Google

CycleGAN: a Master of Steganography

Casey Chu
caseychu@stanford.edu

Andrey Zhmoginov
azhmogin@google.com

Mark Sandler
sandler@google.com

Image-to-Image Translation with CycleGAN

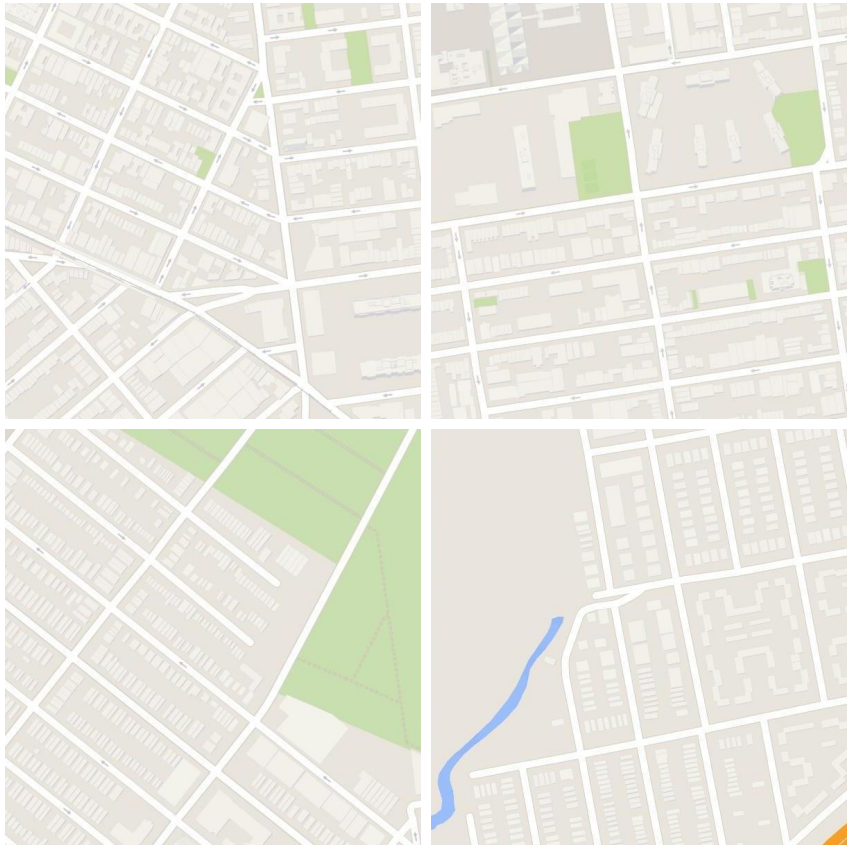
CycleGAN (Zhu et al. 2017) permits *unpaired* translation between two image domains.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. “Unpaired image-to-image translation using cycle-consistent adversarial networks.” (2017).

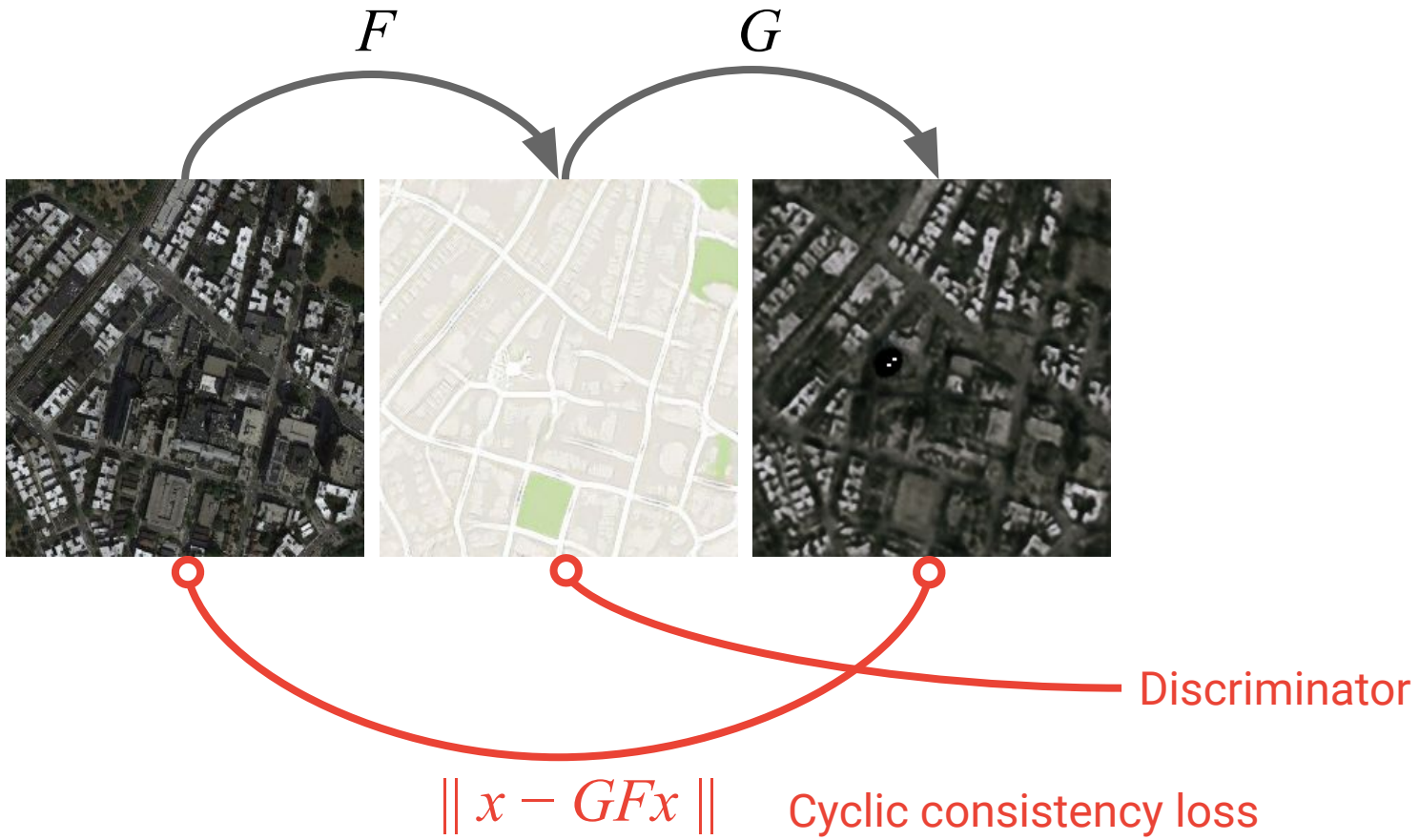
In our experiments, we fix one domain to be aerial photographs and the other domain to be regular maps from Google Maps.

Domain A: **aerial imagery**

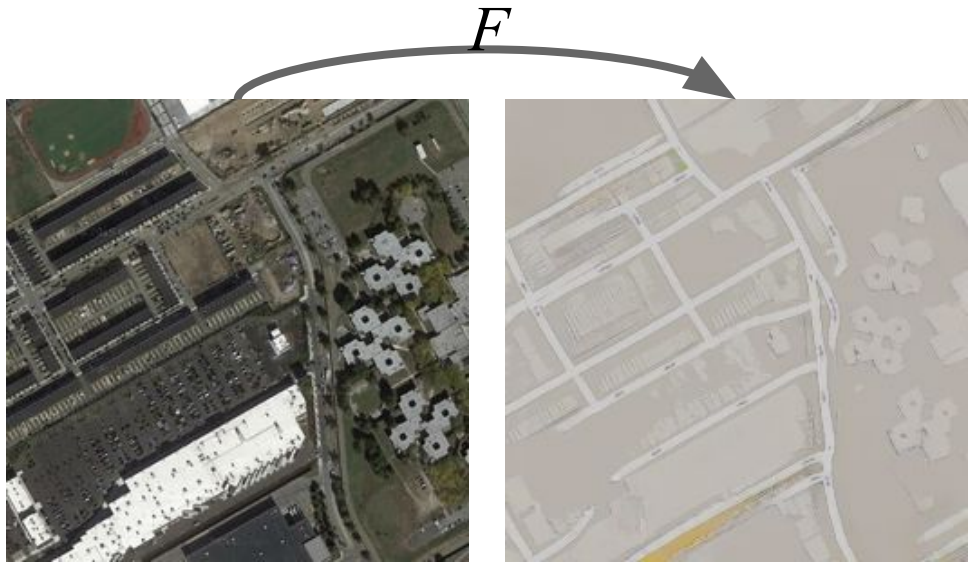
Domain B: **maps**

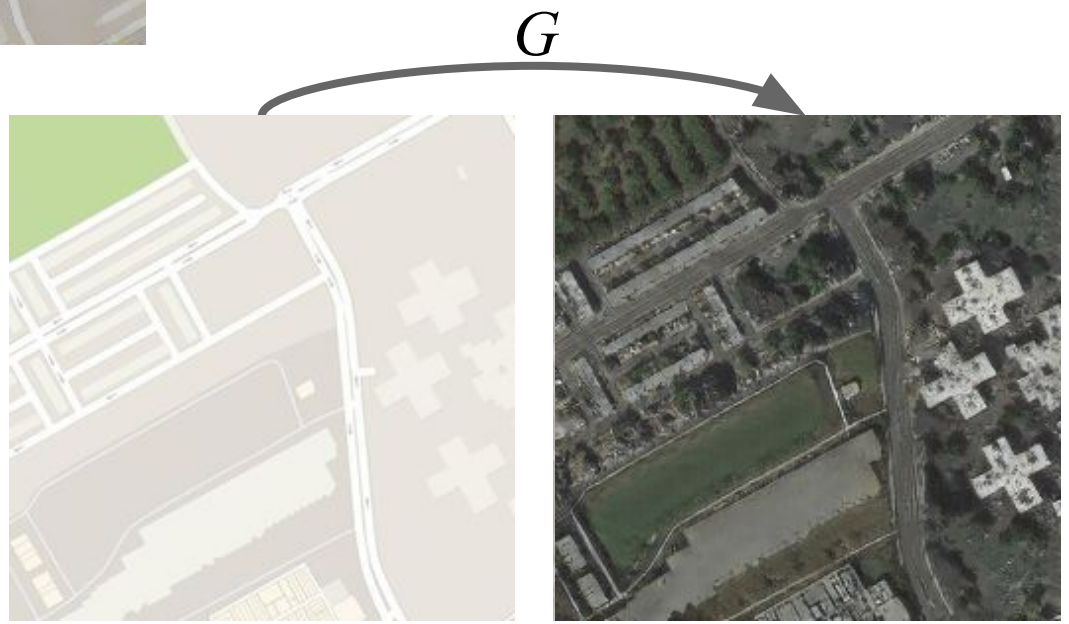


CycleGAN defines two functions, F and G, and places a cyclic *consistency* loss and a GAN loss on them.



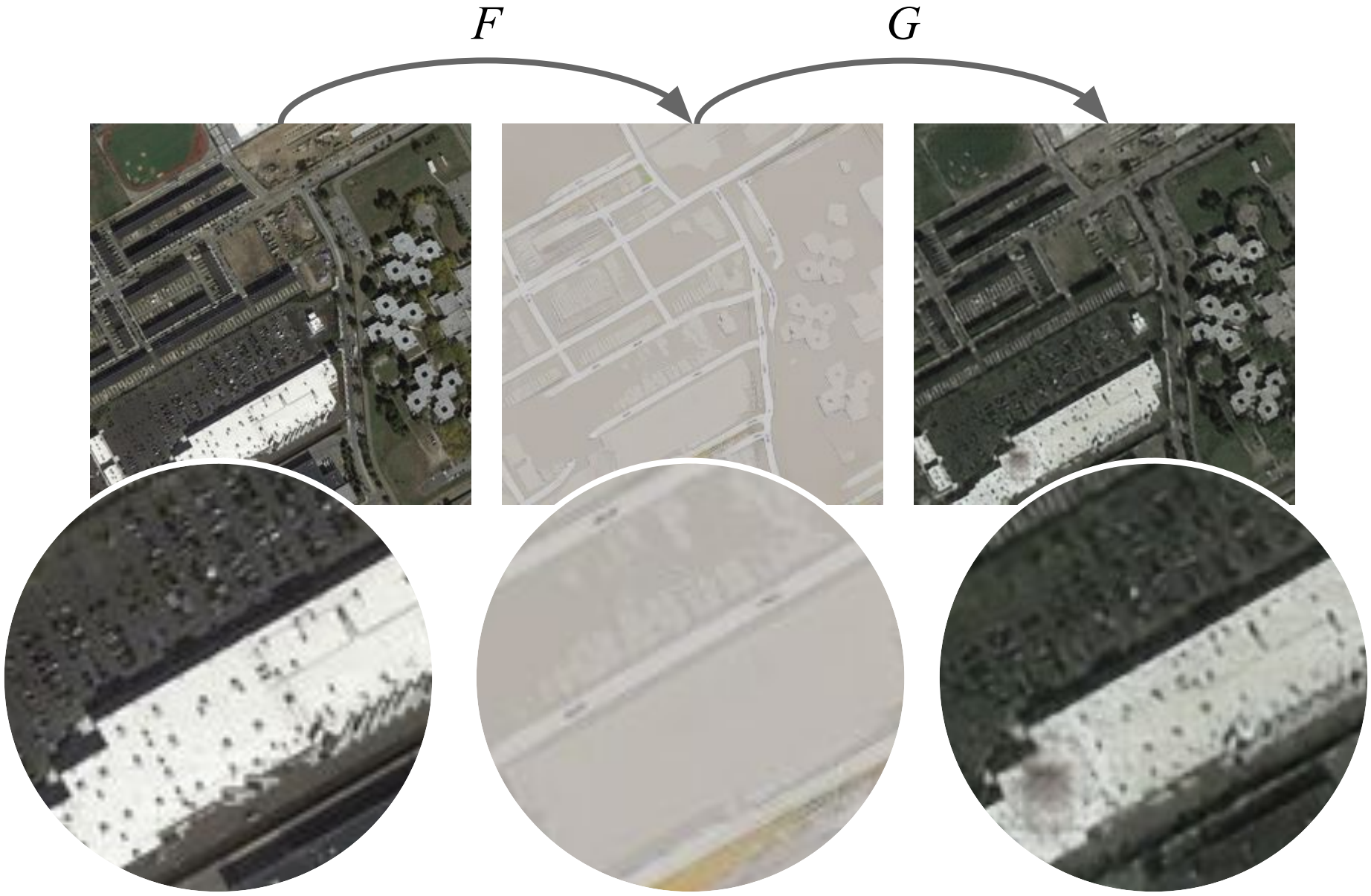
Once trained, the generated samples seem realistic.



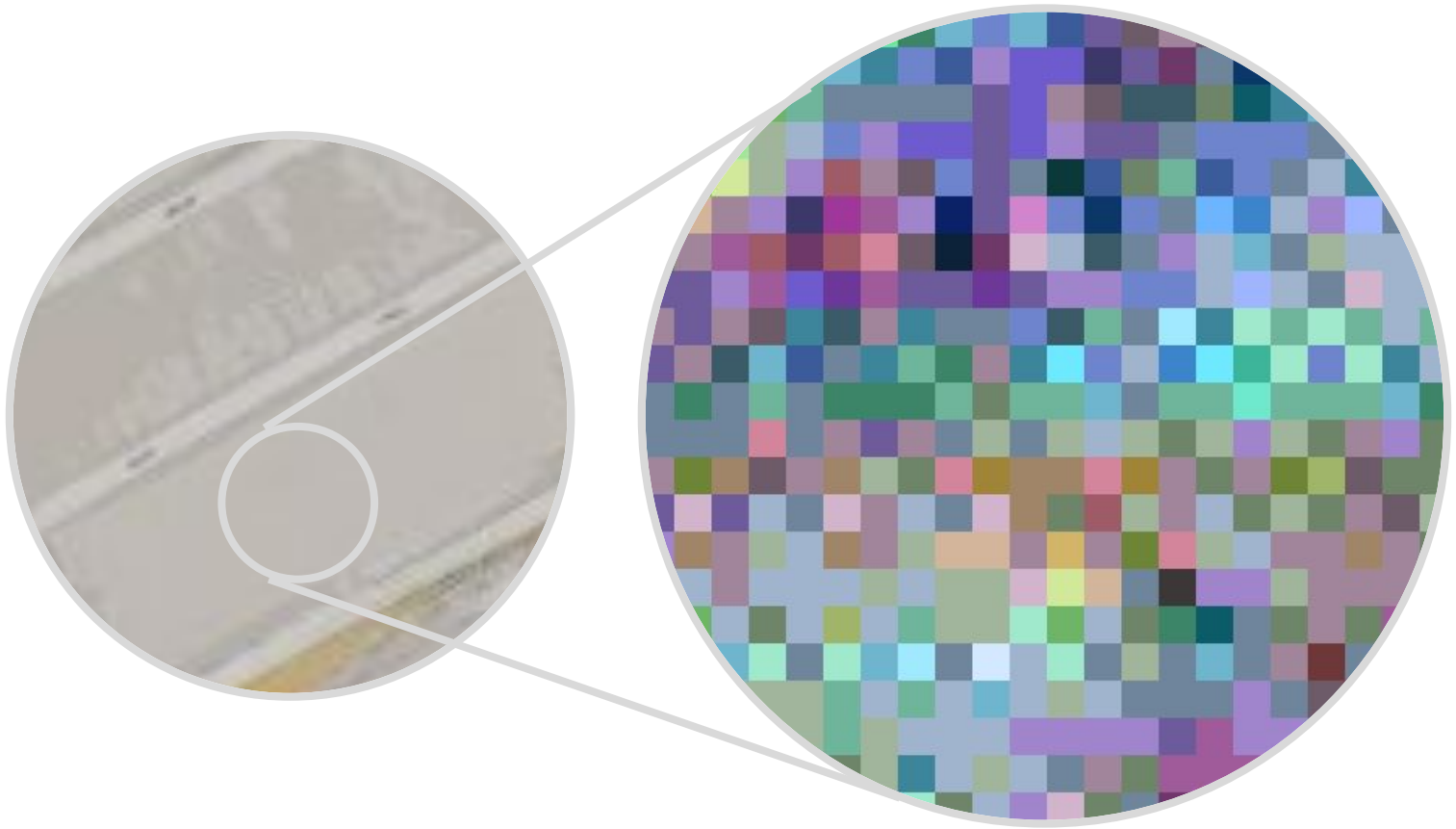


Hidden Information

GFx seems to contain information not present in Fx.



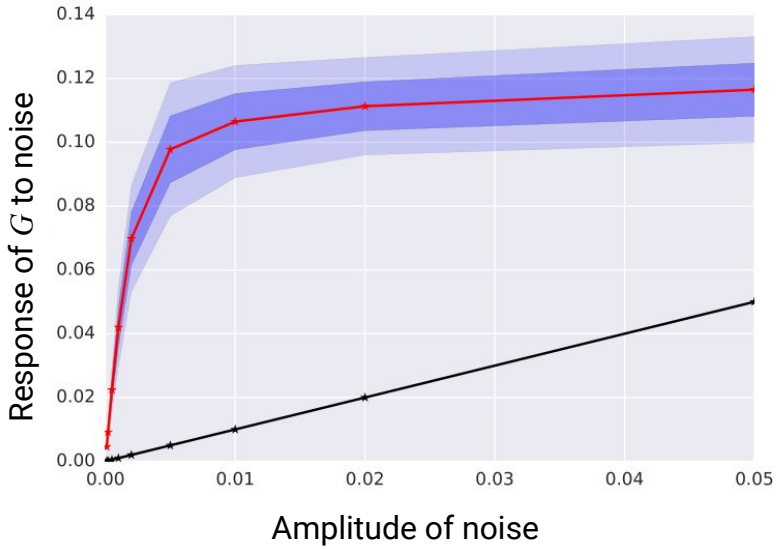
CycleGAN needs to hide information inside the map to satisfy cycle consistency, because aerial photos contain much more information than maps do.

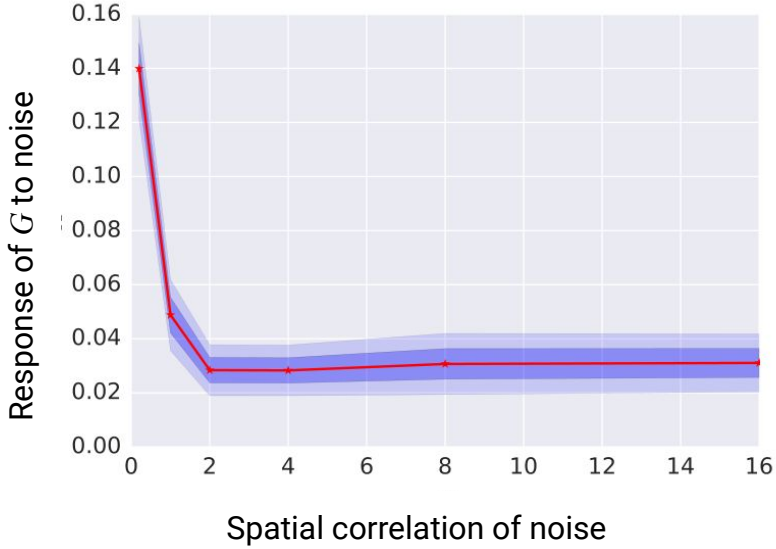


Shown with amplified contrast.

Even JPEG encoding is enough to destroy the hidden information.

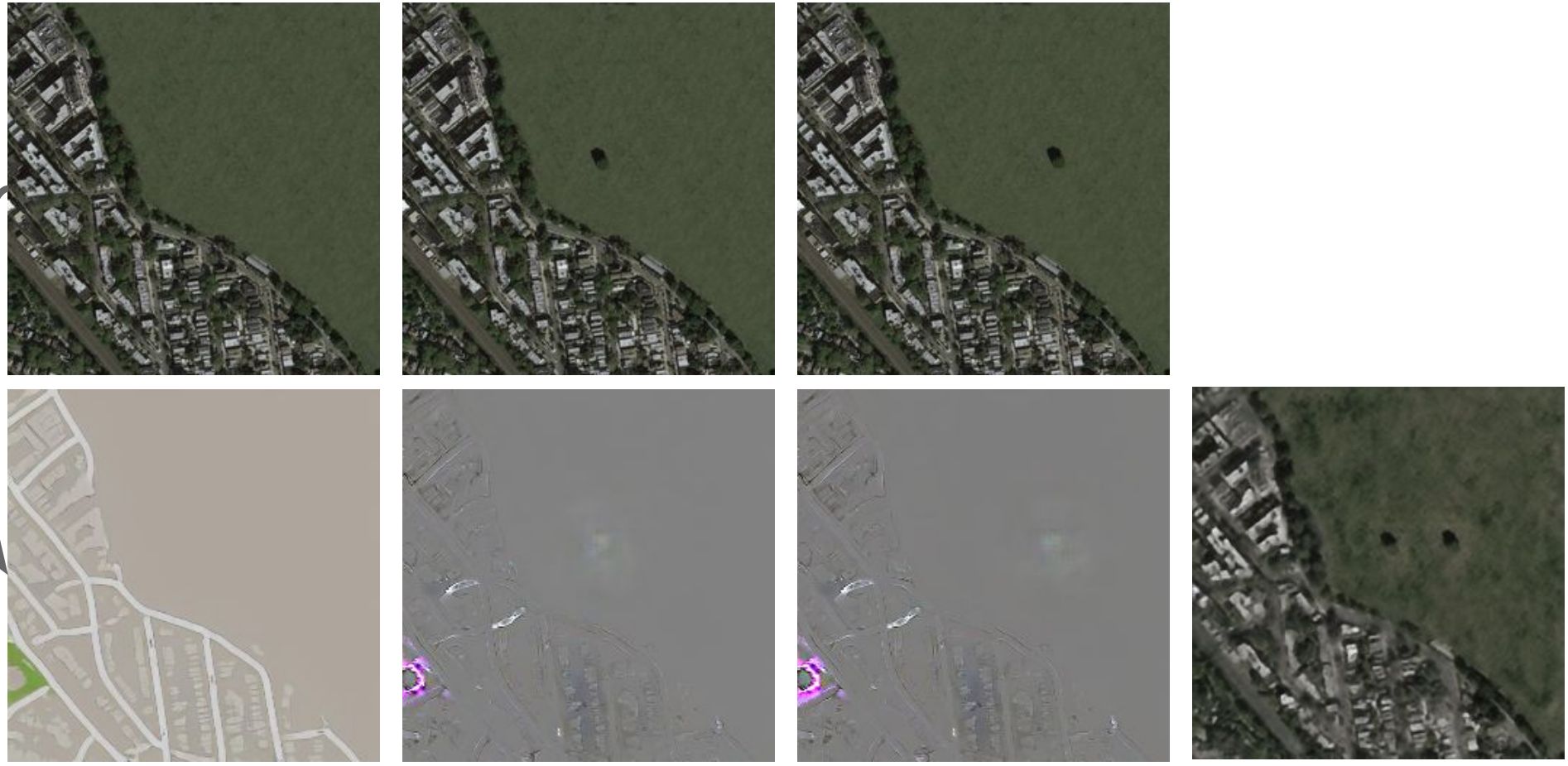
We inject noise into the generated map and study how the reconstruction is destroyed. We conclude that the hidden signal is a high-frequency, low-amplitude one.





Note: image value is scaled to [0, 1]; 0.01 corresponds to approximately 3 RGB values when in the range [0, 255]

The encodings are roughly additive.



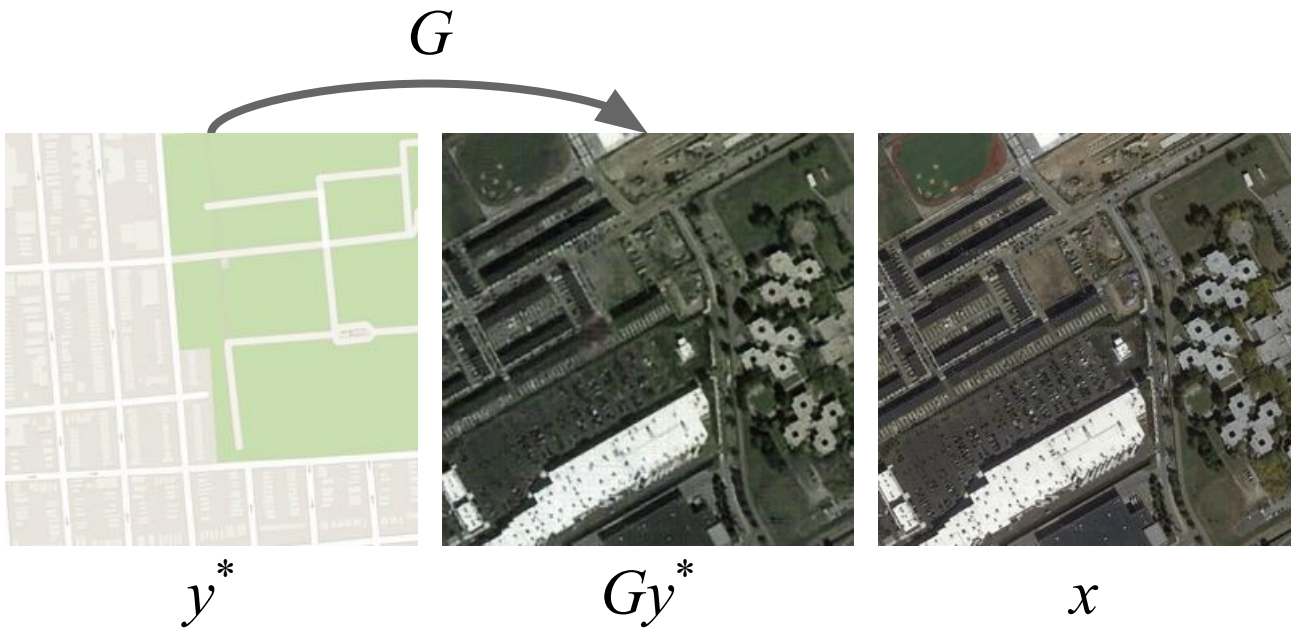
Information Hiding as an Adversarial Attack

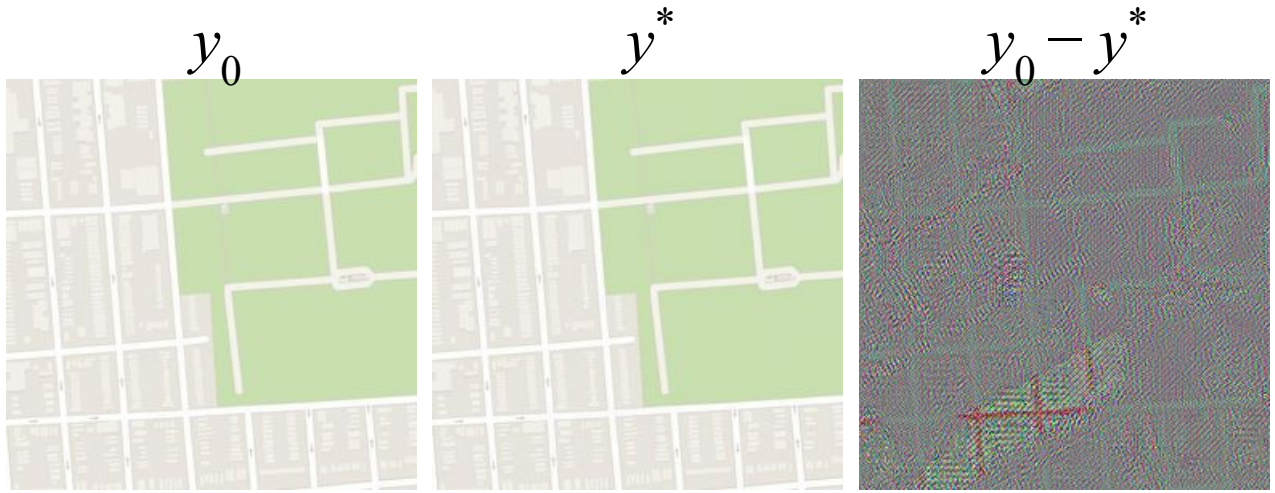
We attempt to hide an arbitrary aerial image inside an arbitrary map by applying gradient descent to the following objective, starting from y_0, similar to an adversarial attack (Szegedy et al. 2014).

y_0 x

$$y^* = \arg\min_y || Gy - x ||$$

It is possible to hide an arbitrary aerial image — in fact, the map changes only imperceptibly.





(amplified for visibility)

We may interpret the CycleGAN training procedure as performing an attack *against itself*, where F provides adversarial examples for G.

Adversarial attack:

$$y^* = \arg\min_y \| Gy - x \|$$

CycleGAN training criterion:

$$F, G = \arg\min_{F, G} \| GFx - x \| + \text{other terms}$$

The fact that we can so easily perform an adversarial attack suggests that this is almost certainly happening during training.

Takeaways

1. Generative models may be vulnerable to adversarial attacks — especially CycleGAN.

These results imply that generative models are not immune to adversarial attacks, so caution is required in deploying them for problems of importance.

This particular vulnerability arises from one domain (aerial imagery) having much larger entropy than the other (maps). Preliminary experiments that artificially increase the entropy of the other domain, by adding a fourth channel of pure noise to the maps, seem to make the model more immune to adversarial attacks.

2. When designing loss functions, be aware that compositions of neural networks may lead to unintuitive results.

Could other models also “cheat” by leveraging adversarial attacks, just as CycleGAN does? For example, in the following GAN loss, could G be seen as attacking D?

$$G = \arg\max_G \mathbb{E}_{z \sim p(z)} [\log D(G(z))]$$

Could applying techniques that guard against adversarial attacks stabilize the training of multi-component models?