# ADDRESSING AND VISUALIZING MISALIGNMENTS IN HUMAN TASK-SOLVING TRAJECTORIES

**Sejin Kim** **Hosung Lee** **Sundong Kim**
Gwangju Institute of Science and Technology (GIST), Republic of Korea
{sjkim7822, confeitohs, sdkim0211}@gmail.com

## ABSTRACT

Understanding misalignments in human task-solving trajectories is critical for improving AI models trained to mimic human reasoning. This study categorizes these misalignments into (1) Functional Inadequacies in Tools, (2) User Unfamiliarity with Tools, and (3) Cognitive Dissonance in Users. We introduce a misalignment detection algorithm and a visualization tool to analyze discrepancies in user trajectories from O2ARC, formalizing intention-aware trajectory modeling. Additionally, we propose an intention prediction algorithm that infers user intentions by identifying frequently visited states and structured transitions. By incorporating intention-aligned supervision into a Decision Transformer-based ARC solver, we demonstrate that aligning AI with inferred human intentions significantly improves task-solving performance. These findings underscore the importance of modeling human task-solving trajectories beyond action sequences and capturing underlying intentions for better AI alignment.

## 1 INTRODUCTION

The Abstraction and Reasoning Corpus (ARC-AGI, hereafter ARC) has become a key benchmark for evaluating AI models' abstract reasoning and generalization abilities (Chollet, 2019). ARC tasks require identifying high-level transformations from minimal examples and applying them to novel inputs, mirroring human cognitive processes (Fig. 1). Despite progress in deep learning (DL) and reinforcement learning (RL), these models still struggle with ARC, underscoring a gap between AI and human reasoning.

To address this, interactive platforms (Borsky, 2021; Johnson et al., 2021; Lab42, 2022; Kim et al., 2022; Shim et al., 2024; Strandgaard, 2024; LeGris et al., 2024) collect human task-solving trajectories, offering insights into human approaches. Among them, O2ARC (Shim et al., 2024) provides extensive trajectory data, forming the basis for ARCLE (Lee et al., 2024), a reinforcement learning environment designed to model human strategies.

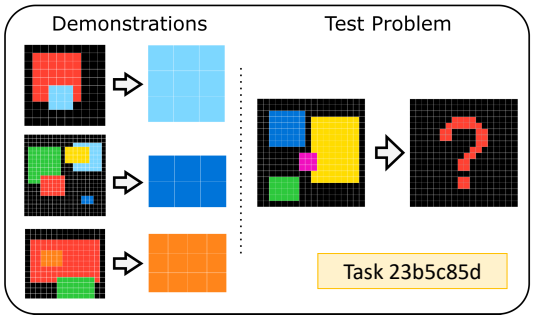



Figure 1: Desirable ARC solvers can infer the rule from input-output pairs and generate the correct output grid for a new input. The correct answer is the $3 \times 3$ magenta-colored rectangle. Solve this task on the O2ARC website: https://o2arc.com/task/23b5c85d.

However, training AI models directly on O2ARC trajectories presents challenges. Recent studies (Park et al., 2023; Kim et al., 2024) utilize these trajectories, but their effectiveness is limited by rigid preprocessing rules, reducing generalization to unseen tasks. We hypothesize that this stems from *misalignments between human intentions and trajectory data*, caused by user errors, tool limitations, and reasoning-to-action gaps. These misalignments can be categorized into:

- **Functional Inadequacies in Tools**: Missing functions make task completion difficult.
- **User Unfamiliarity with Tools**: Users struggle due to limited prior knowledge or experience.
- **Cognitive Dissonance in Users**: Misinterpreted task objectives lead to inefficient actions.

To address this, we introduce an **Intention Prediction Algorithm** (Alg. 1) that aligns trajectories with inferred human intentions. We assume that most human demonstrations follow correct task-solving steps and that frequently visited *popular states* serve as critical waypoints. By identifying these states and encoding transitions between them, we infer structured intention labels for trajectory actions.

Furthermore, we show that incorporating intention-aligned supervision significantly improves AI learning efficiency. Our experiments confirm that models trained with intention labels generalize better, reducing reliance on spurious correlations and encouraging structured decision-making.

**Contributions** We make the following key contributions:

- We analyze and categorize three types of misalignment in human task-solving trajectories.
- We propose an algorithm that infers user intention from logged trajectories.
- We demonstrate that intention-aligned trajectory learning enhances task-solving performance.

By aligning AI decision-making with structured human problem-solving strategies, this research provides a framework for improving trajectory-based learning. Our approach also paves the way for future work on learning-based intention inference and more adaptive task-solving methods.

## 2 Misalignments in Trajectory

The trajectory data collected from O2ARC provide a detailed record of users' decisions and actions as they navigate tasks. Each trajectory is structured using the state-action format defined by the ARC Learning Environment (ARCLE) (Lee et al., 2024), serving as a framework to analyze user interactions and problem-solving strategies.

### 2.1 The Format of O2ARC Trajectory

To formalize ARC tasks and trajectories, we adopt the notation proposed in recent work (Akyürek et al., 2024). A single ARC task $d$ from the set of tasks $\mathcal{D}_{\text{ARC}}$ is defined as:

$$d = \{(\mathbf{x}_k^{\text{train}}, \mathbf{y}_k^{\text{train}})_{k=1}^{K}, (\mathbf{x}^{\text{test}}, \mathbf{y}^{\text{test}})\} \in \mathcal{D}_{\text{ARC}}, \tag{1}$$

where $(\mathbf{x}_k^{\text{train}}, \mathbf{y}_k^{\text{train}})$ are input-output grid pairs provided as demonstrations, and $(\mathbf{x}^{\text{test}}, \mathbf{y}^{\text{test}})$ represent the test problem grid and its answer grid.

A set of user trajectories $\mathcal{T}_d$ are collected as sequences of states and actions while solving a task $d$. A single trajectory $\tau_d$ is represented as:

$$\tau_d = (s_0, a_0, s_1, a_1, \ldots, s_n) \in \mathcal{T}_d, \tag{2}$$

where $s_i$ denotes a state, $a_i$ an action, and $n$ the number of actions in the trajectory $\tau_d$. Every state $s_i$ and action $a_i$ in the trajectory $\tau_d$ follow the same state-action transition function $f(s, a)$ as the yellow arrow in Fig. 2:

$$f(s_i, a_i) = s_{i+1} \tag{3}$$

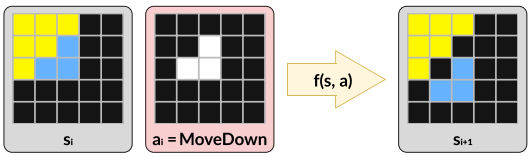



Figure 2: A single state transition step in ARCLE (Lee et al., 2024). An action transforms the current state $s_i$ into the next state $s_{i+1}$ through the transition function $f(s_i, a_i) = s_{i+1}$. In this example, the selected grids masked in white are shifted down by one row.

**States** Each state $s_i$ is derived from $\mathbf{x}^{\text{test}}$ and evolves through the state-action transition function $f(s, a)$ like:

$$s_i = \begin{cases} \mathbf{x}^{\text{test}} & \text{if } i = 0, \\ f(s_{i-1}, a_{i-1}) & \text{if } 1 \leq i \leq n. \end{cases} \tag{4}$$

States encompass the current task grid and additional contextual information, such as object properties and clipboard contents, which enable actions like copy-paste and provide richer data for analyzing user strategies (Lee et al., 2024).

This formalization captures the dynamic problem-solving process in O2ARC and provides a foundation for identifying misalignments between user actions and intentions.

### 2.2 Fundamental Concepts for Trajectory Analysis

O2ARC trajectory does not record user intentions, making it challenging to track user intentions. To address this, we propose a method to infer intentions from trajectories by leveraging the data's inherent structure and introducing fundamental assumptions.

The assumption is that users generating each trajectory $\tau_d$ share similar problem-solving strategies. Although users may adopt diverse approaches for general tasks, ARC tasks' emphasis on precise analogies leads us to hypothesize that most users converge on common intentions. While this assumption involves risks of oversimplification, prior studies (Johnson et al., 2021; LeGris et al., 2024) have observed consistency in user strategies across tasks.

**Popular States** *Popular states*, also called bottleneck states in prior studies (Johnson et al., 2021), are defined as frequently visited states among trajectories. In this study, we determined by a threshold function $\theta(|\mathcal{T}_d|)$ that depends on the total number of trajectories of the task $d$. For example, setting $\theta(x) = \sqrt{x}$ balances precision and robustness in identifying frequently visited states. Formally, the set of popular states $\mathcal{P}_d$ for $d$ is:

$$\mathcal{P}_d = \{s_i \mid N(s_i; \mathcal{T}_d) \geq \theta(|\mathcal{T}_d|)\}, \tag{5}$$

where $N(s_i; \mathcal{T}_d)$ represents the number of trajectories visiting state $s_i$.

**Intention** An *intention* is defined as an *action sequence* that transitions between two popular states. An action sequence $a_{i:j}$ consists of consecutive actions $a_i, a_{i+1}, \ldots, a_{j-1}$, achieving a transition from $s_i$ to $s_j$ while avoiding intermediate popular states. Formally:

$$\begin{aligned} &a_{i:j} = (a_i, a_{i+1}, \ldots, a_{j-1}) \text{ such that} \\ &s_i, s_j \in \mathcal{P}_d \text{ and } s_k \notin \mathcal{P}_d \text{ for } i < k < j. \end{aligned} \tag{6}$$

**Ideal Actions** To simplify the representation of the intention, we define an *ideal action*, denoted as $a^\star_{i:j}$. This ideal action encapsulates the intention behind the sequence $a_{i:j}$, abstracting the sequence into a single, efficient action that transitions directly from $s_i$ to $s_j$.

$$f(s_i, a^\star_{i:j}) = s_j. \tag{7}$$

Here, $a^\star_{i:j}$ is a hypothetical action summarizing the intention of $a_{i:j}$. The set of all such ideal actions for $d$ is:

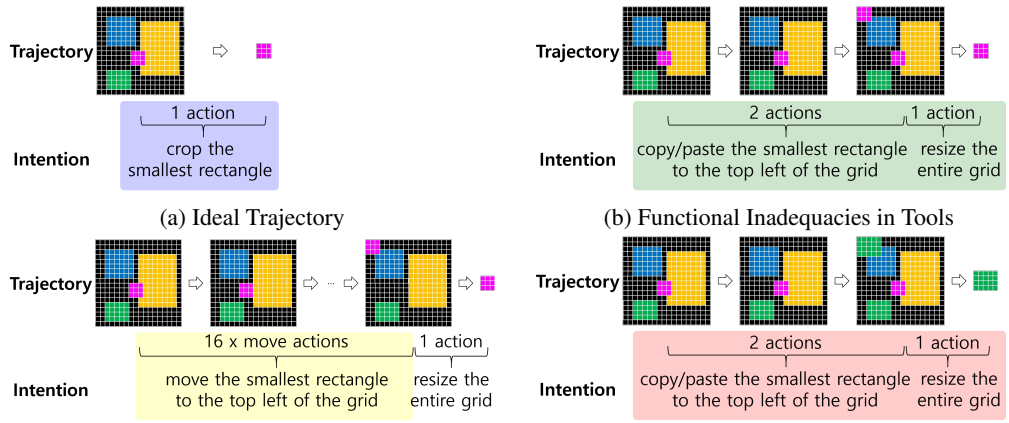
$$\mathcal{A}^\star_d = \{a^\star_{i:j} \mid f(s_i, a^\star_{i:j}) = s_j \text{ where } s_i, s_j \in \mathcal{P}_d\}. \tag{8}$$

**Ideal Trajectory** An *ideal trajectory* is defined as a sequence of popular states and ideal actions transitioning through popular states, satisfying the conditions for an optimal transition. Unlike observed trajectories in $\mathcal{T}_d$, ideal trajectories represent a conceptual framework for understanding optimal user behavior. Thus, the ideal trajectory $\tau_d^\star$ for a task $d$ is represented as:

$$\tau_d^\star = (s_0, a_0, s_1, a_1, \ldots, s_n) \text{ where } \forall s_i \in \mathcal{P}_d, \forall a_i \in \mathcal{A}_d^\star. \tag{9}$$

Here, $\tau_d^\star$ represents the optimal path that fully aligns with the user's intentions.

### 2.3 Three Types of Misalignments in Trajectory




(a) Ideal Trajectory (b) Functional Inadequacies in Tools

(c) User Unfamiliarity with Tools (d) Cognitive Dissonance in Users

Figure 3: Various trajectories of the ARC Task 23b5c85d as shown in Fig. 1. (a) The ideal trajectory transitions directly between popular states with the shortest possible sequence of actions, perfectly representing user intentions. (b) Functional inadequacies influence the trajectory in tools, where the lack of a suitable action requires combining multiple actions, resulting in longer transitions between popular states. (c) A trajectory is caused by user unfamiliarity with tools, where redundant actions reflect inefficiencies despite the existence of a shorter ideal trajectory. (d) Trajectory reflects cognitive dissonance in users, where errors or misinterpretations prevent reaching the correct answer state, deviating from transitions between popular states.

Based on Activity Theory (Engeström, 2015), these misalignments can be categorized into three types, each reflecting contradictions between users, tools, and tasks. We leverage the concepts of popular states, intentions, ideal actions, and ideal trajectories to formalize and analyze these categories.

**Functional Inadequacies in Tools** This misalignment occurs due to contradictions between tools and tasks. The toolset (e.g., O2ARC) lacks an ideal action $a_{i:j}^\star$ that transitions between two popular states $s_i$ and $s_j$:

$$a_{i:j}^\star \notin \mathcal{A}_d. \tag{10}$$

As a result, users must combine a sequence of actions $(a_i, a_{i+1}, \ldots, a_{j-1})$ to achieve the same intention:

$$a_{i:j} = (a_i, a_{i+1}, \ldots, a_{j-1}), \quad a_i, \ldots, a_j \in \mathcal{A}_d. \tag{11}$$

For instance, Fig. 3b shows a trajectory where the absence of a "cropping" action forces the user to rely on sequences like copying, pasting, and resizing. This results in an inefficient trajectory compared to the ideal trajectory shown in Fig. 3a, where direct transitions are achieved using "cropping".

**User Unfamiliarity with Tools** This misalignment arises from contradictions between users and tools. Although an ideal action $a_{i:j}^\star$ exists in the supported action set $\mathcal{A}_d$:

$$a_{i:j}^\star \in \mathcal{A}_d, \tag{12}$$

the user fails to utilize it and instead chooses an inefficient action sequence:

$$a_{i:j} = (a_i, a_{i+1}, \ldots, a_{j-1}), \quad a_i, \ldots, a_j \in \mathcal{A}_d \setminus \mathcal{A}_d^\star. \tag{13}$$

Suppose the trajectory in Fig. 3b represents an ideal trajectory. Fig. 3c then illustrates a scenario where the user, unfamiliar with the copy-paste functionality, attempts to move an object step by step. This results in a redundant action sequence of 16 actions, which could otherwise be represented by two ideal actions, highlighting inefficiency in transitioning between popular states.

**Cognitive Dissonance in Users** This misalignment reflects contradictions between users and tasks. Cognitive Dissonance can manifest in two distinct cases: (1) when users have incorrect intentions, or (2) when users possess the correct intention but execute incorrect actions, leading them to an incorrect final state. However, since the O2ARC trajectories do not explicitly record user intentions, verifying the second case is not feasible. Therefore, we define and detect Cognitive Dissonance solely based on the first case.

Specifically, Cognitive Dissonance occurs when a user's trajectory fails to reach a correct final state, deviating from popular states:

$$s_n \neq \mathbf{y}^{\text{test}}, \quad \text{where } \mathbf{y}^{\text{test}} \in \mathcal{P}_d \tag{14}$$

where $s_n$ represents the final state of the trajectory, $\mathbf{y}^{\text{test}}$ the answer grid of the task $d$, and $\mathcal{P}_d$ the set of popular states for the task $d$.

This focus highlights cases where the user either misunderstood the task objectives or encountered significant errors during problem-solving, resulting in an incorrect solution state.

### 2.4 State Space Graph for Trajectories

Fig. 4 visualizes user trajectories from Fig. 3 in the form of a state space graph. Edges with the same color across Fig. 3a-3d are blurred to highlight their misalignment with ideal trajectories. Importantly, the type of misalignment for some trajectories depends on whether cyan nodes are considered popular states. Regardless of this consideration, red edges consistently represent Cognitive Dissonance in Users (Fig. 3d), as they lead to incorrect states.

If only the blue and green nodes are considered popular states, the dashed blue edge represents an ideal action ($\xrightarrow{crop}$) that is absent from the action set (Fig. 3a). In this case, every edge in the green ($\xrightarrow{copy}\xrightarrow{paste}\xrightarrow{resize}$) and the yellow ($\xrightarrow{move} \times 16 \xrightarrow{resize}$) intentions corresponds to Functional Inadequacies in Tools (Fig. 3b), as they require multiple actions to achieve cropping, a single ideal action.

If cyan nodes are also regarded as popular states, trajectories have three intentions (copying, pasting, resizing), and the hypothetical blue edge that indicates cropping is no longer needed. In this scenario, each green edge corresponds to ideal actions ($\xrightarrow{copy}$, $\xrightarrow{paste}$, $\xrightarrow{resize}$), while every yellow edge ($\xrightarrow{move} \times 16$) indicates User Unfamiliarity with Tools (Fig. 3c), reflecting redundant actions despite the availability of more efficient ideal actions ($\xrightarrow{copy}\xrightarrow{paste}$).

## 3 Misalignment Analysis

This section analyzes misalignments in O2ARC trajectories across four hierarchical levels. We begin with an **action-level** analysis of the frequency and distribution of actions leading to specific states. Next, we examine **intention-level** misalignments between popular states. We then explore **trajectory-level** relationships between misalignment types. Finally, a **task-level** analysis investigates variations in misalignments across tasks.

### 3.1 Action-Level Analysis

This subsection analyzes action distributions across states to identify patterns associated with misalignments. In the state space graph (Fig. 4), node size indicates the in-degree of a state, representing actions leading to it. Smaller average in-degrees suggest more diverse strategies, fewer popular nodes, and greater misalignment potential.

**Average In-Degree** The in-degree distribution reveals differences in task complexity and user behavior. Tasks with smaller average in-degrees, summarized in Table 1, often involve pixel-level

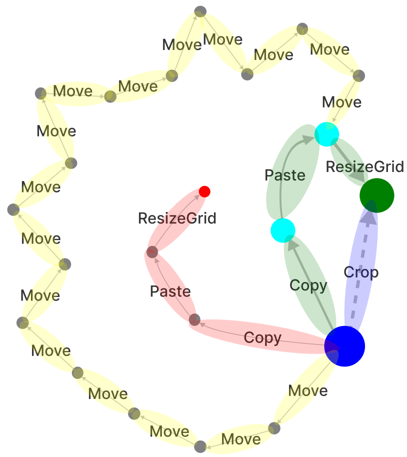



Figure 4: State space graph for Fig. 3. The blue node represents the test input grid, the green node the answer grid, and red nodes incorrect submissions. Node and edge thickness indicate frequency of states and actions.

Table 1: Tasks with the bottom 10 average node sizes. These tasks involve complex patterns, multiple objects, or filling in patterns, requiring diverse strategies. See Sec. 3.1 for discussion

| Task ID (ARC) | Shortest Length | Task Description |
|---|---|---|
| 017c7c7b | 3 | stretch and coloring |
| 0e206a2e | 5 | placing the object |
| 025d127b | 2 | squash the object |
| 0dfd9992 | 10+ | carpet filling |
| 05269061 | 3 | carpet filling |
| 1a07d186 | 10+ | string pulling an object |
| 0962bcdd | 2 | expand the pattern |
| 3428a4f5 | 10+ | Coloring and Bitwise OR |
| 09629e4f | 6 | counting and expansion |
| 00d62c1b | 1 | filling the enclosed part |

adjustments or non-intuitive solutions. For example, Task 0dfd9992 required inefficient pattern filling across large grids, while Task 1a07d186 involved moving numerous pixels individually to implement gravity. These tasks dispersed user actions across more states, leading to lower in-degrees and greater variability.

**Key Insights from Action-Level Analysis** Low average in-degrees highlight Functional Inadequacies in Tools, where available actions fail to efficiently address task requirements. Pixel-level tasks forced users to rely on repetitive actions, emphasizing the need for a more comprehensive action set.

With in-degree distributions, we identified key patterns highlighting Functional Inadequacies in Tools, particularly in tasks requiring repetitive or granular actions. These findings show how user behavior and tool limitations interact at a granular level. This action-level perspective sets the stage for intention-level analysis, where we investigate how sequences of actions between popular states contribute to broader misalignment patterns.

### 3.2 Intention-Level Analysis

This subsection examines misalignments at the intention level by analyzing action sequences between popular states. Each intention corresponds to a sequence of actions connecting two consecutive popular states, enabling the categorization of misalignments within these segments.

**Distribution of Misalignment Types** Table 2 presents the distributions of intentions and actions across different misalignment types. Although 91.11% of intentions are aligned, they represent only 49.57% of the total actions, showing that misaligned intentions involve less optimal actions, particularly in cases of Functional Inadequacy where repetitive pixel-level operations inflate action counts. This imbalance is pronounced in Functional Inadequacy, where repetitive pixel-level adjustments inflate the action proportion for tasks requiring efficient solutions.

**Key Insights from Intention-Level Analysis**

- **Aligned intentions dominate**: The high proportion of aligned intentions (91.11%) demonstrates consistent user behavior across most tasks. This suggests that O2ARC trajectories capture well-structured strategies, making them a reliable dataset for analyzing user behavior.
- **Functional Inadequacy drives inefficiency**: Despite representing only 4.15% of intentions, Functional Inadequacy accounts for 27.54% of all actions, revealing toolset limitations for tasks requiring

repetitive or complex modifications. Addressing these inefficiencies through tool enhancements improves user performance.

- **Overlap in misalignment types**: User Unfamiliarity (2.31% and Cognitive Dissonance (2.43%) exhibited similar intention proportions. Also, their action proportions (11.30% and 11.59%) show similar patterns. This overlap indicates potential shared causes but highlights the need for further investigation into how these misalignments differ in trajectory-level manifestations.

The intention-level analysis revealed that most user strategies align with ideal actions, although misaligned intentions lead to suboptimal actions. This insight highlights how tool inefficiencies and user strategies manifest at an intention level. Building on this, the next analysis extends to the trajectory level, where we explore the cumulative effects of misalignments across entire action sequences.

### 3.3 Trajectory-Level Analysis

This subsection examines misalignments at the trajectory level by analyzing complete sequences of actions. Figure 5 illustrates the distribution of misalignments and their overlaps.

Table 2: Intention-level misalignment, showing their relative occurrence as a proportion of total intentions and actions. Misaligned intentions, although fewer, account for a significant share of suboptimal actions.

| Misalignment Type | Intentions | Actions |
|---|---|---|
| User Unfamiliarity | 2.31% | 11.30% |
| Functional Inadequacy | 4.15% | 27.54% |
| Cognitive Dissonance | 2.43% | 11.59% |
| Not Misaligned | 91.11% | 49.57% |

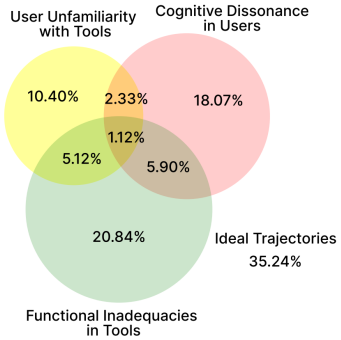



Figure 5: Trajectory-level misalignment distribution.

**Key Insights from Trajectory-Level Analysis**

- **Consistency in ideal trajectories**: The proportion of ideal trajectories (35.24%) highlights structured user strategies, particularly for simpler tasks with intuitive solutions. This suggests that O2ARC trajectory data capture meaningful behavior patterns essential for training AI systems to emulate effective approaches.
- **Prevalence of Functional Inadequacy**: Functional Inadequacy appeared in over 40% of misaligned trajectories, underscoring limitations in the current action set. Tasks involving repetitive operations, such as pixel-level modifications, were particularly affected. Addressing these issues with tool enhancements could reduce misalignments and improve efficiency.
- **Minimal overlap between misalignment types**: Overlaps between misalignment types were minimal, especially between User Unfamiliarity and Cognitive Dissonance. This indicates distinct causes: User Unfamiliarity stems from inefficient tool usage, while Cognitive Dissonance reflects task misinterpretations or incomplete solutions. Differentiating these misalignments enables tailored interventions.

The trajectory-level analysis showed that Functional Inadequacy is the most prevalent misalignment type and that overlaps between other misalignments remain minimal. These findings suggest distinct causes for each misalignment type and reinforce the need to address tool limitations. To further contextualize these patterns, the next section investigates how misalignments vary across tasks, providing a broader view of user behavior and tool challenges.

## 4 EVALUATING INTENTION-ALIGNED TRAJECTORIES

In the previous sections, we analyzed misalignments in O2ARC trajectories and found that a significant proportion—approximately 65% as shown in Fig. 5—exhibit some form of misalignment. This suggests that addressing these misalignments could substantially improve the efficiency of trajectory-based learning. To this end, we propose a method for predicting the implicit intentions within O2ARC trajectories based on popular states. Furthermore, we demonstrate that incorporating these predicted intentions into model training enhances task-solving performance.

### 4.1 INTENTION PREDICTION ALGORITHM

We present Algorithm 1, which formalizes the process of intention prediction by leveraging popular states as key decision points. Our key assumption is that most human demonstrations follow correct task-solving procedures and that popular states represent crucial intermediate points in the task-solving process.

**Algorithm 1:** Intention Prediction Algorithm

**Input:** Trajectories for Solving task $\mathcal{T}_d = (s_0, a_0, \cdots, s_n)$,
Threshold Function $\theta$
**Output:** Task Trajectories with Assigned Intentions $\mathcal{T}$

```
P ← {} ;                                  // Initialize popular states set
// Step 1.  Identify Popular States
for τ_d ∈ T_d do
    for s_i ∈ τ_d do
        if N(s_i) ≥ θ(|T|) then
            P ← P ∪ {s_i} ;

// Step 2.  Assign Intention Edges
for τ_d ∈ T do
    s ← τ_d[s_0] ; for s_i, a_i ∈ τ_d do
        if s_i ∈ P then
            for s_j, a_j ∈ τ_d[s : s_i] do
                a_j[intention] ← (s, s_i) ;
            s ← s_i ;                     // Update current popular state
return T_d
```

Under this assumption, we hypothesize that user intentions exist at the level of transitions between popular states. Specifically, for each trajectory $\tau_d$, we annotate all actions $a_{i:j}$ occurring between two consecutive popular states $(s_i, s_j)$ with an intention tuple $(s_i, s_j)$. This pair of popular states represents an ideal edge $a^{\star}_{i:j}$ connecting key transitions within the trajectory.

Building on existing research that leverages human task-solving trajectories for training (Park et al., 2023), we investigate whether including intention information improves model learning. Our approach augments the training data with intention annotations and evaluates the impact of these additional supervisory signals on task-solving performance.

### 4.2 EXPERIMENTS

To evaluate the effectiveness of intention supervision, we integrated intention annotations into an existing Decision Transformer (DT) framework (Park et al., 2023). Using a predefined transformation policy, the original model was trained on human expert trajectories generated from augmented input-output pairs. The model iteratively predicts state, action, return-to-go, and timestep based on observed inputs during training. Prior research has shown that augmenting state representations with object information improves task-solving performance.

To extend this framework, we modified the model to process intention-labeled inputs and explicitly predict intention transitions. We achieved this by introducing an auxiliary loss term encouraging the

model to learn intention structures alongside state-action transitions. By incorporating this additional supervision, we hypothesize that the model will better capture the structured reasoning underlying human task demonstrations.

Table 3: Incorporating intention information enhances task-solving performance. DT and DT+PnP are baselines without intention. DT represents vanilla Decision Transformer Chen et al. (2021), and DT+PnP represents the baseline with object detection Park et al. (2023).

| **Test Set** | **DT** | **DT + Intention** | **DT + PnP** | **DT + PnP + Intention** |
|---|---|---|---|---|
| 1 | 61.35 | 59.6 | 84.2 | 90.1 |
| 2 | 60.2 | 56.25 | 82.7 | 89.25 |
| 3 | 60.35 | 58.35 | 83.9 | 88.55 |
| 4 | 60.25 | 58.4 | 83.55 | 89.85 |
| **Avg** | 60.54 | 58.15 | 83.59 | 89.44 |

### 4.3 Results

Our results are summarized in Table. 3, demonstrate that incorporating intention supervision significantly enhances task-solving performance. Specifically, adding intention information to the existing model with object features (DT + Object) resulted in an average performance improvement of 5.85% across four datasets, each consisting of 2,000 augmented tasks. Despite the strong baseline performance of DT + Object (83.59%), integrating intention alignment further boosted accuracy to nearly 90%, providing strong evidence that intention-based supervision positively impacts model learning.

However, as seen in the loss curves, training with intention supervision required more samples, suggesting an increased learning complexity. This outcome is expected, as the model must learn a richer representation incorporating both object-level and intention-based reasoning. Notably, when intention information was added to the baseline model without object features (DT + Intention), the model exhibited reduced efficiency in leveraging intention cues, leading to a drop in performance rather than an improvement. This indicates that intention alignment alone is insufficient without a structured representation of object-level information.

These findings underscore the importance of combining structured object representations with intention-based supervision to improve trajectory learning. While intention alignment enhances reasoning over key transitions, its effectiveness depends on the availability of complementary features such as object information. Future research could explore more efficient training strategies or explicit intention annotation methods to further refine this approach.

## 5 Conclusion and Future Works

Understanding the discrepancy between a user's thought process and the externalized steps recorded in trajectories is a critical challenge for developing systems that effectively support problem-solving behavior. This requires considering latent cognitive processes that are not directly observable in trajectory logs but significantly influence task performance.

In this study, we introduced an algorithm to identify three types of misalignment: functional inadequacies in tools, user unfamiliarity with tools, and cognitive dissonance in users. Additionally, we proposed an intention prediction framework based on popular states and demonstrated its effectiveness in improving trajectory-based learning. Our findings suggest that intention-aligned supervision helps models capture higher-level reasoning patterns beyond low-level execution steps.

Building on these contributions, several promising directions emerge for future research. First, advanced reward modeling techniques from Inverse Reinforcement Learning (IRL) and Reward Learning from Human Preferences (RLHP) can help align AI models with human strategies by prioritizing meaningful transitions. Second, improving intention inference using Graph Neural Networks (GNNs) or Transformer-based models could enable the automatic discovery of latent structures in task-solving trajectories. Finally, refining O2ARC tools and scoring metrics using

state-space graph analysis could provide a more robust framework for evaluating trajectory quality beyond task completion time.

As AI systems continue to evolve, understanding the underlying intentions behind user actions will be essential for building more interpretable and generalizable decision-making models. Future work will explore alternative approaches for refining intention prediction and integrating intention-based reasoning into broader AI applications.

## ACKNOWLEDGEMENTS

This work was supported by IITP (RS-2024-00445087), NRF (RS-2024-00451162), GIST (Postdoc value-up) grants funded by the Ministry of Science and ICT, Korea.

## REFERENCES

Ekin Akyürek, Mehul Damani, Linlu Qiu, Han Guo, Yoon Kim, and Jacob Andreas. The Surprising Effectiveness of Test-Time Training for Abstract Reasoning. *arXiv:2411.07279*, 2024.

Alexey Borsky. The ARC Game, 2021. URL `https://volotat.github.io/ARC-Game/`.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *NeurIPS*, 2021.

François Chollet. On the Measure of Intelligence. *arXiv:1911.01547*, 2019.

Yrjö Engeström. *Learning by Expanding*. Cambridge University Press, 2015.

Aysja Johnson, Wai Keen Vong, Brenden M. Lake, and Todd M. Gureckis. Fast and Flexible: Human Program Induction in Abstract Reasoning Tasks. In *CogSci*, 2021.

Subin Kim, Prin Phunyaphibarn, Donghyun Ahn, and Sundong Kim. Playgrounds for Abstraction and Reasoning. In *NeurIPS Workshop on Neuro Causal and Symbolic AI*, 2022.

Yunho Kim, Jaehyun Park, Heejun Kim, Sejin Kim, Byung-Jun Lee, and Sundong Kim. Diffusion-based offline rl for improved decision-making in augmented arc task. *arXiv preprint arXiv:2410.11324*, 2024.

Lab42. ARCreate Playground, 2022. URL `https://arc-editor.lab42.global/playground`.

Hosung Lee, Sejin Kim, Seungpil Lee, Sanha Hwang, Jihwan Lee, Byung-Jun Lee, and Sundong Kim. ARCLE: The Abstraction and Reasoning Corpus Learning Environment for Reinforcement Learning. In *CoLLAs*, 2024.

Solim LeGris, Wai Keen Vong, Brenden M Lake, and Todd M Gureckis. H-ARC: A Robust Estimate of Human Performance on the Abstraction and Reasoning Corpus Benchmark. *arXiv:2409.01374*, 2024.

Jaehyun Park, Jaegyun Im, Sanha Hwang, Mintaek Lim, Sabina Ualibekova, Sejin Kim, and Sundong Kim. Unraveling the ARC Puzzle: Mimicking Human Solutions with Object-Centric Decision Transformer. In *ICML Workshop on Interactive Learning with Implicit Human Feedback*, 2023.

Suyeon Shim, Dohyun Ko, Hosung Lee, Seokki Lee, Doyoon Song, Sanha Hwang, Sejin Kim, and Sundong Kim. O2ARC 3.0: A Platform for Solving and Creating ARC Tasks. In *IJCAI*, 2024. URL `https://o2arc.com`.

Simon Strandgaard. ARC Interactive, 2024. URL `https://neoneye.github.io/arc/`.