A Appendix

A.1 Proof of Lemma 1

Proof. Using the e-ISS property in Assumption 1 we have:

\[
\frac{1}{TN} \sum_{i=1}^{N} \sum_{t=1}^{T} \| x_t^{(i)} \| \leq \frac{1}{TN} \sum_{i=1}^{N} \sum_{t=1}^{T} \left( \gamma \sum_{k=1}^{t-1} \rho^{t-1-k} \| B_k^{(i)} u_k^{(i)} - f_k^{(i)} + w_k^{(i)} \| \right)
\]

\[
\leq (a) \frac{\gamma}{1-\rho} \frac{1}{TN} \sum_{i=1}^{N} \sum_{t=1}^{T} \| B_t^{(i)} u_t^{(i)} - f_t^{(i)} + w_t^{(i)} \|
\]

\[
\leq (b) \frac{\gamma}{1-\rho} \sqrt{\frac{1}{TN} \sum_{i=1}^{N} \sum_{t=1}^{T} \| B_t^{(i)} u_t^{(i)} - f_t^{(i)} + w_t^{(i)} \|^2},
\]

where (a) and (b) are from geometric series and Cauchy-Schwarz inequality respectively. □

A.2 Proof of Lemma 2

This proof is based on the proof of Theorem 4.1 in [28].

Proof. For any \( \Theta \in \mathcal{K}_1 \) and \( \bar{c}^{(1:N)} \in \mathcal{K}_2 \) we have

\[
\sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t^{(i)} (\hat{\Theta}_t, \bar{c}_t^{(i)}) - \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t^{(i)} (\bar{\Theta}, \bar{c}^{(i)})
\]

\[
\leq (a) \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla \ell_t^{(i)} (\hat{\Theta}_t, \bar{c}_t^{(i)}) \cdot (\hat{\Theta}_t - \bar{\Theta}) + \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla \ell_t^{(i)} (\hat{\Theta}_t, \bar{c}_t^{(i)}) \cdot (\bar{c}_t^{(i)} - \bar{c}^{(i)})
\]

\[
= \sum_{i=1}^{N} \left[ G_t^{(i)} (\hat{\Theta}_t) - G_t^{(i)} (\bar{\Theta}) \right] + \sum_{i=1}^{N} \sum_{t=1}^{T} \left[ g_t^{(i)} (\bar{c}_t^{(i)}) - g_t^{(i)} (\bar{c}^{(i)}) \right]
\]

\[
\leq \sum_{i=1}^{N} \min_{\Theta \in \mathcal{K}_1} \sum_{i=1}^{N} G_t^{(i)} (\Theta) - \sum_{i=1}^{N} \sum_{t=1}^{T} g_t^{(i)} (\bar{c}_t^{(i)}) - \sum_{i=1}^{N} \min_{\bar{c}^{(i)} \in \mathcal{K}_2} \sum_{i=1}^{N} g_t^{(i)} (\bar{c}^{(i)}).
\]

where we have (a) because \( \ell_t^{(i)} \) is convex. Note that the total regret of \( \mathcal{A}_1 \) is \( T \cdot o(N) \) because \( G^{(i)} \) is scaled up by a factor of \( T \). □

A.3 Proof of Theorem 3

Proof. Since \( \Theta \in \mathcal{K}_1 \) and \( \bar{c}^{(1:N)} \in \mathcal{K}_2 \), applying Lemma 2 we have

\[
\sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t^{(i)} (\hat{\Theta}_t, \bar{c}_t^{(i)}) = \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_t^{(i)} (\Theta, \bar{c}^{(i)}) \leq T \cdot o(N) + N \cdot o(T)
\]

Recall that the definition of \( \ell_t^{(i)} \) is \( \ell_t^{(i)} (\hat{\Theta}, \bar{c}) = \| F(\phi(x_t^{(i)} ; \hat{\Theta}), \hat{c}) - y_t^{(i)} \|^2 \), and \( y_t^{(i)} = f_t^{(i)} - w_t^{(i)} \).

Therefore we have

\[
\ell_t^{(i)} (\hat{\Theta}_t, \bar{c}_t^{(i)}) = \| f_t^{(i)} - f_t^{(i)} + w_t^{(i)} \|^2 = \| B_t^{(i)} u_t^{(i)} - f_t^{(i)} + w_t^{(i)} \|^2
\]

\[
\ell_t^{(i)} (\Theta, \bar{c}^{(i)}) = \| w_t^{(i)} \|^2 \leq W^2.
\]
The OGD algorithm initializes OGD with learning rates (where \( \eta \)). Then applying Lemma 1 we have upper bounded by \( \eta \). From Lemma 7, using learning rates upper bounded as \( \gamma \). Recall that \( g \), \( R \), and \( \text{ACE} \). Define \( x \) as the total regret of the outer-adapter \( A_1 \), and \( R(A_2) \) as the total regret of the inner-adapter \( A_2 \). Recall that in Theorem 3 we show that \( \text{ACE}(\text{OMAC}) \leq \frac{1}{\sqrt{T}} \). Now we will prove Corollary 4 by analyzing \( R(A_1) \) and \( R(A_2) \) respectively.

**Proof of Corollary 4** Since the true dynamics \( f(x, c) = Y_1(x)\hat{\Theta} + Y_2(x)c \), we have upper bounded as \( \| \nabla_c g_t(i) \| = \| 2Y_2(x_t(i))^T \left( Y_1(x_t(i))\hat{\Theta} + Y_2(x_t(i))c_t(i) - Y_1(x_t(i))\hat{\Theta} - Y_2(x_t(i))c_t(i) + w_t(i) \right) \| \). From Lemma 7 using learning rates \( \eta_i(i) = \frac{2K_i}{c_2\sqrt{T}} \) for all \( i \), the regret of \( A_2 \) at each outer iteration is upper bounded by \( 3K_i^2C_2N \sqrt{T} \). Then the total regret of \( A_2 \) is bounded as \( R(A_2) \leq 3K_iC_2N \sqrt{T} \).
Now let us study $A_1$. Similarly, recall that $G^{(i)}(\hat{\Theta}) = \sum_{t=1}^{T} \nabla_{\Theta} \ell^{(i)}_t(\hat{\Theta}^{(i)}, \hat{c}^{(i)}) \cdot \hat{\Theta}$, which is convex (linear) w.r.t. $\hat{\Theta}$. The gradient of $G^{(i)}$ is upper bounded as

$$\|\nabla_{\hat{\Theta}} G^{(i)}\| = \left\| \sum_{t=1}^{T} 2Y_1(x_t^{(i)})^\top \left( Y_1(x_t^{(i)})\hat{\Theta}^{(i)} + Y_2(x_t^{(i)})\hat{c}^{(i)} - Y_1(x_t^{(i)})\Theta - Y_2(x_t^{(i)})c^{(i)} + w_t^{(i)} \right) \right\|$$

$$\leq T \left( 2K_1^2K_{\Theta} + 2K_1K_c + 2K_2^2K_{\Theta} + 2K_1K_2K_c + 2K_1W \right)$$

Finally using Theorem 3 we have

$$\text{ACE(OMAC)} \leq \frac{\gamma}{1 - \rho} \sqrt{W^2 + \frac{\mathcal{R}(A_1) + \mathcal{R}(A_2)}{TN}}$$

From Lemma 7 using learning rates $\bar{\eta}^{(i)} = \frac{2K_{\Theta}}{T C_1 \sqrt{N}}$, the total regret of $A_1$ is upper bounded as

$$\mathcal{R}(A_1) \leq 3K_{\Theta} T C_1 \sqrt{N}.$$ 

Finally using Theorem 3 we have

$$\text{ACE(baseline adaptive control)} \leq \frac{\gamma}{1 - \rho} \sqrt{W^2 + \frac{\mathcal{R}(A_1) + \mathcal{R}(A_2)}{TN}}$$

Now let us analyze $\text{ACE(baseline adaptive control)}$. To simplify notations, we define $\tilde{Y}(x) = [Y_1(x) \ \ Y_2(x)]: \mathbb{R}^n \to \mathbb{R}^{n \times (p + h)}$ and $\tilde{\alpha} = [\tilde{\Theta}; \tilde{c}] \in \mathbb{R}^{p + h}$. The baseline adaptive controller updates the whole vector $\tilde{\alpha}$ at every time step. We denote the ground truth parameter by $\alpha^{(i)} = [\Theta; c^{(i)}]$, and the estimation by $\hat{\alpha}^{(i)} = [\hat{\Theta}; \hat{c}^{(i)}]$. We have $\|\alpha^{(i)}\| \leq \sqrt{K_{\Theta}^2 + K_c^2}$. Define $\mathcal{K} = \{\hat{\alpha} = [\hat{\Theta}; \hat{c}] : \|\hat{\Theta}\| \leq K_{\Theta}, \|\hat{c}\| \leq K_c\}$, which is a convex set in $\mathbb{R}^{p + h}$.

Note that the loss function for the baseline adaptive control is $\bar{\ell}^{(i)}(\hat{\alpha}) = \|\tilde{Y}(x_t^{(i)})\hat{\alpha} - Y_1(x_t^{(i)})\Theta - Y_2(x_t^{(i)})c^{(i)} + w_t^{(i)}\|^2$. The gradient of $\bar{\ell}^{(i)}$ is

$$\nabla_{\hat{\Theta}} \bar{\ell}^{(i)}(\hat{\alpha}) = 2 \begin{bmatrix} Y_1(x_t^{(i)})^\top \\ Y_2(x_t^{(i)})^\top \end{bmatrix} (Y_1(x_t^{(i)})\hat{\Theta} + Y_2(x_t^{(i)})\hat{c} - Y_1(x_t^{(i)})\Theta - Y_2(x_t^{(i)})c^{(i)} + w_t^{(i)}),$$

whose norm on $\mathcal{K}$ is bounded by

$$\sqrt{4(K_1^2 + K_2^2)(2K_1K_\Theta + 2K_2K_c + W)^2} = \sqrt{C_1^2 + C_2^2}.$$ 

Therefore, from Lemma 7 running OGD on $\mathcal{K}$ with learning rates $\frac{2\sqrt{K_{\Theta}^2 + K_c^2}}{\sqrt{C_1^2 + C_2^2} \sqrt{T}}$ gives the following guarantee at each outer iteration:

$$\sum_{t=1}^{T} \bar{\ell}^{(i)}(\hat{\alpha}^{(i)}) - \bar{\ell}^{(i)}(\alpha^{(i)}) \leq 3\sqrt{K_{\Theta}^2 + K_c^2} \sqrt{C_1^2 + C_2^2} \sqrt{T}.$$ 

Finally, similar as (12) we have

$$\text{ACE(baseline adaptive control)} \leq \frac{\gamma}{1 - \rho} \sqrt{\sum_{i=1}^{N} \sum_{t=1}^{T} \bar{\ell}^{(i)}(\hat{\alpha}^{(i)})}$$

$$\leq \frac{\gamma}{1 - \rho} \sqrt{\sum_{i=1}^{N} 3\sqrt{K_{\Theta}^2 + K_c^2} \sqrt{C_1^2 + C_2^2} \sqrt{T} + \sum_{i=1}^{N} \sum_{t=1}^{T} \bar{\ell}^{(i)}(\alpha^{(i)})}$$

$$\leq \frac{\gamma}{1 - \rho} \sqrt{W^2 + 3\sqrt{K_{\Theta}^2 + K_c^2} \sqrt{C_1^2 + C_2^2} \frac{1}{\sqrt{T}}}.$$ 

Note that this bound does not improve as the number of environments (i.e., $N$) increases.
A.5 Proof of Theorem 6

Proof. For any \( \Theta \in \mathcal{K}_1 \) and \( c^{(1:N)} \in \mathcal{K}_2 \) we have

\[
\sum_{i=1}^N \sum_{t=1}^T \ell_t^{(i)}(\hat{c}_t^{(j)}, c_t^{(j)}) - \sum_{i=1}^N \sum_{t=1}^T \ell_t^{(i)}(\Theta, c_t^{(i)}) = \sum_{i=1}^N \sum_{t=1}^T \left[ \ell_t^{(i)}(\hat{c}_t^{(i)}, c_t^{(i)}) - \ell_t^{(i)}(\Theta, c_t^{(i)}) \right] + \sum_{i=1}^N \sum_{t=1}^T \left[ \ell_t^{(i)}(\hat{c}_t^{(i)}, c_t^{(i)}) - \ell_t^{(i)}(\Theta, c_t^{(i)}) \right]
\]

(24)

Then combining with Lemma 1 results in the ACE bound.

A.6 Proof of Theorem 6

Proof. Note that in this case the available measurement of \( f \) at the end of the outer iteration \( i \) is:

\[
y_t^{(j)} = Y(x_t^{(j)})\Theta c^{(j)} - w_t^{(j)}, \quad 1 \leq j \leq i, 1 \leq t \leq T.
\]

(25)

Recall that the Ridge-regression estimation of \( \Theta \) is given by

\[
\hat{\Theta}^{(i+1)} = \arg \min_{\hat{\Theta}} \|\hat{\Theta}\|_F^2 + \sum_{j=1}^i \sum_{t=1}^T \|Y(x_t^{(j)})\hat{\Theta} c^{(j)} - y_t^{(j)}\|^2
\]

(26)

\[
= \arg \min_{\hat{\Theta}} \|\hat{\Theta}\|_F^2 + \sum_{j=1}^i \sum_{t=1}^T \|Z_t^{(j)} \text{vec}(\Theta) - y_t^{(j)}\|^2.
\]

Note that \( y_t^{(j)} = (c^{(j)})^T \otimes Y(x_t^{(j)}) \cdot \text{vec}(\Theta) - w_t^{(j)} = Z_t^{(j)} \text{vec}(\Theta) - w_t^{(j)} \). Define \( V_i = \lambda I + \sum_{t=1}^i \sum_{t=1}^T Z_t^{(j)^T} Z_t^{(j)} \). Then from the Theorem 2 of [32] we have

\[
\|\text{vec}(\hat{\Theta}^{(i+1)} - \Theta)\|_F \leq R \sqrt{\frac{i}{\delta}} \sqrt{\log\left(1 + \frac{\sqrt{\lambda} T \cdot n K_2^2 K_c^2 / \delta}{\delta}\right) + \sqrt{\lambda} K_0}
\]

(27)

for all \( i \) with probability at least \( 1 - \delta \). Note that the environment diversity condition implies: \( V_i \succ \Omega(i) I \). Finally we have

\[
\|\hat{\Theta}^{(i+1)} - \Theta\|_F^2 = \|\text{vec}(\hat{\Theta}^{(i+1)} - \Theta)\|^2 \leq O\left(\frac{1}{i} O(\log(iT/\delta)) \right) = O\left(\frac{\log(iT/\delta)}{i}\right).
\]

(28)

Then with a fixed \( \hat{\Theta}^{(i+1)} \), in outer iteration \( i + 1 \) we have

\[
y_t^{(i+1)}(\hat{c}) = Y(x_t^{(i+1)})\hat{\Theta}^{(i+1)}\hat{c} - Y(x_t^{(i+1)})\Theta c^{(i+1)} + w_t^{(i+1)}.
\]

(29)

Since \( A_2 \) gives sublinear regret, we have

\[
\sum_{t=1}^T \|Y(x_t^{(i+1)})\hat{\Theta}^{(i+1)}\hat{c} - Y(x_t^{(i+1)})\Theta c^{(i+1)} + w_t^{(i+1)}\|^2
\]

(30)

\[
- \min_{\hat{c} \in \mathcal{K}_2} \sum_{t=1}^T \|Y(x_t^{(i+1)})\hat{\Theta}^{(i+1)}\hat{c} - Y(x_t^{(i+1)})\Theta c^{(i+1)} + w_t^{(i+1)}\|^2 = o(T)
\]

Note that

\[
\min_{\hat{c} \in \mathcal{K}_2} \sum_{t=1}^T \|Y(x_t^{(i+1)})\hat{\Theta}^{(i+1)}\hat{c} - Y(x_t^{(i+1)})\Theta c^{(i+1)} + w_t^{(i+1)}\|^2 \leq \sum_{t=1}^T \|Y(x_t^{(i+1)})\hat{\Theta}^{(i+1)}\hat{c} - Y(x_t^{(i+1)})\Theta c^{(i+1)} + w_t^{(i+1)}\|^2
\]

(31)

\[
\leq TW^2 + T \cdot K_Y^2 \cdot O\left(\frac{\log(iT/\delta)}{i}\right) \cdot K_c^2
\]
where (a) uses (28).

Finally we have

$$
\sum_{t=1}^{T} \| \hat{f}^{(i+1)}_t - f^{(i+1)}_t + w^{(i+1)}_t \|^2 \\
= \sum_{t=1}^{T} \| Y(x^{(i+1)}_t)\hat{\Theta}^{(i+1)}_t - Y(x^{(i+1)}_t)\Theta c^{(i+1)}_t + w^{(i+1)}_t \|^2 \\
\leq o(T) + TW^2 + O(T \frac{\log(iT/\delta)}{i})
$$

(32)

for all $i$ with probability at least $1 - \delta$. (b) is from (30), and (31). Then with Lemma 1 we have (with probability at least $1 - \delta$)

$$
ACE \leq \frac{\gamma}{1 - \rho} \sqrt{\frac{\sum_{i=1}^{N} o(T) + TW^2 + O(T \frac{\log(iT/\delta)}{i})}{TN}} \\
\leq \frac{\gamma}{1 - \rho} \sqrt{\frac{W^2 + \frac{o(T)}{T} + O(\log(NT/\delta)) \sum_{i=1}^{N} \frac{1}{i}}{TN}} \\
\leq \frac{\gamma}{1 - \rho} \sqrt{\frac{W^2 + \frac{o(T)}{T} + O(\log(NT/\delta) \log(N))}{N}}.
$$

(33)

If we relax the environment diversity condition to $\Omega(\sqrt{i})$, in (28) we will have $O(\frac{\log(iT/\delta)}{\sqrt{N}})$. Therefore in (33) the last term becomes $O(\frac{\log(iT/\delta)}{\sqrt{N}} \sum_{i=1}^{N} \frac{1}{\sqrt{i}} \leq O(\frac{\log(iT/\delta)}{\sqrt{N}})$.

A.7 Experimental details

A.7.1 Theoretical justification of Deep OMAC

Recall that in Deep OMAC (Table 4 in Section 5) the model class is $F(\phi(x; \hat{\Theta}), \hat{c}) = \phi(x; \hat{\Theta}) \cdot \hat{c}$, where $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times h}$ is a neural network parameterized by $\hat{\Theta}$. We provide the following proposition to justify such choice of model class.

**Proposition 1.** Let $\hat{f}(x, \hat{c}) : [-1, 1]^n \times [-1, 1]^h \rightarrow \mathbb{R}$ be an analytic function of $[x, \hat{c}] \in [-1, 1]^{n + h}$ for $n, h \geq 1$. Then for any $\epsilon > 0$, there exist $h(\epsilon) \in \mathbb{Z}^+$, a polynomial $\hat{\phi}(x) : [-1, 1]^n \rightarrow \mathbb{R}^{h(\epsilon)}$ and another polynomial $c(\hat{c}) : [-1, 1]^h \rightarrow \mathbb{R}^{h(\epsilon)}$ such that

$$
\max_{[x, \hat{c}] \in [-1, 1]^{n + h}} \| \hat{f}(x, \hat{c}) - \hat{\phi}(x)^T c(\hat{c}) \| \leq \epsilon
$$

and $h(\epsilon) = O((\log(1/\epsilon))^{1/h})$.

Note that here the dimension of $c$ depends on the precision $1/\epsilon$. In practice, for OMAC algorithms, the dimension of $\hat{c}$ or $c$ (i.e., the latent space dimension) is a hyperparameter, and not necessarily equal to the dimension of $\hat{c}$ (i.e., the dimension of the actual environmental condition). A variant of this proposition is proved in [34]. Since neural networks are universal approximators for polynomials, this theorem implies that the structure $\phi(x; \hat{\Theta}) \hat{c}$ can approximate any analytic function $\hat{f}(x, \hat{c})$, and the dimension of $\hat{c}$ only increases polylogarithmically as the precision increases.

A.7.2 Pendulum dynamics model and controller design

In experiments, we consider a nonlinear pendulum dynamics with unknown gravity, damping and external 2D wind $w = [w_x; w_y] \in \mathbb{R}^2$. The continuous-time dynamics model is given by

$$
ml^2 \ddot{\theta} - ml \dot{\theta} \sin \theta = u + f(\theta, \dot{\theta}, c(w)),
$$

(34)
This model generalizes the pendulum with external wind model in [35] by introducing extra modelling mismatches (e.g., gravity mismatch and unknown damping). In this model, \( \alpha_1 \) is the damping coefficient, \( \alpha_2 \) is the air drag coefficient, \( r \) is the relative velocity of the pendulum to the wind, \( F_{\text{wind}} \) is the air drag force vector, and \( \vec{l} \) is the pendulum vector. Define the state of the pendulum as \( x = [\theta; \dot{\theta}] \). The discrete dynamics is given by

\[
x_{t+1} = \begin{bmatrix}
\theta_t + \delta \cdot \frac{\dot{\theta}_t}{m l^2} \\
\dot{\theta}_t + \frac{\theta_t + \delta \cdot \dot{\theta}_t}{m l^2} + f(x_t, c) \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 \\ \delta m l^2 \end{bmatrix} (u_t + ml \ddot{\theta} \sin \theta_t + f(x_t, c)),
\]

where \( \delta \) is the discretization step. We use the controller structure \( u_t = -Kx_t - ml \ddot{\theta} \sin \theta_t - \hat{f} \) for all 6 controllers in the experiments, but different controllers have different methods to calculate \( \hat{f} \) (e.g., the no-adapt controller uses \( \hat{f} = 0 \) and the omniscient one uses \( \hat{f} = f \)). We choose \( K \) such that \( A - BK \) is stable (i.e., the spectral radius of \( A - BK \) is strictly smaller than 1), and then the e-ISS assumption in Assumption [1] naturally holds. We visualize the pendulum experiment results in fig. 2.

### A.7.3 Quadrotor dynamics model and controller design

Now we introduce the quadrotor dynamics with aerodynamic disturbance. Consider states given by global position, \( p \in \mathbb{R}^3 \), velocity \( v \in \mathbb{R}^3 \), attitude rotation matrix \( R \in \text{SO}(3) \), and body angular velocity \( \omega \in \mathbb{R}^3 \). Then dynamics of a quadrotor are

\[
\begin{align*}
\dot{p} &= v, \\
\dot{v} &= mg + Rft + f, \\
\dot{R} &= RS(\omega), \\
J\dot{\omega} &= J\omega \times \omega + \tau,
\end{align*}
\]

where \( m \) is the mass, \( J \) is the inertia matrix of the quadrotor, \( S(\cdot) \) is the skew-symmetric mapping, \( g \) is the gravity vector, \( f_t = [0, 0, T] \) and \( \tau = [\tau_x, \tau_y, \tau_z] \) are the total thrust and body torques from four rotors, and \( f = [f_x, f_y, f_z] \) are forces resulting from unmodelled aerodynamic effects and varying wind conditions. In the simulator, \( f \) is implemented as the aerodynamic model given in [36].

**Controller design.** Quadrotor control, as part of multicopter control, generally has a cascaded structure to separate the design of the position controller, attitude controller, and thrust mixer.
(allocation). In this paper, we incorporate the online learned aerodynamic force \( \hat{f} \) in the position controller via the following equation:

\[
f_d = -mg - m(K_P \cdot p + K_D \cdot v) - \hat{f},
\]

(38)

where \( K_P, K_D \in \mathbb{R}^{3\times3} \) are gain matrices for the PD nominal term, and different controllers have different methods to calculate \( \hat{f} \) (e.g., the omniscient controller uses \( \hat{f} = f \)). Given the desired force \( f_d \), a kinematic module decomposes it into the desired \( R_d \) and the desired thrust \( T_d \) so that

\[
R_d \cdot [0, 0, T_d]^\top \approx f_d.
\]

Then the desired attitude and thrust are sent to a lower level attitude controller (e.g., the attitude controller in [51]).