Towards Evaluating the Robustness of Neural Networks

Nicholas Carlini
Google

David Wagner
UC Berkeley

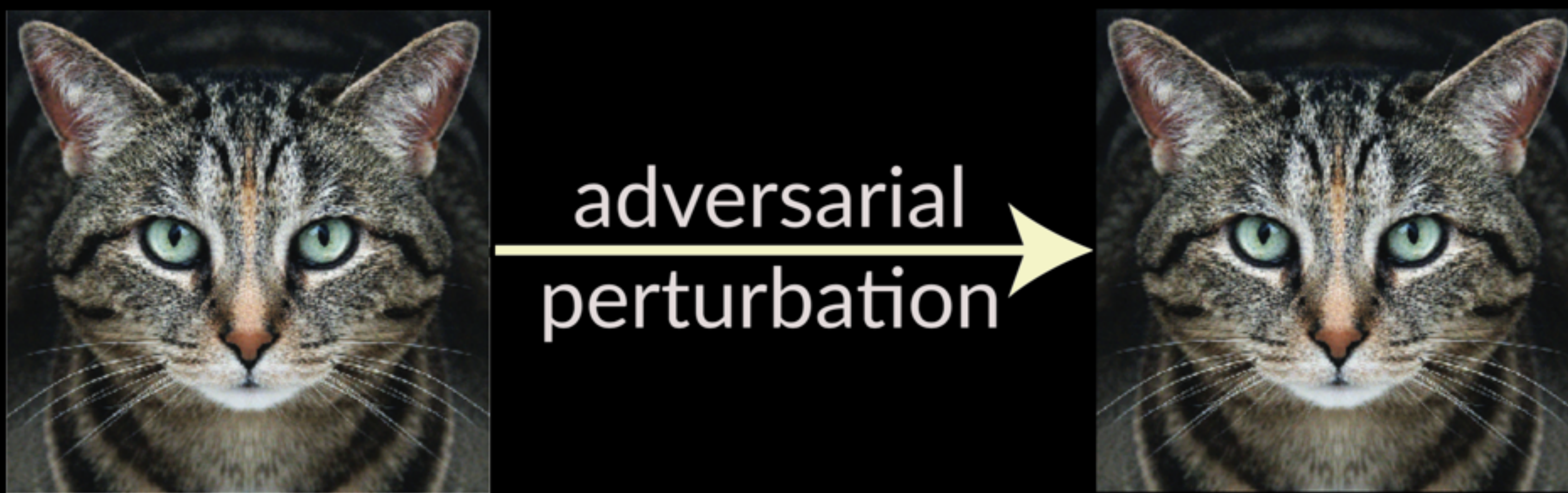
88% **tabby cat**

adversarial
perturbation

88% **tabby cat**

adversarial perturbation

88% **tabby cat**



88% **tabby cat** 99% **guacamole**

Why should we care about adversarial examples?

Make ML ***robust***

Make ML ***better***

Background: Adversarial Examples

- For a classification neural network F(x)
- Given an input X classified as label L ...
- ... it is easy to find an X′ close to X
- ... so that F(X′) != L

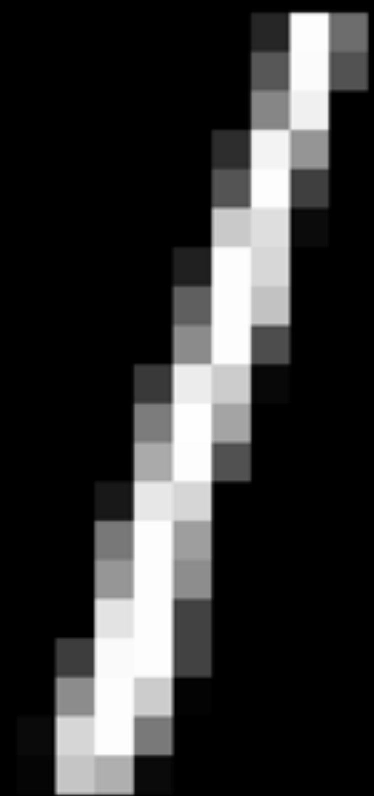
Distance Metrics

- "Adversarial examples are close to the original"
- How do we define **close**?
  - This is what lets us compare attacks.
- In what domain? Images.

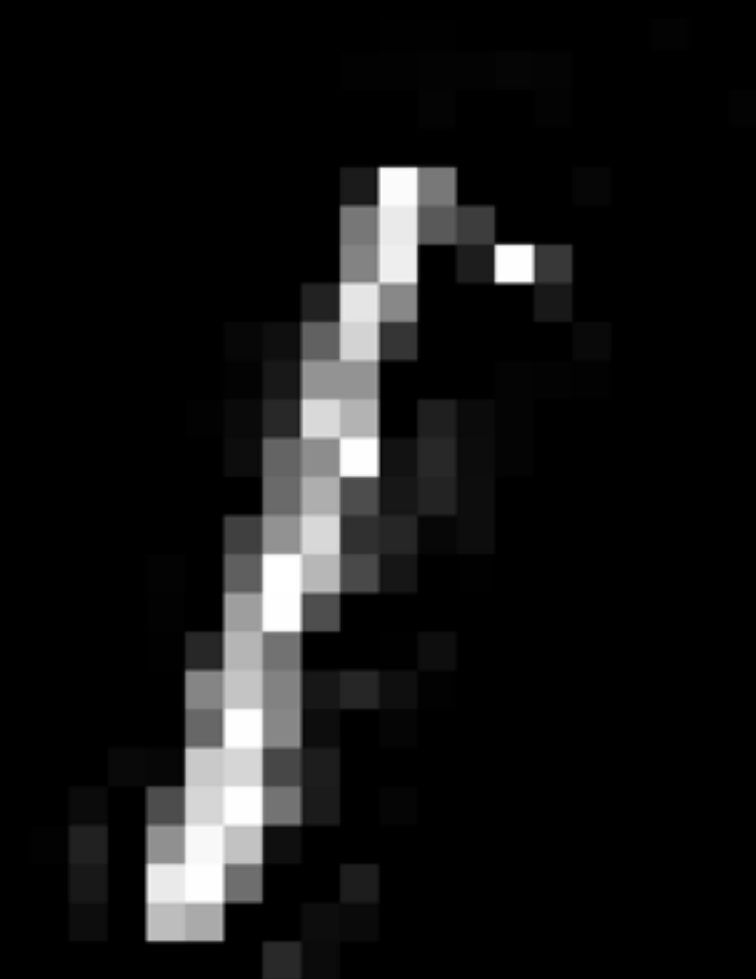
Distance Metrics

- L_p distance metrics:
  - L_0 - number of pixels changed
  - L_2 - standard Euclidian distance
  - $L_{infinity}$ - amount each pixel can be changed

If any L_p distance is small, the two images should be visually similar



Classified as a 1



Classified as a 0

For this talk:

Assume complete knowledge of model parameters

(but lots of work exists for other threat models)

Two ways to evaluate robustness:

1. Construct a proof of robustness
2. Demonstrate constructive attack

Proving Robustness

- It is possible to prove robustness
  - ... for specific input points
  - ... on simple datasets (~~MNIST~~ CIFAR-10)
  - ... for small networks (~~100~~ 10,000 neurons)
  - ... for ReLU activations

Finding Adversarial Examples

- Formulation: given input x, find x′ where
  minimize $d(x,x')$
  such that $F(x') = T$
  x′ is "valid"

- Gradient Descent to the rescue?

- Non-linear constraints are hard

Reformulation

- Formulation:
  minimize $d(x,x') + g(x')$
  such that x' is "valid"

- Where $g(x')$ is some kind of loss function on how close $F(x')$ is to target T

  - $g(x')$ is small if $F(x') = T$

  - $g(x')$ is large if $F(x') \neq T$

Reformulation

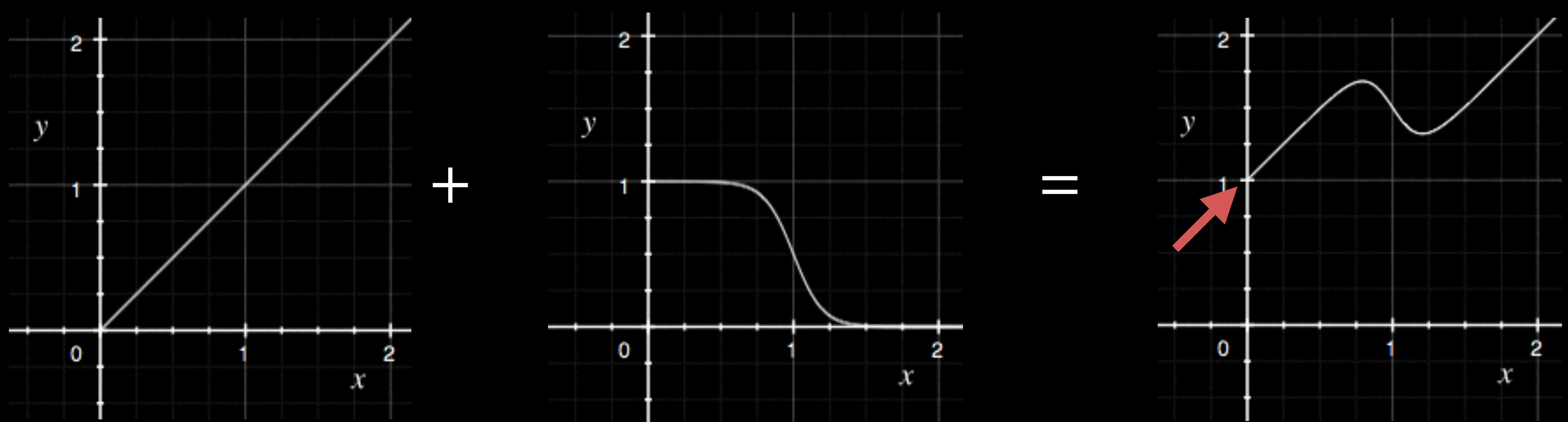
- For example
  - $g(x') = (1-F(x')_T)$
- If $F(x')$ says the probability of T is 1:
  - $g(x') = (1-F(x')_T) = (1-1) = 0$
- $F(x')$ says the probability of T is 0:
  - $g(x') = (1-F(x')_T) = (1-0) = 1$

Does this work?

- Fo
  mi
  su

Problem 1:
Global minimum is not an adversarial example

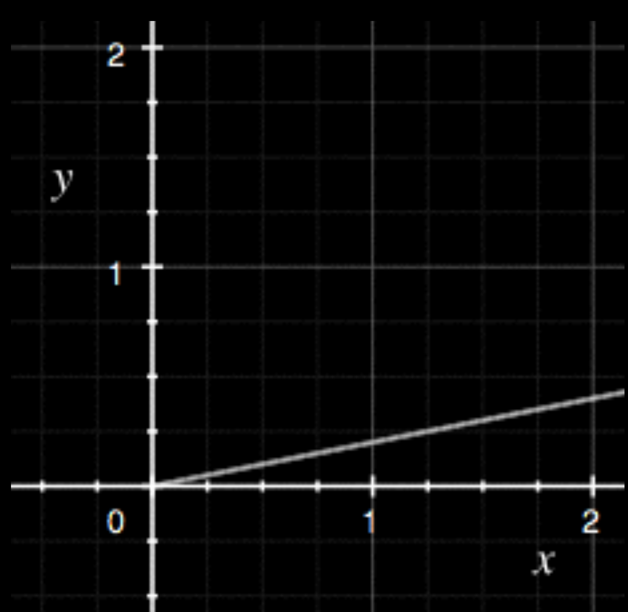
d(x,x′) + g(x′)



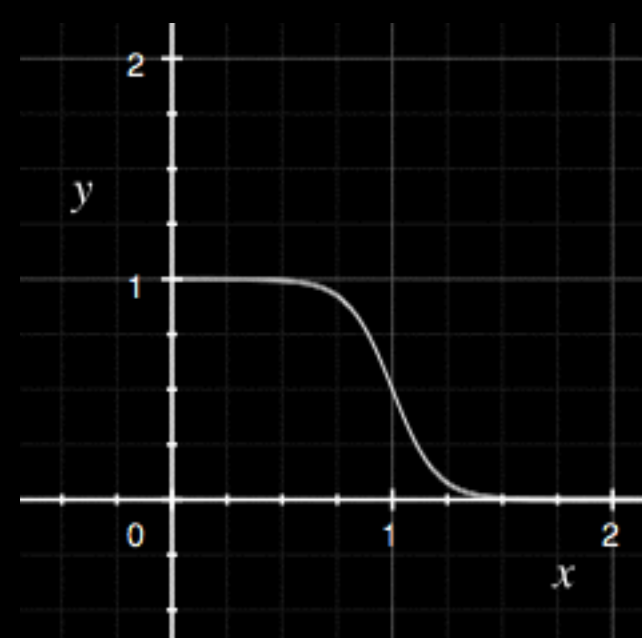
Does this work?

- Formulation:
  minimize $d(x,x')/5 + g(x')$
  such that x' is "valid"

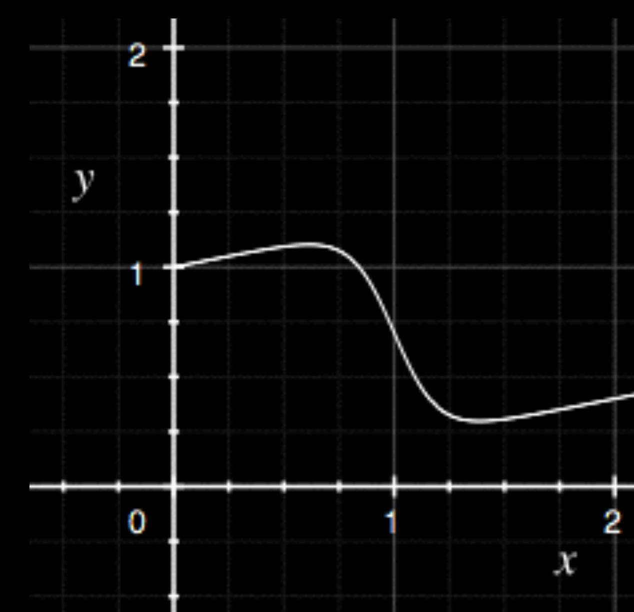
$d(x,x')/5$ + $g(x')$



+



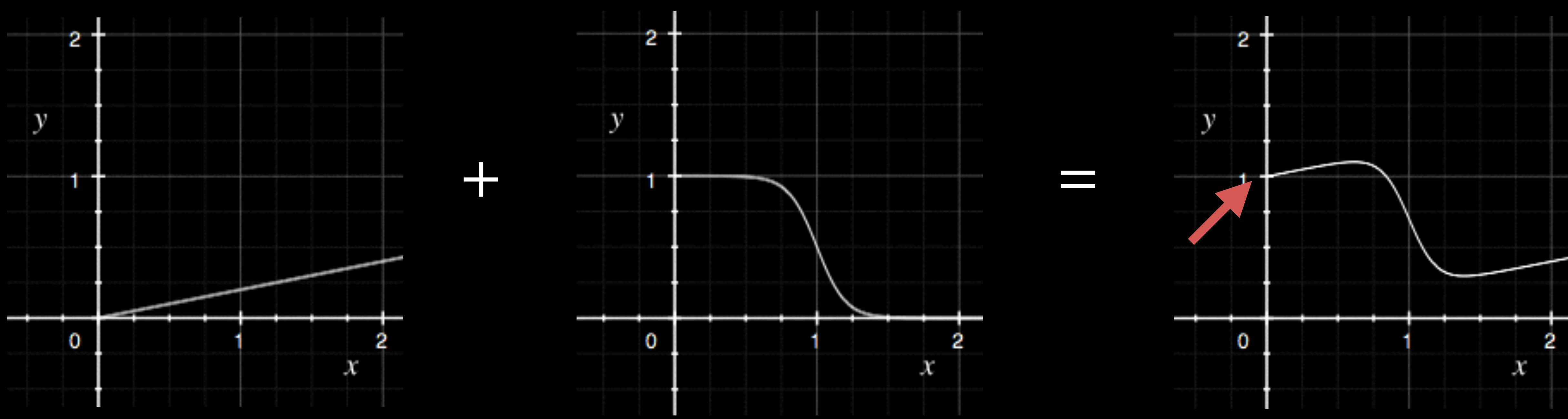
=



Does this work?

Problem 2:
Gradient direction does not point toward the global minimum

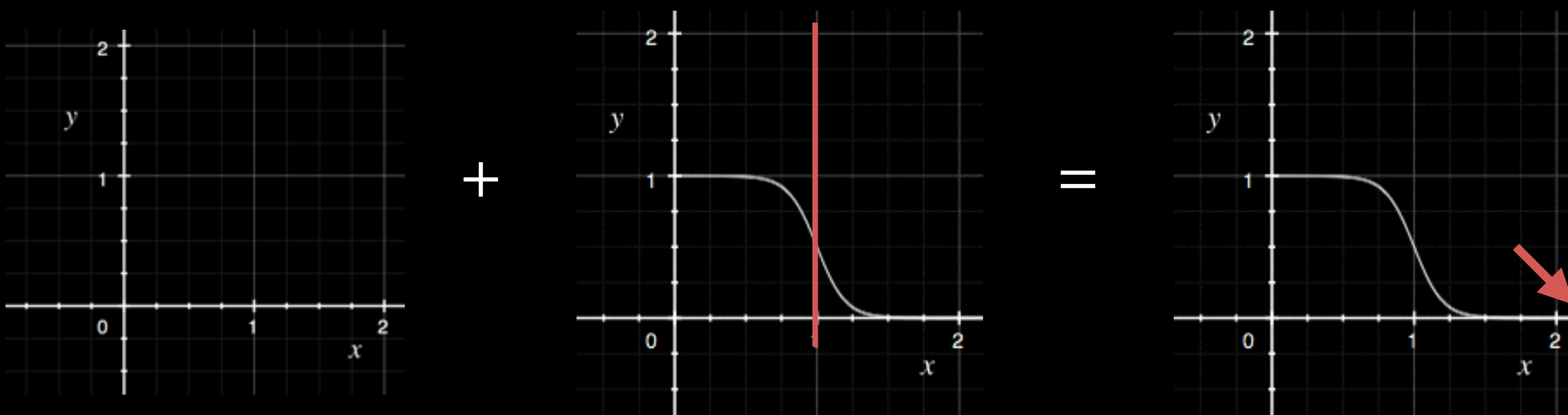
d(x,x′)/5 + g(x′)



Does this work?

Problem 3:
Global minimum is not the minimally perturbed adversarial example

d(x,x′)/1e10 + g(x′)



Constructing a better loss function

1. Global minimum at the decision boundary

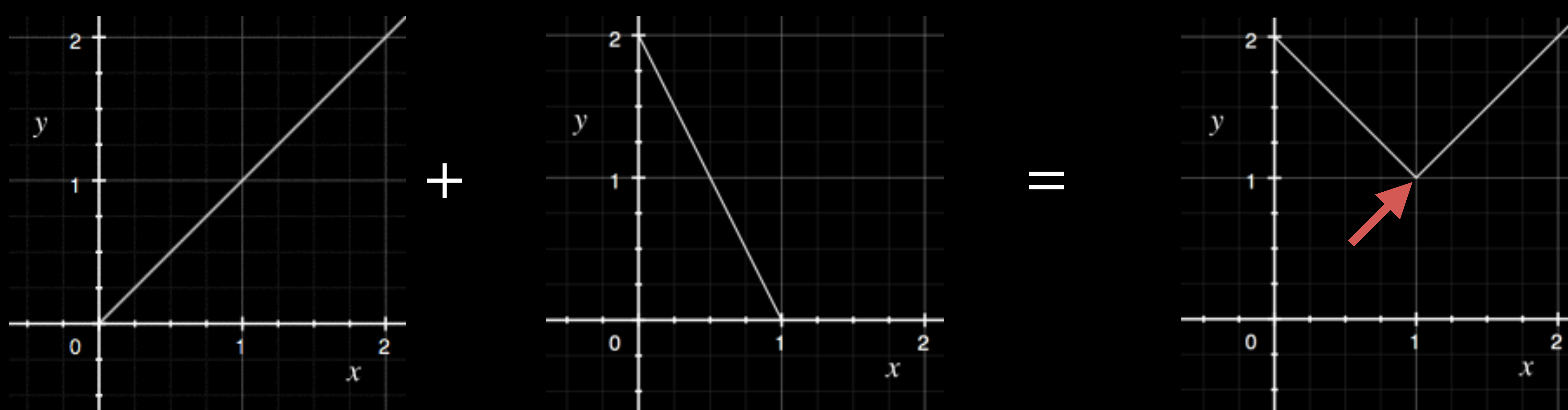
2. Gradient points towards the global minimum

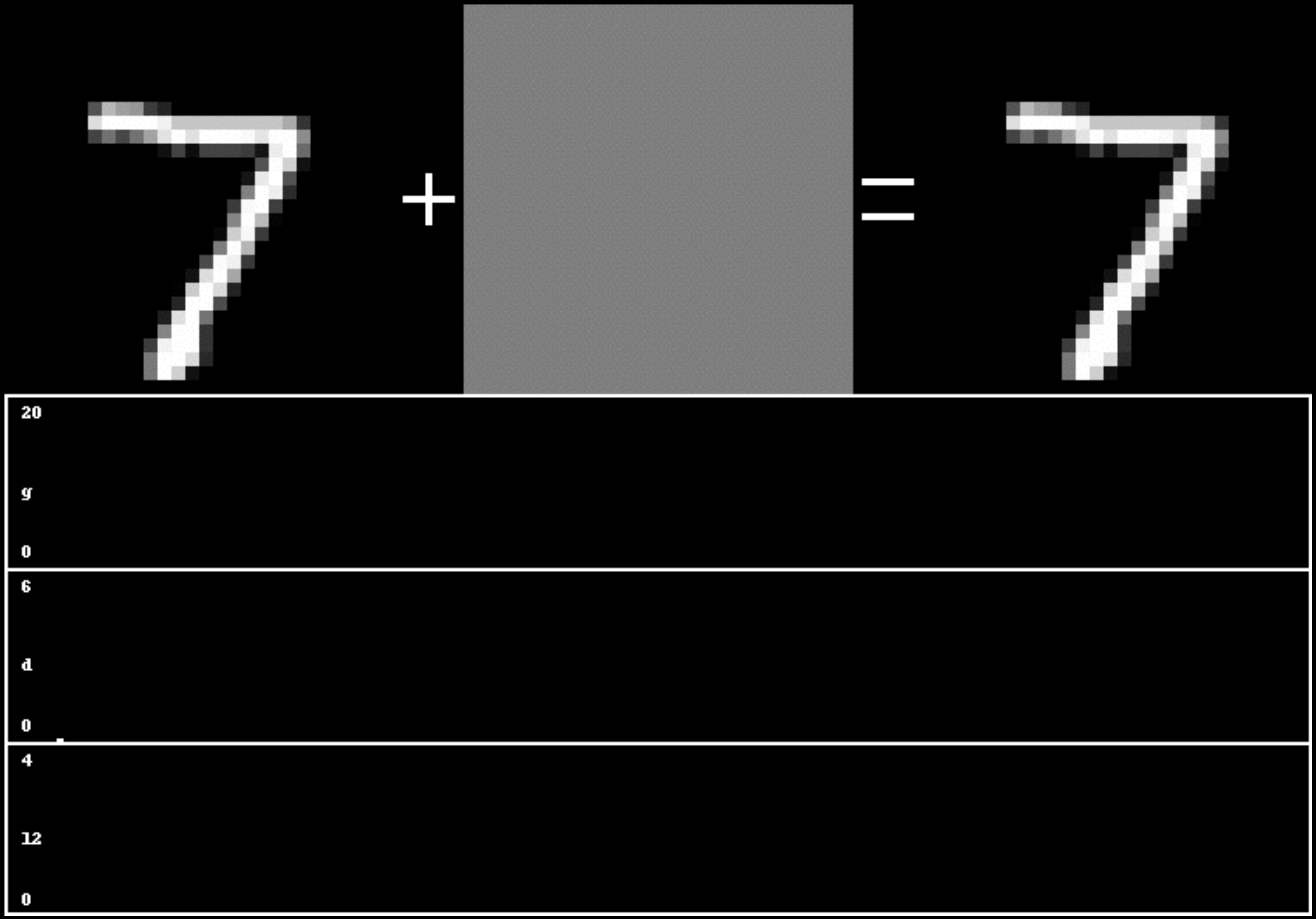
$$\max\left(\max_{t' \neq t}\{\log(F(x)'_t)\} - \log(F(x)_t), 0\right)$$

Improved Formulation

- Formulation:
  minimize $\quad d(x,x') + g(x')$
  such that $\quad x'$ is "valid"

$d(x,x') \quad + \quad g(x')$



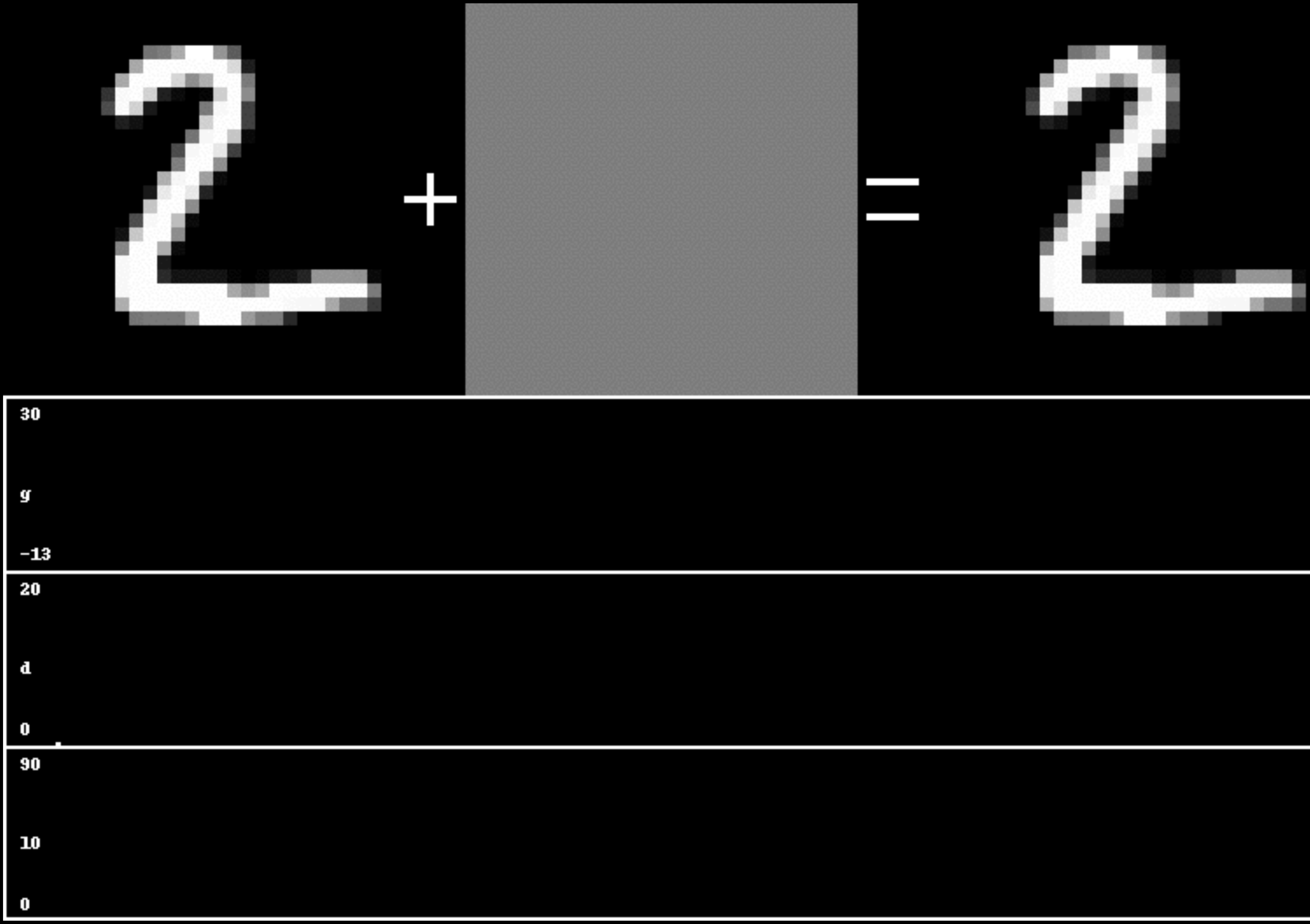

20
g
0
6
d
0
4
12
0

L_0 from L_2

- First attempt:
- minimize $d(x,x') + g(x')$
  such that x' is "valid"
- Where the distance d is the L_0 distance

L_0 from L_2

- Solve the L_2 minimization problem and identify the least changed pixel
- Force that pixel to remain constant
- Re-solve the L_2 minimization problem with that pixel fixed at the initial value
- Repeat, finding the new least-changed pixel



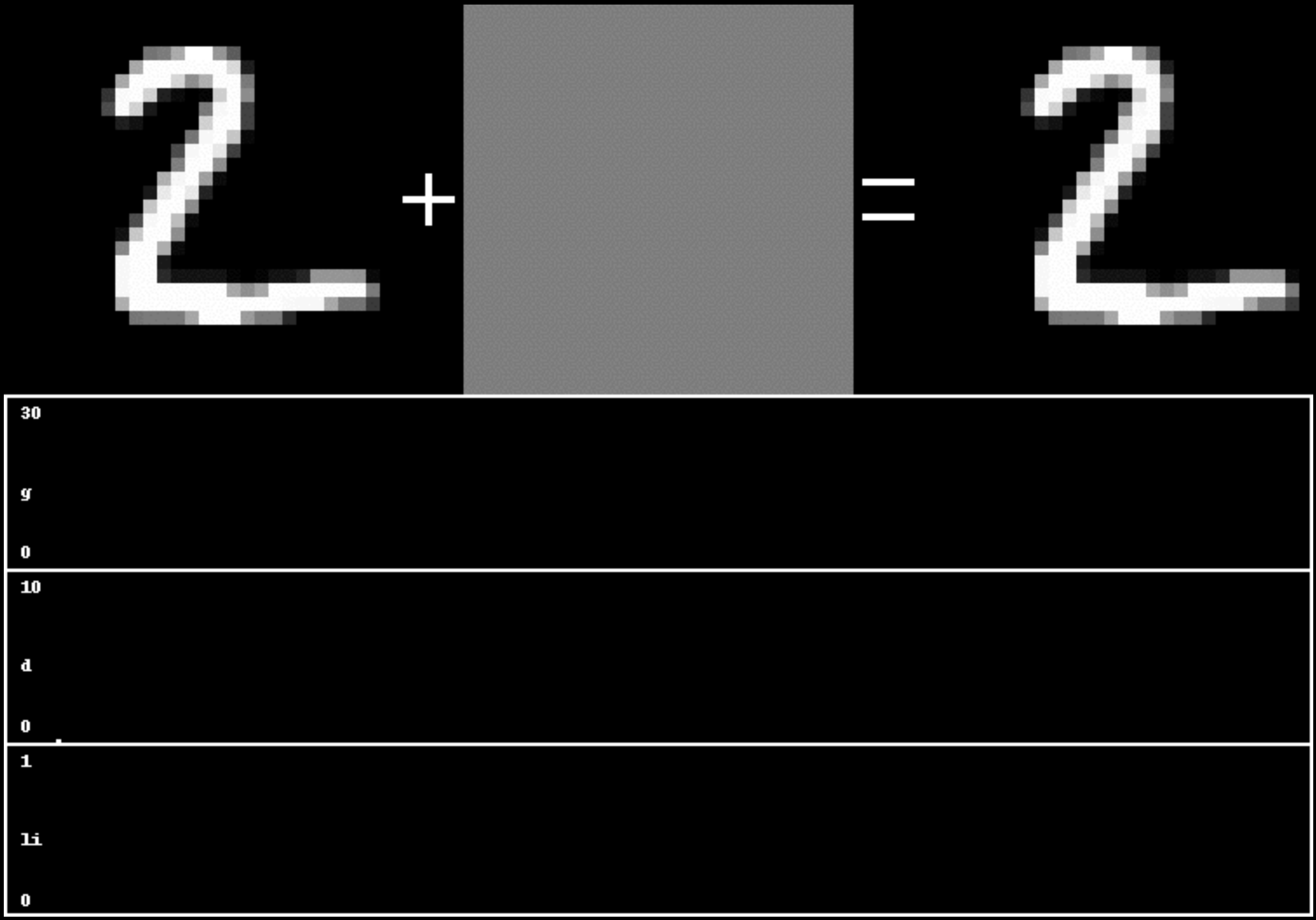
30
g
−13
20
d
0
90
10
0

$L_{infinity}$ from L_2

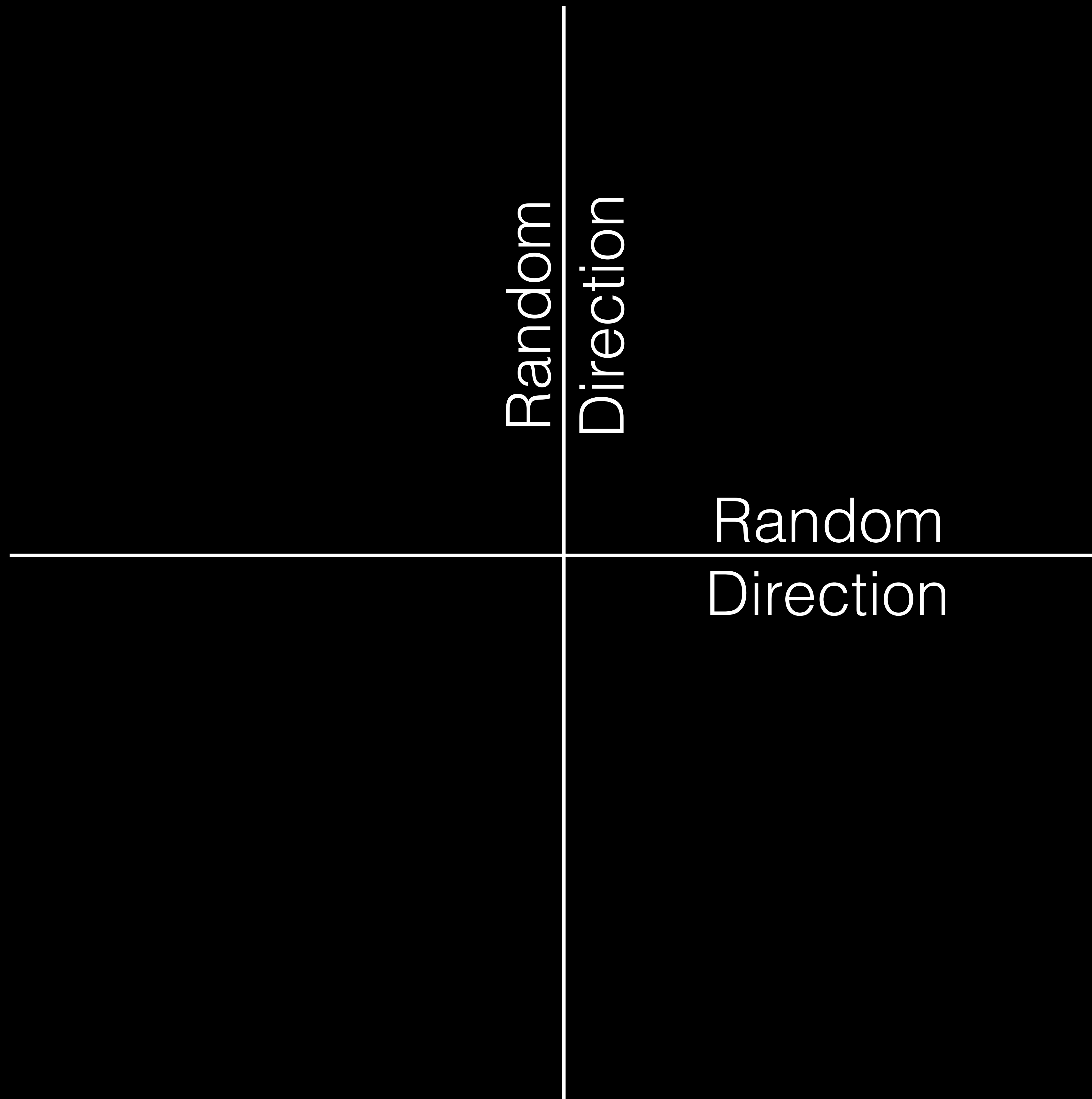
- Formulation:
  minimize $d(x,x') + g(x')$
  such that x is "valid"

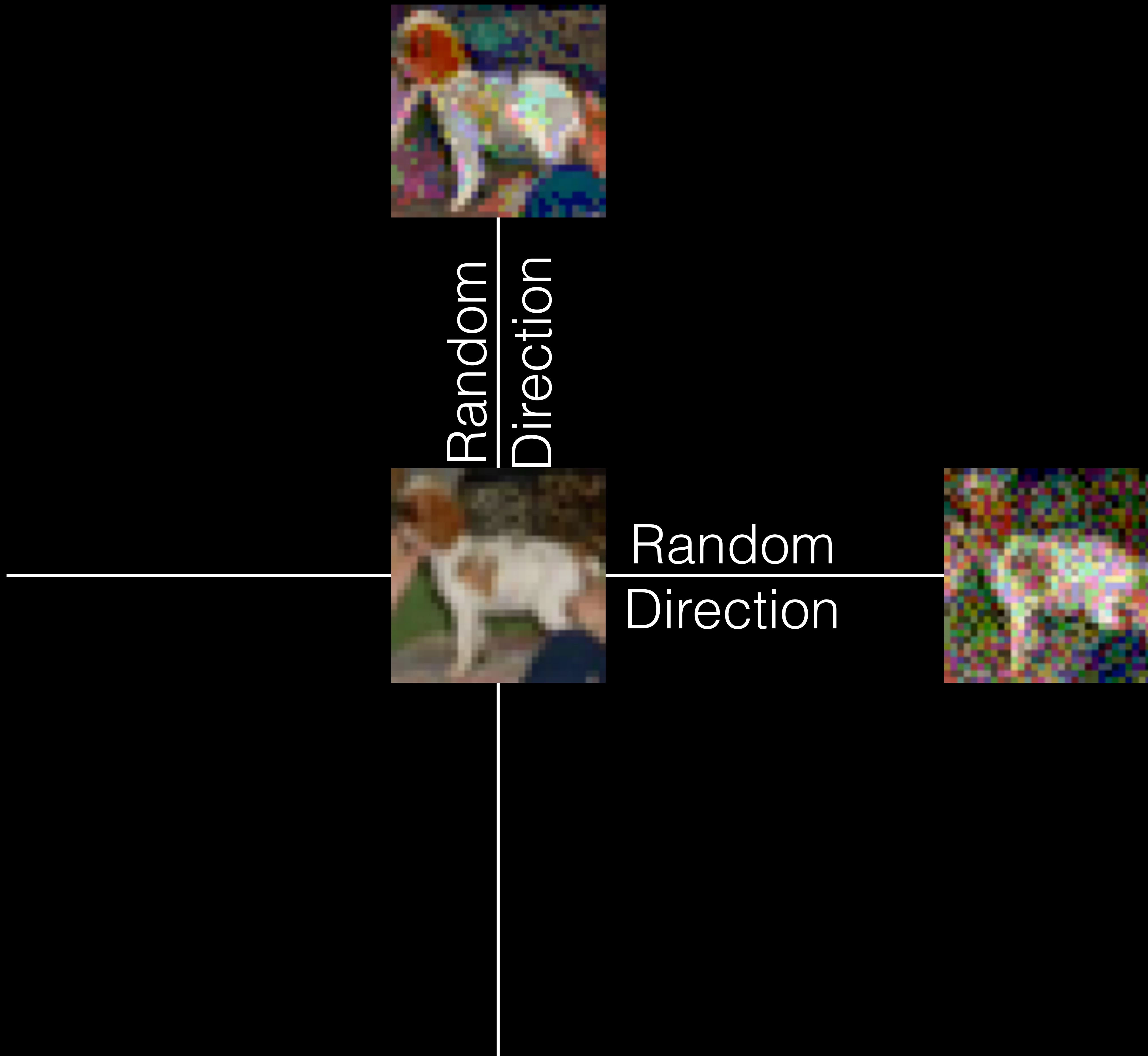
$L_{infinity}$ from L_2

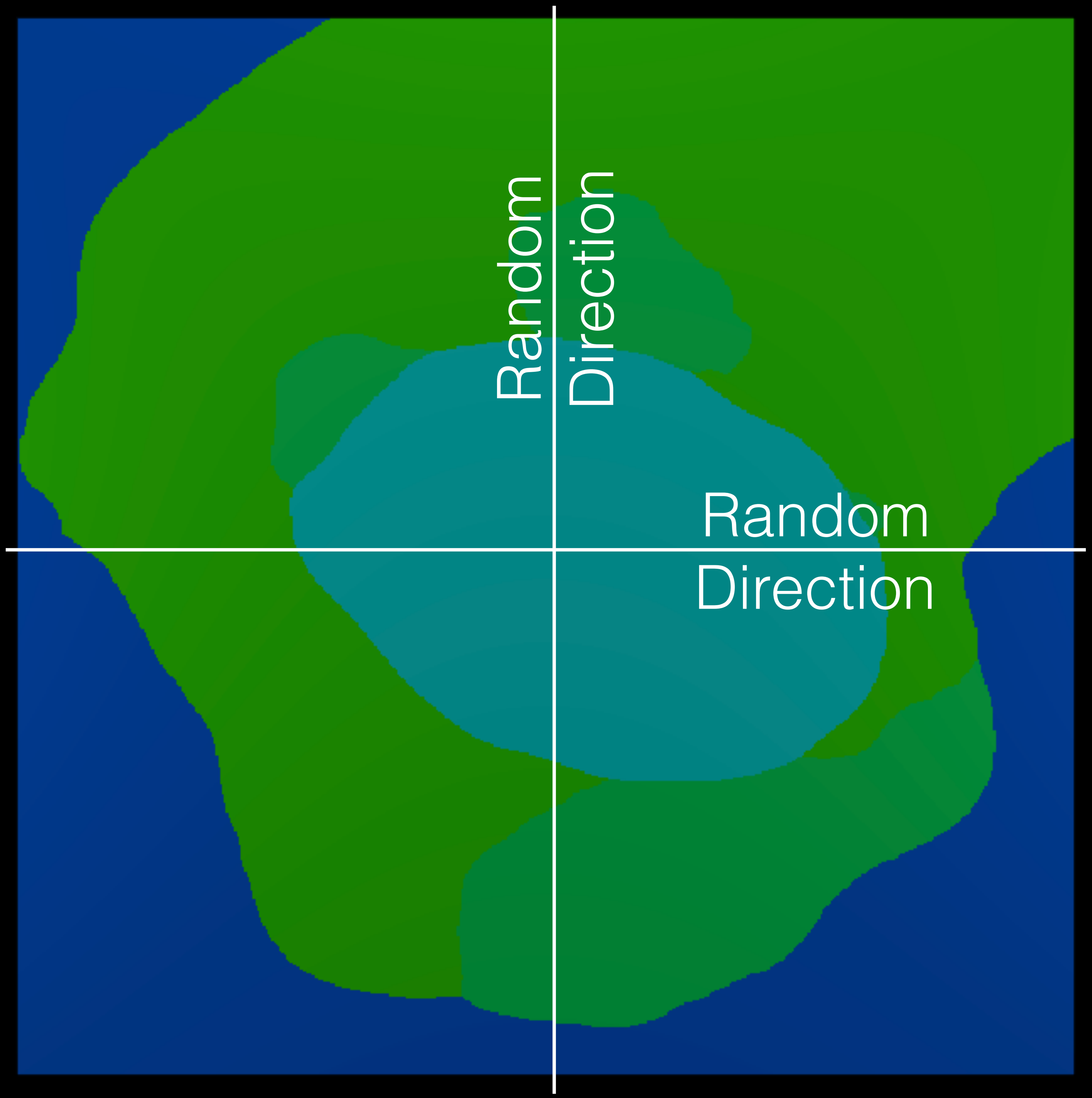
- Initially set a budget Δ=1
- Formulation:
  minimize sum[max(|x_i-x'_i| - Δ, 0)] + g(x')
  such that x is "valid"
- Decrease Δ and solve again

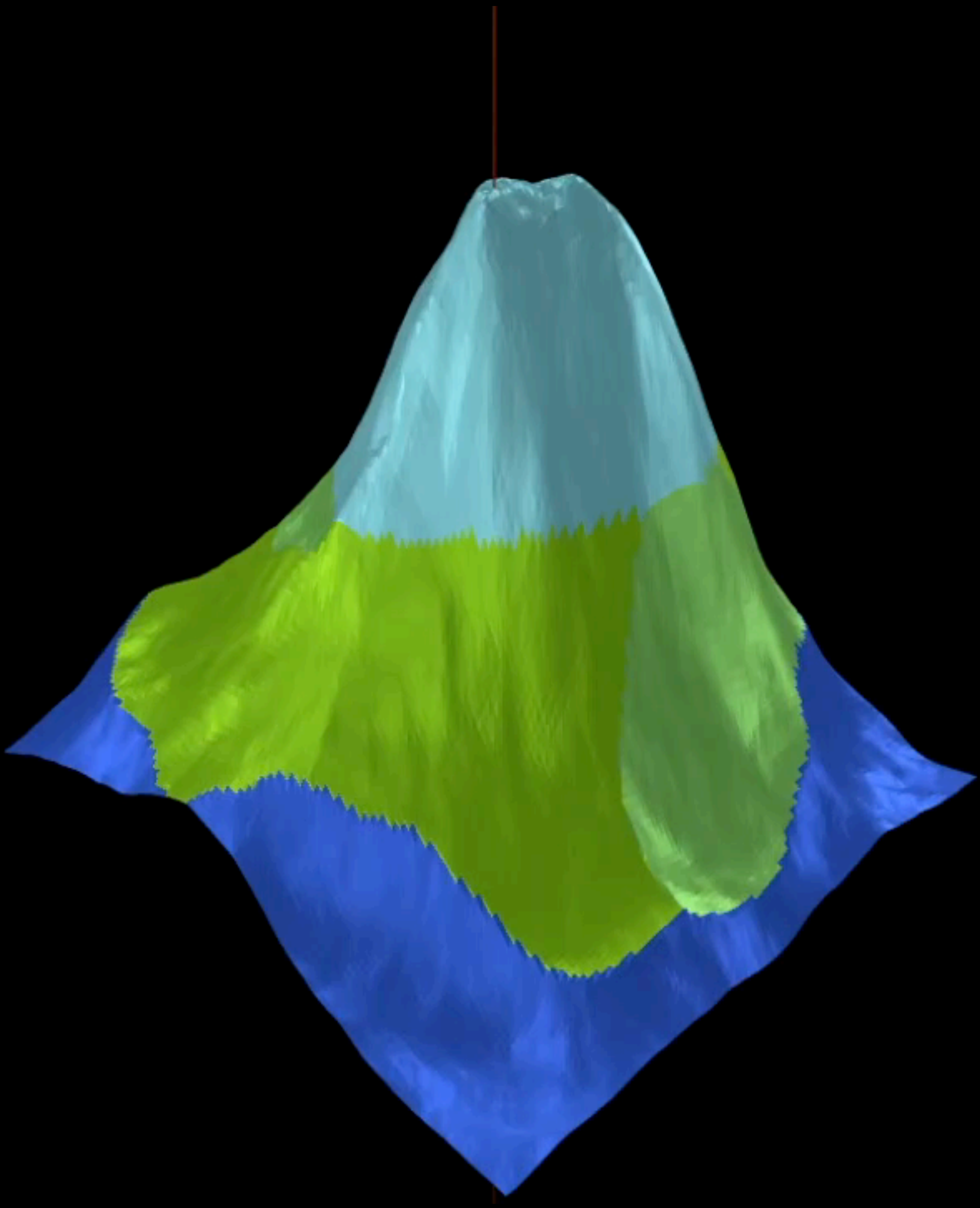

+
=
30
g
0
10
d
0
1
li
0

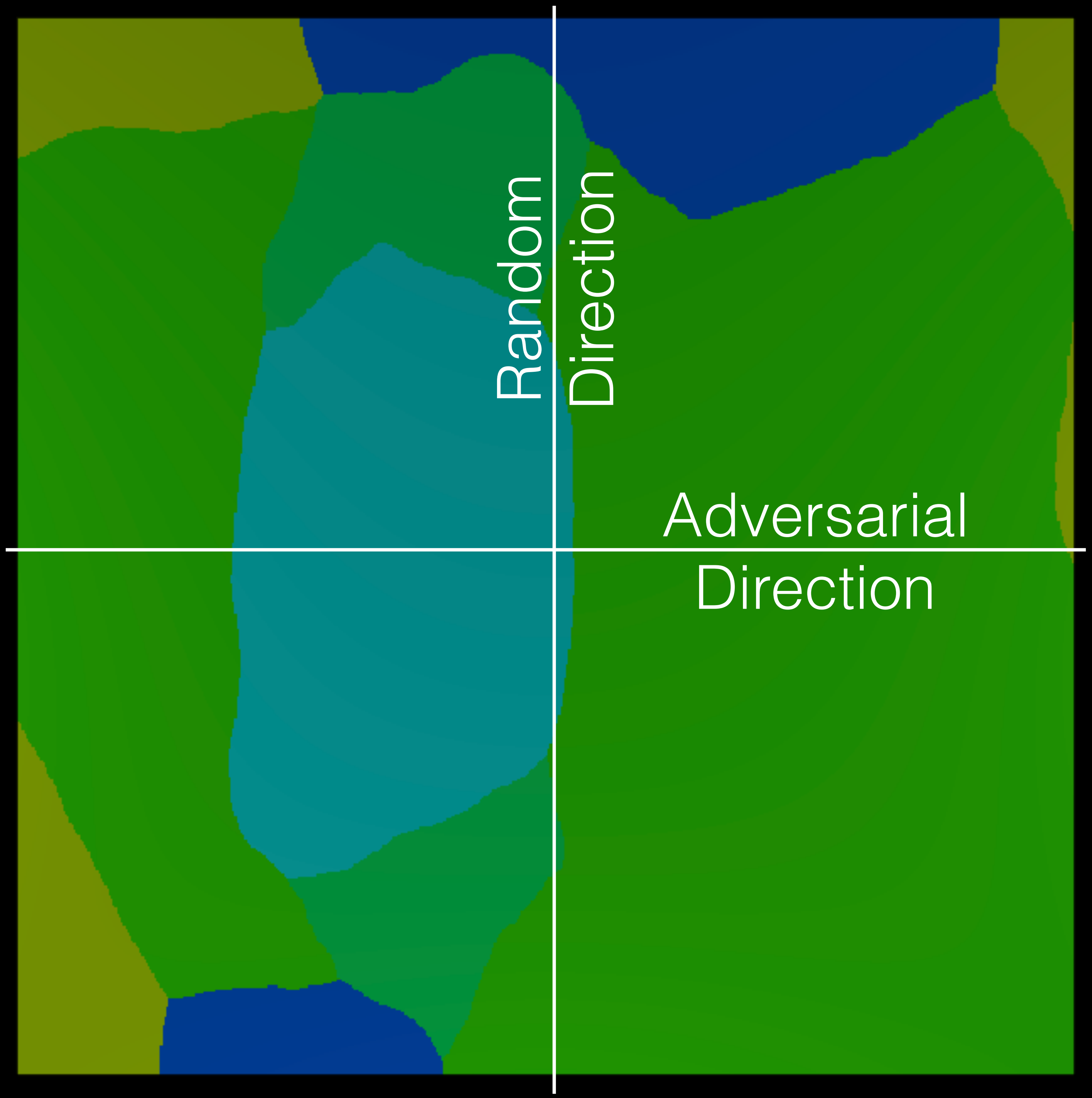
Visualizations

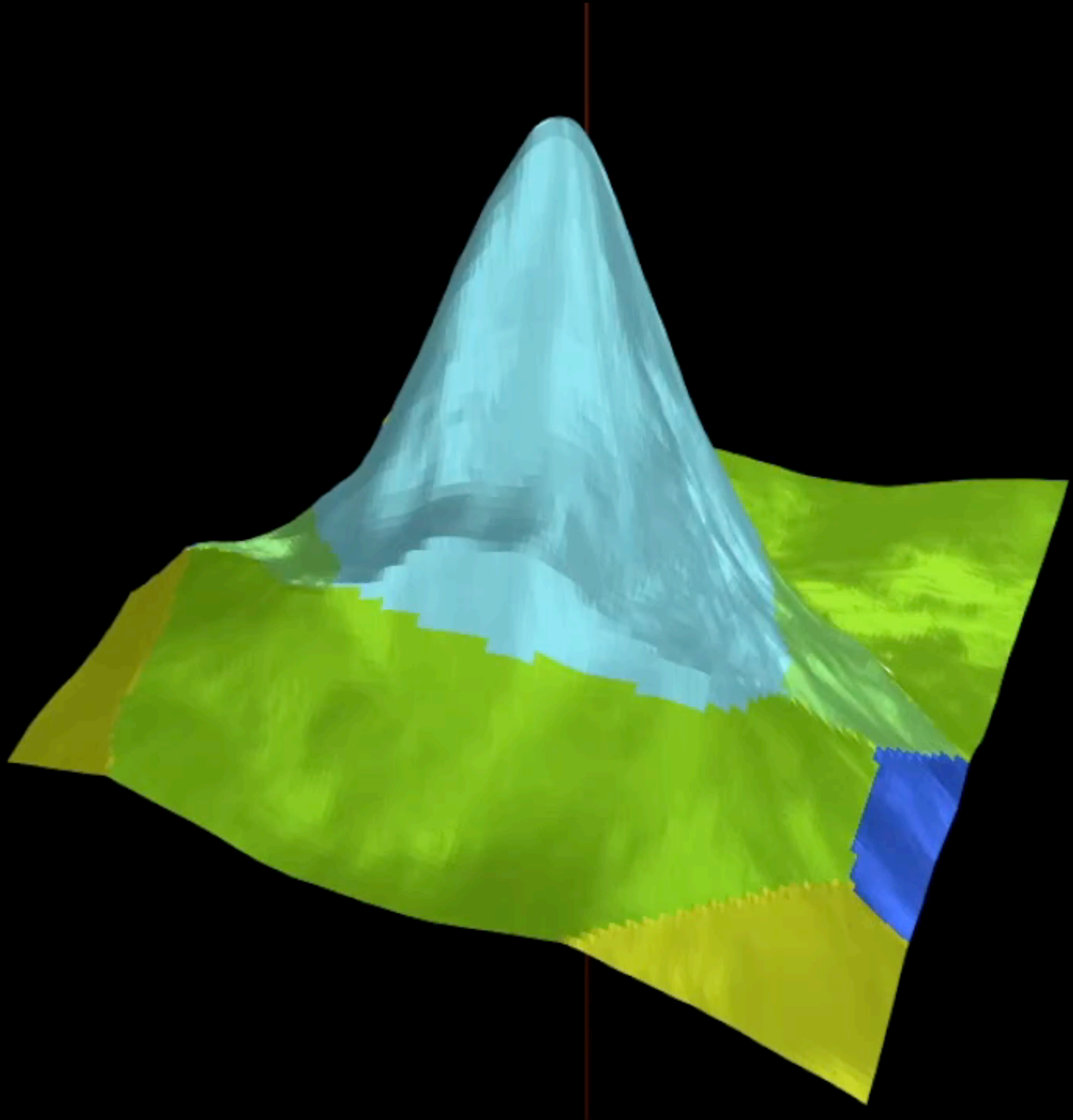

Random Direction
Random Direction


Random Direction
Random Direction


Random Direction
Random Direction




Random Direction
Adversarial Direction



A **defense** is a neural network that

1. Is accurate on the test data
2. Resists adversarial examples

Defense Idea #1: Thermometer Encoding

Claim:
Neural networks don't generalize

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. Towards deep learning models resistant to adversarial attacks. ICLR 2018

Normal Training

( ,7)

( ,3)

Training

Adversarial Training (1)

( ,7)

( ,3)

( ,7)

( ,3)

Attack

Adversarial Training (2)

( ,7)

( ,3)

( ,7)

( ,3)

Training

Defense Idea #2: Thermometer Encoding

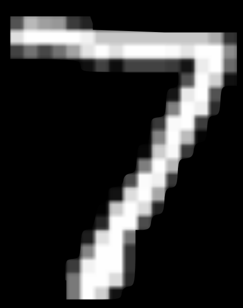
Claim:
Neural Networks are "overly linear"

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. 2018. Thermometer encoding: One hot way to resist adversarial examples. In International Conference on Learning Representations.

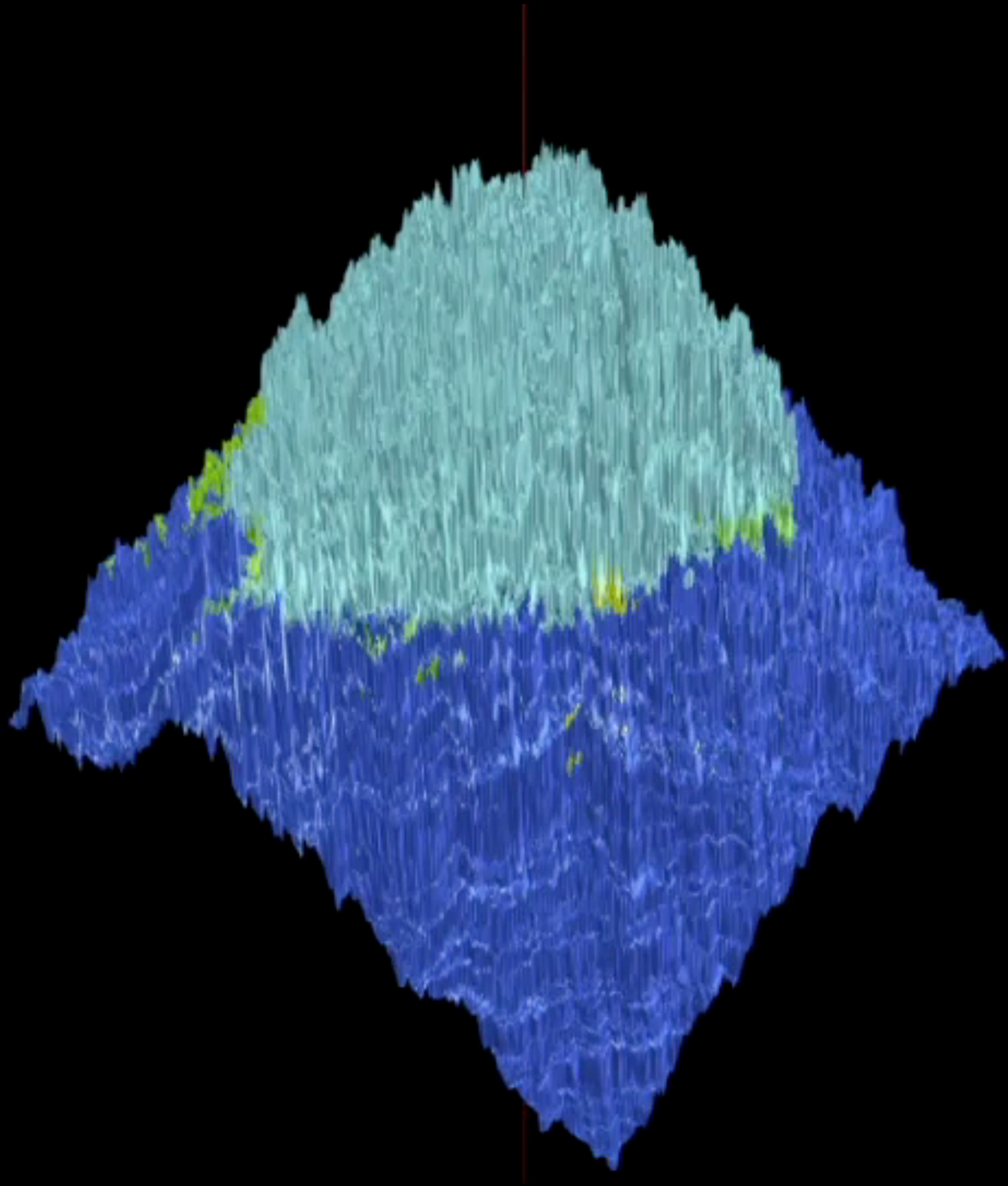
Thermometer Encoding

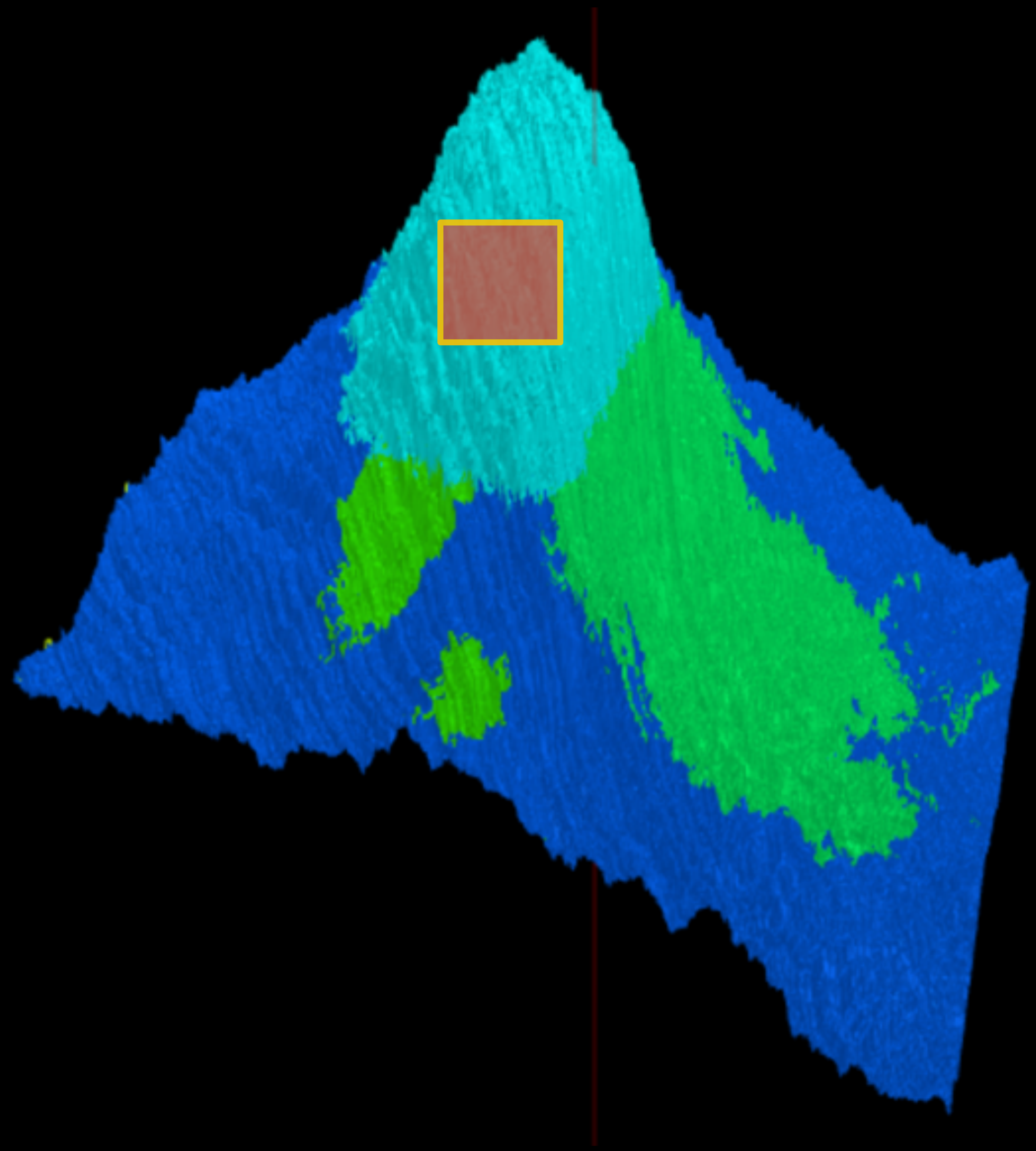
- Break linearity by changing input representation
- T(0.13) = 1 1 0 0 0 0 0 0 0 0
- T(0.66) = 1 1 1 1 1 1 0 0 0 0
- T(0.97) = 1 1 1 1 1 1 1 1 1 1

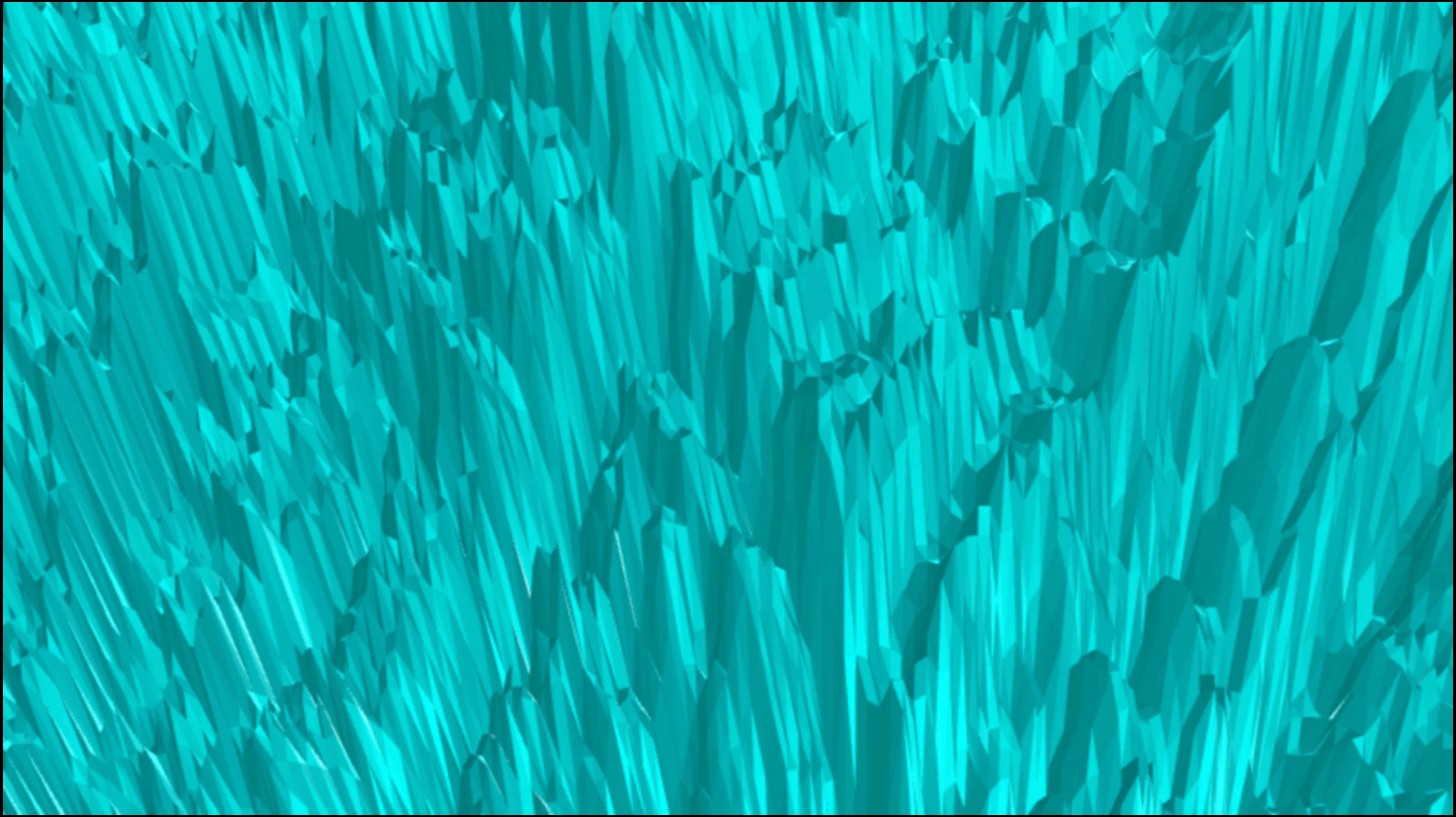
Standard Neural Network

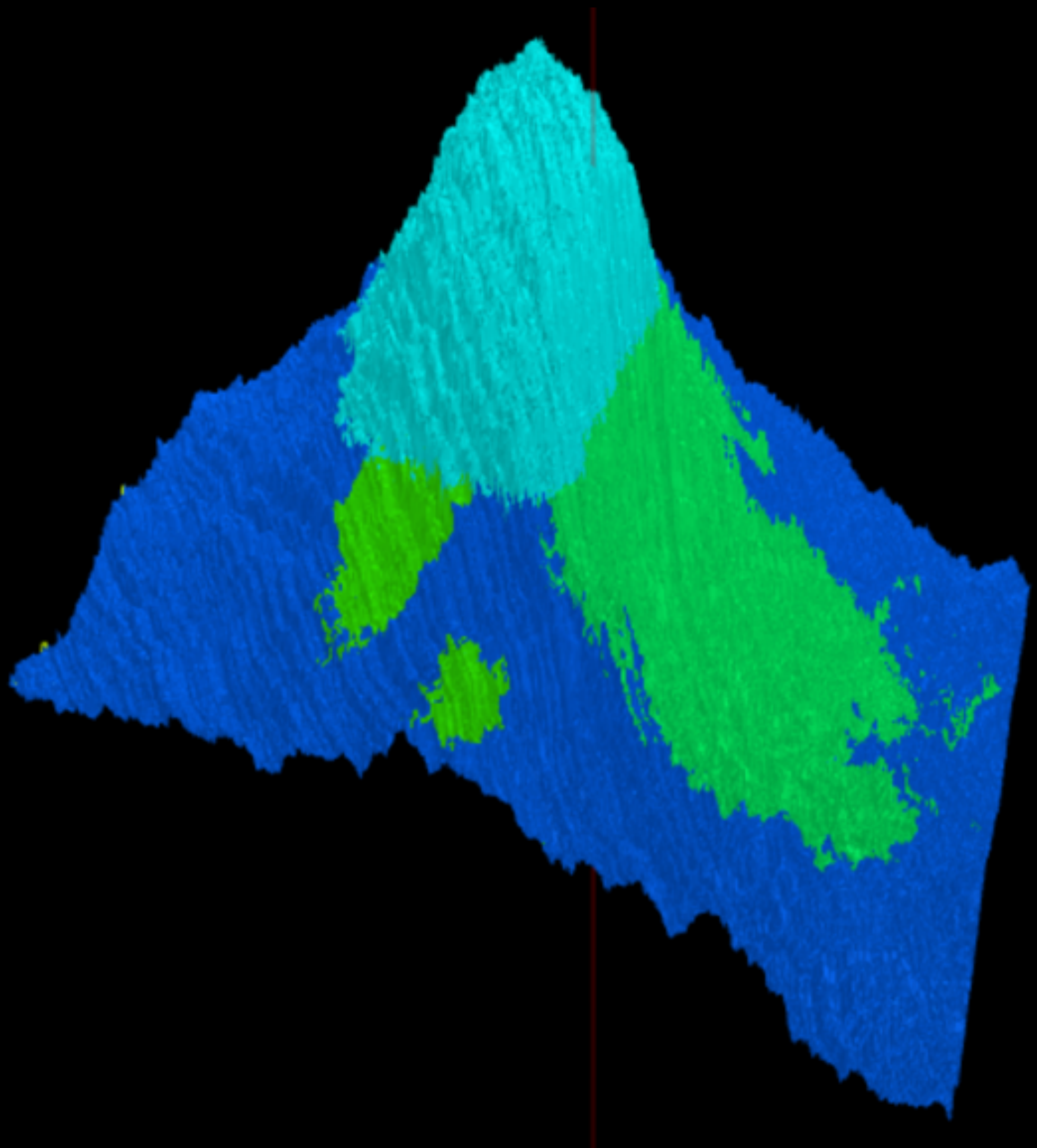


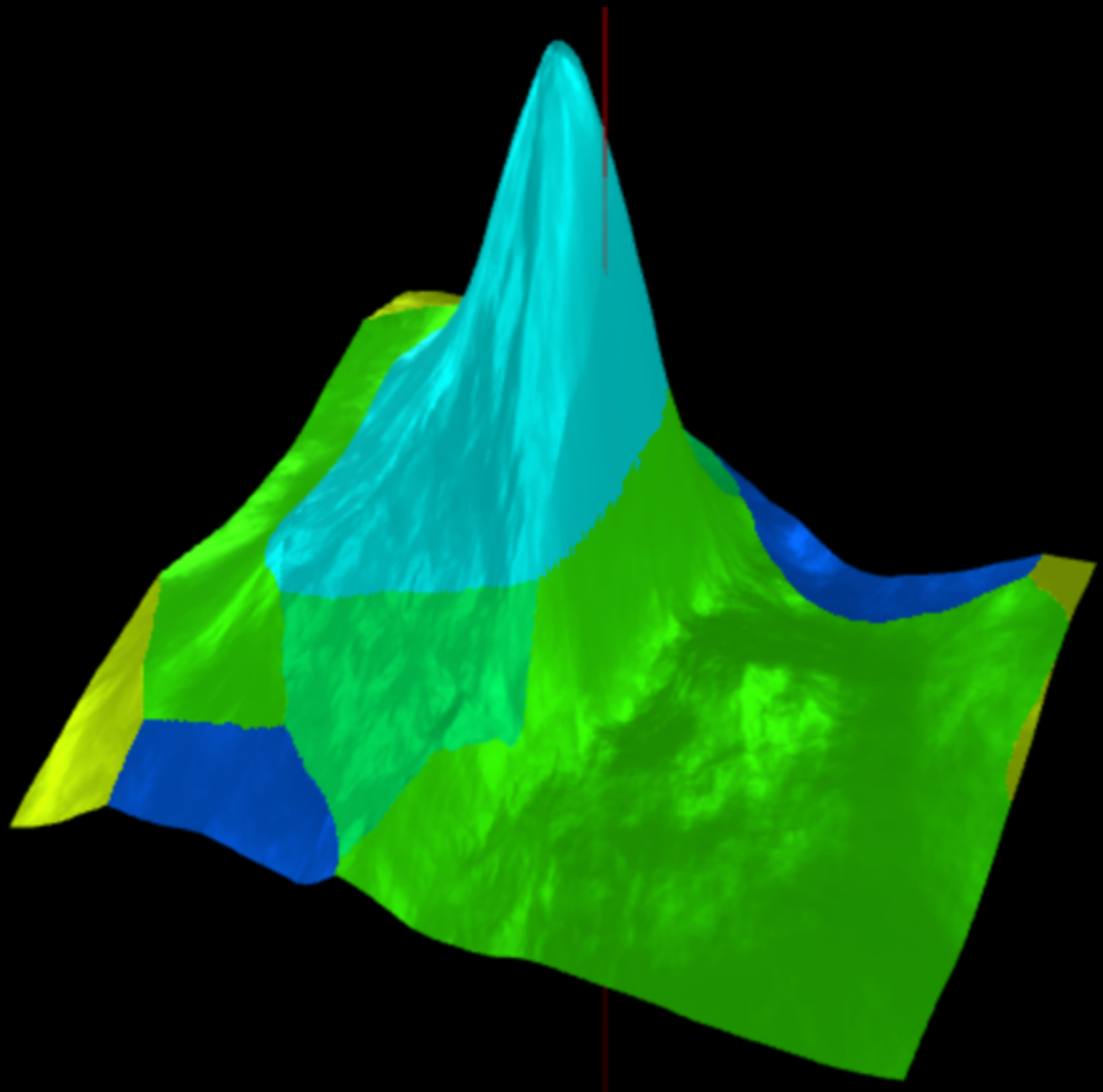
With Thermometer Encoding



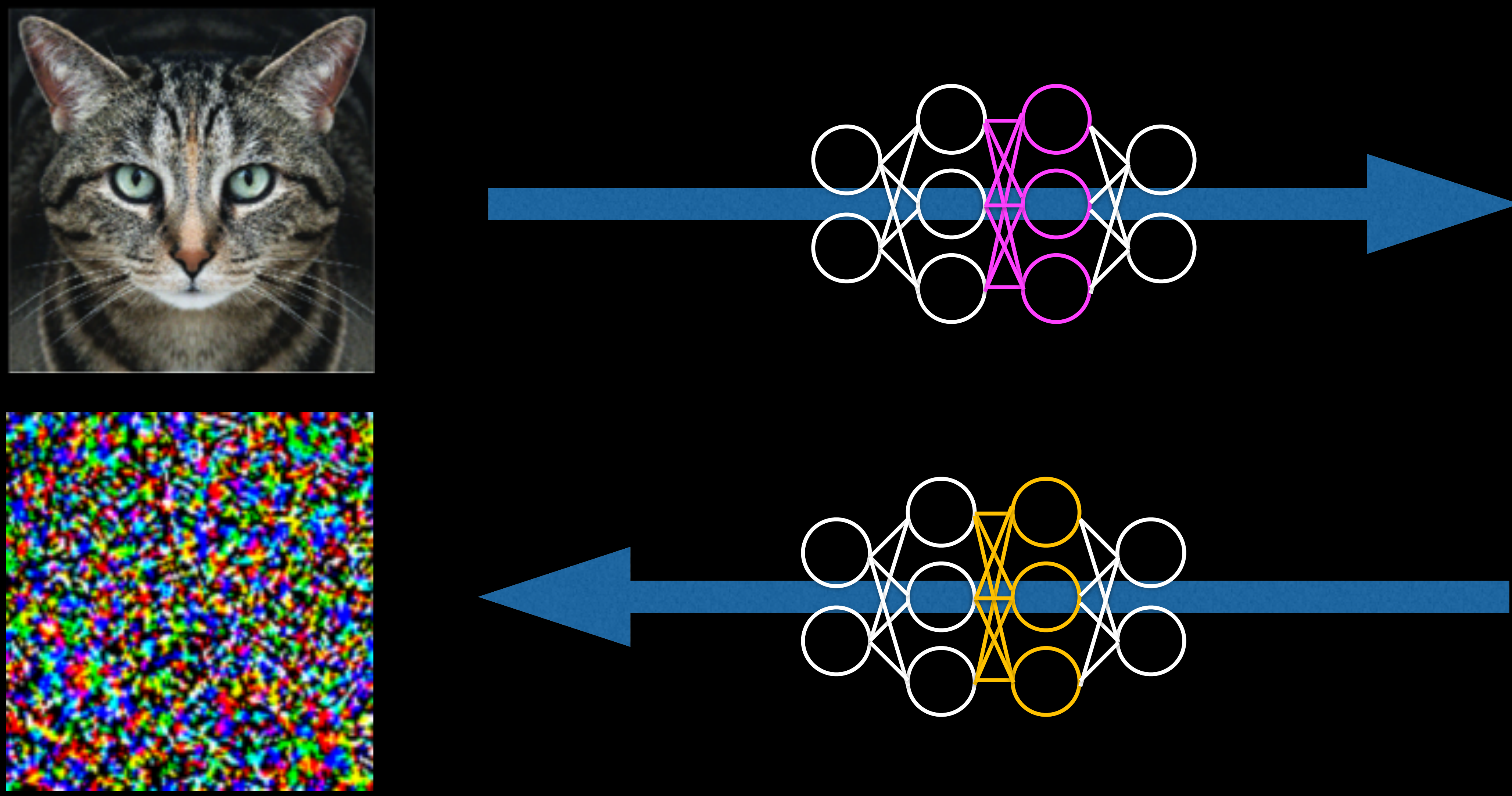






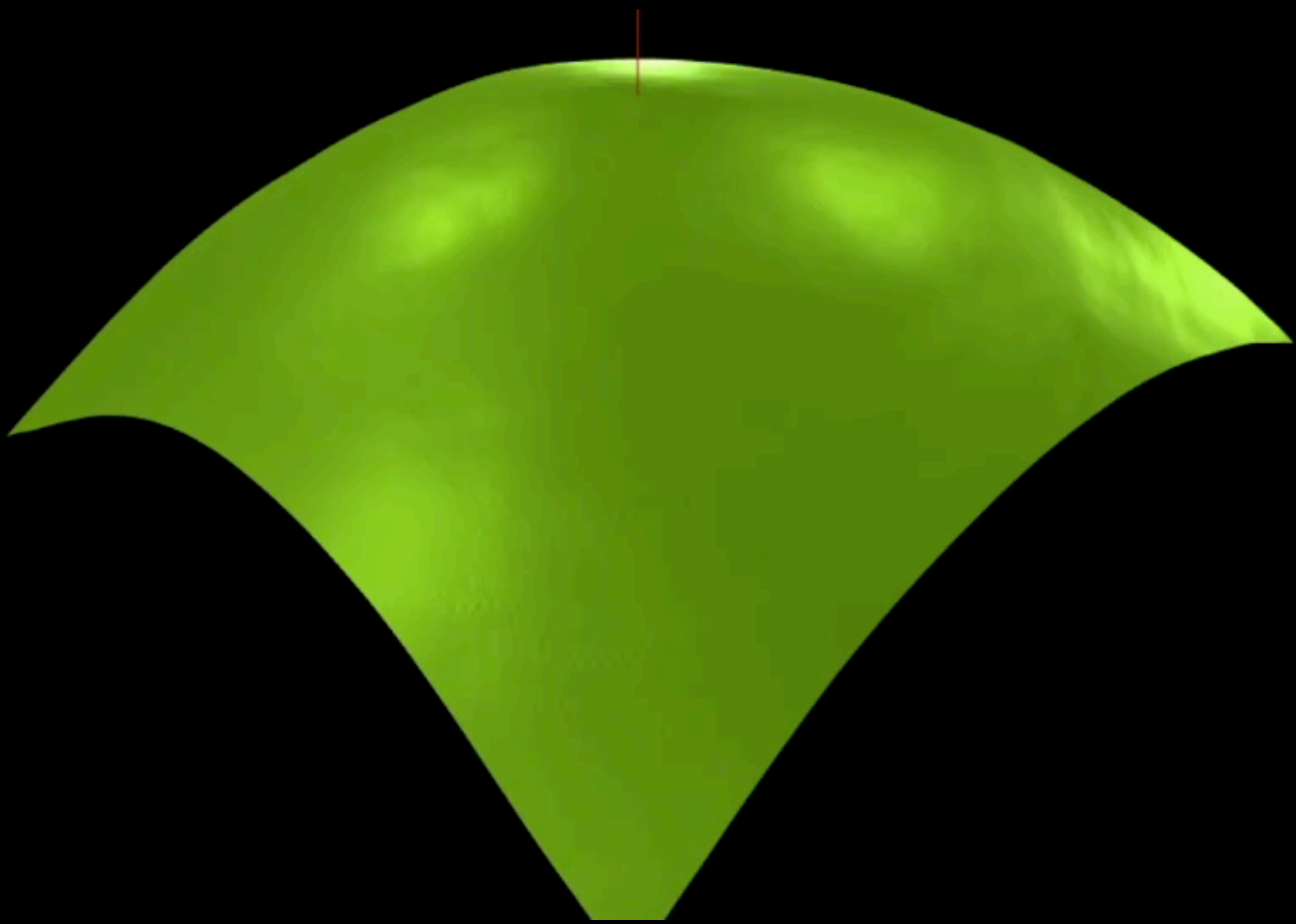


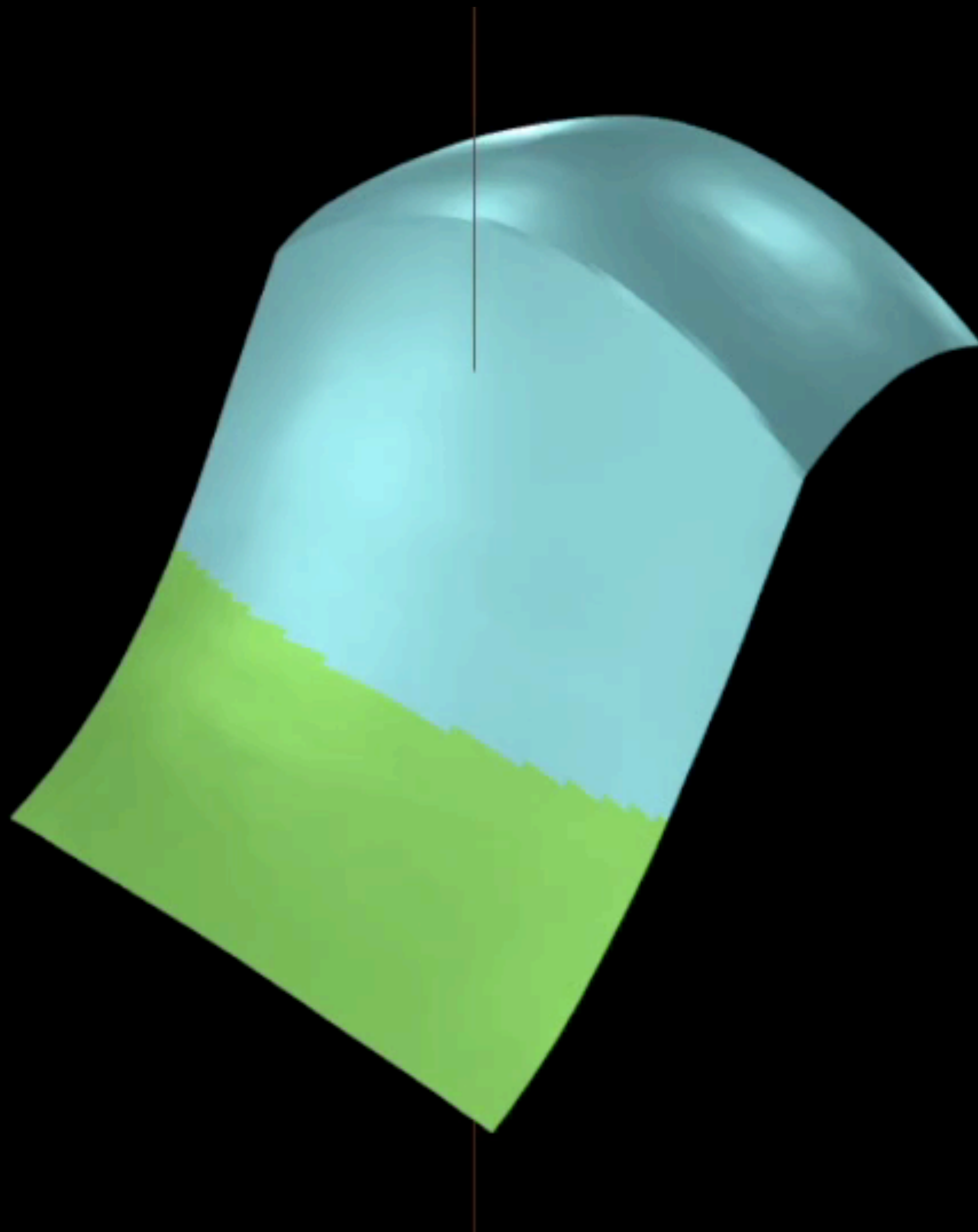
"Fixing" Gradient Descent



[0.1,
0.3,
0.0,
0.2,

What does adversarial training do?





... so that's images

what about other domains?

Audio has these same issues, too

N Carlini and D Wagner. "Audio Adversarial Examples: Targeted Attacks on Speech-to-Text". 2018.

"now I would drift gently
off to dream land"

[adversarial]

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity original or adversarial?

original or adversarial?

On audio, traditional ML methods are not vulnerable to adversarial examples

Questions?

https://nicholas.carlini.com
nicholas@carlini.com