2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Multi-objective Cross-task Learning via Goal-conditioned GPT-based Decision Transformers for Surgical Robot Task Automation

Jiawei Fu, Yonghao Long, Kai Chen, Wang Wei, Qi Dou

Abstract— **Surgical robot task automation has been a promising research topic for improving surgical efficiency and quality. Learning-based methods have been recognized as an interesting paradigm and been increasingly investigated. However, existing approaches encounter difficulties in long-horizon goal-conditioned tasks due to the intricate compositional structure, which requires decision-making for a sequence of sub-steps and understanding of inherent dynamics of goal-reaching tasks. In this paper, we propose a new learning-based framework by leveraging the strong reasoning capability of the GPT-based architecture to automate surgical robotic tasks. The key to our approach is developing a goal-conditioned decision transformer to achieve sequential representations with goal-aware future indicators in order to enhance temporal reasoning. Moreover, considering to exploit a general understanding of dynamics inherent in manipulations, thus making the model's reasoning ability to be task-agnostic, we also design a cross-task pretraining paradigm that uses multiple training objectives associated with data from diverse tasks. We have conducted extensive experiments on 10 tasks using the surgical robot learning simulator SurRoL [1]. The results show that our new approach achieves promising performance and task versatility compared to existing methods. The learned trajectories can be deployed on the da Vinci Research Kit (dVRK) for validating its practicality in real surgical robot settings. Our project website is at: `https://med-air.github.io/SurRoL`.**

I. Introduction

Surgical robot task automation has been increasingly studied for its potential to improve surgical efficiency and augment robot intelligence. Recent advancements have witnessed research on learning-based methods [1]–[5] to promote automation of surgical robots. Still, current performances of the latest methods are impeded in long-horizon goal-conditioned tasks, where a sequence of actions and sub-steps are required until reaching an ultimate goal. Previous algorithms with reinforcement learning [6] and Markov decision process only predict actions from the current state while overlooking information from historical sequential states and actions. This lacks temporal reasoning capability over actions and affects learning of the inherent sequential dynamics which is useful to the final success of a complex task. Despite some works [7], [8] combining task-specific strategies to conduct sub-task decomposition, those customized designs may sacrifice the generalization ability to other surgical robot tasks, thus imposing constraints on their overall efficacy and applicability in the domain.

In this regard, to address long-horizon goal-conditioned tasks in surgical robot learning, the model should understand the contextual dependency of the sequence across diverse tasks, and hold holistic features of the goal-reaching pattern in tasks to accurately predict appropriate decisions. Recently, large language models (LLMs) are very popular through the success of GPT [9], and have been transferred to solving decision-making problems in robotics [10]. Researchers apply the transformer backbone of LLMs for decision-making from historical state and sequential action input. Specifically, decision transformer families [10]–[14] have achieved impressive results in gym- and atari-based environments. They use autoregressive models to yield the trajectories through the reasoning of transformer architectures, and forecast the action based on the history sequence and returns-to-go, which means an accumulated reward from a current state to the ultimate goal state without decay.

Nevertheless, existing decision transformer methods [10]–[14] heavily rely on immediate reward feedback to update returns-to-go, which cannot be satisfied in our considered goal-conditioned paradigm, because the agent would only get rewards when it reaches final goals in surgical tasks. Furthermore, the introduction of task-specific rewards and the loss of cross-task pretraining create varying internal dynamics across tasks, resulting in technical challenges in developing a unified framework for reasoning and decision-making within the goal-reaching paradigm in surgical tasks.

To leverage the advanced GPT-based decision-making frameworks for improving surgical robot task automation, we propose the goal-conditioned decision transformer that embedds goal and time-to-goal as future indicators. Besides, we formulate multiple training objectives: *action prediction*, *dynamics prediction*, *time-to-goal prediction*, and *sequence reconstruction* in our cross-task pretraining process, which fosters a comprehensive representation of the temporal dynamics inherent in goal-conditioned tasks and encourages the model to incorporate diverse temporal reasoning factors. Based on such pretraining, the model is updated on the targeted specific task as a downstream use case. Importantly, the update process does not require any modification on the model architectures, thanks to the shared goal-reaching pattern which is uniform regardless of the individual reward of each task. We rely on the open-source simulator SurRoL [1], [15] to collect data and conduct model training. Experimental results show that our proposed method exceeds existing decision-making algorithms and task-specific methods in

J. Fu, Y. Long, K. Chen, W. Wei, and Q. Dou are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. This work was supported in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 24209223), in part by the Hong Kong Innovation and Technology Fund (Project No. ITS/223/22), in part by a grant from National Natural Science Foundation of China (Project No. 62322318), and in part by InnoHK Multi-scale Medical Robotics Centre.

Corresponding author: Qi Dou (qidou@cuhk.edu.hk)

average performance and versatility among diverse tasks. The contributions are:

- We propose a new learning-based framework with goal-conditioned decision transformer for surgical robot scenarios, which is designed for goal-reaching surgical tasks with a strong reasoning ability.
- We design a multi-objective cross-task pretraining strategy, which learns the sequential context dependency inherent in data from diverse tasks to reason the internal dynamics of goal-conditioned paradigms.
- Experimental results demonstrate superior performance of our proposed approach compared with state-of-the-art learning-based methods. Moreover, we deploy the trajectory of our method into real-world dVRK platform to show its practicality.

II. Related Work

A. Decision-making with transformer architectures

Large-scale transformers have propelled LLMs to achieve breakthroughs in natural language processing tasks [16], [17] with the context learning capacity. Inspired by its long-term trajectory modeling, decision transformer [10] first utilizes transformer architecture for sequential decision-making problems, where sequences constructed from history states, actions and returns-to-go are embedded in transformers to replace the previously used text information. It applies autoregressive models for the sequence and can incorporate the instantaneous task-dependent reward function to allow the model be aware of future states. Yamagata *et al.* [11] learn the Q value additionally to help find the optimal action trajectory based on the decision transformer. Online decision transformer [12] modifies to stochastic policy and applies extra interactions with environments, which improves its performance. Hu *et al.* [14] augment the decision transformer architecture to accommodate tasks characterized by missing frames through the implementation of random masking techniques, alongside the exclusion of temporal span encoding.

This approach is designed to investigate the transformer's capability to decipher the sequence of events and the intricacies involved in the dynamics of the goal-conditioned paradigm. By employing a multi-task pretraining scheme across different tasks, we aim to cultivate a better understanding of the underlying dynamics representative to goal-conditioned tasks, and further improve overall performance.

B. Surgical task automation

In recent years, surgical robot automation has emerged as a popular area of research. Existing works have designed rule-based methods to improve the performance on multiple tasks, such as endoscope manipulation [18]–[20], surgical suturing [21], [22], tissue cutting [23], [24], etc. However, the rule-based methods require sophisticated design for every possible case, which consumes labor and lacks generalization probability to new tasks. To overcome these weaknesses, learning-based algorithms, i.e., reinforcement learning and imitation learning are increasingly studied in surgical robotics [2]–[5], [25]. Huang *et al.* [2] embed expert demonstrations in reinforcement learning to tackle the large exploration burden in learning for completing surgical tasks. To improve the performance of long-horizon tasks, sophisticated task-dependant sub-goals are designed [7] for training a chaining policy to connect the sequence of actions. In addition, Seita *et al.* [3] apply deep imitation learning to achieve sequential fabric smoothing in dVRK platform from raw sensor inputs.

Previous learning-based methods [7], [8] address long sequence tasks with complex task decomposition, which only adapts to a few tasks that are carefully designed and modified. This pattern results in poor generalization performance and needs a significant amount of manually designed work. To alleviate this problem, we utilize the reasoning and good generalization [13] ability of transformers as demonstrated in LLMs for surgical task automation. Our model can be versatile and general without extra task-specific designs.

III. Method

We first provide our target problem in Sec. III-A. We then present the details of our proposed framework in Sec. III-B, which contains trajectory representation, architecture, and the training and evaluation schemes. Next, we show the multi-objective pretraining and downstream task learning in Sec. III-C. Finally, the Sec. III-D describes the offline hindsight data augmentation strategy and our implementation details.

A. Problem formulation

Goal-conditioned surgical automation tasks model those tasks specifying a goal $\mathbf{g}$ for the agent to achieve. The $\mathbf{g}$ can be described by the final states of the surgical robot arms or the manipulated objects, such as desired poses or orientations. Combining the time-ordered observation sequence $\{\mathbf{o}_{t'}\}_{t'=1}^{t}$ and the history executed action sequence $\{\mathbf{a}_{t'}\}_{t'=1}^{t-1}$, our target is to leverage transformer to represent and optimize a policy $\pi(\mathbf{a}_t|\{\mathbf{o}_{t'}\}_{t'=1}^{t},\{\mathbf{a}_{t'}\}_{t'=1}^{t-1},\mathbf{g})$ that produces an action $\mathbf{a}_t$ for current execution and maximize the probability that the agent will achieve $\mathbf{g}$. We note that the goal-reaching pattern is different from the conventional reward-maximizing formulation, where environments set task-specific reward r. The previous transformer-based decision-making methods are designed to construct a policy $\pi(\mathbf{a}_t|\{\mathbf{o}_{t'}\}_{t'=1}^{t},\{\mathbf{a}_{t'}\}_{t'=1}^{t-1})$ with the help of returns-to-go $\hat{R}_t := \sum_{t'=t}^{T} r_{t'}$, where T is total steps to the ultimate goal and t is current timestep. It guides the agent to achieve more accumulative rewards.

B. Policy learning via goal-conditioned decision transformer

Trajectory representation is significant for our method. The selected representation should involve the useful temporal feature and can guide the model to reason for the future. In this regard, we additionally embed the goal $\mathbf{g}$ in the representation. Besides, different from previous methods [10]–[14] using returns-to-go, which depends on the instantaneous feedback rewards from environments, we select time-to-goal $\hat{T}$, which means the number of timesteps from the current state to the final goal in trajectory. The reason for this design comes from that goal-conditioned tasks will not give any

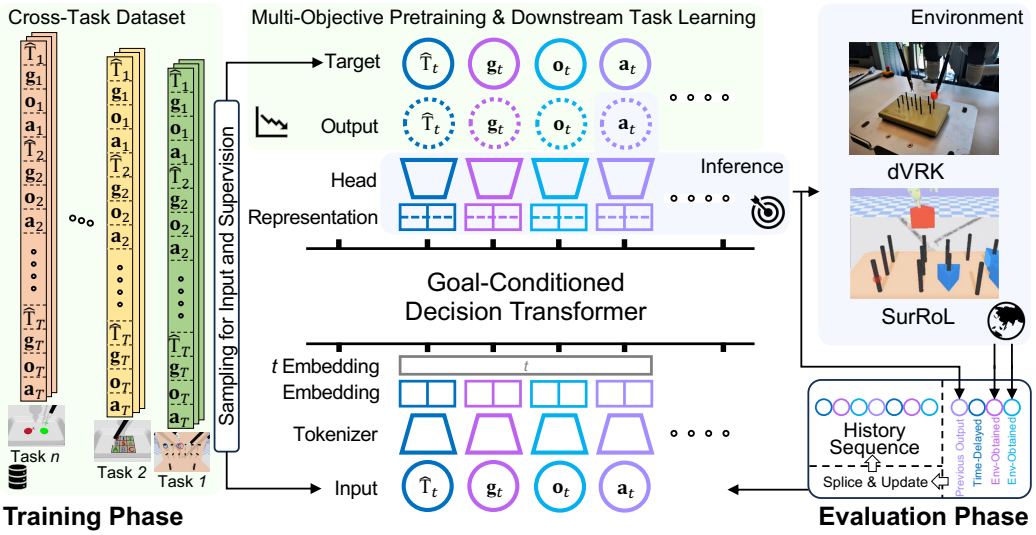

Fig. 1. Illustration of the architecture of the proposed model. For each timestep t, the sequence consists of four items: $\hat{T}_t$ (time-to-goal), $\mathbf{g}_t$ (goal), $\mathbf{o}_t$ (observation), $\mathbf{a}_t$ (action), which are embedded with the embedding of timestep and processed by the GPT architecture transformer backbone. In summary, the GPT backbone processes the input to predict results via specific heads. During pretraining and learning, sequences act as both input and target, guided by training objectives. In evaluation, we update a cached history sequence with model predictions and environmental data to forecast action $\mathbf{a}_t$.

rewards until the final targets, the accumulated future reward of returns-to-go will only be zero until arriving at the goal, which lacks the information to guide the model to be future-aware. To satisfy the aforementioned conditions and improve the sequence reasoning capacity of the model, our trajectory representation consists of four items in temporal order:

$$\xi := \{(\hat{T}_1, \mathbf{g}_1, \mathbf{o}_1, \mathbf{a}_1), ..., (\hat{T}_t, \mathbf{g}_t, \mathbf{o}_t, \mathbf{a}_t), ..., (\hat{T}_T, \mathbf{g}_T, \mathbf{o}_T, \mathbf{a}_T)\},$$

where $\hat{T}_t$, $\mathbf{g}_t$, $\mathbf{o}_t$, $\mathbf{a}_t$ denote the time-to-goal, goal, observation and action at timestep t, respectively. As depicted in Fig. 1, the items in the trajectory undergo tokenization using distinct linear layers. These tokenized representations are subsequently subjected to layer normalization. Next, the timesteps are embedded alongside their corresponding normalized representations from each tokenized element in ξ. This combined input is processed by the transformer backbone, employing the GPT architecture to generate the learned hidden representation for the input sequence. Ultimately, the corresponding heads would organize the output from the learned representation.

Our training process is divided into two stages, i.e., multi-objective cross-task pretraining and downstream task skill learning. Both stages aim to supervise the output decoded components from the corresponding prediction heads using the input sequence and confer sequence reasoning capacity to the model with available data. The details of the pretraining and skill learning for tasks are described in Section III-C. In the evaluation stage, we store the executed history sequence with the same formulation as the training phase and input the history trajectory without the current needed action to the model to get the predicted action from the action head. In the subsequent iteration, we incorporate the newly acquired items into the history sequence and input the updated trajectory to generate forecast action once again. To limit the input sequence length, we abandon the items in the beginning timestep if the size of the input trajectory exceeds the maximum limitation. Moreover, we estimate time-to-goal using a time-delayed rule from the expected overall timestep for tasks, since the loss of ground truth goal-reached timestep in the evaluation phase.

C. Cross-task pretraining and downstream task learning

1) Cross-task multi-objective pretraining: In the pretraining stage, we first augment all individual data from several tasks using the data augmentation method. The augmented cross-task dataset is deployed to train the GPT backbone of our model with task-specific tokenizers and prediction heads. This operation aims to maximize the use of existing data to improve the temporal reasoning and generalizability of our model. Furthermore, based on the deployed GPT backbone, we have designed four training objectives with corresponding masks to offer multiple forms of supervision and enhance the overall contextual reasoning capability of our model. These objectives encompass action prediction, forward dynamics prediction, time-to-goal prediction, and sequence reconstruction. The involved objectives and their masks are depicted in Fig. 2. The first three terms focus on both short-term and long-term relationship understanding for our model. The last term is devised to let our model study global and priori features from hindsight views. All the involved four objectives follow the goal-conditioned paradigm without tasks-specific rewards, which ensures the generalization and transferability among all the training tasks. By optimizing these objectives jointly, the model is compelled to consider a diverse set of temporal factors over sequences and understand the goal-conditioned paradigm behind all tasks to produce precise decisions.

Action prediction: The result of action prediction directly determines the overall decision-making performance of our model. We input the recorded sequence into our model, and the action prediction head outputs the forecast action from the learned representation with the reasoning ability of GPT architecture. We compute the MSE loss between the

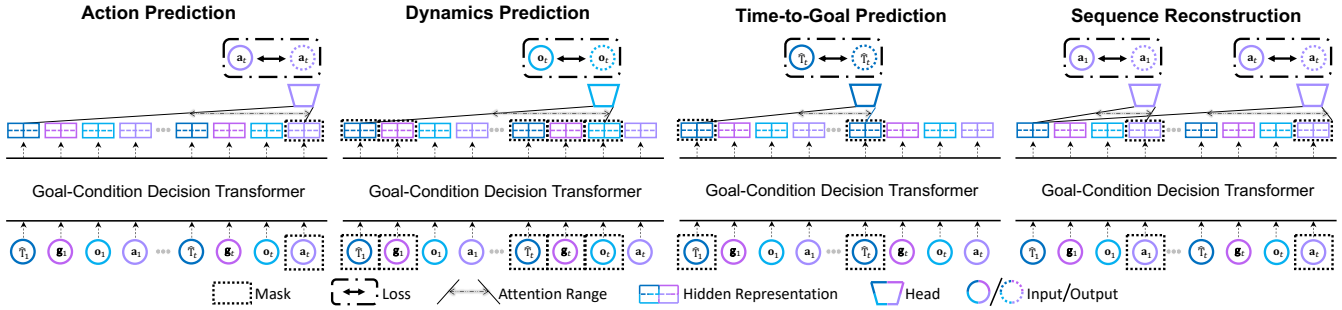

Fig. 2. Illustration of multiple training objectives that boost sequence contextual reasoning and understanding of task-agnostic paradigms of the model.

predicted and original actions in sequence to supervise the learning process.

Forward dynamics prediction: Forward dynamics illustrates the internal physics property of the tasks, which plays a crucial role in boosting the final performance of our method. During this process, we manually mask all $\hat{T}$ and $\mathbf{g}$ in the sequence. Given timestep t, our model will predict $\mathbf{o}_t$ given $\{(\mathbf{o}_{t'}, \mathbf{a}_{t'})\}_{t'=1}^{t-1}$ based on the long sequence modeling ability of GPT. This mechanism helps the model understand and analyze the influence of both short-term and long-term information on the dynamics of tasks.

Time-to-goal prediction: To forecast the time-to-goal $\hat{T}_t$ at timestep t, we mask $\hat{T}$ in the sequence to avoid injection of priori information into the model. We prepare the input sequence as $\{(\mathbf{g}_{t'}, \mathbf{o}_{t'}, \mathbf{a}_{t'})\}_{t'=0}^{t-1}\}$. Then we get $\hat{T}_t$ from the time-to-goal prediction head and calculate the loss to let the model learn to reason how long it still needs to get close to the ultimate goal and complete the task.

Sequence reconstruction: The sequence reconstruction is inspired by BERT [26]. Given the entire input sequence $\{(\hat{T}_t, \mathbf{g}_t, \mathbf{o}_t, \mathbf{a}_t)\}_{t=0}^{T}$, we randomly mask items and let the model rebuild them with the corresponding prediction head from the learned hidden state by the GPT backbone. This objective helps the model understand the total-sequence-level and temporal reasoning between each item and capture long-term interrelationships among the sequence items.

2) Downstream task learning: In the downstream task learning stage, we incorporate augmented data tailored to the specific task at hand and ignore the data from other tasks. The reason is that in the task learning stage, data from other tasks is invalid for achieving better performance in the decision-making problem for the specific downstream task. In the learning process of this period, we exclusively focus on the *action prediction* objective, while disregarding other objectives. This deliberate choice stems from our intention to leverage the inherent temporal reasoning ability and comprehension of goal-conditioned dynamic patterns of the pretrained transformer backbone to optimize action performance to the fullest extent, but not boost the reasoning ability of the GPT backbone.

D. Data augmentation and implementation details

1) Data augmentation of limited demonstrations: We implement a data augmentation strategy through the application of a hindsight relabeling technique [27]. Specifically, given the original dataset denoted as $\mathscr{D}_{\text{original}} := \{\xi^i\}_{i=1}^{N}$, where N is the trajectory number in the dataset. We systematically navigate through $\mathscr{D}_{\text{original}}$, performing truncation on each instance ξ^i at every timestep t. This process involves substituting the intended goal with the achieved goal at that timestep, resulting in a modified instance ξ_{relabel}. The collection of all such modified instances ξ_{relabel}, derived from traversing all ξ across all conceivable truncated timesteps t, is aggregated to form the relabeled dataset $\mathscr{D}_{\text{relabel}}$. Consequently, the comprehensive training dataset is constituted by the union of the original and relabeled datasets, expressed as $\mathscr{D}_{\text{training}} := \mathscr{D}_{\text{original}} \cup \mathscr{D}_{\text{relabel}}$. This augmentation process not only facilitates optimal utilization of the limited dataset but also embodies the principles of hindsight to densify the goal space, thereby significantly enhancing the efficacy of our proposed methodology.

2) Implementation details: Our model is built based on the GPT backbone [9] with 8 layers and 4 heads. All tokenizers and prediction heads are linear single layers. The maximum encoding sequence length is 100. For both the cross-task pretraining and downstream task skill learning, we supervise all training objectives using the MSE loss. We utilize the AdamW optimizer [28] with the initial learning rate 1e-4 and weight decay 1e-4. For real-world deployment, we execute the AI-predicted trajectory from the simulator to the dVRK platform. The code, trained models and implemented details in this work are released and integrated into the policy learning engine of the SurRoL simulator in its updated repo: `https://github.com/med-air/SurRoL`

IV. Experiments

In the experiments, we validate the performance of our proposed method in comparison with state-of-the-art methods using SurRoL simulator [1]. We conduct ablation experiments on key variables of our model to quantify their effects. We finally show real-world dVRK deployment results.

A. Experiment setup

Tasks: We consider skill-training surgical robotic tasks, which give goal states for the robots to reach and implement designed functions. To benchmark the performance of our approach, we adopt 10 state-based tasks as included in SurRoL [37]: NeedleReach, GauzeRetrieve, NeedlePick, PegTransfer, NeedleRegrasp, BiPegTransfer, BiPegBoard, MatchBoard-Panel, PickAndPlace, and MatchBoard. The corresponding dimensions of observation space $\mathbb{O}$, goal space $\mathbb{G}$, and action space $\mathbb{A}$ are described in the first row of Table I. We refer readers to our project website [37] for more detailed

TABLE I

MAIN COMPARISON RESULTS WITH RECENT STATE-OF-THE-ART METHODS USING MEAN AND STANDARD DEVIATION OF SUCCESS RATE.

Task dim(𝕆|𝔾|𝔸)	NeedleReach $\mathbb{R}^7\|\mathbb{R}^3\|\mathbb{R}^5$	GauzeRetrieve $\mathbb{R}^{19}\|\mathbb{R}^3\|\mathbb{R}^5$	NeedlePick $\mathbb{R}^{19}\|\mathbb{R}^3\|\mathbb{R}^5$	PegTransfer $\mathbb{R}^{19}\|\mathbb{R}^3\|\mathbb{R}^5$	NeedleRegrasp $\mathbb{R}^{35}\|\mathbb{R}^3\|\mathbb{R}^{10}$	BiPegTransfer $\mathbb{R}^{35}\|\mathbb{R}^3\|\mathbb{R}^{10}$	BiPegBoard $\mathbb{R}^{35}\|\mathbb{R}^3\|\mathbb{R}^{10}$	MatchBoardPanel $\mathbb{R}^{31}\|\mathbb{R}^6\|\mathbb{R}^5$	PickAndPlace $\mathbb{R}^{31}\|\mathbb{R}^6\|\mathbb{R}^5$	MatchBoard $\mathbb{R}^{20}\|\mathbb{R}^3\|\mathbb{R}^5$
PLAS [29]	**1.00** (±.00)	0.00 (±.00)	0.00 (±.00)	0.16 (±.08)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)
IQL [30]	**1.00** (±.00)	0.40 (±.00)	0.05 (±.04)	0.82 (±.20)	0.04 (±.01)	0.09 (±.05)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)
BC [31]	**1.00** (±.00)	0.07 (±.05)	0.21 (±.06)	0.56 (±.11)	0.09 (±.03)	0.09 (±.05)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)
VINN [32]	0.89 (±.06)	0.01 (±.02)	0.02 (±.02)	0.05 (±.04)	0.01 (±.02)	0.00 (±.00)	0.02 (±.05)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)
DT [10]	0.95 (±.08)	0.68 (±.20)	0.89 (±.10)	0.77 (±.15)	0.66 (±.18)	0.15 (±.12)	0.67 (±.09)	0.29 (±.19)	0.18 (±.04)	0.21 (±.09)
DDPGBC [33]	**1.00** (±.00)	0.63 (±.11)	0.91 (±.05)	0.48 (±.22)	0.05 (±.08)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.04 (±.01)	0.05 (±.03)
AMP [34]	0.99 (±.02)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)
CoL [35]	**1.00** (±.00)	0.71 (±.16)	0.96 (±.05)	0.58 (±.23)	0.04 (±.07)	0.01 (±.02)	0.00 (±.00)	0.00 (±.00)	0.02 (±.05)	0.04 (±.05)
AWAC [36]	0.94 (±.20)	0.43 (±.43)	0.26 (±.33)	0.31 (±.32)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.00 (±.00)	0.07 (±.04)
DEX [2]	**1.00** (±.00)	0.73 (±.12)	0.94 (±.05)	0.73 (±.20)	0.63 (±.19)	0.18 (±.14)	0.02 (±.01)	0.00 (±.00)	0.02 (±.07)	0.05 (±.04)
ViSkill* [7]	-	-	-	-	-	**0.85** (±.08)	0.81 (±.04)	**0.57** (±.06)	-	-
T-STAR* [8]	-	-	-	-	-	0.67 (±.05)	0.65 (±.10)	0.45 (±.04)	-	-
Ours	**1.00** (±.00)	**0.95** (±.05)	**1.00** (±.00)	**0.84** (±.10)	**0.74** (±.12)	0.42 (±.08)	**1.00** (±.00)	0.48 (±.09)	**0.30** (±.12)	**0.37** (±.18)

"*" means the approach leverages manually pre-defined prior information for each specific task, such as task decomposition and chaining. "-" denotes not suitable.

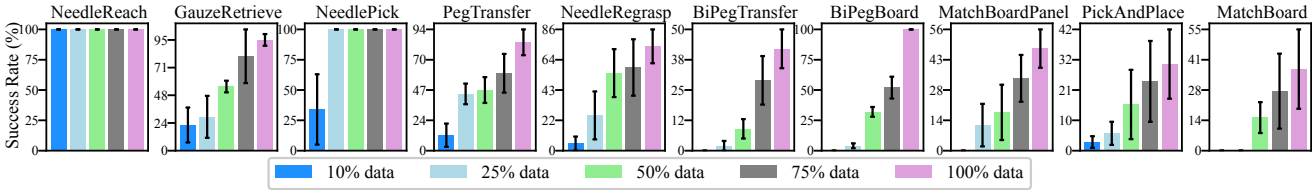

Fig. 3. Ablation results for the available data amount for the 10 different tasks. The mean value and standard deviation of the success rate for all tasks with different amount of available data are visualized to evaluate the performance of the trained model.

descriptions and demos of all the tasks. For every task, we provide 100 trajectories as the original dataset, which are generated from an expert script with priori knowledge.

Evaluation metric: In the evaluation process, we pretrain the policy on all the tasks with the cross-task objectives. For downstream task learning, we update the model on data of the specific task, only with the action prediction objective. When the model training was done, for testing of each task, we randomly set a start state (e.g., randomizing the initial positions of the needle and gripper for NeedlePick task) for 100 times, and count the number of successfully completions of the task. The success rate is the number of successful trials divided by the total number of trials. To evaluate the stability of the model, we set 5 different seeds to generate the random starting states in experiments for each task, and report the average success rate and standard deviation of multiple runs.

B. Comparison with state-of-the-art methods

We provide both offline and online methods as the compared baselines. The offline methods are divided into two categories, offline reinforcement learning and imitation learning. For the offline reinforcement learning methods part, we involve **PLAS** [29] and **IQL** [30]. The offline reinforcement learning methods are implemented from d3rlpy library [38] with original parameters. For the imitation learning methods part, we select **BC** [31] and **VINN** [32]. The involved imitation learning methods are realized from our previous work [2]. We also report the results from **DT** [10], which is a basic decision-making method based on modeling sequences with transformers. For each baseline, we deploy the proposed data augmentation method to maintain entirely fairness.

The main results are shown in Table I. We observe that for the tasks with relatively low dimensions of state and action dimensions, e.g., NeedleReach, almost all the methods could get respectable success rates. For more complicated and long-horizon tasks, e.g., BiPegBoard and MatchBoardPanel, all the offline methods that are independent of the sequence reasoning capacity cannot reach the expected goal in the evaluations. Our model and **DT** [10] achieve success cases even in those sophisticated tasks with the sequence temporal modeling and reasoning ability of large transformer architecture from transformers. Compared with **DT** [10], our method shows an average 0.31 improvement on success rate since our design pretrain on multiple tasks with several extra losses and embeds the time-to-goal indicator. They guide future awareness and help learn the dynamics of the goal-conditioned paradigm to boost ultimate performance.

Although our method does not need to interact with environment, we also compared with the state-of-the-art online methods, which consists of **DDPGBC** [33], **AMP** [34], **CoL** [35], **AWAC** [36], **DEX** [2], **ViSkill** [7], and **T-STAR** [8]. We implement those algorithms based on OpenAI library [39] and our previous work [2], [7]. We present the experimental results in Table I. We find that compared with offline methods, the aggregate performance of the online methods has greatly improved since the active interaction with environments. However, powered by the strong reasoning among long sequences of the transformer architecture, our model still displays considerable performance compared to most of the online algorithms in involved tasks, only using 0.05 of required data in the training process. For those algorithms designed for specific tasks with pre-defined priori knowledge, such as **ViSkill** [7] and **T-STAR** [8] in BiPegTransfer, BiPegBoard, and MatchBoardpanel, our model still surpasses or holds at almost equal levels in two of the tasks. However, those methods [7], [8] that depend on task-specific decomposition cannot be generalized to other

TABLE II

ABLATION STUDY RESULTS FOR DIVERSE PRETRAINING CONFIGURATIONS WITH MEAN AND STANDARD DEVIATION FOR ALL TASKS.

Task	NeedleReach	GauzeRetrieve	NeedlePick	PegTransfer	NeedleRegrasp	BiPegTransfer	BiPegBoard	MatchBoardPanel	PickAndPlace	MatchBoard
w/o DP[1]	1.00 (±.00)	0.80 (±.08)	0.91 (±.04)	0.80 (±.07)	0.67 (±.08)	0.33 (±.05)	0.92 (±.05)	0.35 (±.06)	0.21 (±.04)	0.33 (±.05)
w/o TP[2]	1.00 (±.00)	0.88 (±.07)	0.91 (±.05)	0.71 (±.07)	0.65 (±.11)	0.32 (±.07)	0.83 (±.14)	0.27 (±.08)	0.20 (±.08)	0.29 (±.19)
w/o SR[3]	1.00 (±.01)	0.92 (±.04)	0.93 (±.01)	0.72 (±.08)	0.67 (±.09)	0.31 (±.15)	0.91 (±.07)	0.36 (±.04)	0.26 (±.07)	0.25 (±.06)
w/o Pretrain	0.99 (±.02)	0.79 (±.07)	0.85 (±.11)	0.68 (±.11)	0.64 (±.16)	0.28 (±.07)	0.88 (±.09)	0.33 (±.19)	0.18 (±.08)	0.28 (±.09)
Ours	1.00 (±.00)	0.95 (±.05)	1.00 (±.00)	0.84 (±.10)	0.74 (±.12)	0.42 (±.08)	1.00 (±.00)	0.48 (±.09)	0.30 (±.12)	0.37 (±.18)

[1] without dynamics prediction, [2] without time-to-goal prediction, [3] without sequence reconstruction.

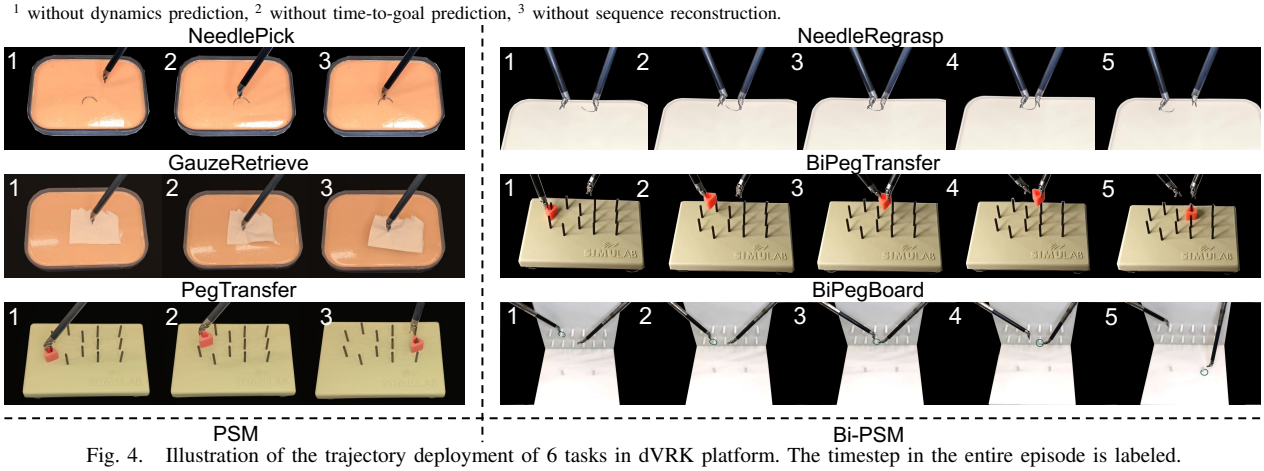

Fig. 4. Illustration of the trajectory deployment of 6 tasks in dVRK platform. The timestep in the entire episode is labeled.

tasks, which limits their valid scenarios and makes the overall performance of this type of algorithm inferior to ours.

C. Ablation study

1) Effect of cross-task pretraining: The ablation results in different tasks of our cross-task pretraining strategy are shown in Table II. We deploy the same model architecture but using different pretraining strategies. Our method improves the average success rate by 0.19 compared to methods that do not use pretraining. In addition, these models are not as robust as the model pretrained to other tasks and show a large variance in different test epochs, which shows that our cross-task pretraining improves stability through the understanding of the general task paradigm.

2) Effect of multi-objective training: We also provide the ablation results for each auxiliary component of the deployed training objectives in the pretraining process in Table II. The evaluation results demonstrate that the involvement of each objective item can help boost the average success rate and limit the instability of the performance, which is attributed to reasoning capacity among sequence items and understanding of task dynamics of the transformer architecture, supported by the designed multiple training objectives.

3) Influence of data amount: In order to explore the impact of training data volume on experimental results, we split the original dataset of each task by the selected typical percentage. The multi-objective pretraining pattern among all the tasks and the proposed data augmentation approach is also deployed. From the results in Fig. 3, we summarize that for the simple task, e.g., NeedleReach, even though the available data is constrained to a low amount, the final performance is still considerable. For other complicated tasks with richer state spaces and long horizons to complete, the performance collapses with the reduction of the data amount. This phenomenon means that more complex tasks require more data for training to ensure effectiveness of the model.

D. Real-world robot deployment

The final trajectory of our model is deployed in dVRK equipment to evaluate its practicality and employability. The environmental scenes come from 3D printing of objects, which are the digital twin of SurRoL simulator [1], [15]. The involved tasks contain Needlepick, GauzeRetrieve, PegTransfer, NeedleRegrasp, BiPegTransfer, and BiPegBoard, where the first three tasks are with patient-sided manipulator (PSM) and the last three are with bimanual patient-sided manipulator (Bi-PSM). The finishing procedures of each task are shown in Fig. 4. The results illustrate that our method can support both PSM and Bi-PSM to manipulate the objects precisely, such as conveying pegs and regrasping needles. These findings illustrate that our model's predicted trajectories can be deployed in the real-world platform of dVRK, thanks to digital twin design of SurRoL.

V. CONCLUSION

We present goal-conditioned decision transformer, which leverages the large-scale transformers from LLMs to complete goal-conditioned surgical robot automation tasks. The proposed method depends on a novel sequence modeling with future indicators to realize temporal reasoning capability in goal-conditioned surgical tasks. Besides, we utilize multi-objective pretraining across multiple tasks to improve the sequence contextual reasoning and comprehension of the general goal-conditioned paradigm to anticipate precise decisions. Experiment results show that our method achieves superior performance and versatility across diverse tasks and the trajectory is deployable in dVRK to verify applicability.

REFERENCES

[1] Y. Long, W. Wei, T. Huang, Y. Wang, and Q. Dou, "Human-in-the-loop embodied intelligence with interactive simulation environment for surgical robot learning," *IEEE Robotics and Automation Letters*, 2023.

[2] T. Huang, K. Chen, B. Li, Y.-H. Liu, and Q. Dou, "Demonstration-guided reinforcement learning with efficient exploration for task automation of surgical robot," in *2023 IEEE International Conference on Robotics and Automation*, 2023, pp. 4640–4647.

[3] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2020, pp. 9651–9658.

[4] H. Su, Y. Hu, Z. Li, A. Knoll, G. Ferrigno, and E. De Momi, "Reinforcement learning based manipulation skill transferring for robot-assisted minimally invasive surgery," in *2020 IEEE International Conference on Robotics and Automation*. IEEE, 2020, pp. 2203–2208.

[5] W. Chi, G. Dagnino, T. M. Kwok, A. Nguyen, D. Kundrat, M. E. Abdelaziz, C. Riga, C. Bicknell, and G.-Z. Yang, "Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning," in *2020 IEEE International conference on robotics and automation*. IEEE, 2020, pp. 2414–2420.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[7] T. Huang, K. Chen, W. Wei, J. Li, Y. Long, and Q. Dou, "Value-informed skill chaining for policy learning of long-horizon tasks with surgical robot," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2023, pp. 8495–8501.

[8] Y. Lee, J. J. Lim, A. Anandkumar, and Y. Zhu, "Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization," in *5th Annual Conference on Robot Learning*, 2021.

[9] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[10] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 15 084–15 097.

[11] T. Yamagata, A. Khalil, and R. Santos-Rodriguez, "Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline RL," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 38 989–39 007.

[12] Q. Zheng, A. Zhang, and A. Grover, "Online decision transformer," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 27 042–27 059.

[13] K.-H. Lee, O. Nachum, M. S. Yang, L. Lee, D. Freeman, S. Guadarrama, I. Fischer, W. Xu, E. Jang, H. Michalewski *et al.*, "Multi-game decision transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 921–27 936, 2022.

[14] K. Hu, R. C. Zheng, Y. Gao, and H. Xu, "Decision transformer under random frame dropping," in *The Eleventh International Conference on Learning Representations*, 2023.

[15] J. Xu, B. Li, B. Lu, Y.-H. Liu, Q. Dou, and P.-A. Heng, "Surrol: An open-source reinforcement learning centered and dvrk compatible platform for surgical robot learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2021.

[16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[17] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[18] T. Osa, C. Staub, and A. Knoll, "Framework of automatic robot surgery system using visual servoing," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 1837–1842.

[19] B. W. King, L. A. Reisner, A. K. Pandya, A. M. Composto, R. D. Ellis, and M. D. Klein, "Towards an autonomous robot for camera control during laparoscopic surgery," *Journal of laparoendoscopic & advanced surgical techniques*, vol. 23, no. 12, pp. 1027–1030, 2013.

[20] I. Rivas-Blanco, C. J. Perez-del Pulgar, C. López-Casado, E. Bauzano, and V. F. Muñoz, "Transferring know-how for an autonomous camera robotic assistant," *Electronics*, vol. 8, no. 2, p. 224, 2019.

[21] K. L. Schwaner, D. Dall'Alba, P. T. Jensen, P. Fiorini, and T. R. Savarimuthu, "Autonomous needle manipulation for robotic surgical suturing based on skills learned from demonstration," in *2021 IEEE 17th international conference on automation science and engineering*. IEEE, 2021, pp. 235–241.

[22] S. Leonard, K. L. Wu, Y. Kim, A. Krieger, and P. C. Kim, "Smart tissue anastomosis robot (star): A vision-guided robotics system for laparoscopic suturing," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1305–1317, 2014.

[23] M. Khadem, C. Rossa, R. S. Sloboda, N. Usmani, and M. Tavakoli, "Mechanics of tissue cutting during needle insertion in biological tissue," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 800–807, 2016.

[24] V. Patel, S. Krishnan, A. Goncalves, C. Chen, W. D. Boyd, and K. Goldberg, "Using intermittent synchronization to compensate for rhythmic body motion during autonomous surgical cutting and debridement," in *2018 International Symposium on Medical Robotics (ISMR)*. IEEE, 2018, pp. 1–6.

[25] R. Bendikas, V. Modugno, D. Kanoulas, F. Vasconcelos, and D. Stoyanov, "Learning needle pick-and-place without expert demonstrations," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3326–3333, 2023.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[27] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[28] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," 2018.

[29] W. Zhou, S. Bajracharya, and D. Held, "Plas: Latent action space for offline reinforcement learning," in *Conference on Robot Learning*. PMLR, 2021, pp. 1719–1735.

[30] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *International Conference on Learning Representations*, 2021.

[31] M. Bain and C. Sammut, "A framework for behavioural cloning." in *Machine Intelligence 15*, 1995, pp. 103–129.

[32] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto, "The surprising effectiveness of representation learning for visual imitation," *arXiv preprint arXiv:2112.01511*, 2021.

[33] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE international conference on robotics and automation*. IEEE, 2018, pp. 6292–6299.

[34] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–20, 2021.

[35] V. G. Goecks, G. M. Gremillion, V. J. Lawhern, J. Valasek, and N. R. Waytowich, "Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2020, pp. 465–473.

[36] A. Nair, A. Gupta, M. Dalal, and S. Levine, "Awac: Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

[37] Med-AIR, https://med-air.github.io/SurRoL/.

[38] T. Seno and M. Imai, "d3rlpy: An offline deep reinforcement learning library," *Journal of Machine Learning Research*, vol. 23, no. 315, pp. 1–20, 2022.

[39] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.