APPENDICES OF SELF-ENSEMBLE ADVERSARIAL TRAINING FOR BETTER ROBUSTNESS

Anonymous authors
Paper under double-blind review

A FURTHER EXPERIMENTS

Here we adopt ResNet18 and / or WRN-32-10 as the backbone model with the same experimental setup as in Sec. 4.1, where we reported the natural accuracy (NAT), PGD-20 and PGD-100 attack (PGD), MIM (PGD with a momentum term), CW attack and each component of AutoAttack. All the experiments are conducted for 5 individual trials and we also report their standard deviations. All the methods were realized by Pytorch 1.5, where we used a single NVIDIA GeForce RTX 3090 GPU.

A.1 ROBUSTNESS AGAINST COMPONENTS OF AUTOATTACK

To broadly demonstrate the robustness of our proposal, we conducted experiments against each component of AutoAttack. We perform each component of AA on CIFAR-10 dataset with both ResNet18 and WRN-32-10, including three parameter-free versions of PGD with the CE, DLR, targeted-CE loss with 9 target classes loss (APGD_{CE}, APGD_{DLR}, APGD_{T}), the targeted version of FAB (FAB_{T}) and an existing complementary Square (**?**). Results are shown in the following Table 1. And it is obvious that our SEAT outperforms other methods against all components of AA.

Table 1: Average robust accuracy (%) and standard deviation against each component of AA on CIFAR-10 dataset with ResNet18 and WRN-32-10.

	ResNet18					WRN-32-10				
	APGD_{CE}	APGD_{DLR}	APGD_{T}	FAB_{T}	Square	APGD_{CE}	APGD_{DLR}	APGD_{T}	FAB_{T}	Square
AT	47.47 ±0.35	48.57 ±0.18	45.14 ±0.31	46.17 ±0.11	54.21 ±0.15	49.17 ±0.26	50.09 ±0.36	47.34 ±0.33	48.00 ±0.43	56.5 ±0.18
TRADES	53.47 ±0.21	50.89 ±0.26	47.93 ±0.36	48.53 ±0.43	55.75 ±0.21	55.38 ±0.43	55.55 ±0.42	52.2 ±0.13	53.11 ±0.72	59.47 ±0.17
MART	52.98 ±0.13	50.36 ±0.3	48.17 ±0.72	49.39 ±0.28	55.73 ±0.51	55.2 ±0.32	55.41 ±0.4	51.99 ±0.3	52.88 ±0.63	59.01 ±0.38
SEAT	**53.87 ±0.17**	**53.35 ±0.24**	**50.88 ±0.27**	**51.41 ±0.37**	**57.77 ±0.22**	**57.57 ±0.18**	**57.74 ±0.29**	**55.06 ±0.27**	**55.53 ±0.36**	**62.26 ±0.23**

A.2 PERFORMANCE ON CIFAR-100

To further demonstrate the robustness of our proposal against adversarial attacks, we benchmark the state-of-the-art robustness with ResNet18 on CIFAR-100 (**?**). We widely investigate the performance of SEAT against the PGD methods (PGD^{20} and PGD^{100}), MIM, CW, AA and its all components. Results shown in Table 2 demonstrate the effectiveness of SEAT for building a robust classifier.

Table 2: Comparison of our algorithm with different defense methods using ResNet18 on CIFAR10. The maximum perturbation is $\varepsilon = 8/255$. Average accuracy rates (in %) and standard deviations have shown that the proposed SEAT method greatly improves the robustness of the model.

Method	NAT	PGD^{20}	PGD^{100}	MIM	CW	APGD_{CE}	APGD_{DLR}	APGD_{T}	FAB_{T}	Square	AA
AT	**60.1 ±0.35**	28.22 ±0.3	28.27 ±0.12	28.31 ±0.41	24.87 ±0.51	26.63 ±0.29	24.13 ±0.22	21.98 ±0.3	23.87 ±0.21	27.93 ±0.12	23.91 ±0.41
TRADES	59.93 ±0.46	29.9 ±0.41	29.88 ±0.11	29.55 ±0.25	26.14 ±0.21	27.93 ±0.44	25.43 ±0.29	23.72 ±0.45	25.16 ±0.15	30.03 ±0.32	24.72 ±0.37
MART	57.24 ±0.64	30.62 ±0.37	30.62 ±0.17	30.83 ±0.28	26.3 ±0.29	29.91 ±0.07	26.32 ±0.24	24.28 ±0.49	24.86 ±0.66	28.28 ±0.39	24.27 ±0.21
SEAT	56.28 ±0.33	**32.15 ±0.17**	**32.12 ±0.26**	**32.62 ±0.15**	**29.68 ±0.26**	**30.97 ±0.18**	**29.62 ±0.22**	**26.88 ±0.23**	**27.71 ±0.24**	**32.35 ±0.34**	**27.87 ±0.24**

A.3 DIFFERENT LEARNING RATE STRATEGIES

Apart from showing the curve of different learning rate schedule in Figure 3 (a) in Sec. 4.3, we also report final results in Table 3. The effect of warming up learning rate is marginal. When compared with the staircase one, the warmup strategy cannot generate diverse models in the later stages so the homogenization of the candidate models we mentioned in Sec. 3.2 cannot be fixed by the warmup strategy. On the contrary, those methods like cosine / linear / cyclic that provide relatively diverse models in the later stages can mitigate the issue, accounting for more robust ensemble models.

Table 3: Average robust accuracy (%) under different learning strategies on CIFAR-10 dataset with ResNet18.

Method	NAT	PGD^{20}	PGD^{100}	MIM	CW	APGD_{CE}	APGD_{DLR}	APGD_{T}	FAB_{T}	Square	AA
SEAT (Staircase)	80.91	54.58	54.56	54.47	49.71	52.39	48.01	45.83	45.11	53.64	45.85
SEAT (Cosine)	83.0	55.09	55.16	56.39	53.43	52.34	52.1	49.51	50.15	56.48	50.48
SEAT (Linear)	**83.7**	**56.02**	**55.97**	**57.13**	**54.38**	**53.87**	**53.35**	**50.88**	**51.41**	**57.77**	**51.3**
SEAT (Warmup)	82.74	55.31	55.35	56.39	53.26	53.55	48.94	45.89	46.6	54.94	45.82
SEAT (Cyclic)	83.14	**56.03**	**55.79**	**56.99**	**54.01**	**53.72**	**53.1**	**50.66**	**51.02**	**57.75**	**51.44**

A.4 DETERIORATION OF VANILLA EMA

As shown in Sec. 3.3, the deteriorated SEAT underperforms SEAT a lot from the perspective of optimization. We also report quantitative results on both ResNet18 and WRN-32-10 shown in the Tables 4 and 5. The deteriorated one does not bring too much boost when compared to the vanilla adversarial training (except for PGD methods). A plausible explanation for the exception of PGD is that the SEAT technique produces an ensemble of individuals that are adversarially trained by PGD with the cross-entropy loss, which means that they are intrinsically good at defending the PGD attack with the cross-entropy loss and its variants even though they suffer from the deterioration. Considering results have greatly improved after using the piecewise linear learning rate strategy, it is fair to say that adjusting learning rate is effective. As we claimed in Proposition 2 and its proof, the staircase will inevitably make the self-ensemble model worsen since $\sum_{t=1}^{T}(\beta_t \tilde{\boldsymbol{\xi}}^{\mathrm{T}})$ will gradually approach to zero, meaning the difference between $\bar{f}_{\mathcal{F}}(x, y)$ and $f_{\bar{\boldsymbol{\theta}}}(x, y)$ achieves the second order of smallness.

Table 4: Average robust accuracy (%) and standard deviation on CIFAR-10 dataset with ResNet18.

Method	NAT	PGD^{20}	PGD^{100}	MIM	CW	APGD_{CE}	APGD_{DLR}	APGD_{T}	FAB_{T}	Square	AA
AT	84.32	48.29	48.12	47.95	49.57	47.47	48.57	45.14	46.17	54.21	44.37
	±0.23	±0.11	±0.13	±0.04	±0.15	±0.35	±0.18	±0.31	±0.11	±0.25	±0.37
SEAT (deteriorated)	80.91	54.58	54.56	54.47	49.71	52.39	48.01	45.83	45.11	53.64	45.85
	±0.38	±0.71	±0.29	±0.39	±0.41	±0.26	±0.18	±0.52	±0.23	±0.44	±0.19
SEAT	83.7	56.02	55.97	57.13	54.38	53.87	53.35	50.88	51.41	57.77	51.3
	±0.13	±0.11	±0.07	±0.12	±0.1	±0.17	±0.24	±0.27	±0.37	±0.22	±0.26

Table 5: Average robust accuracy (%) and standard deviation on CIFAR-10 dataset with WRN-32-10.

Method	NAT	PGD^{20}	PGD^{100}	MIM	CW	APGD_{CE}	APGD_{DLR}	APGD_{T}	FAB_{T}	Square	AA
AT	**87.32**	49.01	48.83	48.25	52.8	54.17	53.09	48.34	49.00	57.5	48.17
	±0.21	±0.33	±0.27	±0.17	±0.25	±0.26	±0.36	±0.33	±0.43	±0.18	±0.48
SEAT (deteriorated)	85.28	55.68	55.57	55.6	53.01	54.12	53.54	49.95	50.02	57.81	49.96
	±0.42	±0.42	±0.19	±0.23	±0.41	±0.54	±0.28	±0.67	±0.75	±0.33	±0.31
SEAT	86.44	**59.84**	**59.8**	**60.87**	**58.95**	**57.57**	**57.74**	**55.06**	**55.53**	**62.26**	**55.67**
	±0.12	**±0.2**	**±0.16**	**±0.1**	**±0.34**	**±0.18**	**±0.29**	**±0.27**	**±0.36**	**±0.23**	**±0.22**

A.5 COMPUTATIONAL COMPLEXITY FOR SEAT

To demonstrate the efficiency of the SEAT method, we use the number of Multiply-Accumulate operations (MACs) in Giga (G) to compute the theoretical amount of multiply-add operations in DNNs, roughly GMACs = 0.5 * GFLOPs. Besides, we also provide the actual running time. As shown in Table 6, the SEAT method takes negligible MACs and training time when compared with standard adversarial training.

Table 6: Evaluation of time complexity of SEAT. Here we use the number of Multiply-Accumulate operations (MACs) in Giga (G) to measure the running time complexity. And we also compute the actual training time with or without the SEAT method using ResNet18 and WRN-32-10 on a single NVIDIA GeForce RTX 3090 GPU.

Method	MACs (G)	Training Time (mins)
ResNet18 (AT)	0.56	272
ResNet18 (SEAT)	0.59	273
WRN-32-10 (AT)	6.67	1534
WRN-32-10 (SEAT)	6.81	1544

B PROOFS OF THEORETICAL RESULTS

B.1 PROOF OF PROPOSITION 1

Proposition 1. *(Restated) Let $f_{\boldsymbol{\theta}}(\cdot)$ denote the predictions of a neural network parametrized by weights $\boldsymbol{\theta}$. Assuming that $\forall \boldsymbol{\theta} \in \Theta$, $f_{\boldsymbol{\theta}}(\cdot)$ is continuous and $\forall (x, y) \in \mathbb{D}$, $f_{\boldsymbol{\theta}}(x, y)$ is at least twice differentiable. Consider two points $\boldsymbol{\theta}_t, \tilde{\boldsymbol{\theta}} \in \Theta$ in the weight space and let $\boldsymbol{\xi} = \boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}$, for $t \in \{1, 2, \cdots, T\}$, the difference between $\bar{f}_{\mathcal{F}}(x, y)$ and $f_{\tilde{\boldsymbol{\theta}}}(x, y)$ is of the second order of smallness if and only if $\sum_{t=1}^{T}(\beta_t \boldsymbol{\xi}^{\mathrm{T}}) = \mathbf{0}$.*

Proof. For the sake of the twice differentiability of $f_{\boldsymbol{\theta}}(x, y)$, based on the Taylor expansion, we can fit a quadratic polynomial of $f_{\tilde{\boldsymbol{\theta}}}(x, y)$ to approximate the value of $f_{\boldsymbol{\theta}_t}(x, y)$:

$$f_{\boldsymbol{\theta}_t}(x, y) = f_{\tilde{\boldsymbol{\theta}}}(x, y) + \boldsymbol{\xi}^{\mathrm{T}} \nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x, y) + \frac{1}{2} \boldsymbol{\xi}^{\top} \nabla_{\boldsymbol{\xi}}^2 f_{\tilde{\boldsymbol{\theta}}}(x, y) \boldsymbol{\xi} + O\left(\Delta^n\right), \tag{1}$$

where $O\left(\Delta^n\right)$ represents the higher-order remainder term. Note that the subscript $\boldsymbol{\xi}$ here stands for a neighborhood where the Taylor expansion approximates a function by polynomials of any point (i.e. $\tilde{\boldsymbol{\theta}}$) in terms of its value and derivatives. So the difference between the averaged prediction of candidate classifiers and the prediction of the ensembled weight classifier can be formulated as:

$$\begin{aligned} \bar{f}_{\mathcal{F}}(x, y) - f_{\tilde{\boldsymbol{\theta}}}(x, y) &= \sum_{t=1}^{T} \beta_t f_{\boldsymbol{\theta}_t}(x, y) - f_{\tilde{\boldsymbol{\theta}}}(x, y) \\ &= \cancel{\sum_{t=1}^{T} \beta_t f_{\tilde{\boldsymbol{\theta}}}(x, y)} + \sum_{t=1}^{T} \beta_t \boldsymbol{\xi}^{\mathrm{T}} \nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x, y) + \sum_{t=1}^{T} \beta_t O\left(\Delta^2\right) \cancel{- f_{\tilde{\boldsymbol{\theta}}}(x, y)} \\ &= \sum_{t=1}^{T} (\beta_t \boldsymbol{\xi}^{\mathrm{T}}) \nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x, y) + O\left(\Delta^2\right). \end{aligned} \tag{2}$$

Therefore, we can claim that the difference between $f_{\boldsymbol{\theta}_t}(x, y)$ and $f_{\tilde{\boldsymbol{\theta}}}(x, y)$ is "almost" at least of the first order of smallness except for some special cases. And we will immediately declare under which condition this difference can achieve the second order of smallness in the following proof of Theorem 1. □

B.2 PROOF OF THEOREM 1

Theorem 1. *(Restated) Assuming that for $i, j \in \{1, \cdots, T\}$, $\boldsymbol{\theta}_i = \boldsymbol{\theta}_j$ if and only if $i = j$. The difference between the averaged prediction of multiple networks and the prediction of SEAT is of the second order of smallness if and only if $\beta_i = (1 - \alpha)\alpha^{i-1}$ for $i \in \{1, 2, \cdots, T\}$.*

Proof. According to Eqn 2, we know that the second order of smallness will achieve when $\sum_{t=1}^{T}(\beta_t \boldsymbol{\xi}^{\mathrm{T}}) = \mathbf{0}$. Thus, we continue deducing from Eqn 2 as:

$$\begin{aligned} \sum_{t=1}^{T}(\beta_t \boldsymbol{\xi}^{\mathrm{T}}) &= \mathbf{0} \\ \sum_{t=1}^{T}\beta_t(\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}) &= \mathbf{0} \\ \sum_{t=1}^{T}\beta_t \boldsymbol{\theta}_t &= \tilde{\boldsymbol{\theta}} \\ \sum_{t=1}^{T}\beta_t \boldsymbol{\theta}_t &= \sum_{t=1}^{T}(1-\alpha)\alpha^{t-1}\boldsymbol{\theta}_t. \end{aligned} \tag{3}$$

To get a further aonclusion, we next use Mathematical Induction (MI) to prove only when $\beta_i = (1-\alpha)\alpha^{i-1}$ for $i \in \{1, 2, \cdots, T\}$ will $\sum_{t=1}^{T}\beta_t \boldsymbol{\theta}_t = \sum_{t=1}^{T}(1-\alpha)\alpha^{t-1}\boldsymbol{\theta}_t$ set up.

Base case: Let $t = 1$, it is clearly true that $\beta_1 = (1-\alpha)\alpha^0$ if and only if $\beta_1\boldsymbol{\theta}_1 = (1-\alpha)\alpha^0\boldsymbol{\theta}_1$, hence the base case holds.

Inductive step: Assume the induction hypothesis that for a particular k, the single case $T = k$ holds, meaning the sequence of $(\beta_1, \beta_2, \cdots, \beta_k)$ is equal to the sequence of $((1-\alpha)\alpha^0, (1-\alpha)\alpha^1, \cdots, (1-\alpha)\alpha^{k-1})$ if $\sum_{t=1}^{k}\beta_t\boldsymbol{\theta}_t = \sum_{t=1}^{k}(1-\alpha)\alpha^{t-1}\boldsymbol{\theta}_t$.

For $T = k+1$, it follows that:

$$\begin{aligned} \sum_{t=1}^{k+1}\beta_t\boldsymbol{\theta}_t &= \sum_{t=1}^{k+1}(1-\alpha)\alpha^{t-1}\boldsymbol{\theta}_t \\ \cancel{\sum_{t=1}^{k}\beta_t\boldsymbol{\theta}_t} + \beta_{k+1}\boldsymbol{\theta}_{k+1} &= \cancel{\sum_{t=1}^{k}(1-\alpha)\alpha^{t-1}\boldsymbol{\theta}_t} + (1-\alpha)\alpha^{k+1-1}\boldsymbol{\theta}_{k+1} \\ \beta_{k+1}\boldsymbol{\theta}_{k+1} &= (1-\alpha)\alpha^{k+1-1}\boldsymbol{\theta}_{k+1} \\ \beta_{k+1} &= (1-\alpha)\alpha^{k+1-1}. \end{aligned} \tag{4}$$

The sequence of normalized scores at the $k+1$-th ensembling at left hand is $(\beta_1, \beta_2, \cdots, \beta_k, \beta_{k+1})$ after adding the new term β_{k+1}. Likewise, the sequence of the right hand is $((1-\alpha)\alpha^0, (1-\alpha)\alpha^1, \cdots, (1-\alpha)\alpha^{k-1}, (1-\alpha)\alpha^k)$. Because every $f_{\theta_t} \in \mathcal{F}$ is different from others and the sequence is ordered, we have $(\beta_1, \beta_2, \cdots, \beta_k, \beta_{k+1}) = ((1-\alpha)\alpha^0, (1-\alpha)\alpha^1, \cdots, (1-\alpha)\alpha^{k-1}, (1-\alpha)\alpha^k)$

Conclusion: Since both the base case and the inductive step have been proved as true, by mathematical induction the statement $\beta_i = (1-\alpha)\alpha^{i-1}$ for $i \in \{1, 2, \cdots, T\}$ holds for every positive integer T. Back to the starting point, the first order term cannot be ignored in most cases. SEAT is hardly be approximate to the averaged prediction of history networks indeed. □

B.3 Proof of Proposition 2

Proposition 2. *(Restated) Assuming that every candidate classifier is updated by SGD-like strategy, meaning $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \tau_t \nabla_{\boldsymbol{\theta}_t} f_{\boldsymbol{\theta}_t}(x', y)$ with $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_T > 0$, the self-ensemble model will inevitably worsen and the degree depends on learning rate schedules.*

Proof. First we discuss a special case - the change at the t-th iteration. Reconsidering the first order term in Eqn 2, we have:

$$\begin{aligned}
&\sum_{t=1}^{T}(\beta_t \boldsymbol{\xi}^{\mathrm{T}})\nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x,y) \\
&= \sum_{t=1}^{T}[\beta_t(\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}})]\nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x,y) \\
&= \sum_{t=1}^{T}[\beta_t((1-(1-\alpha)\alpha^{t-1})\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_{\mathcal{F}\setminus t})]\nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x,y) \\
&= \sum_{t=1}^{T}[\beta_t((1-(1-\alpha)\alpha^{t-1})\boldsymbol{\theta}_t - (1-(1-\alpha)\alpha^{t-1}/\alpha)\boldsymbol{\theta}_{t-1} - \tilde{\boldsymbol{\theta}}_{\mathcal{F}\setminus t,t-1})]\nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x,y)
\end{aligned} \tag{5}$$

Owing to the fact that the decay rate α close to 1 is typically recommended in practice, so $1/\alpha \to 1$. Then we can deduce by combining:

$$\sum_{t=1}^{T}[\beta_t((1-(1-\alpha)\alpha^{t-1})(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}) - \tilde{\boldsymbol{\theta}}_{\mathcal{F}\setminus t,t-1})]\nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x,y) \tag{6}$$

By using SGD to update θ_{t-1}, we have:

$$\sum_{t=1}^{T}[\beta_t((1-(1-\alpha)\alpha^{t-1})(\tau_t \mathbb{E}_{(x,y)}\left[\nabla_{\boldsymbol{\theta}_t}\ell(\boldsymbol{\theta}_t;(x'_k,y))\right]) - \tilde{\boldsymbol{\theta}}_{\mathcal{F}\setminus t,t-1})]\nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x,y) \tag{7}$$

Without changing samples in the t-th minibatch, we can conclude that the output of SEAT will be closer to the averaged prediction of multiple networks when the learning rate τ_t becomes extremely small at the t-th iteration.

To further analyse the whole training process, we construct $\boldsymbol{\theta} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\theta}_t$ and $\tilde{\boldsymbol{\xi}} = \boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}$ to unpack $\tilde{\boldsymbol{\theta}}_{\mathcal{F}\setminus t}$, and then reformulate Eqn 2:

$$\begin{aligned}
&\sum_{t=1}^{T}(\beta_t \tilde{\boldsymbol{\xi}}^{\mathrm{T}})\nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x,y) \\
&= \sum_{t=1}^{T}[\beta_t(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})]\nabla_{\tilde{\boldsymbol{\xi}}} f_{\tilde{\boldsymbol{\theta}}}(x,y) \\
&= \sum_{t=1}^{T}[\frac{\beta_t}{T}((1-(1-\alpha)\alpha^{t-1})\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_{\mathcal{F}\setminus t})]\nabla_{\tilde{\boldsymbol{\xi}}} f_{\tilde{\boldsymbol{\theta}}}(x,y) \\
&= \sum_{t=1}^{T}[\frac{\beta_t}{T}((1-(1-\alpha)\alpha^{t-1})(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}))]\nabla_{\tilde{\boldsymbol{\xi}}} f_{\tilde{\boldsymbol{\theta}}}(x,y) \\
&= \sum_{t=1}^{T}[\frac{\beta_t}{T}((1-(1-\alpha)\alpha^{t-1})(\tau_t \mathbb{E}_{(x,y)}\left[\nabla_{\boldsymbol{\theta}_t}\ell(\boldsymbol{\theta}_t;(x'_k,y))\right]))]\nabla_{\boldsymbol{\xi}} f_{\tilde{\boldsymbol{\theta}}}(x,y)
\end{aligned} \tag{8}$$

When the learning rate τ_t achieves the threshold where the updating amount becomes very small (as shown in Figure 1 (a) in the main body) at a certain iteration t', the learning rate schedules $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_T$ will be divided into $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_{t'}$ and $\tau_{t'+1} \geq \tau_{t'+2} \geq \cdots \geq \tau_T$. Because $(\beta_1, \beta_2, \cdots, \beta_T)$ is a non-decreasing sequence, when $T - t' \to \infty$, $\sum_{t=1}^{T}(\beta_t \tilde{\boldsymbol{\xi}}^{\mathrm{T}})$ will gradually approach to zero, which means the difference between $\bar{f}_{\mathcal{F}}(x,y)$ and $f_{\tilde{\boldsymbol{\theta}}}(x,y)$ achieves the second order of smallness. □

REFERENCES