Structured Agent Distillation for Large Language Model Agents

Jun Liu
Carnegie Mellon University
Pittsburgh, United States
liujun@cmu.edu

Zhenglun Kong
Harvard University
Boston, United States
zhenglun_kong@hms.harvard.edu

Peiyan Dong
MIT
Boston, United States
peggy281@mit.edu

Changdi Yang
Northeastern University
Boston, United States
yang.changd@northeastern.edu

Tianqi Li
Carnegie Mellon University
Pittsburgh, United States
tianqinl@cs.cmu.edu

Yanyue Xie
Northeastern University
Boston, United States
xie.yany@northeastern.edu

Yanfan Gong
Adobe Research
Seattle, United States
yifang@adobe.com

Xuan Shen
Northeastern University
Boston, United States
shenxuan0516@gmail.com

Hao Tang
Carnegie Mellon University & NUS
Pittsburgh, United States
bjdxtanghao@gmail.com

Pu Zhao
Northeastern University
Boston, United States
p.zhao@northeastern.edu

Geng Yuan
University of Georgia
Athens, United States
geng.yuan@uga.edu

Wei Niu
University of Georgia
Athens, United States
wniu@uga.edu

Wenbin Zhang
Florida International University
Miami, United States
wenbin.zhang@fiu.edu

Xue Lin
Northeastern University
Boston, United States
xue.lin@northeastern.edu

Dong Huang
Carnegie Mellon University
Pittsburgh, United States
donghuang@cmu.edu

Yanzhi Wang
Northeastern University
Boston, United States
yanz.wang@northeastern.edu

ABSTRACT

Large language models (LLMs) exhibit strong capabilities as decision-making agents by interleaving reasoning and actions, as seen in ReAct-style frameworks. Yet, their practical deployment is constrained by high inference costs and large model sizes. We propose **Structured Agent Distillation**, the first framework to distill a ReAct-based LLM agent into a smaller model while preserving both reasoning fidelity and action consistency. Our method introduces a structured, span-level distillation strategy that explicitly segments trajectories into reasoning and action spans, enabling fine-grained alignment beyond standard token-level imitation. Unlike other advanced distillation methods, Our method segments trajectories into `[REASON]` and `[ACT]` spans, applying segment-specific losses to align each component with the teacher's behavior. This structure-aware supervision enables compact agents to better replicate the teacher's decision process. Experiments on ALFWorld, HotPotQA-ReAct, and WebShop show that our approach consistently outperforms token-level and imitation learning baselines, achieving significant compression with minimal performance drop. Scaling and ablation results further highlight the importance of span-level alignment for efficient and deployable agents. We will release code upon acceptance.

KEYWORDS

Agent Distillation, Span-Level Alignment, Reasoning and Action Segmentation, Large Language Models (LLMs), Span-Level Alignment

Corresponding Authors: Pu Zhao, Hao Tang.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/OLHJ8062

ACM Reference Format:
Jun Liu, Zhenglun Kong, Peiyan Dong, Changdi Yang, Tianqi Li, Yanyue Xie, Yanfan Gong, Xuan Shen, Hao Tang, Pu Zhao, Geng Yuan, Wei Niu, Wenbin Zhang, Xue Lin, Dong Huang, and Yanzhi Wang. 2026. Structured Agent Distillation for Large Language Model Agents. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 26 pages. https://doi.org/10.65109/OLHJ8062

Table 1: Comparison between LLMs and LLM-based general agents in terms of token efficiency.

Dimension	LLMs	LLM-based General Agents
Application	Static, single-turn tasks (QA, etc.)	Multi-turn interactive tasks (e.g., WebNav)
Objective	Minimize FLOPs in generation	Mini token budget across reasoning + action
Token Source	Fixed input tokens per prompt	Dynamic tokens in reasoning-action steps
Latency Target	Single-step inference efficiency	End-to-end efficiency over agent steps
Output Format	Final text sequence	Task trajectory (reasoning steps + actions)

1 INTRODUCTION

Large language models (LLMs) [52, 79, 84, 85] have recently been extended beyond language modeling into decision-making roles, giving rise to *LLM-based general agents*—systems that solve complex tasks by interleaving multi-step reasoning and tool-augmented actions. Frameworks like ReAct [75], Toolformer [44], and WebGPT [36] demonstrate that LLMs can operate through structured *reasoning-action trajectories*—sequences alternating between deliberation and execution to complete tasks such as planning, web navigation, and multi-hop question answering. Chain-of-thought (CoT) prompting [65, 66] encourages models to decompose complex tasks into intermediate reasoning steps before acting, reinforcing the need to preserve reasoning–action structure during training.

Despite their effectiveness, LLM-based general agents remain costly to deploy due to model size and inference overhead [48, 51, 77, 78]. To address this, recent work distills large agents into smaller student models. However, most approaches rely on token-level supervision [6, 9, 10, 14, 35, 62, 82], which treats the agent trajectory as a flat token sequence and aligns predictions step by step—ignoring its structured composition of reasoning and action.

Limitations of Token-Level Distillation. This paradigm fails to capture the *structural nature* of agent behavior: (i) it overlooks long-range dependencies between reasoning and action [86]; (ii) it lacks span-level supervision, blurring the distinction between planning and execution; (iii) it causes semantic drift during rollouts, degrading coherence and task success.

Our Approach: Structured Agent Distillation. We propose a structure-aware compression framework that explicitly models the compositional structure of agent behavior. Our method segments each trajectory into `[REASON]` and `[ACT]` spans and supervises them with span-specific objectives. By applying segment-aware masking and reasoning–action alignment, our approach preserves both the rationale and the resulting decision—enabling more faithful and coherent student agents.

Our Contributions:

- Our work is the first to distill ReAct-style LLM agents using structured span-level supervision: we segment trajectories into reasoning and action spans and apply span-specific alignment via token-level masks, improving over naive token-level distillation.
- We validate our approach on ALFWorld, HotPotQA-ReAct, and WebShop, achieving consistent gains in task success, planning efficiency, and chain-of-thought (CoT) alignment over strong token-level baselines.
- We conduct comprehensive scaling and ablation studies, demonstrating that segment-level supervision is critical for training compact and robust student agents.

2 MOTIVATION

2.1 Why Token-Level Distillation Falls Short

LLMs vs. General Agents. While LLMs focus on single-turn generation, LLM-based general agents operate in interactive settings where structured reasoning and action unfold over multiple steps. Token efficiency in agents must therefore consider trajectory-level [40, 54] latency and semantic role differentiation. Table 1 summarizes the key distinctions between the two paradigms.

This structured nature of agent trajectories poses unique challenges for compression and acceleration. In particular, existing methods such as token-level distillation [14, 22], originally designed for next-token prediction, fail to capture the hierarchical nature of agent behavior. Token-level distillation supervises the student at each decoding step using cross-entropy [4] or KL divergence [23] between teacher and student outputs. While this is effective for language modeling, it fails to account for the structured nature of agent trajectories—specifically the distinction between intermediate reasoning and final action execution.

Critically, *token-level methods lack structural awareness*, treating all tokens equally without distinguishing their functional roles in the agent trajectory. In practice, trajectories often alternate between internal reasoning steps and external actions—two semantically distinct spans that require different forms of supervision. As a result, the student learns to match surface-level actions while ignoring the underlying rationale, often skipping key planning steps required to complete the task.

2.2 Toward Structured Agent Distillation

We propose **Structured Agent Distillation (SAD)**, which segments trajectories into `[REASON]` and `[ACT]` spans and applies span-specific supervision to improve structural imitation. A curriculum mechanism further enhances stability by ordering training examples by complexity. Table 2 summarizes representative LLM-based agent frameworks in terms of four dimensions: external tool usage, reasoning-action alignment, segment-aware supervision, and curriculum-guided training. While prior methods such as ReAct and Voyager support structured reasoning and tool use, they lack segment-level supervision and curriculum scheduling. In contrast, our SAD framework uniquely supports all four dimensions, enabling more faithful and efficient agent compression.

3 STRUCTURED AGENT DISTILLATION FRAMEWORK

The proposed **SAD** segments teacher trajectories into reasoning (`Reason`) and interaction (`Action/Observation`) spans, each supervised independently to promote phase-specific alignment. As shown in Figure 1, the teacher agent, given an observation and

Table 2: Comparison of LLM agent training frameworks. Only our method supports all four dimensions of structured agent distillation. Tool: supports external API calls or tool use; R–A Align.: aligns structured reasoning and action spans; Seg.-aware Sup.: applies supervision across reasoning–action sequences; Curric.: uses trajectory difficulty for progressive training [24].

Framework	Tool	R–A Align.	Seg.-aware Sup.	Curric.
Token-Level KD [14]	✗	✗	✗	✗
ReAct [75]	✓	✓	✗	✗
Toolformer [44]	✓	✗	✗	✗
Voyager [63]	✓	✓	✗	✗
SAD (Ours)	✓	✓	✓	✓

task prompt, produces `[REASON]`ing traces and `[ACT]`ion outputs, forming a trajectory $\tau = (\texttt{reason}, \texttt{action})$ used for curriculum sampling. The student learns from τ via two losses: (1) *CoT-Policy Alignment* $\mathcal{L}_{\text{CoT}}$ aligns reasoning, and (2) *Action Consistency* $\mathcal{L}_{\text{Act}}$ aligns decisions. Refer to Appendix A for comprehensive analysis.

3.1 Problem Formulation

We aim to distill high-capacity ReAct-style teacher agents into smaller student models while preserving structured decision-making behavior. Each teacher's trajectory is a sequence of interleaved reasoning and action components:

$$\tau = [(r_1, r_2, \ldots, r_k), (a_1, a_2, \ldots, a_m)], \tag{1}$$

where $r_i \in \mathcal{R}$ are reasoning tokens (e.g., CoT steps), and $a_j \in \mathcal{A}$ are action tokens (e.g., tool calls, answers). Given a teacher policy $\pi_T(\tau)$, the goal is to train a compact student policy $\pi_\theta(\tau)$ such that

$$\pi_\theta(\tau) \approx \pi_T(\tau), \tag{2}$$

preserving both semantic reasoning and execution structure beyond token-level matching.

To enable sequence-to-sequence modeling, we linearize each trajectory into a flattened form with segment markers:

$$\tau' = \texttt{[REASON]}\ r_1 \cdots r_k\ \texttt{[ACT]}\ a_1 \cdots a_m.$$

We tokenize this as

$$x = \texttt{Tokenize}(\tau') = (x_1, x_2, \ldots, x_T), \tag{3}$$

and assign each token x_t a segment label $s_t \in \{\text{Reason}, \text{Action}\}$, indicating its span. These labels are used to compute segment-aware losses during training.

For clarity, we adopt the following notation: τ denotes the structured reasoning–action trajectory, τ' its linearized form with explicit segment markers, and $x = \texttt{Tokenize}(\tau')$ the token sequence processed by the model. Accordingly, π_θ always operates on tokenized inputs x, while τ and τ' are used only for segmentation and mask construction.

3.2 Trajectory Segmentation

Given a teacher-generated trajectory τ, we decompose it into two disjoint spans:

$$(\tau^{(r)}, \tau^{(a)}) \leftarrow \text{Segment}(\tau),$$

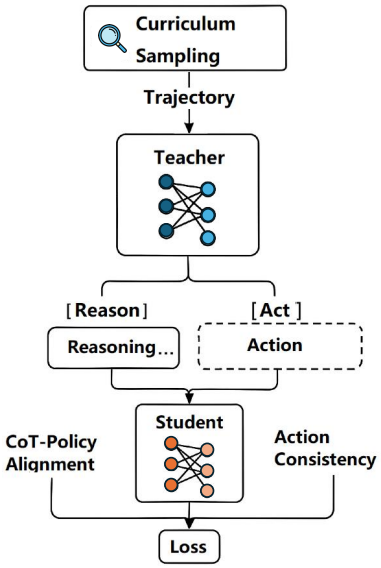

Figure 1: Structured Agent Distillation framework. The teacher provides segmented reasoning–action trajectories. The student aligns CoT traces and actions via span-specific KL losses, with projected gradients and curriculum sampling for stability.

where $\tau^{(r)}$ denotes the reasoning span and $\tau^{(a)}$ denotes the action span. This segmentation is performed via lightweight rule-based parsing based on prompt templates consistent across tasks (see Appendix D). The segmented trajectory is then tokenized into a sequence x defined in Eq. (3). While the above formulation assumes a single reasoning–action pair, SAD naturally extends to multi-step ReAct trajectories $\tau = [(r^{(i)}, a^{(i)}, o^{(i)})]_{i=1}^{K}$ by applying union masks over multiple reasoning/action spans. A detailed construction and example are provided in Appendix M.

3.3 Structured Agent Distillation Objectives

To supervise student agents under SAD, we align binary token masks $m_r(t)$ and $m_a(t)$ with the tokenized sequence $x = (x_1, \ldots, x_T)$, enabling selective supervision over structurally distinct parts of the trajectory.

(1) CoT-Policy Alignment Loss. For reasoning tokens, the student's conditional distribution $p_S(\cdot \mid x_{<t})$ is aligned with the teacher's distribution $p_T(\cdot \mid x_{<t})$ using Kullback–Leibler(KL) divergence:

$$\mathcal{L}_{\text{CoT}} = \sum_{t=1}^{T} m_r(t)\, \text{KL}(p_T(\cdot \mid x_{<t}) \,\|\, p_S(\cdot \mid x_{<t}))\,. \tag{4}$$

(2) Action Consistency Loss. For action tokens, we similarly minimize KL divergence:

$$\mathcal{L}_{\text{Act}} = \sum_{t=1}^{T} m_a(t)\, \text{KL}(p_T(\cdot \mid x_{<t}) \,\|\, p_S(\cdot \mid x_{<t}))\,. \tag{5}$$

CoT-Policy Alignment operates over the full vocabulary during `[REASON]` spans to guide the student's intermediate reasoning steps, encouraging alignment with the teacher's chain-of-thought. In

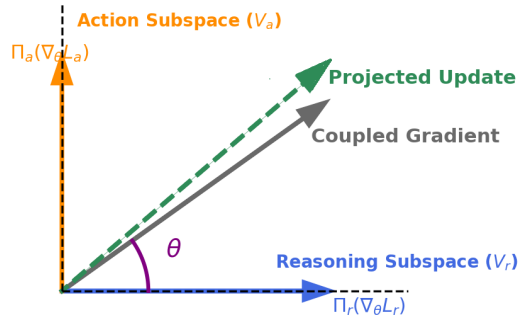

Figure 2: Optimization structure of Structured Agent Distillation (SAD) via gradient projection. Standard token-level KL (gray) couples reasoning and action gradients, leading to a conflict angle θ. SAD resolves this by projecting gradients onto reasoning ($\mathcal{V}_r$) and action ($\mathcal{V}_a$) subspaces, forming an orthogonal decomposition. The resulting projected update (green) follows the composite direction $\Pi_{\text{reason}}(\nabla_\theta \mathcal{L}_{\text{CoT}}) + \Pi_{\text{action}}(\nabla_\theta \mathcal{L}_{\text{Act}})$, eliminating cross-span interference and clarifying SAD's optimization geometry.

contrast, Action Consistency Loss applies KL over a discrete action space during `[ACT]` spans, ensuring that the student replicates the teacher's action decisions.

Final Objective. The total structured loss aggregates these terms:

$$\mathcal{L}_{\text{total}} = \lambda_r \cdot \mathcal{L}_{\text{CoT}} + \lambda_a \cdot \mathcal{L}_{\text{Act}}, \tag{6}$$

where λ_r and λ_a are scalar weights balancing the two objectives. We set $\lambda_r = \lambda_a = 1.0$ to equally weight reasoning and action supervision in the final loss.

Clarification. Although the loss terms are unscaled, this formulation is *not* equivalent to computing a single token-level KL over the entire vocabulary. Token-level KL normalizes over the joint token space $\mathcal{V}_r \cup \mathcal{V}_a$, which couples gradients from frequent reasoning tokens and rare but critical action tokens. In contrast, SAD applies KL over disjoint normalization domains ($\mathcal{V}_r$ for reasoning, $\mathcal{V}_a$ for action), thereby changing both the normalization space and gradient direction. This decomposition alters the optimization geometry and prevents cross-span interference, constituting a fundamental difference from a flat token-level KL even when λ_r and λ_a are equal. While our formulation assumes teacher-forced alignment during training, the loss can be extended to accommodate mismatched trajectories via alignment-based matching, as detailed in Appendix H.

3.4 Optimization View: Gradient Projection and Span Decoupling

To better understand the functional benefit of SAD, we present an *optimization-based interpretation* of how span-specific KL losses reshape the gradient landscape. Rather than assuming a cognitive or semantic split, SAD introduces an optimization structure that avoids interference between reasoning and action supervision signals.

As illustrated in Figure 2, a token-level KL couples heterogeneous gradients into a single update direction (gray arrow), creating a conflict angle θ between reasoning and action forces in parameter space. SAD reformulates this process as an orthogonal gradient projection: reasoning and action components are separately normalized in their respective subspaces ($\mathcal{V}_r, \mathcal{V}_a$) and then recombined geometrically. This projection removes cross-span interference and yields span-specific updates, altering the overall optimization geometry.

Let $\mathcal{V}_r$ and $\mathcal{V}_a$ denote the token domains for reasoning and action, along with $\mathcal{V}_r \cap \mathcal{V}_a = \varnothing$. A standard token-level KL minimizes

$$\nabla_\theta \mathcal{L}_{\text{token}} = \nabla_\theta \, \text{KL}(p_T(\cdot \mid x_{<t}) \,\|\, p_S(\cdot \mid x_{<t})), \tag{7}$$

where normalization over the entire vocabulary $\mathcal{V}_r \cup \mathcal{V}_a$ forces shared probability mass between semantically incompatible tokens. This coupling biases gradients toward frequent reasoning tokens and suppresses rare but task-critical action tokens, explaining why a single distribution cannot disentangle these behaviors.

The proposed SAD resolves this by restricting the KL computation to disjoint subspaces:

$$\nabla_\theta \mathcal{L}_{\text{SAD}} = m_r(t)\, \nabla_\theta \text{KL}_{\mathcal{V}_r}(p_T \,\|\, p_S) + m_a(t)\, \nabla_\theta \text{KL}_{\mathcal{V}_a}(p_T \,\|\, p_S), \tag{8}$$

which performs *gradient projection* onto reasoning and action subspaces,

$$\nabla_\theta \mathcal{L}_{\text{SAD}} = \Pi_{\text{reason}}(\nabla_\theta \mathcal{L}_{\text{CoT}}) + \Pi_{\text{action}}(\nabla_\theta \mathcal{L}_{\text{Act}}). \tag{9}$$

This projection changes both the normalization domain and gradient direction, eliminating cross-span interference and yielding span-specific updates. **Hence, the distinction from token-level KL is geometric, not cosmetic:** SAD introduces a structure-aware gradient decomposition rather than simply applying the KL divergence to a smaller subset of tokens.

3.5 Multi-Step ReAct Trajectories and Multi-Span Masks

Trajectory Model. Extending the single-step formulation in Eq. (1), we generalize SAD to multi-step ReAct episodes composed of alternating reasoning, action, and observation segments:

$$\tau = \left[(r^{(1)}, a^{(1)}, o^{(1)}), (r^{(2)}, a^{(2)}, o^{(2)}), \ldots, (r^{(K)}, a^{(K)}, o^{(K)})\right], \tag{10}$$

where $r^{(i)}$, $a^{(i)}$, and $o^{(i)}$ denote the reasoning trace, executed action, and subsequent observation at step i. Linearization yields

$$\tau' = \prod_{i=1}^{K} \left(\texttt{[REASON]}\, r^{(i)}\, \texttt{[ACT]}\, a^{(i)}\, \texttt{[OBS]}\, o^{(i)}\right), \qquad x = \texttt{Tokenize}(\tau').$$

This formulation extends SAD beyond the single-step assumption, enabling structured supervision across multiple reasoning–action cycles.

Segment-Aware Mask Construction. Each token x_t in the sequence is assigned a binary membership mask:

$$\begin{aligned} m_r(t) &= \mathbf{1}[x_t \in \cup_i r^{(i)}], m_a(t) \quad = \mathbf{1}[x_t \in \cup_i a^{(i)}], \\ m_o(t) &= \mathbf{1}[x_t \in \cup_i o^{(i)}], \end{aligned} \tag{11}$$

We enforce an *exactly-one* constraint for every token:

$$m_r(t) + m_a(t) + m_o(t) = 1, \quad \forall t \in [1, T], \tag{12}$$

ensuring non-overlapping and functionally disjoint spans. Reasoning and action masks (m_r, m_a) are used for supervision, while observation masks (m_o) indicate environmental feedback.

Observation Handling. Each token is assigned to *exactly one* of the three functional categories—reasoning, action, or observation—ensuring disjoint span boundaries. However, only reasoning and action tokens contribute to the distillation loss. Observation

tokens ($m_o(t) = 1$) are excluded from the distillation loss, as they encode deterministic feedback from the environment rather than agent behavior. This exclusion prevents the student from overfitting to static textual observations and focuses learning on reasoning and decision quality. Nonetheless, the framework permits optional extensions: (1) adding an auxiliary cross-entropy term for perceptual grounding, or (2) defining a separate observation head for multimodal tasks. All reported experiments adopt the exclusion setting.

Supervision over Multi-Span Masks. Structured losses defined in Section 3.3 (Eq. (4) and (5)) are applied independently to each reasoning and action span, as detailed in Appendix L. with m_r and m_a computed as the union of all `[REASON]` and `[ACT]` segments. This union-mask construction generalizes SAD to multi-turn reasoning–acting trajectories, preserving disjoint functional roles and preventing cross-span gradient interference.

Illustrative Example. A two-step episode (K=2) from ALFWorld:

> `[REASON]` I will first check the table. `[ACT]` search[tray] `[OBS]` You see a tray.
> `[REASON]` Now I will pick it up. `[ACT]` pickup[tray] `[OBS]` Tray in inventory.

Here, reasoning tokens correspond to m_r=1, action tokens to m_a=1, and observation tokens to m_o=1. Only reasoning and action tokens receive gradient updates, clarifying span-level supervision semantics and aligning with multi-turn ReAct agent behavior.

3.6 Optimization Analysis and Intuition

Why It Works in Practice. The decoupled objectives alter both gradient magnitude and direction across spans. Empirically, SAD reduces step-to-step gradient variance and improves training stability under limited data. This acts as an *implicit curriculum*: reasoning spans provide dense signals for high-level coherence, while action spans yield sparse but decisive grounding signals. The balanced supervision accelerates convergence and strengthens reasoning–action compositionality.

SAD provides a *principled change in optimization geometry* that (1) moves beyond simplified cognitive assumptions, (2) explains why a single token-level distribution fails to separate reasoning and action, and (3) establishes a structure-aware, theoretically grounded distillation objective.

3.7 Semantic Decoupling and Example

While the overall loss is additive in form, our supervision is fundamentally different from flat token-level imitation. We explicitly decompose the learning signal into structurally disjoint spans—`[REASON]` and `[ACT]`—and apply segment-specific losses to each, preserving the semantics of multi-phase agent behavior.

CoT-Policy Alignment Loss ($\mathcal{L}_{\text{CoT}}$) supervises predictions within the reasoning span, promoting coherent multi-step inference aligned with the teacher's thought patterns. **Action Consistency Loss** ($\mathcal{L}_{\text{Act}}$) applies only to the action span, enforcing the accurate replication of grounded decisions.

Each token is assigned to exactly one functional span using binary masks $\{m_r(t), m_a(t)\}$, which gate gradient flow. This masking enforces *semantic separation* during training, ensuring that the student independently learns high-level reasoning and low-level execution. Unlike token-level KL with soft targets, our structure-aware formulation avoids loss interference across phases, better modeling causal dependencies (reason → act). Ablations confirm that removing either component harms performance.

Example.

> **Instruction:** `"Find the tray"`
> **Teacher:** `[REASON]` `"I will first look on the table to check for a tray..."` → `[ACT]` `search[tray]`
> **Student:** `[REASON]` `"Maybe it's on the shelf, I should check there."` → `[ACT]` `search[tray]`

Although the student executes the correct action, their reasoning deviates from the teacher's thought process. $\mathcal{L}_{\text{CoT}}$ penalizes semantic deviations within `[REASON]` via KL divergence between teacher and student token distributions, producing gradients that align multi-step reasoning. $\mathcal{L}_{\text{Act}}$ rewards correct predictions in `[ACT]`, allowing action alignment even when reasoning differs.

Algorithm 1 Structured Agent Distillation (SAD)

```
Initialize teacher policy π_T, student policy π_θ, and curriculum scheduler C
for epoch = 1 to E do
    Sample multi-step trajectory τ = {(r^(i), a^(i), o^(i))}_{i=1}^K ~ C
    Linearize and tokenize τ' = [REASON] r^(i) [ACT] a^(i) [OBS] o^(i)
    Construct binary masks m_r, m_a, m_o over tokens x_{1:T}
    Forward pass: obtain student logits p_S(· | x_{<t})
    Compute losses L_CoT, L_Act
    Aggregate weighted total loss L_total = λ_r L_CoT + λ_a L_Act
    Update parameters θ ← θ − η ∇_θ L_total
end for
```

3.8 Curriculum Sampling in Structured Distillation

To further enhance learning efficiency and stability, we employ curriculum learning [1, 11] based on a trajectory complexity score:

$$C(\tau) = \alpha \cdot \text{len}(r_{1:k}) + \beta \cdot \text{len}(a_{1:m}) + \gamma \cdot \text{entropy}(\pi_T(\tau)), \quad (13)$$

where $\text{len}(r_{1:k})$ and $\text{len}(a_{1:m})$ denote the lengths of the reasoning and action segments, respectively, and $\text{entropy}(\pi_T(\tau))$ reflects teacher uncertainty. The weights α, β, γ balance their relative contributions. During training, trajectories are sorted by $C(\tau)$, allowing the model to start with examples and gradually progress to more complex ones. Detailed analysis in Appendix N.

3.9 Training Algorithm

Algorithm 1 outlines our structured agent distillation process. The student policy π_θ learns to imitate the teacher π_T across reasoning stages, guided by a curriculum scheduler C that samples increasingly complex trajectories. Each trajectory is tokenized into reasoning and action spans, and the student predicts tokens autoregressively. The objective aggregates reasoning and action losses, and gradients are backpropagated to update θ.

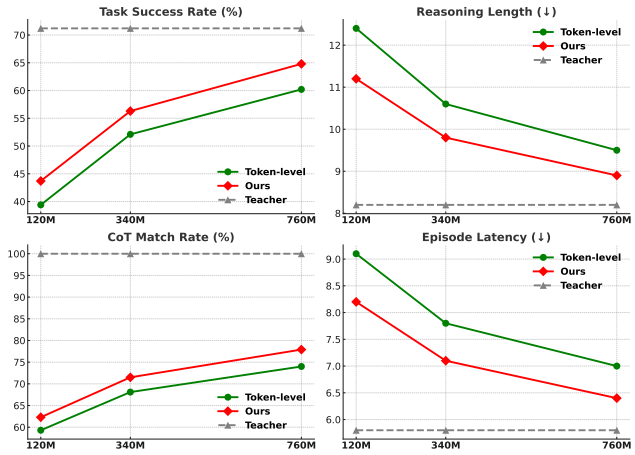

Figure 3: Scaling behavior of student agents across model sizes. Top-Left: Task Success Rate (%). Top-Right: Reasoning Length (tokens). Bottom-Left: Chain-of-Thought (CoT) Match Rate (%). Bottom-Right: Episode Latency (steps). Structured Agent Distillation consistently outperforms the best baseline method (Token-level [10]) and better approaches teacher performance as model capacity increases.

4 EXPERIMENTS

4.1 Experimental Setups

Agent Environments. We evaluate on three representative benchmarks: ① Embodied benchmark ALFWorld [55] for embodied instruction following, ② Web benchmark WebShop [74] towards scalable real-world web interaction with grounded language agents, and ③ multi-hop question benchmarks HotPotQA-ReAct [75] for multi-hop QA with reasoning traces. ALFWorld and WebShop test structured decision-making, while HotPotQA-ReAct requires open-ended multi-hop reasoning and free-form answer generation—thus already covering both discrete and natural-language modalities. All reported results are averaged over 5 independent runs.

Baselines. The baselines include token level KD [10], word-level KD [43, 56], and SeqKD [21, 38, 59, 87].

Dataset Statistics. We adopt standard splits from existing ReAct-based benchmarks. ALFWorld comprises 8,055 instruction-following trajectories (5,400 train / 1,200 val / 1,475 test). WebShop contains 12,000 web interaction samples (8,000 / 2,000 / 2,000), and HotPotQA-ReAct includes 90,447 multi-hop reasoning examples (84,000 / 3,447 / 3,000). The details of Experimental Setups in Appendix B.

4.2 Experimental Results and Analysis

We evaluate student agent performance across three benchmarks—ALFWorld, WebShop, and HotPotQA-ReAct—comparing our proposed Structured Agent Distillation with the baseline. We report results across three evaluation metrics: task success rate, reasoning efficiency, and CoT consistency (Table 3).

Task Success Rate. As shown in Table 3, Structured Agent Distillation consistently outperforms token-level MiniLLM baselines across all student sizes, with especially notable gains at 120M (+4.3%), confirming the effectiveness of trajectory-level supervision over token imitation.

Reasoning Efficiency. Table 3 shows that students trained via Structured Agent Distillation generate shorter reasoning spans.

CoT Consistency (defined in Appendix C). As shown in Table 3, our method achieves higher CoT match rates across all settings, demonstrating stronger structural alignment with teacher reasoning.

Latency. Following prior work [54, 75], we measure the average number of reasoning and action steps per episode. As shown in Table 3, **segment-aware** students consistently exhibit shorter execution traces. Latency is measured in reasoning/action steps rather than wall-clock time, and lower values indicate more concise and efficient decision-making.

Segment Mask Validation. Figure 8 (Appendix F) confirms that token-level masks align accurately with reasoning/action spans across environments.

Span Statistics. As shown in Figure 9, reasoning spans are longer and more variable, while action spans are shorter—justifying span-specific supervision.

Our method outperforms token-level distillation across all benchmarks (Table 4), yielding more accurate and faithful student agents.

5 SCALING ANALYSIS

To assess scalability, we transfer trajectories from a GPT-2-1.5B teacher model into student models with 120M, 340M, and 760M parameters using Structured Agent Distillation, and compare the outcomes against the best-performing baseline method (token-level [10]). Figure 3 summarizes four metrics—task success rate, reasoning efficiency (avg. reasoning length), CoT match rate, and episode latency—on ALFWorld, WebShop, and HotPotQA-ReAct.

Task Success. Success rates improve with model size (top-left), with students trained via Structured Agent Distillation consistently outperforming token-level baselines. At 760M, performance closely approaches the teacher.

Reasoning Efficiency. Students trained via Structured Agent Distillation produce shorter, more efficient reasoning traces (top-right), especially at larger scales.

CoT Match. Students distilled with our method better recover the teacher's reasoning structure (bottom-left), with consistently higher CoT match rates.

Latency. Structured supervision yields lower episode latency (bottom-right), reducing decision steps and accelerating task completion.

Structured Agent Distillation scales effectively with student capacity, enhancing task success, planning efficiency, and structural reasoning alignment. Improvements are most pronounced at smaller scales (e.g., 120M, 340M), where it mitigates the performance degradation commonly observed with token-level imitation. Additional scaling results for OPT and LLaMA models are presented in Appendix G.

6 ABLATION STUDIES

We conduct ablation studies to understand the contribution of each component in our Structured Agent Distillation framework.

Ablation on Reasoning, Action, and Segmentation Components Specifically, we analyze the roles of reasoning supervision, action supervision, and span segmentation. Table 5 confirms that each supervision component plays a critical role:

Table 3: Unified comparison of task success (↑), reasoning length (↓), CoT match (↑), and episode latency (steps) (↓) across ALFWorld, WebShop, and HotPotQA. The teacher is a GPT-2-1.5B ReAct-style agent. Students trained via Structured Agent Distillation consistently outperform token-level KD [10], KD [43, 56], and SeqKD [2, 21, 38, 59, 87] baselines.

Method	Task Success ↑			Reasoning Length ↓			CoT Match Rate ↑			Episode Latency ↓		
	ALF	Web	Hot	ALF	Web	Hot	ALF	Web	Hot	ALF	Web	Hot
Teacher(GPT)	71.2	68.7	78.5	8.2	11.5	10.8	100.0	100.0	100.0	5.8	7.4	6.2
KD-120M	37.3	34.9	46.1	13.2	16.4	15.3	57.1	52.7	63.4	9.3	10.9	9.2
SeqKD-120M	38.5	36.0	47.2	12.9	16.0	15.0	58.3	54.0	64.2	9.2	10.8	9.0
Token-120M	39.4	36.7	48.3	12.4	15.7	14.8	59.3	55.1	65.7	9.1	10.7	8.9
Ours (120M)	**43.7**	**41.2**	**52.8**	**11.2**	**14.6**	**13.8**	**62.3**	**58.7**	**66.2**	**8.2**	**9.5**	**7.8**
KD-340M	49.3	47.1	58.8	11.1	14.9	13.8	66.0	60.5	68.3	8.0	9.6	8.2
SeqKD-340M	50.7	48.6	60.2	10.9	14.5	13.4	67.2	61.4	69.3	7.9	9.5	8.1
Token-340M	52.1	49.8	61.5	10.6	14.1	13.0	68.1	62.4	70.5	7.8	9.4	7.9
Ours (340M)	**56.3**	**54.7**	**65.5**	**9.8**	**13.1**	**12.2**	**71.5**	**66.9**	**74.0**	**7.1**	**8.7**	**7.0**
KD-760M	57.5	54.3	66.2	10.1	13.7	12.6	72.0	67.2	73.1	7.2	8.9	7.4
SeqKD-760M	58.7	55.5	67.4	9.8	13.4	12.3	73.1	68.3	74.3	7.1	8.8	7.3
Token-760M	60.2	57.0	69.1	9.5	13.2	12.0	74.0	69.3	76.4	7.0	8.6	7.2
Ours (760M)	**64.8**	**61.5**	**73.1**	**8.9**	**12.4**	**11.7**	**77.9**	**73.1**	**80.4**	**6.4**	**8.1**	**6.6**

Table 4: Unified comparison of task success (↑), reasoning length (↓), CoT match (↑), and episode latency (↓) across ALFWorld, WebShop, and HotPotQA. The teachers are OPT-13B, LLaMA-13B, and Orca2-13B ReAct-style agents. Students trained via Structured Agent Distillation consistently outperform token-level [10], KD [43, 56], and SeqKD [2, 21, 38, 60, 87] baselines.

Model	Task Success ↑			Reasoning Length ↓			CoT Match Rate↑			Episode Latency ↓		
	ALF	Web	Hot	ALF	Web	Hot	ALF	Web	Hot	ALF	Web	Hot
Teacher (OPT13B)	76.5	73.2	82.7	38.2	35.9	40.7	100.0	100.0	100.0	6.5	5.9	4.8
KD (OPT-1.3B)	45.3	40.7	51.0	47.6	43.9	50.1	58.9	55.0	67.3	8.1	7.3	6.3
SeqKD (OPT-1.3B)	46.2	41.8	52.3	46.3	42.5	49.4	60.2	56.1	68.4	7.9	7.2	6.2
Token-OPT-1.3B	47.8	43.2	54.1	45.7	41.8	48.5	61.5	57.9	69.8	7.8	7.1	6.0
Ours (OPT-1.3B)	**52.3**	**48.7**	**58.5**	**41.2**	**38.0**	**43.6**	**67.2**	**63.8**	**74.4**	**7.0**	**6.4**	**5.3**
KD (OPT-2.7B)	53.1	48.3	60.7	44.3	40.1	46.5	65.1	61.0	73.4	7.5	6.9	5.7
SeqKD (OPT-2.7B)	54.4	49.7	61.3	43.2	39.5	46.0	66.2	61.9	74.1	7.3	6.7	5.6
Token-OPT-2.7B	55.6	51.0	62.9	42.5	39.0	45.2	67.3	62.7	75.5	7.2	6.6	5.6
Ours (OPT-2.7B)	**59.2**	**56.4**	**67.0**	**39.4**	**36.2**	**41.7**	**71.6**	**67.9**	**79.8**	**6.7**	**6.1**	**5.0**
KD (OPT-6.7B)	60.1	55.8	67.2	42.1	38.4	43.9	70.1	66.5	78.1	7.0	6.4	5.4
SeqKD (OPT-6.7B)	61.3	56.9	68.1	41.4	37.8	43.3	71.4	67.0	79.0	6.9	6.3	5.4
Token-OPT-6.7B	62.8	58.6	69.7	40.8	37.2	42.9	72.2	67.9	80.2	6.8	6.2	5.3
Ours (OPT-6.7B)	**67.1**	**63.8**	**73.9**	**38.0**	**35.1**	**40.2**	**76.4**	**72.5**	**84.0**	**6.5**	**5.9**	**4.9**
Teacher (LLaMA13B)	75.3	71.8	81.0	37.5	34.8	39.9	100.0	100.0	100.0	6.4	5.8	4.7
KD (LLaMA-7B)	62.1	56.9	70.1	42.5	38.3	44.0	71.2	66.0	79.0	6.8	6.3	5.3
SeqKD (LLaMA-7B)	63.0	58.1	70.9	41.7	37.9	43.6	72.5	67.1	80.0	6.8	6.2	5.2
Token-LLaMA-7B	64.2	59.3	71.5	41.1	37.5	43.2	73.0	68.3	81.2	6.7	6.1	5.2
Ours (LLaMA-7B)	**68.0**	**64.1**	**75.2**	**38.2**	**34.9**	**39.8**	**77.2**	**72.9**	**84.7**	**6.4**	**5.8**	**4.8**
Teacher (Orca2-13B)	78.1	75.6	84.3	37.0	34.6	39.1	100.0	100.0	100.0	6.3	5.8	4.6
KD (Orca2-7B)	64.0	59.2	72.4	41.6	37.8	43.1	73.1	68.7	81.5	6.7	6.2	5.2
SeqKD (Orca2-7B)	65.2	60.4	73.5	40.9	37.2	42.5	74.3	69.4	82.6	6.6	6.1	5.1
Token-Orca2-7B	66.3	61.7	74.8	40.3	36.9	42.0	75.6	70.2	83.4	6.6	6.0	5.1
Ours (Orca2-7B)	**70.5**	**66.2**	**78.6**	**37.8**	**34.2**	**38.9**	**79.4**	**74.6**	**86.5**	**6.3**	**5.8**	**4.7**

(1) Removing reasoning supervision ($\mathcal{L}_{\text{CoT}}$) significantly degrades CoT matches and increases latency, indicating reduced planning coherence.
(2) Removing action supervision ($\mathcal{L}_{\text{Act}}$) lowers task success and execution fidelity, showing the importance of behavior alignment.
(3) Disabling span segmentation (flat token-level loss) causes uniform degradation across all metrics, suggesting that structural decomposition is essential.

Table 5: Ablation study on ALFWorld using a 340M-parameter student model. Each variant controls whether reasoning, action supervision, and explicit span segmentation are enabled. We report task success rate, CoT match rate, and episode latency(steps).

Method	Reason	Action	Segm	Succ (%) ↑	CoT (%) ↑	Episode Latency ↓
Full Segment-Aware (Ours)	✓	✓	✓	**56.3**	**71.5**	**7.1**
Only Reasoning Supervision	✓	✗	✓	52.7	69.2	7.4
Only Action Supervision	✗	✓	✓	49.5	61.8	7.8
No Span Segmentation	✓	✓	✗	48.2	60.4	8.1
Random Span Masking	✓	✓	~ (random)	45.9	57.7	8.4

Table 6: Ablation study on ALFWorld using a 340M-parameter student model. We vary the loss weight ratios between reasoning (CoT) and action supervision while keeping their sum fixed. Structured Agent Distillation consistently outperforms token-level baselines across all settings.

CoT : Act Ratio	Succ (%) ↑	CoT (%) ↑	Episode Latency ↓
1.0 : 0.0	53.4	69.2	8.2
0.0 : 1.0	52.7	66.1	8.3
0.5 : 1.0	55.8	70.3	7.7
1.0 : 1.0	**56.3**	**71.5**	**7.1**
2.0 : 1.0	56.1	70.8	7.3
Token-Level Baseline	52.1	68.1	8.6

(4) Random span masking further harms performance, highlighting the need for accurate, semantically meaningful segmentation.

These results demonstrate that our method captures structurally distinct signals crucial for reasoning-action alignment.

Ablation on λ_r:λ_a Ratio. We conduct an ablation to examine the balance between reasoning and action supervision by varying the weighting ratio between λ_{CoT} and λ_{Act} while keeping the total loss weight fixed. As shown in Table 6, combining both losses yields the best performance, with a balanced 1:1 ratio achieving the highest success rate, CoT match, and shortest episode length. Even when only one supervision term is used, SAD still outperforms token-level baselines, indicating that reasoning and action signals are complementary yet individually beneficial.

Ablation study on curriculum sampling in Appendix N, Qualitative Examples and Faithfulness in Appendix O.

7 DISCUSSION

Semantic Decoupling Matters. Token-level imitation fails to preserve reasoning–action structure. By segmenting trajectories and supervising each span separately, Structured Agent Distillation enables independent learning of reasoning and execution, which is especially critical under limited capacity.

Loss Design. Although the total loss combines $\mathcal{L}_{\text{CoT}}$ and $\mathcal{L}_{\text{Act}}$, removing either term degrades CoT alignment, task success, and efficiency, confirming their complementary roles beyond token-level KL.

Segmentation Robustness. Our rule-based extraction of `[REASON]` and `[ACT]` spans is lightweight and reliable across benchmarks, requiring no manual annotation while consistently improving student performance.

Scalability. We use 13B teachers (OPT, LLaMA, Orca2) due to resource limitations. SAD is model- and scale-agnostic, operating independently of architecture or tokenizer. It naturally extends to frontier models (e.g., Qwen3-235B-A22B, DeepSeek-V3.1), enabling 235B→13B/7B compression under the same objective.

8 RELATED WORK

LLM Agents. LLM agents unify reasoning and action. ReAct [75] interleaves language and actions. Toolformer [44] self-supervises API calls, while WebGPT [36] reasons via web browsing. AutoGPT [41] chains subtasks autonomously. AgentBench [34], ReWoo [70], and HuggingGPT [53] explore modularity and tool use. CAMEL [26], ChatDev [39], AutoGen [68], CrewAI [5], ToolBench [12], and LangChain [15] focus on multi-agent collaboration and LLM orchestration. GITM [90] integrates LLMs with memory and knowledge to build generally capable agents. Although these agents show strong reasoning-action abilities, their trajectories remain challenging to compress and generalize.

Distillation and Fine-Tuning. Token-level distillation [6, 9, 10, 14, 28, 29, 35, 49, 50, 69, 77, 82] compresses LLMs via soft target alignment and teacher-guided training. Recent work improves multi-granular supervision [10, 27, 33], cross-modal transfer [32, 89], and compact pretraining [30, 58, 80]. However, these methods target static text generation, not structured reasoning-action trajectories.

Trajectory Modeling and Behavioral Cloning. Recent work in sequence-level imitation [7, 20, 25, 46, 57, 65, 72, 73, 76, 81, 83], behavior cloning [3, 13, 31, 42, 45, 47, 61, 67, 71, 88] emphasizes temporal coherence in agent behavior. SpanBERT [19] masks contiguous spans instead of individual tokens, and diffusion planning [16] ConvBERT [17] uses span-based convolutions instead of attention heads. LLM agents require structured objectives to model reasoning and action jointly—captured by our Structured Agent Distillation framework.

9 CONCLUSION

We propose **Structured Agent Distillation**, a compression framework that segments teacher trajectories into reasoning and action spans, allowing students to better replicate high-level reasoning and low-level execution beyond token-level imitation. Our method achieves consistent gains in task success, reasoning efficiency, and CoT alignment over token-level baselines. Scaling and ablation results confirm the benefits of structured supervision under limited capacity. These findings highlight the importance of preserving trajectory structure for training lightweight agents and open avenues for structured knowledge transfer in real-world decision making.

ACKNOWLEDGMENTS

This work is partly supported by the National Science Foundation DRL-2507128.

REFERENCES

[1] Yoshua Bengio, Jean Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML 2009)*. ACM, New York, NY, USA, 41–48. https://dl.acm.org/doi/10.1145/1553374.1553380

[2] Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* 2, 3 (2023), 6.

[3] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. 2018. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, IEEE, Piscataway, NJ, USA, 4693–4700.

[4] Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons, New York.

[5] CrewAI Inc. 2023. CrewAI: Framework for Orchestrating Role-Playing, Autonomous AI Agents. https://github.com/crewAIInc/crewAI. Accessed: 2025-04-15.

[6] Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. 2025. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. AAAI Press, Palo Alto, CA, 23724–23732.

[7] Chris Cundy and Stefano Ermon. 2023. Sequencematch: Imitation learning for autoregressive sequence modelling with backtracking. *arXiv preprint arXiv:2306.05426* 1 (2023).

[8] Marco Cuturi and Mathieu Blondel. 2017. Soft-DTW: a Differentiable Loss Function for Time-Series. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, Brooklyn, NY, USA, 894–903. https://proceedings.mlr.press/v70/cuturi17a.html

[9] Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How Small Can Language Models Be and Still Speak Coherent English? *arXiv preprint arXiv:2305.07759* 1 (2023), 1–10.

[10] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge Distillation of Large Language Models. In *The Twelfth International Conference on Learning Representations*. OpenReview, Addis Ababa, Ethiopia. https://openreview.net/forum?id=5h0qf7IBZZ

[11] Sheng Guo, Weilin Huang, Haozhi Zhang, et al. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European conference on computer vision (ECCV)*. Springer, Munich, Germany, 135–150.

[12] Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. 2024. StableToolBench: Towards Stable Large-Scale Benchmarking on Tool Learning of Large Language Models. arXiv:2403.07714 [cs.CL]

[13] Charles A Hepburn and Giovanni Montana. 2024. Model-based trajectory stitching for improved behavioural cloning and its applications. *Machine Learning* 113, 2 (2024), 647–674.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* 1, 1 (2015), 1–9.

[15] LangChain Inc. 2022. LangChain Documentation. https://python.langchain.com/. Accessed: 2025-04-15.

[16] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. 2022. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991* abs/2205.09991 (2022), 1–10.

[17] Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems* 33 (2020), 12837–12848.

[18] Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. *arXiv preprint arXiv:2401.04925* 1, 1 (2024).

[19] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics* 8 (2020), 64–77.

[20] Liyiming Ke, Yunchu Zhang, Abhay Deshpande, Siddhartha Srinivasa, and Abhishek Gupta. 2023. CCIL: Continuity-based data augmentation for corrective imitation learning. *arXiv preprint arXiv:2310.12972* 1, 1 (2023).

[21] Yoon Kim and Alexander M Rush. 2016. Sequence-Level Knowledge Distillation. In *EMNLP*. Association for Computational Linguistics, Austin, TX, 1317–1327.

[22] Zhenglun Kong, Yize Li, Fanhu Zeng, Lei Xin, Shvat Messica, Xue Lin, Pu Zhao, Manolis Kellis, Hao Tang, and Marinka Zitnik. 2025. Token Reduction Should Go Beyond Efficiency in Generative Models–From Vision, Language to Multimodality. *arXiv preprint arXiv:2505.18227* 1 (2025).

[23] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86. https://doi.org/10.1214/aoms/1177729694

[24] M Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems* 23 (2010), 1234–1242.

[25] Hoang Le, Andrew Kang, Yisong Yue, and Peter Carr. 2016. Smooth imitation learning for online sequence prediction. In *International Conference on Machine Learning*. PMLR, PMLR, New York, NY, USA, 680–688.

[26] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, et al. 2023. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In *Thirty-seventh Conference on Neural Information Processing Systems*. NeurIPS, New Orleans, LA, USA.

[27] Sheng Li, Qitao Tan, Yue Dai, Zhenglun Kong, Tianyu Wang, Jun Liu, Ao Li, Ninghao Liu, Yufei Ding, Xulong Tang, et al. 2025. Mutual Effort for Efficiency: A Similarity-based Token Pruning for Vision Transformers in Self-Supervised Learning. In *The Thirteenth International Conference on Learning Representations*. ICLR, Singapore.

[28] Yanyu Li, Changdi Yang, Pu Zhao, Geng Yuan, et al. 2023. Towards real-time segmentation on the edge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. AAAI, Washington, DC.

[29] Yanyu Li, Pu Zhao, et al. 2022. Pruning-as-search: Efficient neural architecture search via channel pruning and structural reparameterization. *International Joint Conference on Artificial Intelligence (IJCAI-22)* 1, 1 (2022), 1.

[30] Jun Liu, Zhenglun Kong, Peiyan Dong, Xuan Shen, Pu Zhao, Hao Tang, et al. 2025. Rora: Efficient fine-tuning of llm with reliability optimization for rank adaptation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, IEEE, Hyderabad, Inda, 1–5.

[31] Jun Liu, Zhenglun Kong, Changdi Yang, Fan Yang, Tianqi Li, Peiyan Dong, Joannah Nanjekye, Hao Tang, Geng Yuan, Wei Niu, et al. 2025. Rcr-router: Efficient role-aware context routing for multi-agent llm systems with structured memory. *arXiv preprint arXiv:2508.04903* 1, 1 (2025), 1.

[32] Jun Liu, Zhenglun Kong, Pu Zhao, et al. 2024. Tsla: A task-specific learning adaptation for semantic segmentation on autonomous vehicles platform. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 44, 4 (2024), 1406–1419.

[33] Jun Liu, Zhenglun Kong, Pu Zhao, Changdi Yang, et al. 2025. Toward adaptive large language models structured pruning via hybrid-grained weight importance assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. AAAI Press, Philadelphia, Pennsylvania, USA, 18879–18887. https://doi.org/10.1609/aaai.v39i18.34078

[34] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688* 1, 1 (2023).

[35] Benjamin Minixhofer, Edoardo Maria Ponti, and Ivan Vulić. 2025. Cross-Tokenizer Distillation via Approximate Likelihood Matching. *arXiv preprint arXiv:2503.20083* 1, 1 (2025).

[36] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* abs, 2112.09332 (2021), 1–10.

[37] Simon Ott, Konstantin Hebenstreit, Valentin Liévin, Christoffer Egeberg Hother, Milad Moradi, Maximilian Mayrhauser, Robert Praas, Ole Winther, and Matthias Samwald. 2023. ThoughtSource: A central hub for large language model reasoning data. *Scientific data* 10, 1 (2023), 528.

[38] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277* 1, 1 (2023), 1–10.

[39] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. ChatDev: Communicative Agents for Software Development. *arXiv preprint arXiv:2307.07924* 1, 1 (2023). https://arxiv.org/abs/2307.07924

[40] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. 2022. A Generalist Agent. *Transactions on Machine Learning Research* 1 (2022), 1–10.

[41] Toran Bruce Richards and Significant Gravitas. 2023. AutoGPT: An experimental open-source attempt to make GPT-4 autonomous. https://github.com/Significant-Gravitas/AutoGPT. Accessed: 2025-04-15.

[42] Zachary W Robertson and Matthew R Walter. 2020. Concurrent training improves the performance of behavioral cloning from observation. *arXiv preprint arXiv:2008.01205* 1 (2020).

[43] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* 1, 1 (2019). https://arxiv.org/pdf/1910.01108.pdf

[44] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, et al. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems* 36 (2023), 68539–68551.

[45] Xuan Shen, Peiyan Dong, Zhenglun Kong, Yifan Gong, Changdi Yang, Han, et al. 2025. Squat: Quant Small Language Models on the Edge. In *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, IEEE, Munich, Germany, 1–9.

[46] Xuan Shen, Chenxia Han, Yufa Zhou, Yanyue Xie, Yifan Gong, Quanyi Wang, Yiwei Wang, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. 2025. DraftAttention: Fast Video Diffusion via Low-Resolution Attention Guidance. *arXiv preprint arXiv:2505.14708* 1, 1 (2025), 1–10.

[47] Xuan Shen, Weize Ma, et al. 2026. Fastcar: Cache Attentive Replay for Fast Auto-Regressive Video Generation on the Edge. In *The Fourteenth International Conference on Learning Representations*. ICLR, Rio de Janeiro, Brazil, 1–10. https://openreview.net/forum?id=9f3Nukn6BA

[48] Xuan Shen, Weize Ma, Jing Liu, et al. 2025. QuartDepth: Post-Training Quantization for Real-Time Depth Estimation on the Edge. In *CVPR*. IEEE, New Orleans, LA, USA.

[49] Xuan Shen, Zhao Song, Yufa Zhou, et al. 2025. Lazydit: Lazy learning for the acceleration of diffusion transformers. In *AAAI*. AAAI Press, Philadelphia, Pennsylvania.

[50] Xuan Shen, Zhao Song, Yufa Zhou, Bo Chen, et al. 2025. Numerical Pruning for Efficient Autoregressive Models. *Proceedings of the AAAI Conference on Artificial Intelligence* 39, 19 (2025), 20418–20426. https://doi.org/10.1609/aaai.v39i19.34249

[51] Xuan Shen, Pu Zhao, Yifan Gong, Zhenglun Kong, et al. 2024. Search for Efficient Large Language Models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Red Hook, NY, USA.

[52] Xuan Shen, Hangyu Zheng, Yifan Gong, et al. 2025. Sparse Learning for State Space Models on Mobile. In *ICLR*. OpenReview, Singapore, 1. https://openreview.net/forum?id=t8KLjiFNwn

[53] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems* 36 (2023), 38154–38180.

[54] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2023), 8634–8652.

[55] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR, Virtual Only Conference, 1–10. https://arxiv.org/abs/2010.03768

[56] Kaitao Song, Hao Sun, Xu Tan, Tao Qin, Jianfeng Lu, Hongzhi Liu, and Tie-Yan Liu. 2020. LightPAFF: A two-stage distillation framework for pre-training and fine-tuning. *arXiv preprint arXiv:2004.12817* N/A, N/A (2020), –. https://arxiv.org/pdf/2004.12817.pdf

[57] Gokul Swamy, Sanjiban Choudhury, J Bagnell, and Steven Z Wu. 2022. Sequence model imitation learning with unobserved contexts. *Advances in Neural Information Processing Systems* 35 (2022), 17665–17676.

[58] Qitao Tan, Jun Liu, Zheng Zhan, Caiwei Ding, Yanzhi Wang, Xiaolong Ma, Jaewoo Lee, Jin Lu, and Geng Yuan. 2025. Harmony in divergence: Towards fast, accurate, and memory-efficient zeroth-order llm fine-tuning. *arXiv preprint arXiv:2502.03304* 1, 1 (2025).

[59] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

[60] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[61] Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954* 1, 1 (2018).

[62] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-Read Students Learn Better: The Impact of Student Initialization on Knowledge Distillation. *arXiv preprint arXiv:1908.08962* 1, 1 (2019), 1–10.

[63] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* abs/2305.16291, arXiv (2023), 1–10.

[64] Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. Scott: Self-consistent chain-of-thought distillation. *arXiv preprint arXiv:2305.01879* 1, 1 (2023), 1.

[65] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* 1, 1 (2022), 1–10. arXiv preprint.

[66] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[67] Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. 2020. Fighting copycat agents in behavioral cloning from observation histories. *NeurIPS* 33 (2020), 2564–2575.

[68] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, et al. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155* 1, 1 (2023).

[69] Yushu Wu, Yifan Gong, Pu Zhao, et al. 2022. Compiler-aware neural architecture search for on-mobile real-time super-resolution. In *European Conference on Computer Vision*. Springer, Munich, Germany, 92–111.

[70] Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. 2023. Rewoo: Decoupling reasoning from observations for efficient augmented language models. *arXiv preprint arXiv:2305.18323* 1, 1 (2023).

[71] Changdi Yang, Zheng Zhan, Ci Zhang, Yifan Gong, Jun Liu, et al. 2025. FairSMOE: Mitigating Multi-Attribute Fairness Problem with Sparse Mixture-of-Experts. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence*. IJCAI, Montreal, Canada.

[72] Changdi Yang, Pu Zhao, Yanyu Li, et al. 2023. Pruning parameterization with bi-level optimization for efficient semantic segmentation on the edge. In *CVPR*. IEEE, Vancouver, BC, Canada, 15402–15412.

[73] Wenyan Yang, Alexandre Angleraud, Roel S Pieters, Joni Pajarinen, and Joni-Kristian Kämäräinen. 2023. Seq2seq imitation learning for tactile feedback-based manipulation. In *ICRA*. IEEE, IEEE, New York, NY, USA, 5829–5836.

[74] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems* 35 (2022), 20744–20757.

[75] Shunyu Yao, Jeffrey Zhao, , et al. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*. ICLR, Kigali, Rwanda.

[76] Zheng Zhan, Yifan Gong, Pu Zhao, Geng Yuan, et al. 2021. Achieving on-mobile real-time super-resolution with neural architecture and pruning search. In *ICCV*. IEEE, Montreal, Canada, 4821–4831.

[77] Zheng Zhan, Zhenglun Kong, Yifan Gong, et al. 2024. Exploring Token Pruning in Vision State Space Models. In *The Conference on Neural Information Processing Systems*. NeurIPS, New Orleans, LA.

[78] Zheng Zhan, Yushu Wu, et al. 2024. Fast and Memory-Efficient Video Diffusion Using Streamlined Inference. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., New York, NY, USA.

[79] Zheng Zhan, Yushu Wu, Zhenglun Kong, et al. 2024. Rethinking Token Reduction for State Space Models. In *the 2024 Conference on Empirical Methods in Natural Language Processin*. Association for Computational Linguistics, Miami, Florida, USA.

[80] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model.

[81] Ruiyi Zhang, Changyou Chen, Zhe Gan, Zheng Wen, Wenlin Wang, and Lawrence Carin. 2020. Nested-wasserstein self-imitation learning for sequence generation. In *International Conference on Artificial Intelligence and Statistics*. PMLR, PMLR, Brooklyn, NY, USA, 422–433.

[82] Shuoxi Zhang, Hanpeng Liu, and Kun He. 2024. Knowledge distillation via token-level relationship graph based on the big data technologies. *Big Data Research* 36 (2024), 100438.

[83] Yihua Zhang, Yuguang Yao, et al. 2022. Advancing model pruning via bi-level optimization. *Advances in Neural Information Processing Systems* 35 (2022), 18309–18326.

[84] Pu Zhao, Xuan Shen, Zhenglun Kong, Yixin Shen, et al. 2024. Fully Open Source Moxin-7B Technical Report. *arXiv preprint arXiv:2412.06845* 1, 1 (2024).

[85] Pu Zhao, Fei Sun, et al. 2024. Pruning Foundation Models for High Accuracy without Retraining. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, Miami, Florida, USA.

[86] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Huang, et al. 2024. Sglang: Efficient execution of structured language model programs. *Advances in Neural Information Processing Systems* 37 (2024), 62557–62583.

[87] Chunting Zhou, Pengfei Liu, Puxin Xu, , et al. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems* 36 (2023), 55006–55021.

[88] Mingyan Zhou, Biao Wang, and Xiatao Sun. 2024. Developing Trajectory Planning with Behavioral Cloning and Proximal Policy Optimization for Path-Tracking and Static Obstacle Nudging. *arXiv e-prints* 1, 1 (2024), arXiv–2409.

[89] Shengchao Zhou, Weizhou Liu, Chen Hu, Shuchang Zhou, and Chao Ma. 2023. UniDistill: A Universal Cross-Modality Knowledge Distillation Framework for 3D Object Detection in Bird's-Eye View. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New York, NY, USA, 495–504. https://doi.org/10.1109/CVPR52729.2023.00495

[90] Xizhou Zhu, Yuntao Chen, Hao Tian, Tao, et al. 2023. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144* 1, 1 (2023), 1–10.