This paper has been accepted for publication at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Pittsburgh, 2026. ©IEEE

Learning Acrobatic Flight from Preferences

Colin Merk*,[1], Ismail Geles*,[1], Jiaxu Xing[1], Angel Romero[1], Giorgia Ramponi[2], and Davide Scaramuzza[1]

***Abstract*— Preference-based reinforcement learning (PbRL) enables agents to learn control policies without requiring manually designed reward functions, making it well-suited for tasks where objectives are difficult to formalize or inherently subjective. Acrobatic flight poses a particularly challenging problem due to its complex dynamics, rapid movements, and the importance of precise execution. However, manually designed reward functions for such tasks often fail to capture the qualities that matter: we find that hand-crafted rewards agree with human judgment only 60.7% of the time, underscoring the need for preference-driven approaches. In this work, we propose Reward Ensemble under Confidence (REC), a probabilistic reward learning framework for PbRL that explicitly models per-timestep reward uncertainty through an ensemble of distributional reward models. By propagating uncertainty into the preference loss and leveraging disagreement for exploration, REC achieves 88.4% of shaped reward performance on acrobatic quadrotor control, compared to 55.2% with standard Preference PPO. We train policies in simulation and successfully transfer them zero-shot to the real world, demonstrating complex acrobatic maneuvers learned purely from preference feedback. We further validate REC on a continuous control benchmark, confirming its applicability beyond the domain of aerial robotics.**

Supplementary Materials

Video: `https://youtu.be/JiMW_ligKvQ`

I. Introduction

Autonomous drones capable of executing dynamic and agile maneuvers have become a benchmark for progress in robotics, with applications ranging from high-speed navigation [1] and search-and-rescue to inspection [2]. Reinforcement learning (RL) has shown promise in enabling such high-performance control policies, even surpassing expert human pilots in competitive environments [1]. However, a persistent challenge in applying RL to real-world tasks is the design of effective reward functions.

Reward design is not only time-consuming and task-specific, but it also introduces a fundamental bottleneck: many tasks—particularly those involving aesthetics, subjective quality, or high-level intent—are difficult or even impossible to express with a well-defined reward function. This limitation is especially critical in acrobatic flight maneuvers, where desirable behavior may depend on human preferences over trajectory smoothness, timing, or style.

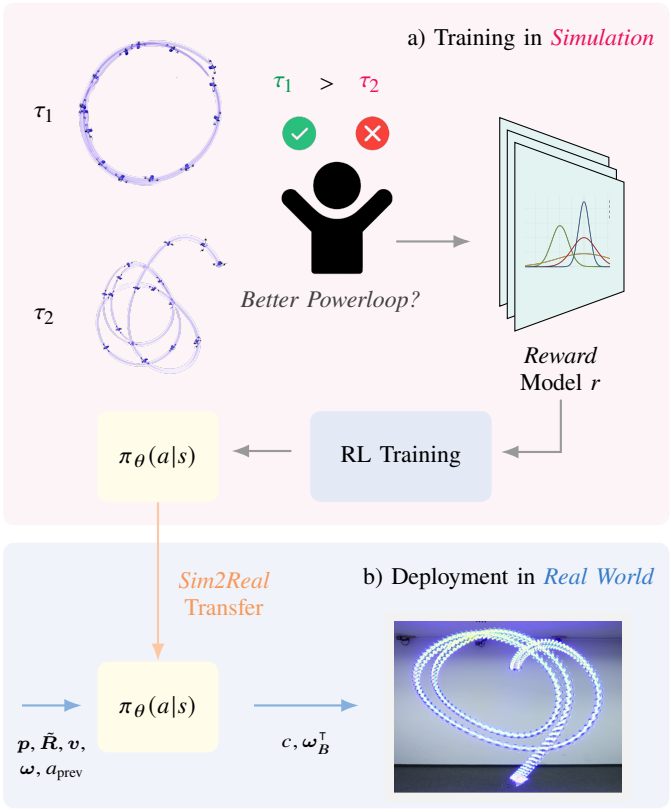


Fig. 1. Overview of the proposed approach. (a) In simulation, trajectory pairs (τ_1, τ_2) are presented to an annotator (human or synthetic) to collect preference labels. These labels train an ensemble of reward models with uncertainty estimation, which provides the reward signal for policy optimization via reinforcement learning. (b) The resulting policy is transferred zero-shot to a real quadrotor to execute acrobatic maneuvers.

Acrobatic flight compounds this difficulty due to its highly nonlinear dynamics, rapid state transitions, and narrow margins for error. Small differences in execution can distinguish a smooth, visually compelling maneuver from a jerky or failed one, yet formalizing such distinctions in a reward function requires extensive domain expertise and iterative tuning. Even carefully engineered rewards may fail to reflect what observers actually prefer: we show that hand-crafted rewards agree with human judgment only 60.7% of the time.

To address this, preference-based reinforcement learning (PbRL) has emerged as a compelling alternative. Rather than specifying explicit rewards, PbRL infers a reward function from comparisons between trajectory segments [3]. This approach enables policy learning for tasks that are hard to define but easy to evaluate by preference. Each acrobatic skill would typically require its own carefully designed reward function in traditional RL, whereas PbRL offers a more general framework that can adapt across different maneuvers

*These authors contributed equally to this work.

[1]These authors are with the Robotics and Perception Group, University of Zurich, Switzerland (`https://rpg.ifi.uzh.ch`). Contact: `geles@ifi.uzh.ch`.

[2]Giorgia Ramponi is with the Autonomous Learning and Predictive Intelligence Lab, University of Zurich, Switzerland.

Supported by the European Union's Horizon Europe programme (grant No. 101120732, AUTOASSESS) and the European Research Council (ERC) (grant No. 864042, AGILEFLIGHT).

without task-specific reward engineering. Unlike imitation learning, which requires high-quality demonstrations, PbRL only requires comparative judgments between behaviors, decoupling the quality of the learned policy from the skill level of the feedback source.

Despite the promise of PbRL, its application in real-world robotics has been limited. Most prior work has focused on simulated benchmarks, and applications to physical systems, particularly agile drones, remain scarce. Existing methods also tend to overlook the uncertainty inherent in preferences, which can lead to instability or suboptimal learning when feedback is noisy or sparse. Without accounting for this uncertainty, reward models can overfit to ambiguous preference labels, leading to brittle policies that may fail to deploy.

In this work, we propose *Reward Ensemble under Confidence* (REC), a probabilistic preference-based reinforcement learning framework for acrobatic drone flight. REC introduces an ensemble of distributional reward models that explicitly capture per-timestep reward uncertainty and propagate it through the preference loss via a Gaussian cumulative distribution function.

The key insight is that preference feedback is inherently probabilistic: when two trajectories are similar in quality, the preferred choice is noisy. Modeling each reward as a distribution lets REC express confidence in its predictions, and the resulting ensemble disagreement drives exploration toward reward regions where the model is least certain, yielding more stable training under limited preference supervision.

We validate our approach through simulation and real-world experiments on acrobatic quadrotor control, as well as on a standard continuous control benchmark. Our contributions are as follows:

1) We propose REC, a probabilistic reward learning framework that models per-timestep reward uncertainty within an ensemble of reward models, replacing the standard Bradley-Terry softmax with a distributional preference model.
2) We demonstrate that REC achieves 88.4% of shaped reward performance on acrobatic quadrotor control, compared to 55.2% with standard Preference PPO, representing a substantial improvement in preference-based learning for agile flight.
3) We successfully transfer learned policies zero-shot to a real 220 g quadrotor, executing acrobatic maneuvers including continuous powerloops and a novel vertical Figure-8 learned purely from preference feedback.
4) We show that hand-crafted reward functions agree with human judgment only 60.7% of the time, highlighting the limitations of manual reward engineering for tasks with subjective objectives.

II. Related Works

Preference-Based Reinforcement Learning. Preference-based reinforcement learning (PbRL) enables agents to learn policies directly from comparisons between trajectory segments, typically provided by human evaluators, bypassing the need for manually designed reward functions. This approach was introduced by [3] and has since inspired numerous advances. Furthermore, limitations in time and sample efficiency have been mitigated through approaches that actively generate pairwise comparisons guided by information gain [4], by selecting batches of comparison pairs [5], and active volume removal [6]. Recent efforts include enhancing exploration efficiency through unsupervised pretraining [7], incorporating transformer-based reward models to capture temporal and non-Markovian aspects of human feedback [8], and exploring implicit reward learning via Q-functions without explicitly modeled rewards [9]. Similarly, [10] leverage reward uncertainty from ensemble disagreement to drive exploration in preference-based learning.

Real-World Deployments of PbRL. Despite its success in simulation, PbRL has seen limited application in real-world robotics, though recent studies show growing interest. One line of research improves sample efficiency by leveraging pre-trained preference models from prior tasks, enabling adaptation to novel tasks on robotic manipulators [11], or using priors obtained from pre-trained LLMs for higher quality candidates in PbRL to achieve expressive behaviors on a quadruped [12]. Another approach combines demonstrations with preference queries to address the inefficiencies of standard PbRL and the limitations of inverse reinforcement learning [13].

Reinforcement Learning in Quadrotor Control. Reinforcement learning has significantly advanced quadrotor control capabilities, with remarkable successes in drone racing [1], [14], robust maneuvering under disturbances [15], and agile flight tasks [16], [17]. Recent works have further demonstrated vision-based RL without using any state information [18], [19]. However, most of these methods rely on manually engineered reward functions. To the best of our knowledge, PbRL has not been applied to quadrotor acrobatics. Our work addresses this gap, demonstrating both simulation efficacy and real-world feasibility using preference feedback from both synthetic and human sources.

III. Methodology

A. Preliminaries

We consider a standard RL setting in which an agent takes an action a_t after receiving an observation o_t and obtains a reward $r_t = r(o_t, a_t)$. In traditional RL this reward is hand-crafted, whereas PbRL replaces it with a reward model trained from preference labels over pairs of trajectories. In online PbRL these labels are collected by repeatedly presenting trajectory pairs to an annotator who selects the preferred one, alternating with policy optimization to iteratively improve the agent's behavior.

Trajectory. We denote a trajectory τ as a sequence of observation–action pairs (o_i, a_i) of length N:

$$\tau = \big((o_1, a_1), (o_2, a_2), \ldots, (o_N, a_N)\big), \tag{1}$$

where o_i is the observation at timestep i and a_i the corresponding action. The total reward of a trajectory $r(\tau)$ is the sum of the rewards at each timestep:

$$r(\tau) = \textstyle\sum_{i=1}^{N} r_i. \tag{2}$$

The reinforcement learning objective can be formalized as maximizing the expected sum of discounted rewards:

$$\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[r(\tau)\right] = \max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{i=1}^{N} \gamma^i r_i\right], \tag{3}$$

where π denotes the policy, $\tau \sim \pi$ represents trajectories sampled from the policy, and $\gamma \in [0, 1]$ is the discount factor.

B. Quadrotor System Model

We consider the problem of learning acrobatic flight policies for a quadrotor platform. We use Flightmare [20] and Agilicious [21] for training in simulation and deploying in the real world. The quadrotor dynamics are modeled as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{p}}_{\mathcal{WB}} \\ \dot{\boldsymbol{q}}_{\mathcal{WB}} \\ \dot{\boldsymbol{v}}_{\mathcal{W}} \\ \dot{\boldsymbol{\omega}}_{\mathcal{B}} \\ \dot{\boldsymbol{\Omega}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_{\mathcal{W}} \\ \boldsymbol{q}_{\mathcal{WB}} \cdot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{\mathcal{B}}/2 \end{bmatrix} \\ \frac{1}{m}\left(\boldsymbol{q}_{\mathcal{WB}} \odot \left(\boldsymbol{f}_{\text{prop}} + \boldsymbol{f}_{\text{aero}}\right)\right) + \boldsymbol{g}_{\mathcal{W}} \\ \boldsymbol{J}^{-1}\left(\boldsymbol{\tau}_{\text{prop}} + \boldsymbol{\tau}_{\text{aero}} - \boldsymbol{\omega}_{\mathcal{B}} \times \boldsymbol{J}\boldsymbol{\omega}_{\mathcal{B}}\right) \\ \frac{1}{k_{\text{mot}}}\left(\boldsymbol{\Omega}_{\text{ss}} - \boldsymbol{\Omega}\right) \end{bmatrix},$$

where $\odot$ denotes quaternion-based rotation, $\boldsymbol{p}_{\mathcal{WB}}$, $\boldsymbol{q}_{\mathcal{WB}}$, $\boldsymbol{v}_{\mathcal{W}}$, and $\boldsymbol{\omega}_{\mathcal{B}}$ are the vehicle's position, orientation, velocity, and angular rates, $\boldsymbol{J}$ is the inertia tensor, k_{mot} the motor time constant, and $\boldsymbol{\Omega}$, $\boldsymbol{\Omega}_{\text{ss}}$ the current and commanded rotor speeds. Propulsion and aerodynamic effects are captured by $\boldsymbol{f}_{\text{prop}}$, $\boldsymbol{\tau}_{\text{prop}}$ and $\boldsymbol{f}_{\text{aero}}$, $\boldsymbol{\tau}_{\text{aero}}$. We employ a quadratic thrust model with data-driven corrections following [1]; see [22] for further details.

The policy observation at time t consists of position $p_t \in \mathbb{R}^3$, linear velocity $v_t \in \mathbb{R}^3$, the first two columns of the rotation matrix $R_t \in \mathbb{R}^6$, angular velocity $\omega_t \in \mathbb{R}^3$, and the previous action $a_{t-1} \in \mathbb{R}^4$. The action space consists of collective thrust and body rates (CTBR), $a_t \in \mathbb{R}^4$. The shaped reward used for the PPO baseline and synthetic preference generation is described in Appendix A.

C. Preference-based Reinforcement Learning

To formalize preference-based reinforcement learning, we first define the concepts of *Preference* and *Preference Modeling*, *Cross Entropy Loss*, and *Synthetic Preference Generation*.

Preference. A preference label $p \in [0, 1]$ is assigned by a *judge* to a pair of trajectories, reflecting the relative preference between them. We assume that preferences are determined by an underlying reward function $r : \tau \to \mathbb{R}$, where

$$p(\tau_1, \tau_2) = \begin{cases} 0, & \text{if } r(\tau_1) > r(\tau_2), \\ 0.5 & \text{if } r(\tau_1) = r(\tau_2), \\ 1, & \text{if } r(\tau_1) < r(\tau_2). \end{cases} \tag{4}$$

Preference Modeling. A commonly used model of preference judgment, as is also being done in [3], is viewing the reward r of a trajectory τ as a latent variable that determines the choices of the annotator. More precisely, the probability of preferring one trajectory over the other is given by the softmax of the two rewards,

$$\hat{p}(\tau_1 > \tau_2) = \frac{\exp(\hat{r}_1)}{\exp(\hat{r}_1) + \exp(\hat{r}_2)}, \tag{5}$$

which is known as the Bradley–Terry (BT) model. In our notation, we refer to a predicted variable by $\hat{x}$, whereas ground truth variables are referred to as x.

Cross Entropy Loss. In Preference PPO [23], the reward model $\hat{r}$ is optimized by minimizing the cross entropy loss between the BT model predictions and the true labels, using the following loss:

$$\begin{aligned} \text{loss}(\hat{r}) = -\sum_{(\tau_1^i, \tau_2^i, p^i)\in D} \Big[& p^i \cdot \log(\hat{p}(\tau_1^i > \tau_2^i)) \\ & + (1 - p^i) \cdot \log(\hat{p}(\tau_2^i > \tau_1^i)) \Big]. \end{aligned} \tag{6}$$

Synthetic Preference Generation. To assess the performance of our preference-based RL method, we use synthetic preferences, which serve as an oracle representing preferences based on the total handcrafted reward of each compared trajectory in the training environment [3]. The preferred trajectory is the one that achieves a higher reward in the given task. The detailed reward functions of each environment are described in Appendix A.

1) Unsupervised Pretraining: As presented in [7] the pretraining step is used to improve the initial exploration of the policy. In preference-based learning, this is especially useful, as there is no guidance for the policy before the reward model has been trained. To train the reward model, some labeled trajectories are necessary. To encourage exploration, an intrinsic reward is given based on the state entropy $H(s) = \mathbb{E}_{s\sim\rho(s)}[\log(\rho(s))]$. As this is intractable, a particle-based entropy estimator [24] is used:

$$\hat{H}(s) \sim \sum_i log(\|s_i - s_i^k\|). \tag{7}$$

Finally, the intrinsic reward for unsupervised exploration r^{int} is given by the distance to the k-th nearest neighbor of the current state s_t:

$$r^{int}(s_t) = log(\|s_t - s_t^k\|). \tag{8}$$

2) Query and Reward Training Scheduling: Querying (collecting new preferences) and reward-model retraining follow the same schedule: the reward model is retrained each time new preferences are gathered. Schedules are specified in terms of n_{steps}, the number of agent-environment interaction steps, and we define an epoch as the interval between two consecutive retraining phases. The preference-based hyperparameters are listed in Table B1 in Appendix B.

3) Query Selection: We select and pair trajectories for preference comparisons using three distinct heuristics. Each method contributes one-third of the total pairs, and each trajectory is used at most once.

a) Random: Pairs are selected at random. This serves as the default during the initial pairing phase before the reward model has been trained.

b) Ensemble Disagreement: All possible pairs are generated from the available trajectories, and each pair receives a preference label from every ensemble member. Pairs are then sorted in descending order by the number of ensemble members that disagree, and the highest-disagreement pairs are selected such that no trajectory is repeated.

c) Current with Previous Epoch: Here, an epoch refers to the interval between two reward retraining phases. A trajectory generated after the most recent reward update is paired with one generated before it.

D. REC Preference PPO

In this section, we introduce REC, a probabilistic reward learning framework that explicitly models uncertainty in the predicted reward. REC consists of three components: a probabilistic loss function for the reward model, an uncertainty-aware reward aggregation strategy, and an ensemble resetting mechanism. The full training procedure is summarized as pseudocode in Appendix C.

1) Loss Function:

a) Probabilistic Cross Entropy Loss.: For the reward model, we use an ensemble of n_{ensemble} (denoted n for brevity) multi-layer perceptrons. Each ensemble member takes the current observation-action pair (o_t, a_t) as input and outputs a scalar reward prediction $r^i_{t,\text{pred}}$. The distributional reward parameters are obtained from the ensemble statistics:

$$r_{\text{mean}}(o_t, a_t) = \frac{1}{n}\sum_{i=1}^{n} r^i_{t,\text{pred}}, \quad r_{\text{std}}(o_t, a_t) = \text{std}\left(\{r^i_{t,\text{pred}}\}_{i=1}^{n}\right). \tag{9}$$

We model the reward at each timestep as a sample from a normal distribution parameterized by these ensemble statistics:

$$r(o_t, a_t) \sim \mathcal{N}\big(r_{\text{mean}}(o_t, a_t), r_{\text{std}}(o_t, a_t)\big). \tag{10}$$

The reward for the whole trajectory is then given by

$$r(\tau) \sim \sum_t \mathcal{N}\big(r_{\text{mean}}(o_t, a_t), r_{\text{std}}(o_t, a_t)\big). \tag{11}$$

Assuming independence across timesteps, we can rewrite this as:

$$\begin{aligned} r(\tau) &\sim \mathcal{N}\left(\sum_t r_{\text{mean}}(o_t, a_t), \sqrt{\sum_t r_{\text{std}}(o_t, a_t)^2}\right) \\ &= \mathcal{N}\Big(r_{\text{mean}}(\tau), r_{\text{std}}(\tau)\Big). \end{aligned} \tag{12}$$

Given the distribution for the trajectory reward $r(\tau)$, we can model the preference between two trajectories. Here, Φ denotes the cumulative distribution function (CDF) of the standard normal, which gives the probability that a sample from $r(\tau_1)$ is larger than one from $r(\tau_2)$:

$$p(\tau_1 > \tau_2) = \Phi\left(\frac{r_{\text{mean}}(\tau_1) - r_{\text{mean}}(\tau_2)}{\sqrt{r_{\text{std}}(\tau_1)^2 + r_{\text{std}}(\tau_2)^2}}\right). \tag{13}$$

Note that this replaces the Bradley-Terry softmax in Eq. 6 with a Gaussian CDF preference model that naturally incorporates reward uncertainty. When ensemble members agree and the standard deviations approach zero, this formulation recovers the deterministic preference model. We derive the preference loss by minimizing the cross-entropy with respect to this probabilistic preference:

$$\begin{aligned} \text{loss}(\hat{r}) = -\sum_{(\tau_1^i, \tau_2^i, p^i) \in D} \Big[& p^i \log \hat{p}(\tau_1^i > \tau_2^i) \\ & + (1 - p^i) \log\left(1 - \hat{p}(\tau_1^i > \tau_2^i)\right)\Big]. \end{aligned} \tag{14}$$

b) Standard Deviation Target Loss.: To encourage more stable standard deviation estimates across ensemble members, we introduce an additional loss term. This term penalizes the squared error between a target standard deviation σ_{target} and the mean standard deviation across all timesteps in the preference dataset, $\bar{\sigma}$:

$$\text{loss}_\sigma = (\bar{\sigma} - \sigma_{target})^2. \tag{15}$$

Without this regularization, ensemble members can collapse to identical predictions with near-zero standard deviation, effectively recovering the deterministic Bradley-Terry model and negating the benefits of probabilistic modeling.

2) Reward Aggregation: To provide the reward signal for policy optimization, we aggregate the ensemble predictions using a noisy aggregation method that increases the reward in regions of high uncertainty, following the approach of [10]:

$$r^{\text{agg}}_{t,\text{pred}} = \frac{1}{n}\sum_{i=1}^{n} r^i_{t,\text{pred}} + |X|, \tag{16}$$

where X is drawn from:

$$X \sim \mathcal{N}\left(0, \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(r^i_{t,\text{pred}} - \frac{1}{n}\sum_{j=1}^{n} r^j_{t,\text{pred}}\right)^2}\right). \tag{17}$$

The absolute value ensures that ensemble disagreement always results in a positive reward bonus, encouraging the policy to visit states where the reward model is uncertain. This drives exploration toward regions where additional preference feedback would be most informative.

3) Ensemble Resetting: Before each reward model retraining, we evaluate each ensemble member separately on the newly added preferences. We then re-initialize the weights of the worst-performing $n_{reset} \in [0, \ldots, n_{\text{ensemble}}]$ ensemble members. This prevents poorly performing members from corrupting the ensemble statistics and helps maintain diversity among the ensemble, which is essential for meaningful uncertainty estimates. Without resetting, ensemble members can converge to similar predictions over time, reducing the effectiveness of both the probabilistic loss and the uncertainty-driven exploration. Unless otherwise specified, we use $n_{reset} = 1$ in our experiments.

IV. Results

A. DM Control Evaluation

To validate REC beyond aerial robotics, we evaluate on the *walker-walk* task from the DM Control Suite [25], using the

implementation of Tai et al. [26]. We ablate each component of REC with respect to Preference PPO across 3 seeds, using the same hyperparameters as in [23]. Fig. 2 shows the mean episode reward during training, and Table I reports the best evaluation reward per method. Each component improves over the Preference PPO baseline, with the probabilistic loss and reward noise yielding the largest gains. Ensemble resetting slightly reduces mean reward on this task but decreases variance from ±55.5 to ±43.3, indicating more consistent training. Its benefit is more pronounced on the quadrotor task (Table II), where maintaining ensemble diversity has greater impact.

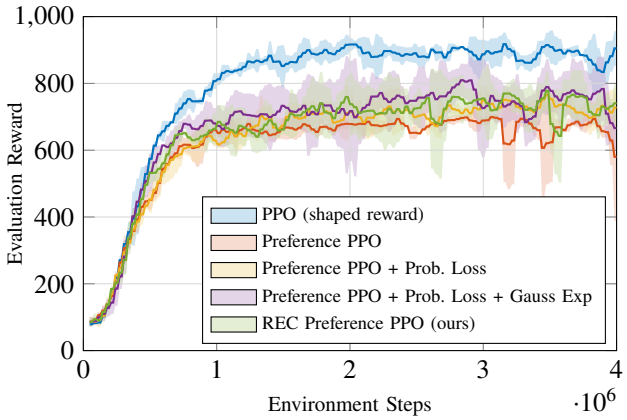


Fig. 2. Average reward on the *walker-walk* task with 1000 synthetic preferences while training. Shaded areas denote standard deviation across 3 seeds. PPO with the shaped environment reward serves as the upper baseline. Ablation components are introduced incrementally to obtain REC Preference PPO.

TABLE I

Maximum evaluation reward during training on the *walker-walk* task. Values represent the highest mean evaluation reward across three seeds (± standard deviation). Components of REC are incrementally introduced starting from the Preference PPO baseline.

Method	**Max. Eval. Rew.**
PPO	930.7 ± 12.0
Preference PPO	719.0 ± 9.3
Preference PPO + Prob. Rew. Loss	777.5 ± 66.2
Preference PPO + Prob. Rew. Loss + Rew. Noise	829.4 ± 55.5
Preference PPO + Prob. Rew. Loss + Rew. Noise + Reset Ensemble — (REC)	815.1 ± 43.3

B. Evaluation on the Quadrotor

We evaluate four training configurations in the Flightmare [20] simulator: PPO with the shaped reward, Preference PPO, and REC Preference PPO with both synthetic and human preferences. Stop-motion visualizations of the best rollouts are shown in Fig. 3, and Table II reports the mean best evaluation reward across three seeds. Fig. 5 shows the evaluation reward over training for the continuous powerloop task using 1000 synthetic preferences. REC Preference PPO achieves a mean evaluation reward of 382.4 ± 80.8, corresponding to 88.4% of the shaped reward baseline (432.4 ± 190.8), compared to 238.9±157.5 (55.2%) for standard Preference PPO. Notably, REC also exhibits substantially lower variance across seeds, indicating more reliable convergence on this challenging exploration problem.

TABLE II

Maximum evaluation reward during training of the continuous powerloop task. Values represent the highest mean evaluation reward across three seeds (± standard deviation). Components of REC are incrementally introduced starting from the Preference PPO baseline.

Method	**Max. Eval. Rew.**
PPO (shaped reward)	432.4 ± 190.8
Preference PPO	238.9 ± 157.5
Preference PPO + Prob. Rew. Loss	281.0 ± 187.4
Preference PPO + Prob. Rew. Loss + Reward Noise	153.9 ± 174.2
Preference PPO + Prob. Rew. Loss + Rew. Noise + Reset Ensemble — (REC)	382.4 ± 80.8

C. Preference Feedback from a Human Annotator

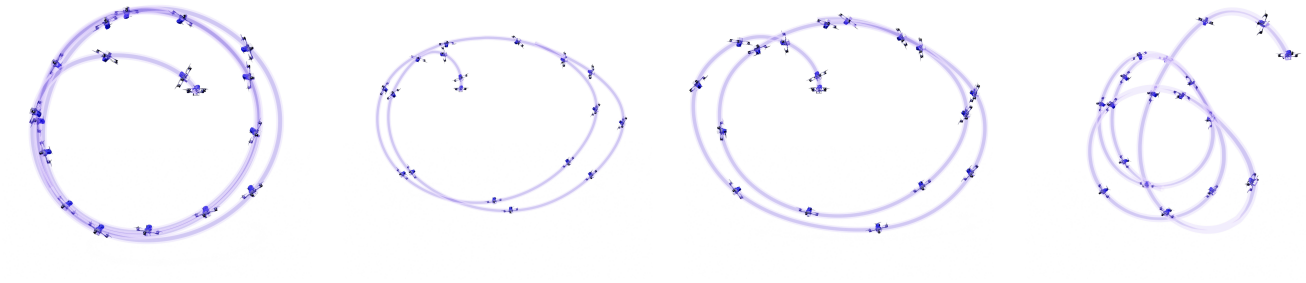
We evaluate REC Preference PPO using preference labels provided by a human annotator. The experimental settings are identical to the synthetic preference experiments, with the annotator comparing trajectory pairs rendered from the simulator. The annotator has the additional option of responding *'I can't tell'*, in which case the pair is discarded and replaced. Of the 1064 comparisons shown, 1000 valid preference labels were collected. The complete training and labeling process took 4 h 43 min. Since no ground-truth reward is assumed to be available in this setting, the policy checkpoint for evaluation and deployment is selected by the annotator rather than by maximum evaluation reward. In this study, the human annotator is the same person who designed the shaped reward used for the synthetic experiments. The disagreement reported below therefore arises despite full familiarity with the intended task, which reinforces that it stems from the limitations of hand-crafted rewards rather than annotator inexperience; studying inter-annotator agreement with independent evaluators remains future work. Table III reports

TABLE III

Agreement between the shaped reward function and the human annotator. Rows indicate the shaped reward's preference, columns the human's choice. The shaped reward never produces ties by construction.

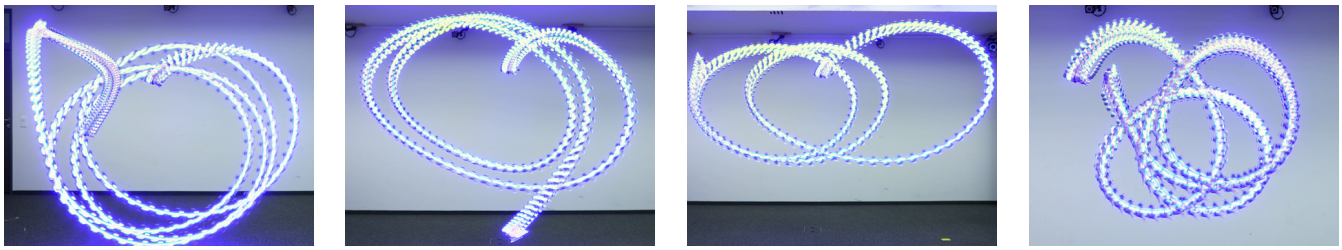
	Human: τ_1	**Human: τ_2**	**Human: tie**
Reward: τ_1	323	167	37
Reward: τ_2	196	238	39

the agreement between the annotator's preferences and those implied by the shaped reward; excluding ties, the two agree only 60.7% of the time. As discussed in Section IV-E, this reflects the limited ability of the shaped reward to capture what an observer values when evaluating acrobatic maneuvers rather than poor annotation quality. Despite this



(a) PPO (shaped reward) (b) Preference PPO (synthetic) (c) REC Preference PPO (synthetic, our algorithm) (d) REC Preference PPO (human, our algorithm)

Fig. 3. Stop-motion visualizations of the highest-reward evaluation rollout in simulation for each training configuration. (a) PPO with shaped rewards. (b) Preference PPO with 1000 synthetic preferences. (c) REC Preference PPO with 1000 synthetic preferences. (d) REC Preference PPO with 1000 human-labeled preferences.



(a) PPO (shaped reward) (b) Preference PPO (synthetic) (c) REC Preference PPO (synthetic, our algorithm) (d) REC Preference PPO (human, our algorithm)

Fig. 4. Long-exposure photographs of real-world deployment across four training configurations. (a) PPO with shaped rewards. (b) Preference PPO with 1000 synthetic preferences. (c) REC Preference PPO with 1000 synthetic preferences. (d) REC Preference PPO with 1000 human-labeled preferences on the powerloop task.

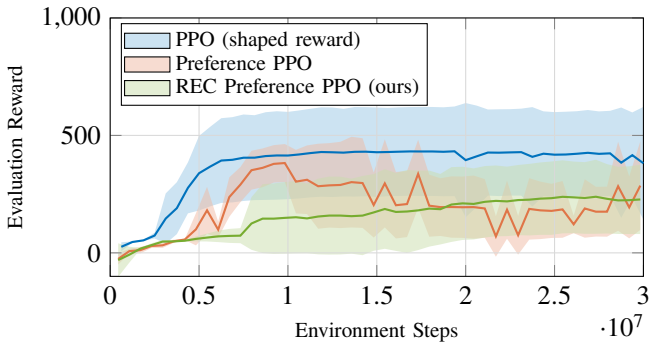


Fig. 5. Mean evaluation reward over 3 seeds during training of the continuous powerloop with 1000 synthetic preferences. Shaded areas indicate standard deviation.

misalignment, the policy trained from human preferences executes the powerloop both in simulation and on the real platform (Section IV-D), indicating that preference feedback captures task-relevant information that the shaped reward does not fully encode.

D. Real-World Deployment and Novel Acrobatic Skills

We deploy the simulation-trained policies on a 220 g quadrotor with a static thrust-to-weight ratio of 6.5, commanding collective thrust and body rates (CTBR) at 50 Hz. For configurations trained with synthetic preferences, we select the checkpoint with the highest evaluation reward. For the human preference experiment, the annotator selects the checkpoint. All policies are transferred zero-shot without any real-world fine-tuning. Deployment results for each training configuration are shown in Fig. 4. All four configurations successfully execute continuous powerloop maneuvers on the real platform, demonstrating that both synthetic and human preferences enable sim-to-real transfer of agile flight. To show the framework's generality, we further train a novel vertical Figure-8 (double powerloop) from human preferences alone, without changing any hyperparameters. The resulting policy, shown in Fig. 6, executes this maneuver on the real drone, illustrating how preference feedback lets non-expert users specify new skills without reward engineering.

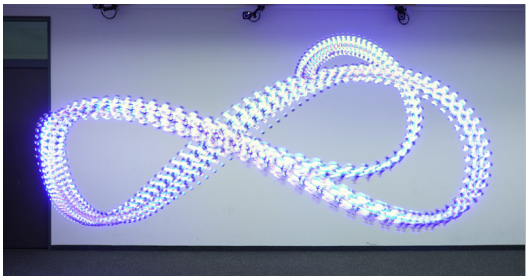


Fig. 6. Vertical Figure-8 (double powerloop) trained purely from human preference feedback using REC Preference PPO without any handcrafted reward function.

E. Discussion

The ablations on both walker-walk and the quadrotor indicate that the probabilistic loss and reward noise contribute the largest individual gains, while ensemble resetting primarily improves consistency; its benefit scales with task difficulty, having a modest effect on walker-walk but a larger one on the quadrotor, where agents frequently fail to explore beyond an initial half-flip. The 60.7% agreement between the shaped reward and human preferences does not indicate poor annotation but rather a fundamental limitation of manual reward engineering: observers judge maneuvers holistically through qualities such as smoothness, timing, and visual impression that are difficult to encode as Markovian terms, yet the human-trained policy still transfers successfully to the real platform. Human evaluation also has limits of its own: annotators perceive geometric trajectory properties but not control-level quantities such as body rates or energy, and the assessment is viewpoint-dependent, so multi-perspective evaluation or control-aware visual feedback is a promising direction. Finally, the high seed variance reflects the exploration difficulty of acrobatic tasks in the absence of a shaping reward, which curriculum or more targeted exploration strategies could mitigate.

V. Conclusion

In this work, we propose REC, a probabilistic reward learning framework for preference-based reinforcement learning that explicitly models reward uncertainty through ensemble distributional predictions. Applied to acrobatic quadrotor control, REC achieves 88.4% of shaped reward performance compared to 55.2% for standard Preference PPO, while reducing training variance across seeds. Policies trained with both synthetic and human preference feedback transfer zero-shot to a real 220 g quadrotor, executing continuous powerloops and a novel vertical Figure-8 maneuver without any manually designed reward function or changes in hyperparameters. The observed 60.7% agreement between shaped rewards and human preferences further indicates that manual reward engineering may be insufficient for tasks where desired behavior is more naturally expressed through comparative judgments.

Several directions remain for future work. The preference collection process would benefit from multi-annotator studies and reduced viewpoint dependency to better characterize the consistency and robustness of preference signals. More broadly, extending REC to a wider range of agile flight maneuvers and robotic domains beyond quadrotors would further establish the generality of probabilistic preference-based learning.

References

[1] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, Aug. 2023, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41586-023-06419-4

[2] J. Xing, G. Cioffi, J. Hidalgo-Carrió, and D. Scaramuzza, "Autonomous power line inspection with drones via perception-aware mpc," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1086–1093.

[3] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[4] E. Biyik, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh, "Asking easy questions: A user-friendly approach to active reward learning," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 1177–1190. [Online]. Available: https://proceedings.mlr.press/v100/b-iy-ik20a.html

[5] E. Biyik, N. Anari, and D. Sadigh, "Batch active learning of reward functions from human preferences," *ACM Transactions on Human-Robot Interaction*, vol. 13, no. 2, pp. 1–27, 2024.

[6] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions," in *Robotics: Science and Systems*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:12226563

[7] K. Lee, L. M. Smith, and P. Abbeel, "Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 6152–6163. [Online]. Available: https://proceedings.mlr.press/v139/lee21i.html

[8] C. Kim, J. Park, J. Shin, H. Lee, P. Abbeel, and K. Lee, "Preference transformer: Modeling human preferences using transformers for RL," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Peot1SFDX0

[9] J. Hejna and D. Sadigh, "Inverse preference learning: Preference-based rl without a reward function," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 18 806–18 827. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/3be7859b36d9440372cae0a293f2e4cc-Paper-Conference.pdf

[10] X. Liang, K. Shu, K. Lee, and P. Abbeel, "Reward uncertainty for exploration in preference-based reinforcement learning," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=OWZVD-l-ZrC

[11] D. J. H. III and D. Sadigh, "Few-shot preference learning for human-in-the-loop RL," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=IKC5TfXLuW0

[12] J. Clark, J. Hejna, and D. Sadigh, "Efficiently generating expressive quadruped behaviors via language-guided preference learning," *preprint*, 2024.

[13] M. Palan, G. Shevchuk, N. C. Landolfi, and D. Sadigh, "Learning reward functions by integrating human demonstrations and preferences," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

[14] Y. Song, A. Romero, M. Mueller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Science Robotics*, p. adg1462, 2023. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.adg1462

[15] C.-H. Pi, W.-Y. Ye, and S. Cheng, "Robust quadrotor control through reinforcement learning with disturbance compensation," *Applied Sciences*, vol. 11, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/7/3257

[16] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *2010 IEEE International Conference on Robotics and Automation*. Anchorage, AK: IEEE, May 2010, pp. 1642–1648. [Online]. Available: http://ieeexplore.ieee.org/document/5509452/

[17] J. Xing, I. Geles, Y. Song, E. Aljalbout, and D. Scaramuzza, "Multi-task reinforcement learning for quadrotors," *IEEE Robotics and Automation Letters*, 2024.

[18] I. Geles, L. Bauersfeld, A. Romero, J. Xing, and D. Scaramuzza, "Demonstrating Agile Flight from Pixels without State Estimation," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[19] J. Xing, A. Romero, L. Bauersfeld, and D. Scaramuzza, "Bootstrapping reinforcement learning with imitation for vision-based agile flight," *Conference on Robot Learning*, 2024.

[20] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," in *Proceedings of the 2020 Conference on Robot Learning*, 2021, pp. 1147–1157.
[21] P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio, and D. Scaramuzza, "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight," *Science Robotics*, vol. 7, no. 67, p. eabl6259, 2022.
[22] L. Bauersfeld, E. Kaufmann, P. Foehn, S. Sun, and D. Scaramuzza, "Neurobem: Hybrid aerodynamic quadrotor model," *RSS: Robotics, Science, and Systems*, 2021.
[23] K. Lee, L. Smith, A. Dragan, and P. Abbeel, "B-pref: Benchmarking preference-based reinforcement learning," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung, Eds., vol. 1, 2021.
[24] H. Liu and P. Abbeel, "Behavior from the void: Unsupervised active pre-training," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 18 459–18 473. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/99bf3d153d4bf67d640051a1af322505-Paper.pdf
[25] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa, "dm control: Software and tasks for continuous control," *Software Impacts*, vol. 6, p. 100022, Nov. 2020.
[26] J. J. Tai, M. Towers, and E. Tower, "Shimmy: Gymnasium and PettingZoo Wrappers for Commonly Used Environments," June 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8140744

APPENDIX

A. Quadrotor Powerloop Shaped Reward

We design a shaped, Markovian reward for the PPO baseline and for synthetic preference generation. The reward consists of three components.

a) Planar Penalty.: Encourages movement in the xz-plane using the L1 distance to the plane. The drone's position is given by $\vec{p}_{drone}$, with p_{planar} denoting the distance from the drone to the circle plane defined by center $\vec{c}_{circle}$ and normal $\vec{n}_{plane}$:

$$p_{planar} = |(\vec{p}_{drone} - \vec{c}_{circle}) \cdot \vec{n}_{plane}|_1. \tag{18}$$

b) Circular Motion Reward.: Rewards tangential movement along a target circle. The vector $\vec{t}$ is the tangent of the circle and $\vec{v}_{circular,sat}$ is the drone's velocity projected onto the circle plane and saturated at a maximum value:

$$r_{circ} = \vec{t} \cdot \vec{v}_{circular,sat}. \tag{19}$$

c) Action Regularization.: Penalizes the high-frequency commands $c_{i,\text{high}} = c_i - \text{LP}(c_i)$, the raw minus the low-pass filtered command, for each action dimension i:

$$p_{reg,i} = |c_{i,\text{high}}| + |c_{i,\text{high}}|^2. \tag{20}$$

The total reward is a weighted sum of these components:

$$r_{total} = \alpha \cdot r_{circ} - \beta \cdot p_{planar} - \sum_i \gamma_i \cdot p_{reg,i}, \tag{21}$$

where $\alpha = 0.5$, $\beta = 3$, $\gamma_{\omega_{xy}} = \gamma_{\omega_z} = 0.008$, $\gamma_c = 0.0001$.

B. Hyperparameters

The hyperparameters used in our experiments are summarized in Tables B1 and B2, listing the preference-based and PPO settings, respectively.

TABLE B1

Preference-based hyperparameters in the *walker-walk* and *flightmare* environments. Pairing is 50% ensemble disagreement and 50% random.

Hyperparameter	flightmare	walker-walk
n_{prefs}	1000	1000
T_{clip}	1.25 s	1.25 s
f_{init}	0.2	0.1
n_{query}	10	10
$\Delta t_{\text{retrain}}$	400K	16K
n_{ensemble}	5	3
n_{reset}	1	1
σ_{target}	0.05	0.05
$d^{\mathcal{R}}_{\text{hidden}}$	256	256
$n^{\mathcal{R}}_{\text{layers}}$	2	2
$\phi^{\mathcal{R}}$	Tanh	Tanh
$n^{\mathcal{R}}_{\text{epochs}}$	100	100
$\eta^{\mathcal{R}}$	0.0003	0.0003

TABLE B2

PPO hyperparameters in the *walker-walk* and *flightmare* environments.

Hyperparameter	flightmare	walker-walk
n_{envs}	50	32
$n_{\text{timesteps}}$	30M	4M
γ	0.995	0.99
λ_{GAE}	0.95	0.92
n_{epochs}	10	20
n_{steps}	250	500
ϵ_{clip}	0.4	0.4
n_{batch}	12,500	64
$\log \sigma_{\text{init}}$	-0.8	0.0
π_{arch}	[128, 128]	[256, 256, 256]
$\pi_{\text{activation}}$	Tanh	Tanh
c_{entropy}	0.001	0.0

C. Pseudocode

The full training procedure is summarized below.

Algorithm 1: REC Pseudocode

Input: env. $\mathcal{E}$, policy π_θ, reward ensemble $\{\hat{r}_{\phi_i}\}_{i=1}^n$, preference budget B, clip length H, annotator $\mathcal{A}$, target std. σ_{target}
Initialize: trajectory memory $\mathcal{M} \leftarrow \emptyset$, preference dataset $\mathcal{D} \leftarrow \emptyset$

Collect diverse initial trajectories by unsupervised exploration; store clips of length H in $\mathcal{M}$.
while training budget not exhausted **do**
  Select informative pairs (τ_0, τ_1) from $\mathcal{M}$ using the reward ensemble.
  Query $\mathcal{A}$ for labels $p \in \{0, 0.5, 1\}$; add (τ_0, τ_1, p) to $\mathcal{D}$.
  Reinitialize worst-performing ensemble member on new labels.
  for each minibatch $(\tau_0, \tau_1, p) \sim \mathcal{D}$ **do**
    Predict per-timestep rewards $\hat{r}_{\phi_i}(o_t, a_t)$, $i = 1, \ldots, n$.
    Trajectory statistics:
      $\mu(\tau) = \sum_t \frac{1}{n} \sum_i \hat{r}_{\phi_i}, \quad \sigma(\tau)^2 = \sum_t \text{std}_i(\hat{r}_{\phi_i})^2.$
    REC preference:
      $\hat{p}(\tau_0 > \tau_1) = \Phi\left(\frac{\mu(\tau_0) - \mu(\tau_1)}{\sqrt{\sigma(\tau_0)^2 + \sigma(\tau_1)^2}}\right).$
    Update $\{\phi_i\}$ via cross-entropy$(\hat{p}, p) + (\bar{\sigma} - \sigma_{\text{target}})^2$.
  end for
  Collect rollouts with $\hat{r}_t^{\text{agg}} = \frac{1}{n} \sum_i \hat{r}_{\phi_i} + |X_t|, \; X_t \sim \mathcal{N}(0, \text{std}_i(\hat{r}_{\phi_i}))$.
  Update π_θ with PPO; append new clips to $\mathcal{M}$.
end while

Output: trained policy π_θ and reward ensemble $\{\hat{r}_{\phi_i}\}_{i=1}^n$