Appendix of Generalist: Decoupling Natural and Robust Generalization

Hongjun Wang[1*] Yisen Wang[1,2†]
[1] National Key Lab of General Artificial Intelligence
School of Intelligence Science and Technology, Peking University
[2] Institute for Artificial Intelligence, Peking University

A. Additional Experiments

A.1. Detailed Configurations

All images are normalized into $[0, 1]$. We train ResNet-18 using SGD with 0.9 momentum for 120 epochs (200 epochs for CIFAR-100) and the weight decay factor is set to $3.5e^{-3}$ for ResNet-18 and $7e^{-4}$ for WRN-32-10. We use the piecewise linear learning rate strategy for performing weight averaging in base-learners. For the base-learner of AT, the initial learning rate for ResNet-18 is set to 0.01 and 0.1 for WRN-32-10 till Epoch 40 and then linearly reduced by 10 at Epoch 60 and 120, respectively. The magnitude of maximum perturbation at each pixel is $\varepsilon = 8/255$ with step size $\kappa = 2/255$ and the PGD steps number in the inner maximization is 10. For the base-learner of NT, we fix the initial learning rate as 0.1 and the weight decay is $5e^{-4}$ for both ResNet-18 and WRN-32-10.

A.2. Experiments on MNIST/SVHN

We conducted experiments on MNIST ($\varepsilon = 0.3$) and SVHN using ResNet-18 with the same setup in Sec. A.1. We ran 5 individual trials and results with standard deviations are shown in Table 2. Our Generalist still achieves the best performance.

Table 2. Comparison of our algorithm with different training methods using ResNet-18 on MNIST and SVHN. The maximum perturbation is $\varepsilon = 8/255$. The best checkpoint is selected based on the tradeoff between clean accuracy and robust accuracy against PGD20 on the test set. We highlight the top two results on each task. Average accuracy rates (in %) have shown that the proposed Generalist method greatly mitigates the tradeoff of the model.

Methods	MNIST			SVHN		
	NAT	PGD20	AA	NAT	PGD20	AA
TRADES	99.07 ±0.13	94.45 ±0.07	92.17 ±0.21	93.1 ±0.25	**55.38 ±0.71**	**45.52 ±0.37**
FAT	99.18 ±0.03	93.54 ±0.1	90.04 ±0.68	93.87 ±0.4	53.61 ±0.88	40.92 ±0.29
Generalist	**99.24 ±0.07**	**96.14 ±0.15**	**92.3 ±0.3**	**94.11 ±0.27**	**55.29 ±0.23**	**45.41 ±0.26**

A.3. Experiments on CIFAR-100

To further demonstrate our proposed Generalist achieves a better tradeoff between accuracy and robustness, we also conduct experiments on CIFAR-100 datasets. Here we still use ResNet-18 as the backbone model with the same configurations as claimed in Sec. A.1. We report the results of natural accuracy and several advanced adversarial attack methods in Table 3. Note that we do not design a specialized strategy for Generalist on CIFAR-100 but Generalist still achieves a gratifying tradeoff, so it still has the potential to perform better.

*Work was done as an internship at Peking University. Now, he is a Ph.D. student at the University of Hong Kong.
†Corresponding Author: Yisen Wang (yisen.wang@pku.edu.cn)

Table 3. Comparison of our algorithm with different training methods using ResNet-18 on CIFAR-100. The maximum perturbation is $\varepsilon = 8/255$. The best checkpoint is selected based on the tradeoff between clean accuracy and robust accuracy against PGD20 on the test set. We highlight the top two results on each task. Average accuracy rates (in %) have shown that the proposed Generalist method greatly mitigates the tradeoff of the model.

Method	NAT	PGD20	PGD100	MIM	CW	APGD_{ce}	APGD_{dlr}	APGD_{t}	FAT_{t}	Square	AA
NT	**65.74**	0.02	0.01	0.02	0.01	0.00	0.00	0.00	0.07	0.37	0.00
AT ($\beta = 1$)	60.10	28.22	28.27	28.31	24.87	26.63	24.13	21.98	23.91	27.93	23.87
AT ($\beta = 1/2$)	60.84	22.64	22.61	23.86	22.28	20.66	21.67	19.2	20.09	25.36	19.17
TRADES ($\lambda = 6$)	59.93	**29.90**	**29.88**	**29.55**	**26.14**	**27.93**	**25.43**	**24.72**	**25.16**	**30.03**	**23.72**
TRADES ($\lambda = 1$)	60.18	28.93	28.91	29.12	25.79	27.07	25.00	23.65	24.31	28.76	23.22
FAT	61.71	22.93	22.87	22.64	23.45	24.78	24.91	20.56	23.16	26.37	20.01
IAT	57.04	21.40	21.39	22.37	19.18	19.63	18.92	15.50	16.63	23.26	15.50
RST	60.30	23.56	23.61	23.71	22.40	24.69	24.18	21.66	23.82	27.05	21.18
Generalist	**62.97**	**29.48**	**29.49**	**30.35**	**27.77**	**27.45**	**27.42**	**24.04**	**25.54**	**31.41**	**23.96**

A.4. Computational Cost and Tradeoff Comparison of Generalist

We compute the actual training time of TRADES and Generalist (serial/parallel version) using ResNet-18 on RTX 3090 GPU in Table 4. We also report the standard deviations over 5 runs to show the sensitivity of Generalist. Neither version of Generalist is slower than TRADES. Generalist does perform both NT and naive AT, but the cost of NT is negligible so the overhead (NT+AT) is smaller than TRADES.

Besides, Table 4 delivers another important message. For the tradeoff between robustness and accuracy, it is hard to obtain acceptable robustness while maintaining clean accuracy above 89% in the joint training framework (TRADES). For every percentage point increase in clean accuracy, the robust accuracy will decrease dramatically (e.g. TRADES can meet 89% on clean accuracy but its robustness against APGD will drop to 30%).

Table 4. Evaluation of time complexity of our algorithm with different training methods using ResNet-18.

Method	NAT	PGD100	APGD	Training Time (mins)
TRADES	89.91 $\pm$0.69	34.25 $\pm$0.56	30.20 $\pm$0.81	414
Generalist (Serial)	89.11 $\pm$0.23	50.12 $\pm$0.12	46.12 $\pm$0.11	397
Generalist (Parallel)	89.09 $\pm$0.34	50.00 $\pm$0.44	46.53 $\pm$0.3	**342**

A.5. Influence of Learning Rate

In this part, we also study the influence of the learning rate for different distribution-aware tasks. For simplicity, we set t', γ and c as their best options according to the main body of the paper. We search the most grid of learning rate configurations in the range of 0.1, 0.01, 0.001 for both natural training and adversarial training. Our Generalist achieves its best and second-best natural accuracy when the learning rate for the clean learner is set to 0.1. And the optimal learning rate for robust accuracy is 0.01. Based on all the observations from Table 5, the learning pace of learners is a little different but the process is compatible.

Table 5. Clean and robust accuracy (%) on CIFAR-10 dataset using ResNet-18 with different learning rates.

	NAT	AA
NT=0.1, AT=0.01	89.09	46.37
NT=0.1, AT=0.1	90.12	41.86
NT=0.1, AT=0.001	**90.45**	43.55
NT=0.01, AT=0.01	88.4	**48.03**
NT=0.01, AT=0.1	88.25	42.98

B. Proofs of Theoretical Results

B.1. Proof of Claim in Section 3.3

Proof. At epoch t, the parameters of the global learner are distributed to the experts and each expert train from this initialization with c steps by calculating the gradients (e.g. using SGD optimizer). Following [3], we approximate the update performed by the initialization based on the Taylor expansion:

$$\begin{aligned} g^{t+c} = \ell'\left(\boldsymbol{\theta}^{t+c}\right) &= \ell'\left(\boldsymbol{\theta}^{t}\right) + \ell''\left(\boldsymbol{\theta}^{t}\right)\left(\boldsymbol{\theta}^{t+c} - \boldsymbol{\theta}^{t}\right) + O\left(\left\|\boldsymbol{\theta}^{t+c} - \boldsymbol{\theta}^{t}\right\|^2\right) \\ &= \bar{g}^{t} + \bar{H}^{t}\left(\boldsymbol{\theta}^{t+c} - \boldsymbol{\theta}^{t}\right) + O\left(\tau^2\right)) \\ &= \bar{g}^{t} - \tau\bar{H}^{t}\sum_{j=t}^{t+c} g^{j} + O\left(\tau^2\right) \\ &= \bar{g}^{t} - \tau\bar{H}^{t}\sum_{j=t}^{t+c} \bar{g}^{j} + O\left(\tau^2\right). \end{aligned} \tag{1}$$

Recalling that $\mathcal{Z}^i$ represents an optimizer that updates the parameter vector at the t-th step: $\mathcal{Z}^i(\boldsymbol{\theta}, \tau) = \boldsymbol{\theta} - \tau\ell'(\boldsymbol{\theta})$. For each base-learner, we approximate the gradient at intervals:

$$\begin{aligned} g_{val} = \frac{\partial}{\partial \boldsymbol{\theta}^{t}}\ell\left(\boldsymbol{\theta}^{t+c}\right) &= \frac{\partial}{\partial \boldsymbol{\theta}^{t}}\ell\left(\mathcal{Z}^{t+c-1}\left(\mathcal{Z}^{t+c-2}\left(\ldots\left(\mathcal{Z}^{t}\left(\boldsymbol{\theta}^{t}\right)\right)\right)\right)\right) \\ &= \mathcal{Z}'^{t}\left(\boldsymbol{\theta}^{t}\right)\cdots\mathcal{Z}'^{t+c-1}\left(\boldsymbol{\theta}^{t+c-1}\right)\ell'\left(\boldsymbol{\theta}^{t+c}\right) \\ &= \left(I - \tau\ell''\left(\boldsymbol{\theta}^{t}\right)\right)\cdots\left(I - \tau\ell''\left(\boldsymbol{\theta}^{t+c-1}\right)\right)\ell'\left(\boldsymbol{\theta}^{t+c}\right) \\ &= \left(\prod_{j=t}^{t+c-1}\left(I - \tau\ell''\left(\boldsymbol{\theta}^{j}\right)\right)\right) g^{t+c}. \end{aligned} \tag{2}$$

Replacing $\ell''\left(\boldsymbol{\theta}^{j}\right)$ with $\bar{H}^{j}$ and substituting g^{t+c} for Eq. 1, we expand to leading order:

$$\begin{aligned} g_{val} &= \left(\prod_{j=t}^{t+c-1}\left(I - \tau\bar{H}^{j}\right)\right)\left(\bar{g}^{t+c} - \tau\bar{H}^{t+c}\sum_{j=t}^{t+c-1}\bar{g}^{j}\right) + O\left(\tau^2\right) \\ &= \left(I - \tau\sum_{j=t}^{t+c-1}\bar{H}^{j}\right)\left(\bar{g}^{t+c} - \tau\bar{H}^{t+c}\sum_{j=t}^{t+c-1}\bar{g}^{j}\right) + O\left(\tau^2\right) \\ &= \bar{g}^{t+c} - \tau\sum_{j=t}^{t+c-1}\bar{H}^{j}\bar{g}^{t+c} - \tau\bar{H}^{t+c}\sum_{j=t}^{t+c-1}\bar{g}^{j} + O\left(\tau^2\right) \end{aligned} \tag{3}$$

Therefore, we take the expectation of g_{val} over steps, and obtain:

$$\mathbb{E}\left[g_{val}\right] = \mathbb{E}\left[\bar{g}^{t+c}\right] - \tau\mathbb{E}\left[\sum_{j=t}^{t+c-1}\bar{H}^{j}\bar{g}^{t+c} - \bar{H}^{t+c}\sum_{j=t}^{t+c-1}\bar{g}^{j}\right] + \mathbb{E}\left[O\left(\tau^2\right)\right] \tag{4}$$

Recalling that $\boldsymbol{\theta}_g$ is mixed by $\boldsymbol{\theta}_n$ and $\boldsymbol{\theta}_r$. For simplicity of exposition, we use p and q to stand for the scalar factors, meaning $\boldsymbol{\theta}_g = p\boldsymbol{\theta}_n + q\boldsymbol{\theta}_r$. Ignoring the higher order terms, for each expert initialized by the global learner (e.g. $\boldsymbol{\theta}_n$), we have:

$$\begin{aligned}\boldsymbol{\theta}_n = \boldsymbol{\theta}_g - \mathbb{E}_n\left[g_{val}\right] &= p\boldsymbol{\theta}_n + q\boldsymbol{\theta}_r - [\mathbb{E}\left[\bar{g}_n^{t+c}\right] + \tau_n \mathbb{E}\left[\sum_{j=t}^{t+c-1} \bar{H}^j \bar{g}_n^{t+c} - \bar{H}^{t+c} \sum_{j=t}^{t+c-1} \bar{g}_n^j\right]] \\ &= [p\boldsymbol{\theta}_n - \mathbb{E}\left[\bar{g}_n^{t+c}\right]] + [q\boldsymbol{\theta}_r - \tau_n \mathbb{E}\left[\bar{H}^{t+c} \sum_{j=t}^{t+c-1} \bar{g}_n^j - \sum_{j=t}^{t+c-1} \bar{H}^j \bar{g}_n^{t+c}\right]] \\ &= [p\boldsymbol{\theta}_n - \sum_{i=t}^{t+c-1} \bar{g}^i] + [q\boldsymbol{\theta}_r - \tau_n \sum_{i=t}^{t+c-1} \sum_{j=1}^{i-1} \bar{H}^i \bar{g}^j] \qquad (\text{for } c \geq 2).\end{aligned} \tag{5}$$

The first term pushes θ_n to move forward the minimum of its assigned loss over its data distribution; while the second one improves generalization by increasing the inner product between gradients of different mini-batches and updating the parameters from the other task. □

B.2. Proof of Theorem 1

Before we present the proof of the Theorem we present useful intermediate results which we require in our proof.

Proposition 1. *Consider a sequence of loss functions $\ell_a : \Theta \mapsto [0,1]_{a\in\mathcal{A}}$ drawn i.i.d. from some distribution $\mathcal{L}$ is given to an algorithm that generates a sequence of hypotheses $\{\boldsymbol{\theta}_a \in \Theta\}_{a\in\mathcal{A}}$ then the following inequality each hold w.p. $1-\delta$:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\ell\sim D} \ell\left(\boldsymbol{\theta}^t\right) \leq \frac{1}{T}\sum_{t=1}^{T} \ell^t\left(\boldsymbol{\theta}^t\right) + \sqrt{\frac{2}{T}\log\frac{1}{\delta}}. \tag{6}$$

Proof. The proof of the Proposition can be directly derived from the Proposition 1 in [2]. □

Then we could immediately obtain the below inequality by the symmetric version of the Azuma-Hoeffding inequality [1]

Remark 1.

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\ell\sim \mathcal{L}} \ell\left(\boldsymbol{\theta}^t\right) \geq \frac{1}{T}\sum_{t=1}^{T} \ell^t\left(\boldsymbol{\theta}^t\right) - \sqrt{\frac{2}{T}\log\frac{1}{\delta}}. \tag{7}$$

Finally, we give the definition of the regret of minimizing any subproblem:

Definition 1. ***(Subproblem Regret)*** *Consider an algorithm generates the trajectory of states $\left\{\boldsymbol{\theta}^t \in \Theta\right\}_{t\in[T]}$, the regret of such an algorithm on loss function $\{\ell^t\}_{t\in[T]}$ is:*

$$\bar{\mathbf{R}} = \sum_{t=1}^{T} \ell^t\left(\boldsymbol{\theta}^t\right) - \inf_{\boldsymbol{\theta}^\star\in\Theta} \sum_{t=1}^{T} \ell^t(\boldsymbol{\theta}). \tag{8}$$

Theorem 1. *(Restated) Consider an algorithm with regret bound R_T that generates the trajectory of states for two base learners, for any parameter state $\boldsymbol{\theta} \in \Theta$, given a sequence of convex surrogate evaluation functions $\ell : \Theta \mapsto [0,1]_{a\in\mathcal{A}}$ drawn i.i.d. from some distribution $\mathcal{L}$, the expected error of the global learner $\boldsymbol{\theta}_g$ on both tasks over the test set can be bounded with probability at least $1-\delta$:*

$$\mathop{\mathbb{E}}_{\ell\sim\mathcal{L}} \ell\left(\boldsymbol{\theta}_g\right) \leq \mathop{\mathbb{E}}_{\ell\sim\mathcal{L}} \ell\left(\boldsymbol{\theta}\right) + \frac{\mathbf{R}_T}{T} + 2\sqrt{\frac{2}{T}\log\frac{1}{\delta}}. \tag{9}$$

Proof. From Theorem 1 and Remark 1, we obtain that

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\ell\sim\mathcal{L}} \ell\left(\boldsymbol{\theta}^t\right) \leq \frac{1}{T}\sum_{t=1}^{T} \ell^t\left(\boldsymbol{\theta}\right) + \frac{\bar{\mathbf{R}}}{T} + \sqrt{\frac{2}{T}\log\frac{1}{\delta}} \leq \mathop{\mathbb{E}}_{\ell\sim\mathcal{L}} \ell\left(\boldsymbol{\theta}\right) + \frac{\bar{\mathbf{R}}}{T} + 2\sqrt{\frac{2}{T}\log\frac{1}{\delta}}. \tag{10}$$

It is obvious that:

$$\frac{\bar{\mathbf{R}}}{T}+\sqrt{\frac{2}{T}\log\frac{1}{\delta}}\leq\frac{\mathbf{R}_T}{T}+\sqrt{\frac{2}{T}\log\frac{1}{\delta}}\quad\text{and}\quad\frac{\bar{\mathbf{R}}}{T}+2\sqrt{\frac{2}{T}\log\frac{1}{\delta}}\leq\frac{\mathbf{R}_T}{T}+2\sqrt{\frac{2}{T}\log\frac{1}{\delta}}. \tag{11}$$

So we obtain:

$$\frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{\ell\sim\mathcal{L}}\ell\left(\boldsymbol{\theta}^t\right)\leq\frac{1}{T}\sum_{t=1}^{T}\ell^t\left(\boldsymbol{\theta}\right)+\frac{\mathbf{R}_T}{T}+\sqrt{\frac{2}{T}\log\frac{1}{\delta}}\leq\mathop{\mathbb{E}}_{\ell\sim\mathcal{L}}\ell\left(\boldsymbol{\theta}\right)+\frac{\mathbf{R}_T}{T}+2\sqrt{\frac{2}{T}\log\frac{1}{\delta}}. \tag{12}$$

Recalling that in Section 3.3, $\boldsymbol{\theta}_g$ can be expressed by the linear combination of $\boldsymbol{\theta}_n$ and $\boldsymbol{\theta}_r$ through $t=1,\cdots,T$ since $\boldsymbol{\theta}_g$ is aggregateed by EMA, so the above inequality can be further derived by the Jensen's inequality (convex surrogate functions could be selected to evaluate the test errors instead of the 0-1 loss):

$$\begin{aligned}\mathop{\mathbb{E}}_{\ell\sim\mathcal{L}}\ell\left(\boldsymbol{\theta}_g\right)=\mathop{\mathbb{E}}_{\ell\sim\mathcal{L}}\ell\left(\sum_{t=1}^{T}\boldsymbol{\theta}^t\right)\leq\frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{\ell\sim\mathcal{L}}\ell\left(\boldsymbol{\theta}^t\right)&\leq\frac{1}{T}\sum_{t=1}^{T}\ell^t\left(\boldsymbol{\theta}\right)+\frac{\mathbf{R}_T}{T}+\sqrt{\frac{2}{T}\log\frac{1}{\delta}}\\&\leq\mathop{\mathbb{E}}_{\ell\sim\mathcal{L}}\ell\left(\boldsymbol{\theta}\right)+\frac{\mathbf{R}_T}{T}+2\sqrt{\frac{2}{T}\log\frac{1}{\delta}}.\end{aligned} \tag{13}$$

Note that this inequality also holds when applying weight averaging technique to the base-learner, because weight averaging is still the linear combination of all history states. □

References

[1] Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19:357–367, 1967. 4

[2] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. Inf. Theory*, 50(9):2050–2057, 2004. 4

[3] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 3