We thank the reviewers for the reviews, providing meaningful insight with constructive feedback. Due to the page limitation, we only provide critical values. In the final manuscript, we will present all the results, alongside modifications reflecting reviewers' comments which are not mentioned in this response.

R1: Result on Humanoid environment. We tested our method on Humanoid-v2 and confirmed our method works properly. The relative baseline and adaptive variance update algorithm, which performs best among proposed algorithms, was tested on the environment and scored **6169** $\pm$ **455** with ten random seeds at 1M steps.

R1: Interaction of ES and PG workers. We measured how many RL and EA actors were contributed in improving the performance, as a summation of the update ratio p (Eq. 6), with higher value indicating more contribution. In our method, RL actors contributed twice more compared to ES actors in HalfCheetah, with values of **214.53** and **105.52**, respectively. The result was reversed in Hopper, where RL contributed **200.86** while EA actors did **363.53**.

R2, R3: Evaluation method for performance and speed. We evaluated our algorithm in two perspectives; performance improvement and speed improvement. For the performance improvement, we evaluated our method as same as the baselines for a fair comparison. In many papers, the final score of the fixed interaction step is frequently used for evaluation metrics. Therefore, all performance result scores are measured in the fixed interaction step. For the speed improvement, we measured execution wall-clock time for the fixed interaction step; the result is presented in Table 1. As the CEM-RL is implemented in a serial-synchronous, we modified the algorithm to a parallel-synchronous version (P-CEM-RL), and then we compared these algorithms with our method to show the efficiency of the asynchronism. We will include the learning curve in the final version.

R2: Ablation study is missing. Our algorithm mainly consists of three aspect; asynchronism, mean and variance update rules. We presented the effect of a simple asynchronous method [25] with the CEM-RL update rule in column "Rank-based" of Table 2. We presented the effect of the variance update rule in Appendix C.3 by comparing the result with a fixed variance setting. Then, we provided all combinations of our proposed mean and variance in Table 2. However, we agree with the reviewer because all these results are shown separately and not discussed thoroughly in the manuscript. We will add a section so that it can be seen at a glance. If these results are still not enough for an ablation study, it would be beneficial for us to consolidate our manuscript if the reviewer can provide us more specific guideline.

R2: Asynchronism in CEM-RL is not "impossible". We used the word "impossible" to emphasize that the update rule of CEM-RL cannot be used exactly the same in an asynchronous setting. CEM-RL spawns all actors at the same time with a fixed number of each agent. In an asynchronous setting, some modifications should be applied, such as alternatively spawning RL and ES actors. We intended to highlight the difference; however, we agree that the word is too aggressive. We will soften the tone in the final version.

R2: (1+1)-ES is not a fitness-based method. We categorized (1+1)-ES as a fitness-based method because it uses fitness values for comparing. However, as the reviewer pointed out, the method is ambiguous to be categorized. We will add detailed explanation in the final version.

R2: No reference or evidence about the statement "Capability of high exploration in fitness-based scheme". Our phrase "high exploration" is intended to empathize with the aggressiveness when there appears a superior individual. However, there might be a misunderstanding about the general meaning of exploration in the policy search field. We will modify the statement in the final version.

R3, R4: Speed up is the factor of 2 or 3. Time efficiency experiment was conducted with only one setting. We used five actors as provided in the section 4.1. The number of actors is not limited, but it requires GPU calculation, limiting the experiment in our setting. We additionally tested our methods with various actors of 2, 3, 4, 5, and 7 in Halfcheetah. The running times were **75%**, **42%**, **37%**, **32%**, and **25%** compared to the execution time of the CEM-RL. It shows that time efficiency is linearly increased as number of actors increases. We will provide broader experiments and discussions about the number of actors, with a table and also a graph.

R3, R4: Ask and Tell based update rules are missing. Using variance instead of covariance. We took a look into the Nevergrad library and read the original papers of implemented algorithms. We will try to merge various algorithms that fit the update scheme of combining ES and RL. It seems though some algorithms are hard to be merged. Our network consists of only three layers with 100k parameters, which is very shallow compared to the networks in the computer vision field. However, algorithms that use covariance like CMA-ES are not appropriate with 100k parameters. It is also the reason why the baseline CEM-RL used variance instead of covariance.

R4: Stability metric is not provided. We defined the term "stability" for consistently showing high performances, thereby having low std value per mean (σ/μ). Our final method AES-RL, relative baseline with the adaptive update, is chosen because its performance is high, and the σ/μ value is low. As shown in Table 2, previous algorithms for asynchronous updates show a higher σ/μ compared to our method, therefore we claimed that our method is relatively stable. We will explicitly explain the metric in the plain text and also emphasize it in the tables.