VIoTGPT: Learning to Schedule Vision Tools Towards Intelligent Video Internet of Things

Yaoyao Zhong, Mengshi Qi, Rui Wang, Yuhan Qiu, Yang Zhang, Huadong Ma*

[1]State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, China
{zhongyaoyao, qms, mhd}@bupt.edu.cn

Abstract

Video Internet of Things (VIoT) has shown full potential in collecting an unprecedented volume of video data. How to schedule the domain-specific perceiving models and analyze the collected videos uniformly, efficiently, and especially intelligently to accomplish complicated tasks is challenging. To address the challenge, we build VIoTGPT, the framework based on LLMs to correctly interact with humans, query knowledge videos, and invoke vision models to analyze multimedia data collaboratively. To support VIoTGPT and related future works, we meticulously crafted the VIoT-Tool dataset, including the training dataset and the benchmark involving 11 representative vision models across three categories based on semi-automatic annotations. To guide LLM to act as the intelligent agent towards intelligent VIoT, we resort to the ReAct instruction tuning method based on VIoT-Tool to learn the tool capability. Quantitative and qualitative experiments and analyses demonstrate the effectiveness of VIoTGPT. We believe VIoTGPT contributes to improving human-centered experiences in VIoT applications.

Projects — https://github.com/zhongyy/VIoTGPT
Long Version — https://arxiv.org/pdf/2312.00401

Introduction

The ubiquitous visual sensors, contemporary communication technologies, and high-capacity networking have enabled the potential and widely-usage of Video Internet of Things (VIoT), *i.e.*, internetworking of large-scale visual sensors, in collecting the unprecedented volume of video data and therefore offering a full ambient environment monitoring (Ma 2011; Mohan et al. 2017; Chen 2020, 2023).

Leveraging the power of perceiving techniques driven by deep learning (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016; Dosovitskiy et al. 2021), the acquired video data is eagerly expected to help with various VIoT applications, such as large geographical area monitoring for smart transportation (Liu et al. 2016; Zantalis et al. 2019; Wei, Wu, and Ma 2019) and public safety (Liu and Ma 2019; Zhang et al. 2022; Liu et al. 2020). As summarized by previous works (Chen 2020, 2023), perceiving techniques are highly varied and wide in terms of analysis targets, understanding granularities, practical applications, from biometric recognition (Sun et al. 2014; Guo et al. 2016; Liu et al. 2017; Deng et al. 2019; Zhong et al. 2021), human performance analysis (Ji et al. 2012; Caba Heilbron et al. 2015; Zhang et al. 2019; Qi et al. 2020a; Yun et al. 2024), to generic scene understanding (Grant and Flynn 2017; Qi et al. 2020b; Lv et al. 2024), *etc*. Therefore, how to schedule these domain-specific perceiving models and analyze the collected videos uniformly, efficiently, and especially intelligently is a main technique challenge.

Although the general-purpose visual models (Kirillov et al. 2023; Zhang et al. 2023) are compelling, they may not possess the domain-specific knowledge needed to replace certain perceiving models, especially the fine-grained ones like biometric recognition. Additionally, they are still too heavy to deal with a large volume of video data. Some recent works (Schick et al. 2023; Yao et al. 2023; Yang et al. 2023a; Wu et al. 2023; Hao et al. 2023; Kim et al. 2023; Qin et al. 2023a) discover the potential ability of the large language models (LLMs) to act as an intelligent agent to use tools, which motivates us to investigate the power of LLMs invoking and scheduling a variety of lightweight visual models to analyze the diverse surveillance videos.

One possible approach is to directly guide powerful LLMs to use the tool by offering concise explanations and demonstrations through in-context prompts, such as VISPROG (Gupta and Kembhavi 2023), ViperGPT (Surís, Menon, and Vondrick 2023) and Visual ChatGPT (Wu et al. 2023). However, these works must rely heavily on strong LLMs like ChatGPT. Alternatively, another approach is to fine-tune LLMs to become proficient in particular tools, such as Toolformer (Schick et al. 2023) and GPT4Tools (Yang et al. 2023a), requiring deep acquaintance with the application domains and tool usages.

Compared with previous tools like calculators, search engines, and usual vision models, *etc*, the visual algorithms for intelligent surveillance can be more "unusual" for LLMs to distinguish, plan, and execute. These algorithms fall under three primary categories: human-centric, vehicle-centric, and event-related. On one hand, LLMs necessitate distinguishing some fine-grained visual tools, *e.g.*, decide to invoke face recognition (Deng et al. 2019), person re-identification (Zheng, Zheng, and Yang 2018) or gait recog-

*Corresponding author: Huadong Ma.
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

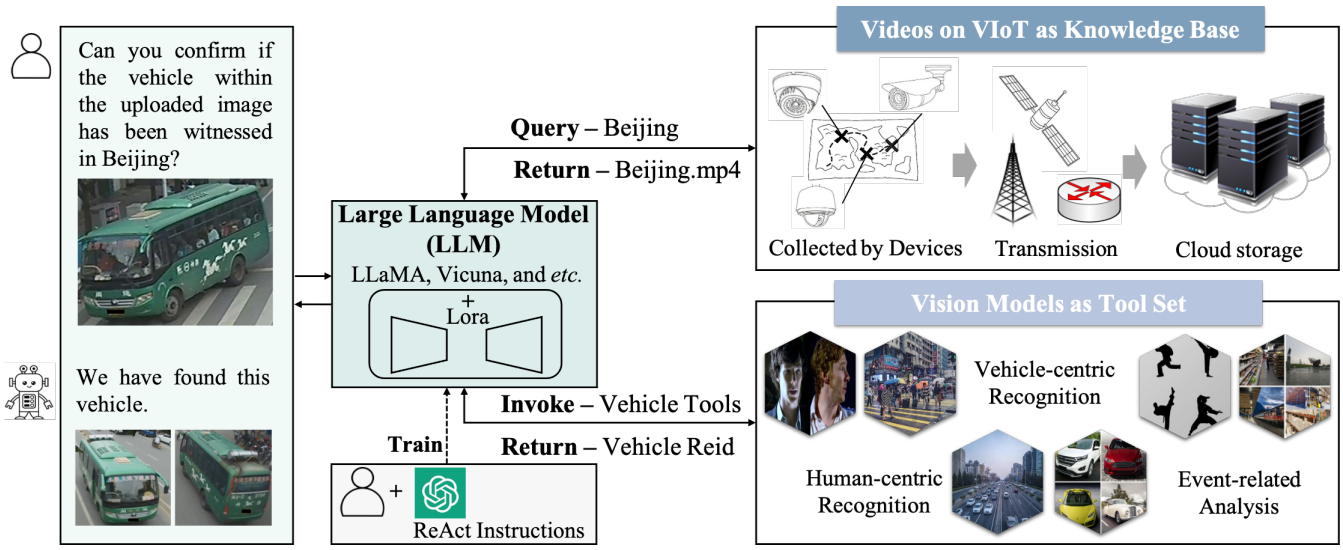

Figure 1: Illustration of VIoTGPT, which mainly consists of three fundamental modules, the videos that contain real-world observations, the human-centric algorithms acting as the tool set, and LLM as the intelligent agent.

nition (Fan et al. 2023) algorithm when they are asked to recognize the person in the video. On the other hand, it is required to invoke multiple interrelated algorithms successively and decide whether to execute the next visual algorithms, *e.g.*, whether to evaluate the impact of behavior after the action recognition (Zheng, Zheng, and Yang 2018).

To address the above challenges, we propose a hierarchical taxonomy of VIoT tools and meticulously crafted the VIoT-Tool dataset involving 11 types of representative vision tools across three primary categories. To lead the LLM to decide actions not only based on the initial human queries but also considering the observation from the environment, we use ReAct instructions while not only using in-context learning or chain-of-thought prompting. We resort to instruction tuning to guide the LLM to learn a variety of instructions to accomplish complicated tasks with a given request. Specifically, we design ReAct instructions based on these representative tools by semi-automatic annotations, and then supervised finetune LLMs to learn the fine-grained difference of similar tools and multi-step reasoning for interrelated tools.

The established framework, named VIoTGPT, is illustrated in Figure 1. VIoTGPT consists of three fundamental components, *i.e.*, LLM as the agent, videos as the knowledge base, and vision models as the tool set. The three fundamental components fulfill their duties and work seamlessly to deliver practical and effective results. To provide both quantitative and qualitative analysis, we establish the corresponding dataset VIoT-Tool based on diverse data including publicly available, web-collected, or self-made surveillance videos. Our major contributions can be summarized as follows:

- We propose VIoTGPT, the framework that applies the customized LLM as the intelligent agent, to interact with videos collected on VIoT, invoke visual models according to queries, and reply to human users. As we know, VIoTGPT is the first intelligent agent system to invoke visual models and analyze video data for intelligent video surveillance.
- To enable VIoTGPT, we propose a hierarchical taxonomy of VIoT, develop the training dataset involving 11 types of representative vision tools across three primary categories, and build the corresponding benchmarks to evaluate the performance of the intelligent agents. The dataset, named VIoT-Tool, will be publicly available to promote further research.
- Demonstrated by experimental results on VIoT-Tool, with instruction tuning, VIoTGPT can schedule the domain-specific perceiving models and analyze the collected videos intelligently to accomplish complicated tasks, especially can learn the fine-grained and interrelated tools scheduling ability.

Related Work

Foundation Model

The artificial intelligence landscape has been dominated by task-specific deep models (Liu et al. 2017; Caba Heilbron et al. 2015; Grant and Flynn 2017), while a new wave of foundation models aims to gain general-purpose vision representations (Chen and He 2021; Kirillov et al. 2023; Zhang et al. 2023), multi-modal representations (Wang et al. 2023; Girdhar et al. 2023; Li et al. 2023b), and even general-purpose generative models (Brooks et al. 2024).

The Segment Anything Model (SAM) (Kirillov et al. 2023), a pixel-level pre-training foundation model, has received widespread attention for its impressive ability in image segmentation. ImageBind (Girdhar et al. 2023) learns joint representation across six different modalities including image, text, audio, depth, thermal, and IMU data, and enables cross-modal retrieval, cross-modal generation, *etc*.

The amazing Sora model (Brooks et al. 2024), a text-conditional diffusion model based on transformer architecture, can generate high-fidelity video up to a minute long. It is trained on videos and images of varying durations, resolutions, and aspect ratios.

Nevertheless, current vision-centered foundation models still struggle to generalize across different vision tasks or different data domains (Tang, Xiao, and Li 2023; Ji et al. 2023; Zhou et al. 2023). The difficulties in multiple-granularity understanding, temporal analysis, and comprehensive data domains and modalities together impede the development of a unified model for all the tasks (Li et al. 2023a). We are eager to see the development of future unified models to replace the task-specific vision models of VIoT, then there is no need to invoke many fragmentary models. However, until then, we have to resort to other technical approaches for intelligence scheduling. We propose to construct a hierarchical taxonomy of the task-specific vision models and use the vision models as tools to collaboratively serve VIoT applications.

LLMs and Tool learning

Recently, large language models (LLMs) have received widespread attention because of their impressive performance on complex natural language understanding tasks (Zhao et al. 2023). LLMs are Transformer language models with billions of parameters trained on massive amounts of text data, which lead to some particularly interesting emergent behaviors including in-context learning (Brown et al. 2020), instruction following (Wei et al. 2022a), and step-by-step reasoning (Wei et al. 2022b). Notable LLMs contains OpenAI's GPT-series (Brown et al. 2020) used in ChatGPT, PaLM of Google (Chowdhery et al. 2022), LLaMa of Meta (Touvron et al. 2023a,b), Vicuna (Chiang et al. 2023), ChatGLM (Du et al. 2022), *etc*.

Despite the deficiencies such as limited corpora knowledge and unsatisfied numerical computation ability (Zhao et al. 2023), recent research has unveiled the potential of LLMs in mastering tools (Qin et al. 2023b; Schick et al. 2023; Shen et al. 2023; Gupta and Kembhavi 2023; Surís, Menon, and Vondrick 2023; Yang et al. 2023b; Wu et al. 2023; Yang et al. 2023a; Gao et al. 2024), enabling them to acquire domain-specific expertise and external knowledge. Without the need for explicit training, it is possible to accomplish tasks solely relying on the in-context learning ability of LLMs. VISPROG (Gupta and Kembhavi 2023) and ViperGPT (Surís, Menon, and Vondrick 2023) have demonstrated that, with a small number of *in-context examples instructions*, a powerful LLM GPT-3 can generate Python-like modular programs and execute them to invoke vision models and other functions for compositional visual tasks. ReAct (Yao et al. 2023) improved prompt engineering by introducing collaborative reasoning and execution, incorporating additional information into inference, and facilitating interaction between LLMs and the external environment. Based on ReAct and the powerful ChatGPT, MM-REACT (Yang et al. 2023b) and Visual ChatGPT (Wu et al. 2023) can integrate the system with vision models using *zero-shot prompting*, to accomplish visual understanding and generation tasks by invoking vision models and receiving the feedback iteratively until reaches the ending condition. CLOVA (Gao et al. 2024) incorporates both correct and incorrect examples for prompts to generate better plans and programs, and design the reflection and learning scheme to update tools. Without the strong LLMs like GPT-3 and ChatGPT, some recent works (Schick et al. 2023; Yang et al. 2023a; Qin et al. 2023b) explore the tool learning based on LLMs with about tens of billions of parameters(*e.g.*, GPT-J (Wang and Komatsuzaki 2021), LLaMA 7B (Touvron et al. 2023a) and Vicuna 13B (Chiang et al. 2023)), with self-instruction tuning (Wang et al. 2022), where instructions generated based on ChatGPT are used to fine-tune LLMs with a limited set of tools and have demonstrated promising results.

Considering the limited computing resources, we also investigate how to leverage LLMs with tens of billions parameters effectively. While different with (Schick et al. 2023; Yang et al. 2023a), we mainly focus on learning to schedule vision tools and query knowledge videos towards intelligent Video Internet of Things (VIoT).

VIoTGPT

Overview

VIoTGPT consists of three components that collaborate to ensure practicality, as shown in Figure 1.

Video Data as Knowledge Base. In practical VIoT applications, videos at various locations are collected by a variety of smart devices in real time. The massive scale videos contain real-world observations and can serve as the information-rich knowledge base denoted as $\mathcal{K} = \{K_1, K_2, \cdots, K_n\}$, which provides a solid foundation for VIoTGPT. For convenience, we use a knowledge base of videos with city-level locations, which can also be extended to more various kinds of tags, such as accurate spatio-temporal information, environmental information, *etc*.

Perceiving Models as Tool Set. We propose a hierarchical taxonomy of VIoT tools. The perceiving algorithms can be concluded as mainly three primary categories: human-centric, vehicle-centric, and event-related algorithms. Each primary category can conclude a variety of secondary categories. Without loss of generality, we define the secondary categories by 11 representative vision algorithms including face recognition, person re-identification, gait recognition, vehicle re-identification, license plate recognition, human pose estimation, human action recognition, crowd counting, scene recognition, fire and smoke detection, and violence detection, as the tool set denoted as $\mathcal{T} = \{T_1, T_2, \cdots, T_m\}$, to demonstrate the feasibility and functionality of VIoTGPT. The agent can invoke one or several tools intelligently to accomplish the tasks, which will be detailed further in Section .

LLM as Agent. The large language model (LLM) with model parameter θ is adopted as the intelligent agent to interact with humans, by firstly summarizing the input instructions and visual information to the pre-defined template, planning, observing, and reasoning with the assistance of the knowledge base and the tool set, and finally providing users with integrated processing information in its reply. Despite the powerful reasoning ability of LLMs such as

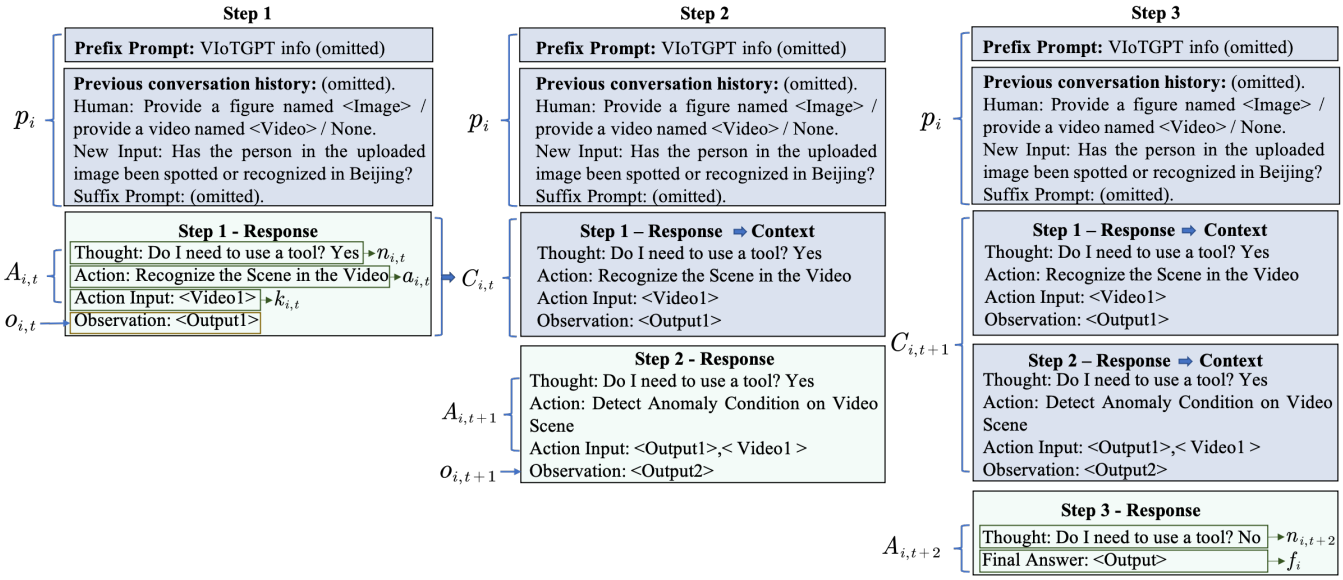

Figure 2: Pre-defined instructions and the response $A_{i,t}$ of the LLM θ at each step. In the intermediate steps, the LLM decides to use the tools ("Thought"), selected tool names ("Action"), and the input of tools ("Action Input"). At the end of each step, the context will be updated by combing previous actions and observations to conversation history $C_{i,t} = (C_{i,t-1}, A_{i,t}, o_{i,t})$. At the final step, the LLM decides not to use tools and returns the final feedback fi.

LLaMa (Touvron et al. 2023a,b) and GPT4Tools (Yang et al. 2023a), it is still difficult for them to be directly used by prompting in VIoTGPT. To help LLMs become more proficient in querying knowledge and invoking tools, we finetune them with the meticulously crafted VIoT-Tool dataset.

Instructions and Training

ReAct Instruction. Considering the intelligent agent should interact with humans, tools, video data, and the environment, we introduce ReAct instruction for VIoTGPT to determine the action in each step not only based on the context but also considering the observation from outputs of previously invoked tools.

Specifically, given the human query q_i with potential visual information v_i, the LLM will summarize and format them and the overall framework information ($\mathcal{T}$ and $\mathcal{K}$) into a new prompt $p_i = (\mathcal{T}, \mathcal{K}, q_i, v_i)$ with the predefined template following Langchain (Yao et al. 2023; Chase 2022), as shown in Figure 2. With the input prompt p_i, at each step, the LLM will determine the action $A_{i,t}$, record observations of tools $o_{i,t}$, memorize the context $C_{i,t} = (A_{i,1}, o_{i,1}, \cdots, A_{i,t}, o_{i,t})$ (including the history of actions and observations), and reason iteratively until it achieves the final answer f_i.

The action of the LLM (θ) at step t is represented by

$$A_{i,t} = \begin{cases} (n_{i,t}, a_{i,t}, k_{i,t}), & t \in \{0, 1, \cdots, T-1\} \\ (n_{i,t}, f_i), & t = T \end{cases} \quad (1)$$

and determined by $P_\theta(A_{i,t}) = P_\theta(A_{i,t}|p_i, C_{i,t-1})$, where $n_{i,t}$ denotes the *decision* of whether to use tools $\mathcal{T}$. Ideally, in the intermediate steps, $n_{i,t}$ represents that the LLM determines to use tools (see $n_{i,t}$ in Figure 2), $a_{i,t}$ represents the selected *tool names*, and $k_{i,t}$ represents the *input* information (usually queried from $\mathcal{K}$) of the selected tool $a_{i,t}$. At the end of each step, the context will be updated $C_{i,t} = (C_{i,t-1}, A_{i,t}, o_{i,t})$ to record the history of actions and observations. At the final step $t = T$, $n_{i,t}$ represents not to use tools, the LLM will return the final feedback fi.

Primary Categories	Secondary Categories	Human Input
Human-centric Recognition	Face Recognition	Image, Question
	Person Re-identification	Image, Question
	Gait Recognition	Video, Question
	Crowd Counting	Question
Vehicle-centric Recognition	Vehicle Re-identification	Image, Question
	License Plate Recognition	Question
Event-related Analysis	Fire Smoke Detection	Question
	Scene Recognition Anomaly Recognition	Question
	Pose Estimation Action Recognition	Question

Table 1: Summarization of VIoT-Tool.

To help the LLM θ perform in this way, we collect the ReAct instruction data $A_{i,t}$ and fine-tune the model. The ReAct instruction data is generated by ChatGPT and human annotations to keep the diversity of instructions and the correctness of tools and knowledge usages.

Supervised Fine-tuning. The learning process of the

Models	Promt	Single Tool Responses				Interrelated Tools Responses				All Responses			
		Decis	Tool	Input	Whole	Decis	Tool	Input	Whole	Decis	Tool	Input	Whole
Llama	Zero-shot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Llama	Few-shot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
VIoTGPT (Llama)	Zero-shot	**89.44**	**41.85**	**75.53**	**41.11**	**84.79**	**69.75**	**69.13**	**67.25**	**87.68**	**48.35**	**74.04**	**47.20**
Vicuna	Zero-shot	86.39	60.64	0.00	0.00	47.38	8.00	0.00	0.00	71.64	48.37	0.00	0.00
Vicuna	Few-shot	28.88	26.98	29.03	22.34	1.00	0.00	0.00	0.00	22.33	20.69	22.26	17.13
VIoTGPT (Vicuna)	Zero-shot	**99.01**	**63.83**	**76.06**	**35.79**	**86.11**	**74.38**	**69.75**	**52.00**	**94.13**	**66.29**	**74.59**	**39.59**
ChatGPT	Zero-shot	99.92	96.80	96.73	93.92	74.16	19.75	15.50	0.00	89.53	78.85	77.80	72.03
ChatGPT	Few-shot	99.85	99.85	99.85	99.85	98.57	95.38	61.63	28.25	99.24	98.81	90.94	83.16

Table 2: Quantitative results on VIoT-Tool. "Decis" represents the accuracy of decisions of whether to use tools $Acc_{n_{i,t}}$. "Tool" represents the accuracy of chosen tool names $Acc_{a_{i,t}}$. "Input" represents the accuracy of input information of tools $Acc_{k_{i,t}}$. "Whole" represents the accuracy of the whole response $Acc_{A_{i,t}}$. "IT" represents instruction tuning.

LLM θ can be formulated as:

$$L_{sft}(\theta) = -\sum_i \sum_t \log P_\theta \left(A_{i,t} | p_i, C_{i,t-1}\right). \quad (2)$$

With the supervised fine-tuning with instructions, LLM can invoke tools and query requisite knowledge of VIoT to perform targeted instructions, particularly in fine-grained tool usage and multi-step reasoning for interrelated tools.

Tools, Training Dataset, and Benchmarks

VIoT-Tool is established based on 11 represented vision tools, as shown in Table 1. Considering privacy and copyright, images and videos used in this paper are publicly available (Zheng et al. 2015), web-collected (Nagrani and Zisserman 2017), or self-made surveillance videos. For convenience, all the videos used in the knowledge base are represented as 125 city-level names.

Dataset Summarization

Statistics. Through meticulous annotation and optimization, we collect the training dataset with training instructions (2.79 billion tokens) constructed by 200K pairs (p_i and $A_{i,t}$) related to the 11 tools across three categories and the corresponding testing datasets with 1,841 pairs. To avoid data bias, the tool data distribution of the training and the test set are almost consistent, which is detailed in the Appendix.

Generalization and Robustness. To evaluate the generalization of LLMs on the knowledge videos, the video data in the testing benchmarks is different from that in the training dataset. To evaluate the semantic robustness, the instructions in the testing datasets are quite different from those in the training dataset.

Fine-grained and Interrelated Tools. In the hierarchical taxonomy of VIoT tools, tools of human-centric recognition or vehicle-centric recognition usually share similar objectives and usage, therefore leading to fine-grained tool differentiation. Event-related analysis typically involves multiple interrelated tools, making them useful as evaluation metrics.

Experiments

Experimental Setting

Models Details and Baselines. With limited computing resources, we use Llama 7B (Touvron et al. 2023a) and Vicuna 7B (Chiang et al. 2023) as base models for the following fine-tuning. Correspondingly, Llama 7B and Vicuna 7B without fine-tuning are used as baselines, which rely on the in-context ability with the same prompt. ChatGPT (gpt-3.5-turbo) is also used to compare our models and demonstrate an optimal performance on VIoT-Tool.

Training Details. To enable training, a parameter-efficient tuning method, *i.e.*, Low-Rank Adaptation (LoRA) (Hu et al. 2022), is used. Specifically, we attach the LoRA modules to the query and key of self-attention layers, with the rank parameter 8, the scaling alpha parameter 16, and the dropout rate 0.05, following the settings of FastChat (Zheng et al. 2023). The maximum length of new tokens is 2,048. We finetune LLMs using an effective batch size of 256 and a learning rate of 5e-5 for 6 epochs with the AdamW optimizer. In the training process, the instruction datasets are randomly divided into training and evaluating sets in a 49:1 proportion. All the experiments are conducted on NVIDIA RTX 4090 GPUs.

Evaluation Metrics. Four main evaluation metrics are used to evaluate the performance, including accuracy of decisions, accuracy of tool names, accuracy of tool inputs, and accuracy of whole response. Specifically, $n_{i,t}$ represents the decision of whether to use tools ("Thought: Do I need to use a tool? Yes/No"), and $Acc_{n_{i,t}}$ measures accuracy of $n_{i,t}$. Then the accuracy of decisions is defined as

$$Acc_{n_{i,t}} = \frac{1}{\sum_i \sum_t} \sum_i \sum_t \{n_{i,t} = \dot{n}_{i,t}\}, \quad (3)$$

where $\dot{n}_{i,t}$ represents the gold label of the decision whether to use tools at step t. Note that, $a_{i,t}$ represents the chosen tool name, and $Acc_{a_{i,t}}$ measures the accuracy of all the chosen tools. Then, the accuracy of tool names is

$$Acc_{a_{i,t}} = \frac{1}{\sum_i \sum_t} \sum_i \sum_t \{a_{i,t} = \dot{a}_{i,t}\}, \quad (4)$$

Figure 3: Illustration of VIoTGPT's capabilities and applications. "FaceRecognition", "PersonReidentification", "GaitRecognition", "LicensePlateRecognition", "VehicleReidentification", "CrowdCounting" and "FireSmokeDetection" represent responses with single tool. While "AnomalyDetection" and "ActionAnalysis" are two event-related pipelines that require scheduling interrelated tools.

where $\dot{a}_{i,t}$ represents the gold label of chosen tool at step t. In addition, $k_{i,t}$ represents the input of tool $a_{i,t}$ including the the queried knowledge information. $Acc_{k_{i,t}}$ measures the accuracy of the input information. Then the accuracy of tool inputs is

$$Acc_{k_{i,t}} = \frac{1}{\sum_i \sum_t} \sum_i \sum_t \{k_{i,t} = \dot{k}_{i,t}\}, \tag{5}$$

where $\dot{k}_{i,t}$ represents the gold label of the input at step t. Finally, $A_{i,t}$ represents the whole response of LLM, and $Acc_{A_{i,t}}$ measures its accuracy. The accuracy of the whole response is

$$Acc_{A_{i,t}} = \frac{1}{\sum_i} \sum_i (\prod_t \{A_{i,t} = \dot{A}_{i,t}\}), \tag{6}$$

where $A_{i,t}$ represents the gold label of response at step t.

Models	Validation Set				Test Set			
	Decis	Tool	Input	Whole	Decis	Tool	Input	Whole
Llama	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
VIoTGPT (Llama)	**84.05**	**61.96**	**81.60**	**60.13**	**89.44**	**41.85**	**75.53**	**41.11**
Vicuna	81.86	61.24	0.00	0.00	86.39	60.64	0.00	0.00
VIoTGPT (Vicuna)	**98.16**	**66.26**	**76.07**	**58.44**	**99.01**	**63.83**	**76.06**	**35.79**

Table 3: Comparison of the single tool performance on validation and test sets of VIoT-Tool respectively.

Models	Validation Set				Test Set			
	Decis	Tool	Input	Whole	Decis	Tool	Input	Whole
Llama	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
VIoTGPT (Llama)	**88.72**	**76.33**	**76.67**	**71.33**	**84.79**	**69.75**	**69.13**	**67.25**
Vicuna	45.01	10.25	0.00	0.00	47.38	8.00	0.00	0.00
VIoTGPT (Vicuna)	**90.08**	**77.67**	**80.33**	**66.00**	**86.11**	**74.38**	**69.75**	**52.00**

Table 4: Comparison of interrelated tools performance on validation and test sets of VIoT-Tool.

Models	Validation Set				Test Set			
	Decis	Tool	Input	Whole	Decis	Tool	Input	Whole
Llama	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
VIoTGPT (Llama)	**85.82**	**65.31**	**80.45**	**62.74**	**87.68**	**48.35**	**74.04**	**47.20**
Vicuna	67.93	49.35	0.00	0.00	71.64	48.37	0.00	0.00
VIoTGPT (Vicuna)	**95.11**	**68.92**	**77.06**	**60.20**	**94.13**	**66.29**	**74.59**	**39.59**

Table 5: Comparison of tools performance on validation and test sets of VIoT-Tool.

Experimental Results

Observations and Discussions

Baselines. Table 2 and Figure 3 report the main experimental results. We found that Llama 7B without fine-tuning could not follow the format requirements with the given prompt, *e.g.*, "Thought: Do I need to use a tool? Yes", let alone use the names of the tools correctly. Vicuna 7B without fine-tuning performs better than Llama since it can make correct decisions using "Thought: Do I need to use a tool? Yes" and even choose appropriate tools, but still cannot correctly query videos in the datasets and failed to accomplish the whole process. We also apply the Vicuna 13B model trained on GPT4Tools (Yang et al. 2023a) as the agent for VIoT. However, it has been observed that this model yields no significant gains compared to the original Vicuna model. We speculate this is due to the different trained tools and the video querying requirements of VIoT.

VIoTGPT. VIoTGPT models have achieved significant performance improvement, as shown in Table 2. For "Llama+IT", we observed that the accuracy of whole response $Acc_{A_{i,t}}$ of single tool responses is mainly limited by the accuracy of tool names $a_{i,t}$, especially some fine-grained differences, *e.g.*, distinguishing license plate recognition and vehicle re-identification, distinguishing gait recognition and person re-identification. This limitation also hinders the performance of "Vicuna+IT". For "Vicuna+IT", another interesting observation is that, although it has invoked tools and querying video knowledge correctly, it failed to return the final feedback (just giving nothing back) to finish the whole process $A_{i,t}$.

ChatGPT. With powerful reasoning ability and in-context prompts, ChatGPT showed strong performance on the individual tool responses in the VIoT-Tool dataset, but it still exhibited significant errors with the interrelated tools, such as invoking tools repeatedly, using the wrong tools and the wrong input, and failing to give final responses due to hallucination. ChatGPT (few-shot) can achieve a nearly satisfactory performance on single-tool responses. However, challenges remain in interrelated tools. Despite this, the performance still demonstrated the reliability of the VIoT-Tool dataset and showed the excellent reasoning ability of ChatGPT, which is what we pursue in the closed system with small-size agents without external APIs.

Additional Experimental Results. We present a comparison of the performance of the validation set and the test set. The experimental results are listed in Table 3, Table 4, and Table 5. We found that there is still scope for improvement on the validation set. The experimental evidence suggests that the implementation of more powerful LLMs is imperative to achieve better results. In addition, there exists a significant performance gap between the validation set and the test set. It is evident that the performance gap is primarily due to the semantic differences in human queries or the diversity of queried knowledge videos. By identifying and addressing these challenges in the future, VIoTGPT can achieve better performance.

Conclusion

In this paper, we presented VIoTGPT, a framework that trains LLMs as intelligent agents to interact with humans, query knowledge videos, and activate vision models to complete complex tasks for intelligent VIoT. We carefully constructed VIoT-Tool dataset to support VIoTGPT and future research. Based on these, we observed promising results. We believe scheduling perceiving models and analyzing videos intelligently will lead VIoT into a bright and smart future.

Acknowledgments

This work is partially supported by the Funds for the NSFC Project under Grants U24B20176, 62402051, and 62202063, Beijing Natural Science Foundation under Grants L223002, and also the China National Postdoctoral Program for Innovative Talents BX20230053.

References

Brooks, T.; Peebles, B.; Holmes, C.; DePue, W.; Guo, Y.; Jing, L.; Schnurr, D.; Taylor, J.; Luhman, T.; Luhman, E.; Ng, C.; Wang, R.; and Ramesh, A. 2024. Video generation models as world simulators.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *NeurIPS*.

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*.

Chase, H. 2022. LangChain, October 2022. *URL https://github. com/hwchase17/langchain*.

Chen, C. W. 2020. Internet of video things: Next-generation IoT with visual sensors. *IEEE IoTJ*, 7(8).

Chen, C.-W. C. 2023. Internet of Video Things: Technical Challenges and Emerging Applications. In *ACMMM*.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *https://vicuna. lmsys. org*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2022. Palm: Scaling language modeling with pathways. *arXiv:2204.02311*.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; and Tang, J. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *ACL*.

Fan, C.; Liang, J.; Shen, C.; Hou, S.; Huang, Y.; and Yu, S. 2023. OpenGait: Revisiting Gait Recognition Towards Better Practicality. In *CVPR*.

Gao, Z.; Du, Y.; Zhang, X.; Ma, X.; Han, W.; Zhu, S.-C.; and Li, Q. 2024. CLOVA: A closed-loop visual assistant with tool usage and update. In *CVPR*.

Girdhar, R.; El-Nouby, A.; Liu, Z.; Singh, M.; Alwala, K. V.; Joulin, A.; and Misra, I. 2023. Imagebind: One embedding space to bind them all. In *CVPR*.

Grant, J. M.; and Flynn, P. J. 2017. Crowd Scene Understanding from Video: A Survey. *TOMM*, 13(2): 19:1–19:23.

Guo, Y.; Zhang, L.; Hu, Y.; He, X.; and Gao, J. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*.

Gupta, T.; and Kembhavi, A. 2023. Visual programming: Compositional visual reasoning without training. In *CVPR*.

Hao, S.; Liu, T.; Wang, Z.; and Hu, Z. 2023. ToolkenGPT: Augmenting Frozen Language Models with Massive Tools via Tool Embeddings. *arXiv:2305.11554*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2012. 3D convolutional neural networks for human action recognition. *IEEE TPAMI*, 35(1): 221–231.

Ji, W.; Li, J.; Bi, Q.; Li, W.; and Cheng, L. 2023. Segment anything is not always perfect: An investigation of sam on different real-world applications. *arXiv:2304.05750*.

Kim, C.; Seo, Y.; Liu, H.; Lee, L.; Shin, J.; Lee, H.; and Lee, K. 2023. Guide Your Agent with Adaptive Multimodal Rewards. In *NeurIPS*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. *arXiv:2304.02643*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Li, C.; Gan, Z.; Yang, Z.; Yang, J.; Li, L.; Wang, L.; and Gao, J. 2023a. Multimodal foundation models: From specialists to general-purpose assistants. *arXiv:2309.10020*.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv:2301.12597*.

Liu, K.; and Ma, H. 2019. Exploring background-bias for anomaly detection in surveillance videos. In *ACMMM*.

Liu, K.; Zhu, M.; Fu, H.; Ma, H.; and Chua, T.-S. 2020. Enhancing anomaly detection in surveillance videos with transfer learning from action recognition. In *ACMMM*.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*.

Liu, X.; Liu, W.; Ma, H.; and Fu, H. 2016. Large-scale vehicle re-identification in urban surveillance videos. In *ICME*.

Lv, C.; Zhang, S.; Tian, Y.; Qi, M.; and Ma, H. 2024. Disentangled counterfactual learning for physical audiovisual commonsense reasoning. *NeurIPS*.

Ma, H.-D. 2011. Internet of things: Objectives and scientific challenges. *Journal of Computer science and Technology*, 26(6): 919–924.

Mohan, A.; Gauen, K.; Lu, Y.-H.; Li, W. W.; and Chen, X. 2017. Internet of video things in 2030: A world with many cameras. In *ISCAS*.

Nagrani, A.; and Zisserman, A. 2017. From Benedict Cumberbatch to Sherlock Holmes: Character Identification in TV series without a Script. In *BMVC*.

Qi, M.; Qin, J.; Zhen, X.; Huang, D.; Yang, Y.; and Luo, J. 2020a. Few-shot ensemble learning for video classification with slowfast memory networks. In *ACMMM*.

Qi, M.; Wang, Y.; Li, A.; and Luo, J. 2020b. STC-GAN: Spatio-temporally coupled generative adversarial networks for predictive scene parsing. *TIP*, 29: 5420–5430.

Qin, Y.; Hu, S.; Lin, Y.; Chen, W.; Ding, N.; Cui, G.; Zeng, Z.; Huang, Y.; Xiao, C.; Han, C.; et al. 2023a. Tool learning with foundation models. *arXiv:2304.08354*.

Qin, Y.; Hu, S.; Lin, Y.; Chen, W.; Ding, N.; Cui, G.; Zeng, Z.; Huang, Y.; Xiao, C.; Han, C.; et al. 2023b. Tool learning with foundation models. *arXiv:2304.08354*.

Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv:2302.04761*.

Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; and Zhuang, Y. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv:2303.17580*.

Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification. In *NIPS*.

Surís, D.; Menon, S.; and Vondrick, C. 2023. Vipergpt: Visual inference via python execution for reasoning. *arXiv:2303.08128*.

Tang, L.; Xiao, H.; and Li, B. 2023. Can sam segment anything? when sam meets camouflaged object detection. *arXiv:2304.04709*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; and et al. 2023a. Llama: Open and efficient foundation language models. *arXiv:2302.13971*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; and et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*.

Wang, B.; and Komatsuzaki, A. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Wang, W.; Bao, H.; Dong, L.; Bjorck, J.; Peng, Z.; Liu, Q.; Aggarwal, K.; Mohammed, O. K.; Singhal, S.; Som, S.; and Wei, F. 2023. Image as a foreign language: BEiT pretraining for vision and vision-language tasks. In *CVPR*.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv:2212.10560*.

Wei, J.; Bosma, M.; Zhao, V.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022a. Finetuned Language Models are Zero-Shot Learners. In *ICLR*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*.

Wei, W.; Wu, H.; and Ma, H. 2019. An autoencoder and LSTM-based traffic flow prediction method. *Sensors*, 19(13): 2946.

Wu, C.; Yin, S.; Qi, W.; Wang, X.; Tang, Z.; and Duan, N. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv:2303.04671*.

Yang, R.; Song, L.; Li, Y.; Zhao, S.; Ge, Y.; Li, X.; and Shan, Y. 2023a. Gpt4tools: Teaching large language model to use tools via self-instruction. *arXiv:2305.18752*.

Yang, Z.; Li, L.; Wang, J.; Lin, K.; Azarnasab, E.; Ahmed, F.; Liu, Z.; Liu, C.; Zeng, M.; and Wang, L. 2023b. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv:2303.11381*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. React: Synergizing reasoning and acting in language models. *ICLR*.

Yun, W.; Qi, M.; Wang, C.; and Ma, H. 2024. Weakly-Supervised Temporal Action Localization by Inferring Salient Snippet-Feature. In *AAAI*.

Zantalis, F.; Koulouras, G.; Karabetsos, S.; and Kandris, D. 2019. A review of machine learning and IoT in smart transportation. *Future Internet*, 11(4): 94.

Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J.; and Zheng, N. 2019. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE TPAMI*, 41(8): 1963–1978.

Zhang, S.; Gong, M.; Xie, Y.; Qin, A. K.; Li, H.; Gao, Y.; and Ong, Y.-S. 2022. Influence-aware attention networks for anomaly detection in surveillance videos. *IEEE TCSVT*, 32(8): 5427–5437.

Zhang, Y.; Huang, X.; Ma, J.; Li, Z.; Luo, Z.; Xie, Y.; Qin, Y.; Luo, T.; Li, Y.; Liu, S.; et al. 2023. Recognize Anything: A Strong Image Tagging Model. *arXiv:2306.03514*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv:2303.18223*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv:2306.05685*.

Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q. 2015. Scalable person re-identification: A benchmark. In *ICCV*.

Zheng, Z.; Zheng, L.; and Yang, Y. 2018. Pedestrian alignment network for large-scale person re-identification. *IEEE TCSVT*, 29(10): 3037–3045.

Zhong, Y.; Deng, W.; Hu, J.; Zhao, D.; Li, X.; and Wen, D. 2021. SFace: sigmoid-constrained hypersphere loss for robust face recognition. *IEEE TIP*, 30: 2587–2598.

Zhou, T.; Zhang, Y.; Zhou, Y.; Wu, Y.; and Gong, C. 2023. Can sam segment polyps? *arXiv:2304.07583*.