Common Q: Relation between Zipper and Theorem 3.2. Zipper divides the tradeoff problem into two independent tasks. $\mathbf{R}_T$ is the sum of the difference between the parameters of each base-learner and the theoretically optimal parameters for each task, so any strategy beneficial to reducing the error of each task will make $\mathbf{R}_T$ smaller and further lower the error bound of the global learner. Theorem 3.2 guarantees the upper bound of the risks given by the global learner trained by Zipper will get lower once the error for each task becomes lower.

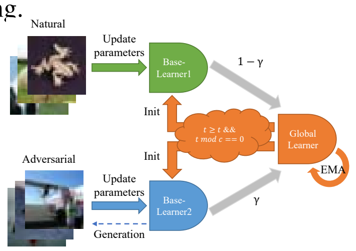
To Reviewer #2 (R2)

Q1: About the improvement and cost brought by Zipper. We compute the actual training time of TRADES and Zipper (serial/parallel version) using ResNet18 on RTX 3090 GPU in following Table. We also report the standard deviations over 5 runs to show the sensitivity of Zipper. Neither version of Zipper is slower than TRADES. Zipper does perform both NT and naive AT, but the cost of NT is negligible so the overhead (NT+AT) is smaller than TRADES. For the improvement of Zipper, it is hard to obtain acceptable robustness while maintaining clean accuracy above 89% in the joint training framework. 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%).

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

To Reviewer #4 (R4)

Q1: A detailed figure for the proposed pipeline. Valuable suggestion! We will add the following Figure for better understanding.



To Reviewer #5 (R5)

Q1: Meaning of Figure 1. Figure 1 shows natural examples are regularly clustered while the adversarial are weirdly scattered for they are elaborate. So we need different strategies to fit them separately rather than treat them as the same.

Q2: Things need to be rigorous. Adversarial examples are generated by fooling the base learner responsible for the adversarial task as Algorithm2 shown. As you said, the generation of adversarial examples does depend on the model so an overlap of two distributions may exist, but this does not affect our algorithm to deal with them separately. Our method is adaptive to this situation. For Line 165, we will add some constraints (e.g. the acceleration is applicable to the given assignment for its corresponding base learner only) to make it more rigorous.

Q3: What is the complementary manner. We assign different tasks to each base learner. The inputs or loss value in one task only affect its own parameters. And the global learner always collects parameters of both base learners. In Line 214, base-learners at the early stage are not well trained which may mislead the training procedure and further accumulate bias when mixing them.

Q4: Question about improvement, cost and Theorem 3.2. Please refer to the reply to Common Q and R2. We should emphasize the final obtained model is the same size as other trained models are. It is still a fair competition considering its competitive training time. The eval function in Theorem 3.2 can be the averaged 0-1 loss or other surrogate functions.

Q5: Other minor concerns. t stands for the t-th iteration and c is frequency of communication. $\boldsymbol{\theta}$ represents *any* parameter vector in Θ and $\boldsymbol{\theta}_g$ is the parameter vector of the global learner. In Line 196, the oracle state represents the theoretically optimal parameters for each task. We select the best checkpoint based on the sum of clean accuracy and robust accuracy following the common settings in the adversarial training field. For space limitation, we omit standard deviations as they are very small ($< 0.5\%$) for Zipper. We will add them in the revision.

To Reviewer #6 (R6)

Q1: Questions about experimental results. We conducted experiments on MNIST ($\varepsilon = 0.3$) and SVHN using ResNet-18 with the same setup in Section 4.1. We ran 5 individual trials and results with standard deviations are shown in the following Table. As for the marked gain on WRN-32-10, WA technique can greatly boost the robustness on a larger model [1, 2] so the tradeoff performance on WRN-32-10 further benefits than ResNet-18. And we apply techniques in [3] as you suggested to WRN-32-10 whose NAT/PGD20/AA achieve 83.1%/56.37%/55.01%. If we force NAT to be higher than 90%, accuracy against PGD20/AA will decrease to 51.28%/50.31%, which are much inferior to our Zipper.

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

Q2: Concerns about motivation and discussion on learning rate. Although the best results still come from using SGD, the learning rate for different tasks can be customized which is not feasible in the joint framework. We add the analysis of the learning rate in the following Table.

	N:0.1/A:0.01	N:0.1/A:0.1	N:0.1/A:0.001	N:0.01/A:0.01	N:0.01/A:0.1
NAT/AA	89.09/46.07	90.12/41.86	90.45/43.55	88.4/48.03	88.25/42.98

Q4: Other minor concerns. We fix the typos and use different symbols to define distributions.We select γ and c for the best tradeoff performance and they are insensitive to the validation set we select. γ is a scalar but we dynamically adjust the value along the training process using a piecewise linear function to decrease. λ follows the best setting in the original TRADES paper. Z_n or Z_r are optimizers for naturally and adversarially trained models. And we will simplify the notation and avoid using the bar to differentiate the characters.

Reference: [1] Fixing Data Augmentation to Improve Adversarial Robustness [2] Improving Robustness using Generated Data. [3] Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples.