Supplementary Materials

In this supplementary, Section A shows the detailed definition of **Theorem 1** *(Contraction Metric)*, and proof of **Proposition 2** *(Policy Evaluation)*, **Proposition 3** *(Policy Improvement)*, **Theorem 4** *(Convergence)* and **Theorem 5** *(Training Acceleration)*. Section B shows the detailed nominal dynamic model of quadrotor, and description of parameters. Section C shows the algorithm details of QuaDUE, combining with DDPG algorithm. Section D shows the algorithm details of the reference trajectory generation, i.e., Kino-JSS. Section E gives the implementation details of QuaDUE-CCM.

A Detailed definition of Contraction; and proof of Proposition 2, Proposition 3, Theorem 4 and Theorem 5

Contraction is described as follows. We firstly consider general time-variant autonomous systems $\dot{x} = f_c(x,t)$. Then we have $\delta\dot{x} = \frac{\partial f_c}{\partial x}(x,t)\delta x$, where δx is a virtual displacement. Next, two neighboring trajectories are considered in the field $\dot{x} = f_c(x,t)$. The square distance between these two trajectories is defined as $\delta x^T\delta x$, where the rate of change is given by $\frac{\mathrm{d}}{\mathrm{d}t}(\delta x^T\delta x) = 2\delta x^T\delta\dot{x} = 2\delta x^T\frac{\partial f_c}{\partial x}\delta x$. Let $\lambda_m(x,t) < 0$ be the largest eigenvalue of the symmetrical part of the Jacobian $\frac{\partial f_c}{\partial x}$ such that there exists $\frac{\mathrm{d}}{\mathrm{d}t}(\delta x^T\delta x) \le 2\lambda_m\delta x^T\delta x$. Therefore, we have $\|\delta x\| \le e^{\int_0^t \lambda_m(x,t)\mathrm{d}t}\|\delta x_o\|$. Such a system can be called **contracting**. Therefore, we have the following **Theorem 1**.

Theorem 1 *(Contraction Metric)*: In a control-affine system, if there exists: $\dot{M} + \mathrm{sym}(M(A + BK)) + 2\lambda M \prec 0$, then the inequality $\|\boldsymbol{x}(t) - \boldsymbol{x}_{ref}(t)\| \le Re^{-\lambda t}\|\boldsymbol{x}(0) - \boldsymbol{x}_{ref}(0)\|$ with $\forall t \ge 0$, $R \ge 1$ and $\lambda > 0$ holds, where $A = A(\boldsymbol{x},\boldsymbol{u})$ and $B = B(\boldsymbol{x})$ are defined above, and $\dot{M} = \partial_{f(\boldsymbol{x})+g(\boldsymbol{x})u}M = \sum_{i=1}^n \frac{\partial M}{\partial x^i}\dot{x}^i$, $\mathrm{sym}(M) = M + M^T$, and $\boldsymbol{x}_{ref}(t)$ is a sequential reference trajectory. Therefore, the system is contracting.

Proof can be found in [1, 2].

Proposition 2 *(Policy Evaluation)*: Given a deterministic policy π, a quantile approximator Π_{W_1} and $Z_{k+1}(\boldsymbol{s},\boldsymbol{a}) = \Pi_{W_1}\mathcal{T}^\pi Z_k(\boldsymbol{s},\boldsymbol{a})$, the sequence $Z_k(\boldsymbol{s},\boldsymbol{a})$ converges to a unique fixed point $\widetilde{Z}_\pi$ under the maximal form of the ∞-Wasserstein metric $\bar{d}_\infty$.

Proof: The combined operator $\Pi_{W_1}\mathcal{T}^\pi$ is an ∞-contraction [3], as there exists:

$$\bar{d}_\infty(\Pi_{W_1}\mathcal{T}^\pi Z_1, \Pi_{W_1}\mathcal{T}^\pi Z_2) \le \bar{d}_\infty(Z_1, Z_2) \tag{1}$$

Based on the Banach's fixed point theorem, we have a unique fixed point $\widetilde{Z}_\pi$ of $\mathcal{T}^\pi$. Since all moments of Z are bounded in $Z_\theta(\boldsymbol{s},\boldsymbol{a}) := \frac{1}{N}\sum_{i=1}^{N}\delta_{q_i(\boldsymbol{s},\boldsymbol{a})}$, the sequence $Z_k(\boldsymbol{s},\boldsymbol{a})$ converges to $\widetilde{Z}_\pi$ in $\bar{d}_\infty$ for $p \in [1,\infty]$. ■

Proposition 3 *(Policy Improvement)*: Denoting an old policy by $\boldsymbol{\pi_{old}}$ and a new policy by $\boldsymbol{\pi_{new}}$, there exists $\mathbb{E}[Z(s,a)]^{\boldsymbol{\pi_{new}}}(s,a) \ge \mathbb{E}[Z(s,a)]^{\boldsymbol{\pi_{old}}}(s,a)$, $\forall s \in \mathcal{S}$ and $\forall a \in \mathcal{A}$.

Proof: We firstly denote the expectation of $Z(s,a)$ by $Q(s,a)$. Then based on:

$$\begin{gathered}\mathcal{T}Z(\boldsymbol{s},\boldsymbol{a}) \stackrel{D}{:=} R(\boldsymbol{s},\boldsymbol{a}) + \gamma Z(\boldsymbol{s'}, \mathrm{arg}\underset{a'}{max}\underset{\boldsymbol{p},R}{\mathbb{E}}[Z(\boldsymbol{s'},\boldsymbol{a'})]) \\ Z_\theta(\boldsymbol{s},\boldsymbol{a}) := \frac{1}{N}\sum_{i=1}^{N}\delta_{q_i(\boldsymbol{s},\boldsymbol{a})}\end{gathered} \tag{2}$$

there exists:

$$V^\pi(s_t) = \mathbb{E}_\pi Q^\pi(s_t,\pi(s_t)) \le \underset{a\in\mathcal{A}}{\max}\mathbb{E}_\pi Q^\pi(s_t,a) = \mathbb{E}_{\pi'}Q^\pi(s_t,\pi'(s_t)) \tag{3}$$

where $\mathbb{E}_{\pi}[\cdot] = \sum_{a\in A} \boldsymbol{\pi}(a|s)[\cdot]$, and $V^{\pi}(s) = \mathbb{E}_{\pi}\mathbb{E}[Z_k(s,a)]$ is the value function. According to Equation 2 and Equation 3, it yields:

$$
\begin{aligned}
Q^{\boldsymbol{\pi_{old}}} &= Q^{\boldsymbol{\pi_{old}}}(s_t, \boldsymbol{\pi_{new}}(s_t)) = r_{t+1} + \gamma \mathbb{E}_{s_{t+1}} \mathbb{E}_{\boldsymbol{\pi_{old}}} Q^{\boldsymbol{\pi_{old}}}(s_{t+1}, \boldsymbol{\pi_{old}}(s_{t+1})) \\
&\leq r_{t+1} + \gamma \mathbb{E}_{s_{t+1}} \mathbb{E}_{\boldsymbol{\pi_{new}}} Q^{\boldsymbol{\pi_{old}}}(s_{t+1}, \boldsymbol{\pi_{new}}(s_{t+1})) \\
&\leq r_{t+1} + \mathbb{E}_{s_{t+1}} \mathbb{E}_{\boldsymbol{\pi_{new}}} [\gamma r_{t+2} + \gamma^2 \mathbb{E}_{s_{t+2}} Q^{\boldsymbol{\pi_{old}}}(s_{t+2}, \boldsymbol{\pi_{new}}(s_{t+2}))]] \\
&\leq r_{t+1} + \mathbb{E}_{s_{t+1}} \mathbb{E}_{\boldsymbol{\pi_{new}}} [\gamma r_{t+2} + \gamma^2 r_{t+3} + ...] = r_{t+1} + \mathbb{E}_{s_{t+1}} V^{\boldsymbol{\pi_{new}}}(s_{t+1}) \\
&= Q^{\boldsymbol{\pi_{new}}}
\end{aligned} \tag{4}
$$

Thus, there exists $\mathbb{E}[Z(s,a)]^{\boldsymbol{\pi_{new}}}(s,a) \geq \mathbb{E}[Z(s,a)]^{\boldsymbol{\pi_{old}}}(s,a)$. ■

Theorem 4 *(Convergence)*: Denoting the policy of the i-th policy improvement by $\boldsymbol{\pi^i}$, there exists $\boldsymbol{\pi^i} \to \boldsymbol{\pi}^*$, $i \to \infty$, and $\mathbb{E}[Z_k(s,a)]^{\boldsymbol{\pi}^*}(s,a) \geq \mathbb{E}[Z_k(s,a)]^{\boldsymbol{\pi^i}}(s,a)$, $\forall s \in \mathcal{S}$ and $\forall a \in \mathcal{A}$.

Proof: **Proposition 3** shows that $\mathbb{E}[Z(s,a)]^{\boldsymbol{\pi^i}} \geq \mathbb{E}[Z(s,a)]^{\boldsymbol{\pi^{i-1}}}$, thus $\mathbb{E}[Z(s,a)]^{\boldsymbol{\pi^i}}$ is monotonically increasing. The immediate reward is defined as:

$$
R_{t+1}(s,a,\theta) = R_{contraction}(s,a,\theta) + R_{track}(s,a) \tag{5}
$$

Extending to Equation 5, the reward function is defined as:

$$
\begin{aligned}
R_{contraction}(s,a,\theta) = &- \omega_{c,1}[\underline{m}I - M]_{ND}(s) - \omega_{c,2}[M - \overline{m}I]_{ND}(s) \\
&- \omega_{c,3}[\widehat{C}_m + 2\lambda M]_{ND}(s,a,\theta) \\
R_{track}(s,a) = &- (\boldsymbol{x}_t(s,a) - \boldsymbol{x}_{ref,t})^{\mathrm{T}} H_1 (\boldsymbol{x}_t(s,a) - \boldsymbol{x}_{ref,t}) - \boldsymbol{u}_t^{\mathrm{T}}(s,a) H_2 \boldsymbol{u}_t(s,a)
\end{aligned} \tag{6}
$$

where $\underline{m}$, $\overline{m}$ are hyper-parameters, H_1 and H_2 are positive definite matrices, and $[A]_{ND}$ is for penalizing positive definiteness where $[A]_{ND} = 0$ iff. $A \prec 0$, and $[A]_{ND} \geq 0$ iff. $A \succeq 0$.

According to Equation 1, Equation 5 and Equation 6, the first moment of Z, i.e., $\mathbb{E}[Z(s,a)]^{\boldsymbol{\pi^i}}$, is upper bounded. Therefore, the sequential $\mathbb{E}[Z(s,a)]^{\boldsymbol{\pi^i}}$ converges to an upper limit $\mathbb{E}[Z(s,a)]^{\boldsymbol{\pi}^*}$ satisfying $\mathbb{E}[Z_k(s,a)]^{\boldsymbol{\pi}^*}(s,a) \geq \mathbb{E}[Z_k]^{\boldsymbol{\pi^i}}$. ■

Theorem 5 *(Training Acceleration)*: In the training process of the distribution RL (i.e. QuaDUE):

1) Let sampling steps $T = 4kl^2 E_J(\theta_0)/\tau^2$, if there exists $\kappa \leq 0.25\tau^2/\sigma^2$, the J_θ optimized by stochastic gradient descend converges to a τ-stationary point.

2) Let sampling steps $T = kl^2 E_J(\theta_0)/(\kappa\tau^2)$, if there exists $\kappa > 0.25\tau^2/\sigma^2$, the J_θ optimized by stochastic gradient descent will not converge to a τ-stationary point.

Proof: As $J_\theta(p^{s,a}, a_\theta^{s,a})$ is kl^2-smooth, we have:

$$
\begin{aligned}
E_J(\theta_{t+1}) - E_J(\theta_t) &\leq \langle \nabla E_J(\theta_t), \theta_{t+1} - \theta_t \rangle + (kl^2/2) \left\|\theta_{t+1} - \theta_t\right\|^2 \\
&= -\iota \langle \nabla E_J(\theta_t), \nabla J_\theta(p^{s,a}, a_\theta^{s,a}) \rangle + (kl^2 \iota^2/2) \left\|\nabla J_\theta(p^{s,a}, a_\theta^{s,a})\right\|^2 \\
&\leq -(\iota/2) \left\|\nabla E_J(\theta_t)\right\|^2 + (\iota/2) \left\|\nabla E_J(\theta_t) - J_\theta(p^{s,a}, a_\theta^{s,a})\right\|^2
\end{aligned} \tag{7}
$$

Next we take the expected value of Equation 7 and consider T steps:

$$
\begin{aligned}
\mathbb{E}[E_J(\theta_T) - E_J(\theta_0)] &\leq \mathbb{E}[\sum_{t=0}^{T-1} -(\iota/2) \left\|\nabla E_J(\theta_t)\right\|^2] + \mathbb{E}[\sum_{t=0}^{T-1} (\iota/2) \left\|\nabla E_J(\theta_t) - J_\theta(p^{s,a}, a_\theta^{s,a})\right\|^2] \\
&\leq -(\iota/2) \sum_{t=0}^{T-1} \mathbb{E}[\left\|\nabla E_J(\theta_t)\right\|^2] + (\iota/2)[(1 - 1/(1+\kappa))\sigma^2 + (\kappa/(1+\kappa))\sigma^2]
\end{aligned} \tag{8}
$$

According to Equation 8 above, if we have a first-order stationary point at T step, then:

$$
(1/T) \sum_{t=0}^{T-1} \mathbb{E}[\left\|\nabla E_J(\theta_t)\right\|^2] \leq 2kl^2 E_J(\theta_0)/T + 2\kappa\sigma^2 \tag{9}
$$

If we have $T = 4kl^2 E_J(\theta_0)/\tau^2$ and $\kappa \leq 0.25\tau^2/\sigma^2$, then J_θ converges to a τ-stationary point with $(1/T)\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla E_J(\theta_t)\|^2] \leq \tau^2$. Thus, (1) is proved. If we have $T = kl^2 E_J(\theta_0)/(\kappa\tau^2)$ and $\kappa > 0.25\tau^2/\sigma^2$, then $(1/T)\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla E_J(\theta_t)\|^2] \leq 4\kappa\tau^2$, where the stationary point is dependent on κ, i.e., the degree of the approximation. Therefore, (2) is proved. ■

B Nominal Dynamic Model of Quadrotor

The quadrotor is assumed as a six Degrees of Freedom (DoF) rigid body of mass m, i.e., three linear motions and three angular motions [4]. Different from [5, 6], the aerodynamic effect (disturbance) $\boldsymbol{e}_f$ is integrated into the quadrotor dynamic model as follows [7]:

$$
\begin{aligned}
\dot{\boldsymbol{P}}_{WB} &= \boldsymbol{V}_{WB} \\
\dot{\boldsymbol{V}}_{WB} &= \boldsymbol{g}_W + \frac{1}{m}(\boldsymbol{q}_{WB} \odot \boldsymbol{c} + \boldsymbol{e}_f) \\
\dot{\boldsymbol{q}}_{WB} &= \frac{1}{2}\Lambda(\boldsymbol{\omega}_B)\boldsymbol{q}_{WB} \\
\dot{\boldsymbol{\omega}}_B &= \boldsymbol{J}^{-1}(\boldsymbol{\tau}_B - \boldsymbol{\omega}_B \times J\boldsymbol{\omega}_B)
\end{aligned}
\tag{10}
$$

where $\boldsymbol{P}_{WB}$, $\boldsymbol{V}_{WB}$ and $\boldsymbol{q}_{WB}$ are the position, linear velocity and orientation expressed in the world frame, and $\boldsymbol{\omega}_B$ is the angular velocity expressed in the body frame [7]; $\boldsymbol{c}$ is the collective thrust $\boldsymbol{c} = [0, 0, \sum T_i]^{\mathrm{T}}$; the operator $\odot$ denotes a rotation of the vector by the quaternion; $\boldsymbol{\tau}_B$ is the body torque; $\boldsymbol{J} = diag(j_x, j_y, j_z)$ is the diagonal moment of inertia matrix; $\boldsymbol{g}_W = [0, 0, -g]^{\mathrm{T}}$; and, the skewsymmetric matrix $\Lambda(\boldsymbol{\omega})$ is defined as:

$$
\Lambda(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}_x & -\boldsymbol{\omega}_y & -\boldsymbol{\omega}_z \\ \boldsymbol{\omega}_x & 0 & -\boldsymbol{\omega}_z & -\boldsymbol{\omega}_y \\ \boldsymbol{\omega}_y & -\boldsymbol{\omega}_z & 0 & \boldsymbol{\omega}_x \\ \boldsymbol{\omega}_z & \boldsymbol{\omega}_y & -\boldsymbol{\omega}_x & 0 \end{bmatrix}
\tag{11}
$$

Then we reformulate Equation 10 in its control-affine form:

$$
\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{e}_{\boldsymbol{f}_k}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{w}
\tag{12}
$$

where $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $\boldsymbol{g} : \mathbb{R}^n \mapsto \mathbb{R}^{n \times m}$ are assumed by the standard as Lipschitz continuous [8, 9]. $\boldsymbol{x} = [\boldsymbol{P}_{WB}, \boldsymbol{V}_{WB}, \boldsymbol{q}_{WB}, \boldsymbol{\omega}_B]^T \in \mathbb{X}$, $\boldsymbol{u} = T_i \forall i \in (0, 3) \in \mathbb{U}$ and $\boldsymbol{w} \in \mathbb{W}$ are the state, input and additive uncertainty of the dynamic model, where $\mathbb{X} \subseteq \mathbb{R}^n$, $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ and $\mathbb{W} \subseteq \mathbb{R}^{n_w}$ are compact sets as the state, input and uncertainty space, respectively. Given an input : $\mathbb{R}^{\geq 0} \mapsto \mathbb{U}$ and an initial state $x_0 \in \mathbb{X}$, our goal is to design a quadrotor tracking controller $\boldsymbol{u}$ such that the state trajectory $\boldsymbol{x}$ can track *any* reference state trajectory $\boldsymbol{x}_{ref}$ (satisfied the quadrotor dynamic limits) under a bounded uncertainty $\boldsymbol{w} : \mathbb{R}^{\geq 0} \mapsto \mathbb{W}$.

C The detailed algorithm of QuaDUE

The objective of this work is to design a distributional-RL-based estimator for CCM uncertainty, which we define as combined wind estimation and CCM uncertainty estimation, for tracking the reference state $\boldsymbol{x}_{ref}$ of the nominal model (Equation 10). The detailed algorithm of QuaDUE is shown in Algorithm 1.

D The detailed algorithm of Kino-JSS

Quadrotor route searching primarily focuses on robustness, feasibility and efficiency. The Kino-RS algorithm [10] is a robust and feasible online searching approach. However, the searching loop is derived from the hybrid-state A* algorithm, making it relatively inefficient in obstacle-dense environments. On the other hand, JPS offers robust route searching, and runs at an order of magnitude

Algorithm 1 QuaDUE

Input: $\boldsymbol{s}_k$, $\boldsymbol{s}_{k+1}$, $\boldsymbol{u}_k$, θ^{μ}, θ^{Q}
Output: $\boldsymbol{a}_k$

Initialize:
- $\theta^{\mu^t} \leftarrow \theta^{\mu}$, $\theta^{Q^t} \leftarrow \theta^{Q}$ update the target parameters from the predicted parameters
- the replay memory $D \leftarrow D_{k-1}$
- the batch B, and its size
- a small threshold $\xi \in \mathbb{R}_+$
- the random option selection probability ϵ - the option termination probability β
- quantile estimation functions $\{q_i\}_{i=1,...,N}$

Repeat
for each sampling step from D **do**
  Select a candidate option z_k from $\left\{z^0, z^1, ..., z^M\right\}$
  $z_k \leftarrow \begin{cases} z_{k-1} & w.p.\ 1-\beta \\ \text{random option} & w.p.\ \beta\epsilon \\ \text{argmax}_z Q(\boldsymbol{s}_k, z) & w.p.\ \beta(1-\epsilon) \end{cases}$
  Execute w_k, get reward r_k and the next state $\boldsymbol{s}_{k+1}$
  $D.\mathbf{Insert}([\mathbf{s_k}, \mathbf{u_k}, \mathbf{r_k}, \mathbf{s_{k+1}}])$
  $B \leftarrow D.\mathbf{sampling}$
  $y_{k,i} \leftarrow \rho_{\tau_i}^{\mathcal{K}}(r_k + \gamma q_i'(\boldsymbol{s}_{k+1}, w_k^*)$
  $J_{\theta^{\mu}} \leftarrow \frac{1}{N} \sum\limits_{i=1}^{N} \sum\limits_{i'=1}^{N} [y_{k,i'} - q_i(\boldsymbol{s}_k, w_k)]$
  $y \leftarrow \beta \text{argmax}_{z'} Q(\boldsymbol{s}_{k+1}, z') + (1-\beta) Q(\boldsymbol{s}_{k+1}.z_k)$
  $J_{\theta^Q} \leftarrow (r_t + \gamma y - Q(\boldsymbol{s}_t, z_t))^2$
  $\theta^{\mu} \leftarrow \theta^{\mu} - l_{\mu} \nabla_{\theta^{\mu}} J_{\theta^{\mu}}$
  $\theta^{Q} \leftarrow \theta^{Q} - l_{\theta} \nabla_{\theta^{Q}} J_{\theta^{Q}}$
end for
Until convergence, , $J_Q^{\theta} < \xi$

faster than the A* algorithm [11]. A common problem of geometric methods such as JPS and A* is that, unlike kinodynamic searching, they consider heuristic cost (e.g., distance) but not the quadrotor dynamics and feasibility (e.g., line 5 of Algorithm 3 and line 10 of Algorithm 4) when generating routes [12]. $\mathbf{checkFea}()$ is the feasibility check to judge the acceleration and velocity constrains based on the quadrotor dynamics. Kino-JSS, proposed in [7], generates a safe and efficient route in unknown environments with aerodynamic disturbances. In [7], Kino-JSS, described by Algorithms 2, 3 and 4, is demonstrated to run an order of magnitude faster than Kino-RS [10] in obstacle-dense environments, whilst maintaining comparable system performance.

Algorithm 2 Kinodynamic Jump Space Search [7]

INPUT: s_{cur}
OUTPUT: $KinoJSSRoute$

$\mathbf{initialize}()$
$openSet.\mathbf{insert}(s_{cur})$
while !$openSet.\mathbf{isEmpty}()$ **do**
  $s_{cur} \leftarrow openSet.\mathbf{pop}()$
  $closeSet.\mathbf{insert}(s_{cur})$
  if $\mathbf{nearGoal}(s_{cur})$ **then**
    return $KinoJSSRoute$
  end if
  $\mathbf{KinoJSSRecursion}()$
end while

s_{cur} denotes the current state, s_{pro} denotes the propagation of current state under the motion m_i, and E_f denotes the aerodynamic disturbance estimated by VID-fusion [13]. In Algorithm 3, $motionSet$,

Algorithm 3 KinoJSSRecursion [7]

INPUT: s_{cur}, E_f, $openSet$, $closeSet$
OUTPUT: $void$

$motions \leftarrow \textbf{JSSMotion}(s_{cur}, E_f)$
for each $m_i \in motions$ **do**
  $s_{pro} \leftarrow \textbf{statePropagation}(s_{cur}, m_i))$
  $inClose \leftarrow closeSet.\textbf{isContain}(s_{pro})$
  if isFree$(s_{pro}) \bigwedge$ **checkFea**$(s_{pro}, m_i) \bigwedge inClose$ **then**
    if checkOccupiedAround(s_{pro}) **then**
      $s_{pro}.neighbors \leftarrow \textbf{JSSNeighbor}(s_{pro})$
      $cost_{pro} \leftarrow s_{cur}.cost + \textbf{edgeCost}(s_{pro})$
      $cost_{pro} \leftarrow cost_{pro} + \textbf{heuristic}(s_{pro})$
      if !$openSet.\textbf{isContain}(s_{pro})$ **then**
        $openSet.\textbf{insert}(s_{pro})$
      else if $s_{pro}.cost \leq cost_{pro}$ **then**
        continue
      end if
      $s_{pro}.parent \leftarrow s_{cur}$
      $s_{pro}.cost \leftarrow cost_{pro}$
    else
      KinoJSSRecursion$()$
    end if
  else
    continue
  end if
end for

Algorithm 4 JSSMotion [7]

INPUT: s_{cur}, E_f
OUTPUT: $motions$

$E_{fcor} \leftarrow E_f + \textbf{GaussianNoise}()$
for each $m_i \in motionSet$ **do**
  $m_{cor} \leftarrow m_i + E_{fcor}$
  $motions \leftarrow \textbf{push\_back}(m_{cor})$
end for
$neighSize \leftarrow s_{cur}.neighbors.\textbf{size}()$
while $neighSize \neq 0$ **do**
  $neighSize = neighSize - 1$
  $neighMotion \leftarrow \textbf{posToMotion}(s_{cur}.neighbors)$
  if checkFea$(s_{cur}, neighMotion)$ **then**
    $motions \leftarrow \textbf{push\_back}(neighMotion)$
  end if
end while
return $motions$

Table 1: Parameters of QuaDUE-CCM

Parameters	Definition	Values
l_{θ_a}	Learning rate of actor	0.0015
l_{θ_c}	Learning rate of critic	0.0015
θ_a	Actor neural network: fully connected with two hidden layers (128 neurons per hidden layer)	-
θ_c	Critic neural network: fully connected with two hidden layers (128 neurons per hidden layer)	-
D	Replay memory capacity	10^4
B	Batch size	256
γ	Discount rate	0.9995
-	Training episodes	1000
T_s	MPC Sampling period	50ms
N	Time steps	20

which is defined as a pyramid shown, offers improved efficiency whilst retaining the advantages of Kino-RS [10].

E The implementation details of QuaDUE-CCM

The performance of our proposed QuaDUE-CCM is evaluated using a DJI Manifold 2-C (Intel i7-8550U CPU) for real-time computation. We use RotorS MAVs simulator [14], where programmable aerodynamic disturbances can be generated. The nominal force $\boldsymbol{n_f}$ is estimated by VID-Fusion [13]. The noise bound of aerodynamic forces is set as 0.5 m/s^2, based on the benchmark established in [15]. Since the update frequency of the aerodynamic force $\boldsymbol{e_f}$ estimation is much higher than our QuaDUE-CCM framework frequency, we sample $\boldsymbol{e_f}$ based on our framework frequency. We also assume the collective thrust $\boldsymbol{c}$ is a true value, which is tracked ideally in the simulation platform.

The parameters of our proposed framework are summarized in Table 1. Then we operate a training process by generating external forces in RotorS [14], where the programmable external forces are in the horizontal plane with range [-3,3] (m/s^2). The training process has 1000 iterations where the quadrotor state is recorded at 16 Hz. The training process occurs over 1000 iterations. The matrices H_1 and H_2 in Equation 6 are chosen as $H_1 = diag\{2.5e^{-2}, 2.5e^{-2}, 2.5e^{-2}, 1e^{-3}, 1e^{-3}, 1e^{-3}, 2.5e^{-3}, 2.5e^{-3}, 2.5e^{-3}, 2.5e^{-3}, 1e^{-5}, 1e^{-5}, 1e^{-5}\}$ and $H_2 = diag\{1.25e^{-4}, 1.25e^{-4}, 1.25e^{-4}, 1.25e^{-4}\}$, respectively.

References

[1] W. Lohmiller and J.-J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[2] D. Sun, S. Jha, and C. Fan. Learning certified control using contraction metric. *arXiv preprint arXiv:2011.12569*, 2020.

[3] W. Dabney, M. Rowland, M. Bellemare, and R. Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[4] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza. Data-driven mpc for quadrotors. *IEEE Robotics and Automation Letters*, 6(2):3769–3776, 2021.

[5] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart. Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system. In *Robot operating system (ROS)*, pages 3–39. Springer, 2017.

[6] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza. Pampc: Perception-aware model predictive control for quadrotors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[7] Y. Wang, J. O'Keeffe, Q. Qian, and D. Boyle. Kinojgm: A framework for efficient and accurate quadrotor trajectory generation and tracking in dynamic environments. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11036–11043. IEEE, 2022.

[8] J. Choi, F. Castaneda, C. J. Tomlin, and K. Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. *arXiv preprint arXiv:2004.07584*, 2020.

[9] C. Dawson, S. Gao, and C. Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods. *arXiv preprint arXiv:2202.11762*, 2022.

[10] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen. Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robotics and Automation Letters*, 4(4):3529–3536, 2019.

[11] D. Harabor and A. Grastien. Online graph pruning for pathfinding on grid maps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, 2011.

[12] W. Ding, W. Gao, K. Wang, and S. Shen. An efficient b-spline-based kinodynamic replanning framework for quadrotors. *IEEE Transactions on Robotics*, 35(6):1287–1306, 2019.

[13] Z. Ding, T. Yang, K. Zhang, C. Xu, and F. Gao. Vid-fusion: Robust visual-inertial-dynamics odometry for accurate external force estimation. *arXiv preprint arXiv:2011.03993*, 2020.

[14] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart. Rotors—a modular gazebo mav simulator framework. In *Robot operating system (ROS)*, pages 595–625. Springer, 2016.

[15] Y. Wu, Z. Ding, C. Xu, and F. Gao. External forces resilient safe motion planning for quadrotor. *arXiv preprint arXiv:2103.11178*, 2021.