Based on the analysis of descriptive capacity and model diversity, we can see that generated ghost networks can provide accurate yet diverse descriptions of the data manifold, which is beneficial to learn transferable adversarial examples as we will experimentally prove below.

Single-model Attack

Firstly, we evaluate the ghost networks in single-model attack, where attackers can only access one base model B trained from scratch. To demonstrate the effectiveness of our method, we design five experimental comparisons. The setting, black-box attack success rate, and properties are shown in Table 1. The difference among five experiments is the type of model to attack, the number of models ensembled by standard ensemble (**?**) in each iteration, and the number of models ensembled by longitudinal ensemble in each branch of the standard ensemble. For example, Exp. S5 combines two ensemble methods, that is, we do a standard ensemble of 10 models for each iteration of attack and a longitudinal ensemble of 10 models. Therefore, in Exp. S5, the intrinsic number of models is 100.

We attack 6 normally-trained networks and test on all the 9 networks (include 3 adversarially-trained networks). The attack rate is shown in Table 1. To save space, we report the average attack rate for black-box models. Individual cases are shown in Fig. 6.

As can be drawn from Table 1, a single ghost network is worse than the base network (Exp. S2 *vs.* Exp. S1), because the descriptive power of ghost networks is inferior to base networks. However, by leveraging the longitudinal ensemble, our work achieves a much higher attack rate in most settings (Exp. S3 *vs.* Exp. S1). This observation firmly demonstrates the effectiveness of ghost networks in learning transferable adversarial examples. It should be mentioned that the computational cost of Exp. S3 almost remains the same as Exp. S1 for two reasons. First, the 10 ghost networks used in Exp. S3 are not trained but eroded from the base model and used on-the-fly. Second, multiple ghost networks are fused via the longitudinal ensemble, instead of the standard ensemble method in **?** (**?**).

The proposed ghost networks can also be fused via the standard ensemble method, as shown in Exp. S4. In this case, we achieve a higher attack rate at the sacrifice of computational efficiency. This observation inspires us to combine the standard ensemble and the longitudinal ensemble as shown in Exp. S5. As we can see, Exp. S5 consistently beats all the compared methods in all the black-box settings. Of course, Exp. S5 is as computational expensive as Exp. S4. However, the additional computational overhead stems from the standard ensemble rather than longitudinal ensemble.

Note that in all the experiments presented in Table 1, we use only one individual base model. Even in the case of Exp. S3, all the to-be-fused models are ghost networks. However, the generated ghost networks are never stored or trained, meaning no extra space complexity. Therefore, the benefit of ghost networks is obvious. Especially when comparing Exp. S5 and Exp. S1, ghost networks can achieve a substantial improvement in black-box attack.

Based on the experimental results above, we arrive at a similar conclusion as **?** (**?**): the number of intrinsic models is essential to improve the transferability of adversarial examples. However, a different conclusion is that it is less necessary to train different models independently. Instead, ghost networks is a computationally cheap alternative enabling good performance. When the number of intrinsic models increases, the attack rate will increase. We will further exploit this phenomenon in multi-model attack.

In Fig. 6, we select two base models, *i.e.*, Res-50, and Inc-v3, to attack and present their performances when testing on all the 9 base models. It is easy to observe the improvement of transferability by adopting ghost networks.

Multi-model Attack

We evaluate ghost networks in multi-model setting, where attackers have access to multiple base models.

Same Architecture and Different Parameters We firstly evaluate a simple setting of multi-model attack, where base models share the same network architecture but have different weights. The same three Res-50 models as in the **Analysis of Ghost Networks** section are used. The settings of 6 experiments are shown in Table 2. Besides a new parameter #B (the number of trained-from-scratch models), others are the same as the single model attack setting. When #B is 1, we will use Res-50-A as the only one base model, and settings are the same as single-model attack. When #B is 3, #S is always 3, and each branch of the standard ensemble is assigned to a different base model. In Exp. M4 and Exp. M6, the ghost network(s) in each standard ensemble branch will be generated by the base model assigned to that branch.

The adversarial examples generated by each method are used to test on all the 9 models. We report the average attack rates in Table 2. It is easy to understand that Exp. M2 performs better than Exp. M1, Exp. M3, and Exp. M4 as it has three independently trained models. However, by comparing Exp. M5 with Exp. M2, we observe a significant improvement of attack rate. For example, By using MI-FGSM as the attack method, Exp. M5 beats Exp. M2 by 6.70. Although Exp. M5 only has 1 base model and Exp. M2 has 3, Exp. M5 actually fuses 30 intrinsic models. Such a result further supports our previous claim that the number of intrinsic models is essential, but it is less necessary to obtain them by training from scratch independently. Similarly, Exp. M6 yields the best performance as it has 3 independently trained models and 30 intrinsic models.

Different Architectures Besides the baseline comparison above, we then evaluate ghost networks in the multi-model setting following **?** (**?**). We attack an ensemble of 5 out of 6 normally-trained models in this experiment, then test the hold-out network (black-box setting). We also attack an ensemble of 6 normally-trained models and test on the 3 adversarially-trained networks to evaluate the transferability of the generated adversarial examples in black-box attack.

The results are summarized in Table 3, the performances in black-box attack are significantly improved. For example, when holding out Res-50, our method improves the performance of I-FGSM from 71.08 to 80.22, and that of

Methods	Hold-out						Ensemble		
	-Res-50	-Res-101	-Res-152	-IncRes-v2	-Inc-v3	-Inc-v4	Inc-v3$_{\text{ens3}}$	Inc-v3$_{\text{ens4}}$	IncRes-v2$_{\text{ens}}$
I-FGSM	71.08	71.16	67.92	46.60	59.98	50.86	15.94	8.54	13.72
I-FGSM + **ours**	80.22	79.80	77.02	60.20	73.18	67.84	25.80	13.56	21.42
MI-FGSM	79.32	79.14	77.26	64.24	72.22	66.64	29.98	16.66	26.16
MI-FGSM + **ours**	**87.14**	**86.14**	**84.64**	**74.18**	**82.06**	**79.18**	**39.56**	**21.24**	**32.68**

Table 3: The attack rate (%) comparison of multi-model attack. "Ensemble" means attack all 6 naturally-trained models. "Hold-out" means attack 5 out of 6 models. The sign "-" indicates the name of the hold-out model.

Methods	Black-box Attack				White-box Attack			
	TsAIL	iyswim	Anil Thomas	Average	Inc-v3_adv	IncRes-v2_ens	Inc-v3	Average
No.1 Submission	13.60	43.20	43.90	33.57	94.40	93.00	**97.30**	94.90
No.1 Submission+**ours**	**14.80**	**52.28**	**51.68**	**39.59**	**97.62**	**96.00**	95.48	**96.37**

Table 4: The attack rate (%) comparison in the NeurIPS 2017 Adversarial Challenge.

MI-FGSM from 79.32 to 87.14. When testing on the three adversarially-trained networks, the improvement is more notable. These results further testify the ability of ghost networks to learn transferable adversarial examples.

NeurIPS 2017 Adversarial Challenge

Finally, we evaluate our method in a benchmark test of the NeurIPS 2017 Adversarial Challenge (**?**). For performance evaluation, we use the top-3 defense submissions (black-box models), *i.e.*, TsAIL[1], iyswim[2] and Anil Thomas[3], and three official baselines (white-box models), *i.e.*, Inc-v3$_{\text{adv}}$, IncRes-v2$_{\text{ens}}$ and Inc-v3. The test dataset contains 5000 images with the same 1000-class labels as ImageNet (**?**). Following the experimental setting of the No.1 attack submission (**?**), we attack on an ensemble of Inc-v3, IncRes-v2, Inc-v4, Res-152, Inc-v3$_{\text{ens3}}$, Inc-v3$_{\text{ens4}}$, IncRes-v2$_{\text{ens}}$ and Inc-v3$_{\text{adv}}$ (**?**). The ensemble weights are set to 1/7.25 equally for the first seven networks and 0.25/7.25 for Inc-v3$_{\text{adv}}$. The total iteration number is set to 10, and the maximum perturbation ϵ is randomly selected from $\{4, 8, 12, 16\}$. The step size $\alpha = \epsilon/10$. The results are summarized in Table 4. Consistent with previous experiments, we observe that by applying ghost networks, the performance of the No. 1 submission can be significantly improved, especially with black-box attack. For example, the average performance of black-box attack is changed from 33.57 to 39.59, an improvement of 6.02. The most remarkable improvement is achieved when testing on iyswim, where ghost networks leads to an improvement of 9.08. This suggests that our proposed method can generalize well to other defense mechanisms.

Conclusion

This paper focuses on learning transferable adversarial examples for adversarial attacks. We propose, for the first time, to exploit network erosion to generate a kind of virtual models called ghost networks. Ghost networks, together with the coupled longitudinal ensemble strategy, is an effective and efficient tool to improve existing methods in learning transferable adversarial examples. Extensive experiments have firmly demonstrated the efficacy of ghost networks. Meanwhile, one can potentially apply erosion to residual unit by other methods or **densely** erode other typical layers (*e.g.*, batch norm (**?**) and relu (**?**)) **through** a neural network. We suppose these methods could improve the transferability as well, and leave these issues as future work. noindent**Acknowledgements** This paper is supported by ONR award N00014-15-1-2356.

In sapiente aspernatur temporibus deleniti accusamus repudiandae quae, mollitia inventore tempora rerum eligendi totam minima explicabo deserunt quo?Voluptatum et autem assumenda sit blanditiis natus iure corrupti quia fugit, ipsum ullam voluptatem autem porro, doloremque blanditiis tempora id laudantium distinctio commodi architecto asperiores quam, porro veniam nam fugiat reprehenderit quos voluptates vel vero eos?Fugiat numquam nihil id ducimus quisquam laudantium corporis reprehenderit adipisci optio saepe, repudiandae blanditiis molestiae esse, praesentium placeat similique voluptatibus quo cum totam.Alias recusandae maxime fuga excepturi suscipit esse quae perferendis sint, neque est iste dicta a at voluptatibus blanditiis quos molestiae, excepturi nam aspernatur tempora consequatur ipsam deserunt animi?Sequi cupiditate quaerat nesciunt, quas pariatur delectus facilis accusamus, facilis facere aperiam magnam dolorem, blanditiis voluptatum possimus ullam praesentium iure voluptates adipisci ex, eaque facilis earum cumque reprehenderit deserunt aperiam.Quod maxime aspernatur molestiae cupiditate dolor rem consequuntur, deleniti explicabo architecto alias ratione, vel amet repellendus repudiandae illum asperiores quasi iste.Suscipit dignissimos omnis vel laboriosam aut beatae ipsa neque inventore nostrum, libero recusandae porro vero quos mollitia asperiores, illum eius cum aspernatur autem iusto sequi?Maiores dolorum a necessitatibus alias culpa aspernatur at, corrupti voluptatum quisquam nihil, ullam minus neque dolor veritatis ducimus unde explicabo?Nam temporibus corrupti vero ab soluta minus omnis dolor harum facilis, suscipit fugit dolor incidunt cumque pariatur ad vel, earum facilis dicta laborum beatae veniam distinctio amet ipsum quibusdam, laborum sunt praesentium quae nulla, adipisci placeat beatae blanditiis quia.Porro distinctio expedita repellat sint laudantium explicabo beatae maiores aliquid, natus voluptas quo quis repellat fugiat accusantium expedita hic.Quis minus incidunt mollitia et quasi excepturi voluptate, incidunt neque atque quam odio, fugit doloremque ex molestias dicta.Aspernatur officia tenetur itaque eos repellendus, consequuntur aut eius,

[1] https://github.com/lfz/Guided-Denoise

[2] https://github.com/cihangxie/NIPS2017_adv_challenge_defense

[3] https://github.com/anlthms/nips-2017/tree/master/mmd