

## 472 A Hyperparameters and Infrastructures

473 For all experiments on the SMAC benchmark, we use the default reward and observation settings. For  
 474 our method, we set the discount factor  $\gamma$  to 0.99 for all experiments. We use RMSprop with  $\alpha = 0.99$   
 475 and no momentum or weight decay for the optimization of both the critic and actors. The learning  
 476 rate for the critic is 0.0001 and the learning rate for actors is 0.0005. The critic and actors have the  
 477 same network architecture as DOP [51]. 16 episodes are sampled from the on-policy buffer each time  
 478 to train both the critic and actors. The on-policy buffer has a buffer size of 32. We run 4 parallel  
 479 environments to collect data.  $\epsilon$ -greedy is used during exploration. We let  $\epsilon$  first anneal linearly from  
 480 1.0 to 0.05 over 500k time steps and then keep constant for the rest of the training. All experiments  
 481 are conducted on NVIDIA GEFORCE RTX 3090 GPUs and Intel Xeon Gold 6248R CPUs. We use 1  
 482 GPU for each experiment.

## 483 B Choice of $\beta$

484 When computing  $\hat{\delta}$ , we select a  $\beta$  such that  $D_{\text{KL}}(\pi'_k \| \pi_k) \leq KL^U$ . In this way,  $\mathcal{L} = -f_A H(\pi'_k) -$   
 485  $f_A D_{\text{KL}}(\pi'_k \| \pi_k) - (1 - f_A) H(\pi_k) \geq -f_A H(\pi'_k) - f_A KL^U - (1 - f_A) H(\pi_k)$  can be bounded.  
 486 There can be multiple values of  $\beta$  such that  $D_{\text{KL}}(\pi'_k \| \pi_k) \leq KL^U$  is satisfied. We set  $\beta$  to the largest  
 487 one. To find such a  $\beta$ , we first prove in Lemma. B.1 that  $D_{\text{KL}}(\pi'_k \| \pi_k)$  monotonically increases as  $\beta$   
 488 increases.

489 **Lemma B.1.** *Let  $N$  be the number of actions, i.e.  $N = |A|$ . Suppose the policy  $\pi$  is the softmax of*  
 490 *the logits  $l$ , and  $\hat{\delta}$  is a vector of the same dimension as  $l$ . Let  $\pi'(\beta) = \text{softmax}(l + \beta\hat{\delta})$ . Then for*  
 491  *$\beta \geq 0$ ,  $\frac{dD_{\text{KL}}(\pi'(\beta) \| \pi)}{d\beta} \geq 0$ .*

492 *Proof.* Let  $Z = \sum_{a=1}^N \exp(l(a))$ ,  $Z'(\beta) = \sum_{a=1}^N \exp(l(a) + \beta\hat{\delta}(a))$ , then

$$\frac{dD_{\text{KL}}(\pi'(\beta) \| \pi)}{d\beta} = \frac{d}{d\beta} \left[ \sum_{a=1}^N \frac{\exp(l(a) + \beta\hat{\delta}(a))}{Z'(\beta)} \log \frac{\exp(l(a) + \beta\hat{\delta}(a))/Z'(\beta)}{\exp(l(a))/Z} \right] \quad (14)$$

$$= \sum_{a=1}^N \frac{d}{d\beta} \left( \frac{\exp(l(a) + \beta\hat{\delta}(a))}{Z'(\beta)} \right) \log \frac{\exp(l(a) + \beta\hat{\delta}(a))/Z'(\beta)}{\exp(l(a))/Z} +$$

$$\sum_{a=1}^N \frac{\exp(l(a) + \beta\hat{\delta}(a))}{Z'(\beta)} \frac{d}{d\beta} \left( \log \frac{\exp(l(a) + \beta\hat{\delta}(a))/Z'(\beta)}{\exp(l(a))/Z} \right). \quad (15)$$

493 Because

$$\frac{d}{d\beta} \left( \frac{\exp(l(a) + \beta\hat{\delta}(a))}{Z'(\beta)} \right) = \frac{\exp(l(a) + \beta\hat{\delta}(a))\hat{\delta}(a)}{Z'(\beta)} - \frac{\exp(l(a) + \beta\hat{\delta}(a))}{Z'(\beta)^2} \frac{dZ'(\beta)}{d\beta} \quad (16)$$

494 and

$$\frac{d}{d\beta} \left( \log \frac{\exp(l(a) + \beta\hat{\delta}(a))/Z'(\beta)}{\exp(l(a))/Z} \right) = \frac{d}{d\beta} \log \left( \frac{\exp(l(a) + \beta\hat{\delta}(a))}{Z'(\beta)} \right) \quad (17)$$

$$= \frac{\frac{d}{d\beta} \exp(l(a) + \beta\hat{\delta}(a))}{\exp(l(a) + \beta\hat{\delta}(a))} - \frac{\frac{d}{d\beta} Z'(\beta)}{Z'(\beta)} \quad (18)$$

$$= \hat{\delta}(a) - \frac{\frac{d}{d\beta} Z'(\beta)}{Z'(\beta)}, \quad (19)$$

495 by incorporating Eq. 16 and 19 into Eq. 15, we have

$$\frac{dD_{\text{KL}}(\pi'(\beta) \| \pi)}{d\beta} \quad (20)$$

$$= \sum_{a=1}^N \left[ \frac{\exp(l(a) + \beta\hat{\delta}(a))\hat{\delta}(a)}{Z'(\beta)} - \frac{\exp(l(a) + \beta\hat{\delta}(a))}{Z'(\beta)^2} \frac{dZ'(\beta)}{d\beta} \right] \left[ \log \frac{\exp(l(a) + \beta\hat{\delta}(a))/Z'(\beta)}{\exp(l(a))/Z} \right] +$$

$$\sum_{a=1}^N \frac{\exp(l(a) + \beta \hat{\delta}(a))}{Z'(\beta)} [\hat{\delta}(a) - \frac{d}{d\beta} Z'(\beta)] \quad (21)$$

$$= \sum_{a=1}^N \frac{\exp(l(a) + \beta \hat{\delta}(a))}{Z'(\beta)} [\hat{\delta}(a) - \frac{1}{Z'(\beta)} \frac{dZ'(\beta)}{d\beta}] [\beta \hat{\delta}(a) + \log \frac{Z}{Z'(\beta)}] + \sum_{a=1}^N \frac{\exp(l(a) + \beta \hat{\delta}(a))}{Z'(\beta)} [\hat{\delta}(a) - \frac{1}{Z'(\beta)} \frac{dZ'(\beta)}{d\beta}] \quad (22)$$

$$= \beta \sum_{a=1}^N \frac{\exp(l(a) + \beta \hat{\delta}(a))}{Z'(\beta)} [\hat{\delta}(a) - \frac{1}{Z'(\beta)} \frac{dZ'(\beta)}{d\beta}] \hat{\delta}(a) + [\log \frac{Z}{Z'(\beta)} + 1] \sum_{a=1}^N \frac{\exp(l(a) + \beta \hat{\delta}(a))}{Z'(\beta)} [\hat{\delta}(a) - \frac{1}{Z'(\beta)} \frac{dZ'(\beta)}{d\beta}]. \quad (23)$$

496 Let  $\pi'(a; \beta) = \frac{\exp(l(a) + \beta \hat{\delta}(a))}{Z'(\beta)}$ , then

$$\frac{1}{Z'(\beta)} \frac{dZ'(\beta)}{d\beta} = \sum_{a=1}^N \frac{\exp(l(a) + \beta \hat{\delta}(a)) \hat{\delta}(a)}{Z'(\beta)} = \sum_{a=1}^N \pi'(a; \beta) \hat{\delta}(a). \quad (24)$$

497 Incorporating Eq. 24 to Eq. 23, we have

$$\begin{aligned} \frac{dD_{\text{KL}}(\pi'(\beta) \parallel \pi)}{d\beta} &= \beta \sum_{a=1}^N \pi'(a; \beta) [\hat{\delta}(a) - \sum_{a'=1}^N \pi'(a'; \beta) \hat{\delta}(a')] \hat{\delta}(a) + \\ &\quad [\log \frac{Z}{Z'(\beta)} + 1] \sum_{a=1}^N \pi'(a; \beta) [\hat{\delta}(a) - \sum_{a'=1}^N \pi'(a'; \beta) \hat{\delta}(a')] \end{aligned} \quad (25)$$

$$= \beta [(\sum_{a=1}^N \pi'(a; \beta) \hat{\delta}(a)^2) - (\sum_{a=1}^N \pi'(a; \beta) \hat{\delta}(a))^2] \quad (26)$$

$$= \beta [(\sum_{a=1}^N \pi'(a; \beta) \hat{\delta}(a)^2) (\sum_{a=1}^N \pi'(a; \beta)) - (\sum_{a=1}^N \pi'(a; \beta) \hat{\delta}(a))^2]. \quad (27)$$

498 By Cauchy–Schwarz inequality,  $(\sum_{a=1}^N \pi'(a; \beta) \hat{\delta}(a)^2) (\sum_{a=1}^N \pi'(a; \beta)) - (\sum_{a=1}^N \pi'(a; \beta) \hat{\delta}(a))^2 \geq$   
 499 0. Because  $\beta \geq 0$ ,  $\frac{dD_{\text{KL}}(\pi'(\beta) \parallel \pi)}{d\beta} \geq 0$ .  $\square$

500 Based on this lemma, we can use a binary search algorithm to find the desired  $\beta$  with efficiency.  
 501 The algorithm is shown in Alg. 1. In this algorithm, we use two hyperparameters,  $BSN_1$  and  
 502  $BSN_2$ , to control the number of iterations of the binary search algorithm. In our experiments, we set  
 503  $BSN_1 = 55$  and  $BSN_2 = 15$ .

## 504 C Experimental Settings

505 In this section, we describe the detailed settings of MPE tasks in our experiments.

506 **Spread:** There are 3 agents and 3 landmarks in a  $5 \times 5$  grid. Agents need to occupy all 3 landmarks  
 507 at the same time. Agents can observe both its location and the relative location of other agents and all  
 508 landmarks. For each landmark  $i$ , let  $d_i$  be the distance from the nearest agent. The reward is  $-\sum d_i$   
 509 subtracting the number of collisions.

510 **Gather:** There are 3 agents and 1 landmark in a  $5 \times 5$  grid. Agents need to gather at the landmark  
 511 simultaneously to get a reward. Agents can observe both its location and the relative location of other  
 512 agents and the landmark. The reward is the sum of the negative distance between every agent and the  
 513 landmark.

---

**Algorithm 1** Find  $\beta$  by Binary Search
 

---

**Input:**  $l_k, \hat{\delta}, KL^U, BSN_1, BSN_2$ 
**Output:**  $\beta$ 

```

1:  $\pi_k := \text{softmax}(l_k)$ 
2:  $l := 0, r := 1, bsn := 0$ 
3: for  $i = 1$  to  $BSN_1$  do
4:    $\pi'_k := \text{softmax}(l_k + r\hat{\delta})$ 
5:    $bsn := i - 1$ 
6:   if  $D_{\text{KL}}(\pi'_k \| \pi_k) > KL^U$  then
7:     Break
8:   end if
9:    $r := r * 2$ 
10: end for
11: for  $i = 1$  to  $BSN_2 + bsn$  do
12:    $m := \frac{l+r}{2}$ 
13:    $\pi'_k := \text{softmax}(l_k + m\hat{\delta})$ 
14:   if  $D_{\text{KL}}(\pi'_k \| \pi_k) > KL^U$  then
15:      $r := m$ 
16:   else
17:      $l := m$ 
18:   end if
19: end for
20: return  $l$ 

```

---

514 **Formation:** There are 4 agents and 1 landmark in a  $5 \times 5$  grid. Agents need to form a square  
 515 whose center is the landmark. Agents can observe both its location and the relative location of other  
 516 agents and the landmark. For each agent  $i$ , let  $d_i$  be the distance from the landmark, and  $\alpha_i$  be the  
 517 angle formed by the  $x$ -axis and the segment connecting it and the landmark. When calculating the  
 518 reward, we first rearrange the order of agents so that  $\alpha_i \leq \alpha_{i+1}$ . Let  $\beta_i = \alpha_i - i\frac{2\pi}{4}$ . The reward is  
 519  $-\sum_{i=1}^4 (|\beta_i - \bar{\beta}| + |d_i - \bar{d}|)$ , where  $\bar{\beta} = \frac{1}{4} \sum_{i=1}^4 \beta_i$  and  $\bar{d} = \frac{1}{4} \sum_{i=1}^4 d_i$ .

## 520 D More Experimental Results

521 **Influence of the attack frequency and the KL-divergence upper bound:** We provide more  
 522 experiments under different attack budgets in Fig. 7 and Fig. 8. From these results, we can see that  
 523 better attack performance can be obtained with a higher attack frequency or a larger KL upper bound  
 524 for our methods. By contrast, random attacks are less affected. These results are in line with our  
 525 observation in the main text that our method find a better attack direction.

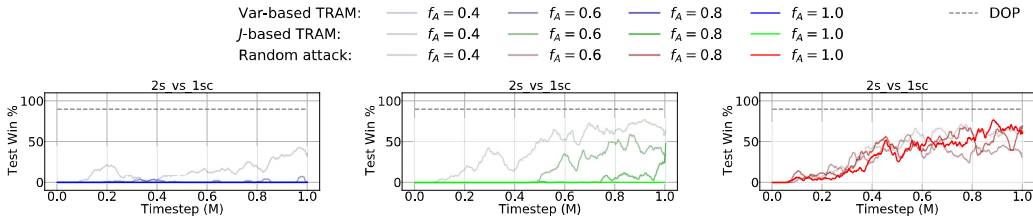


Figure 7: Performance of our methods and the random attack baseline under different attack frequencies on the map 2s\_vs\_1sc. For these experiments,  $KL^U = 6$ .

526 We also show the log likelihood of being abnormal due to attacks for our methods and the random  
 527 attack baseline under different attack frequencies and KL-divergence upper bounds in Table 2 and  
 528 Table 3. A larger likelihood value in these tables indicates that the attacked policy is more like

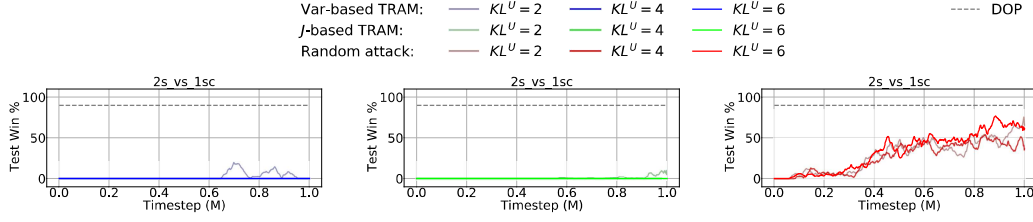


Figure 8: Performance of our methods and the random attack baseline under different KL-divergence upper bounds on the map 2s\_vs\_1sc. For these experiments,  $f_A = 1$ .

the intact policy. From these results, we can draw a similar conclusion as in the main text that our methods make a better use of the limited attack budget.

Table 2: Log likelihood under different attack frequencies. For these experiments,  $KL^U = 6$ .

map	2s3z				3m				6m				2s_vs_1sc			
Frequency	0.4	0.6	0.8	1.0	0.4	0.6	0.8	1.0	0.4	0.6	0.8	1.0	0.4	0.6	0.8	1.0
Var-based TRAM	-1.24	-1.90	-2.31	-3.41	-1.23	-1.76	-2.13	-3.87	-1.12	-2.21	-3.01	-4.32	-1.89	-2.79	-3.27	-3.36
J-based TRAM	-0.72	-0.96	-1.27	-1.65	-0.80	-1.13	-1.67	-2.43	-1.03	-1.42	-1.81	-2.88	-0.98	-1.34	-1.82	-3.08
Random attack	-0.74	-1.03	-1.35	-1.76	-0.77	-1.05	-1.28	-1.53	-0.97	-1.35	-1.75	-2.28	-1.09	-1.53	-1.96	-2.45

Table 3: Log likelihood under different KL upper bounds. For these experiments,  $f_A = 1$ .

map	2s3z			3m			6m			2s_vs_1sc		
$KL^U$	2	4	6	2	4	6	2	4	6	2	4	6
Var-based TRAM	-2.09	-2.50	-3.41	-2.43	-3.74	-3.87	-2.82	-4.20	-4.32	-3.01	-3.23	-3.36
J-based TRAM	-1.13	-1.65	-1.65	-1.53	-2.48	-2.43	-1.68	-2.76	-2.88	-1.67	-3.06	-3.08
Random attack	-1.21	-1.54	-1.76	-1.26	-1.58	-1.53	-1.26	-2.16	-2.28	-1.74	-2.38	-2.45

**Attacking other MARL algorithms:** We change the attack target algorithm from DOP to another policy-based MARL algorithm, COMA [9] and show the result of both our methods and random attack on several SMAC maps in Fig. 9. From these results we can see that on COMA, Var-based TRAM is more effective than J-based TRAM, and random attack is the weakest attack.

**TRAM on higher dimensional environments:** We compare our methods with the random attack baseline (all attacking DOP) on two environments with higher dimension, 15m and 20m, to justify the scalability of our methods. The result is shown in Fig. 10. Var-based TRAM has the most significant influence, and J-based TRAM also undermines the performance a lot. The random attack baseline has little influence on DOP and even slightly improves the training performance on 20m.

**Results with confidence intervals of experiments under different attack budgets** In Fig. 5 and Fig. 6, because curves are distinguished from each other by transparencies and showing confidence intervals may make some curves unclear, we hide the confidence intervals. We show these two figures with confidence intervals in Fig. 11 and Fig. 12.

## E Limitations and Future Directions

In our work, we make an approximation for the first term of Eq. 8,  $\nabla_{\pi} \rho(s, \mathbf{a})$ . This can lead to an inaccurate value of  $\hat{\delta}$  computed by our methods. This might be exaggerated especially for long-horizon tasks, because a contaminated action may change the experience distribution of all following timesteps. One way to alleviate this issue can be training an FDM (forward dynamics model) for the environment to accurately model the change of state-action distribution caused by policy changes.

Another limitation of our method is that in both attack methods, we use the critic (or advantage function  $A$ ) and the policy  $\pi$  to compute  $\hat{\delta}$ . This makes an additional assumption that the agents' critic and policy should be known. In some realistic scenarios where this information is not available, our attack methods will not be effective. One way to solve this issue can be using imitation learning to approximate the policy and the advantage function.

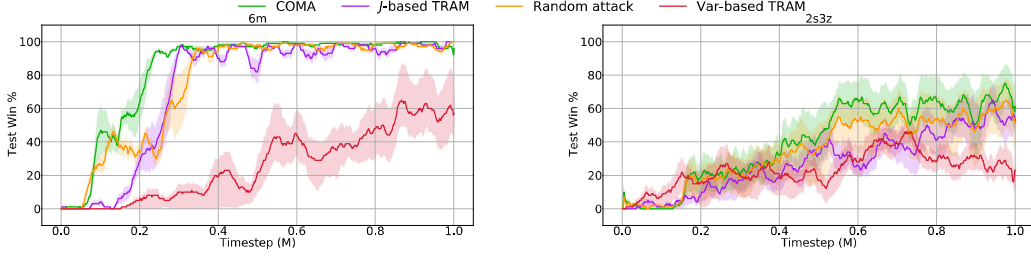


Figure 9: Performance of our methods and a random attack baseline attacking COMA on SMAC maps.

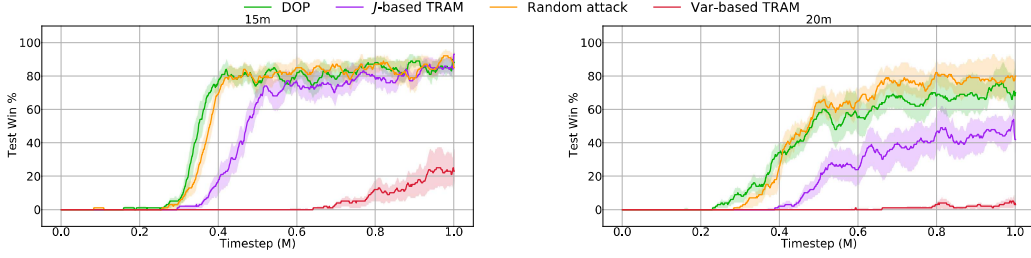


Figure 10: Performance of our methods and a random attack baseline (attacking DOP) on higher dimensional environments.

The two methods proposed in this work are actually maximizing one optimization goal:

$$E_{\rho(s, \mathbf{a}|\delta)}[|g(s, \mathbf{a}) - \nabla_{\theta^t} J(\theta_t)|^2] = E_{\rho(s, \mathbf{a}|\delta)}[|g(s, \mathbf{a})|^2 - 2g(s, \mathbf{a})^\top \nabla_{\theta^t} J(\theta_t) + |\nabla_{\theta^t} J(\theta_t)|^2]$$

555 , where  $g(s, \mathbf{a}) = A(s, \mathbf{a}) \nabla_{\theta^t} \log \pi_{\theta^t}(\mathbf{a}|\tau)$  and  $\rho(\cdot|\delta)$  follows the same definition as  $\rho(\cdot)$  in the  
 556 main text (just to emphasize the influence of  $\delta$  to  $\rho$ ). The variance-based method increases the first  
 557 term ( $E_{\rho(s, \mathbf{a}|\delta)}[|g(s, \mathbf{a})|^2]$ ) on the right hand side (RHS), the J-based method decreases the second  
 558 term ( $E_{\rho(s, \mathbf{a}|\delta)}[g(s, \mathbf{a})^\top \nabla_{\theta^t} J(\theta_t)]$ ), and the third term ( $E_{\rho(s, \mathbf{a}|\delta)}[|\nabla_{\theta^t} J(\theta_t)|^2]$ ) is not affected by  
 559  $\delta$ . Intuitively, maximizing this optimization goal increases the expected distance between the  
 560 contaminated and the original policy gradients. Therefore, another future direction could be to  
 561 investigate the attack when our methods are incorporated.

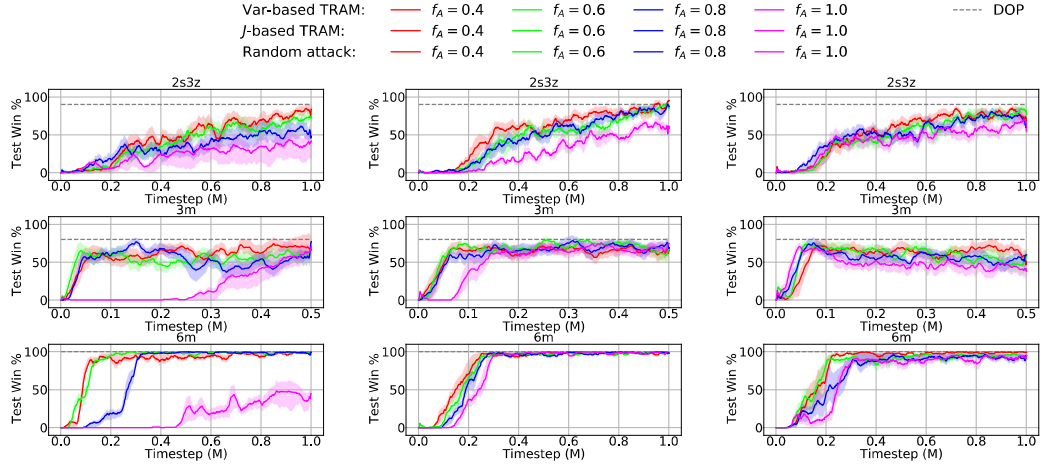


Figure 11: Include confidence intervals of Fig. 5.

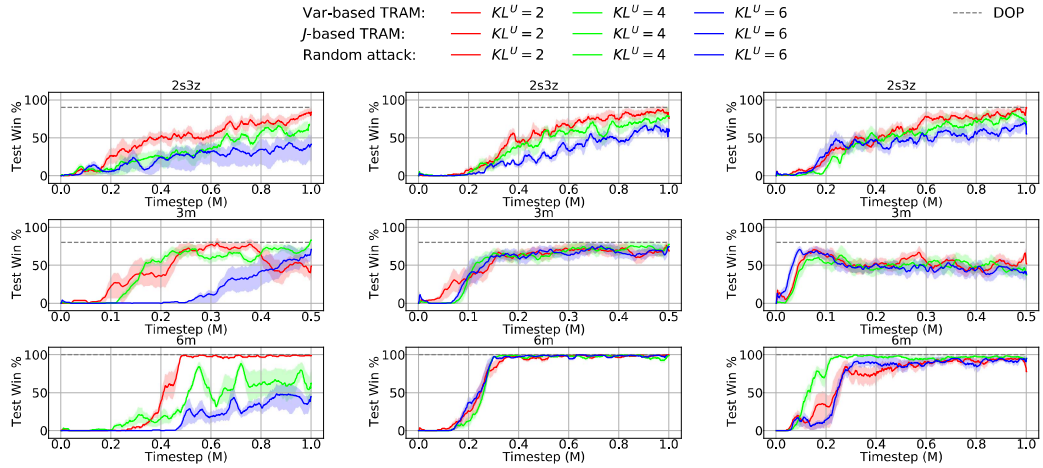


Figure 12: Include confidence intervals of Fig. 6..