Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Review:

Transformer in reinforcement learning for decision-making: a survey*

Weilin YUAN†1, Jiaxing CHEN[2], Shaofei CHEN[2], Dawei FENG[3], Zhenzhen HU[2], Peng LI[2], Weiwei ZHAO†‡1

[1]*College of Information and Communication, National University of Defense Technology, Wuhan 430014, China*
[2]*College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410072, China*
[3]*Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410072, China*
†E-mail: yuanweilin12@nudt.edu.cn; zhaozww@163.com

Received Aug. 14, 2023; Revision accepted Nov. 24, 2023; Crosschecked May 25, 2024

Abstract: Reinforcement learning (RL) has become a dominant decision-making paradigm and has achieved notable success in many real-world applications. Notably, deep neural networks play a crucial role in unlocking RL's potential in large-scale decision-making tasks. Inspired by current major success of Transformer in natural language processing and computer vision, numerous bottlenecks have been overcome by combining Transformer with RL for decision-making. This paper presents a multiangle systematic survey of various Transformer-based RL (TransRL) models applied in decision-making tasks, including basic models, advanced algorithms, representative implementation instances, typical applications, and known challenges. Our work aims to provide insights into problems that inherently arise with the current RL approaches, and examines how we can address them with better TransRL models. To our knowledge, we are the first to present a comprehensive review of the recent Transformer research developments in RL for decision-making. We hope that this survey provides a comprehensive review of TransRL models and inspires the RL community in its pursuit of future directions. To keep track of the rapid TransRL developments in the decision-making domains, we summarize the latest papers and their open-source implementations at https://github.com/williamyuanv0/Transformer-in-Reinforcement-Learning-for-Decision-Making-A-Survey.

Key words: Transformer; Reinforcement learning (RL); Decision-making (DM); Deep neural network (DNN); Multi-agent reinforcement learning (MARL); Meta-reinforcement learning (Meta-RL)
https://doi.org/10.1631/FITEE.2300548 **CLC number:** TP18

1 Introduction

Decision-making (DM) is a high-level agent activity that significantly impacts artificial intelligence (AI) advancements in improving the quality of decisions and enhancing agent capability (Phillips-Wren, 2012; Kochenderfer et al., 2022). In the AI community, a central concern is how to effectively train agents to learn the optimal strategies for complex tasks in simulators or physical environments. To resolve this problem, many advanced approaches (Shoham and Leyton-Brown, 2008; Liu T et al., 2019; Alquier, 2020; Zhao YP et al., 2022) have been proposed, and reinforcement learning (RL) (Sutton and Barto, 2018) has become a dominant paradigm because of its state-of-the-art performance in a wide variety of applications, such as autonomous driving, robotic manipulation, robotic navigation, and gaming AI. The fundamental framework for all these models is usually the deep neural network (DNN)

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. 62376280)
ORCID: Weilin YUAN, https://orcid.org/0000-0001-9894-5253; Weiwei ZHAO, https://orcid.org/0009-0002-6989-8536

(Srinidhi et al., 2021). In particular, in solving large-scale DM tasks, such as Go (Silver et al., 2017b), Texas hold'em (Moravčík et al., 2017), StarCraft (Vinyals et al., 2019), and Dota 2 (Berner et al., 2019), DNNs play a key role in unlocking the potential of RL. Although simple DNN models (e.g., fully connected networks and multilayer perceptions) benefit from RL producing competitive results by fitting the state value, or by mapping the abstract feature to policies, many researchers have proposed various types of networks for different types of tasks. For example, convolutional neural networks (CNNs) (Krizhevsky et al., 2012; Kim, 2014) introduce convolutional operations and pooling operations for processing shift-invariant data (e.g., images). Recurrent neural networks (RNNs) (Zaremba et al., 2014) use recurrent cells to process ordinal and temporal data (e.g., sentences). Transformer (Dong et al., 2021) is a new type of neural network that uses mainly the self-attention mechanism (Ambartsoumian and Popowich, 2018) to extract intrinsic features and has great potential for extensive use in AI-architected applications (Han K et al., 2023). It has become a prevalent architecture in several domains. Fig. 1 summarizes the milestones in the development of Transformer.

1.1 Transformer development: from natural language processing and computer vision to decision-making

Transformer has been applied in natural language processing (NLP) and computer vision (CV) and has achieved considerable success. Transformer was first proposed as a simple new architecture to handle sequential data in NLP tasks (Vaswani et al., 2017), and attained state-of-the-art results. Subsequently, many researchers actively developed Transformer variants taking advantage of the effective encoding of information over long time horizons and the scaling of large amounts of data, which led Transformer to become the dominant deep learning architecture in a broad set of NLP tasks, such as question answering (Lewis P et al., 2021) and language inference (Guo et al., 2019). A collection of Transformer-based language models, such as BERT (Devlin et al., 2019), GPT series (Radford et al., 2018, 2019; Brown et al., 2020), ChatGPT (Zhou et al., 2023), XLNet (Yang ZL et al., 2019), RoBERTa (Liu YH et al., 2019), ALBERT (Lan et al., 2020), and BART (Lewis M et al., 2020), have been proposed recently and are highly competitive with the state-of-the-art approaches. Meanwhile, in CV tasks, the Transformer model has become an attractive solution to various vision problems, such as object detection, semantic segmentation, image processing, and video understanding. Many advanced Transformer-based vision models have been proposed successively, such as iGPT (Chen M et al., 2020), ViT (Dosovitskiy et al., 2021), DERT (Carion et al., 2020), iPT (Chen HT et al., 2021), VQGAN (Esser et al., 2021), CLIP (Radford et al., 2021), and TransGAN (Jiang et al., 2021).

Inspired by Transformer's major success in the fields of CV and NLP, researchers have made efforts to explore the benefit of Transformer models in solving DM tasks. Parisotto and Salakhutdinov (2021)

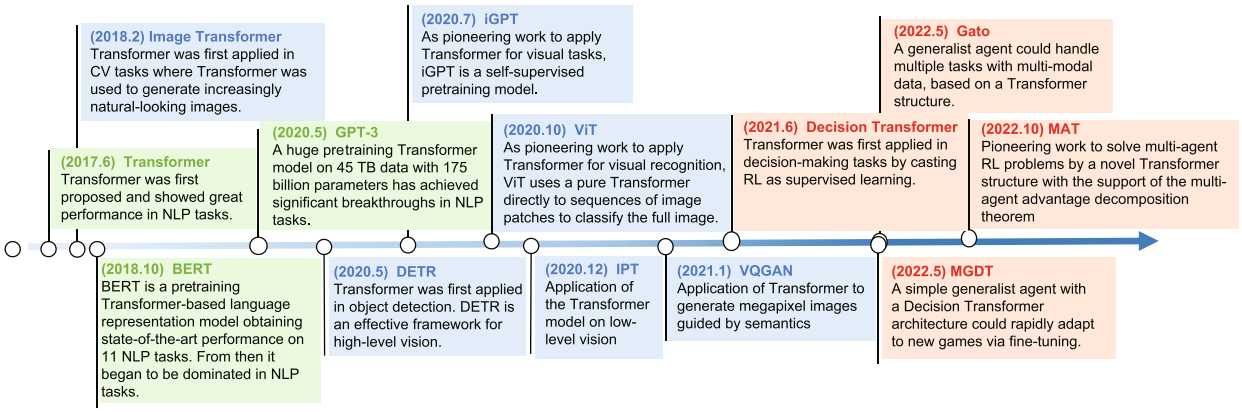

Fig. 1 Milestones in the development of Transformer. Transformer-based natural language processing (NLP) models are marked in green, Transformer-based computer vision (CV) models in blue, and decision-making (DM) Transformer models in red. References to color refer to the online version of this figure

developed an actor–learner distillation procedure to model Transformer as a learner, in which the ability of Transformer to process long-range dependencies results in a huge performance boost in RL. Chen LL et al. (2021) proposed a simple new model called Decision Transformer (DT), which implements optimal actions leveraging a causally masked Transformer, primarily viewing the offline RL problem as a pure sequence-to-sequence model. Similarly, Janner et al. (2021) provided a Trajectory Transformer from an analogous perspective that views an offline RL as a generic sequence generation problem, adopting a Transformer architecture to predict distributions over trajectories. Owing to the remarkable performance of Transformer, an increasing number of researchers are proposing Transformer-based models for integration with RL to improve decision results in a wide range of DM tasks. Presently, Transformer-based RL (TransRL) is a hot topic in the DM community. We have collected thousands of publications in recent years, and the growth of the number of TransRL publications is shown in Figs. 2 and 3. The growth of the number of TransRL-related publications shows a continuous linear progression since 2017. In particular, Fig. 3 shows that an increasing number of TransRL methods are being developed in both the RL and Transformer communities.

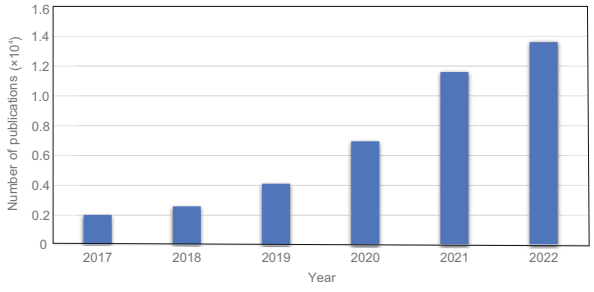

Fig. 2 Increase of the number of TransRL publications with time

1.2 Related surveys

With the dramatic increase in the number of TransRL investigations, keeping track of the latest progress is becoming increasingly difficult. As such, a survey of the existing TransRL works is an urgent necessity and would be helpful for the AI community. In this paper, we review the existing survey works on relevant topics to provide researchers with pointers to other papers on more specific domains

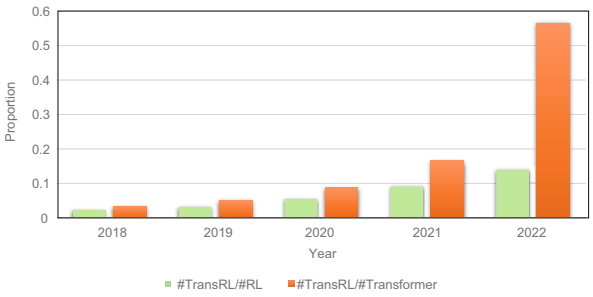

Fig. 3 Proportions of TransRL publications to the total RL publications and to the total Transformer publications

and to outline how our work differs from them. We compare these surveys in Table 1.

1. The surveys of Aleissaee et al. (2023), Han K et al. (2023), and Shamshad et al. (2023) heavily focus on Transformer models and relevant CV task applications, such as remote sensing and medical imaging.

2. Similar to our work, Keneshloo et al. (2020) summarized sequence-to-sequence problems from an RL perspective and discussed the different challenges in using RL methods to train a sequence-to-sequence model. However, Keneshloo et al. (2020) focused on NLP and CV applications, considering condition-based prediction problems as particular instances of RL, in which the reward functions in RL are the classic evaluation metrics (e.g., ROUGE (Lin CY CY, 2004), BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016)) in the sequence-to-sequence problem.

3. The survey of Imhof (2022) provides only a detailed introduction to the original TransRL method, DT (Chen LL et al., 2021), without any Transformer variants or practical application discussions.

4. The survey of Li XX et al. (2023) systematically summarizes various adversarial game models and the corresponding strategy generation approaches from a game theory perspective, and provides some real-world applications such as security, poker, and politics.

5. The survey of Lu YL and Li (2022) focuses on a particular application in a DM domain, the game AI system, considering three typical modern AI games (perfect information games, imperfect information games, and multi-agent games), providing a review of the solution techniques, paradigms, and applications.

6. Lin TY et al. (2022) provided a systematic literature review on a great variety of Transformer variants focusing on architectural modification, pre-training paradigms, and applications, where the variants are just for NLP and CV tasks.

Table 1 Comparison of related surveys

Survey	Domain			Approach	
	NLP	CV	DM	RL	Transformer
This paper			✓	✓	✓
Han K et al., 2023		✓			✓
Aleissaee et al., 2023		✓			✓
Shamshad et al., 2023		✓			✓
Keneshloo et al., 2020	✓	✓		✓	✓
Imhof, 2022				✓	✓
Li XX et al., 2023			✓		
Lu YL and Li, 2022		✓	✓		
Lin TY et al., 2022					✓

NLP: natural language processing; CV: computer vision; DM: decision-making; RL: reinforcement learning

1.3 Main contributions

In this paper, we focus on associating advances in NLP and CV to unify ideas in Transformer models and RL for DM tasks, by drawing upon the simplicity and scalability of the Transformer structure. We hope that our work will inspire more exploration into using TransRL in practical applications. Our main research contributions are summarized as follows:

1. provision of a comprehensive summary of TransRL methods in the context of DM and a new classification of TransRL,

2. provision of a detailed introduction on several kinds of representative TransRL models applied in DM tasks, and an analysis of their advantages and disadvantages,

3. summarization of different practical applications of TransRL in DM based on traditional RL applications, and

4. provision of guidelines to improve the TransRL method from different aspects.

An overview of the organization of this paper is shown in Fig. 4.

2 Formulations

2.1 Reinforcement learning methods

RL is a dominant paradigm in DM, and we will briefly introduce the RL models. In RL, a sequential Markov decision process (MDP) is considered to model interactions between the agent and the environment. Let a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma, T >$ represent an infinite-horizon MDP, where $\mathcal{S} = \{s\}$ denotes the state space and $\mathcal{A} = \{a\}$ denotes the action space. Given a state s_t, the agent chooses an action a_t by the policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Then, the agent will

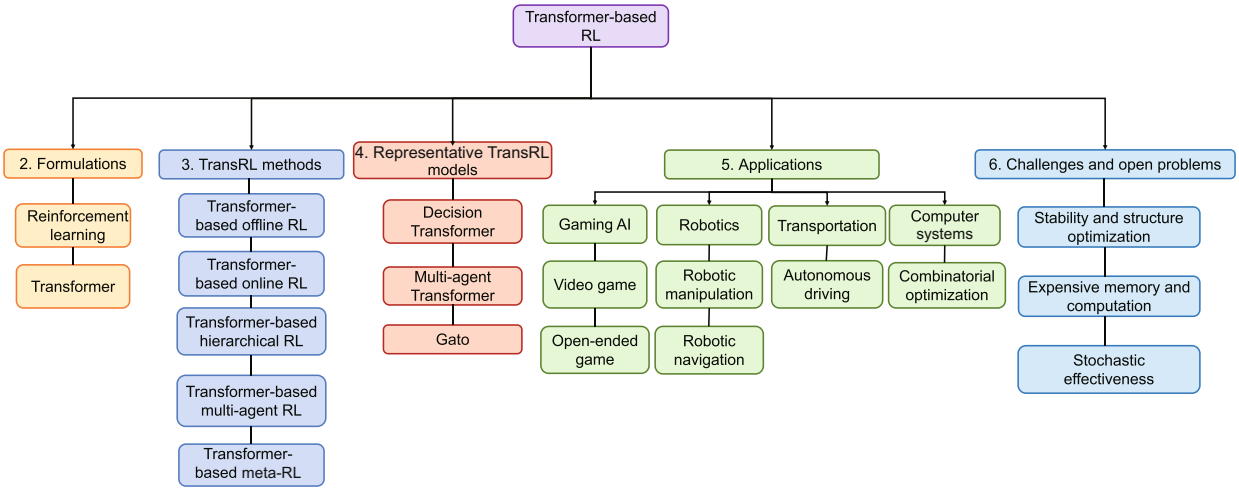

Fig. 4 Overview of the paper organization. The taxonomy of TransRL methods (Section 3) is based on reinforcement learning (RL) settings. In representative TransRL models (Section 4), we introduce the standard TransRL model, Decision Transformer, and two extensions (multi-agent Transformer is an extension to the multi-agent setting and Gato is an extension to the multi-task setting). The applications (Section 5) are a subset of RL applications in decision-making (DM) domains

obtain a reward r_t, and the next state s_{t+1} is calculated with its probability by the transition function $\mathcal{T}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$. The agent tries to maximize its total expected return $R = \sum_{t=0}^{T} \gamma^t r_i^t$, where γ is a discount factor and T is the time horizon. Therefore, the optimal policy could be represented as $\pi^* = \arg\max E[\sum_{t=0}^{T} \gamma^t r_i^t]$.

In the multi-agent setting with a partial observation, MDPs are usually extended to decentralized partially observable MDPs (Dec-POMDPs) (Bernstein et al., 2002; Oliehoek et al., 2008). A Dec-POMDP is formulated as a tuple $< N, \mathcal{S}, \mathcal{O}, \{\mathcal{A}_i\}, \{r_i\}, \mathcal{T}, \gamma >$, where $N = \{1, 2, \cdots, n\}$ $(n > 1)$ denotes the interacting agents, $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$ denotes the set of global private states, $\mathcal{O} = \{o_1, o_2, \cdots, o_n\}$ is the set of observations by all agents, and o_i is agent i's individual observation. The joint action space is denoted by $\mathcal{A} = \mathcal{A}_1 \cdot \mathcal{A}_2 \cdot \cdots \cdot \mathcal{A}_N$, which is the Cartesian product of the individual agent's action space $\mathcal{A}_i$. r_i is the reward obtained from the environment. The definitions of state transition function $\mathcal{T}$ and discount factor γ are similar to those in MDP. Each agent i tends to select an action according to its policy π_i. Denote a joint policy $\pi = \{\pi_i, \pi_{-i}\}$ as the set of all individual strategies, where $-i$ represents all agents except agent i. In contrast to the single-agent setting, the optimal strategy in the multi-agent setting is determined by the individual strategy and the other agents' strategies.

2.2 Transformer

Transformer is a new kind of neural architecture that sequentially encodes the input sequence as powerful features via the attention mechanism and generates an output sequence after the final encoder output is passed in parallel to a stack of decoders. This subsection provides an overview of vanilla Transformer models (Vaswani et al., 2017) (Fig. 5). Transformers are formed by stacking multiple Transformer blocks and are generally used in the encoder–decoder mode. Each Transformer block consists of a multi-head self-attention block, a position-wise feed-forward network, layer normalization modules, and residual connectors.

2.2.1 Encoder–decoder structure

Most competitive Transformer models are primarily based on three kinds of structures (encoder–decoder, encoder-only, and decoder-only). The structure used in a Transformer model depends on the goal application, and details of the various Transformer structures can be found in Raffel et al. (2020) and Tay et al. (2023). The encoder–decoder (Fig. 5) is a general framework based on neural networks adept at dealing with highly structured input and output (Niu et al., 2021).

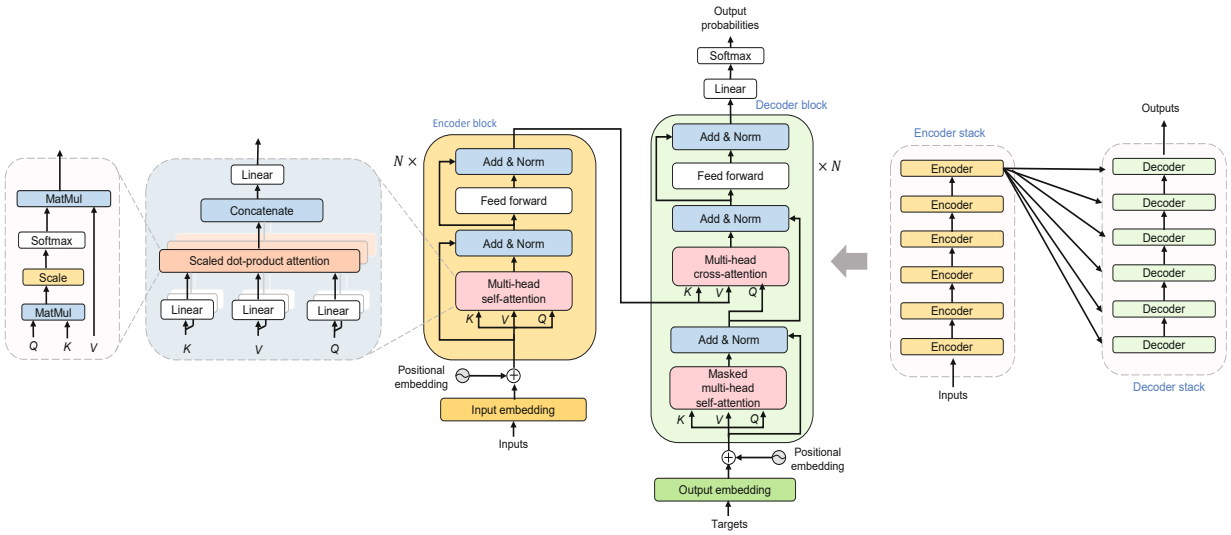

Fig. 5 Structure of Transformer (Vaswani et al., 2017; Tay et al., 2023). Left: detailed components in Transformer including the self-attention network, multi-head attention network, encoder blocks, and decoder block; right: overview of the Transformer framework

Transformer encoder: It maps an input sequence of symbol representations $x = (x_1, x_2, \ldots, x_n)$ to a sequence of representation vectors $X_{n,d}$ (n and d are the length and dimension of the input sequence, respectively) through an embedding layer and $N = 6$ identical encoder blocks, where the symbol representations are generated additively by token embeddings composed of positional embeddings. Notably, positional embeddings contain information about the relative position (such as sinusoidal positional encodings (Gehring et al., 2017)) or absolute position (Shaw et al., 2018) of the tokens in the sequence, which are essential since Transformer completely avoids recurrence and convolutions. There are two main components in each encoder block: the first component is multi-head self-attention, and the subsequent component is a position-wise feed-forward network. Each component contains a residual connection (He et al., 2016) around them, followed by layer normalization (Ba et al., 2016).

Transformer decoder: It also consists of a stack of $N = 6$ similar decoder blocks. Given target embeddings $y = (y_1, y_2, \cdots, y_m)$ and outputs of the encoder, vector $X_{n,d}$, the decoder outputs the predicted next-token probabilities. The decoder is an autoregressive (Graves, 2013) model that uses the previously generated token as an additional input when generating the next predicted token. In addition to the two main components in each encoder block, the decoder block introduces a new component that performs multi-head attention over the output of the encoder block. Similar to the encoder, each component in the decoder block contains a residual connection around it, and is followed by layer normalization. A slight difference in the first sublayer is that the decoder modifies multi-head self-attention with a masked operation to ensure that the predictions for position i can rely only on the available outputs at the previous positions.

2.2.2 Attention

The attention mechanism (Bahdanau et al., 2015; Niu et al., 2021) is a critical defining characteristic of Transformer models. Classical approaches to sequence processing use vanilla recurrent models, e.g., RNNs and long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). In contrast, Transformer bypasses recurrence and depends entirely on the attention mechanism to capture the long-range dependencies between the sequence data.

Consider three input sequences as key vector $k = (k_1, k_2, \ldots, k_n)$, query vector $q = (q_1, q_2, \ldots, q_n)$, and value vector $v = (v_1, v_2, \ldots, v_n)$, and the output z of the attention is a weighted average of the value vectors:

$$z = \sum_{j=1}^{n} \alpha_j v_j f(k_j, q), \tag{1}$$

where weight vector $\alpha = \{\alpha_j\}$ is determined by the compatibility function between queries and keys:

$$\alpha_j = \frac{\exp f(k_j, q)}{\sum_{i=1}^{n} \exp f(k_i, q)}. \tag{2}$$

Note that the keys and values can be different sets of vectors. For the compatibility function, the two commonly used ones are additive attention (Bahdanau et al., 2015) and dot-product attention. The vanilla Transformer uses a modified scaled dot-product function

$$f(k, q) = kq^{\mathrm{T}}/\sqrt{d_k}, \tag{3}$$

where $1/\sqrt{d_k}$ is a scaling factor as an additional item in vanilla Transformer, and d_k is the dimension of k.

Self-attention: Self-attention is the process of applying the attention mechanism to every position of the source sequence (Ambartsoumian and Popowich, 2018), considering a set of query vectors, key vectors, and value vectors, which are packed together into matrices Q, K, and V, respectively. Then, the output matrix is computed as

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V. \tag{4}$$

Multi-head attention: Rather than performing single self-attention once for (Q, K, V) of dimension d_m, the multi-head self-attention block calculates the self-attention results in parallel across many distinct heads, whose outputs are concatenated to serve as the input to a linear projection module, as follows:

$$\begin{aligned}&\mathrm{MultiHead}(Q, K, V)\\ &= \mathrm{Concat}(\mathrm{head}_1, \mathrm{head}_2, \ldots, \mathrm{head}_h)W^O,\end{aligned} \tag{5}$$

where

$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{6}$$

$i = 1, 2, ..., h$, and the projections are parameter matrices, $W_i^Q \in \mathbb{R}^{d_m \times d_k}$, $W_i^K \in \mathbb{R}^{d_m \times d_k}$, $W_i^V \in \mathbb{R}^{d_m \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_m}$. Multi-head attention provides the model with the ability to jointly deal with information across different positions from different representation subspaces, which can be considered as a graph-like inductive bias that correlates all the tokens in a sequence with a pooling operation.

2.2.3 Transformer training

Given a source sequence, the encoder and decoder of the vanilla Transformer model are jointly trained to maximize the conditional probability of a target sequence. The training progress is shown in Algorithm 1, and the use of a trained Transformer for prediction can be found in Phuong and Hutter (2022).

Algorithm 1 Training a simple Transformer model

Require: input sequences x.
Ensure: trained Transformer parameters $\hat{\theta}$.
Initial Transformer parameters θ
for $i = 1, 2, \cdots, N_{\text{epoch}}$ **do**
  for $n = 1, 2, \cdots, N_{\text{data}}$ **do**
    $l \leftarrow \text{length}(x_n)$
    $p(\theta) \leftarrow \text{Transformer}(z_n, x_n|\theta)$
    $\text{loss}(\theta) = -\sum_{t=1}^{l-1} \log p(\theta)[x_n[t+1], t]$
    $\theta \leftarrow \theta - \eta \nabla \text{loss}(\theta)$
  end for
end for
return $\hat{\theta} = \theta$

3 TransRL methods

Reinforcement learning (RL) is currently thriving and has been applied successfully in many tasks. In this section, we provide insight into the advantages and disadvantages of combining Transformer with various RL tasks, such as offline RL, online RL, hierarchical RL, meta-RL, and multi-agent RL (MARL). We abstract several basic frameworks of TransRL in Fig. 6, where ①–⑨ represent Transformer-based offline RL, Transformer-based online RL (offline pretraining and online fine-tuning), Transformer-based hierarchical RL, general TransRL, Transformer-based meta-RL (pretraining and prompt-tuning), Transformer-based MARL based on QMIX (mixing network), Transformer-based MARL based on QMIX (agent network), Transformer-based MARL based on actor–critic, and Transformer-based MARL, separately.

3.1 Transformer-based offline reinforcement learning

Because of the inherited superiority of the vanilla Transformer architecture, recent Transformer-based offline RL works perform better on generative tasks such as next-action prediction. Notably, the encoder–decoder Transformer has been popularized to learn richer representations in offline RL. From the combination mode perspective, we summarize two main ways to associate offline RL with Transformers in recent DM tasks.

3.1.1 Augmenting RL components with Transformer

Transformer has a clear advantage in the ability to capture long-horizon dependencies and surpasses state-of-the-art recurrent memory architectures (such as LSTM and RNN) (Parisotto and Salakhutdinov, 2021), especially in Markovian environments (Parisotto and Salakhutdinov, 2021). A naive combination mode implementation is to apply the Transformer model to represent and augment the components (such as value functions, behavior policies, and dynamics models) in standard RL algorithms. For example, Esslinger et al. (2022) used a decoder-only Transformer in DQN to stabilize Q-learning with data augmentation and replaced the standard CNNs (Hansen et al., 2021). Similarly, inspired by Transformer's superior performance over LSTMs and the growing availability of implementations, Upadhyay et al. (2019) proposed a deep Transformer Q-network (DTQN), which uses an encoder-only Transformer with the input of an agent's history to represent the Q-value. The core steps in designing a TranRL with this combination mode are: (1) Clarify what components in standard RL to represent and augment. Transformer provides a more efficient fit to represent sequential features (shown as the sequences in Tables 2 and 3), and has become a strong substitution of RNN to fit the RL components, such as state values, strategies, and interim elements. (2) Design a specific Transformer form. Transformers use a vast amount of attention toward designing more efficient variant structures where the embedded input sequence passes through various Transformer blocks and projects an output to the action space. In particular, gains from increased Transformer model

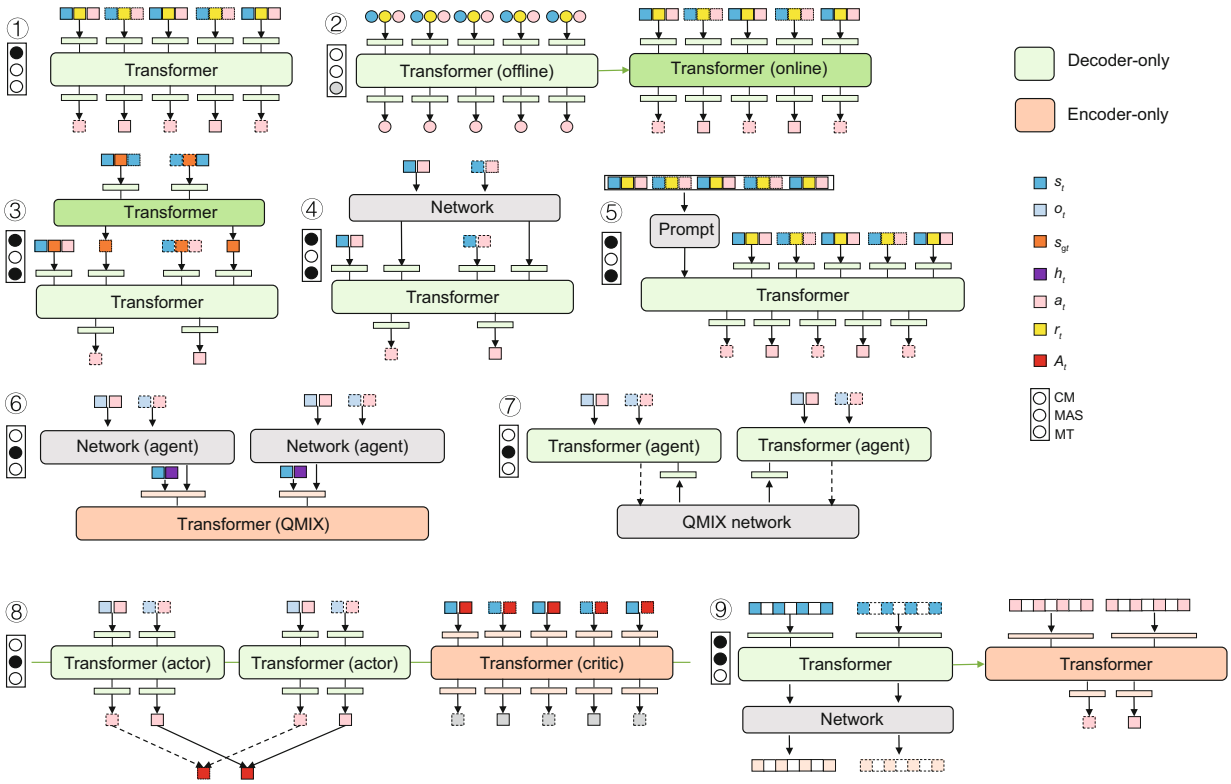

Fig. 6 Typical TransRL frameworks. CM represents the combination mode, white color means replacing the RL component with Transformer, and black color means converting RL to the Transformer framework. Gray color represents that the generality is middle. MAS represents a multi-agent setting. MT represents the framework's generality (or multi-task adaptation). Transformer in orange represents the encoder-only structure and Transformer in green the decoder-only structure. The symbols s_t, o_t, s_{gt}, h_t, a_t, r_t, and A_t denote state, observation, subgoal, history, action, reward, and actions of all agents, respectively. Notably, circle and square of the same color represent different modal data. Dashed and solid geometrical shapes of the same color represent different time-steps. References to color refer to the online version of this figure

scales have yet to saturate. (3) Design the training techniques and paradigm. In this combination model, Transformer updates network parameters in the inner loop, and RL trains hyperparameters in the outer loop. Notably, a downside to Transformer compared to LSTM models is its significant computational cost, especially in challenging memory environments (e.g., partially observable settings).

The bottleneck of this combination method is that the augmented model still optimizes parameters within the RL paradigm (such as Q-learning), which leads to the model still relying on standard RL algorithmic advances to improve its performance.

3.1.2 Converting RL to the Transformer framework

The continual achievements of the Transformer architecture in the NLP and CV domains make it an preferred candidate for RL problems since there are many similarities between supervised learning (SL) and offline RL. Both offline RL and SL are, essentially, statistical processes in which a general function is created by learning from samples. In SL, the function is considered a classifier or predictor, which could be a value function or a policy in RL. Converting RL to the existing Transformer framework (Fig. 6①), which is widely used in language and vision modeling to solve the RL problem, brings simplicity, generalization, robustness, and scalability of large-scale unsupervised learning to RL (Srivastava et al., 2019; Brown et al., 2020; Janner et al., 2021), and avoids issues such as nonstationary learning, bootstrapping, value overestimation (Levine et al., 2020), credit assignment, and discount factors arising in traditional RL. For example, Srivastava et al. (2019) showed an alternative

Table 2 Representative works of Transformer-based multi-agent reinforcement learning for decision-making tasks

Method	Sequence	Technique	Mode	Offline	Online	Benchmark
MADT (Meng et al., 2021)	*o-a* (actor) *s-A* (critic)	Pretrain-finetune, CTDE, actor–critic	D (actor) E (critic)	✓	✓	SMAC
TransMix (Khan et al., 2022)	*s-Q-h*	CTDE	E	✓		SMAC
trans_mix (Wang HB et al., 2023)	*Q*	CTDE	E	✓		SMAC
UPDeT (Hu et al., 2021)	*O*	CTDE	E	✓		SMAC
MAT (Wen et al., 2022)	*O-A*	Multi-agent advantage decomposition	E-D	✓		SMAC, Bi-DexHands, multi-agent MuJoCo, Google Research Football
ATM (Yang YD et al., 2022)	*O-M*	Memory updating schema	E	✓		SMAC
T3OMVP (Yuan Z et al., 2022)	*S*	CTDE	E	✓		Urban multi-intersection environment
T-MAAC (Wang MR et al., 2022)	*O*	Auxiliary-task training	E	✓		MAPDN
DA3-X (Motokawa and Sugawara, 2022)	*o*	DQN, IQN	E	✓		Grid-map

In the Sequence column, s represents the state, a the action, o the observation, h the history, O the observations of all agents in the multi-agent setting, A the actions of all agents in the multi-agent setting, Q the state–action values, and M the memory slots. In the Mode column, E and D represent the encoder and decoder, respectively

upside-down reinforcement learning (UDRL), which primarily sidesteps traditional RL algorithms and uses supervised learning techniques to solve RL problems. Chen LL et al. (2021) showed a simple new framework, DT, which casts the RL problem as a conditional sequence model. In DT, future actions are autoregressively generated by conditioning on sequences of past states, actions, and returns. Details will be introduced in Section 4.1. Wang KR et al. (2022) provided a perspective by viewing offline RL as a generic sequence generation problem, adopting the Transformer architecture to model distributions over trajectories.

The technique of converting DM problems into sequence modeling problems opens a new avenue for solving RL tasks. The great advantage of this combination mode is that the advances in sequence models can be directly applied to RL problems and is not subject to the RL algorithm framework (Lin QJ et al., 2022), and the effectiveness is determined by the representational capacity of the sequence model rather than by algorithmic sophistication. However, as a newly introduced model in the field of DM, Transformer itself still has many shortcomings and needs to be continuously improved, such as by implementing reward design by hand and model failure in a large stochastic environment, which we will introduce in detail in Section 4.1 and Section 6.3.

3.2 Transformer-based online reinforcement learning

Offline RL usually greatly suffers from data inefficiency (Kapturowski et al., 2023) since a standard RL paradigm tends to learn from large-scale offline data, and optimizes neural networks tabula rasa with random initialization. To tackle this issue, the pretrain-finetune paradigm is introduced from the machine learning domain: it actively leverages the foundation model (Bommasani et al., 2021) with transferable knowledge from large-scale pretraining to facilitate downstream tasks. In the context of RL, transferable knowledge generally consists of good representations that facilitate the agent's perception of the world (i.e., a better state space) and reusable skills from which the agent can quickly build complex behaviors given the DM task descriptions (i.e., a better action space) (Lu K et al., 2022). Fig. 7 shows the pipeline of offline pretraining and online fine-tuning.

Table 3 Representative works of TransRL for single-agent decision-making tasks

Method	Sequence	Technique	Mode	Offline	Online	MT	Benchmark
DT (Dosovitskiy et al., 2021)	R-s-a	–	D	✓			Atari, D4RL, Key-to-Door
TT (Janner et al., 2021)	s-a-r	Beam search	D	✓			D4RL
DTQN (Esslinger et al., 2022)	h-o	DQN	D	✓			Gym-Gridverse, Car Flag, Memory Cards, Hallway
DTQN (Upadhyay et al., 2019)	s	DQN	E	✓			OpenAI Gym
ODT (Zheng et al., 2022)	R-s-a	Pretrain-finetune	D		✓		D4RL
Aplor (Reid et al., 2022)	R-s-a	Pretrain-finetune	E-D	✓	✓	✓	Atari, Gym
MineDojo (Fan et al., 2022)	s-r	Pretrain-finetune	E-D	✓	✓	✓	Minecraft game
Gato (Reed et al., 2022)	s-a	Pretrain-finetune, prompt-tuning	D	✓	✓	✓	DM Lab, ALE, BabyAI, DMC Meta-World, Sokoban
HDT (Correia and Alexandre, 2022)	s-s_g (high) s-s_g-a (low)	Subgoal selection	D	✓		✓	OpenAI Gym, D4RL, RoboMimic
TrMRL (Melo, 2022)	$\Phi(s,a,r,\eta)$	–	E	✓		✓	MuJoCo, Meta-World
CMT (Lin RJ et al., 2022)	z-s-a-r	Prompt-tuning	E-D	✓	✓	✓	D4RL, MuJoCo, SMAC
Promt-DT (Xu et al., 2022)	z-s-r-a	Prompt-tuning	D	✓	✓	✓	MuJoCo
TransTSP (Goh et al., 2022)	s	Prompt-tuning	E-D				TSP instances
SwitchTT (Lin QJ et al., 2022)	i-R-s-a	–	D	✓		✓	Gym-mini-grid
SWAT (Hong et al., 2022)	s	TD3	E-D	✓		✓	MTRL
MGDT (Lee et al., 2022)	o-R-a-r	Pretrain-tunning, expert action inference	D	✓	✓	✓	Atari
IL-DT (Pan YW et al., 2022)	a	Imitation learning, rule-based	D	✓		✓	ManiSkill
StARformer (Shang et al., 2022)	g-h_i (long) s-a-r (short)	–	E-D	✓		✓	Atari, DMC
BooT (Wang KR et al., 2022)	s-a-r-R	Data augmentation, teacher-forcing strategy	D	✓			D4RL
CDT (Furuta et al., 2022)	s-a	State-marginal matching	D	✓		✓	Gym, MuJoCo
BDT (Furuta et al., 2022)	s-a	–	D	✓		✓	Gym, MuJoCo
Scene-Rep (Liu HC et al., 2022)	h-a (SLT) s (MST)	SAC	E-D	✓			SMARTS
SPLT (Villaflor et al., 2022)	s-a	–	E-D	✓			CARLA
Trans-REIN (Kool et al., 2019)	s	REINFORCE	E-D				TSP instances
STT (Yang YM et al., 2022)	s	–	E		✓		Gym, MuJoCo, CausalWorld
Catformer (Davis et al., 2021)	–	R2D2	D	✓			Quake III Arena engine
GTrXL (Parisotto et al., 2020)	s	–	D	✓			DMLab-30, Memory Maze
ESPER (Paster et al., 2022)	s-a	–	E	✓			Gambling, Connect Four, 2048 (Cirulli, 2014)
AdA (Bauer et al., 2023)	o-a-r	Meta-RL auto-curriculum learning	E	✓	✓	✓	XLand 2.0

MT represents the multi-task environment. In the Sequence column, R represents the return-to-go, s the state, a the action, r the reward, o the observation, h the history, O the observations of all agents in a multi-agent setting, A the actions of all agents in a multi-agent setting, s_g the subgoal, Q the state–action values, M the memory slots, g StAR (Shang et al., 2022), h_i the outputs of the previous sequence Transformer layer (Shang et al., 2022), η a Boolean flag to identify whether this is a terminal state, z the policy prompt, and i the task id. In the Mode column, E and D represent the encoder and decoder, respectively

Practical instantiation of Transformer-based offline RL involves an online component (Zheng et al., 2022), where a pretrained strategy can be quickly applied to a specific task with online fine-tuning. For example, Online Decision Transformer (ODT) (Zheng et al., 2022) is proposed as a simple and

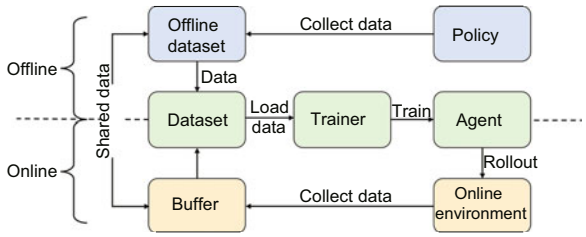

Fig. 7 Pipeline of offline pretraining and online fine-tuning

robust algorithm for fine-tuning a pretrained DT in an online setting. To explore the generality of pretraining, Reid et al. (2022) explored the transferability of pretrained sequence models on SL domains (e.g., NLP and CV) when finetuned on different tasks (e.g., offline RL tasks), where the alignment techniques between the language representations and offline RL elements allow Transformer to handle language and trajectories simultaneously. Fan et al. (2022) trained a large pretrained video–language model with Internet-scale knowledge from Minecraft videos, tutorials, Wiki pages, and forum discussions, which benefits to suit thousands of diverse open-ended tasks. Pretraining is helpful in improving data efficiency and model generality, which highlights the potential of leveraging generic sequence modeling techniques and pretrained models for RL. In particular, pretraining in different modalities on Transformer-based offline RL assists in building a universal computation engine (Lu K et al., 2022). For example, relying on the pretrain-finetune paradigm, Gato (Reed et al., 2022) learns a multimodal, multi-task, multiembodiment generalist policy on various tasks from control environments, vision datasets, and language datasets in a supervised manner. We generalize this pretraining TransRL as a general structure shown in Fig. 6②.

Pretraining is an essential technique for avoiding higher computational costs from using more expressive models such as Transformers (Reid et al., 2022), which is also helpful for MARL problems since online exploration in multi-agent settings may not be feasible in many scenarios. Extending existing pretraining techniques to the multi-agent scenario is nontrivial. Meng et al. (2021) explored the use of DT in the context of MARL and proposed a novel architecture, Multi-Agent Decision Transformer (MADT). However, training data tend to be the main bottleneck for effective RL pretraining. Unlike pretraining in NLP and CV fields where a wealth of unlabeled data can be collected with minimum supervision, RL usually requires a highly task-specific reward design to label training data, which hinders the scaling up of pretraining for large-scale applications. Another issue raised in multimodel and multi-task settings is that these pretraining models might suffer from detrimental gradient interference (Yu TH et al., 2020a) between the various modalities and tasks due to the incurred optimization challenges. To mitigate this issue, Xie et al. (2022) provided some directions.

3.3 Trasnformer-based hierarchical reinforcement learning

Long-horizon tasks (Pateria et al., 2022) pose a major challenge in standard RL, because learning in large state and action spaces without sophisticated exploration techniques always results in poor performance (Pateria et al., 2022). To address this issue, many researchers have introduced hierarchical reinforcement learning (HRL) to decompose a challenging long-term RL task into a sequence of subtasks in different hierarchies, where a subtask is usually a simple RL problem that could be easier to solve by learning a lower-level policy due to its short horizon. Based on these subtasks, learning a higher-level policy to play the task by choosing optimal subtasks as higher-level actions rewards the performance of HRL. Inspired by HRL's high performance on long-horizon problems, Correia and Alexandre (2022) proposed a Hierarchical Decision Transformer (HDT), in which a high-level planner aims to find a path built with subgoal states that drive the agent toward the main task goal by conditioning a low-level controller to try to achieve each subgoal (as shown in Fig. 8 and Fig. 6③). HDT removes the need for the specification of desired rewards by using demonstration learning to provide an extra supervision signal. Therefore, the high-level Transformer receives sequences of the previous states and subgoals, and tries to predict the next subgoal state in the sequence. The low-level Transformer receives sequences of the previous states, subgoals, and actions and tries to predict the next action in the sequence.

Associating Transformer with HRL enriches the scope of Transformer-based RL domains and mitigates the reward design issue in the original DT. However, more Transformer-based HRL methods are encouraged for various scenarios, such as multi-task and multi-agent settings. Additionally, there

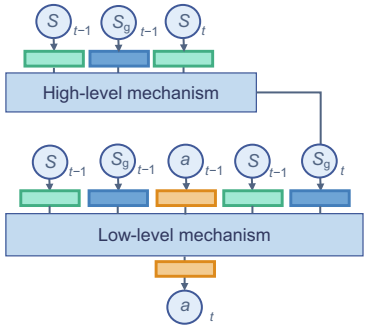

Fig. 8 Structure of the Hierarchical Decision Transformer (HDT)

are many key issues to be mitigated in single-agent RLs, such as subgoal selection, subtask representation (Parr and Russell, 1997), and nonstationarity (resulting from learning multiple levels of policies simultaneously) (Levy et al., 2019).

3.4 Transformer-based multi-agent reinforcement learning

Multi-agent reinforcement learning (MARL) (Yang YD and Wang, 2020) is an attractive problem due to its grand challenges (Gronauer and Diepold, 2022) such as partial observability, cooperation, credit assignment, and nonstationarity, which arise from agent learning to coordinate their behavior to benefit the whole team while conditioning only on each observation (Samvelyan et al., 2019). Currently, many advanced MARL methods try to mitigate the above challenges, under the training framework of centralized training and decentralized execution (CTDE) (Oliehoek et al., 2008; Lowe et al., 2017), which boosts the developments of methods that directly inherit a single-agent RL, such as COMA (Foerster et al., 2018), MADDPG (Lowe et al., 2017), QMIX (Rashid et al., 2020), and multi-agent proximal policy optimization (MAPPO) (Yu C et al., 2022).

How can MARL studies be enhanced with powerful sequential modeling techniques? Meng et al. (2021) proposed a comprehensive architecture, MADT, which uses Transformer for each agent to fit a joint MARL policy. Inherited from the actor–critic (AC) paradigm, CTDE consists of n Transformer-based actor networks and a Transformer-based critic network (as shown in Fig. 9, and we generalize it into a general structure shown in Fig. 6⑧). Different from a single-agent RL, MADT considers both globally shared and private information in policy evaluations.

Credit assignment creates challenges for agents in determining whether it is the behavior selection itself that obtains the rewards or another agent's behavior. The existing works on MARL focus primarily on building a centralized Q-function to guide the training of the individual value function (Rashid et al., 2020; Yang YD et al., 2020). TransMix (Khan et al., 2022) associates Transformer with QMIX (Rashid et al., 2020) to learn a richer mixing function for combining the agents' individual value functions (Fig. 6⑥), which mitigates the credit assignment problem by using a Transformer mixing network to factorize joint action values into individual action values for every agent. Wang HB et al. (2023) showed a similar framework with a different specific mixing network design based on Transformer (as shown in Fig. 9, and we generalize it into a general structure shown in Fig. 6⑦). Unlike the above action-value function methods, UPDeT (Hu et al., 2021) replaces the RNN-based component in the individual value function with Transformer to optimize the policy at an action-group level and fits tasks with different observations and action configuration requirements, which offers significant improvements on the transfer capability of a multi-agent system, especially in hard and complex multi-agent tasks.

Unlike the above studies that just replace partial components in MARL methods with Transformer to improve one certain type of performance, Wen et al. (2022) used the multi-agent advantage decomposition theorem to transform multi-agent joint policy optimization into a pure sequence model. They conducted some pioneering work and proposed an MARL model, multi-agent Transformer (MAT) (as shown in Fig. 11, and we provide a general structure in Fig. 6⑨), which is characterized by treating a team of agents as a sequence and guided by the multi-agent advantage decomposition theorem to ease the reward assignment problem.

Some other researchers make initial explorations involving partial observability and generality in MARL problems. For example, Yang YD et al. (2022) alleviated the partial observability focus on the working memory updating schema and action parsing, and provided an agent Transformer memory (ATM) network to calculate individual Q-values or policy logits.

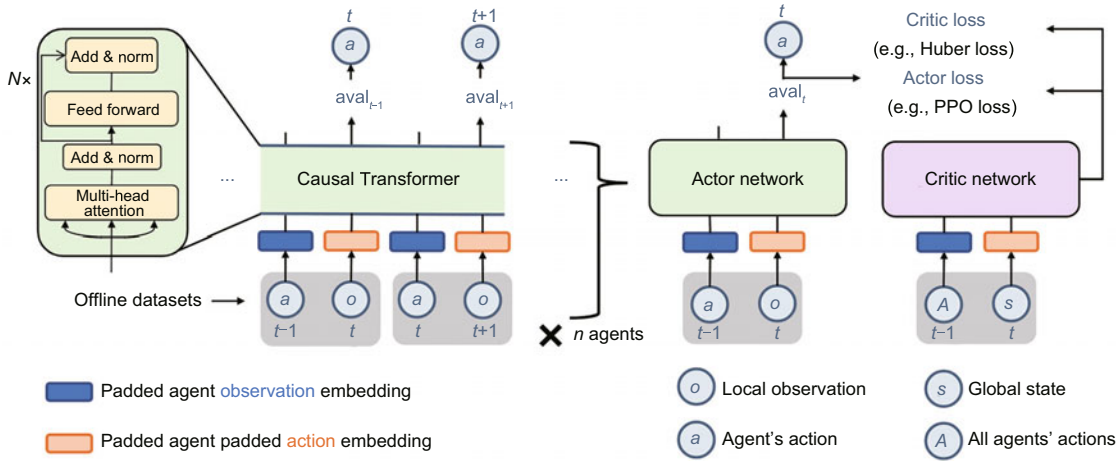

Fig. 9 Detailed model structure for offline and online Multi-Agent Decision Transformer (MADT) (PPO: proximal policy optimization) (Meng et al., 2021)

Associating Transformer with the current advanced MARL boosts the performance in multiple dimensions. However, successful developments of current Transformer-based MARL methods that directly inherit single-agent TransRL methods may lack convergence guarantees in the MARL setting. Finally, we provide a summary of Transformer-based MARL methods in Table 2.

3.5 Transformer-based meta-reinforcement learning

Standard meta-reinforcement learning (meta-RL) aims to train the agent in a set of training tasks to learn a sufficiently powerful strategy that can quickly adapt to new unseen tasks in independent and identically distributed or out-of-distribution (OOD) tasks (Yu TH et al., 2020b).

Inherited meta-learning (Hospedales et al., 2022), a branch of meta-RL, is developed from memory-based architectures (Ortega et al., 2019), such as RL^2 (Duan et al., 2016) and E-RL^2 (Stadie et al., 2018). The recently rising attention-based architectures excel at reasoning and can recognize how situations are related to the multi-head attention mechanism. One natural idea is to use them as a replacement for RNNs. Melo (2022) developed TrMRL, which integrates the Transformer structure with a memory recovery mechanism, associates previous tasks, dynamically represents a task, and creates a memory sequence using layers of recursion. TrMRL is a pioneering work that presents substantial improvements in associating Transformer with meta-RL. However, the generality of TrMRL is encouraged to be improved because its verification task is similar to the training task. Pinon et al. (2022) developed a model-based algorithm, in which a Transformer encoder was used to fit the symbolic environmental dynamics (Wang J et al., 2021), and applied an online planner to the learned model.

How can we unlock the potential of the Transformer structure to entirely adopt OOD tasks rather than simply replacing Transformer with a component of meta-RL? A collection of methods have been implemented to enable generalizations to unseen work. For example, Lin RJ et al. (2022) proposed a novel Transformer-based meta-RL algorithm, named Contextual Meta Transformer (CMT), in which a pre-training and fine-tuning paradigm is used to solve offline RL problems in the offline setting. CMT aims to conquer multiple tasks and generalizations in one shot in the offline setting from the perspective of sequence modeling. Introducing the prompt-based framework from NLP and adapting it to the context of offline RL, Prompt-based Decision Transformer (Prompt-DT) (Xu et al., 2022) (the framework is shown in Fig. 6⑤) incorporates the advantages of the Transformer architecture in sequential modeling and the prompt framework in few-shot adaptation, achieving excellent generality in offline meta-RL, which is analogous to supervised learning with the pretraining and fine-tuning paradigm. Comparison of the standard meta-RL and offline meta-RL refers to Mitchell et al. (2021).

In the context of MARL, associating Transformer with the conventional meta-RL is a promising way to generalize to OOD environments. However, note that adapting Transformer with a specific paradigm to RL problems is nontrivial. For example, it is easy to pretrain language models and to learn adequate knowledge from the pretraining corpus. In RL, it is questionable whether a pretrained model has enough knowledge to solve an unforeseen task because of the inherent differences between the different tasks (Xu et al., 2022). Finally, we provide a summary of TransRL for single-agent DM tasks in Table 3.

4 Representative TransRL models

This section is aimed to provide insights into the details of three representative implementation instances of TransRL models and to discuss their characteristics, benefits, and limits. The goal here is not to exhaustively detail all TransRL models but to cover a series of representative samples of TransRL models.

4.1 Decision Transformer

DT (Dosovitskiy et al., 2021) is the first successful application of the Transformer model to DM tasks, which shifts the focus to sequential modeling and converts offline RL into pure supervised learning tasks. DT avoids the need for computing cumulative rewards (including the discount factor γ) through dynamic programming, and it also avoids the stability issues related to bootstrapping for long-term credit assignment and reward sparsity. Unlike traditional RL, DT models the reward as the returns-to-go $R_t = \sum_{t'=t}^{T} r_{t'}$ instead of the expected return in an MDP. A trajectory for DT is represented as $\tau = (R_1, s_1, a_1, R_2, s_2, a_2, \cdots, R_T, s_T, a_T)$, in which the visual states are usually an encoder with a convolution operation. DT employs a Transformer architecture (Fig. 10) that consists of stacked self-attention layers with residual connections. In the training loop, $3K$ tokens (K for each modality: return-to-go, state, or action) will be sampled from a dataset of offline trajectories and fed in DT with a same-time-step embedding which is a slight difference from a standard positional embedding. The training objective is to minimize the mean square error (MSE) loss between the ground truth action a_t and predicted action $\hat{a}_t$, that is, $\mathcal{L}(\theta) = \frac{1}{N}\sum_i^N (\hat{a}_t - a_t)^2$. In the testing loop, DT interacts with the environment with the predicted action $\hat{a}_t$, which is generated based on input conditioning information.

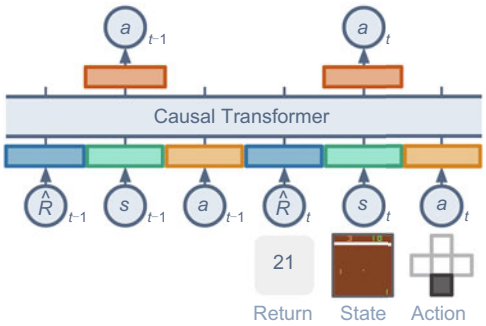

Fig. 10 Structure of the original decision Transformer (Chen LL et al., 2021)

Benefiting from the structure of Transformer, DT achieves a good generalization ability on Atari tasks. Siebenborn et al. (2022) replaced Transformer with an LSTM model while keeping the other components unchanged, and performed ablation experiments to show the strong ability in the DT inherited from Transformer in the same environments. However, DT loses a stitching ability, which is an important ability for an offline RL to learn the optimal policy from suboptimal trajectories, and is significant particularly when the offline dataset collects only suboptimal trajectories. Yamagata et al. (2023) addressed the shortcomings of DT by leveraging the benefits of dynamic programming (such as Q-learning). Another drawback of DT is that goal (e.g., return-to-go) needs to be engineered at hand since DT ignores the instant reward r signal. To explicitly mitigate this issue, Shang et al. (2022) proposed a novel model, StARformer, based on the original DT, which could easily operate on the stepwise reward (that is, the immediate reward generated by an environment in each step), since the Step Transformer in StARformer operates on "*s*-*a*-*r*" tokens in the short term and Sequence Transformer in StARformer operates on learned intermediate StAR-representation in the long term. To implicitly address this issue, Furuta et al. (2022) developed a new Transformer structure to avoid the need for complicated reward engineering and extracted a learning signal from each trajectory data.

4.2 Multi-agent Transformer

MAT (Wen et al., 2022) is a pioneering work that extends TransRL to the multi-agent setting. Unlike previous works that focus on replacing the component in MARL with Transformer, MAT leverages the multi-agent advantage decomposition theorem (Kuba et al., 2021) to entirely transform the joint policy optimization problem into a sequential DM process, and designs an encoder–encoder Transformer structure (Fig. 11) to enhance the parallelization for proximal policy optimization (PPO). In this subsection, we discuss the key techniques behind MAT and provide a promising direction for combining MARL with Transformer models.

Markov games provide an analysis tool for cooperative multi-agent DM problems; however, they are unsuitable for sequence-to-sequence methods since all agents take actions simultaneously based on their observation without sequential dependencies. The multi-agent advantage decomposition theorem plays a key role in building a surprising connection between MARL and sequence models, which guarantees that the joint advantage function can be decomposed into a summation of each agent's local advantages. Underpinned by the multi-agent advantage decomposition theorem, it is trivial to convert a Markov game to a multi-agent sequential decision model, in which agents take actions by following a sequential order, and each agent considers decisions from the preceding agents. This sequential decision progress is similar to DT in a single-agent setting, which provides a new angle to solve the MARL problem by using a Transformer model. As shown in Fig. 11, the Transformer encoder block learns expressive representations of the joint observation, and is followed by multilayer perceptron (MLP) to fit the value functions. The encoder outputs actions for each agent in an autoregressive manner.

In training steps, the encoder is updated by the temporal difference error (Sutton and Barto, 2018):

$$\mathcal{L}_{\text{Encoder}}(\phi) = \frac{1}{Tn}\sum_{m=1}^{n}\sum_{t=0}^{T-1}\left[R\left(o_t,a_t\right)+\gamma V_{\bar{\phi}}\left(\hat{o}_{t+1}^{i_m}\right)-V_{\phi}\left(\hat{o}_t^{i_m}\right)\right]^2. \tag{7}$$

The decoder is trained by minimizing the following clipping PPO (Schulman et al., 2017) objective:

$$\mathcal{L}_{\text{Decoder}}(\theta) = -\frac{1}{Tn}\sum_{m=1}^{n}\sum_{t=0}^{T-1}\min\left(r_t^{i_m}(\theta)\hat{A}_t,\text{clip}\left(r_t^{i_m}(\theta),1\pm\epsilon\right)\hat{A}_t\right), \tag{8}$$

where

$$r_t^{i_m}(\theta)=\frac{\pi_{\theta}^{i_m}\left(a_t^{i_m}\mid\hat{o}_t^{i_{1:n}},\hat{a}_t^{i_{1:m-1}}\right)}{\pi_{\theta_{\text{old}}}^{i_m}\left(a_t^{i_m}\mid\hat{o}_t^{i_{1:n}},\hat{a}_t^{i_{1:m-1}}\right)}. \tag{9}$$

MAT leverages the multi-agent advantage decomposition theorem to bridge MARL with the sequence model, which brings some remarkable advantages: (1) Sequence modeling reduces the complexity growth of the MARL problems with increase in

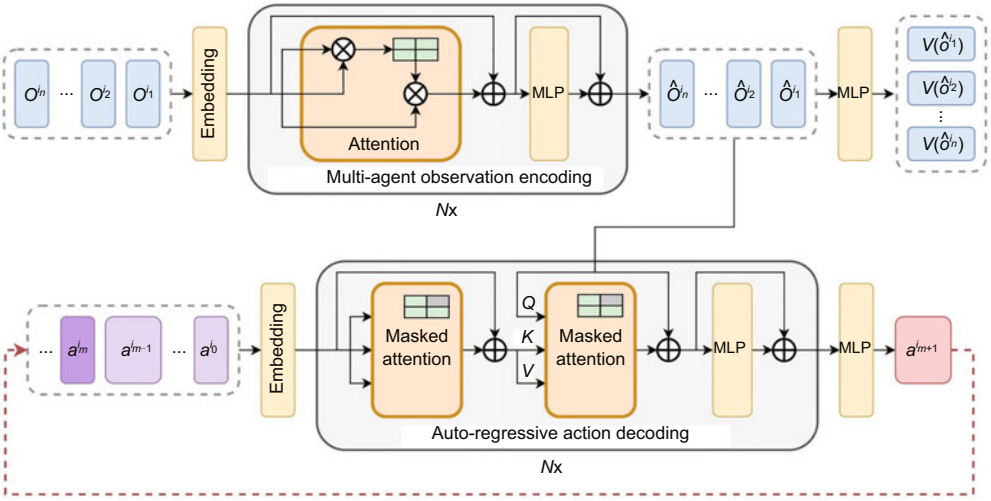

Fig. 11 Structure of multi-agent Transformer (MLP: multilayer perceptron) (Wen et al., 2022)

the number of agents from multiplicative to additive, thus rendering linear complexity; (2) By treating a team of agents as a sequence, the Transformer architecture allows us to model teams of agents with variable numbers and types while avoiding the drawbacks of MAPPO or heterogeneous-agent PPO (HAPPO).

4.3 Gato

Gato is a general-purpose agent for a wide range of tasks (exactly 604 distinct tasks), such as video games (Bellemare et al., 2013), simulated control, chat, caption images, and block stacking with a real robot arm. The remarkable performance of Gato is attributed to the large-scale Transformer model (with 1.2 billion parameters) and the expensive computation, which costs 4 d with 256 tensor processing units (TPUs). In this subsection, we introduce the key techniques of Gato and discuss its benefits and limits. Gato is characterized by sensing and acting with multimodal and multiembodiment (Reed et al., 2022), and learning a generalist policy that can handle multiple tasks by a single network with the same weights. The backbone of Gato is a large decoder-only Transformer with 24 layers, an embedding size of 2048, and a post-attention feed-forward hidden size of 8196. In the training phase, Gato uses different tokenization methods (Kudo and Richardson, 2018; Dosovitskiy et al., 2021) to process diverse modality data, including text, images, discrete values, and continuous values. Note that Gato adds position (including temporal and spatial information) by a parameterized embedding function, which takes different operations on different modalities. For output targets, each token is potentially a target label on the condition of the previous tokens under autoregressive training. Additionally, the target labels could be obtained by prompting with expert demonstrations, which is useful in helping Gato infer the relevant tasks from the observations and actions in the prompt. Given a sequence of tokens $s_{1:L}=\{s_1,s_2,\cdots,s_L\}$, the training loss for a batch $\mathcal{B}$ is

$$\mathcal{L}(\theta,\mathcal{B})=-\sum_{b=1}^{|\mathcal{B}|}\sum_{l=1}^{L}m(b,l)\log p_\theta\left(s_l^{(b)}\mid s_1^{(b)},s_2^{(b)},\ldots,s_{l-1}^{(b)}\right), \tag{10}$$

where $m(b,l)\to\{0,1\}$ is a masking function.

Gato handles tasks across several domains (such as NLP, CV, and DM), which shows an inspiring prospect for reducing the need for handcrafting policy models. However, Gato relies on extremely large diverse modality datasets (which require several weeks or even months of data collection), where the main challenge is the lack of available data. Natural language or image datasets are relatively easy to collect from the Web; however, a Web-scale dataset for control tasks is currently problematic, especially when scaling Gato to a higher number of parameters. Another obvious limit of Gato is that it is computationally expensive; we will discuss this issue in detail in Section 6.2.

5 Applications

Based on the familiar RL applications (Xiang XC and Foo, 2021) (Table 4), here we review how TransRL has been exploited in gaming AI, robotics, transportation, and computer systems. We summarize the applications of TransRL in video games, robotic manipulation, robotic navigation, autonomous driving, and combinational optimization.

Table 4 Existing applications of TransRL (TransRL ⊆ RL ⊆ Domain)

Domain	RL	TransRL
Gaming AI	Board games, card games, video games	Video games (open-ended games)
Robotics	Sim-to-real, control	Robotic manipulation, robotic navigation
Transportation	Traffic control	Autonomous driving
Computer systems	Resource assignment, security	Combinatorial optimization

5.1 Gaming artificial intelligence

In the RL community, games are ideal benchmarks with features that have simple rules and well-defined boundaries (Yuan WL et al., 2021). Gaming AI is referred to as the drosophila of AI (Omidshafiei et al., 2020), which is an open problem that drives research on the AI frontiers. In recent years, gaming AI has achieved superhuman performance in DM games (Silver et al., 2017a; Jaderberg et al., 2019; Vinyals et al., 2019; Badia et al., 2020; Li JJ et al., 2020; Zha et al., 2021; Zhao EM et al., 2022), such as card

games (e.g., Texas hold'em poker, DouDizhu, and Bridge), board games (e.g., Go), and video games (e.g., Gym-like game (Brockman et al., 2016), Atari, StarCraft II, Quake III Arena, and Google Research Football (Kurach et al., 2020)). There have been few studies of TransRL in board games and card games, and researchers currently focus on video games, since video games are characterized by real-time interaction, data generation with low cost, high-level visual inputs, and long-time horizon DM, being a perfect application instance of TransRL. However, in principle, there is no reason why TransRL models could not be trained with either board games or card games. TransRL (Reed et al., 2022) is currently one of the state-of-the-art algorithms in Gym-like and Atari games (participating type games with near state-of-the-art performance) in the single-agent setting. However, for complex multiplayer games, TransRL can be applied only in partial scenarios or some subtasks, rather than by playing the large-scale full game. For example, current Transformer-based MARL methods (Meng et al., 2021; Khan et al., 2022; Yang YD et al., 2022; Wang HB et al., 2023) focus solely on micromanagement in StarCraft II, which is concerned with the fine-grained control of individual units instead of a high-level strategy. One of the main reasons is that the large-scale adversarial game brings great practical challenges involving the nontransmit game (Czarnecki et al., 2020). However, not all games are bounded by rules. For example, Minecraft (Guss et al., 2019) is an open-ended game with a sandbox design. Current TransRL algorithms (Reid et al., 2022) take a revolutionary application in Minecraft, which is beneficial to drive AI to express creative behavior by building structures that adapt to the world around it (Barthet et al., 2023). Like Minecraft, XLand (including versions 1.0 and 2.0) (Open Ended Learning Team et al., 2021) is a productive game that consists of a huge variety of tasks in complex worlds with embedded hidden production rules. DeepMind provides an adaptive agent (Bauer et al., 2023) in XLand, which was created on a large Transformer model AdA with meta-RL. We summarize the applications of TransRL in gaming AI in Table 5.

Table 5 Applications of TransRL in gaming AI

Type	Game	SA	MA	TransRL
Board game	Go, Chess, Shogi	✓		–
Card game	Texas hold'em poker,		✓	–
	DouDizhu, Bridge	✓	✓	–
Video game	Gym-like, Atari,	✓		✓
	StarCraft II,		✓	✓
	Quake III Arena,		✓	✓
	Google Research Football		✓	✓
Open-ended game	Minecraft, XLand	✓	✓	✓

SA: single-agent setting; MA: multi-agent setting

5.2 Robotics

Robotic tasks are characterized by high-dimensional, continuous states and actions (Singh et al., 2022). The wide application of TransRL in robotics focuses mainly on video/image/language-conditioned robotic manipulation and navigation. Robotic manipulation (Fig. 12) requires a robot move an object to a location relative to another object (Pan C et al., 2023), with visual perception or language understanding as the major input to make decisions in the simulator or physical environment (Boularias et al., 2015; Oh et al., 2015; Duvallet et al., 2016). For example, a cooking robot may need to place a lasagna in an oven. Key skills for robotic manipulation include understanding and placing objects in task-specific locations and generalizing to novel objects (such as taking new steak off the grill or sweeping the beans into the new red dustpan). Traditional approaches for handling robotic manipulation tasks try to divide the problem into subtasks such as Assistive Tele-op (Clever et al., 2022), legged locomotion (Yang RH et al., 2022), and grasping (Han YH et al., 2021), and solve each subtask independently (Sanchez et al., 2022). Currently, many new approaches for robotic manipulation have been proposed, aiming to learn end-to-end policy mapping from high-level visual observations to low-level robot actions, among which TransRL is one of the most promising and effective methods (Xu et al., 2022). The original DT (Dosovitskiy et al., 2021) is the first TransRL method applied to simple robotic manipulation tasks, such as the key-to-door scenario (Mesnard et al., 2021). Based on the original DT, TransRL-related studies extend to meta-RL (Melo, 2022), MARL (de Witt et al., 2020), and even general agent settings (Reed et al., 2022). Furthermore, some advanced TransRL algorithms consider even richer applications, such as intense spatiotemporal coupling states (Yang YM

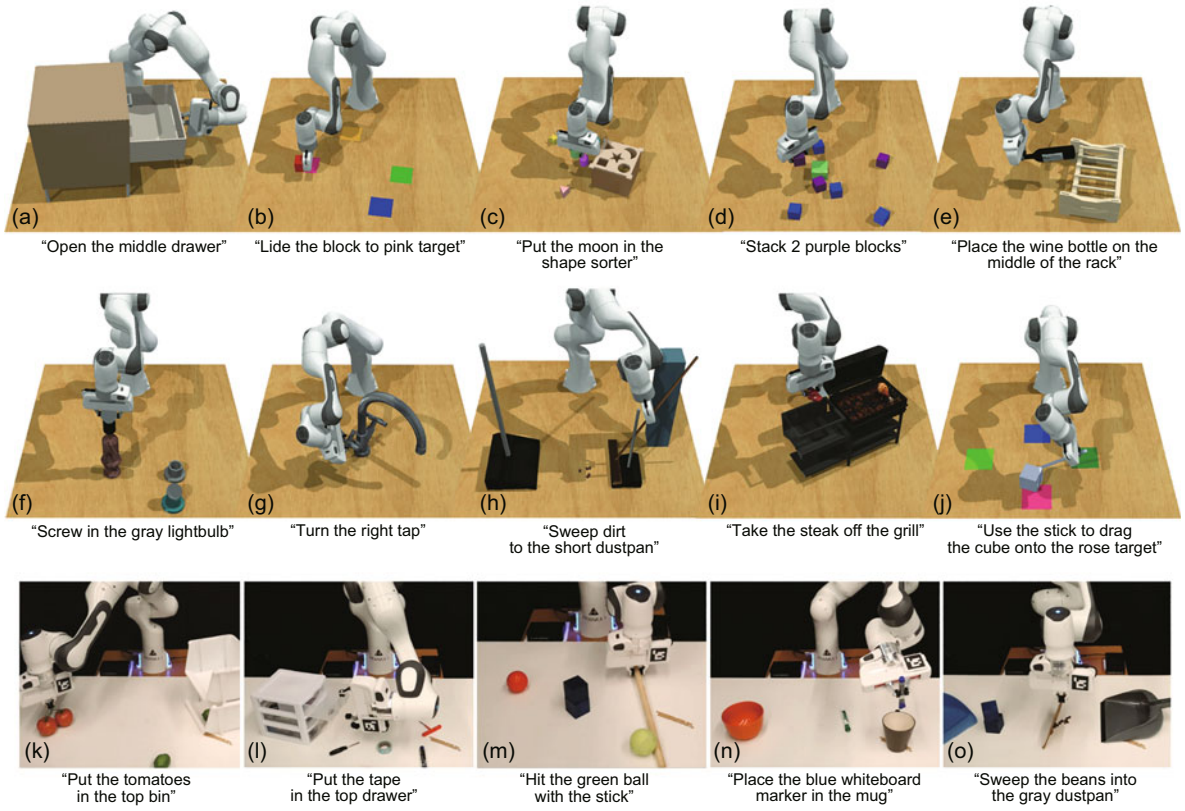

Fig. 12 Illustration on different manipulation tasks (a)–(o) (Shridhar et al., 2023)

et al., 2022), three-dimensional (3D) environments (Hermann et al., 2017; Mees et al., 2022), and imperfect perception, adapting to user preferences (Jain et al., 2023) and language instructions (Guhur et al., 2022).

Traditional robotic navigation can be divided into static tasks and dynamic tasks according to the environmental properties. Static robotic navigation tasks, such as empty outdoor scenarios, require the agent reach the destination while avoiding obstacles in the fixed environment without moving obstacles. In contrast, dynamic scenarios contain interactive objects that are generally inherent to real human environments, such as toys and shoes, which require the robot interact with the environment by pushing or moving the obstacles away to clear the path (Li WY et al., 2022). TransRL has received much attention in natural-language-grounded visual navigation (Fig. 13), an emerging field of robotic navigation, due to its powerful encoding ability on high-level information and multimodal Transformer data. Natural-language-grounded visual navigation contains mainly two research directions, vision-and-language navigation (VLN) (Anderson et al., 2018; Gu et al., 2022) and vision-and-dialog navigation (VDN). In VLN, a robot with first-person views for observations moves to a goal location according to step-by-step natural language instructions. Specifically, the navigation procedure can be viewed as a sequential DM process, where a robot is placed at an initial location (such as in the living room), receives a series of task information in the language instruction form, and then moves to the destination (such as in the kitchen) following the instruction. The given language instruction describes the agent's trajectory in detail, such as "Walk toward the table," "When you get to the bed," and "Turn left and exit the room." Similar to VLN, VDN considers more complex scenarios where robots need to interact with humans; the interactions include understanding human instructions and active inquiries. Owing to the limitations of collecting real-world data and the safety concerns with real robots (Zhao WS et al., 2020), robotic simulators are popular in training agents, such as Gym, ManipulaTHOR (Ehsani et al., 2021), dm_control (Tunyasuvunakool et al.,

2020), SAPIEN (Xiang FB et al., 2020), CausalWorld (Ahmed et al., 2021), and RLBench (James et al., 2020), which greatly accelerate the development of manipulation and navigation methods. However, TransRL focuses on applications in simulators, and few TransRL studies currently explore the simulation-to-reality (sim-to-real) (Zhao WS et al., 2020) transfer process, which leads to a gap between the simulated and real worlds. For example, employing excessive force to real objects might cause elastic deformation and even damage, while grasping can flop with a lack of force. This gap degrades the performance of trained models when they are transferred into real robots.

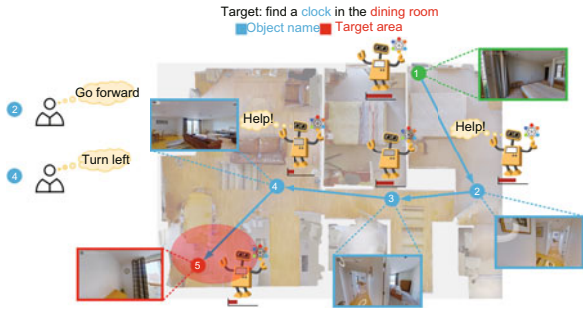

Fig. 13 An illustration of natural-language-grounded visual navigation tasks (Li X et al., 2022)

5.3 Transportation

In urban driving scenarios, a major challenge for DM arises from the stochastic nature of interactive traffic participants and the complexity of road structures (Yurtsever et al., 2020; Liu HC et al., 2022). Many TransRL-related methods outperform traditional RL with the powerful representation ability of Transformers, especially when dealing with multimodal data. For example, Liu HC et al. (2022) developed a two-layer encoder TransRL framework. The Multi-Stage Transformer (MST) was first used to extract latent features (e.g., latent agent interactions) from a multimodal scene representation (including map information, neighbor information, and dynamic interactions among agents). The Sequential Latent Transformer (SLT) follows, aiming to capture latent agent interactions by a self-supervised learning method. MST and soft actor–critic (SAC) (Haarnoja et al., 2018) are used to make driving decisions during the inference phase, while SLT is employed only to speed up training in the training phase. Yuan Z et al. (2022) considered a multivehicle pursuit (MVP) problem under the background of the Internet-of-Vehicles system (Mohamed et al., 2021; Wu TH et al., 2021). The MVP game arises from a real-world application of the police department's pursuit of suspicious vehicles, where multiple vehicles work together to capture mobile targets, which is usually characterized by partial observations and multi-agent dynamic interactions. Yuan Z et al. (2022) proposed a Transformer-based time and team reinforcement learning scheme (T3OMVP) to solve MVP, in which QMIX is used to address multi-agent reward assignments and the agents are trained under the CTDE paradigm. Notably, Transformer-based methods jointly model the states and actions as a pure generic sequence generation problem, and try to disentangle the effects of the policy and environment dynamics on the return. Thus, in adversarial or stochastic environments, these methods lead to overly optimistic behaviors, which tend to be unsafe and unreliable in critical systems such as autonomous driving. However, in autonomous driving, which is a safety-critical domain, understanding the stochastic environment is critical for safe and robust autonomous driving. The abovementioned TransRL methods act optimistically without considering the uncontrollable factors in the environment, which leads to an entanglement between the effectiveness of the policy and stochastic factors on the outcomes. Villaflor et al. (2022) attempted to address this optimism bias by proposing a separated latent trajectory Transformer (SPLT Transformer), which uses two separate policies and world variational autoencoder (VAE) models to efficiently perform a robust search at the test phase.

5.4 Computer systems

In the computer systems domain, we focus on combinatorial optimization problems as the main applications of TransRL. Many industry problems are combinatorial by nature (Bresson and Laurent, 2021). Combinatorial optimization (Kool et al., 2019; Vesselinova et al., 2020; Wang Q and Tang, 2021) aims to find an optimal object from a finite set of objects under constraint conditions (Deudon et al., 2018), which usually leads to NP-hard problems (Hartmanis, 1982). Classical combinatorial optimization problems include the mixed-integer linear program (MILP) (Wolsey, 2020), traveling salesman problem (TSP) (Bello et al., 2017), vehicle routing

problem (VRP) (Toth and Vigo, 2014), and orienteering problem (OP) (Golden et al., 1987). Approaches to these combinatorial optimization problems can be divided into exact methods and heuristic (or approximate) methods (Anbuudayasankar et al., 2014; Kool et al., 2019). We take an example to illustrate the differences between these two classes of methods. In TSP, the best-known exact method is the dynamic programming (DP) algorithm, which guarantees optimal solutions. However, DP has a complexity of $O(2^n n^2)$ for TSP, which makes it infeasible to scale up to large instances ($n > 40$). Nevertheless, handcrafted heuristics that search the space of feasible solutions in an efficient manner, making a trade-off between optimality and computational cost, can provably handle optimality symmetric TSP instances with thousands of city nodes (Deudon et al., 2018). However, the solution from scratch with hand-crafted features and human-engineered heuristics is difficult to transfer to similar tasks. With the development of DNNs, many approaches with general-purpose frameworks have been proposed and successfully applied in combinatorial optimization problems, among which TransRL is a successful approach that is able to learn heuristics to produce high-quality solutions. Mazyavkina et al. (2021) outlined the recent advancements in applying RL to combinatorial optimization problems; the advancements consist mainly of an encoder architecture (e.g., RNN, LSTM, and attention) to represent the state and a learned decoder module (e.g., DQN, PPO, and REINFORCE (Zhang et al., 2021)) to find the solution. Bresson and Laurent (2021) adapted a Transformer architecture to the TSP with RL training (denoted as Trans-TSP), which casts TSP as a translation problem with quadratic complexity $O(n^2 L)$. Trans-TSP encodes the input (2D cities) in the same way as the source language in translation tasks, and the target label is a tour with the minimum length. Wu YX et al. (2022) designed a Transformer-based network to learn to choose nodes for the 2-Opt heuristics, in which the learning was under an actor–critic framework. Deudon et al. (2018) focused on TSP, using Transformer to encode cities and decode the probability of a tour, and heuristic learning was performed by REINFORCE. Kool et al. (2019) replaced RNN with an encoder–decoder Transformer and achieved better performances on both TSP and CVRP.

6 Challenges and open problems

6.1 Stability and structure optimization

Although TransRL's remarkable performance has been attributed mostly to the ability of Transformer to effectively model long-range dependencies, understanding the fundamental mechanism behind the success of Transformer networks and guiding the design of a stable network are still open problems, likely due to the highly complex and non-convex structure of Transformer networks. Guided by general principles in deep learning, solid empirical studies (Voita et al., 2019; Takase et al., 2022) have analyzed the underlying factors (e.g., attention mechanism, feed-forward network, and layer normalization) behind Transformer's success. For example, Liu BY et al. (2021) analyzed the loss landscape and optimization of attention models, and discussed how regularization, concentration on attention, and over-parameterization in attention weight matrices can further improve the attention model. Vashishth et al. (2019) discussed the relationship between the explainability of attention weights and the model outputs. Dong et al. (2021) provided new insights into the operation and inductive bias of networks built by stacking multiple self-attention layers. Furthermore, some works developed tools to understand the inner workings of Transformers from other angles. Ergen et al. (2022) tried to improve the understanding and optimization of the Transformer networks form the perspective of convex optimization. Davis et al. (2021) introduced the concept of sensitivity to guide a stable Transformer design. Effective analysis tools contribute to designing more robust future models. Conversely, a blind design easily leads to instability in training. Liu LY et al. (2020) tried to explore the factors that complicate Transformer training from the perspective of the structure of Transformer; they compared the training of Transformer with different layer norm positions (Post-LN and Pre-LN variants (Baevski and Auli, 2018; Xiong et al., 2020)). Davis et al. (2021) introduced a new sensitivity concept to measure stability, which is a function of architectures that measure the effect of random parameter perturbations on the output variance. Transformer is picky in configurating the model, optimizer, and other hyperparameters, resulting in the standard Transformer architecture being difficult to optimize in a supervised learning setting, which becomes

especially pronounced with RL objectives. There is still lack of research on understanding the difficulty of TransRL, guiding the design of an effective model and suitable training frameworks. Further improvements of the TransRL structure may lie in the following potential directions: (1) theoretical analysis of TransRL ability—exploring the theoretical reason why TransRL trained with sufficient data has better performance than deep RL with RNN networks, will be beneficial to design the TransRL structure; (2) global interaction mechanism design—better global interaction mechanisms with a lightweight architecture beyond attention might be alternative approaches worth exploring.

6.2 Expensive memory and computation

Recent research shows a trend toward large-scale Transformers. The capacity has substantially increased from millions of parameters (Devlin et al., 2019; Conneau et al., 2020) to billions (Radford et al., 2019; Brown et al., 2020; Raffel et al., 2020; Wang HY et al., 2022) and even trillions (Du et al., 2022). Empirical evidence indicates that the scaling of model parameters is beneficial for model performance improvements (Kaplan et al., 2020). For example, the original Transformer has only millions of model parameters, while Gato has 1.2 billion model parameters with superior performance. However, such a large-scale model suffers from instability, expensive memory, and high computational costs. Some works on scaling Transformers focus on inference with better parallelization or optimization of the network to improve training. Miao et al. (2022) focused on data and model parallelism and presented a novel automatic parallel Transformer training system, Galvatron, over multiple graphical processing units (GPUs). To stabilize extremely deep Transformers, Wang HY et al. (2022) proposed a new normalization function to modify the residual connection in Transformers, which plays a key role in scaling up the network to 1000 layers. Ma et al. (2022) provided an open-source toolkit TorchScale (https://aka.ms/torchscale) that allows efficient and effective scaling up of Transformers. Yao et al. (2022) proposed a layerwise token-dropping method (Random-LTD), which skips the computation of a subset of the input tokens at all middle layers. Also, an advanced training method in CV is worthy of consideration to solve the training issue. Li CL et al. (2022) explored an efficient training method, which accelerates Vision Transformer (ViT) training by introducing progressive learning to achieve lossless acceleration by automatically increasing the training overload on the fly. Although some skills are developed to speed up training stably, effort-light training is still a big challenge for TransRL, which is also an opportunity to explore new possibilities to advance the state of the art. Furthermore, with the development of the intelligent terminal, lightweight structures become an urgent requirement for TransRL deployed on physical devices. To satisfy this requirement, search-based neural network structure design (Zoph et al., 2018) and automated model compression (Cheng et al., 2020) will be promising directions to further improve the ability of TransRL.

6.3 Stochastic effectiveness

TransRL methods that directly convert RL to Transformer are limited in largely stochastic domains (Villaflor et al., 2022), since most prior works focused on the mainly deterministic environment (such as D4RL) and a variety of weakly stochastic environments (such as Atari). First, in adversarial or stochastic environments, the state–action transit function will result in different trajectories although following the same action based on the same state, which obstructs reaching the desired results. Second, we often need to perform different optimizations over exploration and exploitation. Generally, we want to find the optimal actions that maximize the total expected rewards or take the best response to the worst-case scenario. Thus, in critical domains or adversarial games, it is safe to perform at maximum over the potential actions and at minimum over the possible futures in the environment. Paster et al. (2022) drew the same conclusion that DT can fail dramatically in stochastic environments, since trajectories that bring a return may have achieved only that return due to luck. Paster et al. (2021) described a counterexample to illustrate that the RvS method (reducing RL to a prediction task that is solved via supervised learning) does not converge in a stochastic environment. Ozair et al. (2021) demonstrated similar issues in chess when deploying MuZero (Schrittwieser et al., 2020) with different MCTS frameworks (Coulom, 2007). They observed that playing with a single-agent variant of MuZero that treats the other agent as an unknown part of

the environment results in a severe drop in performance relative to the traditional one-to-one adversarial framework. To improve stochastic effectiveness, further research on TransRL mechanism design for the stochastic environment is beneficial to extend TransRL's application scenario.

7 Conclusions

We summarized the research related to Transformer in RL for DM. A large number of TransRL algorithms were classified into Transformer-based offline RL, Transformer-based online RL, Transformer-based hierarchical RL, Transformer-based multi-agent RL, and Transformer-based meta-RL. We discussed the improvements and limits of these methods. Then some representative TransRL methods were introduced in detail. Next, according to the applications of RL in DM, we summarized the applications of TransRL in DM domains, such as gaming AI, robotics, autonomous driving, and combinatorial optimization. Finally, we analyzed the challenges raised by TransRL from the perspectives of stability and structure optimization, memory and computation, and stochastic effectiveness. We hope that this survey can bring inspiration to the RL community for future directions.

Contributors

Weilin YUAN and Jiaxing CHEN drafted the paper. Shaofei CHEN, Dawei FENG, Zhenzhen HU, Peng LI, and Weiwei ZHAO helped organize the paper. Weilin YUAN and Weiwei ZHAO revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

References

Ahmed O, Träuble F, Goyal A, et al., 2021. CausalWorld: a robotic manipulation benchmark for causal structure and transfer learning. Proc 9th Int Conf on Learning Representations.

Aleissaee AA, Kumar A, Anwer RM, et al., 2023. Transformers in remote sensing: a survey. *Remote Sens*, 15(7):1860. https://doi.org/10.3390/rs15071860

Alquier P, 2020. Approximate Bayesian inference. *Entropy*, 22(11):1272. https://doi.org/10.3390/e22111272

Ambartsoumian A, Popowich F, 2018. Self-attention: a better building block for sentiment analysis neural network classifiers. Proc 9th Workshop on Computational Approaches to Subjectivity, p.130-139.

Anbuudayasankar SP, Ganesh K, Mohapatra S, 2014. Survey of methodologies for TSP and VRP. In: Anbuudayasankar SP, Ganesh K, Mohapatra S (Eds.), Models for Practical Routing Problems in Logistics: Design and Practices. Springer, Cham, p.11-42. https://doi.org/10.1007/978-3-319-05035-5_2

Anderson P, Fernando B, Johnson M, et al., 2016. SPICE: semantic propositional image caption evaluation. Proc 14th European Conf on Computer Vision, p.382-398. https://doi.org/10.1007/978-3-319-46454-1_24

Anderson P, Wu Q, Teney D, et al., 2018. Vision-and-language navigation: interpreting visually-grounded navigation instructions in real environments. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3674-3683. https://doi.org/10.1109/CVPR.2018.00387

Ba JL, Kiros JR, Hinton GE, 2016. Layer normalization. https://arxiv.org/abs/1607.06450

Badia AP, Piot B, Kapturowski S, et al., 2020. Agent57: outperforming the Atari human benchmark. Proc 37th Int Conf on Machine Learning, p.507-517.

Baevski A, Auli M, 2018. Adaptive input representations for neural language modeling. Proc 7th Int Conf on Learning Representations.

Bahdanau D, Cho K, Bengio Y, 2015. Neural machine translation by jointly learning to align and translate. Proc 3rd Int Conf on Learning Representations.

Banerjee S, Lavie A, 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. Proc ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, p.65-72.

Barthet M, Liapis A, Yannakakis GN, 2023. Open-ended evolution for Minecraft building generation. *IEEE Trans Games*, 15(4):603-612. https://doi.org/10.1109/TG.2022.3189426

Bauer J, Baumli K, Behbahani F, et al., 2023. Human-timescale adaptation in an open-ended task space. Proc 40th Int Conf on Machine Learning, p.1887-1935.

Bellemare MG, Naddaf Y, Veness J, et al., 2013. The arcade learning environment: an evaluation platform for general agents. *J Artif Intell Res*, 47:253-279. https://doi.org/10.1613/jair.3912

Bello I, Pham H, Le QV, et al., 2017. Neural combinatorial optimization with reinforcement learning. Proc 5th Int Conf on Learning Representations.

Berner C, Brockman G, Chan B, et al., 2019. Dota 2 with large scale deep reinforcement learning. https://arxiv.org/abs/1912.06680

Bernstein DS, Givan R, Immerman N, et al., 2002. The complexity of decentralized control of Markov decision processes. *Math Oper Res*, 27(4):819-840. https://doi.org/10.1287/moor.27.4.819.297

Bommasani R, Hudson DA, Adeli E, et al., 2021. On the opportunities and risks of foundation models. https://arxiv.org/abs/2108.07258

Boularias A, Duvallet F, Oh J, et al., 2015. Grounding spatial relations for outdoor robot navigation. Proc IEEE Int Conf on Robotics and Automation, p.1976-1982. https://doi.org/10.1109/ICRA.2015.7139457

Bresson X, Laurent T, 2021. The Transformer network for the traveling salesman problem. https://arxiv.org/abs/2103.03012

Brockman G, Cheung V, Pettersson L, et al., 2016. OpenAI Gym. https://arxiv.org/abs/1606.01540

Brown TB, Mann B, Ryder N, et al., 2020. Language models are few-shot learners. Proc 34th Int Conf on Neural Information Processing Systems, Article 159.

Carion N, Massa F, Synnaeve G, et al., 2020. End-to-end object detection with Transformers. Proc 16th European Conf on Computer Vision, p.213-229. https://doi.org/10.1007/978-3-030-58452-8_13

Chen HT, Wang YH, Guo TY, et al., 2021. Pre-trained image processing Transformer. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.12299-12310. https://doi.org/10.1109/cvpr46437.2021.01212

Chen LL, Lu K, Rajeswaran A, et al., 2021. Decision Transformer: reinforcement learning via sequence modeling. Proc 34th Int Conf on Neural Information Processing Systems, p.15084-15097.

Chen M, Radford A, Child R, et al., 2020. Generative pretraining from pixels. Proc 37th Int Conf on Machine Learning, p.1691-1703.

Cheng Y, Wang D, Zhou P, et al., 2020. A survey of model compression and acceleration for deep neural networks. https://arxiv.org/abs/1710.09282

Cirulli G, 2014. 2048. https://play2048.co/ [Accessed on Aug. 1, 2023].

Clever HM, Handa A, Mazhar H, et al., 2022. Assistive Tele-op: leveraging Transformers to collect robotic task demonstrations. https://arxiv.org/abs/2112.05129

Conneau A, Khandelwal K, Goyal N, et al., 2020. Unsupervised cross-lingual representation learning at scale. Proc 58th Annual Meeting of the Association for Computational Linguistics, p.8440-8451.

Correia A, Alexandre LA, 2022. Hierarchical Decision Transformer. https://arxiv.org/abs/2209.10447

Coulom R, 2007. Efficient selectivity and backup operators in Monte-Carlo tree search. Proc 5th Int Conf on Computers and Games, p.72-83. https://doi.org/10.1007/978-3-540-75538-8_7

Czarnecki WM, Gidel G, Tracey B, et al., 2020. Real world games look like spinning tops. Proc 34th Int Conf on Neural Information Processing Systems, Article 1463.

Davis JQ, Gu A, Choromanski K, et al., 2021. Catformer: designing stable Transformers via sensitivity analysis. Proc 38th Int Conf on Machine Learning, p.2489-2499.

Deudon M, Cournut P, Lacoste A, et al., 2018. Learning heuristics for the TSP by policy gradient. Proc 15th Int Conf on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, p.170-181. https://doi.org/10.1007/978-3-319-93031-2_12

Devlin J, Chang MW, Lee K, et al., 2019. BERT: pre-training of deep bidirectional Transformers for language understanding. Proc Conf on North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.4171-4186. https://doi.org/10.18653/v1/N19-1423

de Witt CS, Peng B, Kamienny PA, et al., 2020. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. https://arxiv.org/abs/2003.06709v2

Dong YH, Cordonnier JB, Loukas A, 2021. Attention is not all you need: pure attention loses rank doubly exponentially with depth. Proc 38th Int Conf on Machine Learning, p.2793-2803.

Dosovitskiy A, Beyer L, Kolesnikov A, et al., 2021. An image is worth 16 × 16 words: Transformers for image recognition at scale. Proc 9th Int Conf on Learning Representations.

Du N, Huang YP, Dai AM, et al., 2022. GLaM: efficient scaling of language models with mixture-of-experts. Proc 39th Int Conf on Machine Learning, p.5547-5569.

Duan Y, Schulman J, Chen X, et al., 2016. RL^2: fast reinforcement learning via slow reinforcement learning. https://arxiv.org/abs/1611.02779

Duvallet F, Walter MR, Howard T, et al., 2016. Inferring maps and behaviors from natural language instructions. In: Hsieh MA, Khatib O, Kumar V (Eds.), Experimental Robotics: 14th Int Symp on Experimental Robotics. Springer, Cham, p.373-388. https://doi.org/10.1007/978-3-319-23778-7_25

Ehsani K, Han W, Herrasti A, et al., 2021. ManipulaTHOR: a framework for visual object manipulation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4497-4506. https://doi.org/10.1109/CVPR46437.2021.00447

Ergen T, Neyshabur B, Mehta H, 2022. Convexifying Transformers: improving optimization and understanding of Transformer networks. https://arxiv.org/abs/2211.11052

Esser P, Rombach R, Ommer B, 2021. Taming Transformers for high-resolution image synthesis. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.12873-12883. https://doi.org/10.1109/cvpr46437.2021.01268

Esslinger K, Platt R, Amato C, 2022. Deep Transformer Q-networks for partially observable reinforcement learning. https://arxiv.org/abs/2206.01078

Fan LX, Wang GZ, Jiang YF, et al., 2022. MineDojo: building open-ended embodied agents with internet-scale knowledge. Proc 36th Int Conf on Neural Information Processing Systems, p.18343-18362.

Foerster J, Farquhar G, Afouras T, et al., 2018. Counterfactual multi-agent policy gradients. Proc 32nd AAAI Conf on Artificial Intelligence, p.2974-2982. https://doi.org/10.1609/aaai.v32i1.11794

Furuta H, Matsuo Y, Gu SS, 2022. Generalized decision Transformer for offline hindsight information matching. Proc 10th Int Conf on Learning Representations.

Gehring J, Auli M, Grangier D, et al., 2017. Convolutional sequence to sequence learning. Proc 34th Int Conf on Machine Learning, p.1243-1252.

Goh YL, Lee WS, Bresson X, et al., 2022. Combining reinforcement learning and optimal transport for the traveling salesman problem. https://arxiv.org/abs/2203.00903

Golden BL, Levy L, Vohra R, 1987. The orienteering problem. *Nav Res Log*, 34(3):307-318. https://doi.org/10.1002/1520-6750

Graves A, 2013. Generating sequences with recurrent neural networks. https://arxiv.org/abs/1308.0850

Gronauer S, Diepold K, 2022. Multi-agent deep reinforcement learning: a survey. *Artif Intell Rev*, 55(2):895-943. https://doi.org/10.1007/s10462-021-09996-w

Gu J, Stefani E, Wu Q, et al., 2022. Vision-and-language navigation: a survey of tasks, methods, and future directions. Proc 60th Annual Meeting of the Association

for Computational Linguistics, p.7606-7623. https://doi.org/10.18653/v1/2022.acl-long.524

Guhur PL, Chen SZ, Pinel RG, et al., 2022. Instruction-driven history-aware policies for robotic manipulations. Proc 6th Conf on Robot Learning, p.175-187.

Guo MS, Zhang Y, Liu T, 2019. Gaussian Transformer: a lightweight approach for natural language inference. Proc 33rd AAAI Conf on Artificial Intelligence, p.6489-6496. https://doi.org/10.1609/aaai.v33i01.33016489

Guss WH, Houghton B, Topin N, et al., 2019. MineRL: a large-scale dataset of Minecraft demonstrations. Proc 28th Int Joint Conf on Artificial Intelligence, p.2442-2448.

Haarnoja T, Zhou A, Abbeel P, et al., 2018. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. Proc 35th Int Conf on Machine Learning, p.1856-1865.

Han K, Wang YH, Chen HT, et al., 2023. A survey on vision Transformer. *IEEE Trans Patt Anal Mach Intell*, 45(1):87-110. https://doi.org/10.1109/TPAMI.2022.3152247

Han YH, Yu KL, Batra R, et al., 2021. Learning generalizable vision-tactile robotic grasping strategy for deformable objects via Transformer. https://arxiv.org/abs/2112.06374

Hansen N, Su H, Wang XL, 2021. Stabilizing deep Q-learning with ConvNets and vision Transformers under data augmentation. Proc 34th Int Conf on Neural Information Processing Systems, p.3680-3693.

Hartmanis J, 1982. Computers and intractability: a guide to the theory of NP-completeness (Michael R. Garey and David S. Johnson). *SIAM Rev*, 24(1):90-91. https://doi.org/10.1137/1024022

He KM, Zhang XY, Ren SQ, et al., 2016. Deep residual learning for image recognition. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.770-778. https://doi.org/10.1109/cvpr.2016.90

Hermann KM, Hill F, Green S, et al., 2017. Grounded language learning in a simulated 3D world. https://arxiv.org/abs/1706.06551

Hochreiter S, Schmidhuber J, 1997. Long short-term memory. *Neur Comput*, 9(8):1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hong S, Yoon D, Kim KE, 2022. Structure-aware Transformer policy for inhomogeneous multi-task reinforcement learning. Proc 10th Int Conf on Learning Representations.

Hospedales T, Antoniou A, Micaelli P, et al., 2022. Meta-learning in neural networks: a survey. *IEEE Trans Patt Anal Mach Intell*, 44(9):5149-5169. https://doi.org/10.1109/TPAMI.2021.3079209

Hu SY, Zhu FD, Chang XJ, et al., 2021. UPDeT: universal multi-agent reinforcement learning via policy decoupling with Transformers. https://arxiv.org/abs/2101.08001

Imhof T, 2022. A Review of the Decision Transformer Architecture: Framing Reinforcement Learning as a Sequence Modeling Problem. https://api.semanticscholar.org/CorpusID:248941921

Jaderberg M, Czarnecki WM, Dunning I, et al., 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859-865. https://doi.org/10.1126/science.aau6249

Jain V, Lin YX, Undersander E, et al., 2023. Transformers are adaptable task planners. Proc 6th Conf on Robot Learning, p.1011-1037.

James S, Ma ZC, Arrojo DR, et al., 2020. RLBench: the robot learning benchmark & learning environment. *IEEE Robot Autom Lett*, 5(2):3019-3026. https://doi.org/10.1109/LRA.2020.2974707

Janner M, Li QY, Levine S, 2021. Offline reinforcement learning as one big sequence modeling problem. Proc 34th Int Conf on Neural Information Processing Systems, p.1273-1286.

Jiang YF, Chang SY, Wang ZY, 2021. TransGAN: two pure Transformers can make one strong GAN, and that can scale up. Proc 34th Int Conf on Neural Information Processing Systems, p.14745-14758.

Kaplan J, McCandlish S, Henighan T, et al., 2020. Scaling laws for neural language models. https://arxiv.org/abs/2001.08361

Kapturowski S, Campos V, Jiang R, et al., 2023. Human-level Atari 200× faster. Proc 11th Int Conf on Learning Representations.

Keneshloo Y, Shi T, Ramakrishnan N, et al., 2020. Deep reinforcement learning for sequence-to-sequence models. *IEEE Trans Neur Netw Learn Syst*, 31(7):2469-2489. https://doi.org/10.1109/TNNLS.2019.2929141

Khan MJ, Ahmed SH, Sukthankar G, 2022. Transformer-based value function decomposition for cooperative multi-agent reinforcement learning in StarCraft. Proc 18th AAAI Conf on Artificial Intelligence and Interactive Digital Entertainment, p.113-119. https://doi.org/10.1609/aiide.v18i1.21954

Kim Y, 2014. Convolutional neural networks for sentence classification. Proc Conf on Empirical Methods in Natural Language Processing, p.1746-1751.

Kochenderfer MJ, Wheeler TA, Wray KH, 2022. Algorithms for Decision Making. MIT Press, Cambridge, USA.

Kool W, van Hoof H, Welling M, 2019. Attention, learn to solve routing problems! Proc 7th Int Conf on Learning Representations.

Krizhevsky A, Sutskever I, Hinton GE, 2012. ImageNet classification with deep convolutional neural networks. Proc 25th Int Conf on Neural Information Processing Systems, p.1097-1105.

Kuba JG, Wen MN, Meng LH, et al., 2021. Settling the variance of multi-agent policy gradients. Proc 34th Int Conf on Neural Information Processing Systems, p.13458-13470.

Kudo T, Richardson J, 2018. SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing. Proc Conf on Empirical Methods in Natural Language Processing: System Demonstrations, p.66-71. https://doi.org/10.18653/v1/D18-2012

Kurach K, Raichuk A, Stańczyk P, et al., 2020. Google Research Football: a novel reinforcement learning environment. Proc 34th AAAI Conf on Artificial Intelligence, p.4501-4510. https://doi.org/10.1609/aaai.v34i04.5878

Lan ZZ, Chen MD, Goodman S, et al., 2020. ALBERT: a lite BERT for self-supervised learning of language representations. Proc 8th Int Conf on Learning Representations.

Lee KH, Nachum O, Yang MJ, et al., 2022. Multi-game decision Transformers. Proc 36th Int Conf on Neural Information Processing Systems, p.27921-27936.

Levine S, Kumar A, Tucker G, et al., 2020. Offline reinforcement learning: tutorial, review, and perspectives on open problems. https://arxiv.org/abs/2005.01643

Levy A, Konidaris GD, Platt RJr, et al., 2019. Learning multi-level hierarchies with hindsight. Proc 7th Int Conf on Learning Representations.

Lewis M, Liu YH, Goyal N, et al., 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Proc 58th Annual Meeting of the Association for Computational Linguistics, p.7871-7880. https://doi.org/10.18653/v1/2020.acl-main.703

Lewis P, Stenetorp P, Riedel S, 2021. Question and answer test-train overlap in open-domain question answering datasets. Proc 16th Conf on European Chapter of the Association for Computational Linguistics, p.1000-1008. https://doi.org/10.18653/v1/2021.eacl-main.86

Li CL, Zhuang BH, Wang GR, et al., 2022. Automated progressive learning for efficient training of vision Transformers. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.12486-12496. https://doi.org/10.1109/cvpr52688.2022.01216

Li JJ, Koyamada S, Ye QW, et al., 2020. Suphx: mastering Mahjong with deep reinforcement learning. https://arxiv.org/abs/2003.13590

Li WY, Hong RX, Shen JW, et al., 2022. Learning to navigate in interactive environments with the Transformer-based memory. https://api.semanticscholar.org/CorpusID:249980271

Li X, Zhang Y, Yuan WL, et al., 2022. Incorporating external knowledge reasoning for vision-and-language navigation with assistant's help. *Appl Sci*, 12(14):7053. https://doi.org/10.3390/app12147053

Li XX, Meng M, Hong YG, et al., 2023. A survey of decision making in adversarial games. *Sci China Inform Sci*, early access. https://doi.org/10.1007/s11432-022-3777-y

Lin CY, 2004. ROUGE: a package for automatic evaluation of summaries. Proc Text Summarization Branches Out, p.74-81.

Lin QJ, Liu H, Sengupta B, 2022. Switch Trajectory Transformer with distributional value approximation for multi-task reinforcement learning. https://arxiv.org/abs/2203.07413

Lin RJ, Li Y, Feng XD, et al., 2022. Contextual Transformer for offline meta reinforcement learning. https://arxiv.org/abs/2211.08016

Lin TY, Wang YX, Liu XY, et al., 2022. A survey of Transformers. *AI Open*, 3:111-132. https://doi.org/10.1016/j.aiopen.2022.10.001

Liu BY, Balaji Y, Xue LZ, et al., 2021. Analyzing attention mechanisms through lens of sample complexity and loss landscape. Proc Int Conf on Learning Representations.

Liu HC, Huang ZY, Mo XY, et al., 2022. Augmenting reinforcement learning with Transformer-based scene representation learning for decision-making of autonomous driving. https://arxiv.org/abs/2208.12263

Liu LY, Liu XD, Gao JF, et al., 2020. Understanding the difficulty of training Transformers. Proc Conf on Empirical Methods in Natural Language Processing, p.5747-5763.

Liu T, Wang JH, Zhang X, et al., 2019. Game theoretic control of multiagent systems. *SIAM J Contr Optim*, 57(3):1691-1709. https://doi.org/10.1137/18M1177615

Liu YH, Ott M, Goyal N, et al., 2019. RoBERTa: a robustly optimized BERT pretraining approach. https://arxiv.org/abs/1907.11692

Lowe R, Wu Y, Tamar A, et al., 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. Proc 31st Int Conf on Neural Information Processing Systems, p.6382-6393.

Lu K, Grover A, Abbeel P, et al., 2022. Frozen pretrained Transformers as universal computation engines. Proc 36th AAAI Conf on Artificial Intelligence, p.7628-7637. https://doi.org/10.1609/aaai.v36i7.20729

Lu YL, Li WX, 2022. Techniques and paradigms in modern game AI systems. *Algorithms*, 15(8):282. https://doi.org/10.3390/a15080282

Ma SM, Wang HY, Huang SH, et al., 2022. TorchScale: Transformers at scale. https://arxiv.org/abs/2211.13184

Mazyavkina N, Sviridov S, Ivanov S, et al., 2021. Reinforcement learning for combinatorial optimization: a survey. *Comput Oper Res*, 134:105400. https://doi.org/10.1016/j.cor.2021.105400

Mees O, Hermann L, Rosete-Beas E, et al., 2022. CALVIN: a benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robot Autom Lett*, 7(3):7327-7334. https://doi.org/10.1109/LRA.2022.3180108

Melo LC, 2022. Transformers are meta-reinforcement learners. Proc 39th Int Conf on Machine Learning, p.15340-15359.

Meng LH, Wen MN, Yang YD, et al., 2021. Offline pre-trained multi-agent decision Transformer: one big sequence model tackles all SMAC tasks. https://arxiv.org/abs/2112.02845

Mesnard T, Weber T, Viola F, et al., 2021. Counterfactual credit assignment in model-free reinforcement learning. Proc 38th Int Conf on Machine Learning, p.7654-7664.

Miao XP, Wang YJ, Jiang YH, et al., 2022. Galvatron: efficient Transformer training over multiple GPUs using automatic parallelism. *Proc VLDB Endow*, 16(3):470-479. https://doi.org/10.14778/3570690.3570697

Mitchell E, Rafailov R, Peng XB, et al., 2021. Offline meta-reinforcement learning with advantage weighting. Proc 38th Int Conf on Machine Learning, p.7780-7791.

Mohamed N, Al-Jaroodi J, Lazarova-Molnar S, et al., 2021. Applications of integrated IoT-fog-cloud systems to smart cities: a survey. *Electronics*, 10(23):2918. https://doi.org/10.3390/electronics10232918

Moravčík M, Schmid M, Burch N, et al., 2017. DeepStack: expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508-513. https://doi.org/10.1126/science.aam6960

Motokawa Y, Sugawara T, 2022. Distributed multi-agent deep reinforcement learning for robust coordination against noise. Proc Int Joint Conf on Neural Networks, p.1-8. https://doi.org/10.1109/IJCNN55064.2022.9892253

Niu ZY, Zhong GQ, Yu H, 2021. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48-62. https://doi.org/10.1016/j.neucom.2021.03.091

Oh J, Suppé A, Duvallet F, et al., 2015. Toward mobile robots reasoning like humans. Proc 29th AAAI Conf on Artificial Intelligence, p.1371-1379. https://doi.org/10.1609/aaai.v29i1.9383

Oliehoek FA, Spaan MTJ, Vlassis N, 2008. Optimal and approximate Q-value functions for decentralized POMDPs. *J Artif Intell Res*, 32(1):289-353.

Omidshafiei S, Tuyls K, Czarnecki WM, et al., 2020. Navigating the landscape of multiplayer games. *Nat Commun*, 11(1):5603. https://doi.org/10.1038/s41467-020-19244-4

Open Ended Learning Team, Stooke A, Mahajan A, et al., 2021. Open-ended learning leads to generally capable agents. https://arxiv.org/abs/2107.12808

Ortega PA, Wang JX, Rowland M, et al., 2019. Meta-learning of sequential strategies. https://arxiv.org/abs/1905.03030

Ozair S, Li YZ, Razavi A, et al., 2021. Vector quantized models for planning. Proc 38th Int Conf on Machine Learning, p.8302-8313.

Pan C, Okorn B, Zhang H, et al., 2023. TAX-pose: task-specific cross-pose estimation for robot manipulation. Proc 6th Conf on Robot Learning, p.1783-1792.

Pan YW, Li YH, Zhang YH, et al., 2022. Silver-bullet-3D at ManiSkill 2021: learning-from-demonstrations and heuristic rule-based methods for object manipulation. Proc Int Conf on Learning Representations.

Papineni K, Roukos S, Ward T, et al., 2002. BLEU: a method for automatic evaluation of machine translation. Proc 40th Annual Meeting of the Association for Computational Linguistics, p.311-318. https://doi.org/10.3115/1073083.1073135

Parisotto E, Salakhutdinov R, 2021. Efficient Transformers in reinforcement learning using actor-learner distillation. Proc 9th Int Conf on Learning Representations.

Parisotto E, Song F, Rae J, et al., 2020. Stabilizing Transformers for reinforcement learning. Proc 37th Int Conf on Machine Learning, p.7487-7498.

Parr R, Russell S, 1997. Reinforcement learning with hierarchies of machines. Proc 10th Int Conf on Neural Information Processing Systems, p.1043-1049.

Paster K, McIlraith SA, Ba J, 2021. Planning from pixels using inverse dynamics models. Proc 9th Int Conf on Learning Representations.

Paster K, McIlraith S, Ba J, 2022. You can't count on luck: why decision Transformers and RvS fail in stochastic environments. Proc 36th Int Conf on Neural Information Processing Systems, p.38966-38979.

Pateria S, Subagdja B, Tan AH, et al., 2022. Hierarchical reinforcement learning: a comprehensive survey. *ACM Comput Surv*, 54(5):109. https://doi.org/10.1145/3453160

Phillips-Wren G, 2012. AI tools in decision making support systems: a review. *Int J Artif Intell Tools*, 21(2):1240005. https://doi.org/10.1142/S0218213012400052

Phuong M, Hutter M, 2022. Formal algorithms for Transformers. https://arxiv.org/abs/2207.09238

Pinon B, Delvenne JC, Jungers R, 2022. A model-based approach to meta-reinforcement learning: Transformers and tree search. https://arxiv.org/abs/2208.11535

Radford A, Narasimhan K, Salimans T, et al., 2018. Improving language understanding by generative pre-training. https://api.semanticscholar.org/CorpusID:49313245

Radford A, Wu J, Child R, et al., 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Radford A, Kim JW, Hallacy C, et al., 2021. Learning transferable visual models from natural language supervision. Proc 38th Int Conf on Machine Learning, p.8748-8763.

Raffel C, Shazeer N, Roberts A, et al., 2020. Exploring the limits of transfer learning with a unified text-to-text Transformer. *J Mach Learn Res*, 21(1):140.

Rashid T, Samvelyan M, de Witt CS, et al., 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *J Mach Learn Res*, 21(1):178.

Reed S, Zolna K, Parisotto E, et al., 2022. A generalist agent. *Trans Mach Learn Res*, 2022:2835-8856.

Reid M, Yamada Y, Gu SS, 2022. Can Wikipedia help offline reinforcement learning? https://arxiv.org/abs/2201.12122

Samvelyan M, Rashid T, de Witt CS, et al., 2019. The StarCraft multi-agent challenge. Proc 18th Int Conf on Autonomous Agents and Multiagent Systems, p.2186-2188.

Sanchez FR, Redmond S, McGuinness K, et al., 2022. Towards advanced robotic manipulation. Proc 6th IEEE Int Conf on Robotic Computing, p.302-305. https://doi.org/10.1109/IRC55401.2022.00058

Schrittwieser J, Antonoglou I, Hubert T, et al., 2020. Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature*, 588(7839):604-609, https://doi.org/10.1038/s41586-020-03051-4

Schulman J, Wolski F, Dhariwal P, et al., 2017. Proximal policy optimization algorithms. https://arxiv.org/abs/1707.06347

Shamshad F, Khan S, Zamir SW, et al., 2023. Transformers in medical imaging: a survey. *Med Image Anal*, 88:102802. https://doi.org/10.1016/j.media.2023.102802

Shang JH, Kahatapitiya K, Li X, et al., 2022. StARformer: Transformer with state-action-reward representations for visual reinforcement learning. Proc 17th European Conf on Computer Vision, p.462-479, https://doi.org/10.1007/978-3-031-19842-7_27

Shaw P, Uszkoreit J, Vaswani A, 2018. Self-attention with relative position representations. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.464-468. https://doi.org/10.18653/v1/N18-2074

Shoham Y, Leyton-Brown K, 2008. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, New York, USA.

Shridhar M, Manuelli L, Fox D, 2023. Perceiver-actor: a multi-task Transformer for robotic manipulation. Proc 6th Conf on Robot Learning, p.785-799.

Siebenborn M, Belousov B, Huang JN, et al., 2022. How crucial is Transformer in Decision Transformer? https://arxiv.org/abs/2211.14655

Silver D, Hubert T, Schrittwieser J, et al., 2017a. Mastering Chess and Shogi by self-play with a general reinforcement learning algorithm. https://arxiv.org/abs/1712.01815

Silver D, Schrittwieser J, Simonyan K, et al., 2017b. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354-359. https://doi.org/10.1038/nature24270

Singh B, Kumar R, Singh VP, 2022. Reinforcement learning in robotic applications: a comprehensive survey. *Artif Intell Rev*, 55(2):945-990. https://doi.org/10.1007/s10462-021-09997-9

Srinidhi CL, Ciga O, Martel AL, 2021. Deep neural network models for computational histopathology: a survey. *Med Image Anal*, 67:101813. https://doi.org/10.1016/j.media.2020.101813

Srivastava RK, Shyam P, Mutz F, et al., 2019. Training agents using upside-down reinforcement learning. https://arxiv.org/abs/1912.02877

Stadie BC, Yang G, Houthooft R, et al., 2018. Some considerations on learning to explore via meta-reinforcement learning. https://arxiv.org/abs/1803.01118

Sutton RS, Barto AG, 2018. Reinforcement Learning: an Introduction (2nd Ed.). MIT Press, Cambridge, USA.

Takase S, Kiyono S, Kobayashi S, et al., 2022. On layer normalizations and residual connections in Transformers. https://arxiv.org/abs/2206.00330v1

Tay Y, Dehghani M, Bahri D, et al., 2023. Efficient Transformers: a survey. *ACM Comput Surv*, 55(6):109. https://doi.org/10.1145/3530811

Toth P, Vigo D, 2014. Vehicle Routing: Problems, Methods, and Applications (2nd Ed.). Society for Industrial and Applied Mathematics. Mathematical Optimization Society, Philadelphia, USA.

Tunyasuvunakool S, Muldal A, Doron Y, et al., 2020. dm_control: software and tasks for continuous control. *Softw Impacts*, 6:100022. https://doi.org/10.1016/j.simpa.2020.100022

Upadhyay U, Shah N, Ravikanti S, et al., 2019. Transformer based reinforcement learning for games. https://arxiv.org/abs/1912.03918

Vashishth S, Upadhyay S, Tomar GS, et al., 2019. Attention interpretability across NLP tasks. https://arxiv.org/abs/1909.11218

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Vedantam R, Lawrence Zitnick C, Parikh D, 2015. CIDEr: consensus-based image description evaluation. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4566-4575. https://doi.org/10.1109/CVPR.2015.7299087

Vesselinova N, Steinert R, Perez-Ramirez DF, et al., 2020. Learning combinatorial optimization on graphs: a survey with applications to networking. *IEEE Access*, 8:120388-120416. https://doi.org/10.1109/ACCESS.2020.3004964

Villaflor AR, Huang Z, Pande S, et al., 2022. Addressing optimism bias in sequence modeling for reinforcement learning. Proc 39th Int Conf on Machine Learning, p.22270-22283.

Vinyals O, Babuschkin I, Czarnecki WM, et al., 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350-354. https://doi.org/10.1038/s41586-019-1724-z

Voita E, Talbot D, Moiseev F, et al., 2019. Analyzing multi-head self-attention: specialized heads do the heavy lifting, the rest can be pruned. Proc 57th Annual Meeting of the Association for Computational Linguistics, p.5797-5808. https://doi.org/10.18653/v1/P19-1580

Wang HB, Xie XD, Zhou LK, 2023. Transform networks for cooperative multi-agent deep reinforcement learning. *Appl Intell*, 53(8):9261-9269. https://doi.org/10.1007/s10489-022-03924-3

Wang HY, Ma SM, Dong L, et al., 2022. DeepNet: scaling Transformers to 1,000 layers. https://arxiv.org/abs/2203.00555

Wang J, King M, Porcel N, et al., 2021. Alchemy: a benchmark and analysis toolkit for meta-reinforcement learning agents. Proc 1st Neural Information Processing Systems Track on Datasets and Benchmarks.

Wang KR, Zhao HY, Luo XF, et al., 2022. Bootstrapped Transformer for offline reinforcement learning. Proc 36th Int Conf on Neural Information Processing Systems, p.34748-34761.

Wang MR, Feng MX, Zhou WG, et al., 2022. Stabilizing voltage in power distribution networks via multi-agent reinforcement learning with Transformer. Proc 28th ACM SIGKDD Conf on Knowledge Discovery and Data Mining, p.1899-1909. https://doi.org/10.1145/3534678.3539480

Wang Q, Tang CL, 2021. Deep reinforcement learning for transportation network combinatorial optimization: a survey. *Knowl-Based Syst*, 233:107526. https://doi.org/10.1016/j.knosys.2021.107526

Wen MN, Kuba JG, Lin RJ, et al., 2022. Multi-agent reinforcement learning is a sequence modeling problem. Proc 36th Int Conf on Neural Information Processing Systems, p.16509-16521.

Wolsey LA, 2020. Integer Programming (2nd Ed.). Wiley, New Jersey, USA.

Wu TH, Jiang MZ, Han YH, et al., 2021. A traffic-aware federated imitation learning framework for motion control at unsignalized intersections with Internet of Vehicles. *Electronics*, 10(24):3050. https://doi.org/10.3390/electronics10243050

Wu YX, Song W, Cao ZG, et al., 2022. Learning improvement heuristics for solving routing problems. *IEEE Trans Neur Netw Learn Syst*, 33(9):5057-5069. https://doi.org/10.1109/TNNLS.2021.3068828

Xiang FB, Qin YZ, Mo KC, et al., 2020. SAPIEN: a SimulAted Part-based Interactive ENvironment. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.11097-11107. https://doi.org/10.1109/CVPR42600.2020.01111

Xiang XC, Foo S, 2021. Recent advances in deep reinforcement learning applications for solving partially observable Markov decision processes (POMDP) problems: part 1—fundamentals and applications in games, robotics and natural language processing. *Mach Learn Knowl Extr*, 3(3):554-581. https://doi.org/10.3390/make3030029

Xie ZH, Lin ZC, Li JY, et al., 2022. Pretraining in deep reinforcement learning: a survey. https://arxiv.org/abs/2211.03959

Xiong RB, Yang YC, He D, et al., 2020. On layer normalization in the Transformer architecture. Proc 37th Int Conf on Machine Learning, p.10524-10533.

Xu MD, Shen YK, Zhang S, et al., 2022. Prompting Decision Transformer for few-shot policy generalization. Proc 39th Int Conf on Machine Learning, p.24631-24645.

Yamagata T, Khalil A, Santos-Rodríguez R, 2023. Q-learning decision Transformer: leveraging dynamic programming for conditional sequence modelling in offline RL. Proc 40th Int Conf on Machine Learning, Article 1625.

Yang RH, Zhang MH, Hansen N, et al., 2022. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal Transformers. Proc 10th Int Conf on Learning Representations.

Yang YD, Wang J, 2020. An overview of multi-agent reinforcement learning from game theoretical perspective. https://arxiv.org/abs/2011.00583

Yang YD, Wen Y, Wang JH, et al., 2020. Multi-agent determinantal Q-learning. Proc 37th Int Conf on Machine Learning, Article 997.

Yang YD, Chen GY, Wang WX, et al., 2022. Transformer-based working memory for multiagent reinforcement learning with action parsing. Proc 36th Int Conf on Neural Information Processing Systems, p.34874-34886.

Yang YM, Xing DP, Xu B, 2022. Efficient spatiotemporal Transformer for robotic reinforcement learning. *IEEE Robot Autom Lett*, 7(3):7982-7989. https://doi.org/10.1109/LRA.2022.3186494

Yang ZL, Dai ZH, Yang YM, et al., 2019. XLNet: generalized autoregressive pretraining for language understanding. Proc 33rd Int Conf on Neural Information Processing Systems, Article 517.

Yao ZW, Wu XX, Li CL, et al., 2022. Random-LTD: random and layerwise token dropping brings efficient training for large-scale Transformers. https://arxiv.org/abs/2211.11586

Yu C, Velu A, Vinitsky E, et al., 2022. The surprising effectiveness of PPO in cooperative multi-agent games. Proc 36th Int Conf on Neural Information Processing Systems, p.24611-24624.

Yu TH, Kumar S, Gupta A, et al., 2020a. Gradient surgery for multi-task learning. Proc 34th Int Conf on Neural Information Processing Systems, Article 489.

Yu TH, Quillen D, He ZP, et al., 2020b. Meta-World: a benchmark and evaluation for multi-task and meta reinforcement learning. Proc Conf on Robot Learning, p.1094-1100.

Yuan WL, Hu ZZ, Luo JR, et al., 2021. Imperfect information game in multiplayer no-limit Texas hold'em based on mean approximation and deep CFVnet. Proc China Automation Congress, p.2459-2466. https://doi.org/10.1109/CAC53003.2021.9727939

Yuan Z, Wu TH, Wang QW, et al., 2022. T3OMVP: a Transformer-based time and team reinforcement learning scheme for observation-constrained multi-vehicle pursuit in urban area. *Electronics*, 11(9):1339. https://doi.org/10.3390/electronics11091339

Yurtsever E, Lambert J, Carballo A, et al., 2020. A survey of autonomous driving: common practices and emerging technologies. *IEEE Access*, 8:58443-58469. https://doi.org/10.1109/ACCESS.2020.2983149

Zaremba W, Sutskever I, Vinyals O, 2014. Recurrent neural network regularization. https://arxiv.org/abs/1409.2329

Zha DC, Xie JR, Ma WY, et al., 2021. DouZero: mastering DouDizhu with self-play deep reinforcement learning. Proc 38th Int Conf on Machine Learning, p.12333-12344.

Zhang JZ, Kim J, O'Donoghue B, et al., 2021. Sample efficient reinforcement learning with REINFORCE. Proc 35th AAAI Conf on Artificial Intelligence, p.10887-10895. https://doi.org/10.1609/aaai.v35i12.17300

Zhao EM, Yan RY, Li JQ, et al., 2022. AlphaHoldem: high-performance artificial intelligence for heads-up no-limit poker via end-to-end reinforcement learning. Proc 36th AAAI Conf on Artificial Intelligence, p.4689-4697. https://doi.org/10.1609/aaai.v36i4.20394

Zhao WS, Queralta JP, Westerlund T, 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. Proc IEEE Symp Series on Computational Intelligence, p.737-744. https://doi.org/10.1109/SSCI47803.2020.9308468

Zhao YP, Zhao J, Hu XH, et al., 2022. DouZero+: improving DouDizhu AI by opponent modeling and coach-guided learning. Proc IEEE Conf on Games, p.127-134. https://doi.org/10.1109/CoG51982.2022.9893710

Zheng QQ, Zhang A, Grover A, 2022. Online decision Transformer. Proc 39th Int Conf on Machine Learning, p.27042-27059.

Zhou J, Ke P, Qiu XP, et al., 2023. ChatGPT: potential, prospects, and limitations. *Front Inform Technol Electron Eng*, early access. https://doi.org/10.1631/FITEE.2300089

Zoph B, Vasudevan V, Shlens J, et al., 2018. Learning transferable architectures for scalable image recognition. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.8697-8710. https://doi.org/10.1109/cvpr.2018.00907