Intent-aligned Autonomous Spacecraft Guidance via Reasoning Models

Anonymous CVPR submission

Paper ID *****

Abstract

Future spacecraft operations require autonomy that can interpret high-level mission intent while preserving safety. However, existing trajectory optimization still relies heavily on expert-crafted formulations and does not support intent-conditioned decision-making. This paper proposes an intent-aligned spacecraft guidance framework that links high-level reasoning and safe trajectory optimization through explicit intermediate abstractions, based on behavior sequences and waypoint constraints. A foundation model first predicts an intent-aligned behavior plan, a waypoint generation model then converts it into waypoint constraints, and the safe trajectory is computed via optimization. This decomposition enables scalable supervision without sacrificing safety. Numerical experiments in close-proximity operation scenarios demonstrate that the proposed pipeline achieves over 90% SCP convergence and yields a $1.5\times$ *higher rate of generating trajectories that satisfy the top intent-prioritized performance criteria than heuristic decision-making. These results support the use of intermediate behavior abstraction as a practical interface between foundation-model reasoning and safety-critical onboard spacecraft autonomy.*

1. Introduction

Safe and autonomous guidance and control are essential for future in-space logistics. During critical phases, spacecraft must transition from ground supervision to goal-driven onboard autonomy, requiring systems that reason over high-level objectives while guaranteeing real-time safety and dynamic feasibility. Although numerical optimal control methods [19] enable fast nonconvex trajectory generation, they face two key limitations: (i) significant domain expertise is required to translate mission intent into safe, executable trajectories, and (ii) proximity operations still rely heavily on heuristic waypoint-based strategies [6]. While effective, these approaches confine strategic reasoning to human design processes or rigid rule-based logic.

Concurrently, foundation models (FMs) have demonstrated general reasoning capabilities and are increasingly adapted for physical AI through multimodal perception-to-action pipelines [15, 18, 26]. Nonetheless, an end-to-end reasoning pipeline that ensures the safety and feasibility of the generated trajectory is lacking in space applications.

This paper proposes an intent-aligned spacecraft guidance framework that translates high-level mission intent into a safe, dynamically feasible, and semantically grounded trajectory through three stages: (i) behavior planning with a reasoning trace, (ii) structured waypoint generation, and (iii) constrained nonconvex trajectory optimization. In this hierarchy, behavior sequences provide the key intermediate abstraction between foundation-model reasoning and safe trajectory optimization, while waypoint constraints geometrically ground the selected behavior plan. This decomposition enables scalable supervision, systematic incorporation of expert knowledge, and interpretable intervention points at the behavior and waypoint levels. As a result, the framework supports progressive autonomy and practical deployment while maintaining operational oversight.

The main contributions of this paper are as follows:

- An intent-aligned spacecraft guidance architecture that links foundation model reasoning to safe trajectory generation through behavior and waypoint abstractions.
- A staged bootstrapping strategy that constructs scalable supervision for waypoint generation and behavior-level reasoning from optimization-derived trajectory outcomes.
- End-to-end validation in a close-proximity inspection scenario, showing over 90% SCP convergence and improved intent-aligned trajectory generation over heuristics.

2. Related Work

Learning-Based Nonconvex Trajectory Generation: Sequential convex programming (SCP) [19] has emerged as a leading framework for nonconvex trajectory generation due to its convergence guarantees [3, 20, 21] and ability to accommodate complex constraints [2, 7, 22]. To mitigate sensitivity to initialization, learning-based warm-start strategies have therefore been increasingly explored [1, 5, 11, 16, 23]. These approaches leverage neural networks to predict initial trajectories or control sequences that accelerate convergence and improve robustness. Recent work extends beyond enhancement and demonstrates the fast generation of multi-objective trade spaces [25]. This paper is inspired by prior work on

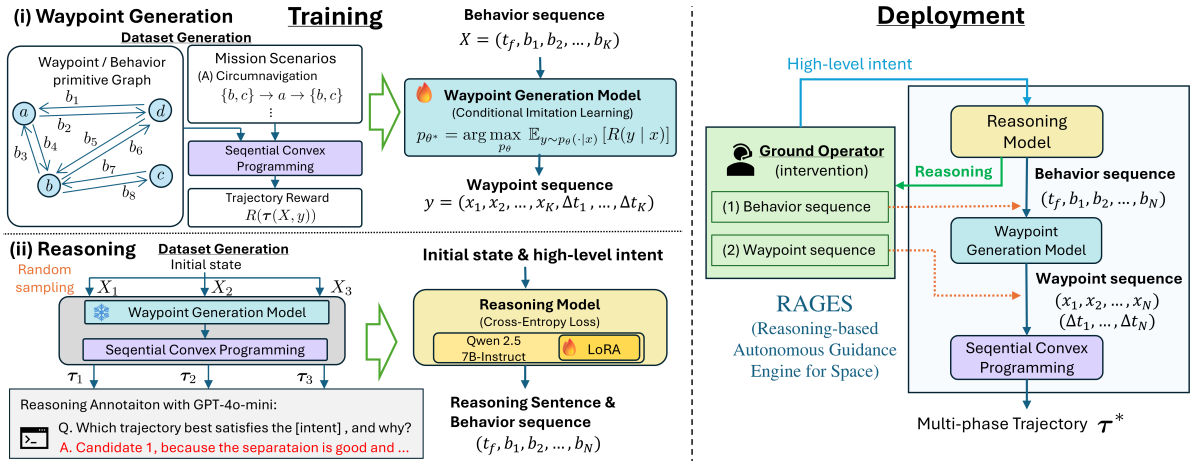


Figure 1. **Training and deployment of the proposed intent-to-trajectory pipeline.** The framework comprises (i) a reasoning model that predicts a behavior sequence from scenario context and high-level intent, (ii) a waypoint generator that produces waypoint constraints, and (iii) an SCP solver that enforces dynamics and safety. *Training:* the waypoint generator is trained first using SCP rollouts; the reasoning model is then fine-tuned using the training data generated by evaluating candidate behavior sequences through the frozen waypoint generator and SCP, with automated annotation by a pretrained LLM. *Deployment:* the reasoning model and waypoint generator run online, and SCP produces the final feasible trajectory. The operator can override the generated decision at both the behaviors and the waypoints.

semantic trajectory generation conditioned on text inputs via language encoders, in which safety is enforced via a subsequent SCP [24]. While effective, it relies on text–trajectory pairs constructed from a limited, predefined waypoint set, limiting generalization and scalability.

Foundation Models for Robotic Reasoning: Foundation model (FM)–based reasoning has gained traction in robotics as a scalable mechanism for encoding cross-domain knowledge and task priors [13, 15, 26]. In space applications, recent works explore language-conditioned control and semantic reasoning [4, 8, 14], but typically map commands directly to control inputs without enforcing dynamically feasible, constraint-satisfying trajectories. High-stakes space operations further require interpretability and operator oversight, which fully end-to-end architectures often obscure.

3. Methodology

The objective of this paper is to map a high-level mission intent, specified by the operator and encoded with performance priorities, to a dynamically feasible trajectory. To enable this *intent-to-trajectory* pipeline, a tri-level decision-making architecture is proposed (cf. Fig. 1), which comprises: (i) a FM that interprets the scenario context and infers a mid-level behavior sequence; (ii) a waypoint generation model that converts this sequence into structured waypoint constraints; and (iii) a nonconvex trajectory generation module that synthesizes a feasible trajectory. The hierarchy is enabled by the intermediate representations: behavior primitives and waypoint constraints. Because high-level objectives are inherently under-constrained and admit many feasible trajectories, direct intent-to-trajectory learning can be unstable and data-inefficient. Decomposing intent into structured behavior sequences increases decision specificity and improves the robustness of trajectory synthesis, while providing multiple checkpoints for operator intervention.

Training follows a staged bootstrapping process aligned with the hierarchy. The waypoint generator is first trained using the trajectories solved via SCP and labeled with performance metrics. The reasoning model is then trained with data that couples high-level intent and trajectory-level quality via the trained waypoint model. This sequential procedure propagates optimization-level supervision to higher-level semantic reasoning. The subsequent sections introduce the modules in reverse inference order: trajectory generation via SCP, waypoint generation, and finally the reasoning model.

3.1. Trajectory generation via SCP

The foundation of the proposed architecture is the nonconvex trajectory generation problem $\mathcal{P}$, which is shown as:

$$\min_{\{x_j,u_j\}_{j=1}^N} \quad \sum_{j=1}^{N} \|u_j\|_2 \tag{1a}$$

$$\text{subject to} \quad x_{j+1} = \Phi(t_{j+1}, t_j)(x_j + \Gamma_j u_j), \forall j \in \mathbb{Z}_{1:N-1}, \tag{1b}$$

$$x_1 = \hat{x}_0, \quad x_{j_k} = \tilde{x}_k, \qquad \forall k \in \mathbb{Z}_{1:K}, \tag{1c}$$

$$g(x_j, u_j) \leq 0, \qquad \forall j \in \mathbb{Z}_{1:N-1}. \tag{1d}$$

The state vector $x \in \mathbb{R}^6$ is in quasi-nonsingular Relative Orbital Elements (qnsROE) [6], and the objective minimizes total fuel expenditure. Eq. (1b) is discrete-time dynamics with impulsive inputs $u \in \mathbb{R}^3$ applied in the Radial–Tangential–Normal (RTN) frame; Eq. (1c) is the waypoint-passage constraints parameterized by $(\tilde{x}_k, j_k)$ and the initial state $\hat{x}_0$; and Eq. (1d) is a nonconvex safety constraint. In particular, chance-constrained passive safety [10, 23] is enforced

to guarantee separation from the ellipsoidal keep-out zone (KOZ) with a probabilistic margin, even under sudden loss of control authority. This problem is solved via SCP [19].

3.2. Waypoint generation model

The waypoint generation model predicts a waypoint placement policy conditioned on an intermediate sequence of *behavior primitives*. Its output is $y = \left(\{\tilde{x}_k, d_k\}_{k=1}^{K}\right)$, where $\tilde{x}_k$ denotes the k-th waypoint state and d_k is the corresponding transfer duration in discrete steps. With a fixed time grid of size Δt, the total maneuver time is $t_f = \sum_{k=1}^{K} d_k \, \Delta t$. The model is conditioned on $X = \left(\hat{x}_0,\, t_f,\, \{b_k\}_{k=1}^{K},\, \boldsymbol{z}\right)$, where $\hat{x}_0$ is the initial belief state, t_f is the maneuver horizon, $\{b_k\}_{k=1}^{K}$ is the behavior-primitive sequence, and $\boldsymbol{z} = \left((t_0),\, r_{\mathrm{KOZ}},\, \beta\right)$ collects scenario parameters: target orbital elements, the keep-out-zone radius, and a scalar uncertainty parameter, respectively.

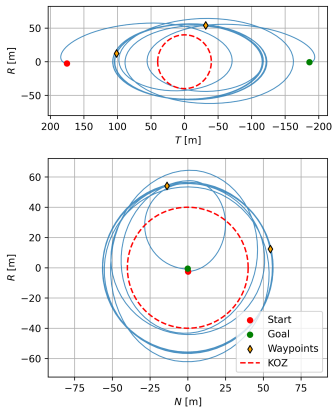
Waypoint generation is intended to approximate the trade-offs made by mission designers to maximize overall trajectory quality. As manual annotation of high-quality waypoint strategies is not scalable, a proxy reward is defined based on the solution of a trajectory optimization problem $\mathcal{P}(X, y)$. Given input X and waypoint configuration y, solving $\mathcal{P}(X, y)$ via SCP yields a feasible trajectory $\boldsymbol{\tau}^{*}_{\mathcal{P}(X,y)} = \{x_j^{*}, u_j^{*}\}_{j=1}^{N}$, from which a scalar reward R is computed. In particular, this paper defines R as a weighted sum of the control cost and trajectory-wide observation quality. By restricting the design space to waypoint placement, the optimization of R becomes tractable.

The waypoint generator is modeled as a conditional Gaussian distribution whose parameters are predicted by a multi-layer perceptron (MLP). It is trained via weighted maximum likelihood estimate, where each sample is weighted according to the reward of the corresponding SCP solution.

A dataset of (X, y, R) tuples is constructed via structured scenario generation. Each behavior primitive corresponds to a transfer between relative orbital domains, inducing a directed graph in the qnsROE space: nodes represent admissible domains, and edges encode behavior-induced transitions. Three representative on-orbit inspection scenarios (*Circumnavigation*, *Flyby*, and *Ducking*) define distinct connectivity structures over this graph, each associated with admissible transfer-time windows. Behavior sequences and corresponding durations are sampled under these structural constraints. This rule-based curation promotes scenario diversity while excluding semantically inconsistent transitions.

3.3. Reasoning model

The reasoning model predicts a dynamically feasible behavior sequence consistent with mission intent. It maps the input $\zeta = [\hat{x}_0,\, \boldsymbol{z},\, \text{intent}]$ to the output $\eta = [\text{reasoning},\, t_f,\, b_1, \ldots, b_K]$, which includes a behavior sequence, the total transfer time, and a concise reasoning trace. In this work, *intent* is represented as an ordered priority set,



Input	
x_0 [m]	[0.0, 175.0, 0.0, 2.5, 0.0, 2.5]
Intent	{fuel, time, observation, safety margin}
$\mathrm{oe}(t_0)$	[6.74×10^{6}, 5.58×10^{-4}, 0.90, −1.03, 0.46, 0.94]
r_{KOZ} [m]	40.0
β	1.000
Output	
$[b_1, b_2, b_3]$	8 (Approach from +V-bar) → 1 (Station-Keeping) → 9 (Retreat to -V-bar)
N	68
Reasoning	It seems to lead to low delta-v while maintaining a reasonable transfer time.
$\{\tilde{x}_j\}$ [m]	[[0.0, −9.4, 0.0, 56.7, 0.0, 56.0], [0.0, −4.4, 0.0, 55.8, 0.0, 56.5], [0.0, −187.6, 0.0, 0.9, 0.0, 0.0]]
$\{d_j\}$	[20, 26, 22]

Figure 2. Representative Trajectory Generation.

priority = {fuel, time, observation, safety margin}, and the reasoning trace is a single sentence that references one or two top-priority metrics. This structured interface enables scalable dataset construction. Importantly, at deployment, the reasoning model does not observe downstream waypoint- or trajectory-level metrics; it must infer a behavior sequence solely from the scenario context and the intent specification.

To construct paired data (ζ, η), an annotation pipeline converts optimization-derived performance comparisons into supervision for behavior-level decision-making (cf. Fig. 1). For fixed $(\hat{x}_0, \boldsymbol{z}, \text{intent})$, $M = 4$ candidate mid-level decisions are sampled, each specifying a valid behavior-primitive sequence and total time t_f. Each candidate is passed through the frozen waypoint generator and the SCP that solves Eq. 1, treated as a black-box simulator. This yields a feasible trajectory and four quantitative metrics corresponding to the intent priority list. A pretrained LLM is used during dataset construction to convert structured candidate comparisons into a selected behavior plan and concise rationale consistent with the ordered intent priorities. This produces structured targets that contain both the selected behavior decision and its reasoning trace; trajectory-level metrics are used only for supervision during training.

The reasoning model is trained via supervised fine-tuning of a pretrained LLM (Qwen2.5-7B-Instruct). A token-wise cross-entropy loss is applied to the autoregressive output sequence, covering both behavior tokens and reasoning tokens, with a weighting factor controlling their relative contributions. Low-Rank Adaptation (LoRA) [12] is employed for parameter-efficient fine-tuning by introducing trainable low-rank updates to frozen transformer layers.

4. Results

Figure 2 summarizes representative inputs and outputs, including the inferred intermediate decisions and the resulting multi-phase RTN-frame trajectory. The solution generates a behavior sequence aligned with the highest-priority intent metric, followed by waypoint constraint construction and SCP-based trajectory optimization to enforce passive safety.

		Neural Module		Heuristic
		Unweighted	Weighted	
SCP Success [%]		**90.8**	90.6	72.2
Reward mean / std.		-2.99 ± 0.95	**-2.91 ± 0.96**	-3.28 ± 1.14
max. waypoint error [%]	0 m	**62.4**	56.4	100
	< 5 m	83.8	**81.6**	100
	< 10 m	89.4	**98.8**	100

Table 1. Statistical performance of the waypoint generation.

Reasoning Waypoint Gen.	Heur. Heur.	Heur. Neural	Neural Heur.	Neural Neural
Behavior-Seq Feasibility [%]		–	100	
SCP Success [%]	75.2	**91.6**	65.2	90.2
Intent-Reasoning match (2/2) [%]		–	52.4	
Intent-Reasoning match: (1/2) [%]		–	47.6	
Intent-Reasoning match: (0/2) [%]		–	0.0	
Reason-Traj. Dual-Metric Win [%]	5.0	12.4	3.8	**20.4**
Reason-Traj. Single-Metric Win [%]	38.6	41.4	25.4	**45.6**
Intent-Traj. Dual-Metric Win [%]	7.4	13.0	3.6	**18.0**
Intent-Traj. Single-Metric Win [%]	35.6	42.2	24.4	**44.0**

Table 2. End-to-end performance analysis with the consistency between the high-level intent, reasoning, and final trajectory, across four decision-making pipelines based on heuristics (Heur.) or trained neural modules (Neural).

4.1. Performance of waypoint generation

The waypoint generation model is first evaluated to validate the effectiveness of the bootstrapped training procedure. A total of 50,000 (X, y, R) samples are generated with a 90–10 train–validation split, and performance is assessed on 500 independently sampled test cases using deterministic mean predictions. Three approaches are compared: reward-weighted/unweighted imitation learning and a heuristic waypoint placement method based on relative orbit geometry. In the heuristic baseline, waypoints are randomly sampled within the admissible domains defined by the waypoint–behavior primitive graph, subject to predefined geometric constraints. This guarantees that all sampled waypoints remain within the targeted domain.

As shown in Table 1, both neural models improve the downstream SCP success rate to approximately 90%, compared to the 72% in the heuristic baseline. The reward-weighted model achieves the highest mean reward with comparable variance, indicating modest but consistent gains in trajectory quality when reward information is incorporated during training. Furthermore, roughly 60% of generated waypoint sequences lie strictly within the prescribed geometric waypoint domains. However, over 98% of waypoints fall within a 10 m tolerance of their designated domains for the weighted imitation learning, demonstrating strong structural alignment even when exact domain membership is not satisfied.

4.2. End-to-end evaluation

Finally, an end-to-end statistical evaluation of the proposed hierarchical pipeline is conducted across four configurations, defined by combinations of heuristic and trained neural modules for behavior-sequence reasoning and waypoint generation. For heuristic reasoning, a feasible behavior sequence initiating from the waypoint domain to which the initial state belongs is randomly sampled. Three categories of metrics are considered: (i) behavior-sequence feasibility, assessing whether the reasoning module produces an admissible behavior sequence; (ii) success rate in the downstream SCP convergence; and (iii) semantic consistency.

Semantic consistency is quantified by measuring alignment between high-level intent, reasoning traces, and resulting trajectory performance. For each scenario, up to two prioritized metrics are extracted from the reasoning trace using GPT-4o-mini and compared against the top two priorities specified in the high-level intent to quantify intent–reasoning alignment. The final converged trajectory is then compared with the three alternatives generated from the same scenario input ζ. A *dual (single)-metric win* is recorded when two (one) out of two prioritized metrics outperform those of the other candidates. During evaluation, both the reasoning and waypoint generation models operate deterministically to ensure reproducibility.

Table 2 summarizes the results over 500 test scenarios for all combinations of heuristic and trained (neural) reasoning and waypoint modules. The results show that the trained reasoning model achieves 100% feasibility in generating feasible behavior sequences. Also, the SCP convergence rate is largely governed by the waypoint generation, where the trained model substantially improves the success rate under both reasoning settings.

In terms of semantic alignment, more than 50% of reasoning traces match both top-priority intent metrics, and all match at least one. The fully trained configuration also achieves the highest dual-metric win rates for both intent–trajectory and reasoning–trajectory alignment, yielding approximately a $1.5\times$ relative improvement (18.0% and 20.4%) over heuristic reasoning with the same trained waypoint generator (13.0% and 12.4%), which demonstrates the effectiveness of introducing a neural module for generating mid-level behavior sequences.

Overall, the results show that the hierarchical architecture successfully propagates high-level intent through intermediate behavior representations to trajectory-level performance. In particular, training the waypoint generator improves SCP convergence, and bootstrapping the reasoning model with trajectories from the trained waypoint model enhances semantic consistency, enabling the intent-aligned safe trajectory generation.

5. Conclusion and Future Work

This paper presents an intent-aligned autonomous spacecraft guidance framework that links reasoning and safe trajectory generation through behavior-level abstraction and waypoint generation. Future work will incorporate learning-based warm-start strategies and a more rigorous analysis of the generalization properties of the elicited reasoning capability.

References

[1] Somrita Banerjee, Thomas Lew, Riccardo Bonalli, Abdulaziz Alfaadhel, Ibrahim Abdulaziz Alomar, Hesham M Shageer, and Marco Pavone. Learning-based Warm-Starting for Fast Sequential Convex Programming and Trajectory Optimization. In *IEEE Aerospace Conference*, pages 1–8, 2020. 1

[2] Lars Blackmore. Autonomous precision landing of space rockets. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, pages 15–20. The Bridge Washington, DC, USA, 2016. 1

[3] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone. GuSTO: Guaranteed sequential trajectory optimization via sequential convex programming. In *IEEE International Conference on Robotics and Automation (ICRA)*, Montreal, May, 2019. 1

[4] Alejandro Carrasco, Victor Rodriguez-Fernandez, and Richard Linares. Large language models as autonomous spacecraft operators in kerbal space program. *Advances in Space Research*, 2025. 2

[5] Davide Celestini, Amirhossein Afsharrad, Daniele Gammelli, Tommaso Guffanti, Gioele Zardini, Sanjay Lall, Elisa Capello, Simone D'Amico, and Marco Pavone. Generalizable spacecraft trajectory generation via multimodal learning with transformers. In *2025 American Control Conference (ACC)*, pages 3558–3565. IEEE, 2025. 1

[6] S. D'Amico. *Autonomous Formation Flying in Low Earth Orbit.* PhD Thesis, Delft University, 2010. 1, 2

[7] Purnanand Elango, Dayou Luo, Abhinav G Kamath, Samet Uzun, Taewan Kim, and Behçet Açıkmeşe. Continuous-time successive convexification for constrained trajectory optimization. *Automatica*, 180:112464, 2025. 1

[8] Matthew Foutter, Daniele Gammelli, Justin Kruger, Ethan Foss, Praneet Bhoj, Tommaso Guffanti, Simone D'Amico, and Marco Pavone. Space-llava: A vision-language model adapted to extraterrestrial applications. In *2025 IEEE Aerospace Conference*, pages 1–23, 2025. 2

[9] Clarence R Gates. A simplified model of midcourse maneuver execution errors. Technical report, 1963. 1

[10] T. Guffanti and S. D'Amico. Passively Safe and Robust Multi-Agent Optimal Control with Application to Distributed Space Systems. *AIAA Journal of Guidance, Control, and Dynamics*, 46(8):1448–1469, 2023. 2

[11] Tommaso Guffanti, Daniele Gammelli, Simone D'Amico, and Marco Pavone. Transformers for trajectory optimization with application to spacecraft rendezvous. In *IEEE Aerospace Conference*, pages 1–13, 2024. 1

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1 (2):3, 2022. 3

[13] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 2

[14] Amit Jain and Richard Linares. Autonomous reasoning for spacecraft control: A large language model framework with group relative policy optimization. *arXiv preprint arXiv:2601.04334*, 2026. 2

[15] Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 2025. 1, 2

[16] Taewan Kim, Purnanand Elango, Danylo Malyuta, and Behçet Açıkmeşe. Guided policy search using sequential convex programming for initialization of trajectory optimization algorithms. In *2022 American Control Conference (ACC)*, pages 3572–3578. IEEE, 2022. 1

[17] Adam W Koenig, Tommaso Guffanti, and Simone D'Amico. New state transition matrices for spacecraft relative motion in perturbed orbits. *Journal of Guidance, Control, and Dynamics*, 40(7):1749–1768, 2017. 1

[18] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020. 1

[19] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Açıkmeşe. Convex Optimization for Trajectory Generation: A Tutorial on Generating Dynamically Feasible Trajectories Reliably and Efficiently. *IEEE Control Systems Magazine*, 42(5):40–113, 2022. 1, 3

[20] Yuanqi Mao, Michael Szmuk, and Behçet Açıkmeşe. Successive convexification of non-convex optimal control problems and its convergence properties. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 3636–3641. IEEE, 2016. 1

[21] Kenshiro Oguri. Successive convexification with feasibility guarantee via augmented lagrangian for non-convex optimal control problems. In *Proc. IEEE Conf. on Decision and Control*, pages 3296–3302. IEEE, 2023. 1

[22] Michael Szmuk, Taylor P Reynolds, and Behçet Açıkmeşe. Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints. *Journal of Guidance, Control, and Dynamics*, 43 (8):1399–1413, 2020. 1

[23] Yuji Takubo, Tommaso Guffanti, Daniele Gammelli, Marco Pavone, and Simone D'Amico. Towards robust spacecraft trajectory optimization via transformers. In *IEEE Aerospace Conference*, 2025. 1, 2

[24] Yuji Takubo, Arpit Dwivedi, Sukeerth Ramkumar, Luis A Pabon, Daniele Gammelli, Marco Pavone, and Simone D'Amico. Semantic trajectory generation for goal-oriented spacecraft rendezvous. In *AIAA SCITECH 2026 Forum*, 2026. 2

[25] Yuji Takubo, Daniele Gammelli, Marco Pavone, and Simone D'Amico. Agile tradespace exploration for space rendezvous mission design via transformers. In *IEEE Aerospace Conference*, 2026. 1

[26] Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025. 1, 2

Intent-aligned Autonomous Spacecraft Guidance via Reasoning Models

Supplementary Material

6. Nonconvex Trajectory Generation

The nonconvex trajectory optimization in Eq. (1) solves for an optimal open-loop trajectory that is robust to the uncertainty. Discrete-time dynamics of the qnsROE space are expressed in Eq. (1b). In particular, $\Phi(t_{j+1}, t_j) \in \mathbb{R}^{6\times 6}$ is the state transition matrix of qnsROE with the secular J_2 effect [17] and $\Gamma_j = \Gamma(t_j) \in \mathbb{R}^{6\times 3}$ is the control input matrix [6].

Furthermore, the nonconvex safety constraint in Eq. (1d) ensures the spacecraft is outside of the ellipsoidal safety domains around the target not only for the entire duration of the controlled trajectory $t \in [0, t_f]$ but also for a predefined duration of the drift trajectories $\tau \in [0, \tau^s]$ after the abrupt loss of control. Formally, this is expressed as the chance-constraint form as [25]:

$$\Pr(x_{j0}^\top S_{ki} x_{j0} \geq 1) \geq 1 - \Delta_j, \tag{2a}$$
$$\Phi_{ji} = \Phi(t_j + \tau_i, t_j), \quad \Psi_{ji} = \Psi(t_j + \tau_i), \tag{2b}$$
$$S_{ki} = (\Psi_{ji}\Phi_{ji})^\top P \Psi_{ji}\Phi_{ji}, \tag{2c}$$
$$x_{j0} = x_j + \Gamma_j u_j, \tag{2d}$$

where $\Psi(t) \in \mathbb{R}^{6\times 6}$ is the first-order mapping between the qnsROE and the relative position and velocity vector resolved in the RTN frame [6], Δ_j is the prescribed risk factor, and $P \in \mathbb{R}^{6\times 6}$ is a diagonal matrix with its components corresponding to the semi-axes of the KOZ in the RTN frame. Under the assumption of Gaussian uncertainty, this chance-constraint is rewritten to a deterministic form as follows:

$$1 - x_{j0}^\top S_{ki} x_{k0} + q(\Delta_k)\sqrt{\boldsymbol{g}_{ki}^\top \Sigma_{ji} \boldsymbol{g}_{ji}} \leq 0 \tag{3a}$$
$$\boldsymbol{g}_{ji} = -2\Phi_{ji}^{-\top}(S_{ji} x_{j0}), \tag{3b}$$
$$\Sigma_{j(i+1)} = \Phi_{ji}\Sigma_{ji}\Phi_{ji}^\top + Q_{ji}, \quad \Phi_{ji} = \Phi(t_j + \tau_{i+1}, t_j + \tau_i), \tag{3c}$$
$$\Sigma_{j0} = \Sigma_{\text{nav}}(x_j) + \Gamma_j \Sigma_{\text{exe}}(u_j)\Gamma_j^\top, \tag{3d}$$

where $q(\cdot)$ is the inverse of the normal cumulative distribution function; Q_{ji} is the constant discrete-time process noise covariance for unmodeled acceleration; $\Sigma_{\text{nav}}(x_j)$ is the navigation uncertainty, and $\Sigma_{\text{exe}}(u_j)$ is the control execution error modeled by the Gates model [9]. The navigation covariance in qnsROE is modeled as a function of the range between the spacecraft and the target $\rho(x_j)$ as:

$$\Sigma_{\text{nav}}(x_j) = \rho(x_j)\boldsymbol{s}\boldsymbol{s}^\top, \tag{4a}$$
$$\boldsymbol{s} = \beta \cdot 10^{-3} \cdot [0.1, 4.0, 2.0, 2.0, 2.0, 2.0][\text{m}], \tag{4b}$$

where scalar variable β defines the navigation accuracy.

7. Waypoint Generation Model

7.1. Dataset generation with waypoint/behavior primitive graph

The generated dataset is constructed from five domains in the qnsROE space, summarized in Table 3. Note that $\delta a = \delta e_x = \delta i_x = 0$ is set so that each waypoint does not have an along-track drift, having no first-order J_2-perturbation, and having the maximal RN-plane separation by leveraging the E/I-vector separation [6]. This reduces the sampling domain from originally a six-dimensional space to a two-dimensional space. Table 4 defines the behavior primitives that enable transitions between these qnsROE domains, while Table 5 specifies the campaign-level qnsROE sequences. Together, these three tables define the waypoint–behavior primitive graph from which waypoint sequences are sampled during dataset generation.

Table 3. Canonical relative orbit domains expressed in the qnsROE space.

Name	$\delta\lambda$	$\delta e_y = \delta i_y$
(a) Central, passively safe orbit	$[-5,\ 5]$	$[30,\ 70]$
(b) +V-bar, passively safe orbit	$[100,\ 250]$	$[30,\ 70]$
(c) +V-bar axis	$[100,\ 250]$	$[-5,\ 5]$
(d) -V-bar, passively safe orbit	$[-250,\ -100]$	$[30,\ 70]$
(e) -V-bar axis	$[-250,\ -100]$	$[-5,\ 5]$

Table 4. Behavior-mode transitions.

Behavior Mode	Transition
(1) Station-Keeping	Self-loop at all nodes: (a)–(e)
(2) Drift +V-dir.	(d)→(a), (a)→(b), (d)→(b)
(3) Drift -V-dir.	(b)→(a), (a)→(d), (b)→(d)
(4) Expand R/N separation	(e)→(d), (c)→(b)
(5) Shrink R/N separation	(b)→(c), (d)→(e)
(6) Approach from -V-bar	(e)→(a)
(7) Retreat to +V-bar	(a)→(c)
(8) Approach from +V-bar	(c)→(a)
(9) Retreat to -V-bar	(a)→(e)
(10) Ducking (fast drift) +V-dir.	(e)→(c)
(11) Ducking (fast drift) -V-dir.	(c)→(e)

In addition to the construction of the waypoint-behavior primitive graph, Table 6 defines the parameter ranges taken for the randomization of the input X.

7.2. Reward model

The reward model used in the waypoint generation model is defined distinctly from the simple fuel cost in Eq. (1a) as

Table 5. Admissible orbital-region sequences by mission type. The symbol "·" denotes no transfer (no-op) and is not counted as a region transition.

Mission Type	Phase 1	Phase 2	Phase 3
(A) Circumnav.	(b|c|d|e)→(a)	(a)→(a)	(a)→(b|c|d|e)
(B) Flyby	(c)→(b) or (b)→·	(b)→(d)	(d)→(e) or (d)→·
	(e)→(d) or (d)→·	(d)→(b)	(b)→(c) or (b)→·
(C) Ducking	(b)→(c) or (c)→·	(c)→(e)	(e)→(d) or (c)→·
	(d)→(e) or (e)→·	(e)→(c)	(c)→(b) or (c)→·

Table 6. Simulation and model parameters.

parameter	value
Δt [s]	900
$N_{\max}$	100 ($\sim t_f \leq 15$ hr.)
r_{KOZ} [m]	$\{20.0,\ 30.0,\ 40.0\}$
β	$\{0.75,\ 1.0,\ 1.25,\ 1.5,\ 2.0\}$
$(t_0) = [a,\ e,\ i,\ \Omega,\ \omega,\ M]$	
a [m]	6.73814×10^6
e	5.581×10^{-4}
i [°]	51.64
Ω [°]	301.04
ω [°]	26.18
M [rad]	linspace(0, 2π, 20)

follows:

$$R(y; X) = -\lambda R_c(\boldsymbol{\tau}^*_{\mathcal{P}(X,y)}) + R_o(\boldsymbol{\tau}^*_{\mathcal{P}(X,y)}), \tag{5}$$

where $\boldsymbol{\tau}^*_{\mathcal{P}(X,y)} = \{x^*_j, u^*_j\}_{j=1}^N$ denotes a minimum-fuel solution of a nonconvex trajectory generation problem $\mathcal{P}(X, y)$; R_c denotes the control cost, R_o denotes the observational reward associated with the trajectory, and λ denotes the weighting hyperparameter. Note that, when desired, this can be further extended and set $\boldsymbol{\tau}^*_{\mathcal{P}(X,y)} = \{x^*_j, u^*_j\}_{j=1}^N$ to be a high-fidelity trajectory, where the navigation and control error based on Monte-Carlo simulation (e.g., ΔV_{99}) are part of the reward function.

The control cost and the observational reward are defined as:

$$R_c = \sum_{j=1}^{N} \|u_j\|_2, \tag{6}$$

$$R_o = -\frac{1}{N} \sum_{j=\min\{j:\rho(x_j)\leq r_{\mathrm{KOZ}}+\Delta r\}}^{\max\{j:\rho(x_j)\leq r_{\mathrm{KOZ}}+\Delta r\}} \rho(x_j). \tag{7}$$

This construction selects the contiguous time interval spanning the first and last epochs at which the ego spacecraft enters the observation region of radius $r_{\mathrm{KOZ}} + \Delta r$. Within this interval, the observational reward increases as the spacecraft remains closer to the target. The cost is then normalized by the total number of timesteps N.

7.3. Training

In order to bias the policy toward higher-reward samples, a weighted imitation learning objective is adopted as:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{B} w_i \log p_\theta\left(y^{(i)} \mid X^{(i)}\right), \tag{8a}$$

$$w_i = \frac{R_i - R_{\min}}{\sum_{j=1}^{B}(R_j - R_{\min})}. \tag{8b}$$

where $w_i \geq 0$ denotes the normalized weight assigned to the i-th sample in a batch of size B. The rewards are shifted by $R_{\min}$ to ensure nonnegativity before normalization.

8. Reasoning Model

8.1. Dataset generation

First, four quantitative evaluation metrics corresponding to the high-level intent components {fuel, time, observation, safety margin} are introduced as follows:

- **Fuel:** R_c
- **Time:** $N = \sum_{k=1}^{K} d_k$
- **Observation quality:** R_o
- **Safety margin:** $\min_j \rho(x_j) - r_{\mathrm{KOZ}}$

Higher values of the observation metric R_o and the safety margin are preferred, whereas lower values of fuel R_c and total time N are desirable.

8.2. Supervised Fine-tuning

The reasoning model is trained based on supervised fine-tuning of a transformer-based pretrained LLM (Qwen2.5-7B-Instruct). To reduce memory and computational overhead, the base model is loaded in 4-bit quantized form, and LoRA layers are applied to the attention projection matrices of the self-attention layers. Training is performed on structured reasoning traces containing a behavior sequence and the total transfer time, formatted as JSON. The model parameters are optimized by minimizing the token-level cross-entropy loss over the serialized output sequence.

9. Supplementary Results

9.1. Waypoint generation

Fig. 3 presents the histogram of rewards obtained from the 500 test cases, corresponding to the summary statistics reported in Table 1. The superior reward distribution achieved by the imitation learning–based methods (both weighted and unweighted) relative to the heuristic baseline is clearly evident.

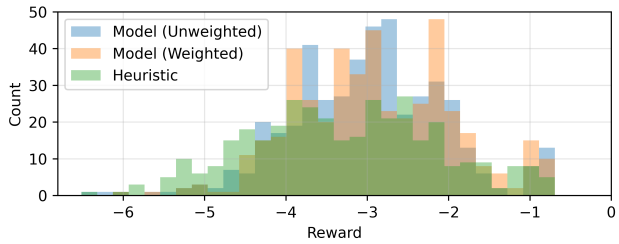


Figure 3. Distribution of the reward function $R(y; X)$ across different waypoint generation models.

10. LLM Prompts

10.1. Annotation of reasoning traces from tabularized metrics

Generation of behavior sequence with reasoning (GPT-4o-mini)

```
System:
You're an expert spacecraft operator for rendezvous
     missions.
You select one trajectory candidate from metric
     tables.
Follow the priority order (lexicographic), not
     weighted sum.
Output only valid JSON.
For one_line_reason, use probabilistic wording to
     describe why the selected candidate seems
     favorable based on its metrics and the intent
     priority.
Avoid absolute superlatives or explicit comparisons,
     as the candidates may have tradeoffs and there
     are no guarantees.
Focus on the strengths of the chosen candidate in
     relation to the mission intent, without
     directly stating it is the best or comparing it
     to others.

User:
Priority order: <intent_1> > <intent_2> > <intent_3>
     > <intent_4>
Metrics: <metric_1>, <metric_2>, <metric_3>,
     <metric_4>
Rules:
- Lower is better for fuel_dv and time_sec.
- Higher is better for obs and safety_margin.
- All candidates are already safe; safety_margin is
     a metric of conservatism.

Candidates CSV:
id,policy,fuel_dv,time_sec,obs,safety_margin
<...rows...>

Return JSON with keys: {"best_candidate_id": <int>,
     "one_line_reason": "<short sentence>"}

Reasoning style constraints:
- one_line_reason must be exactly one short sentence.
- Do not mention candidate IDs or names.
- Avoid comparative/superlative words: lower,
     higher, lowest, highest, better, best, worse,
     worst, more, less.
- Avoid ranking symbols or explicit comparisons: >,
     <, >=, <=, versus, than.
- Prefer probabilistic phrasing like: it seems to
     lead to low delta-v, is expected to keep
     transfer time short, has a high chance of
     supporting observation.
```

10.2. Generation of behavior sequence with a reasoning trace

Generation of behavior sequence with reasoning (Qwen2.5-7B-Instruct)

```
System:
You are an assistant that selects trajectory-level
     decisions from structured mission context.
Follow intent priority and constraints, reason
     briefly, and return a strict JSON answer.

User:
You're an expert spacecraft operator for rendezvous
     missions.
Task: choose (b_seq, tf) based on mission context
     and intent priority.
Then provide one-line reasoning and justification.

x0_roe_m = <x0>
r_koz = <r_koz>
beta = <beta>
intent_priority = <intent_priority>

Return JSON with keys: reasoning, tf, b_seq.
```

Structured output from the reasoning model (JSON format)

```
<|think|>
<reasoning text>
<|answer|>
{"reasoning":"...","tf":...,"b_seq":[...]}
<|end|>
```

10.3. Extraction of two top performance metrics from the generated reasoning trace

LLM prompt for metric extraction (GPT-4o-mini)

```
System:
Extract metric names exactly from the allowed list.

User:
From the reasoning sentence, select up to two
     metrics mentioned in order of appearance.
Allowed metrics: fuel_dv, transfer_time_sec,
     observation_score, safety_margin_m.
Some words to check for each metric:
- fuel_dv: fuel, delta-v, control cost
- transfer_time_sec: time, transfer time, tof
- observation_score: observation
- safety_margin_m: safety, safety margin, clearance
Return strict JSON only: {"focused_metrics": ["..."]}
Reasoning sentence:
<REASONING_SENTENCE>
```