

Surfing: Iterative Optimization Over Incrementally Trained Deep Networks

Ganlin Song, Zhou Fan, John Lafferty
Yale University

Main Idea

We minimize

$$f_{\theta}(x) = \frac{1}{2} \|AG_{\theta}(x) - Ay\|^2 \quad (1)$$

$A \in \mathbb{R}^{m \times n}$ compression matrix ($m \ll n$), $G_{\theta} : \mathbb{R}^k \mapsto \mathbb{R}^n$ generative network.

Goal: Find the minimizer \hat{x} and recover y by $G_{\theta}(\hat{x})$.

PROBLEM:

- $f_{\theta}(x)$ is usually **non-convex**.
- $f_{\theta}(x)$ has nice landscape only when G has **random** parameters.

OUR RESULTS:

- We propose a novel algorithm **surfing** that outperforms regular gradient descent in practice.
- Theoretical analysis is provided to ensure its convergence.

Surfing Algorithm

The landscape of $f_{\theta}(x)$ is nice at the beginning and getting wavy and bumpy as G_{θ} is trained.

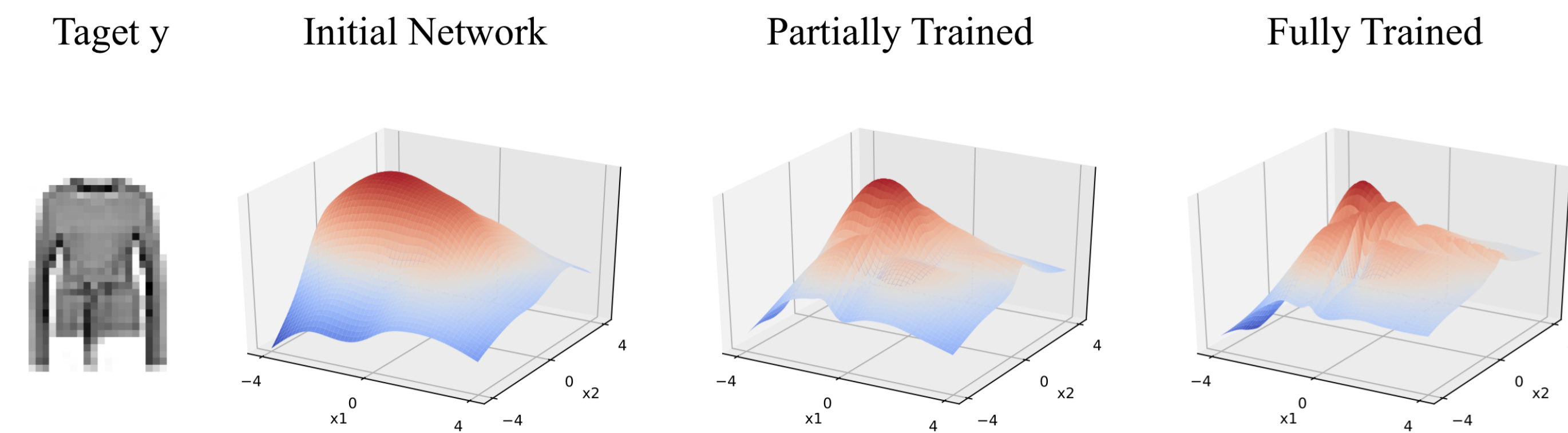


Figure 1: Behavior of the surfaces $x \mapsto -\frac{1}{2} \|G_{\theta}(x) - y\|^2$ for an image y from Fashion-MNIST over three levels of training. The network is trained by VAE.

Motivations:

- Use intermediate networks when training G_{θ}
- Apply gradient descent on each of these networks
- Keep track of the global optimum

Algorithm:

Consider a sequence of network G_0, G_1, \dots, G_T , G_0 has random weights θ_0 and G_T is trained. Write

$$f_t(x) = \frac{1}{2} \|AG_t(x) - Ay\|^2, \quad \forall t \in [T]. \quad (2)$$

First minimize f_0 to obtain the minimizer x_0 , then apply gradient descent on f_t for $t = 1, 2, \dots, T$ successively, **starting from the minimizer x_{t-1} for the previous network**.

Theoretical Results

Consider G with ReLU activation,

$$G(x, \theta) = V \sigma(W_d \dots \sigma(W_2 \sigma(W_1 x + b_1) + b_2) \dots + b_d).$$

Our results:

1. If θ_0 is **Gaussian** and G is **sufficiently expansive**, then w.h.p., all critical points of $f_0(x)$ belong to **a small neighborhood around 0**.

Proof technique:

- G is piecewise linear, gradient of f_0 has explicit expression.
- Concentration bounds give $\nabla f_0(x) = 2^{-d}x + O(\epsilon(1 + \|x\|))$.
- Find v s.t. directional derivative $D_v f_0(x) < 0$ for all $\|x\| > O(\epsilon)$.

2. Consider a network flow $G^s(x) = G(x, \theta(s))$ for $s \in [0, S]$ and corresponding objective $f^s(x) = f(x, \theta(s))$. If the weights of G^s is **bounded** and the global minimizer of f^s is **unique and Lipschitz-continuous**, then the projected-gradient surfing can **keep track of the minimizer** with a small time discretization step δ of G^s .

Projected-gradient surfing:

- Identify all the linear pieces $\{P_1, \dots, P_l\}$ for current G_t that could contain global minimizer of f_t .
- Apply projected gradient descent $x \leftarrow \text{Proj}_P(x - \eta \nabla f_t(x))$ for each P_i .

Experiments

1. $f(x) = \frac{1}{2} \|G(x) - G(x_*)\|^2$. (Invert a network)

Compared with regular gradient descent (ADAM), surfing has

- Higher proportion of convergence
- Comparable computations

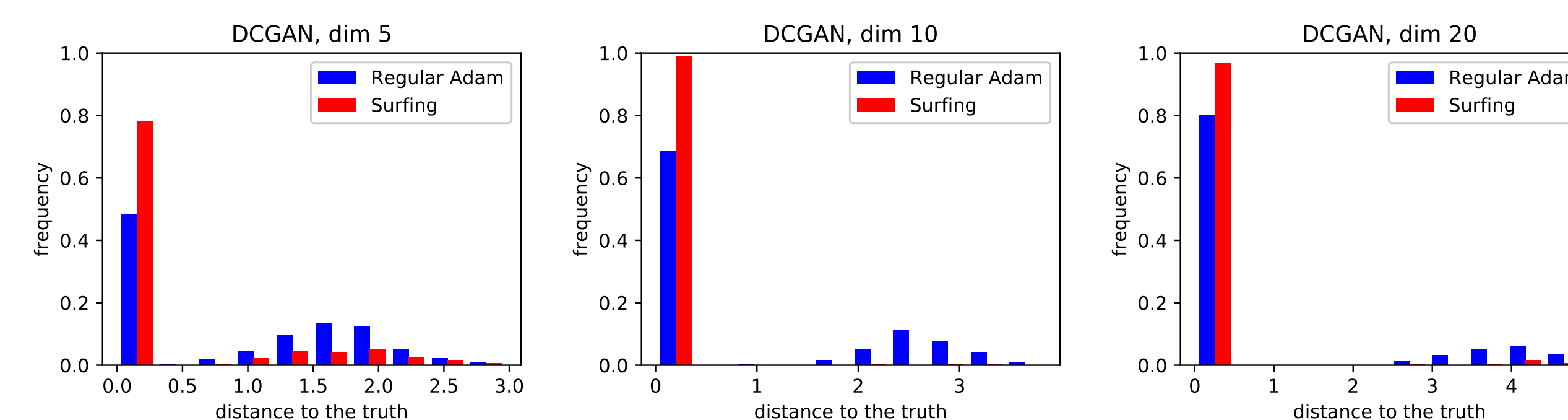


Figure 2: Distribution of distance between solution \hat{x} and the truth x_* .

Input dimension		5	10	20	5	10	20	5	10	20
	Model	DCGAN			WGAN			WGAN-GP		
% successful	Regular Adam	48.3	68.7	80.0	56.0	84.3	90.3	47.0	64.7	64.7
	Surfing	78.3	98.7	96.3	81.7	97.3	99.3	83.7	95.7	97.3
# iterations	Regular Adam	618	4560	18937	464	1227	3702	463	1915	15445
	Surfing	741	6514	33294	547	1450	4986	564	2394	25991

Table 1: Percentages of solutions \hat{x} satisfying $\|\hat{x} - x_*\| < 0.01$.

2. $f(x) = \frac{1}{2} \|AG(x) - AG(x_*)\|^2$. (Compressed sensing, y in the range of G)
Surfing find global optimum more easily than regular GD.

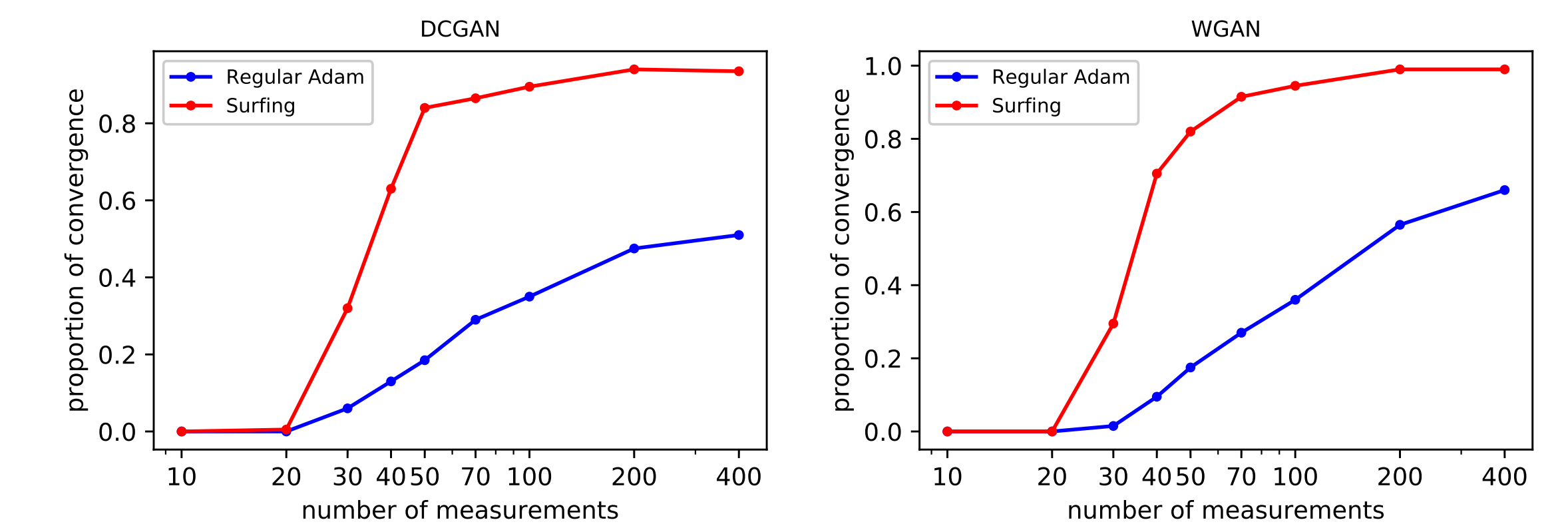


Figure 3: Compressed sensing setting for exact recovery.

3. $f(x) = \frac{1}{2} \|AG(x) - Ay\|^2$, y from test data.

Surfing improves the reconstruction of given signal y .

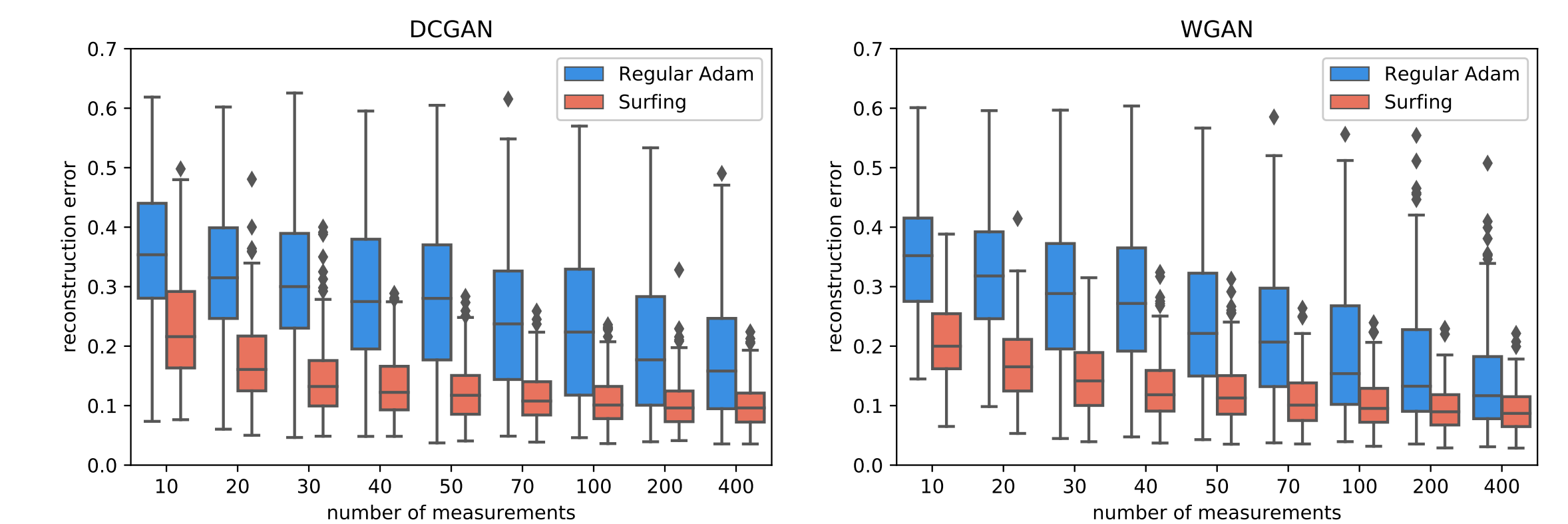


Figure 4: Compressed sensing setting for approximation, or rate-distortion.

The reconstruction error is given by $\sqrt{\frac{1}{n} \|G(\hat{x}) - y\|^2}$.

Future Work

1. **Bridge the gap** between practice and theoretical analysis for surfing.
2. Sometimes, **gradient descent works** on (1) with trained G . How do we understand this?
3. **Regularize the training process** of G so that (1) can always be minimized by simple gradient descent.
4. **Apply idea of surfing to other optimization problems**, where the objective has favorable properties at initial stage and evolves incrementally.