*, 2022] proposes to bootstrap Transformer to generate data while optimizing it for sequential decision-making. Bootstrapping Transformer for data augmentation can expand the amount and coverage of offline datasets, and hence achieve performance improvement. More specifically, BooT compares different data generation schemes and bootstrapping schemes to analyze how BooT can benefit policy learning. The results show that it can generate data consistent with the underlying MDP without additional explicit conservative constraints.

Different choices of conditioning

While conditioning on return-to-go is a practical choice to incorporate future trajectory information, one natural question is whether other kinds of hindsight information can benefit sequential decision-making. To this end, Furuta *et al.* [2021] propose Hindsight Information Matching (HIM), a unified framework that can formulate variants of hindsight RL problems. More specifically, HIM converts hindsight RL into matching any pre-defined statistics of future trajectory information w.r.t. the distribution induced by the learned conditional policy. Furthermore, this work proposes Generalized DT (GDT) for arbitrary choices of statistics and demonstrates its applications in two HIM problems: offline multi-task state-marginal matching and imitation learning.

Specifically, one drawback of conditioning on return-to-go is that it will lead to sub-optimal actions in stochastic environments. This is because the training data may contain sub-optimal actions that result in high rewards by luck due to the stochasticity of transitions. Paster *et al.* [2022] identify this limitation in general RvS methods. They further formulate RvS as an HIM problem and discover that RvS policies can achieve goals in consistency if the information statistics are independent of transitions' stochasticity. Based on this implication, they propose environment-stochasticity-independent

Method	Setting	Hindsight Info	Inference	Additional Structure/Usage
DT [Chen *et al.*, 2021]	Offline	return-to-go	conditioning	basic Transformer structure
TT [Janner *et al.*, 2021]	IL/GCRL/Offline	return-to-go	beam search	basic Transformer structure
BeT [Shafiullah *et al.*, 2022]	BC	none	conditioning	basic Transformer structure
BooT [Wang *et al.*, 2022]	Offline	return-to-go	beam search	data augmentation
GDT [Furuta *et al.*, 2021]	HIM	arbitrary	conditioning	anti-causal aggregator
ESPER [Paster *et al.*, 2022]	Offline (stochastic)	expected return	conditioning	adversarial clustering
DoC [Yang *et al.*, 2022]	Offline (stochastic)	learned representation	conditioning	additional latent value func.
QDT [Yamagata *et al.*, 2022]	Offline	relabelled return-to-go	conditioning	additional Q func.
StARformer [Shang *et al.*, 2022]	IL/Offline	return-to-go/reward	conditioning	Step Transformer
ConDT [Konan *et al.*, 2022]	Offline	learned representation	conditioning	return-dependent transformation
SPLT [Villaflor *et al.*, 2022]	Offline	none	min-max search	separate models for world and policy
ODT [Zheng *et al.*, 2022]	Online finetune	return-to-go	conditioning	trajectory-based entropy
MADT [Meng *et al.*, 2021]	Online finetune (multi-agent)	none	conditioning	separate models for actor and critic

Table 1: A summary of Transformers for sequential decision-making.

representations (ESPER), an algorithm that first clusters trajectories and estimates average returns for each cluster, and then trains a policy conditioned on the expected returns. Alternatively, Dichotomy of Control (DoC) [Yang *et al.*, 2022] proposes to learn a representation that is agnostic to stochastic transitions and rewards in the environment via minimizing mutual information. During inference, DoC selects the representation with the highest value and feeds it into the conditioned policy.

In addition to exploring different hindsight information, another approach to enhance return-to-go conditioning is to augment the dataset. Q-learning DT (QDT) [Yamagata *et al.*, 2022] proposes to use a conservative value function to relabel return-to-go in the dataset, hence combining DT with dynamic programming and improving its stitching capability.

Improving the structure of Transformers

Apart from studying different conditioned information, there are also some works to improve the structure of DT. To solve visual input tasks, StARformer [Shang *et al.*, 2022] proposes learning an additional Step Transformer for local per-timestep representation and using this representation for sequence modeling. Konan *et al.* [2022] believe that different levels of return-to-go throughout the task procedure are the identification of the sub-tasks during task execution, and the tokenization requirements of each sub-task are distinct. To address this problem, they propose Contrastive Decision Transformer (ConDT) structure, where a return-dependent transformation is applied to state embedding and action embedding before putting them into a causal Transformer. The return-dependent transformation intuitively captures features specific to the current sub-task and is learned with an auxiliary contrastive loss to strengthen the correlation between transformation and return. Villaflor *et al.* [2022] analyze one disadvantage of implementing model prediction and policy network in the same model as TT. In safety-critical scenarios with long-term planning, the preference between predicting future states and making action decisions is often contradictory. Concretely, it is necessary to find the best action in the worst future, which is difficult to complete in one model. Therefore they propose SeParated Latent Trajectory Transformer (SPLT Transformer), consisting of two independent Transformer-based CVAE structures of the world model and policy model, with the trajectory as the condition. SPLT Transformer searches the latent variable space to minimize return-to-go in the world model and to maximize return-to-go in the policy model during planning, similar to the min-max search procedure.

Extending DT beyond offline RL

Although most of the works around Transformers for sequential decision-making focus on the offline setting, there are several attempts to adapt this paradigm to online and multi-agent settings. Online Decision Transformer (ODT) [Zheng *et al.*, 2022] replaces the deterministic policy in DT with a stochastic counterpart and defines a trajectory-level policy entropy to help exploration during online finetuning. Besides, such a two-stage paradigm (offline pre-training with online finetuning) is also applied to Multi-Agent Decision Transformer (MADT) [Meng *et al.*, 2021], where a decentralized DT is pre-trained with offline data from the perspective of individual agents and is used as the policy network in online finetuning with MAPPO [Yu *et al.*, 2021a].

4.4 Transformers for generalist agents

In view of the fact that Decision Transformer has already flexed its muscles in various tasks with offline data, several works turn to consider whether Transformers can enable a generalist agent to solve multiple tasks or problems, as in the CV and NLP fields.

Generalize to multiple tasks

Some works draw on the ideas of pre-training on large-scale datasets in CV and NLP, and try to abstract a general policy from large-scale multi-task datasets. Multi-Game Decision Transformer (MGDT) [Lee *et al.*, 2022], a variant of DT, learns DT on a diverse dataset consisting of both expert and non-expert data and achieves close-to-human performance on multiple Atari games with a single set of parameters. In order to obtain expert-level performance with a dataset containing non-expert experiences, the expert action inference mechanism is designed in MGDT, which calculates an expert-level return-to-go posterior distribution from the prior distribution of return-to-go and a preset expert-level return-to-go likelihood proportional according to Bayesian formula. Likewise, Switch Trajectory Transformer (SwitchTT) [Lin *et al.*, 2022], a multi-task extension to TT, exploits a sparsely activated model that replaces the FFN layer with a mixture-of-expert layer for efficient multi-task offline learning. Besides, a distributional trajectory value estimator is adopted to model the uncertainty of value estimates. With these two enhanced features, SwitchTT achieves improvement over TT across multiple tasks in terms of both performance and training speed. MGDT and SwitchTT exploit experiences collected from multiple tasks and various performance-level policies to learn a general policy. Yet, constructing a large-scale multi-task dataset is non-trivial. Unlike large-scale datasets in CV or NLP, which are usually constructed with massive public data from the Internet and simple manual labeling, action information is always absent from public sequential decision-making data and is not facile to label. Thus, Baker *et al.* [2022] propose a semi-supervised scheme to utilize large-scale online data without action information and the key is to learn a Transformer-based Inverse Dynamic Model (IDM), which predicts the action information with past and future observations and is consequently capable of labeling massive unlabeled online video data. IDM is learned on a small-scale dataset containing manually labeled actions and is accurate enough to provide action labels of videos for effective behavior cloning and fine-tuning.

The efficacy of prompting [Brown *et al.*, 2020] for adaptation to new tasks has been proven in many prior works in NLP. Following this idea, several works aim at leveraging prompting techniques for DT-based methods to enable fast adaptation. Prompt-based Decision Transformer (Prompt-DT) [Xu *et al.*, 2022] samples a sequence of transitions from the few-shot demonstration dataset as prompt, and shows that it can achieve few-shot policy generalization on offline meta RL tasks. Reed *et al.* [2022] further exploit prompt-based architecture to learn a generalist agent (Gato) via auto-regressive sequence modeling on a super large-scale dataset covering natural language, image, temporal decision-making, and multi-modal data. Gato is capable of a range of tasks from various domains, including text generation and decision-making. Specifically, Gato unifies multi-modal sequences in a shared tokenization space and adapts prompt-based inference in deployment to generate task-specific sequences. Despite being effective, Laskin *et al.* [2022] point out that one limitation of the prompt-based framework is that the prompt is demonstrations from a well-behaved policy, as contexts in both works are not sufficient to capture policy improvement. Instead, they propose Algorithm Distillation (AD) [Laskin *et al.*, 2022], which instead trains a Transformer on across-episode sequences of the learning progress of single-task RL algorithms. Therefore, even in new tasks, the Transformer can learn to gradually improve its policy during the auto-regressive generation.

Generalize to multiple domains

Beyond generalizing to multiple tasks, Transformer is also a powerful "universal" model to unify a range of domains related to sequential decision-making. Motivated by advances in masked language modeling [Devlin *et al.*, 2018] technique in NLP, Carroll *et al.* [2022] propose Uni[MASK], which unifies various commonly-studied domains, including behavioral cloning, offline RL, GCRL, past/future inference, and dynamics prediction, as one mask inference problem. Uni[MASK] compares different masking schemes, including task-specific masking, random masking, and finetune variants. It is shown that one single Transformer trained with random masking can solve arbitrary inference tasks. More surprisingly, compared to the task-specific counterpart, random masking can still improve performance in the single-task setting.

In addition to unifying sequential inference problems in the RL domain, Reid *et al.* [2022] find it beneficial to finetune DT with Transformer pre-trained on language datasets or multi-modal datasets containing language modality. Concretely, Reid *et al.* [2022] find out that pre-training Transformer with language data while encouraging similarity between language and RL-based representations can help improve the performance and convergence speed of DT. This finding implies that even knowledge from non-RL fields can benefit RL training via Transformers. Additionally, Huang *et al.* [2022] discover that pre-trained large-scale language models are capable of generating reasonable high-level plans to accomplish complex tasks without further finetuning. With proper correction and prompting, the Transformer can generate valid actions in the embodied environment. Furthermore, similarly to Gato, RT-1 [Brohan *et al.*, 2022] leverages large-scale datasets with diverse robotics experiences and language instructions to train a Transformer as well as a tokenizer, which achieves high performance on downstream tasks.

5 Summary and Future Perspectives

This paper briefly reviews advances in Transformers for RL. We provide a taxonomy of these advances: *a)* Transformers can serve as a powerful module of RL, e.g., acting as a representation module or a world model; *b)* Transformers can serve as a sequential decision-maker; *c)* Transformers can benefit generalization across tasks and domains. While we cover representative works on this topic, the usage of Transformers in RL is not limited to our discussions. Given the prosperity of Transformers in the broader AI community, we believe that combining Transformers and RL is a promising trend. To conclude this survey, in the following, we discuss future perspectives and open problems for this direction.

Combining Reinforcement Learning and (Self-) Supervised Learning. Retracing the development of TransformRL, the training methods involve both RL and (self-)supervised learning. When serving as a representation module that is trained under the conventional RL framework, optimization of the Transformer architecture is usually unstable. When using Transformers to solve decision-making problems via sequence modeling, the "deadly triad problem" [Van Hasselt *et al.*, 2018] is eliminated due to (self-)supervised learning paradigm. Under the framework of (self-)supervised learning, the performance of policy is deeply bounded to offline-data quality and the explicit trade-off between exploitation and exploration no longer exists. Therefore, a better policy may be learned when we combine RL and (self-)supervised learning in Transformer learning. Some works [Zheng *et al.*, 2022; Meng *et al.*, 2021] have tried out the scheme of supervised pre-training and RL-involved finetuning. However, exploration can be limited with a relatively fixed policy [Nair *et al.*, 2020], which is one of the bottlenecks to be resolved. Also, along this line, the tasks used for performance evaluation are relatively simple. It is worthwhile to further explore whether Transformers can scale such (self-)supervised learning up to larger datasets, more complex environments, and real-world applications. Further, we expect future work to provide more theoretical and empirical insights to characterize under which conditions this (self-)supervised learning is expected to perform well [Brandfonbrener *et al.*, 2022].

Bridging Online and Offline Learning via Transformers. Stepping into Offline RL is a milestone in TransformRL. Practically, utilizing Transformers to capture dependencies in decision sequence and to abstract policy is mainly inseparable from the support of considerable offline data used. However, it is unfeasible for some decision-making tasks to get rid of the online framework in real applications. On the one hand, it is not that easy to obtain expert data in certain tasks. On the other hand, some environments are open-ended (e.g., Minecraft), which means the policy has to continually adjust to deal with unseen tasks during online interaction. Therefore, we believe that bridging online learning and offline learning is necessary. However, most research progress following Decision Transformer focuses on the offline learning framework. Several works have attempted to adopt the paradigm of offline pre-training and online fine-tuning [Xie *et al.*, 2022]. Yet, the distribution shift in online fine-tuning still exists as that in offline RL algorithms, we thereby expect some special designs for Decision Transformer to address this issue. In addition, how to train an online Decision Transformer from scratch is an interesting open problem.

Transformer Structure Tailored for Decision-making Problems. The Transformer structures in the current Decision Transformer series methods are mainly vanilla Transformer, which is originally designed for the text sequence and may not fit the nature of decision-making problems. For example, is it appropriate to adopt the vanilla self-attention mechanism for trajectory sequences? Whether different elements in the decision sequence or different parts of the same elements need to be distinguished in position embedding? In addition, as there are many variants of representing trajectory as a sequence in different Decision Transformer algorithms, how to choose from them still lacks systematic research. For instance, how to select robust hindsight information when deploying such algorithms in the industry? Furthermore, the vanilla Transformer is a structure with huge computational cost, which makes it expensive at both training and inference stages, and high memory occupation, which constrains the length of the dependencies it captures. To alleviate these, some works in NLP [Zhou *et al.*, 2021] have improved the structure from these aspects, and it is also worth exploring whether similar structures can be used in the decision-making problem.

Towards More Generalist Agents with Transformers. Our review on Transformers for generalist agents has shown the potential of Transformers as a general policy (Section 4.4). In fact, the design of Transformers allows multiple modalities (e.g., image, video, text, and speech) to be processed using similar processing blocks and demonstrates excellent scalability to very large-capacity networks and huge datasets. Recent works have made substantial progress in training agents that can be capable of performing multiple and cross-domain tasks. However, given that these agents are trained on huge amounts of data, it is still uncertain whether they merely memorize the dataset and if they can perform efficient generalization. Therefore, how to learn an agent that can generalize to unseen tasks without strong assumptions is a problem worth studying [Boustati *et al.*, 2021]. Moreover, we are curious about whether Transformer is strong enough to learn a general world model that can be used in different tasks and scenarios.

RL for Transformers. While we have discussed how RL can benefit from the usage of Transformers, the reverse direction, i.e., using RL to benefit Transformer training is an intriguing open problem yet less explored. We see that, very recently, Reinforcement Learning from Human Feedback (RLHF) [Ouyang *et al.*, 2022] learns a reward model and uses RL algorithms to finetune Transformer for aligning language models with human intent. In the future, we believe RL can be a useful tool to further polish Transformer's performance in other domains.

References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, et al. What matters for on-policy deep actor-critic methods? a large-scale study. In *International conference on learning representations*, 2020.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2019.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (VPT): Learning to act by watching unlabeled online videos. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

Andrea Banino, Adria Puigdomenech Badia, Jacob C Walker, Tim Scholtes, Jovana Mitrovic, and Charles Blundell. Coberl: Contrastive bert for reinforcement learning. In *International Conference on Learning Representations*, 2021.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al.

Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Ayman Boustati, Hana Chockler, and Daniel C McNamee. Transfer learning with causal counterfactual reasoning in decision transformers. *arXiv preprint arXiv:2110.14355*, 2021.

David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. When does return-conditioned supervised learning work for offline reinforcement learning? *arXiv preprint arXiv:2206.01079*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Micah Carroll, Orr Paradise, Jessy Lin, Raluca Georgescu, Mingfei Sun, David Bignell, Stephanie Milani, Katja Hofmann, Matthew Hausknecht, Anca Dragan, et al. Unimask: Unified inference in sequential decision problems. *arXiv preprint arXiv:2211.10869*, 2022.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019.

Sudeep Dasari and Abhinav Gupta. Transformers for one-shot visual imitation. In *Conference on Robot Learning*, pages 2071–2084. PMLR, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888. IEEE, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International conference on learning representations*, 2019.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

Hiroki Furuta, Yutaka Matsuo, and Shixiang Shane Gu. Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364*, 2021.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. *arXiv preprint arXiv:1912.06088*, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*, 2015.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Siyi Hu, Fengda Zhu, Xiaojun Chang, and Xiaodan Liang. Updet: Universal multi-agent rl via policy decoupling with transformers. In *International Conference on Learning Representations*, 2020.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*, 2022.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. Going beyond linear transformers with recurrent fast weight programmers. *Advances in Neural Information Processing Systems*, 34:7703–7717, 2021.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Michael Janner, Qiyang Li, and Sergey Levine. Reinforcement learning as one big sequence modeling problem. In *ICML 2021 Workshop on Unsupervised Reinforcement Learning*, 2021.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

Sachin G Konan, Esmaeil Seraj, and Matthew Gombolay. Contrastive decision transformers. In *6th Annual Conference on Robot Learning*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Vitaly Kurin, Maximilian Igl, Tim Rocktäschel, Wendelin Boehmer, and Shimon Whiteson. My body is a cage: the role of morphology in graph-based incompatible control. *arXiv preprint arXiv:2010.01856*, 2020.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*, 2022.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Kuang-Huei Lee, Ofir Nachum, Sherry Yang, Lisa Lee, C. Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, and Igor Mordatch. Multi-game decision transformers. In *Advances in Neural Information Processing Systems*, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Shuang Li, Xavier Puig, Yilun Du, Clinton Wang, Ekin Akyurek, Antonio Torralba, Jacob Andreas, and Igor Mordatch. Pre-trained language models for interactive decision-making. *arXiv preprint arXiv:2202.01771*, 2022.

Zichuan Lin, Li Zhao, Derek Yang, Tao Qin, Tie-Yan Liu, and Guangwen Yang. Distributional reward decomposition for reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Qinjie Lin, Han Liu, and Biswa Sengupta. Switch trajectory transformer with distributional value approximation for multi-task reinforcement learning. *arXiv preprint arXiv:2203.07413*, 2022.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

Ricky Loynd, Roland Fernandez, Asli Celikyilmaz, Adith Swaminathan, and Matthew Hausknecht. Working memory graphs. In *International conference on machine learning*, pages 6404–6414. PMLR, 2020.

Luckeciano C Melo. Transformers are meta-reinforcement learners. In *International Conference on Machine Learning*, pages 15340–15359. PMLR, 2022.

Linghui Meng, Muning Wen, Yaodong Yang, Chenyang Le, Xiyun Li, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, and Bo Xu. Offline pre-trained multi-agent decision transformer: One big sequence model conquers all starcraftii tasks. *arXiv preprint arXiv:2112.02845*, 2021.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Kei Ota, Tomoaki Oiki, Devesh Jha, Toshisada Mariyama, and Daniel Nikovski. Can increasing input dimensionality improve deep reinforcement learning? In *International Conference on Machine Learning*, pages 7424–7433. PMLR, 2020.

Kei Ota, Devesh K Jha, and Asako Kanezaki. Training larger networks for deep reinforcement learning. *arXiv preprint arXiv:2102.07920*, 2021.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Sherjil Ozair, Yazhe Li, Ali Razavi, Ioannis Antonoglou, Aaron Van Den Oord, and Oriol Vinyals. Vector quantized models for planning. In *International Conference on Machine Learning*, pages 8302–8313. PMLR, 2021.

Emilio Parisotto and Ruslan Salakhutdinov. Efficient transformers in reinforcement learning using actor-learner distillation. *arXiv preprint arXiv:2104.01655*, 2021.

Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pages 7487–7498. PMLR, 2020.

Keiran Paster, Sheila McIlraith, and Jimmy Ba. You can't count on luck: Why decision transformers fail in stochastic environments. *arXiv preprint arXiv:2205.15967*, 2022.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Machel Reid, Yutaro Yamada, and Shixiang Shane Gu. Can wikipedia help offline reinforcement learning? *arXiv preprint arXiv:2201.12122*, 2022.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pretraining from videos. In *International Conference on Machine Learning*, pages 19561–19579. PMLR, 2022.

Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *arXiv preprint arXiv:2206.11251*, 2022.

Jinghuan Shang, Kumara Kahatapitiya, Xiang Li, and Michael S Ryoo. Starformer: Transformer with state-action-reward representations for visual reinforcement learning. In *European Conference on Computer Vision*, pages 462–479. Springer, 2022.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Samarth Sinha, Homanga Bharadhwaj, Aravind Srinivas, and Animesh Garg. D2rl: Deep dense architectures in reinforcement learning. *arXiv preprint arXiv:2010.09163*, 2020.

Jiankai Sun, De-An Huang, Bo Lu, Yun-Hui Liu, Bolei Zhou, and Animesh Garg. Plate: Visually-grounded planning with transformers in procedural tasks. *IEEE Robotics and Automation Letters*, 7(2):4924–4930, 2022.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Yujin Tang and David Ha. The sensory neuron as a transformer: Permutation-invariant neural networks for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:22574–22587, 2021.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.

Hado Van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil. Deep reinforcement learning and the deadly triad. *arXiv preprint arXiv:1812.02648*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Adam R Villaflor, Zhe Huang, Swapnil Pande, John M Dolan, and Jeff Schneider. Addressing optimism bias in sequence modeling for reinforcement learning. In *International Conference on Machine Learning*, pages 22270–22283. PMLR, 2022.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.

Kerong Wang, Hanye Zhao, Xufang Luo, Kan Ren, Weinan Zhang, and Dongsheng Li. Bootstrapped transformer for offline reinforcement learning. *arXiv preprint arXiv:2206.08569*, 2022.

Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *arXiv preprint arXiv:2205.14953*, 2022.

Zhihui Xie, Zichuan Lin, Junyou Li, Shuai Li, and Deheng Ye. Pretraining in deep reinforcement learning: A survey. *arXiv preprint arXiv:2211.03959*, 2022.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *International Conference on Machine Learning*, pages 24631–24645. PMLR, 2022.

Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. *arXiv preprint arXiv:2209.03993*, 2022.

Mengjiao Yang, Dale Schuurmans, Pieter Abbeel, and Ofir Nachum. Dichotomy of control: Separating what you can control from what you cannot. *arXiv preprint arXiv:2210.13435*, 2022.

Deheng Ye, Guibin Chen, Wen Zhang, Sheng Chen, Bo Yuan, Bo Liu, Jia Chen, Zhao Liu, Fuhao Qiu, Hongsheng Yu, et al. Towards playing full moba games with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:621–632, 2020.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, et al. Mastering complex control in moba games with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6672–6679, 2020.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, et al. Deep reinforcement learning with relational inductive biases. In *International conference on learning representations*, 2018.

Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, et al. Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830*, 2018.

Qinqing Zheng, Amy Zhang, and Aditya Grover. Online decision transformer. *arXiv preprint arXiv:2202.05607*, 2022.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.