Constructing and Interpreting Digital Twin Representations for Visual Reasoning via Large Language Models and Reinforcement Learning

Yiqing Shen[1(✉)][0000−0001−7866−3339] and Mathias Unberath[1]

[1]Johns Hopkins University, Baltimore, MD, USA
{yshen92,unberath}@jhu.edu

Abstract. Visual reasoning may require models to interpret images/videos and respond to implicit text queries across diverse output formats, from pixel-level segmentation masks to natural language descriptions. Existing approaches rely on supervised fine-tuning with task-specific architectures. For example, reasoning segmentation, grounding, summarization, and visual question answering each demand distinct model designs and training, preventing unified solutions and limiting cross-task/-modality generalization. Hence, we propose DT-R1, a reinforcement learning framework that trains large language models (LLMs) to construct digital twin (DT) representations of complex multi-modal visual inputs and then reason over these high-level representations as a unified approach to visual reasoning. Specifically, we train DT-R1 using GRPO with a novel reward that validates both structural integrity and output accuracy. Evaluations in six visual reasoning benchmarks, covering two modalities and four task types, demonstrate that DT-R1 consistently achieves improvements over state-of-the-art task-specific models. DT-R1 opens a new direction where visual reasoning emerges from reinforcement learning on with DT representations.

1 Introduction

Visual reasoning supports embodied AI systems to interpret visual data (*e.g.*, images or videos) and respond to implicit text queries in various applications, such as robotics and autonomous driving [7]. Previous work on visual reasoning has primarily focused on reasoning segmentation [14,39], where targeted objects are segmented based on implicit text queries like "*segment the object used to hold hot beverages*" rather than explicit expressions [12] such as "*segment the coffee cup*" or point prompts [13,24]. However, real-world applications may require diverse output formats beyond pixel-level segmentation masks [29,30]. For instance, assistive systems for visually impaired users need textual descriptions of scenes rather than masks or bounding box; while it is more efficient to use bounding box in grounding. Reasoning visual tasks (RVTs) [28] then expand on reasoning segmentation to a diverse family of visual reasoning problems, including reasoning grounding [50] for bounding box generation, reasoning summarization for

natural language descriptions, reasoning visual question answering (VQA) [45] for textual responses, *etc.*

Yet, current approaches for RVTs face three limitations. First, RVT methods such as LISA [14], VISA [39], and ScanReason [50] encode visual features as tokens for multimodal LLM processing. This token-based encoding breaks apart continuous spatial-temporal relationships into discrete tokens, causing the loss of fine-grained geometric information necessary for reasoning, resulting in suboptimal performance [27,31]. Second, most existing RVT methods rely on supervised fine-tuning (SFT), which produces output without explicit reasoning chains [35]. This prevents decision verification from the users' side, and limits their ability to generalize beyond training distributions as demonstrated by poor performance on out-of-domain data [18, 29]. Finally, existing RVT methods remain specific to task types or data modalities, where each RVT requires its own specialized model architecture and training procedure. These limitations motivate the need for a unified framework that preserves spatial-temporal information while learning reasoning capabilities without task-specific architectural modifications.

Recent work on just-in-time digital twins (DT) [31] shows that replacing token embedding with DT representations can preserve spatial-temporal continuity for reasoning segmentation in both image and video [29], but it is performed in a zero-shot manner. Consequently, to bridge the gap, we propose DT-R1, a reinforcement learning (RL) framework that trains LLMs to construct and reason over task-specific DT representations, allowing diverse RVTs across image and video to be handled within a single unified model through structured rollout sequences. Specifically, DT-R1 trains LLMs to analyze implicit text queries and autonomously generate plans that specify which vision foundation models to apply for DT representation construction, thereby eliminating the need for manual effort in domain-specific design processes. Subsequently, the LLM learns to perform iterative reasoning over these DT representations by generating explicit CoTs [35] and executable Python code for qualitative reasoning when required, before identifying the appropriate task type and producing the corresponding output format. Rather than relying on SFT with annotated reasoning chain, DT-R1 gains reasoning capabilities in diverse domains without requiring task-specific tuning through pure RL through Group-Relative Policy Optimization (GRPO) [26], together with a novel reward that combines format validation for structured outputs and accuracy validation for the final output.

The major contributions are three-fold. First, we propose DT-R1, an RL framework that trains a single LLM to handle diverse RVTs including reasoning segmentation, grounding, summarization, and visual question answering across two visual modalities, as a unified solution to eliminate the need for task-specific architecture design and tuning. Specifically, it designs a structured rollout that trains the LLM to generate DT representation construction plans specifying vision foundation model dependencies, explicit reasoning chains, executable Python code for spatial-temporal computations, task-type identification, and task-appropriate final responses. Second, we introduce a novel rule-based reward for DT-R1 training that takes into account the output format and ac-

curacy, including the validity of the DT representation construction plan and the executability of the code. Third, we demonstrate that pure RL enables the development of reasoning capabilities without requiring explicit supervision on visual data with the self-constructed DT representations, which suggests a new potential venue for scaling.

2 Related Works

Reasoning Visual Tasks (RVTs) Reasoning segmentation aims to identify the target object and generates the corresponding pixel-level masks by interpreting implicit text queries through multi-step reasoning [30]. Reasoning segmentation methods can be classified into two primary paradigms. End-to-end methods, such as LISA [14] and VISA [39], employ specialized tokens such as `<seg>` to integrate the perception and reasoning capabilities of multimodal LLMs (MLLMs) with the segmentation abilities of separate decoders, but may encounter catastrophic forgetting during fine-tuning. Disentangled architectures, such as LLM-Seg [34], CoReS [3], LLaVASeg [42] and Seg-Zero [18], address these limitations by maintaining the separation between reasoning and segmentation by transforming the reasoning output into explicit prompts for segmentation networks [13, 24] at the cost of additional computational complexity. Just-in-time DT representation has further disentangled perception from reasoning [31] by constructing structured DT representation that preserve fine-grained semantic, spatial, and temporal information, while enabling LLMs to generate executable code for complex reasoning operations in a zero-shot manner, eliminating the need for LLM fine-tuning. Since reasoning segmentation remains limited to mask generation, reasoning visual tasks (RVTs) [28] are introduced as a unified formulation. Specifically, it extends the reasoning paradigm beyond segmentation to encompass multiple output formats, including bounding box generation, natural language summarization, and visual question answering, while preserving the concept of processing implicit text queries.

Digital Twin Representation Digital twins (DTs) in the context of foundation models have been defined as "*outcome-driven virtual replicas of physical processes that capture task-specific entities and their interactions to analyze and optimize their real-world counterparts*" [27]. DT representations are the building blocks of these virtual DT replicas [27], usually comprising structured data formats that encode semantic, spatial, and temporal relationships extracted from raw input. In terms of RVTs, previous work has shown the success of DT representation as an intermediate layer between visual perception and reasoning [31], where LLMs dynamically plan the construction of such representations from video using specialized vision foundation models like SAM2 [24], DepthAnything2 [40], OWLv2 [20] and DINOv2 [21], based on query.

Reinforcement Learning for LLMs RL is a training paradigm for LLMs to gain reasoning capabilities, as shown by DeepSeek-R1 [8] with GRPO. Similarly,

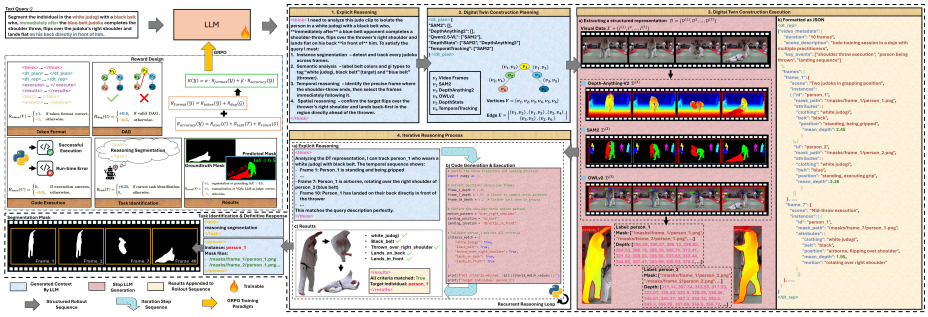

Fig. 1: Overall framework of DT-R1 for unified reasoning visual tasks. Given an implicit text query and visual input (image or video), DT-R1 generates a structured rollout sequences through reinforcement learning.

ToRL [15] and Search-R1 [11] have shown that RL training can further improve mathematical reasoning or LLM search-based reasoning by iterative reward optimization without requiring explicit supervision. Seg-Zero [18] and Seg-R1 [43] applied pure RL to visual reasoning, achieving superior out-of-domain generalization in terms of reasoning segmentation. ViGoRL achieves spatially grounded visual reasoning via RL by anchoring each reasoning step to specific visual coordinates [25]. However, existing RL-based approaches remain constrained to text-only reasoning domains or individual reasoning visual tasks, whereas our DT-R1 establishes an integration of RL training with DT representations to enable unified reasoning across diverse RVTs.

3 Methods

Oveview We propose DT-R1, an RL framework that trains LLMs to construct and reason over DT representations as a unified method for reasoning visual tasks [28] in the context of both image and video modalities, as shown in Fig. 1. Rather than relying on token representation of visual features [17] or supervised fine-tuning with explicit reasoning data, DT-R1 learns to bridge the gap between high-level text query reasoning and low-level visual perception through DT representations [29, 31].

Structured Rollout Sequence DT-R1 trains LLM to plan, construct, and reason on DT representations from the given implicit text query Q via structured output. Specifically, it begins the generation of the output sequence by explicit reasoning within `<think>` and `</think>` tokens, where the LLM analyzes the requirements of the given text query Q and identifies the types of information needed. Afterwards, the LLM then generates the DT representation construction plan within specialized tokens a to specify which vision foundation models should be applied and establish their execution dependencies in terms of a directed acyclic graph (DAG) $\mathcal{G}$. The DT representation construction plan is then

executed by applying the specified vision foundation models to the given visual data $\mathcal{X} = \{I^{(1)}, I^{(2)}, \ldots, I^{(T)}\}$, where T represents the number of frames and $T = 1$ for the image modality. The planned DT construction is then executed, applying the specified vision foundation models to extract relevant visual information from the input visual data. The resulting structured representation $\mathcal{D} = \{D^{(1)}, D^{(2)}, \ldots, D^{(T)}\}$ is formatted as JSON and integrated into the rollout sequence within <dt_rep> and </dt_rep> tokens. Upon completion of the DT construction, DT-R1 trains LLM to enter an iterative reasoning process to address the reasoning visual task. This begins with another round of explicit reasoning within <think> and </think> tokens, where the LLM reasons over the self-constructed DT representation. When computational operations are required in spatial or temporal reasoning, LLM generates executable Python code within <execute> and </execute> tokens, using libraries such as OpenCV. The execution results are also appended to the rollout sequence within the <results> and </results> tokens. This iterative reasoning process continues until the termination conditions are met: either the maximum token limit is reached, or the model generates a final response by determining the appropriate task type and output format within <task> and </task> tokens, specifying whether the response should contain segmentation masks, bounding boxes, natural language summaries, or textual answers. Finally, the definitive response is then generated within <answer> and </answer> tokens. The complete structured output sequence can be formulated as:

$$\begin{aligned}\mathcal{Y} = \langle &\texttt{<think>}\mathcal{R}_0\texttt{</think>}, \texttt{<dt\_plan>}\mathcal{G}\texttt{</dt\_plan>}, \\ &\texttt{<dt\_rep>}\mathcal{D}\texttt{</dt\_rep>}, (\texttt{<think>}\mathcal{R}_i\texttt{</think>}, \\ &\texttt{<execute>}\mathcal{C}_i\texttt{</execute>}, \ \texttt{<results>}\mathcal{O}_i\texttt{</results>})_{i=1}^{m}, \\ &\texttt{<task>}\mathcal{T}\texttt{</task>}, \texttt{<answer>}\mathcal{S}\texttt{</answer>}\rangle\end{aligned}$$

where $\mathcal{R}_0$ represents the initial analysis, $\mathcal{G}$ denotes the construction plan in terms of DAG, $\mathcal{D}$ is the DT representation, $\mathcal{R}_i$, $\mathcal{C}_i$, and $\mathcal{O}_i$ represent the results of reasoning, code, and execution in iteration i, $\mathcal{T}$ specifies the type of task, $\mathcal{S}$ contains the final answer, and m indicates the number of iterative reasoning cycles performed.

Digital Twin Representation Planning and Construction This plan for DT representation construction is formulated by a text-encoded DAG [32] $\mathcal{G} = (V, E)$ in tokens <dt_plan> and </dt_plan>, which defines the execution sequence and dependencies for the required vision foundation models. The graph structure comprises vertices $V = \{v_1, v_2, \ldots, v_K\}$ representing the selected vision foundation models and edges $E = \{(v_i, v_j) | v_i, v_j \in V\}$ denoting directed dependencies between models. Each edge $(v_i, v_j) \in E$ establishes that the computational output of model v_i serves as a pre-requisite for model v_j. During both training and inference, the LLM acquires knowledge of available vision foundation models through prompts that detail each model's specific capabilities, input format requirements, and output specifications.

For example, SAM2 [13, 24] typically functions as a primary computational node for instance segmentation, generating mask collections $\mathcal{M}^{(t)} = \{m_i^{(t)}\}_{i=1}^{N^{(t)}}$

where each $m_i^{(t)}$ represents a binary segmentation mask for the instance of the object i in the temporal frame t, with $N^{(t)}$ indicating the total number of detected instances. To enable processing by LLM, only the path of the mask collections is encoded in the DT representation instead of the mask itself. DepthAnything2 [40] operates as a dependent computational node that processes video frame input to generate dense depth maps $\mathcal{Z}^{(t)}$, while subsequent depth statistics computation operations depend on both the SAM2-generated masks and DepthAnything2 depth output for spatial analysis. Following previous work [28, 31], VLMs such as Qwen2.5-VL [2] function as semantic analysis nodes that maintain partial dependencies on SAM2 outputs to facilitate multi-level semantic information extraction across instance, frame, and video levels. In this case, the DAG is represented in text format as a structured dependency specification: *{"SAM2": [], "DepthAnything2": [], "DepthStats": ["SAM2", "DepthAnything2"], "SemanticAnalysis": ["SAM2"]}*, where each key is a vision foundation model (*i.e.*, vertex in the DAG) and the corresponding list specifies its prerequisite dependencies (*i.e.*, the edges).

When `</dt_plan>` is detected in the rollout sequence, the text generation is paused for DT representation generation. Then, the resulting DT representation $\mathcal{D}$ is inserted between the `<dt_rep>` and `</dt_rep>` tokens as a JSON structure that encompasses three levels. The video-level metadata captures global scene descriptions and temporal information throughout the video sequence. Frame-level information provides scene and spatial descriptions for each timestamp t. Instance-level attributes contain detailed masks, depth statistics, semantic descriptions, and visual features for individual objects. For example, for each instance i, depth statistics $d_i^{(t)} = \{\mathcal{Z}^{(t)}(p) | p \in m_i^{(t)}\}$ are computed on all pixels within the instance mask, incorporating mean depth $\mu_i^{(t)} = \frac{1}{|m_i^{(t)}|} \sum_{p \in m_i^{(t)}} \mathcal{Z}^{(t)}(p)$ and standard deviation. After `</dt_rep>`, LLM continues to generate the subsequent sequence.

Reasoning on Digital Twin Representation Following DT representation construction, LLM performs iterative reasoning using $\mathcal{D}$ as the perception of $\mathcal{X}$. Each reasoning iteration $\mathcal{R}_i$ within tokens `<think>` and `</think>` examines specific components of $\mathcal{D}$ to extract relevant information to resolve the reasoning visual task described in the implicit text query, performing semantic reasoning related to object functions, analyzing attribute relationships, and interpreting the context of the scene. When spatial reasoning (determining geometric relationships, relative positions, and distance calculations between objects) or temporal reasoning (analyzing motion patterns, tracking object state changes, and understanding sequential events across frames) is required, the LLM generates executable Python code $\mathcal{C}_i$ within `<execute>` and `</execute>` tokens. After detection of `</execute>`, LLM generates pauses and the generated code code is then operated directly on the DT representation, performing operations such as distance calculations using depth statistics $d_i^{(t)}$ and mean depth values $\mu_i^{(t)}$, geometric analysis on segmentation masks $m_i^{(t)}$, or temporal correlation analysis across frame sequences. The results of the execution $\mathcal{O}_i$ are appended between

Table 1: Performance comparison of video reasoning segmentation on JiTBench [31] using region similarity ($\mathcal{J}$) and contour accuracy ($\mathcal{F}$) for semantic, spatial, and temporal reasoning at three difficulty levels.

Methods	Level 1 (Basic)								Level 2 (Intermediate)								Level 3 (Advanced)							
	Semantic		Spatial		Temporal		Avg.		Semantic		Spatial		Temporal		Avg.		Semantic		Spatial		Temporal		Avg.	
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
LISA-7B [14]	0.635	0.226	0.398	0.706	0.283	0.451	0.420	0.480	0.442	0.213	0.198	0.490	0.268	0.273	0.284	0.344	0.274	0.229	0.229	0.322	0.282	0.307	0.244	0.304
LISA-13B [14]	0.669	0.258	0.237	0.756	0.313	0.320	0.388	0.463	0.472	0.230	0.176	0.524	0.283	0.256	0.293	0.354	0.301	0.234	0.177	0.353	0.280	0.259	0.237	0.297
GSVA [37]	0.587	0.431	0.218	0.541	0.324	0.214	0.412	0.360	0.534	0.353	0.202	0.487	0.237	0.115	0.363	0.280	0.502	0.289	0.134	0.480	0.215	0.108	0.308	0.268
LLM-Seg [34]	0.423	0.315	0.184	0.535	0.345	0.278	0.307	0.386	0.210	0.201	0.120	0.437	0.258	0.247	0.177	0.314	0.187	0.154	0.119	0.319	0.218	0.218	0.153	0.252
V* [36]	0.141	0.071	0.104	0.123	0.055	0.082	0.105	0.087	0.170	0.090	0.060	0.153	0.084	0.044	0.107	0.094	0.118	0.095	0.033	0.109	0.072	0.026	0.082	0.069
Seg-Zero [18]	0.598	0.445	0.361	0.622	0.478	0.334	0.468	0.453	0.524	0.398	0.241	0.548	0.432	0.309	0.388	0.430	0.471	0.352	0.223	0.523	0.421	0.281	0.349	0.408
VISA-7B [39]	0.563	0.521	0.354	0.585	0.563	0.327	0.479	0.492	0.487	0.473	0.235	0.514	0.510	0.303	0.398	0.442	0.432	0.411	0.218	0.497	0.499	0.277	0.354	0.424
VISA-13B [39]	0.599	0.532	0.389	0.621	0.587	0.381	0.503	0.526	0.521	0.505	0.267	0.547	0.543	0.334	0.431	0.475	0.463	0.441	0.248	0.528	0.531	0.306	0.384	0.455
JiT [31]	0.865	0.789	0.721	0.795	0.831	0.793	0.792	0.806	0.841	0.752	0.705	0.801	0.819	0.784	0.766	0.801	0.810	0.741	0.690	0.801	0.792	0.737	0.747	0.777
DT-R1	**0.917**	**0.836**	**0.764**	**0.843**	**0.883**	**0.848**	**0.837**	**0.850**	**0.887**	**0.793**	**0.742**	**0.845**	**0.864**	**0.827**	**0.808**	**0.845**	**0.851**	**0.778**	**0.725**	**0.841**	**0.832**	**0.774**	**0.785**	**0.816**

Table 2: Performance comparison of video reasoning segmentation methods on RVT-Bench [28] segmentation part.

Methods	Level 1 (Basic)								Level 2 (Intermediate)								Level 3 (Advanced)								Level 4 (Expert)							
	Semantic		Spatial		Temporal		Avg.		Semantic		Spatial		Temporal		Avg.		Semantic		Spatial		Temporal		Avg.		Semantic		Spatial		Temporal		Avg.	
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
LISA-7B [14]	0.458	0.432	0.416	0.422	0.407	0.394	0.435	0.408	0.449	0.421	0.390	0.412	0.396	0.357	0.420	0.388	0.421	0.406	0.366	0.394	0.381	0.332	0.398	0.369	0.404	0.383	0.343	0.366	0.355	0.310	0.377	0.343
LISA-13B [14]	0.485	0.465	0.448	0.459	0.447	0.425	0.466	0.444	0.478	0.457	0.421	0.452	0.433	0.394	0.452	0.426	0.458	0.444	0.391	0.431	0.418	0.367	0.431	0.405	0.439	0.426	0.368	0.413	0.401	0.346	0.411	0.387
GSVA [37]	0.098	0.095	0.209	0.069	0.065	0.158	0.134	0.097	0.100	0.093	0.133	0.070	0.064	0.094	0.108	0.076	0.101	0.098	0.129	0.072	0.069	0.090	0.109	0.077	0.095	0.094	0.107	0.064	0.063	0.067	0.099	0.065
LLM-Seg [34]	0.095	0.096	0.200	0.130	0.129	0.244	0.130	0.168	0.095	0.092	0.129	0.132	0.129	0.168	0.105	0.143	0.096	0.094	0.117	0.132	0.130	0.157	0.103	0.140	0.094	0.091	0.106	0.129	0.127	0.145	0.097	0.134
V* [36]	0.041	0.039	0.085	0.030	0.027	0.068	0.055	0.041	0.042	0.038	0.057	0.030	0.026	0.041	0.046	0.032	0.044	0.042	0.054	0.031	0.029	0.039	0.046	0.033	0.040	0.038	0.043	0.027	0.025	0.029	0.040	0.027
Seg-Zero [18]	0.479	0.446	0.413	0.461	0.437	0.399	0.446	0.422	0.459	0.424	0.387	0.443	0.416	0.372	0.423	0.410	0.438	0.408	0.365	0.422	0.398	0.349	0.404	0.390	0.413	0.389	0.347	0.398	0.375	0.328	0.383	0.367
VISA-7B [39]	0.453	0.439	0.395	0.437	0.422	0.380	0.429	0.413	0.440	0.422	0.373	0.425	0.407	0.356	0.412	0.396	0.425	0.406	0.352	0.410	0.393	0.336	0.394	0.379	0.401	0.383	0.331	0.387	0.371	0.315	0.372	0.358
VISA-13B [39]	0.512	0.488	0.444	0.491	0.473	0.426	0.481	0.463	0.495	0.472	0.419	0.477	0.458	0.401	0.462	0.445	0.477	0.456	0.396	0.461	0.442	0.378	0.443	0.427	0.452	0.431	0.372	0.436	0.417	0.354	0.418	0.402
JiT [31]	0.548	0.521	0.475	0.526	0.506	0.457	0.515	0.496	0.529	0.504	0.448	0.510	0.490	0.429	0.491	0.476	0.510	0.487	0.423	0.493	0.472	0.404	0.474	0.456	0.483	0.461	0.399	0.467	0.447	0.379	0.448	0.431
DT-R1	**0.725**	**0.692**	**0.649**	**0.702**	**0.679**	**0.627**	**0.688**	**0.670**	**0.698**	**0.667**	**0.614**	**0.675**	**0.653**	**0.592**	**0.660**	**0.640**	**0.673**	**0.642**	**0.587**	**0.649**	**0.629**	**0.568**	**0.634**	**0.615**	**0.648**	**0.618**	**0.562**	**0.622**	**0.601**	**0.544**	**0.609**	**0.589**

<results> and </results> for the subsequent reasoning. However, if execution errors are encountered, only relevant error information is provided (*i.e.*, final line of error messages), where verbose tracebacks containing file path information are dropped [15], will be appended to the <results> and </results>.

As complex reasoning visual tasks often require multi-step reasoning where the initial findings inform subsequent computational needs and computational results may reveal additional relationships requiring further reasoning, this reasoning process continues iteratively until the LLM determines the appropriate task type $\mathcal{T}$ within <task> and </task> tokens. The task type $\mathcal{T}$ specifies both the category of reasoning visual task (reasoning segmentation, reasoning grounding, reasoning summarization, or reasoning visual question answering) and the corresponding output format requirements. The final response $\mathcal{S}$ is then generated within <answer> and </answer> tokens, containing the task-specific output determined by $\mathcal{T}$. For reasoning segmentation, $\mathcal{S}$ includes the identified instance names and their corresponding segmentation mask file paths, while for reasoning grounding, $\mathcal{S}$ provides bounding box coordinates with instance names, for reasoning summarization or visual question answering, $\mathcal{S}$ contains natural language descriptions of relevant visual elements.

Reward Design Following previous work [8, 11, 15, 18], our reward is based on rules to avoid the computational overhead of training neural reward models. The total reward $R(\mathcal{Y})$ comprises one component that focuses on the formatting and another on the performance, denoted as:

$$R(\mathcal{Y}) = \alpha \cdot R_{\text{format}}(\mathcal{Y}) + \beta \cdot R_{\text{accuracy}}(\mathcal{Y}), \tag{1}$$

where α and β are the balancing parameters.

Format rewards $R_{\text{format}}(\mathcal{Y})$ evaluate the structural integrity of the rollout sequence by accessing the appearance of all required tokens in the correct order, as well as the validity of content within DT representation planning. Formally, the format reward is computed by $R_{\text{format}}(\mathcal{Y}) = R_{\text{token}}(\mathcal{Y}) + R_{\text{dag}}(\mathcal{G})$, where the token format reward $R_{\text{token}}(\mathcal{Y})$ verifies the presence and sequential ordering of all required pair of tokens, including <think>, <dt_plan>, <execute>, <task>, and <answer> with their corresponding closing tokens. The token format reward is defined as $+1$ if all tokens are correctly formatted, otherwise -1 to provide strong positive reinforcement for correct formatting while penalizing malformed outputs. The DAG format reward $R_{\text{dag}}(\mathcal{G})$ ensures that the DT construction plan within <dt_plan> and </dt_plan> tokens forms a valid DAG by $R_{\text{dag}}(\mathcal{Y}) = \mathbb{I}[\text{Valid Format}] \cdot \mathbb{I}[\text{Acyclic}] \cdot \mathbb{I}[\text{Valid Dependencies}] - 0.5$ where the indicators $\mathbb{I}(\cdot)$ verify that the plan is valid formatted, contains no circular dependencies, and specifies only available vision foundation models with correct dependency relationships. Therefore, $R_{\text{dag}}(\mathcal{G})$ becomes $+0.5$ if the DAG is valid otherwise -0.5. We use 0.5 instead of 1.0 to provide more granular feedback, since DAG validation represents a more complex structural constraint than simple token presence, requiring the model to learn intricate relationships between vision foundation models.

Accuracy rewards $R_{\text{accuracy}}(\mathcal{Y})$ evaluate the performance quality of the generated output in three dimensions: (1) code executability, (2) correctness of task identification, and (3) precision of the final result, namely $R_{\text{accuracy}}(\mathcal{Y}) = R_{\text{exec}}(\mathcal{C}) + R_{\text{task}}(\mathcal{T}) + R_{\text{result}}(\mathcal{S})$. The code executability reward $R_{\text{exec}}(\mathcal{C})$ validates that Python operations within the <execute> and </execute> tokens can be successfully executed without syntax or run-time errors. Following previous work [15], we define it as $R_{\text{exec}}(\mathcal{C}) = 0$ if the code executes successfully, otherwise -0.5 to penalize non-executable operations. The task identification reward $R_{\text{task}}(\mathcal{T})$ verifies that the LLM correctly identifies the appropriate reasoning visual task type from the implicit query Q, providing $+0.25$ if the task specification within <task> and </task> tokens matches the ground truth task category, otherwise 0. The final result accuracy reward $R_{\text{result}}(\mathcal{S})$ evaluates the correctness of the final output within <answer> and </answer> tokens, providing $+1$ for correct responses and -1 for incorrect ones to establish strong performance incentives. To be more concrete, for reasoning segmentation and reasoning grounding, correctness is determined by whether the IoU between the predicted and ground truth exceeds 0.5 following previous work [18]. Since we do not train a segmentation decoder [14] with in the framework, we focus on whether the LLM successfully identifies the correct object from the DT representation by making IoU greater than 0.5 an indicator of successful object identification and localization. For reasoning summarization and reasoning visual question answering tasks, we employ an LLM-as-a-judge [49] evaluation method that assesses the semantic correctness and factual accuracy of natural language responses generated against reference answers.

Training The training employs GRPO [26] to optimize LLM through the reward in Eq. (1). Specifically, during each training iteration, LLM generates multiple

DT-R1 Instruction Template
Solve the visual reasoning task through digital twin construction and analysis. Begin by reasoning within <think> and </think> to examine the query and determine the information requirements for addressing it. Subsequently, develop a digital twin construction plan enclosed in <dt_plan> and </dt_plan> by selecting appropriate vision foundation models and defining their interdependencies using a directed acyclic graph in JSON format. You have access to SAM2 for instance segmentation, DepthAnything2 for depth estimation, Qwen2.5-VL for semantic analysis, DINO-2 for visual feature extraction, OWLv2 for object detection, and OpenCV for frame-level processing. Following plan execution, the constructed digital twin representation becomes available within <dt_rep> and </dt_rep> for subsequent analysis. Proceed with reasoning over this structured representation, utilizing iterative analysis cycles as needed. For spatial or temporal computations, implement Python code within <execute> and </execute>, with outcomes appearing in <results> and </results>. Upon completing the reasoning, specify the task category within <task> and </task>, selecting from reasoning segmentation, grounding, summarization, or visual question answering, then deliver the final answer within <answer> and </answer>.
Query: *query*

Fig. 2: Prompt template for DT-R1. *query* will be replaced with the actual one during training and inference.

rollout sequences by sampling the current policy π_θ (*i.e.*, LLM), with the reward $R(\mathcal{Y})$ evaluating each sequence. Policy updates leverage these reward signals to reinforce successful reasoning patterns while discouraging format violations and inaccurate predictions, with GRPO's variance reduction ensuring stable convergence while maintaining exploration capabilities necessary for discovering effective reasoning across diverse visual scenarios. To prevent undesirable learning from optimizing externally generated content, including the DT representations $\mathcal{D}$ within the tokens <dt_rep> and </dt_rep> and the execution results $\mathcal{O}_i$ within the tokens <results> and </results>, we exclude these contexts from the training. Finally, we utilize a standardized prompt template for both training and inference, as illustrated in Fig. 2.

4 Experiments

Datasets The training of DT-R1 requires datasets organized as triplets $(\mathcal{X}, Q, \mathcal{T}, \mathcal{S})$, where $\mathcal{X}$ represents the input visual data (image or video), Q denotes the implicit text query, $\mathcal{T}$ specifies the task type, and $\mathcal{S}$ contains the ground truth output. The task type $\mathcal{T}$ encompasses four RVT categories: reasoning segmentation, reasoning grounding, reasoning summarization, and reasoning visual question answering, which can be derived from the dataset source itself. For training, we utilize four, including the training sets of ReasonSeg [14] for image reasoning segmentation, ReVOS [39] for video reasoning segmentation, GroundMore [5] for video grounding, along with 1,000 samples from ActivityNet-QA [44] for video

VQA and SumMe [9] for video summarization. For evaluation, we utilize six datasets, categorized into in-domain and out-of-domain assessments. In-domain evaluation utilizes test sets from training dataset sources: ReasonSeg [14] for image reasoning segmentation, and ReVOS [39] test set for video reasoning segmentation. Out-of-domain evaluation demonstrates generalization capabilities across four datasets: LLM-Seg40K [34] test set for image reasoning segmentation; JiTBench [30, 31] for video reasoning segmentation; and RVTBench [28], which provides evaluation across all four RVT types within the video modality.

Implementation Details We use DeepSeek-R1-Distill-Qwen-7B [8] as the backbone LLM for DT-R1 RL training. For DT representation construction, we follow previous works [28, 31] by including SAM2 [24], OWLv2 [20], DepthAnything [40], DINOv2 [21], Qwen2.5-VL [2] and OpenCV operators. All executable Python code execution utilizes Sandbox Fusion as the interpreter [15]. The GRPO [26] training configuration employs Low-Rank Adaptation (LoRA) [10] with rank 16 to enable efficient parameter updates, with a batch size of 32 and sample number of 8. We implement the training on 16 NVIDIA 4090 GPUs with 24GB memory using DeepSpeed [23] and trl libraries. The learning rate is set to 5e-5, and training converges within 8 epochs. While we use DeepSeek-R1-Distill-Qwen-7B as the default backbone, the framework is adaptable with Qwen2.5-7B achieves $\mathcal{J}$=0.798 (vs 0.808), and replacing GRPO with PPO yields $\mathcal{J}$=0.794, confirming that DT-R1 is backbone- and RL-algorithm-agnostic. Regarding computational cost, DT representation construction takes approximately 0.3 min/video on an RTX 4090, producing representations smaller than 1MB (approximately 2000 tokens/video). Inference requires 1.54s/query, which is substantially faster than JiT's 15.24s/query.

Table 3: Performance comparison of video reasoning grounding on RVTBench-grounding [28] via cIoU and gIoU.

Methods	Level 1 (Basic)							Level 2 (Intermediate)							Level 3 (Advanced)							Level 4 (Expert)						
	Semantic		Spatial		Temporal		Avg.	Semantic		Spatial		Temporal		Avg.	Semantic		Spatial		Temporal		Avg.	Semantic		Spatial		Temporal		Avg.
	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU gIoU
LISA-7B [14]	0.429	0.403	0.367	0.410	0.384	0.348	0.400 0.381	0.412	0.389	0.351	0.393	0.370	0.332	0.384 0.365	0.395	0.372	0.333	0.376	0.353	0.314	0.367 0.348	0.376	0.355	0.317	0.357	0.336	0.298	0.349 0.330
LISA-13B [14]	0.460	0.434	0.398	0.441	0.415	0.379	0.431 0.412	0.443	0.420	0.382	0.424	0.401	0.363	0.415 0.396	0.426	0.403	0.364	0.407	0.384	0.345	0.398 0.379	0.407	0.386	0.348	0.388	0.367	0.329	0.380 0.361
GSVA [37]	0.185	0.172	0.198	0.192	0.179	0.205	0.185 0.192	0.178	0.165	0.191	0.185	0.172	0.198	0.178 0.185	0.171	0.158	0.184	0.178	0.165	0.191	0.171 0.178	0.164	0.151	0.177	0.171	0.158	0.184	0.164 0.171
LLM-Seg [34]	0.218	0.235	0.294	0.245	0.262	0.318	0.249 0.275	0.211	0.228	0.287	0.238	0.255	0.311	0.242 0.268	0.204	0.221	0.280	0.231	0.248	0.304	0.235 0.261	0.197	0.214	0.273	0.224	0.241	0.297	0.228 0.254
V* [36]	0.065	0.071	0.089	0.072	0.078	0.096	0.075 0.082	0.061	0.067	0.085	0.068	0.074	0.092	0.071 0.078	0.058	0.064	0.082	0.065	0.071	0.089	0.068 0.075	0.055	0.061	0.079	0.062	0.068	0.086	0.065 0.072
Seg-Zero [18]	0.384	0.362	0.328	0.395	0.373	0.339	0.358 0.369	0.371	0.349	0.315	0.382	0.360	0.326	0.345 0.356	0.358	0.336	0.302	0.369	0.347	0.313	0.332 0.343	0.345	0.323	0.289	0.356	0.334	0.300	0.319 0.330
VISA-7B [39]	0.318	0.294	0.267	0.325	0.301	0.274	0.293 0.300	0.305	0.281	0.254	0.312	0.288	0.261	0.280 0.287	0.292	0.268	0.241	0.299	0.275	0.248	0.267 0.274	0.279	0.255	0.228	0.286	0.262	0.235	0.254 0.261
VISA-13B [39]	0.491	0.465	0.429	0.472	0.446	0.410	0.462 0.443	0.474	0.451	0.413	0.456	0.432	0.394	0.446 0.427	0.457	0.434	0.395	0.438	0.415	0.376	0.429 0.410	0.438	0.417	0.379	0.419	0.398	0.360	0.411 0.392
JiT [31]	0.518	0.492	0.456	0.499	0.473	0.437	0.489 0.470	0.501	0.479	0.440	0.483	0.459	0.421	0.473 0.454	0.484	0.461	0.422	0.465	0.442	0.403	0.456 0.437	0.467	0.444	0.405	0.448	0.425	0.386	0.439 0.420
GroundMore [3]	0.542	0.518	0.485	0.524	0.500	0.467	0.515 0.497	0.526	0.504	0.469	0.508	0.486	0.451	0.500 0.482	0.509	0.487	0.452	0.491	0.469	0.434	0.483 0.465	0.492	0.470	0.435	0.474	0.452	0.417	0.466 0.448
DT-R1	**0.699**	**0.673**	**0.638**	**0.680**	**0.654**	**0.619**	**0.670 0.651**	**0.674**	**0.651**	**0.615**	**0.656**	**0.632**	**0.597**	**0.647 0.628**	**0.653**	**0.628**	**0.594**	**0.634**	**0.609**	**0.575**	**0.625 0.606**	**0.629**	**0.606**	**0.571**	**0.610**	**0.587**	**0.552**	**0.602 0.583**

Evaluations of Video Reasoning Segmentation As shown in Table 1, DT-R1 achieves substantial improvements on JiTBench [31], achieving the highest $\mathcal{J}$ scores of 0.837, 0.808, and 0.785 across difficulty levels 1-3 respectively, surpassing the previous best method by 4.5%, 4.2%, and 3.8%. Similarly, Table 2 demonstrates DT-R1's superior performance on RVTBench-segmentation [28], where it achieves the highest $\mathcal{J}$ scores of 0.688, 0.660, 0.634, and 0.609 across reasoning difficulty levels 1-4, with improvements of 17.3%, 16.6%, 16.0%, and 16.1% over JiT [31]. On ReVoS (Table 5), DT-R1 exhibits particularly strong

Table 4: Performance comparison on reasoning VQA and summarization tasks with RVTBench [28] across BLEU-4 [22], ROUGE-L [16], BertScore [46], and CIDEr [33].

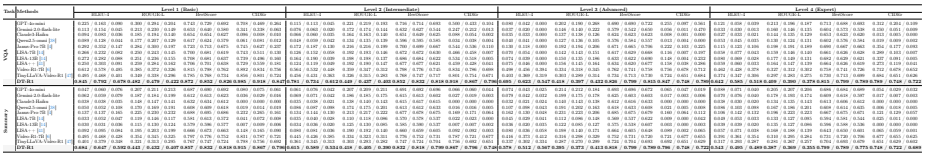

Table 5: Comparison of video reasoning segmentation on ReVoS [39], where DT-R1 also achieves the best performance.

Methods	Referring		Reasoning		Overall	
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
LISA-7B [14]	0.443	0.471	0.338	0.384	0.391	0.427
LISA-13B [14]	0.452	0.479	0.343	0.391	0.398	0.435
GSVA [37]	0.445	0.465	0.340	0.395	0.418	0.433
LLM-Seg [34]	0.402	0.410	0.305	0.331	0.354	0.381
V* [36]	0.219	0.209	0.287	0.256	0.234	0.265
Seg-Zero [18]	0.487	0.512	0.361	0.398	0.424	0.455
VISA-7B [39]	0.556	0.591	0.420	0.467	0.488	0.529
VISA-13B [39]	0.573	0.608	0.437	0.481	0.505	0.545
JiT [31]	0.691	0.723	0.618	0.652	0.655	0.688
DT-R1	**0.911**	**0.887**	**0.814**	**0.805**	**0.875**	**0.863**

gains in reasoning tasks with a $\mathcal{J}$ score of 0.814 compared to JiT's 0.618, while maintaining high performance on referring tasks. These consistent improvements across diverse benchmarks support that learning to construct and reason over DT representations through RL enables more effective preservation of spatial-temporal relationships compared to token-based approaches. An analysis of failure cases on ReVOS reveals that 68% of reasoning errors stem from ambiguous spatial references (*e.g.*, "closest" with equidistant objects), while 32% arise from incomplete SAM2 masks in occluded or crowded scenes.

Evaluations of Image Reasoning Segmentation To evaluate DT-R1's capability in handling image-based reasoning segmentation tasks, we evaluate its performance on both in-domain and out-of-domain benchmarks. Table 6 shows that DT-R1 achieves a gIoU of 0.746 and 0.812 on ReasonSeg's short and long queries respectively, representing improvements of 12.8% and 12.9% over the previous best method JiT. Importantly, DT-R1 also demonstrates generalization capabilities on the out-of-domain LLM-Seg40K [34] dataset with a gIoU of 0.641, outperforming JiT's 0.485 by 15.6%. These results show that DT-R1 applies to both image and video modalities.

Evaluations of Video Reasoning Grounding Video reasoning (spatial-temporal) grounding aims to localize objects through bounding boxes with respect to a specific time segment of video based on given implicit text queries. Table 3 reveals that DT-R1 consistently achieves the best performance across all four difficulty levels of RVTBench-grounding [28], with cIoU scores of 0.670, 0.647, 0.625, and

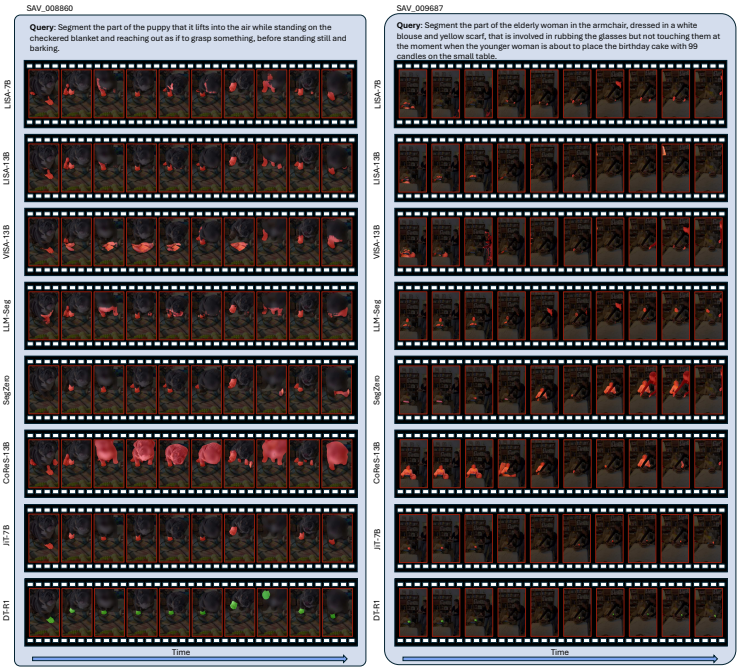

Fig. 3: Qualitative comparison of video reasoning segmentation results on JiTBench examples. Each row shows the segmentation results from different methods across time. Green masks indicate correct segmentation where the predicted region matches the ground truth, while red masks denote incorrect predictions. DT-R1 consistently produces accurate segmentations across both temporal sequences, correctly identifying the target regions throughout the video frames.

Table 6: Performance evaluation of image reasoning segmentation on ReasonSeg [14] and LLM-Seg40K [34] via gIoU and cIoU.

Methods	ReasonSeg [14]				LLM-Seg40K [34]	
	Short Query		Long Query		Overall	
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
LISA-7B [14]	0.483	0.463	0.579	0.597	0.376	0.485
LISA-13B [14]	0.554	0.506	0.632	0.653	0.392	0.502
GSVA [37]	0.238	0.218	0.314	0.305	0.185	0.202
LLM-Seg [34]	0.210	0.203	0.253	0.248	0.455	0.542
V* [36]	0.438	0.430	0.483	0.495	0.312	0.338
Seg-Zero [18]	0.531	0.498	0.587	0.612	0.442	0.478
VISA-7B [39]	0.453	0.482	0.453	0.455	0.358	0.394
VISA-13B [39]	0.497	0.521	0.498	0.502	0.381	0.417
JiT [31]	0.618	0.584	0.683	0.701	0.485	0.528
DT-R1	**0.746**	**0.713**	**0.812**	**0.834**	**0.641**	**0.677**

0.602 from levels 1 through 4. These results substantially exceed the current state-of-the-art method such as GroundMore [5], which is specifically designed for grounding tasks. Notably, DT-R1 maintains balanced performance across all three reasoning dimensions, achieving semantic/spatial/temporal cIoU scores of

Table 7: Ablation study on reward components using JiTBench [31] Level 2.

Configuration	Format Rewards		Accuracy Rewards		Performance		Training Success Rate
	R_{token}	R_{dag}	R_{exec}	$R_{\text{task}} + R_{\text{result}}$	$\mathcal{J}$	$\mathcal{F}$	
w/o Format rewards			✓	✓	0.412	0.438	31.5%
w/o DAG validation	✓		✓	✓	0.687	0.712	78.3%
w/o Code execution	✓	✓		✓	0.724	0.756	85.7%
w/o Accuracy rewards	✓	✓			0.523	0.548	62.1%
Only task accuracy				✓	0.298	0.315	22.8%
Full DT-R1	✓	✓	✓	✓	**0.808**	**0.845**	94.2%

0.699/0.673/0.638 at level 1, compared to GroundMore's 0.542/0.518/0.485. The substantial margins of DT-R1 over both general-purpose and task-specific baselines demonstrate that our unified RL framework successfully develops grounding capabilities without requiring specialized architectural modifications or task-specific tuning.

Evaluations of Video Reasoning VQA and Summarization We evaluate DT-R1's performance on reasoning VQA and summarization on the RVTBench [28] in Table 4, where DT-R1 demonstrates better performance across all evaluation metrics. This performance advantage remains consistent as task complexity increases, with DT-R1 maintaining strong results even at the difficulty level 4, where it achieves 0.642 on BLEU-4 compared to the next best method, Video-R1-7B [6], which scores 0.462. Recent RL-based video reasoning models, such as Video-R1-7B [6], TinyLLaVA-Video-R1 [47], and DT-R1 demonstrate improvements over traditional approaches such as LISA [14] and its variants. For the summarization task, while overall scores are lower than VQA across all methods, DT-R1 maintains its commanding lead with scores ranging from 0.684 at the difficulty level 1 to 0.543 at level 4 for BLEU-4. The consistent performance gap between DT-R1 and other methods across all difficulty levels underscores the effectiveness of our DT representation based reasoning framework across diverse RVTs.

Ablation Study We conduct an ablation study on the contribution of each reward component on JiTBench Level 2, as presented in Tab. 7. The results demonstrate that each component of our reward function helps achieve optimal performance and training stability. Removing format rewards entirely leads to substantial performance degradation, with $\mathcal{J}$ and $\mathcal{F}$ scores dropping to 0.412 and 0.438 respectively, alongside a reduction in training success rate to 31.5%. The DAG validation component proves particularly important, as its removal reduces performance to 0.687 $\mathcal{J}$ and 0.712 $\mathcal{F}$, while the code execution reward also contributes meaningfully to the final results. The configuration using only task accuracy performs poorly across all metrics (0.298 $\mathcal{J}$, 0.315 $\mathcal{F}$), highlighting the necessity of our reward design. We further compare with Seg-R1 [43], another RL-based method, on ReasonSeg: DT-R1 achieves gIoU=0.746 vs Seg-R1's 0.628 (+11.8%). Additionally, individual specialist model ablation on JiTBench Level 2 shows that removing SAM2 drops $\mathcal{J}$ to 0.684 (−12.8%), DepthAnything to

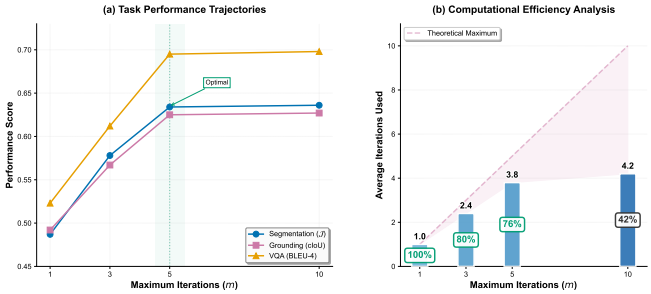

Fig. 4: Analysis of iterative reasoning cycles impact on DT-R1 performance. (a) It demonstrates the performance across segmentation, grounding, and visual question answering tasks as maximum iterations increase from 1 to 10, revealing performance saturation at 5 iterations. (b) It illustrates computational efficiency, where the average number of reasoning iterations actually utilized versus the theoretical maximum, with efficiency percentages indicating the model's adaptive termination behavior.

0.756 (−5.6%), OWLv2 to 0.732 (−8.0%), and Qwen2.5-VL to 0.778 (−3.4%), confirming each perception component contributes to the overall performance. Additionally, we explore the impact of iterative reasoning cycles on performance, as shown in Fig. 4. The DR-R1 demonstrates computational efficiency by adaptively utilizing an average of 3.8 iterations when permitted up to 5, indicating that it learns to terminate reasoning cycles appropriately without exhausting the maximum allowed iterations.

5 Conclusion

We present DT-R1, a RL framework that trains LLMs to construct and reason over digital twin representations as a unified solution for diverse reasoning visual tasks. Despite requiring multiple vision foundation models, DT-R1 achieves 1.54s/query inference time, which is 10× faster than JiT's 15.24s/query. While the performance of the DT-R1 approach seems promising, the current design requires executing multiple vision foundation models during inference for DT representation construction; hence one future direction can look into efficient implementation such as distilled models [1, 48] or caching [19] to reduce computations. Additionally, DT-R1 currently processes only visual and textual modalities, yielding opportunities to incorporate audio and other sensory data into the DT representation as another future direction. Future work can also explore scaling to larger LLMs to enhance reasoning capabilities and enable more complex DT construction that capture complex spatial-temporal relationships. By demonstrating that pure RL can train LLMs to show visual reasoning capabilities, DT-R1 shows a promise that can complement the current paradigm [27] of maintaining separate models for each reasoning visual task.

Acknowledgments. Y. Shen was supported in part by the JHU Amazon Initiative for Artificial Intelligence (AI2AI) fellowship program.

Disclosure of Interests. The authors have no competing interests in the paper.

References

1. Fastsam3d: An efficient segment anything model for 3d volumetric medical images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 542–552. Springer (2024) 14
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025) 6, 10
3. Bao, X., Sun, S., Ma, S., Zheng, K., Guo, Y., Zhao, G., Zheng, Y., Wang, X.: Cores: Orchestrating the dance of reasoning and segmentation. In: European Conference on Computer Vision. pp. 187–204. Springer (2024) 3
4. Chen, X., Wu, Z., Liu, X., Pan, Z., Liu, W., Xie, Z., Yu, X., Ruan, C.: Janus-pro: Unified multimodal understanding and generation with data and model scaling (2025), https://arxiv.org/abs/2501.17811 11
5. Deng, A., Chen, T., Yu, S., Yang, T., Spencer, L., Tian, Y., Mian, A.S., Bansal, M., Chen, C.: Motion-grounded video reasoning: Understanding and perceiving motion at pixel level. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 8625–8636 (2025) 9, 10, 12
6. Feng, K., Gong, K., Li, B., Guo, Z., Wang, Y., Peng, T., Wu, J., Zhang, X., Wang, B., Yue, X.: Video-r1: Reinforcing video reasoning in mllms. arXiv preprint arXiv:2503.21776 (2025) 11, 13
7. Gao, D., Ji, L., Zhou, L., Lin, K.Q., Chen, J., Fan, Z., Shou, M.Z.: Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn. arXiv preprint arXiv:2306.08640 (2023) 1
8. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025) 3, 7, 10
9. Gygli, M., Grabner, H., Riemenschneider, H., Van Gool, L.: Creating summaries from user videos. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13. pp. 505–520. Springer (2014) 10
10. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022) 10
11. Jin, B., Zeng, H., Yue, Z., Yoon, J., Arik, S., Wang, D., Zamani, H., Han, J.: Search-r1: Training llms to reason and leverage search engines with reinforcement learning. arXiv preprint arXiv:2503.09516 (2025) 4, 7
12. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 787–798 (2014) 1
13. Kirillov, A., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023) 1, 3, 5

14. Lai, X., Tian, Z., Chen, Y., et al.: Lisa: Reasoning segmentation via large language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9579–9589 (2024) 1, 2, 3, 7, 8, 9, 10, 11, 12, 13
15. Li, X., Zou, H., Liu, P.: Torl: Scaling tool-integrated rl. arXiv preprint arXiv:2503.23383 (2025) 4, 7, 8, 10
16. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-1013/ 11
17. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26296–26306 (2024) 4
18. Liu, Y., Peng, B., Zhong, Z., Yue, Z., Lu, F., Yu, B., Jia, J.: Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. arXiv preprint arXiv:2503.06520 (2025) 2, 3, 4, 7, 8, 10, 11, 12
19. Liu, Z., Liu, B., Wang, J., Dong, Y., Chen, G., Rao, Y., Krishna, R., Lu, J.: Efficient inference of vision instruction-following models with elastic cache. In: European Conference on Computer Vision. pp. 54–69. Springer (2024) 14
20. Minderer, M., Gritsenko, A., Houlsby, N.: Scaling open-vocabulary object detection. Advances in Neural Information Processing Systems **36** (2024) 3, 10
21. Oquab, M., Darcet, T., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) 3, 10
22. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. p. 311–318. ACL '02, Association for Computational Linguistics, USA (2002). https://doi.org/10.3115/1073083.1073135, https://doi.org/10.3115/1073083.1073135 11
23. Rasley, J., Rajbhandari, S., Ruwase, O., He, Y.: Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 3505–3506 (2020) 10
24. Ravi, N., Gabeur, V., Hu, Y.T., et al.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024) 1, 3, 5, 10
25. Sarch, G., Saha, S., Khandelwal, N., Jain, A., Tarr, M.J., Kumar, A., Fragkiadaki, K.: Grounded reinforcement learning for visual reasoning. arXiv preprint arXiv:2505.23678 (2025) 4
26. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024) 2, 8, 10
27. Shen, Y., Ding, H., Seenivasan, L., Shu, T., Unberath, M.: Position: Foundation models need digital twin representations. arXiv preprint arXiv:2505.03798 (2025) 2, 3, 14
28. Shen, Y., Li, C., Fan, C., Unberath, M.: Rvtbench: A benchmark for visual reasoning tasks. arXiv preprint arXiv:2505.11838 (2025) 1, 3, 4, 6, 7, 10, 11, 13
29. Shen, Y., Li, C., Liu, B., Li, C.Y., Porras, T., Unberath, M.: Operating room workflow analysis via reasoning segmentation over digital twins. arXiv preprint arXiv:2503.21054 (2025) 1, 2, 4
30. Shen, Y., Li, C., Xiong, F., Jeong, J.O., Wang, T., Latman, M., Unberath, M.: Reasoning segmentation for images and videos: A survey. arXiv preprint arXiv:2505.18816 (2025) 1, 3, 10

31. Shen, Y., Liu, B., Li, C., Seenivasan, L., Unberath, M.: Online reasoning video segmentation with just-in-time digital twins. arXiv preprint arXiv:2503.21056 (2025) 2, 3, 4, 6, 7, 10, 11, 12, 13
32. Shi, W., Shen, Y.: Reinforcement fine-tuning for reasoning towards multi-step multi-source search in large language models. arXiv preprint arXiv:2506.08352 (2025) 5
33. Vedantam, R., Zitnick, C.L., Parikh, D.: Cider: Consensus-based image description evaluation (2015), https://arxiv.org/abs/1411.5726 11
34. Wang, J., Ke, L.: Llm-seg: Bridging image segmentation and large language model reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1765–1774 (2024) 3, 7, 10, 11, 12
35. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022) 2
36. Wu, P., Xie, S.: V*: Guided visual search as a core mechanism in multimodal llms. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13084–13094 (2024) 7, 10, 11, 12
37. Xia, Z., et al.: Gsva: Generalized segmentation via multimodal large language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3858–3869 (2024) 7, 10, 11, 12
38. Xu, J., Guo, Z., He, J., Hu, H., He, T., Bai, S., Chen, K., Wang, J., Fan, Y., Dang, K., et al.: Qwen2.5-omni technical report. arXiv preprint arXiv:2503.20215 (2025) 11
39. Yan, C., Wang, H., Yan, S., et al.: Visa: Reasoning video object segmentation via large language models. arXiv preprint arXiv:2407.11325 (2024) 1, 2, 3, 7, 9, 10, 11, 12
40. Yang, L., Kang, B., Huang, Z., et al.: Depth anything v2. arXiv preprint arXiv:2406.09414 (2024) 3, 6, 10
41. Yang, S., Qu, T., Lai, X., Tian, Z., Peng, B., Liu, S., Jia, J.: An improved baseline for reasoning segmentation with large language model. CoRR (2023) 11
42. Yang, Y., Jiang, P.T., Wang, J., Zhang, H., Zhao, K., Chen, J., Li, B.: Empowering segmentation ability to multi-modal large language models. arXiv preprint arXiv:2403.14141 (2024) 3
43. You, Z., Wu, Z.: Seg-r1: Segmentation can be surprisingly simple with reinforcement learning. arXiv preprint arXiv:2506.22624 (2025) 4, 13
44. Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., Tao, D.: Activitynet-qa: A dataset for understanding complex web videos via question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 9127–9134 (2019) 9
45. Zakari, R.Y., Owusu, J.W., Wang, H., Qin, K., Lawal, Z.K., Dong, Y.: Vqa and visual reasoning: An overview of recent datasets, methods and challenges. arXiv preprint arXiv:2212.13296 (2022) 2
46. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert (2020), https://arxiv.org/abs/1904.09675 11
47. Zhang, X., Wen, S., Wu, W., Huang, L.: Tinyllava-video-r1: Towards smaller lmms for video reasoning. arXiv preprint arXiv:2504.09641 (2025) 11, 13
48. Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., Wang, J.: Fast segment anything. arXiv preprint arXiv:2306.12156 (2023) 14
49. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al.: Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems **36**, 46595–46623 (2023) 8

50. Zhu, C., Wang, T., Zhang, W., Chen, K., Liu, X.: Scanreason: Empowering 3d visual grounding with reasoning capabilities. In: European Conference on Computer Vision. pp. 151–168. Springer (2024) 1, 2