USPilot: An Embodied Robotic Assistant Ultrasound System with Large Language Model Enhanced Graph Planner

Mingcong Chen[1,3] Siqi Fan[2,3] Guanglin Cao[3,4] Hongbin Liu[3,4]

Abstract—In the era of Large Language Models (LLMs), embodied artificial intelligence presents transformative opportunities for robotic manipulation tasks. Ultrasound imaging, a widely used and cost-effective medical diagnostic procedure, faces challenges due to the global shortage of professional sonographers. To address this issue, we propose USPilot, an embodied robotic assistant ultrasound system powered by an LLM-based framework to enable autonomous ultrasound acquisition. USPilot is designed to function as a virtual sonographer, capable of responding to patients' ultrasound-related queries and performing ultrasound scans based on user intent. By fine-tuning the LLM, USPilot demonstrates a deep understanding of ultrasound-specific questions and tasks. Furthermore, USPilot incorporates an LLM-enhanced Graph Neural Network (GNN) to manage ultrasound robotic APIs and serve as a task planner. Experimental results show that the LLM-enhanced GNN achieves unprecedented accuracy in task planning on public datasets. Additionally, the system demonstrates significant potential in autonomously understanding and executing ultrasound procedures. These advancements bring us closer to achieving autonomous and potentially unmanned robotic ultrasound systems, addressing critical resource gaps in medical imaging.

Index Terms—Medical robots, Robotic ultrasound, Large language models (LLMs), Planning

I. INTRODUCTION

Ultrasound is widely used in medical imaging. It provides real-time images of internal organs, blood vessels, and tissues without exposing patients to ionizing radiation. This technique is particularly useful for diagnosing conditions such as pregnancy, heart disease, and tumors. However, hospital sonographers face significant challenges due to the increasing demand for ultrasound services and the rapid pace of technological advancements. There are over 64,751 sonographers employed in the United States, but the number will need to increase by 27,600 according to a prediction by the Bureau of Labor Statistics [1]. In addition, sonographers commonly experience physical strain from their work. Repetitive motions often lead to joint discomfort, particularly in the wrists, upper back, and shoulders [2]. Research on diagnostic medical sonographers indicates that a significant majority—up to 91%—experience discomfort or pain while conducting ultrasound examinations [3]. To mitigate this issue, it is imperative to consider the implementation of automation, particularly in light of the
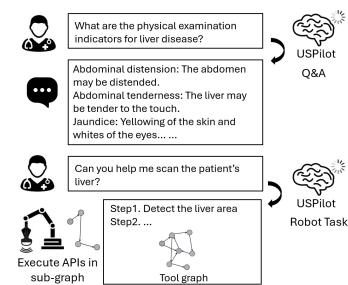
Fig. 1: The overview of USPilot: Answer the sonographer's normal medical questions or perform an automated ultrasound scan.

global shortage of ultrasound imaging professionals and the increasing demand for ultrasound-based diagnostic imaging.

The introduction of automated diagnostic and intraoperative ultrasound systems, which are either guided or assisted by robotic technology, represents a promising solution to the challenges previously outlined. By diminishing the workload of ultrasound technicians, these automated systems pave the way for more sophisticated image acquisition processes characterized by digital and numerical consistency. There are currently several robotic assistant US scanning systems for specific tasks [4] such as reinforcement learning [5], [6], imitation learning [7], [8] and predefined tasks [9]. However, these systems still require extensive domain knowledge or reward engineering, requiring doctors or engineers to select the specific functions individually. The human-robot interaction methods in these systems may also increase surgeons' workload or distract them from their current tasks, which do not align with their operating habits.

With the rise of large language models (LLMs), a subsequent surge in embodied artificial intelligence has been observed. LLMs have learned rich semantic knowledge of the world, so they can be naturally used for planning without complex definitions or additional training. This ability of LLMs to understand and generate human-like text based on vast amounts of training data allows them to be seamlessly integrated into various applications where decision-making
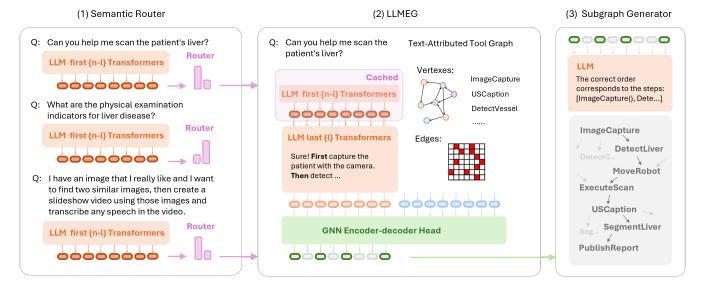
Fig. 2: The structure of USPilot: (1) The semantic router recognizes the user's intent. (2) If the intent is an executable task, LLMEG is invoked using the cached, unadapted Transformer to select potential APIs. (3) The LLM-based subgraph generator reorders the selected APIs into a directed graph.

and planning are crucial. Some research, such as Octopus [10] and TAPA [11], illustrated the ability of task reasoning and planning for daily instructions in simulated environments. Some research has trained on robot manipulation data and acting the robot arm with the pre-defined vision feedback function [12]–[14]. [15] demonstrating its ability to perform common surgical sub-tasks through human language instruction. However, these methods achieve low-level robot control by using LLMs to code under the guidance of prompt engineering, which will face the illusion of the LLMs. The illusion of LLMs in low-level robot control coding may cause safety issues, especially in medical robotics applications.

In this paper, we present USPilot, a comprehensive framework designed for autonomous ultrasound acquisition. The framework incorporates a semantic router powered by Large Language Models (LLMs) that interpret user queries related to ultrasound procedures and translate natural language into instructions for robotic ultrasound system operation. A key innovation of our framework is the implementation of an LLM-enhanced Graph Neural Network (GNN) for robotic tool selection and planning. By leveraging adapter-based ultrasound knowledge integration, USPilot demonstrates successful autonomous ultrasound image acquisition on a physical robotic platform, marking a significant step toward fully autonomous robotic ultrasound systems. This advancement suggests promising possibilities for unmanned medical imaging applications.

II. RELATED WORKS

LLMs as Planners For robots, language serves as a natural medium to guide high-level control. Many methods have attempted to apply the prior knowledge of LLMs and Vision Language Models (VLMs) to robot tasks. These models can assist in breaking down long-horizon tasks [16] and responding in interactive environments. Techniques like Chain-of-Thought [17] and few-shot prompting are useful for ensuring that each step of a plan is well-reasoned and atomic. However, LLM illusions and associations can still introduce instability and errors, which is unacceptable in ultrasound tasks. To address these issues, researchers have designed prompts or trained LLMs to adapt to frameworks like linear temporal logic [18], code generation [16], [19], [20], and behavior trees [21]. These constraints help preserve associations while ensuring stability. In our approach, we use a text-attributed graph (TaG) to constrain the task space for LLM planning.

TAG with LLMs In TaGs, textual descriptions of entities act as vertexes. At the same time, the edges represent relationships between neighbors, which, in our case, correspond to robot tasks in the ultrasound scan and their execution logic. To leverage the power of LLMs on the text in graphs, existing methods assign LLMs three roles: aligner, predictor, and enhancer. Researchers [22], [23] align the output from Graph Neural Networks (GNN) with LLMs to create a shared latent space for knowledge transformation. Similar results can be achieved using a projector applied to the GNN encoder, like GraphGPT [24], [25], which helps make the graph embedding more understandable for the LLM predictor. However, aligner LLMs can introduce illusions into graphs, and LLMs as predictors require high-quality labeled datasets to ensure accurate projection. Since LLMs can significantly enhance or refine textual features, they are used in [26], [27] to improve text features for GNNs. In our ultrasound robot task, this structure enriches the original task information while limiting the LLM cognition domain.

III. METHODOLOGY

A. Problem Formulation

In this work, the system adapts to the user's instruction types to perform ultrasound robot manipulation or answer ultrasound-related questions. During usage, the LLM-based
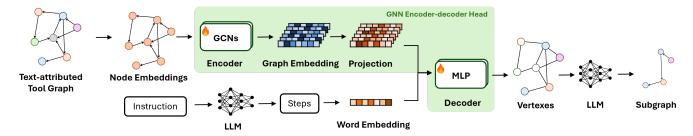
Fig. 3: The structure of LLMEG: Select a toolchain G^* from G based on the textual information provided by I.

planner receives a text instruction I and a task description graph G. The proposed system can generate a text answer A or a serial executable robotic command G^* based on the user intent perception policy π. The answering process can be formulated as $A \bigvee G^* = \pi(I, G)$.

Robotic ultrasound manipulation can be represented as a task planning problem on a graph, where each step in the system is highly dependent on its preceding steps. The graph is defined as a TaG with the structure $G = (V, E, T)$. The vertex $v \in V$ represents a task in the ultrasound scan, which is defined as an application programming interface (API) in the robotic system. The API description text $t \in T$ provides a linguistic description of the function of each API. The edge $(x, y) \in E$ indicates the dependency between two APIs. The graph-based task planning aims to identify an optimal subgraph G^* that can complete the given task instruction I.

B. Structure Overview

The overview of USPilot is depicted in Fig 2. The semantic router is a dynamic routing mechanism connected to the LLM transformer structure. It enables user intent recognition, which helps the system load the appropriate knowledge or graphs. Related subject knowledge is trained into the LLM through an adapter structure, allowing the model to better respond to user intent by adjusting the gating factor of the adapter layers.

If the user intent involves executable tasks, the router switches to the Large Language Model Enhanced Graph (LLMEG). The LLMEG comprises an LLM-enhanced Graph Neural Network (GNN) head and a subgraph generator. The LLM-enhanced GNN head selects APIs by sending LLM-generated word embeddings and GNN-generated graph embeddings to the decoder. The LLM-based subgraph generator defines the path for API execution. When the user intent involves a general medical question, the router switches the inference path to a pre-trained ultrasound-related question-answer adapter to fulfill the medical inquiry requirements. The semantic router, adapters, and LLMEG are all hot-plugging in our USPilot, which makes it suitable for separate training, upgrading, and adapting.

C. Large Language Model Enhanced Graph

LLMs possess a remarkable ability to interpret user requests and decompose complex commands into simpler subtasks. This task decomposition capability is leveraged using the LLM to generate subtasks from input instructions, which are then converted into word embeddings, denoted as e_w. The vertex features F_v are the embeddings of each vertex's textual description generated by the LLM. The graph adjacency matrix A is a $2 \times N$ matrix representing the connectivity between vertexes $x, y \in E(x, y)$. In LLMEG, the GNN encoder is implemented using two layers of graph convolution network (GCN) with the inputs as vertex features F_v and graph adjacency matrix A. The encoder outputs graph embeddings e_g of size $N \times F'$, computed as:

$$e_g = \sigma(WAF_v) \tag{1}$$

where N is the number of vertexes, F' is the dimension of the GCN layer, W is a learned weighted matrix and σ is the LeakyReLU activation function. The graph embedding is concatenated with the word embeddings to form the feature embedding $e_f = [\eta e_g; e_w]$ as the input to the decoder, where η is a trainable projection matrix that maps e_g to the same latent space as e_w. The decoder consists of multi-layer perceptron (MLP) layers followed by a sigmoid activation function, producing the output:

$$o = sigmoid(MLPs(e_g)) \tag{2}$$

The output, with size N, indicates whether each vertex should be selected for the given task. To train the LLMEG encoder-decoder, the cross entropy loss $CE = -[v log(\hat{v}) + (1-v) log(1-\hat{v})]$ is used, where v is the ground truth label indicating whether a vertex should be selected, and $\hat{v}$ is the predicted probability.

The result o from the LLMEG result is used to construct an undirected graph that identifies the APIs required to complete the task. To determine the sequence of API executions, classical path-searching algorithms, such as Depth First Search (DFS), can be applied. However, DFS lacks the ability to select the appropriate starting point due to its lack of semantic understanding. To address this limitation, the LLM is utilized again to transform the undirected graph into a directed graph G^*, incorporating semantic information from the subtasks. The system then executes the vertexes sequentially based on the order defined in the directed graph G^*.

D. Ultrasound knowledge embedded and user intent understanding

To adapt the model for understanding ultrasound tasks, a lightweight adaptation method, referred to as the LLaMA Adapter, is employed to efficiently fine-tune the original LLM
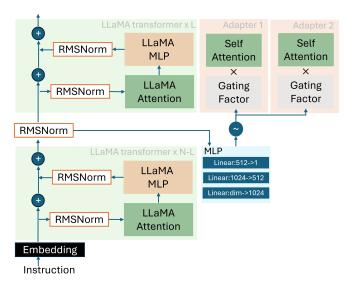
Fig. 4: The dynamic routing mechanism switches the forward path among adapters according to the user's instructions.

into an instruction-following model specific to ultrasound-related tasks. The LLaMA Adapter achieves this by inserting prompted parameters into the top L layers of the N-layers LLaMA transformer model, enabling the model to acquire a new semantic understanding. In the USPilot framework, two adapters are trained: one based on ultrasound-related questions-answer pairs and the other tailored for ultrasound scanning tasks.

Upon activating the LLaMA Adapter, the model experiences a reduced capability to handle general questions. As a comprehensive language model, retaining proficiency across a wide range of queries, including those outside the specific domain of robotic tasks is essential. However, the multi-adapter structure may compromise the model's performance in general or medical question-answering scenarios while focusing on robotic tasks. A dynamic routing mechanism is integrated to address this limitation and restore versatility. The core component of this router is a three-layer MLP with a sigmoid activation gate. The complete architecture, including the dynamic router and adapters, is detailed in Fig4. The adaption prompt for the l-th layer of the LLaMA adapter can be defined as the concatenated instruction knowledge learned prompt P_l and word tokens T_l as

$$[P_l; T_l] \quad (3)$$

, where the learned prompt P_l acts as a prefix turning.The dynamic router manages and routes tasks effectively based on input. The learned adaption prompt can be illustrated as

$$P_l = sigmoid(MLP(T_{n-l})) * P_0 + sigmoid(MLP(T_{n-l})) * P_1 \quad (4)$$

, where P_0 and P_1 represent prompt vectors from different adapters. The training dataset is divided strategically between general question-answering tasks and specific robotic task instructions. This division enables the dynamic router to adapt efficiently between these two domains, balancing specialized and general capabilities. The router mitigates overfitting by

appropriately routing tasks, ensuring the model retains general applicability while specializing in robotic tasks. Furthermore, the architecture demonstrates the potential for routing multiple questions or tasks using multiclass activation functions such as softmax. In this case, the routing weights can be expressed as: $P = softmax(MLP(T_{n-l})) * [P_0; P1; P_2...P_n]$.

IV. EXPERIMENT

In this section, we present several experiments designed to assess the performance of the USPilot's components. The LLMEG module was evaluated on a publicly available dataset compared to other methods. The semantic router and ultrasound knowledge adapter were also validated on a pre-collected dataset. Furthermore, the USPilot system was integrated with our automated robotic ultrasound scanning system, demonstrating its functionality in a physical environment.

A. Comparison Experiment for LLMEG

We evaluated the proposed LLMEG with three public datasets from TaskBench [28] for instructed API selection or task planning. **Multimedia dataset**: This dataset includes user-centric tasks such as file downloading and video editing. It is represented as a graph with 40 vertexes and 449 edges. **HuggingFace dataset**: This dataset contains various AI models covering tasks across language, vision, audio, and video domains. Each task is treated as a tool for addressing specific problems, forming a graph with 23 vertexes and 225 edges. **Dailylife APIs**: This dataset comprises tools for daily services such as web search and shopping. It is represented as a graph with 40 vertexes and 1,560 edges.

We employed 3 metrics to evaluate the methods' performance. Accuracy is the successful rate of completing the given tasks. The vertex F1 score evaluates the labels assigned to individual vertexes, which can be indicated as

$$F1_{vertex} = \frac{2 \times Precision_{vertex} \times Recall_{vertex}}{Precision_{vertex} + Recall_{vertex}} \quad (5)$$

, where the $Precision_{vertex}$ is the number of correctly identified vertexes among all predicted vertexes. $Recall_{vertex}$ is the number of correctly identified vertexes out of the total actual vertexes, representing correct API calls. The edge F1 measures the model's ability to predict valid or expected API call sequences, which can be assigned as
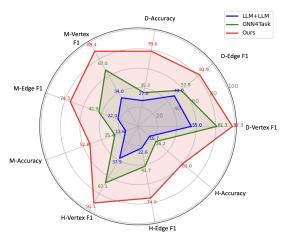
$$F1_{edge} = \frac{2 \times Precision_{edge} \times Recall_{edge}}{Precision_{edge} + Recall_{edge}} \quad (6)$$

, where $Precision_{edge}$ is the number of correctly identified edges among all predicted edges, and $Recall_{edge}$ is the number of correctly identified edges out of the total actual edges in API call sequences.

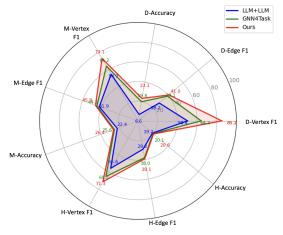In the compression experiment, we employed two alternative methods: **LLM+LLM:** The LLM+LLM method involves collaboration between two LLMs. Initially, the text description of all the tools and the user request is provided to the first LLM, which generates the necessary substeps to complete the task. Subsequently, the generated substeps are combined with the tool descriptions to form another prompt for the second

TABLE I: Vertex F1 score, edge F1 score and accuracy across three datasets (Dailylife, Multimedia and Huggingface) for GNN4Task with GPT4-turbo, LLMEG (LLaMA3-8b) without subgraph, and LLMEG (LLaMA3-8b) with subgraph
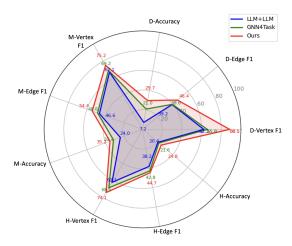
Method	LLM	Parameters	Dailylife			Multimedia			Huggingface		
			Vertex F1	Edge F1	Acc	Vertex F1	Edge F1	Acc	Vertex F1	Edge F1	Acc
GNN4TaskPlan	GPT4-turbo	8*222b	97.10	88.52	-	88.56	69.60	-	77.01	50.49	-
Ours w/o subgraph	LLaMA3	8b	97.29	26.35	39.12	89.37	50.91	28.0	91.09	52.59	33.8
Ours	LLaMA3	8b	97.29	88.87	78.64	89.37	74.35	54.6	91.09	74.94	59.6

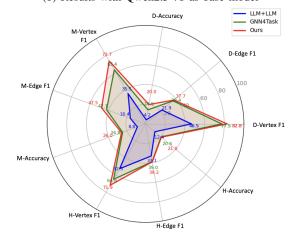
(a) Results with LLaMA3-8b as base model

(b) Results with Qwen2.5-7b as base model

(c) Results with Qwen2.5-3b as base model

(d) Results with Gemma2-2b as base model

Fig. 5: Comparesion of LLM+LLM, GNN4Task, and our proposed method across Dailylife, Multimedia, and Huggingface datasets, where 'M-' indicates the Multimedia dataset, 'D-' indicates the Dailylife dataset and 'H-' represents the Huggingface dataset.

LLM, which selects the tools to be utilized. The selected tools are considered as vertexes, while two adjacent vertexes can be regarded as edges. **GNN4Task:** The GNN4Task generates subtasks using LLMs and employs a small language model (e5-355M) to embed each subtask. A GNN processes each embedded subtask to identify vertexes, with edges defined as connections between adjacent vertexes.

We employed the 8-billion-parameter LLaMA3 model in our proposed method LLMEG and compared it with GNN4Task across the Dailylife, Multimedia, and Hugging-Face datasets. Notably, GNN4task is equipped with one of the powerful LLMs, GPT4-tubo, a mixture of eight 222 billion parameters experts. As shown in Table I, the proposed method LLMEG achieves superior performance across

all three datasets compared with the GNN4Task method. In the Dailylife dataset, LLMEG achieves vertex and edge F1 scores of 97.29% and 88.87%, respectively, outperforming GNN4Task (97.10% and 88.52%). Both methods perform well on this dataset due to the inherent ability of LLMs to understand daily service descriptions. For the Multimedia dataset, our method improves vertex and edge F1 scores by 0.81% and 4.75%, respectively, over GNN4Task. On the more professional HuggingFace dataset, our proposed method significantly enhances vertex selection and edge prediction by 14.8% and 24.45%, respectively. We also compared our method with a variant that replaces the LLM-generated subgraph with depth-first search (DFS) for planning. Using subgraph planning,

the edge F1 score increases by 62%, 23.4%, and 22.4%, respectively, while the total accuracy increases by 39.52%, 26.6%, and 25.8% in the datasets. These results highlight the importance of semantic understanding in generating correct API execution sequences, even when API selection is accurate. The improvement in edge prediction varies between datasets and is particularly pronounced in the Dailylife dataset, which contains an average of 39 edges per vertex. As graph complexity increases, the significance of semantic information becomes more evident.

After employing the LLMEG, we conducted experiments using different LLMs with varying parameter sizes to assess planning capabilities. As shown in Fig. 5a, our proposed method outperforms the original LLM-based method and GNN4Task in both vertex and edge F1 scores by over 40% across all datasets when using LLaMA3-8b and its embeddings. For other LLMs, LLMEG also abstracts word embedding with 137M parameters small language model (nomic-embed-textv1.5). Our proposed method achieves better performance across all metrics for the similar size LLM Qwen2.5-7b. However, when integrated with smaller LLMs shown in Fig. 5c and Fig. 5d, our LLMEG can result a high value only in vertex F1 score but struggles with edge F1 scores and accuracy. This is due to smaller LLMs' limited semantic understanding capabilities, which are insufficient for effective subgraph replanning.

B. User intent understanding and ultrasound knowledge embedded

The user intent prediction MLP was trained with 1.2K instructions on robot tasks, general question-answering (QA), and ultrasound-related questions with an equal ratio of 1:1:1. We used sigmoid as the activation function to classify the user inputs as QA inquiries or ultrasound robot commands. As shown in Table II, the model was evaluated on a 14K dataset, comprising 9K QA instances and 5K robotic commands. The results demonstrate the MLP classification layer can reach an accuracy of 100% for QA intent and 97% for robotic command intent.

In addition, the ultrasound robotic adapter was fine-tuned using 2K instructions on seven body parts: carotid vessel, thyroid, liver, kidney, spleen, femoral artery, and gallbladder. The dataset comprises instructions and expected subtasks, in which the instructions have the patient's direct command and symptom description, such as "Scan the patient's liver" and "The patient has jaundice; check him with ultrasound." An evaluation set was prepared for six body parts and functional commands to assess planning capabilities. Furthermore, the bladder and pancreas, which did not appear in the training set, were considered unseen senses for evaluation. The Success Rate (SR) is the fraction of successfully completed instructions. As Table III indicates, USPilot achieved a 99% success rate for single API executions. The fine-tuned adapter can comprehend the instructions, achieving a 95% success rate for direct commands and 78.5% for symptom-based descriptions involving seen body parts. For unseen body parts (bladder and pancreas), USPilot leveraged the LLM's generalization

TABLE II: Accuracy of user intent classification by USPilot

Type	Num	Accuracy
QA	9167	100%
Robot	5418	97%

TABLE III: Success rate of the fine-tuned USPilot adapter in direct commanding and symptom description across seven seen body parts, four operation situations, and two unseen body parts
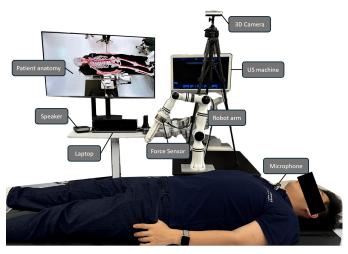
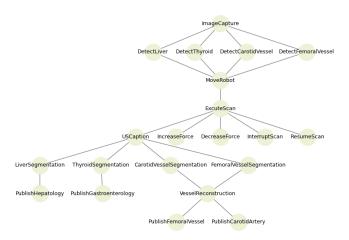|	Direct Command	Symptom description
Seen		
Carotid Artery	95%	96%
Thyroid	100%	85%
Liver	100%	85%
Kidney	95%	75%
Spleen	95%	85
Femoral Artery	90%	75%
Gallbladder	85%	80%
Total	**95%**	**78.5%**
Unseen		
Bladder	37.5%	15.5%
Pancreas	94.7%	47.3%
Total	**71.4%**	**25.7%**
Single API		
Interrupt	100%	-
Continue	100%	-
Increase force	100%	-
Decrease force	95%	-
Total	**99%**	-

ability to achieve a 71.4% success rate for direct commands. However, it achieved only a 25.7% success rate for symptom-based descriptions due to the absence of specific knowledge about these body parts in the training data. Instead of effectively generalizing to entirely unseen body parts, it exhibits knowledge transfer between related body parts, such as from the gallbladder to the pancreas or from the kidney to the bladder, which results in lower success rates.

C. USPilot in the real world ultrasound senario

The system comprises an ultrasound robotic system, a 3D camera system, and a high-performance computing server. To ensure patient safety, the ultrasound robotic system performs pre-defined trajectory scanning with admittance control. It adjusts the probe's orientation based on the contact point to achieve an optimal contact angle with the skin, ensuring the acquisition of clear ultrasound images [29]. To determine the scanning trajectory and cover various body parts, the system integrates a patient anatomy avatar reconstruction model [30], which provides the positions of key anatomical structures such as the carotid artery, thyroid, and liver. USPilot receives instructions from the sonographer, who can pose medical questions or command the robotic system to perform ultrasound scans.

As shown in Fig. 6a, the physical setup consists of an ultrasound system (Pioneer H20, Angell, China) equipped with a linear probe and a convex probe mounted on a 6-axis collaborative robotic arm (RM65, Realman, China). A probe fast assemble machine is positioned adjacent to the robot system. A 3D camera (Realsense D455, Intel, US) mounted on the robot platform provides anatomical structure positioning

(a) The Physical setup of USPilot.

(b) The graph structure of USPilot with 21 API vertex and 24 edges.

Fig. 6: USPilot in a Real World Ultrasound Senario

TABLE IV: Results from the real-world ultrasound scan with USPilot

Type	Num	Instruction example	SR
Carotid vessel	10	I got dizzy recently, the doctor said my carotid vessel may blocked.	10/10
Liver	10	My physical examination report shows that my transaminase is elevated. Please help me check my body.	10/10
Force adjust	5	You pressed me a little pain.	3/5
Interrupt	5	I feel not good.	5/5

and patient orientation information, serving as part of the system's environmental feedback. With the patient anatomy structure, the system determines the initial scanning position for different targets. A 6-axis force sensor (M3815D, SRI, China) is integrated into the robot's end-effector to ensure patient safety and provide orientation feedback for the system. There are 21 APIs and 24 edges defined in the USPilot system, as depicted in the graph structure in Fig 6b. The system employs PicoVoice as a wake-word listener for user-machine interaction to ensure accessibility and ease of use. The user can activate the system by calling 'Doctor' or 'Hey, Doctor'. Upon activation, the user's speech is processed by the VOSK speech recognition toolkit and transcribed into text to USPilot. The system can then convert the generated text or pre-defined API text into audio using EdgeTTS, which is played back to the user.

In the real-world scenes, the USPilot was evaluated on two targets with 10 testing rounds per target and 5 additional rounds for single API commands. The robot was required to perform automatic scanning based on varying instructions for each target. We employed the success rate for evaluation, defined as the proportion of testing rounds in which robots successfully completed all assigned tasks. In the scanning task, the robot can successfully locate the commanded target and perform the scan with a 100% success rate. However, for the force adjustment task, USPilot achieved only a 60% success rate, consistent with the results in Table III. This limitation is attributed to the semantic interpretation of the "feeling uncomfortable" instruction, which prompts the system to interrupt the scan rather than solely reducing the contact force.

V. DISCUSSION & FUTURE WORK

The results in Table. I show that the proposed LLMEG method with LLaMA3-8b outperforms GNN4TaskPlan with GPT-4turbo-8*222b. As can be seen from Fig. 5, a reduction in word embedding size significantly impacts the performance of the proposed method, preventing it from achieving the same metrics as the LLaMA3-8b model. Additionally, as the size of the LLMs decreases, the performance of the proposed method declines substantially, indicating that the quality of steps generated by LLMs strongly influences the GNN classification model. Even with a good API selection, the order of the API calls heavily depends on the LLM's ability to plan subgraphs in the proposed method. Therefore, future research should focus on API selection and optimizing the planning of API call sequences.

As discussed in Section IV-B, the proposed USPilot system provides the potential for an autonomous robotic ultrasound platform. The system effectively distinguishes between user inputs as either questions or robot-executable commands. Nevertheless, the fine-tuned adapter in robotic ultrasound manipulation does provide good performance in executing direct commands and interpreting symptom descriptions. However, the dataset is still not large enough to ensure reliable and stable performance in such a medical scenario. Expanding the dataset with more diverse language instructions would be essential for further fine-tuning.

In section IV-C, we demonstrate the USPilot with the physical setup to perform real ultrasound scans on the human body. The current robotic ultrasound system can achieve unsupervised scans on two simple organs. However, the physical setup still could not provide enough function for medical diagnosis

and could not cover as many body parts as the simulated data. Consequently, developing additional applications to address more complex medical scenarios is imperative. Moreover, the current proposed system lacks of the ability of replanning, which should also be discussed in future works.

VI. Conclusion

This paper introduces USPilot, an embodied framework for an autonomous ultrasound robotic system powered by an LLM-enhanced GNN. Through evaluations of LLMEG, semantic routers on datasets, and robot performance in real-world scenarios, USPilot shows stable performance in ultrasound robotic tasks and enhanced capabilities in managing ultrasound-related QA inquiries. By reimplementing existing methods across datasets derived from TaskBench and modifying the base model, the system exhibits a degree of knowledge transfer and adaptability to different domains. However, the size of the base model parameters remains a critical factor influencing performance, and the system lacks the ability to learn from past errors. Addressing these limitations represents a promising direction for future research on autonomous ultrasound robotic systems.

References

[1] S. Murphey, "Work related musculoskeletal disorders in sonography," 2017.

[2] G. Harrison and A. Harris, "Work-related musculoskeletal disorders in ultrasound: Can you reduce risk?," *Ultrasound*, vol. 23, no. 4, pp. 224–230, 2015.

[3] K. Evans, S. Roll, and J. Baker, "Work-related musculoskeletal disorders (wrmsd) among registered diagnostic medical sonographers and vascular technologists: a representative sample," *Journal of Diagnostic Medical Sonography*, vol. 25, no. 6, pp. 287–299, 2009.

[4] Y. Bi, Z. Jiang, F. Duelmer, D. Huang, and N. Navab, "Machine learning in robotic ultrasound imaging: Challenges and perspectives," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, 2024.

[5] K. Li, J. Wang, Y. Xu, H. Qin, D. Liu, L. Liu, and M. Q.-H. Meng, "Autonomous navigation of an ultrasound probe towards standard scan planes with deep reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8302–8308, IEEE, 2021.

[6] G. Ning, X. Zhang, and H. Liao, "Autonomic robotic ultrasound imaging system based on reinforcement learning," *IEEE transactions on biomedical engineering*, vol. 68, no. 9, pp. 2787–2797, 2021.

[7] Q. Men, C. Teng, L. Drukker, A. T. Papageorghiou, and J. A. Noble, "Multimodal-guidenet: Gaze-probe bidirectional guidance in obstetric ultrasound scanning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 94–103, Springer, 2022.

[8] X. Deng, Y. Chen, F. Chen, and M. Li, "Learning robotic ultrasound scanning skills via human demonstrations and guided explorations," in *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 372–378, IEEE, 2021.

[9] M. Chen, Y. Huang, J. Chen, T. Zhou, J. Chen, and H. Liu, "Fully robotized 3d ultrasound image acquisition for artery," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2690–2696, IEEE, 2023.

[10] J. Yang, Y. Dong, S. Liu, B. Li, Z. Wang, C. Jiang, H. Tan, J. Kang, Y. Zhang, K. Zhou, *et al.*, "Octopus: Embodied vision-language programmer from environmental feedback," *arXiv preprint arXiv:2310.08588*, 2023.

[11] Z. Wu, Z. Wang, X. Xu, J. Lu, and H. Yan, "Embodied task planning with large language models," *arXiv preprint arXiv:2307.01848*, 2023.

[12] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*, pp. 287–318, PMLR, 2023.

[13] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[14] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[15] M. Moghani, L. Doorenbos, W. C.-H. Panitch, S. Huver, M. Azizian, K. Goldberg, and A. Garg, "Sufia: Language-guided augmented dexterity for robotic surgical assistants," *arXiv preprint arXiv:2405.05226*, 2024.

[16] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530, IEEE, 2023.

[17] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022.

[18] Z. Yang, S. S. Raman, A. Shah, and S. Tellex, "Plug in the safety chip: Enforcing constraints for llm-driven robot agents," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14435–14442, IEEE, 2024.

[19] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10608–10615, IEEE, 2023.

[20] S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, and H. Li, "Instruct2act: Mapping multi-modality instructions to robotic actions with large language model," *arXiv preprint arXiv:2305.11176*, 2023.

[21] H. Zhou, Y. Lin, L. Yan, J. Zhu, and H. Min, "Llm-bt: Performing robotic adaptive tasks based on large language models and behavior trees," *arXiv preprint arXiv:2404.05134*, 2024.

[22] X. Ren, W. Wei, L. Xia, L. Su, S. Cheng, J. Wang, D. Yin, and C. Huang, "Representation learning with large language models for recommendation," in *Proceedings of the ACM on Web Conference 2024*, pp. 3464–3475, 2024.

[23] T. Zou, L. Yu, Y. Huang, L. Sun, and B. Du, "Pretraining language models with text-attributed heterogeneous graphs," *arXiv preprint arXiv:2310.12580*, 2023.

[24] R. Chen, T. Zhao, A. Jaiswal, N. Shah, and Z. Wang, "Llaga: Large language and graph assistant," *arXiv preprint arXiv:2402.08170*, 2024.

[25] J. Tang, Y. Yang, W. Wei, L. Shi, L. Su, S. Cheng, D. Yin, and C. Huang, "Graphgpt: Graph instruction tuning for large language models," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 491–500, 2024.

[26] S. Sun, Y. Ren, C. Ma, and X. Zhang, "Large language models as topological structure enhancers for text-attributed graphs," *arXiv preprint arXiv:2311.14324*, 2023.

[27] H. Liu, J. Feng, L. Kong, N. Liang, D. Tao, Y. Chen, and M. Zhang, "One for all: Towards training one graph model for all classification tasks," *arXiv preprint arXiv:2310.00149*, 2023.

[28] Y. Shen, K. Song, X. Tan, W. Zhang, K. Ren, S. Yuan, W. Lu, D. Li, and Y. Zhuang, "Taskbench: Benchmarking large language models for task automation," *arXiv preprint arXiv:2311.18760*, 2023.

[29] G. Cao, M. Chen, J. Hu, and H. Liu, "An ultra-fast intrinsic contact sensing method for medical instruments with arbitrary shape," *IEEE Robotics and Automation Letters*, 2023.

[30] T. Zhou, M. Chen, G. Cao, J. Hu, and H. Liu, "Inverse kinematics embedded network for robust patient anatomy avatar reconstruction from multimodal data," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3395–3402, 2024.