Appendix

1 Convergence Analysis

This section describes assumptions the proposed model relies on and proves the convergence of the learning process with the help of theories from ordinary differential equation(ode).

Since AC produces stochastic policies, each update of parameters contain deviations, then Equation 17, 18 and 21 can be re-formulated as follows:

$$\lambda = \begin{cases} \lambda_{min}, & if\ \lambda < \lambda_{min} \\ \lambda_{max}, & if\ \lambda > \lambda_{max} \\ \lambda + \eta_1 \nabla_\lambda L + M_\lambda, & if\ \lambda_{min} < \lambda < \lambda_{max} \end{cases} \tag{1}$$

$$\theta = \begin{cases} \theta_{min}, & if\ \theta < \theta_{min} \\ \theta_{max}, & if\ \theta > \theta_{max} \\ \theta + \eta_2 \nabla_\theta L + M_\theta, & if\ \theta_{min} < \theta < \theta_{max} \end{cases} \tag{2}$$

$$\nu = \begin{cases} \nu_{min}, & if\ \nu < \nu_{min} \\ \nu_{max}, & if\ \nu > \nu_{max} \\ \nu + \eta_3 \nabla_\nu \boldsymbol{A^{\pi_\theta}} + M_\nu, & if\ \nu_{min} < \nu < \nu_{max} \end{cases} \tag{3}$$

where $M_\lambda, M_\theta, M_\nu$ represent uncertainties caused by estimation error, environment noise etc. The above equations can be also regarded as the approximation scheme of the following odes:

$$\dot{\lambda} = \nabla_\lambda L \tag{4}$$

$$\dot{\theta} = \nabla_\theta L \tag{5}$$

$$\dot{\nu} = \nabla_\nu L \tag{6}$$

When the gradients are near 0, the learning process is near stable, that is, a local-optimal point is found for the Lagrangian formula. The following content states that, under certain conditions, the approximation scheme of Equation 30, 31 and 32 enables to learn a sub-optimal policy that nearly satisfies all given constraints.

Assumption 1. *There exists sub-optimal points for Equation 13 and all the points nearly satisfy the given constraints.*

Assumption 2. *Equation 17, 18 and 21 are Lipschitz and the Markov Decision Process is fully observable.*

Assumption 3.$\{M_\theta\}, \{M_\lambda\}, \{M_\nu\}$ *are martingale difference sequences, that is,* $E(M_\theta, M_\lambda, M_\nu) = 0$*. demonstraing that the mean value of the environment noise is 0 and that the magnitude of the noise is not large enough to disturb the convergence of the parameters.*

Assumption 4.*For each iteration,* λ, θ, ν *remain bounded. The dimension of the state or action space is less than 100.*

Theorem 1. *Equation 33, 34 and 35 are well defined, that is, for each initial parameter, there exist solutions for them and their trajectories converge to the equilibrium point.*

Proof 1. According to Assumption 2, all the update formulas are Lipshitz, then from [1] it can be concluded that Equation 33, 34 and 35 are well defined. And according to Assumption 4, all the parameters remain bounded so their trajectories oscillate around the equilibrium point and finally converge.

Theorem 2. $\{\lambda_n\}, \{\theta_n\}, \{\nu_n\}$*, which are the sequences of updated parameters following Equation 30, 31 and 32, track respectively odes of Equation 36, 37 and 38.*

Proof 2. When θ and ν remain unchanged, construct interpln:

$$\bar{\lambda_t} = \lambda_n + (\lambda_{n+1} - \lambda_n)\frac{t - t(n)}{t(n+1) - t(n)} \tag{7}$$

It can be observed that $sup_{t\geq 0} \left\|\bar{\lambda}(t)\right\| = sup_n \left\|\lambda_n\right\| < \infty$. Then let $\lambda^s(t), t \geq s$ denote the solution with respect to the ode where t starts at s:

$$\dot{\lambda}^s(t) = h(\lambda^s(t)), t \tag{8}$$

where $\lambda^s(s) = \bar{\lambda}(s)$. Similarly, let $\lambda_s(t), t \leq s$ denotes the solution of the ode where t ends at s:

$$\dot{\lambda}_s(t) = h(\lambda_s(t)), t \geq s \tag{9}$$

where$\lambda_s(s) = \bar{\lambda}(s)$. Let

$$\zeta_n = \sum_{m=0}^{n-1} a(m) M_{m+1}, n \geq 1 \tag{10}$$

by the assumption that M is a martingale sequence, it can be concluded that $\zeta_n, n \geq 1$ is also a martingale sequence. With the additional assumption concerning the learing rate and the bounded interation it can be concluded that:

$$\sum_{n\geq 0} E[\left\|\zeta_{n+1} - \zeta_n\right\|^2] = \sum_{n\geq 0} a(n)^2 E[\left\|M_{n+1}\right\|^2] < \infty \tag{11}$$

and from [2] it is concluded that:

$$\lim_{s\to\infty} sup_{t\in[s,s+t]} \left\|\bar{\lambda}(t) - \lambda(t)\right\| = 0 \tag{12}$$

which means that the interpln will eventually move close to the ode. θ and ν follow the same pattern.

Theorem 3. *If $\eta_1(t), \eta_2(t), \eta_3(t)$ satisfy $\sum_t \eta_1(t) = \infty, \sum_t \eta_2(t) = \infty, \sum_t \eta_3(t) = \infty, \sum_t \eta_1^2(t) < \infty, \sum_t \eta_2^2(t) < \infty, \sum_t \eta_3^2(t) < \infty$ and $\eta_3 > \eta_2$, and Algorithm 3 is implemented where the sensitivity measure is used, Algorithm 2 finally converges in finite time steps.*

Proof 3. Consider a coupled iteration with the singularly perturbed ode:

$$\dot{x}(t) = \frac{1}{\sigma} h(x(t), y(t)) \tag{13}$$

$$\dot{y}(t) = g(x(t), y(t)) \tag{14}$$

When σ tends to be 0, $x(t)$ is a fast transient and $y(t)$ is a much slower transient which can be treated as near static with respect to $x(t)$, that is, nearly a constant. Then for the ode:

$$\dot{x}(t) = h(x(t), y) \tag{15}$$

y can be treated as a constant parameter. Suppose also that the ode has a stable point $\lambda(y)$ where λ is Lipshitz. Then when σ is small enough, $x(t)$ tracks $\lambda(y(t))$ for $t > 0$, wich means that the ode:

$$\dot{y}(t) = g(\lambda(y(t)), y(t)) \tag{16}$$

approximates $y(t)$. For the requirements of the summed learning rates, see[2]. When $\eta_3 > \eta_2$, the update procedures of them form a two-time-scale approximation: since η_2 changes slowly compared with η_3, when η_3 is updated, it sees η_2 nearly as a constant; and when η_3 is updated, η_2 has already converged. According to Algorithm 2, η_1 is updated only when the constraint performances remain stable which indicated that η_2 and η_3 remain stable, or move away from a feasible solution. Then the whole procedure behaves the same as a three-time-scale approach and finally converges.

2 Exploration Mechanism

This section introduces an exploration mechanism where the agent explores more effectively than with stochastic exploration. A set container is maintained during the learning process to record environment states visited by the agent and the number of visits of each state. Each time when a state is visited, the set container appends the state not visited before, otherwise it updates the information of the corresponding state. Since the states of a robot is usually a vector whose elements are continuous, each element in the vector is processed as follows:

$$s^{'} = round(\frac{s - s_m}{s_{max}}, n) \tag{17}$$

where s is the original state, $s^{'}$ is the processed state, $round$ is the rounding operation controlled by precision n, s_m is the average of all elements of the state vector, s_{max} is the largest element of the state vector. The equation indicates that each element will be in range (0,1) and that similar state vectors are cast into the same $s^{'}$. n controls the overall number of state vectors to be recorded in the set container which reflects the trade-off between the learning performance and memory space.

Exploration Reward The exploration reward is generated with respect to the goal point characterized by the environment states that are rarely visited. The goal point can be regarded as a bridge that connects the explored area and unknown area. So when the agent reaches the goal point, there is high probability that more novel states will be visited. At the end of each episode during the learning process, the agent update its goal point when the current goal point has been reached.The exploration reward is calculated as follows:

$$r_e = \frac{1}{n_s} + min(max(d_2 - d_1, 0), 1) \tag{18}$$

where n_s is the number of visits of state s, d_1 and d_2 are the distance between the current agent position and the goal point. Then the overall exploration performance can be estimated following Equation 20 and be treated as a constraint.

3 Experiment Settings and Results

The proposed model in this study is tested on robotic tasks from PyBullet Gymperium from OpenAI, including Walker2D, Humanoid, Hopper, Halfcheetah, Ant, Snakebot and Minitaur. The agent needs to learn a policy for the robot to remain balanced and walk towards a certain location. The reward structure of each robot consists of five parts: the aliveness, the progress, the electronic cost, the stuck joint cost and the foot collision cost. During the experiment, the collision cost is always 0 and at later stage of the learning process the aliveness reward tends to be a constant so the the progress, the electronic cost and the stuck joint cost are regarded as effective constraints and the sum of the five parts is regarded as the main objective. Then the criterion for evaluating model performance is whether the model satisfies all given constraint thresholds and ensures the main objective is maximized. The exploration performance is measured by the number of position points visited by the robot.

The state space of each robot is a 22-dimenstion vector and the action space is a 12-demenstion vector.

Parameter Settings All model parameters are described and initialized as in Table 1. Note that θ and ν are initialized by the network.

Constraints Classification The constraints and main objective mentioned above are initially grouped by running Algorithm 2 and the average results of five environments are shown in Table 2, where the positive or negative number implies that two constraints are positively or negatively correlated and the absolute value represents the degree of the relation. According to the results, the main objective and the progress are highly positive-correlated, and the electronic cost and joint cost are positive-correlated although the joint cost is relatively independent of others. Then the main objective and progress form a group and the electronic cost and joint cost form another group.

Algorithm Settings We compare SCPOCA with PPO, RCPO and PPO with reward shaping, that is, learning with a predefined λ. Since original RCPO concerns only one constraint, in the experiment, it is augmented with constraint aggregation, so is the PPO with reward shaping. The constraints are imposed on the basis of original PPO where the electronic cost and joint cost are required to improve 20% while the performance of progress and the main objective are required to decrease by no more than 20%.

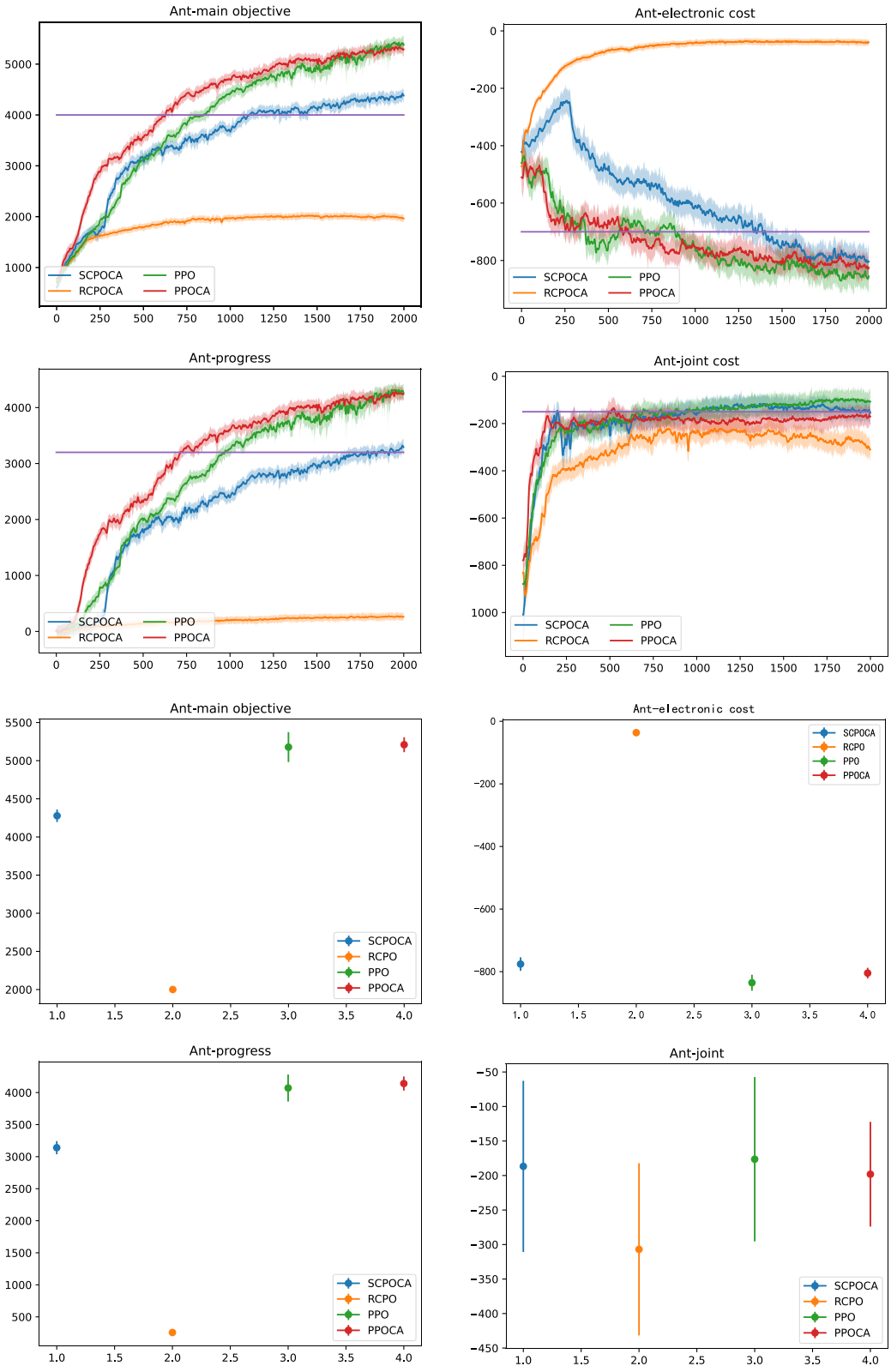

Figure 1: Average Results of Ant

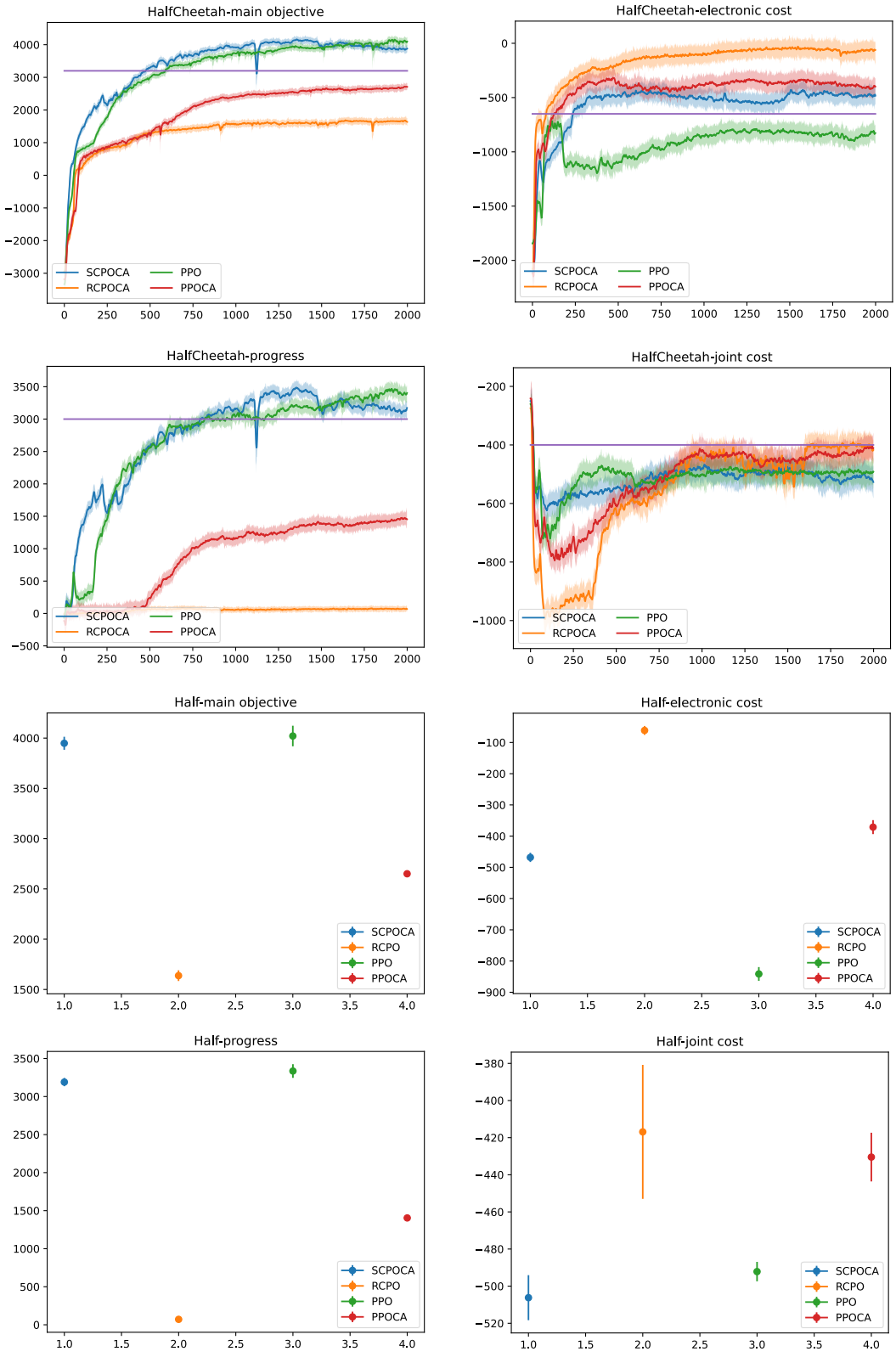

Figure 2: Average Results of Halfcheetah

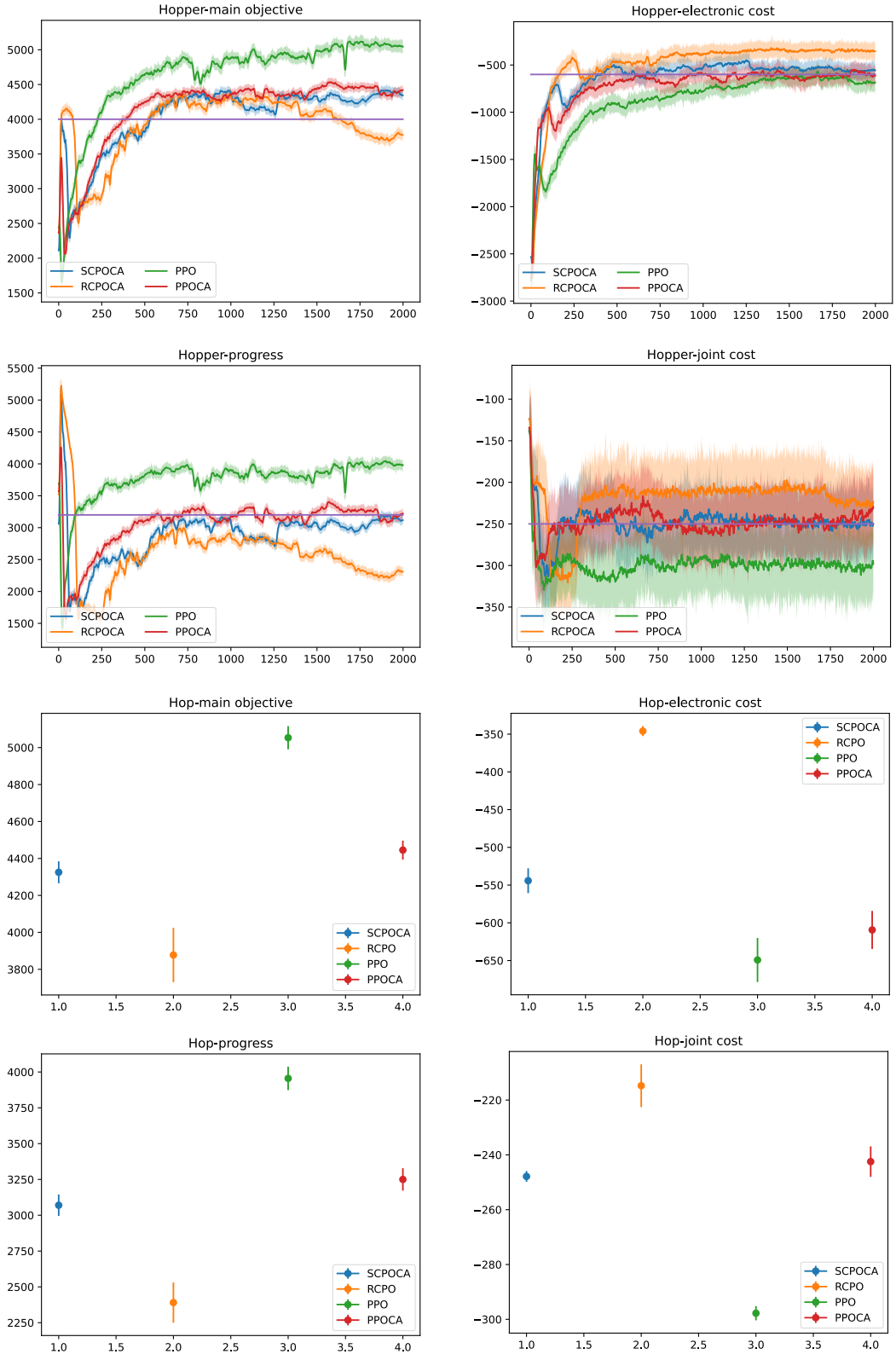

Figure 3: Average Results of Hopper

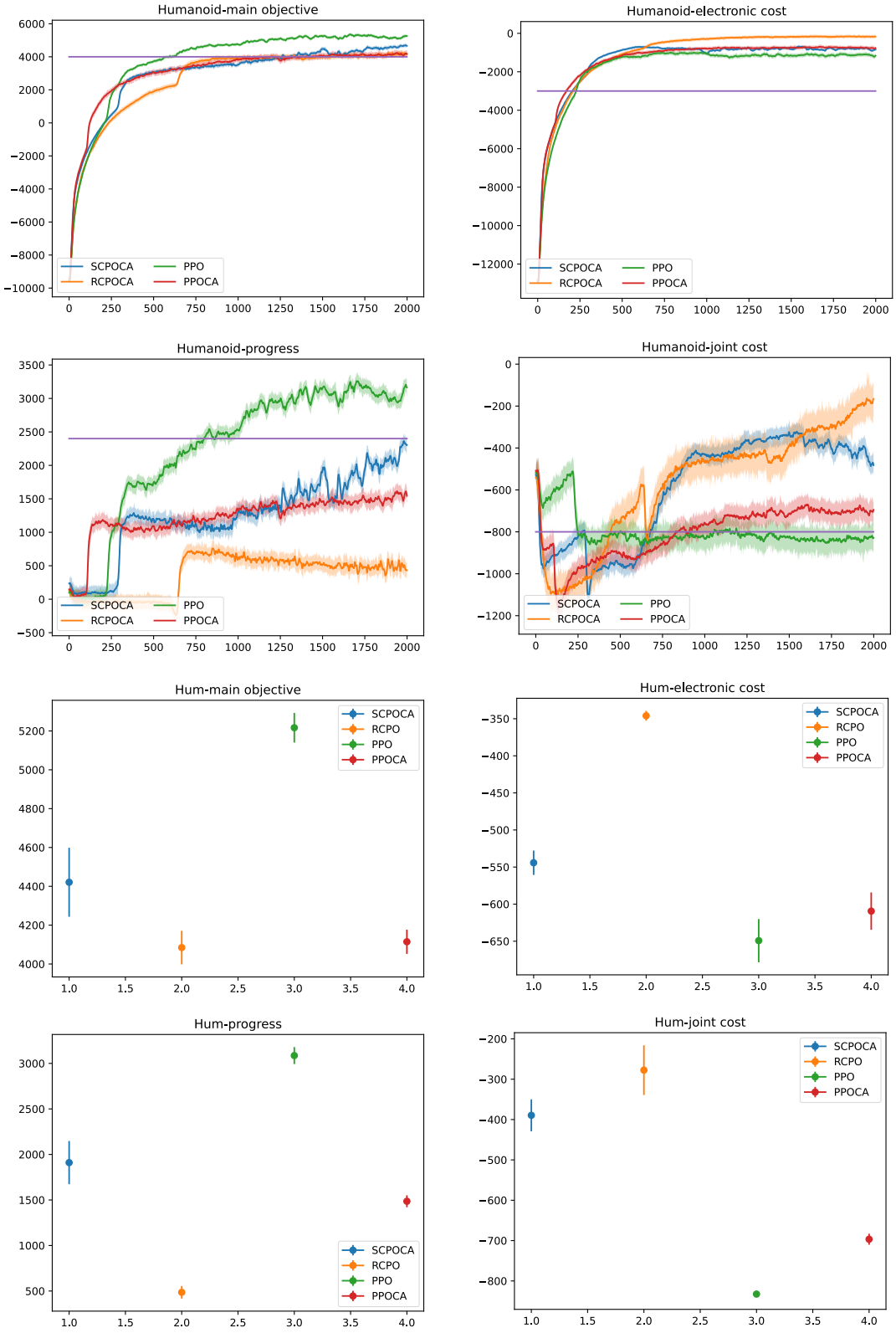

Figure 4: Average Results of Humanoid

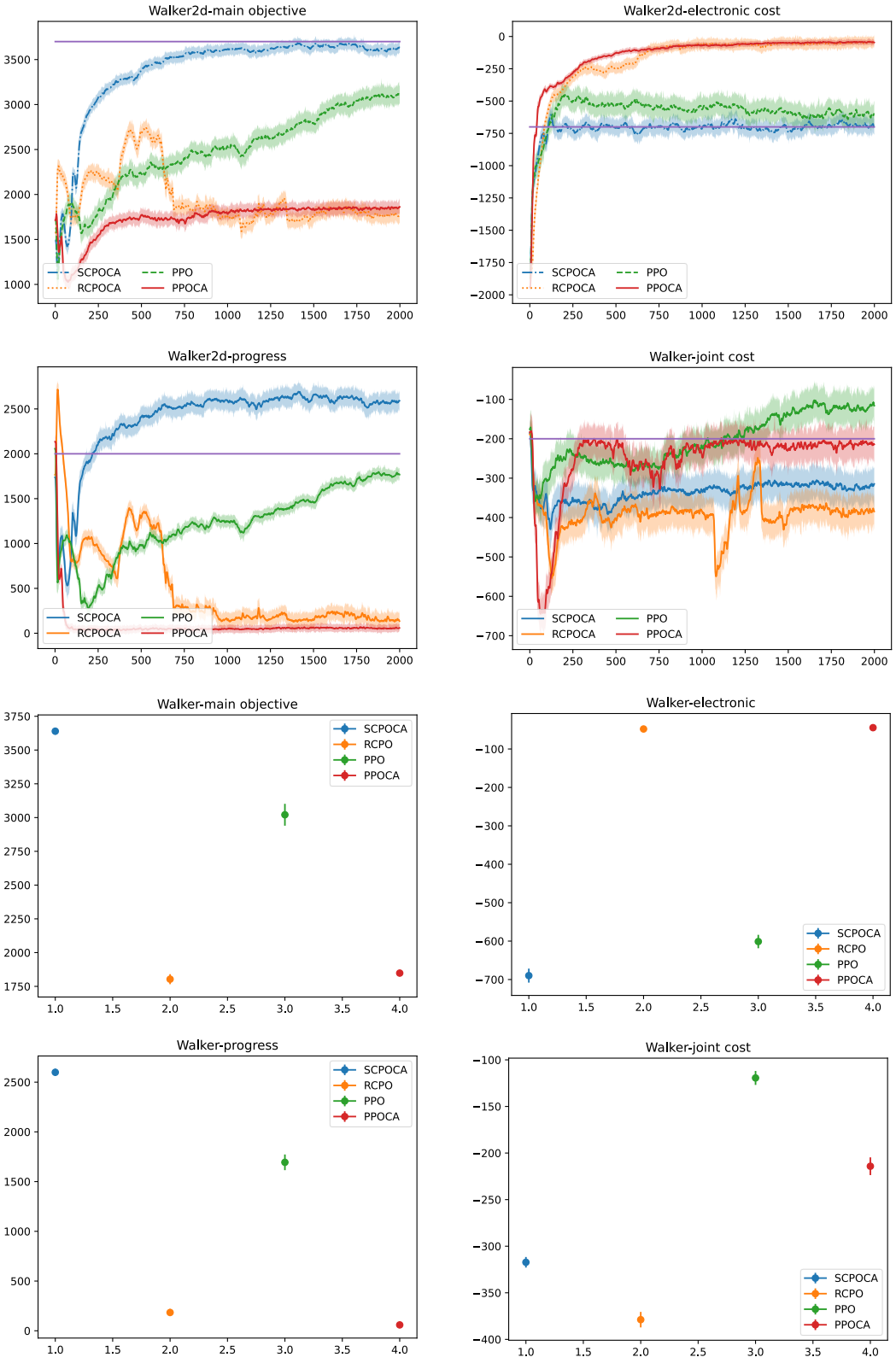

Figure 5: Average Results of Walker2D

Table 1: Values of model parameters

	definition	**value**
$\boldsymbol{U}$	update frequency	200
$\boldsymbol{EP\_NUM}$	number of episodes	2000
$\boldsymbol{H}$	number of time steps for each episode	2048
η_1	learning rates of λ	1×10^{-7}
η_2	learning rates of θ	1×10^{-6}
η_3	learning rates of ν	1×10^{-5}
C_1, C_2	stabilization parameters	1000
-	hidden layer size of neural networks	128×1
λ_0	initial value of λ	0.5

Table 2: relations of different constraints and the main objective

	main objective	**progress**	**electronic cost**	**joint cost**
main objective	-	265.013	-59.091	0.134
progress	265.013	-	-79.191	-0.267
electronic cost	-59.091	-79.191	-	0.097
joint cost	0.134	-0.267	0.097	-

References

[1] Hamed Fathalizadeh Parapari and Mohammad Bagher Menhaj. Solving nonlinear ordinary differential equations using neural networks. pages 351–355, Qazvin, Iran, 2016. IEEE.

[2] V. S. Borkar. An actor-critic algorithm for constrained markov decision processes. 54:207–213, 2005.