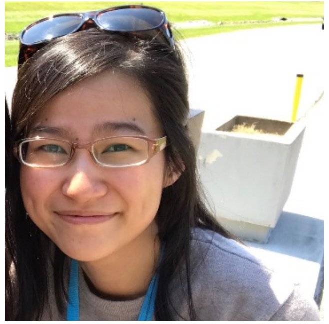
PROVEN:
Verifying Robustness of Neural Networks with a Probabilistic Approach

Tsui-Wei (Lily) Weng[1]

Pin-Yu Chen[2]*, Lam M. Nguyen[2]*, Mark S. Squillante[2]*, Akhilan Boopathy[1], Ivan Oseledets[3], Luca Daniel[1]

MIT[1], IBM Research Yorktown[2], Skoltech[3], alphabetical order*

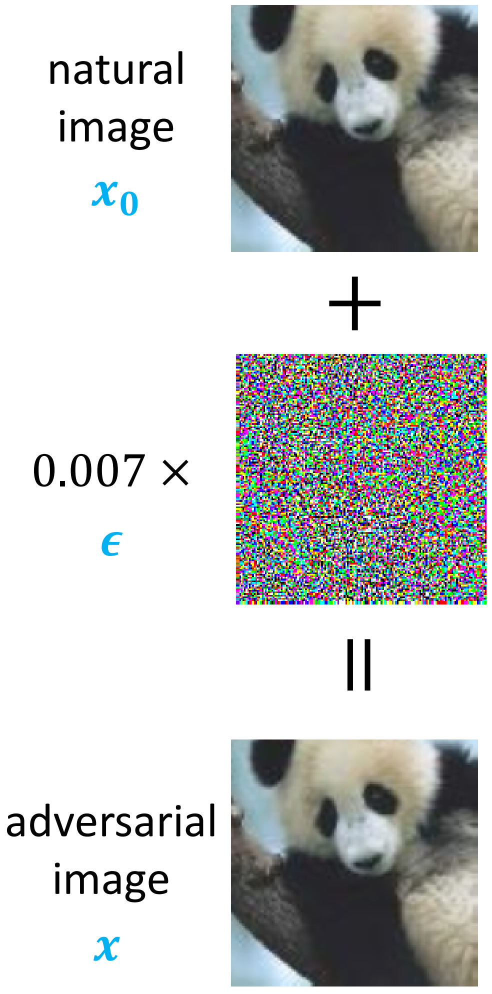
★**Arxiv:** https://arxiv.org/abs/1812.08329 ★**GitHub:** https://github.com/lilyweng/proven







Neural networks are vulnerable to adversarial attacks



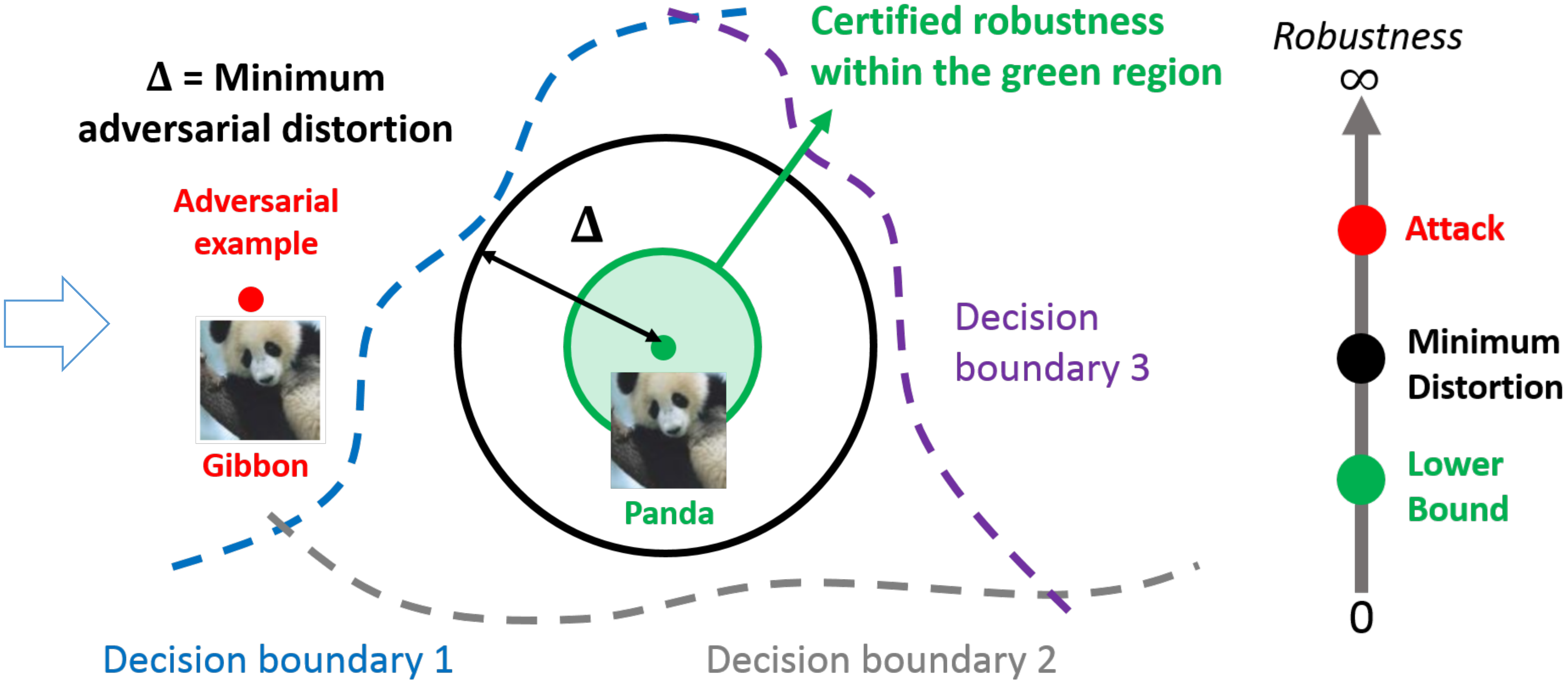


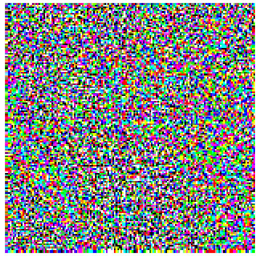
Photo: Goodfellow et al, Explaning and harnessing adversarial examples, ICLR 2015

Existing robustness certification algorithms compute a certified lower bound of min adversarial distortions

natural image x_0

$+$

$0.007 \times \epsilon$



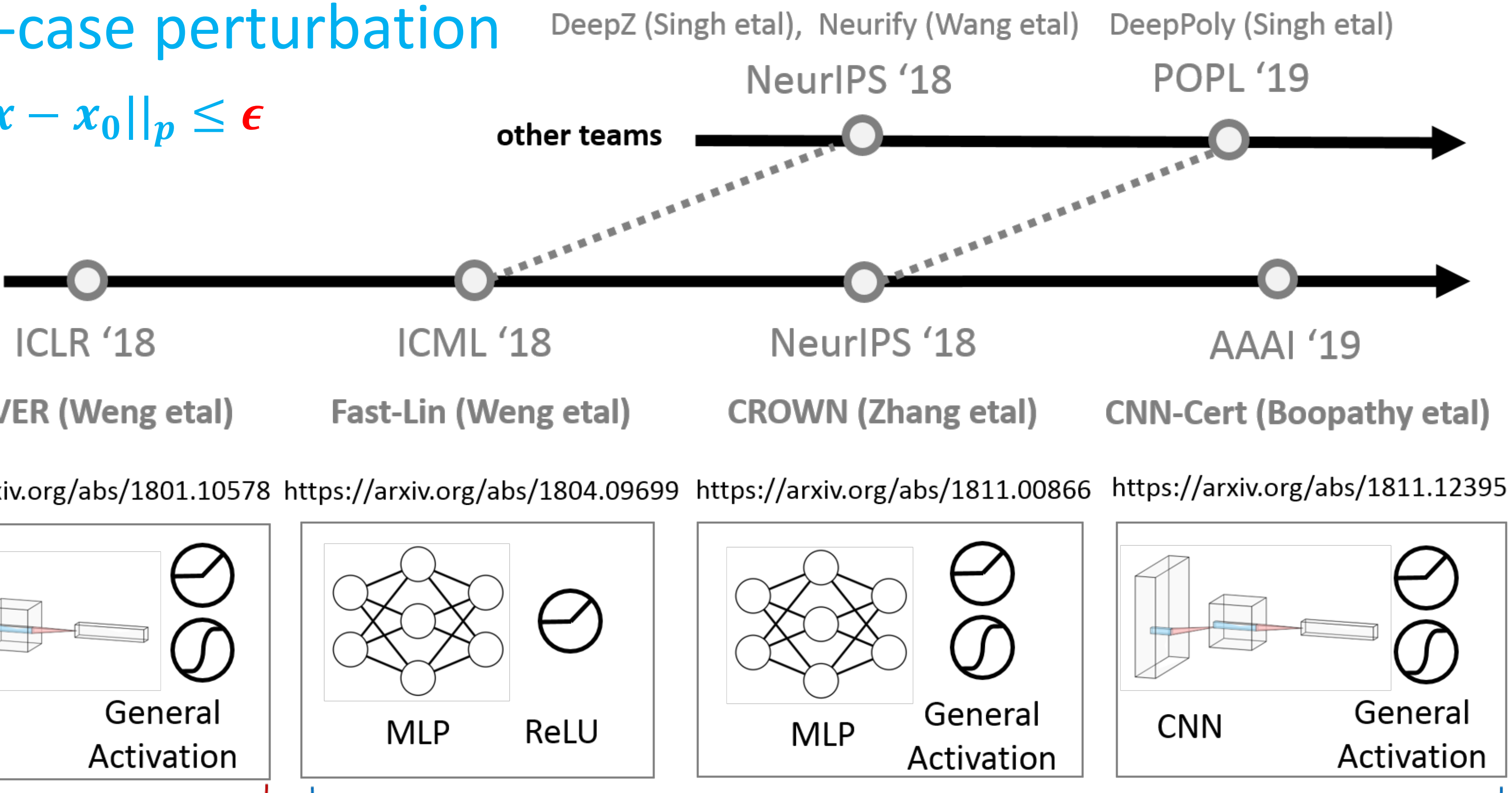
$=$

Do not exist!

adversarial image x

Worst-case perturbation

$$||x - x_0||_p \le \epsilon$$



Robustness Estimation

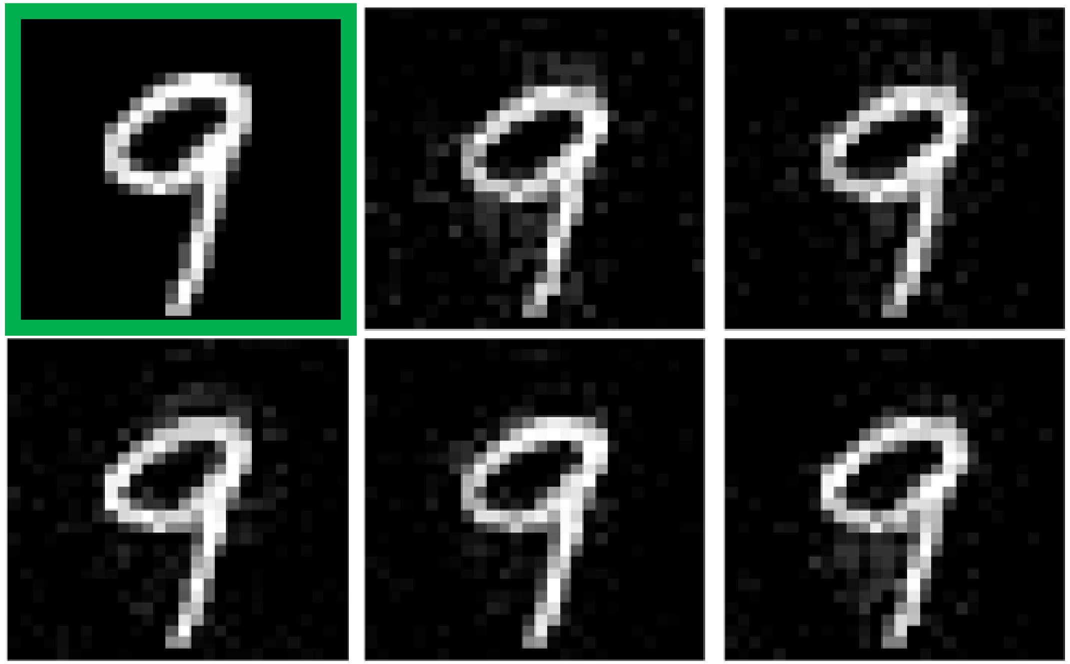
Robustness Certification

Photo: Goodfellow et al, Explaning and harnessing adversarial examples, ICLR 2015

Neural networks are also vulnerable to random noises

LeNet is fooled by Gaussian noises (Bibi etal, CVPR 2018)

True image: 9 | Adv image: 2 | Adv image: 3



Adv image: 4 | Adv image: 7 | Adv image: 8

VGG-F is fooled by uniform noises (Fawzi etal, NIPS 2016)

True image: cauliflower

Adv image: artichoke

Neural networks are also vulnerable to random noises

Attacks with Uniform & Bernoulli noises:

Perturbed ℓ_∞ magnitude	$\epsilon = 0.25$		$\epsilon = 0.20$	
MNIST model	Uniform	Bernoulli	Uniform	Bernoulli
2-layer CNN, ReLU	25%	72%	15%	65%
2-layer CNN, tanh	91%	99%	83%	98%
2-layer CNN, sigmoid	92%	100%	15%	44%
2-layer CNN, arctan	7%	44%	22%	22%
3-layer CNN, ReLU	69%	90%	53%	99%
3-layer CNN, tanh	11%	25%	0%	41%
3-layer CNN, sigmoid	14%	24%	30%	76%
3-layer CNN, arctan	24%	83%	55%	73%
Perturbed ℓ_∞ magnitude	$\epsilon = 0.025$		$\epsilon = 0.020$	
CIFAR model	Uniform	Bernoulli	Uniform	Bernoulli
5×[2048], ReLU	15%	16%	13%	15%
6×[2048], ReLU	17%	20%	14%	20%
5-layer CNN, ReLU	22%	31%	17%	28%

Success rate over randomly selected 100 images can be up to 100%

Existing approaches analyzing neural networks + random noises

Existing works

- Assumptions on locally approximately flat decision boundaries (Franceschi etal, AIstats 2018)
- Assumptions on Gaussian distributed latent input vectors (Fawzi etal, 2018)
- Estimate probability of rare events via Monte Carlo approach (Webb etal, ICLR 2019)

Our goal

Provide a certificate of neural network robustness under random noises

✓ Bounded Subgaussian Noises (e.g. Uniform, Bernoulli)
✓ Gaussian Noises (w/ and w/o Correlations)

Key Idea

Leverage prior robustness certification frameworks (Fast-Lin[1], CROWN[2], CNN-Cert[3]) on adversarial perturbations

[1] Weng etal, "Toward Fast Computation of Certified Robustness for ReLU Networks", ICML'18
[2] Zhang etal, "Efficient Neural Network Robustness Certification with General Activation Functions", NeurIPS'18
[3] Boopathy etal, "CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks", AAAI'19

Worst-case robustness certification algorithms

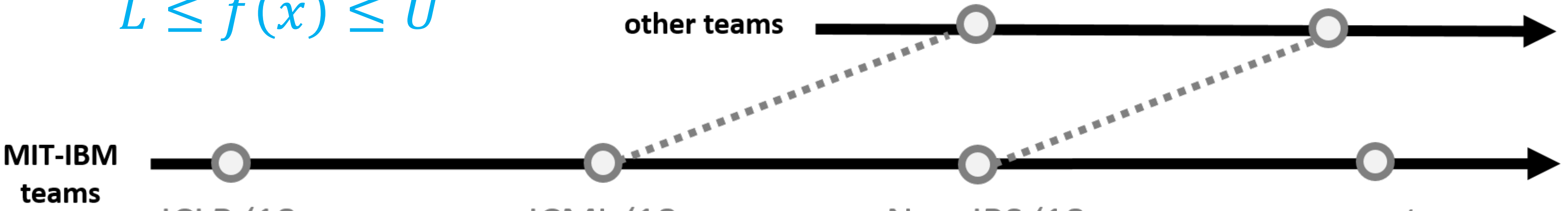
$f(x)$ = NN, and x_0= Original image, x= Perturbed image, $\|x - x_0\| \le \varepsilon$

Worst-case perturbation

$$L \le f(x) \le U$$

DeepZ (Singh etal), Neurify (Wang etal) DeepPoly (Singh etal)

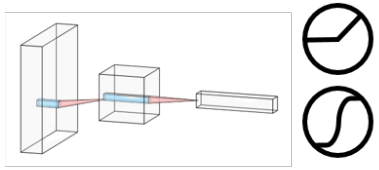
NeurIPS '18 POPL '19



ICLR '18 ICML '18 NeurIPS '18 AAAI '19

CLEVER (Weng etal) **Fast-Lin (Weng etal)** **CROWN (Zhang etal)** **CNN-Cert (Boopathy etal)**

https://arxiv.org/abs/1801.10578 https://arxiv.org/abs/1804.09699 https://arxiv.org/abs/1811.00866 https://arxiv.org/abs/1811.12395



CNN General Activation

$$L = Ax + B_L$$
$$U = Ax + B_U$$

MLP ReLU

$$L = A_L x + B_L$$
$$U = A_U x + B_U$$

MLP General Activation

$$L = A_L * x + B_L$$
$$U = A_U * x + B_U$$

CNN General Activation

Robustness Estimation

Robustness Certification

Our proposal: PRObabilistically VErify NN robustness

$f(x)$ = NN, and x_0= Original image, x= Perturbed image, $\|x - x_0\| \le \varepsilon$

Worst-case perturbation

$$L \le f(x) \le U$$

DeepZ (Singh etal), Neurify (Wang etal) — NeurIPS '18

DeepPoly (Singh etal) — POPL '19

other teams

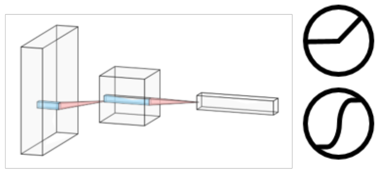
MIT-IBM teams

ICLR '18 — **CLEVER (Weng etal)** — https://arxiv.org/abs/1801.10578

ICML '18 — **Fast-Lin (Weng etal)** — https://arxiv.org/abs/1804.09699

NeurIPS '18 — **CROWN (Zhang etal)** — https://arxiv.org/abs/1811.00866

AAAI '19 — **CNN-Cert (Boopathy etal)** — https://arxiv.org/abs/1811.12395



CNN General Activation

$$L = Ax + B_L$$
$$U = Ax + B_U$$

MLP ReLU

$$L = A_L x + B_L$$
$$U = A_U x + B_U$$

MLP General Activation

$$L = A_L * x + B_L$$
$$U = A_U * x + B_U$$

CNN General Activation

Robustness Estimation

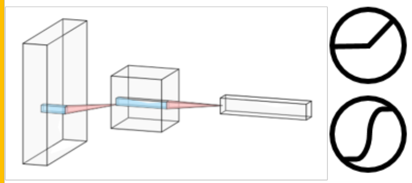
Robustness Certification

Random noises

$$X - x_0 \sim D_\epsilon$$

ICML '19

PROVEN



CNN General Activation

Probabilistic Robustness Certification

PROVEN bounds the probability of NN output

$f(x)$ = NN, and x_0= Original image, x= Perturbed image, $\|x - x_0\| \le \varepsilon$

PROVEN: $\underline{P[L > a]} \le P[f(X) > a] \le \underline{P[U > a]}$

Lower bound on the probability

Upper bound on the probability

$$X - x_0 \sim D_\varepsilon, a \in R, L = A_L * X + B_L, U = A_U * X + B_U$$

To find $P[L > a]$ & $P[U > a]$:

Case (I): X_i independent

(a) direct convolution

(b) probabilistic inequalities

$$\text{Lower bound} \ge \begin{cases} 1 - \exp\left(-\frac{(\mu_L - a)^2}{2\epsilon^2 \|A_{t,:}^L\|_2^2}\right) & , otherwise \\ 0 & , if\ \mu_L - a \ge 0 \end{cases}$$

Case (II): X is multivariate Gaussian

$$\text{Lower bound} \approx \frac{1}{2} - \frac{1}{2} erf\left(\frac{a - \mu_L}{\sigma_L \sqrt{2}}\right)$$

$$\text{Upper bound} \approx \frac{1}{2} - \frac{1}{2} erf\left(\frac{a - \mu_U}{\sigma_U \sqrt{2}}\right)$$

Experiment results

- We compute the robustness lower bound ϵ with various confidence for
  - Input noises: bounded SubGaussian noises and Gaussian noises
  - Networks: various MLP, CNN architectures/activations
  - Training method: standard/adversarial training

- We observed the following interesting results
  - Compared to the worst-case certified lower bound (with 100% provable guarantees), the lower bound with provable 99.99% confidence level can be much larger
    - up to 3.5×-5.4× larger for standard networks, and up to 7× larger for robust networks
  - With better (tighter) robustness certification algorithms, the robustness lower bound is also larger
    - up to 1.3× larger

Conclusion

1) PROVEN is **general**
it compute robustness of general convolutional neural networks with certified probability when input perturbations are random noises

2) PROVEN is **efficient**
it builds on top of existing robustness certification framework (Fast-Lin, CROWN, CNN-Cert) with little overhead

Questions? Come to Tuesday poster #70!

★**Paper:** http://proceedings.mlr.press/v97/weng19a.html, ★**GitHub:** https://github.com/lilyweng/proven