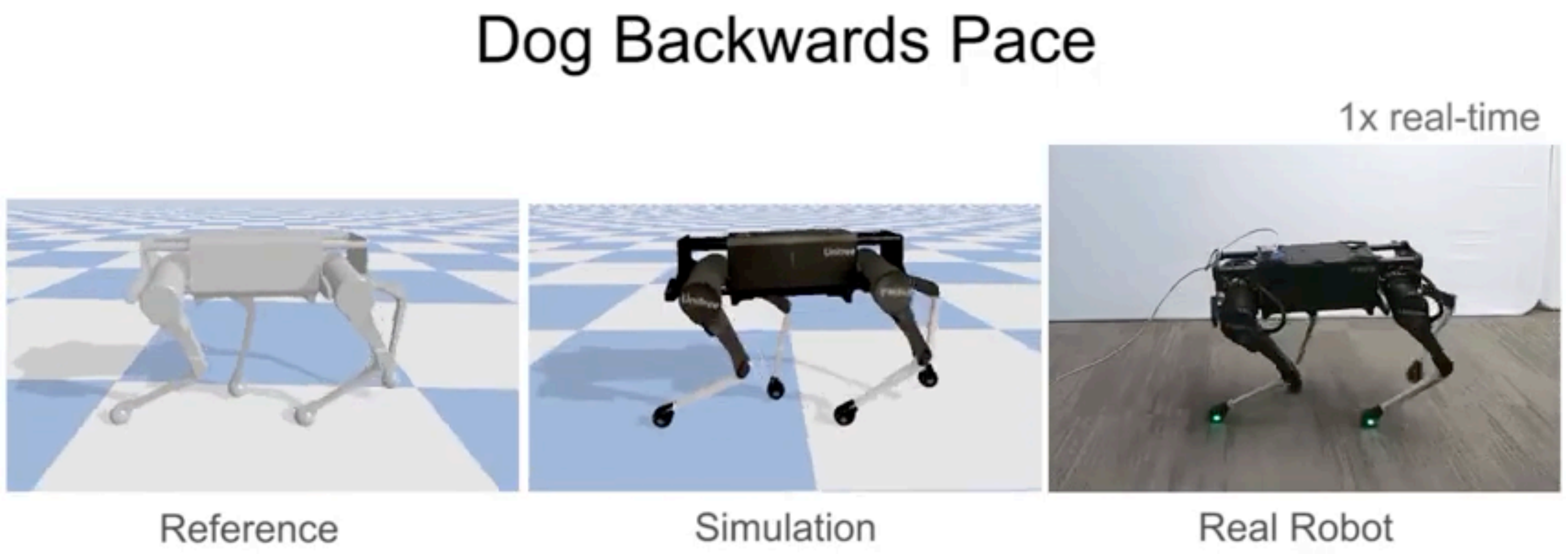
Correct-by-Learning Methods for Reliable Control

Sicun Gao
UCSD

Neural Network Controllers



(L) Deep Drone Acrobatics, Kaufmann et al., RSS'20
(R) Learning Agile Robotic Locomotion Skills by Imitating Animals, Peng et al., RSS'20

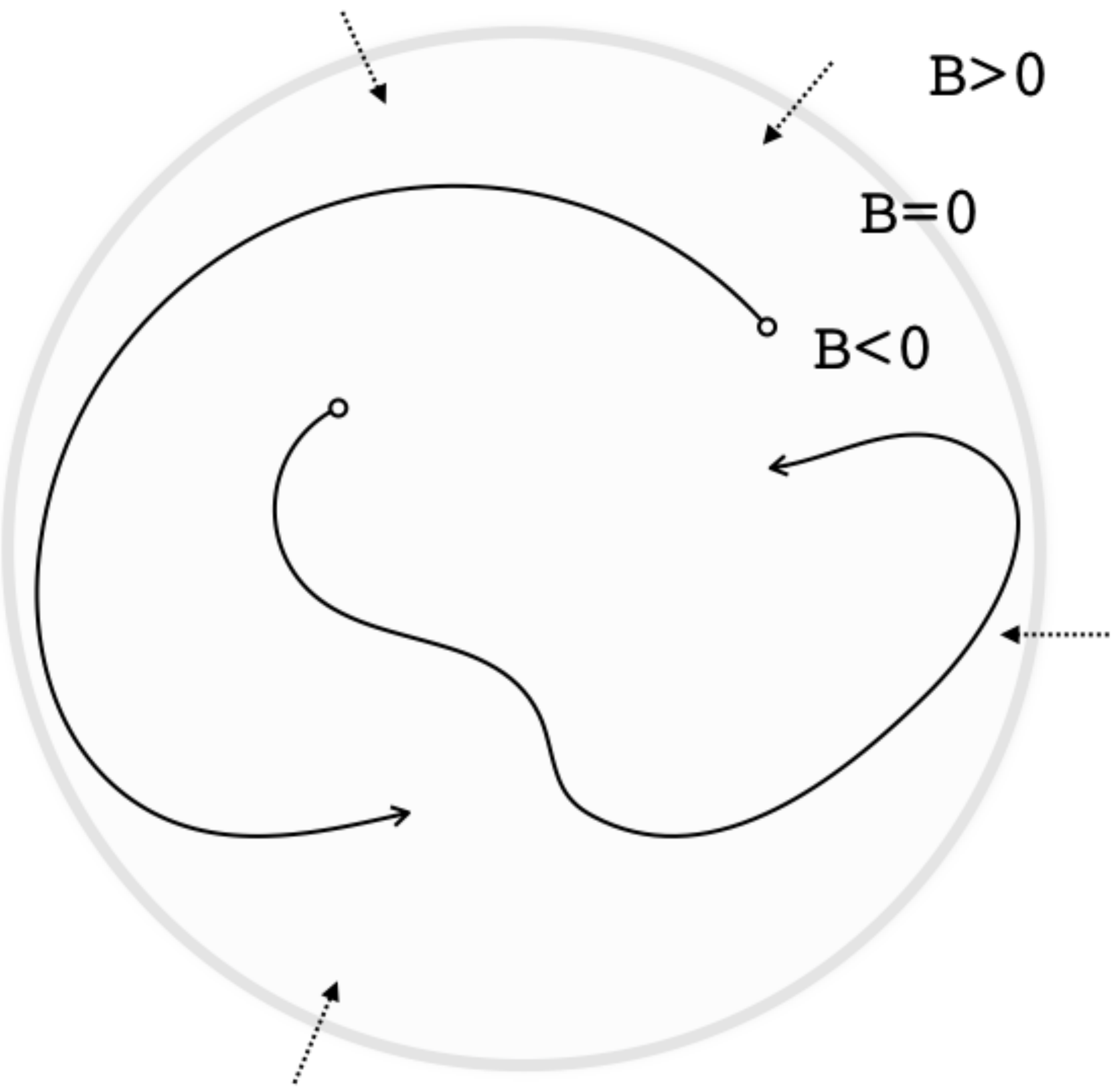
Neural Network Controllers

- Neural networks memorize well. Inputs pass through random activation patterns that create enough degrees of freedom to fit any output.
- For the same reason, never expect it to generalize easily
- We should differentiate "neural" (representations) and "learning" (methods)

Towards rigorous neural control

- Neural control is opening up exciting new directions as general and practical nonlinear control methods
  - More simulation and sampling (data)
  - More demand for scalable optimization (algorithms)
  - More demand for certification (proofs)
- Inductive correct-by-construction methods are the key
  - Convergence of many different areas (FM/Control/ML)

Safety: Barrier Functions

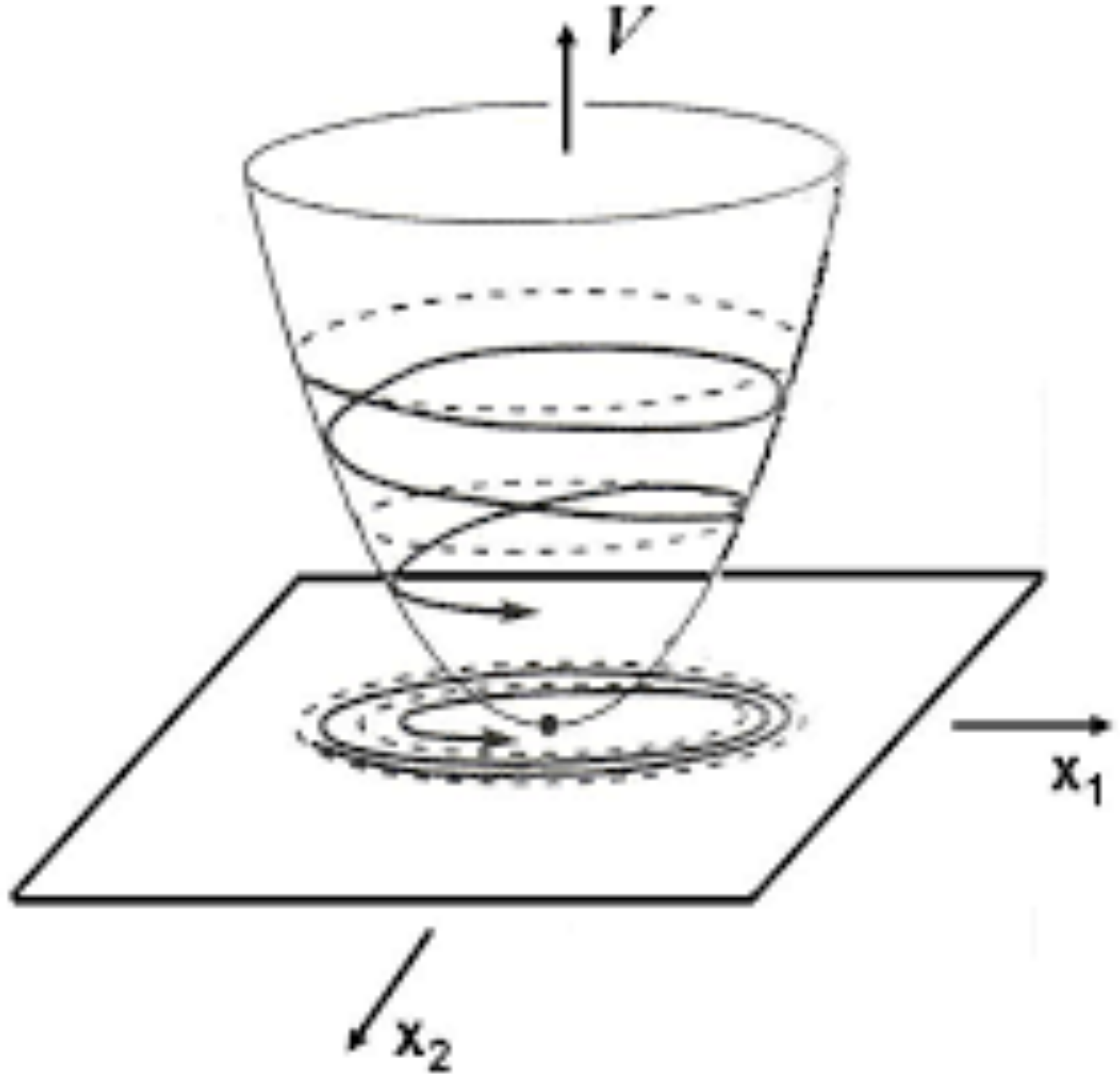


$$B(x) = 0 \rightarrow \mathsf{L}_f B(x) < 0$$

$$\forall x_0 \in X_0 \forall t \in T\left(x_t = \phi(x_0, t) \rightarrow \text{safe}(x_t) \right)$$

- A system is safe if we can construct a forward invariant set to show that the system's orbits can never escape some boundary
- The key is to find the right shape of the barrier function and certify the Lie derivative conditions

Stability: Lyapunov Functions



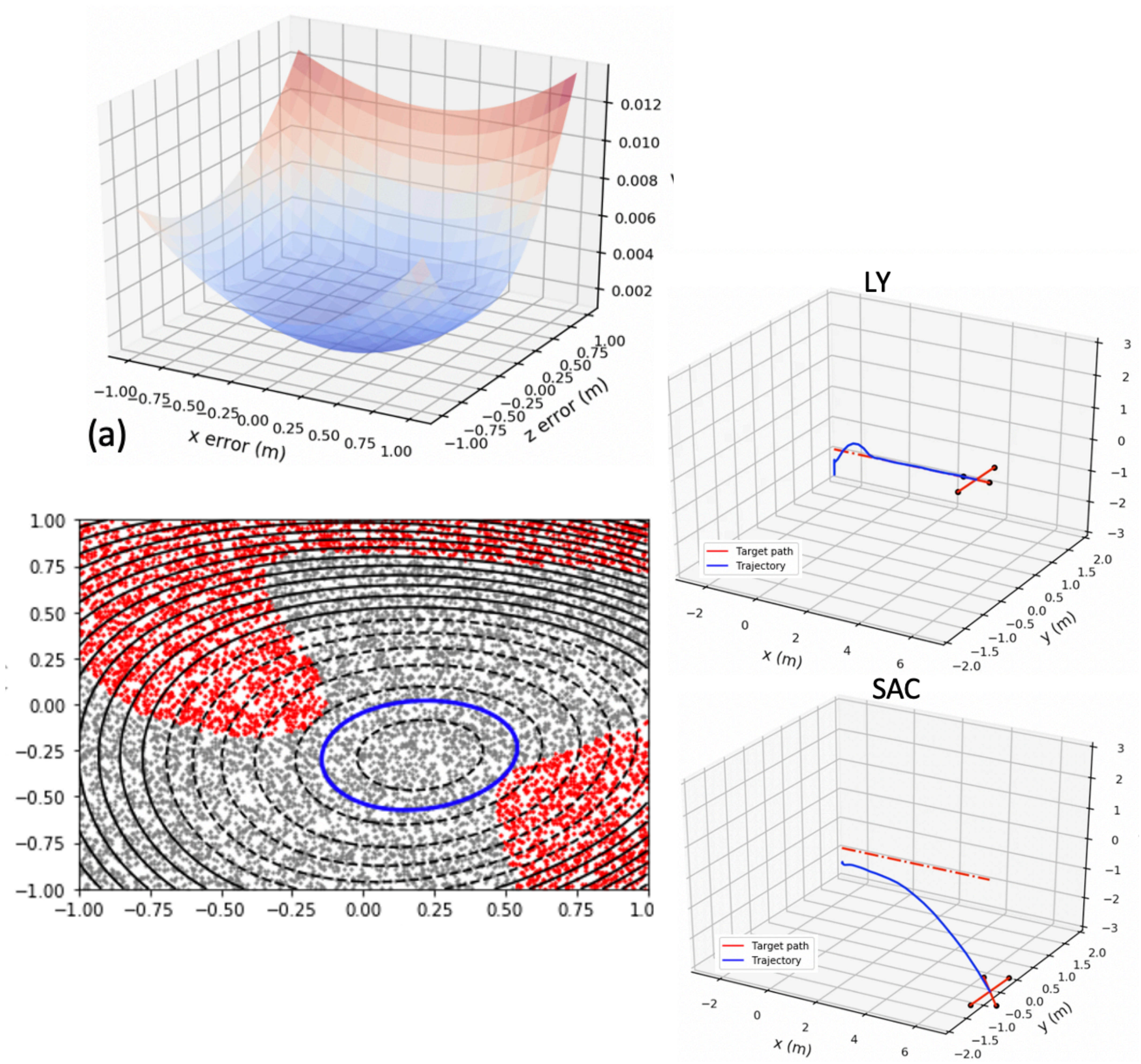
$$V(x) > 0 \wedge \mathsf{L}_f V(x) < 0$$

$$\forall \varepsilon \exists \delta \forall x_0 \forall t$$

$$\left(\left(\|x_0\| < \delta \wedge x_t = \phi(x_0, t)\right) \rightarrow \|x_t\| < \varepsilon\right) \wedge \lim_{t\to\infty} \|\phi(x_0, t)\| = 0$$

- A system is stable if we can construct a Lyapunov function to show that the system has to converge to the stable point

Stability: Lyapunov Functions



- A system is stable if we can construct a Lyapunov function to show that the system has to converge to the stable point
- The key is to find the right shape of the Lyapunov function and certify its Lie derivative conditions

Neural Certificates

- The key to ensuring safety and stability is to find certificate V and control law g to produce inductive proofs by satisfying

$$\exists V \exists g \forall x \; \Phi_f(V, g, x)$$

- With learning−based methods, f can be highly nonlinear dynamics and g can be a deep neural network

- We should need neural network V as well

Neural Certificates

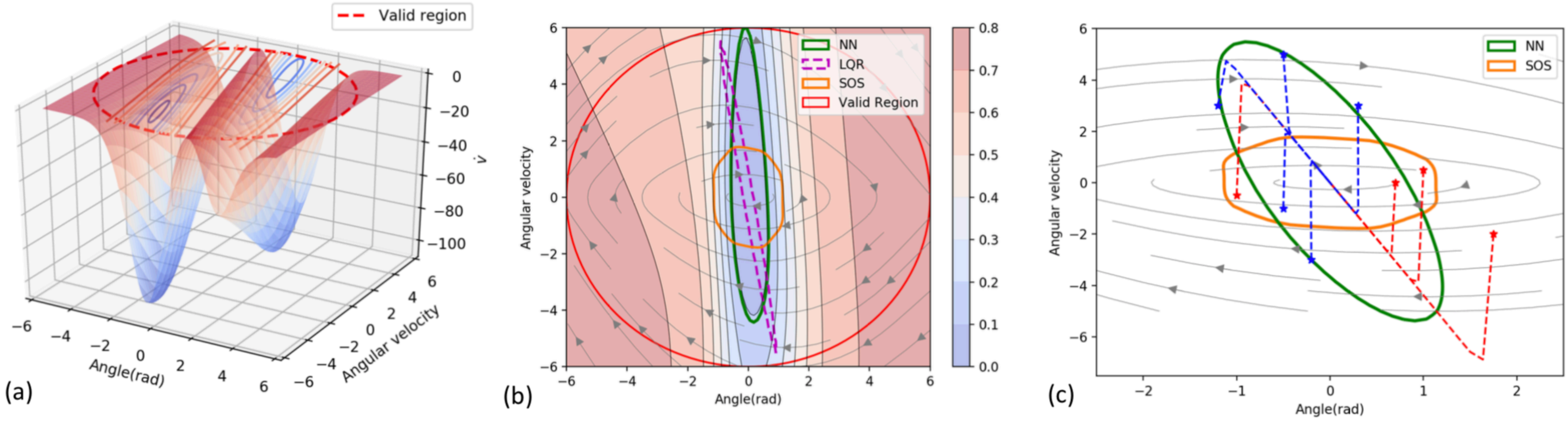
$$\exists V \exists g \forall x \; \Phi_f(V, g, x)$$

- Key to the success of neural certificates:
  - Expressive function approximators
  - Scalable optimization (for $\exists$)
  - Scalable certification (for $\forall$)

Neural Lyapunov Control [NeurIPS'19]

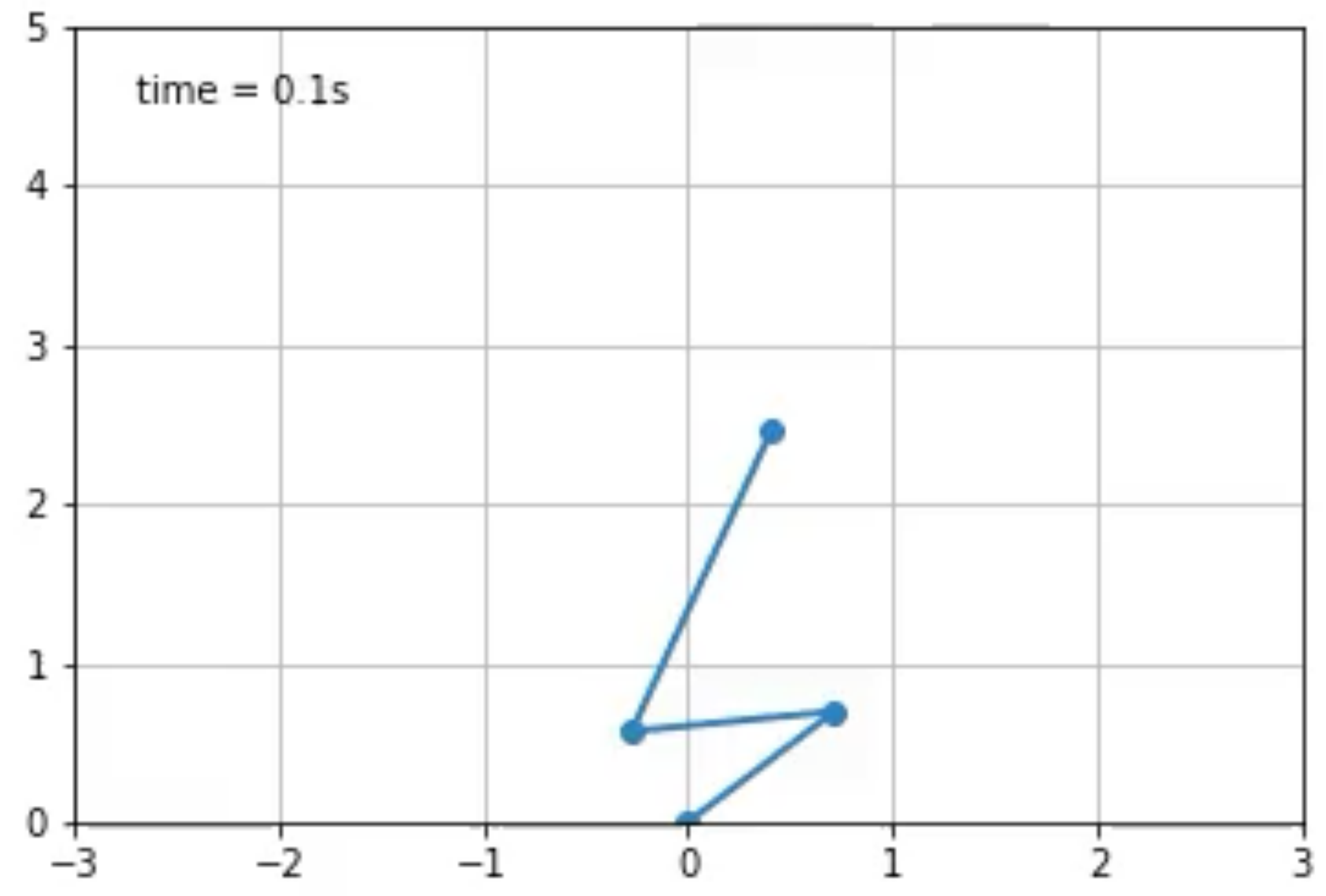
- Learn neural network Lyapunov functions purely from samples (the $\exists$ part) and then give it to solver to directly certify (the $\forall$ part)
  - Turns out dReal can often handle reasonably small tanh neural networks better than polynomials

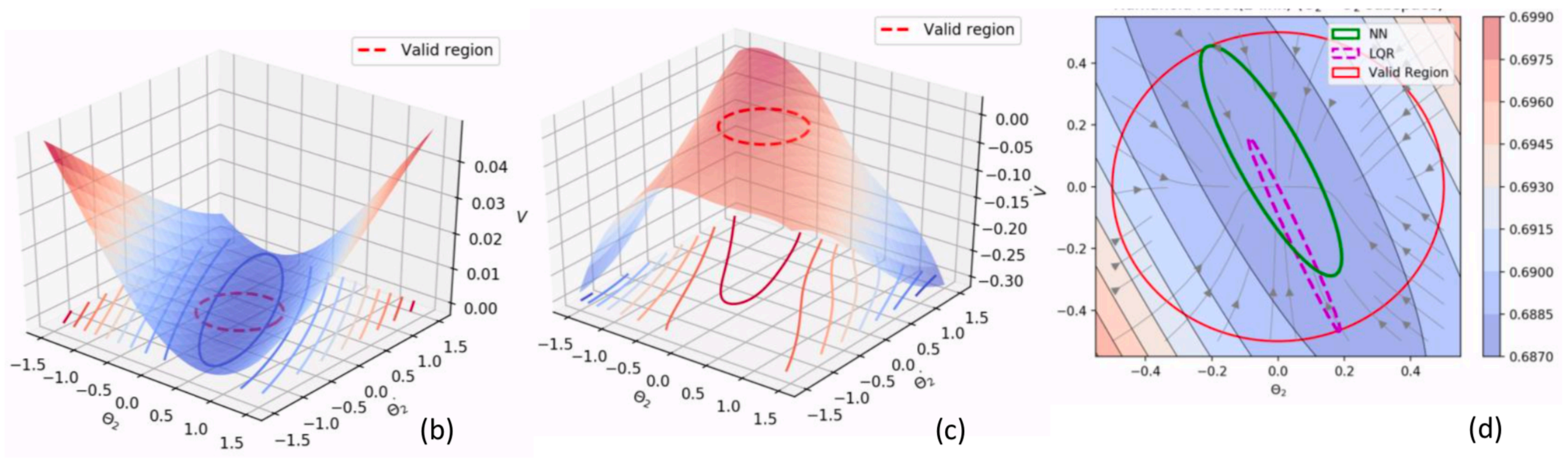
Neural Lyapunov Control



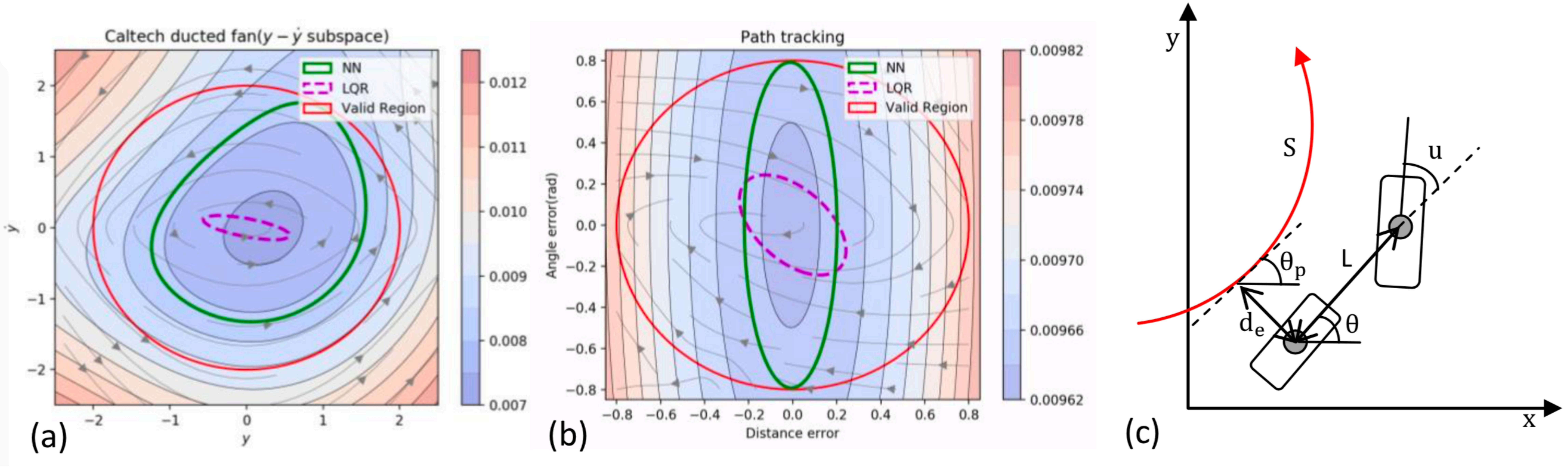
- Learned Lyapunov landscape (showing Lie−derivatives in (a)) for inverted pendulum

Neural Lyapunov Control





Neural Lyapunov Control



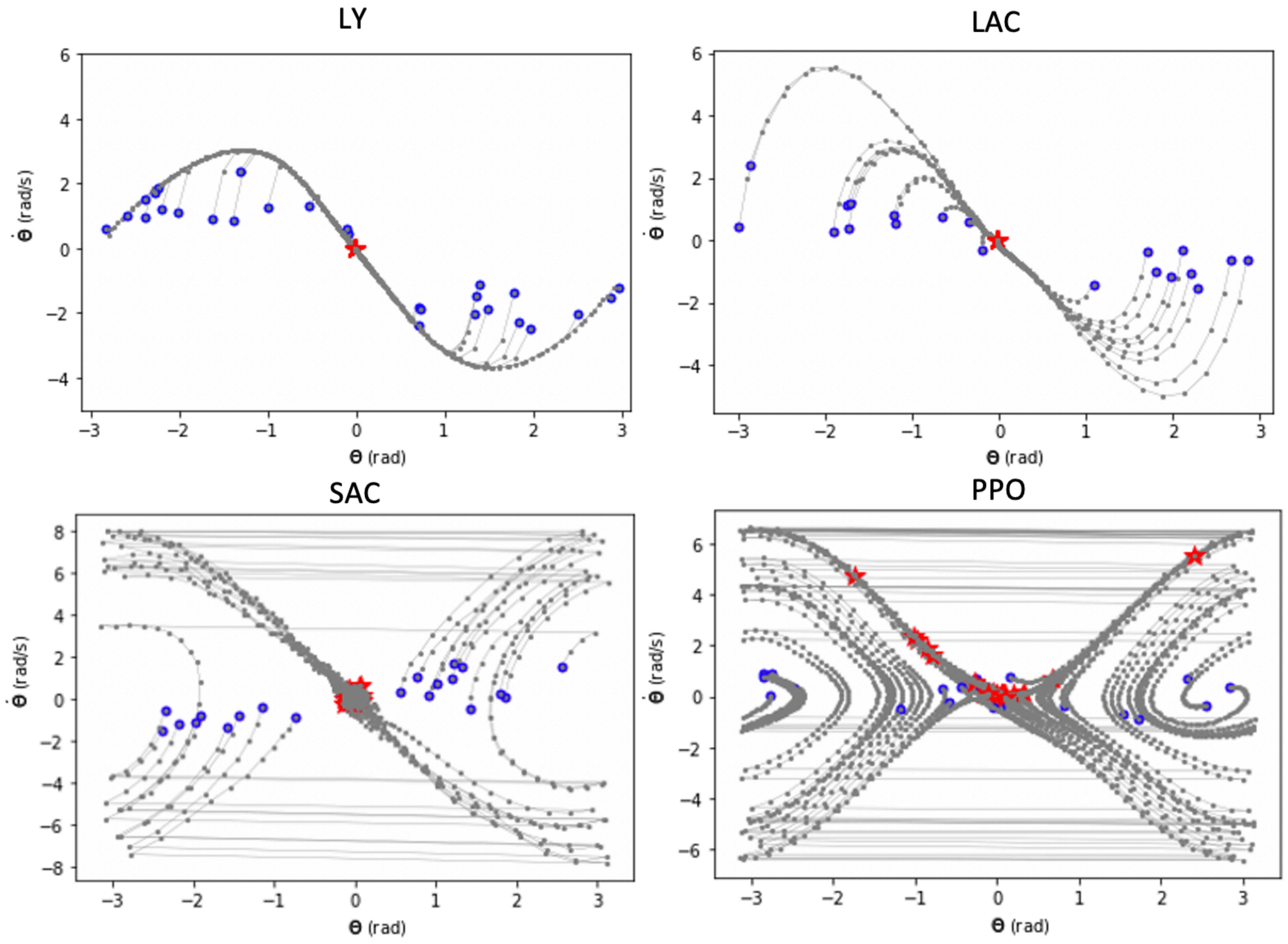
Neural Almost-Lyapunov Critics [ICRA'21]

- After seeing the capability of neural Lyapunov functions, we became more bold in pushing it to full "neural control" setting:
  - model-free learning
  - integrate in policy optimization
  - sampling-based certification

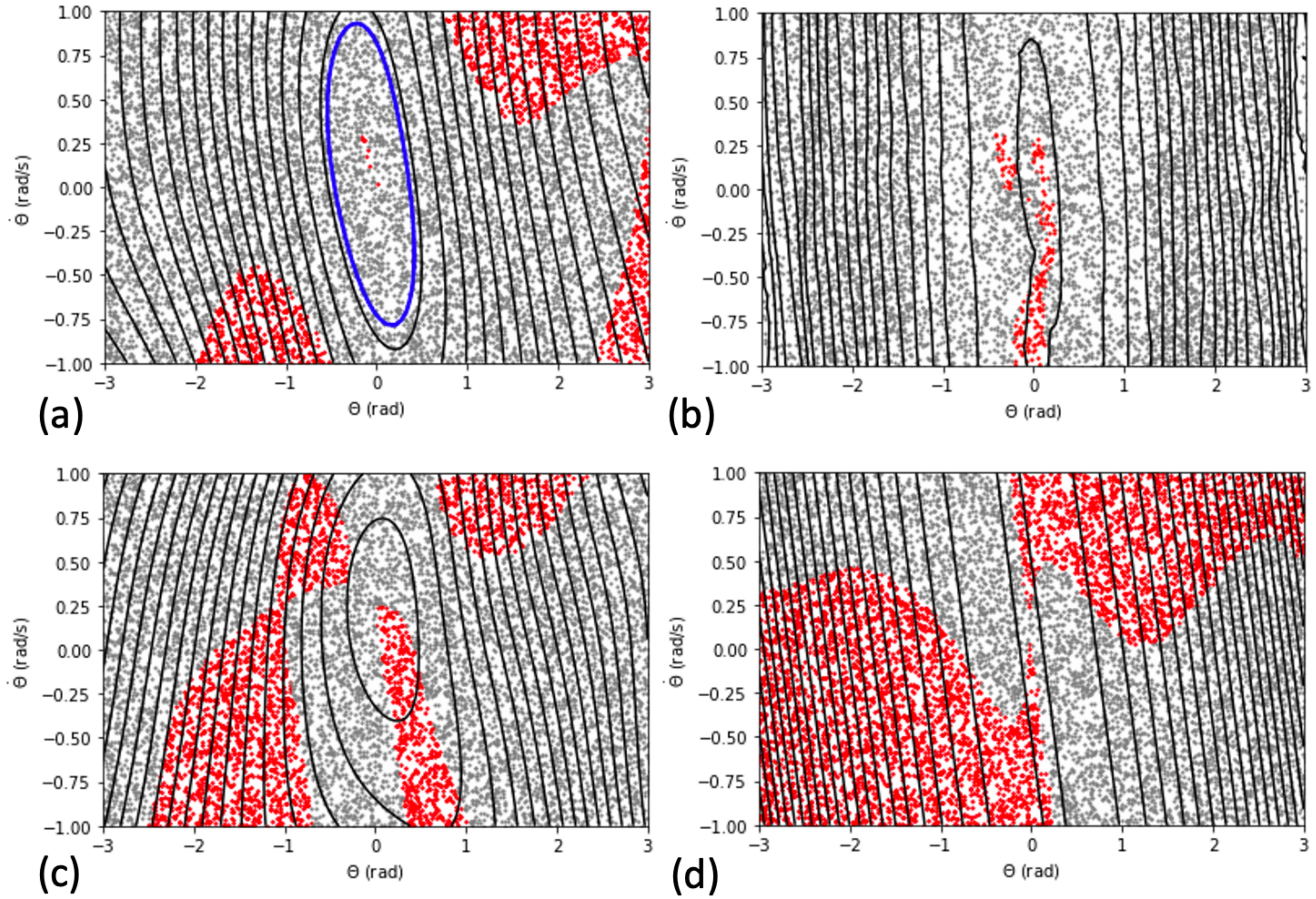
Neural Almost–Lyapunov Critics

- We ask the agent to fit a temporary “neural Lyapunov function” in the critic step
- The policy optimization steps move actions towards more negative Lie derivatives for the temporarily frozen Lyapunov critic function
- Basically the agent attempts to formulate its own stability proof and learns policy to support that

Neural Almost-Lyapunov Critics



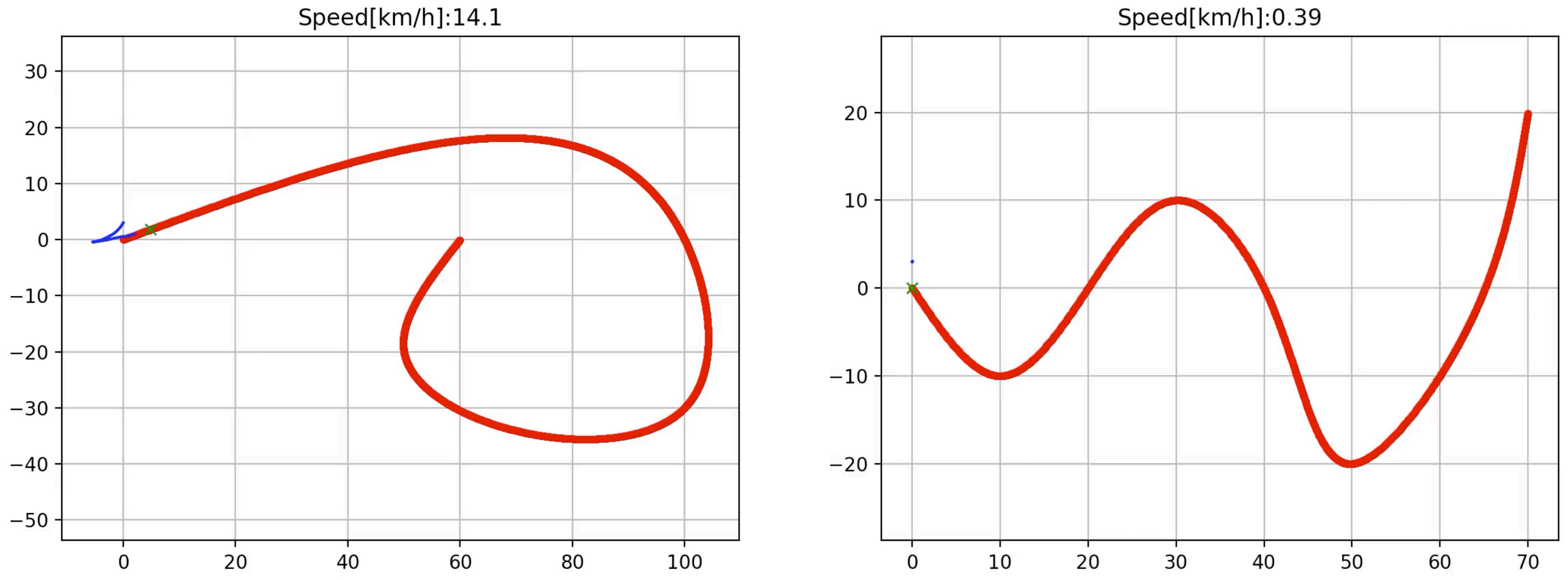
phase plot



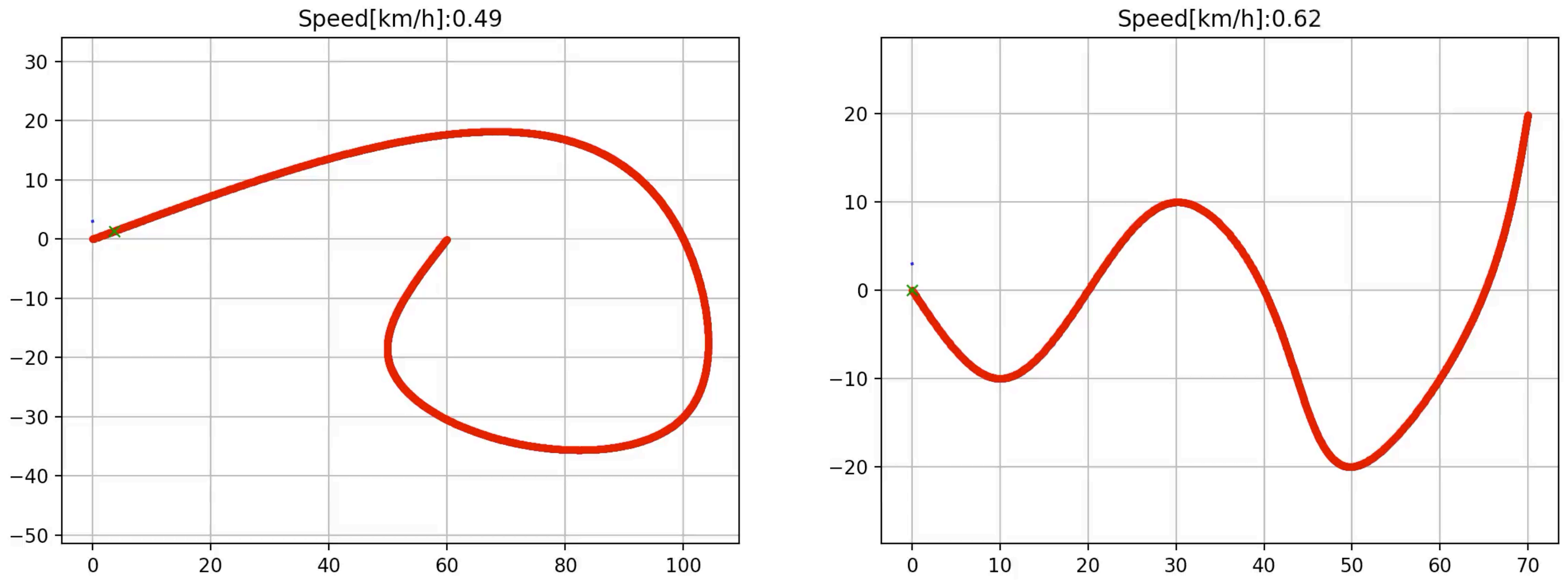
Lyapunov analysis of region of attraction

Neural Almost-Lyapunov Critics

PPO in trained vs. untrained env

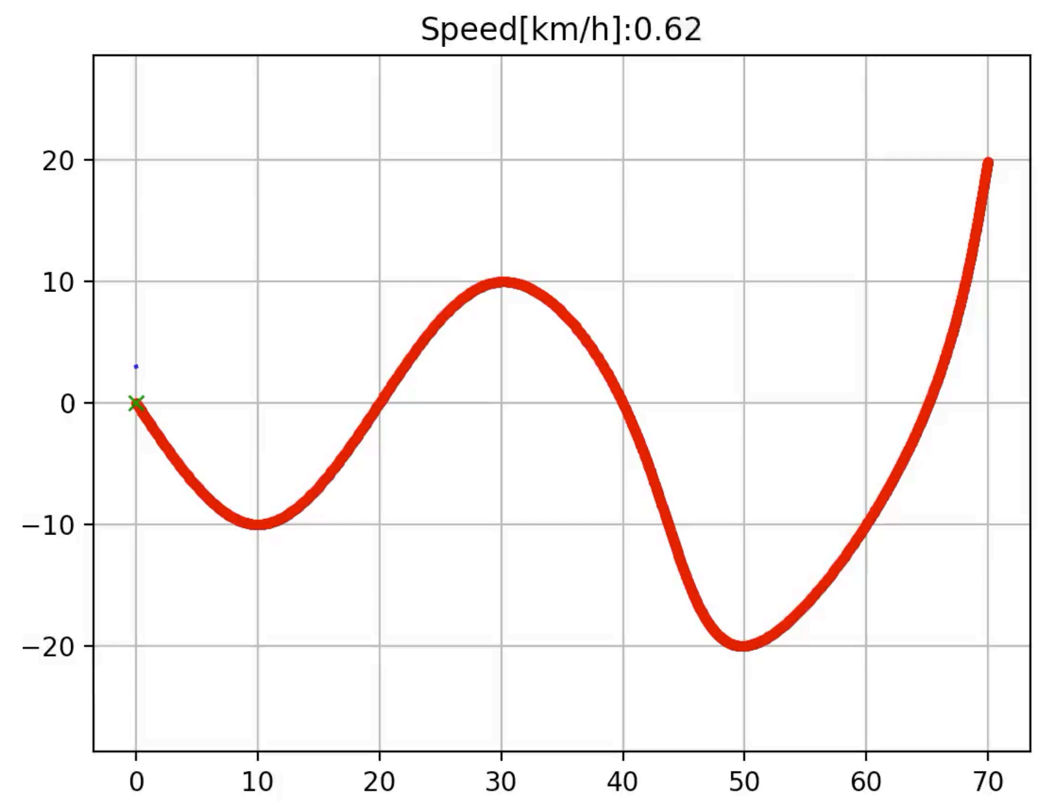


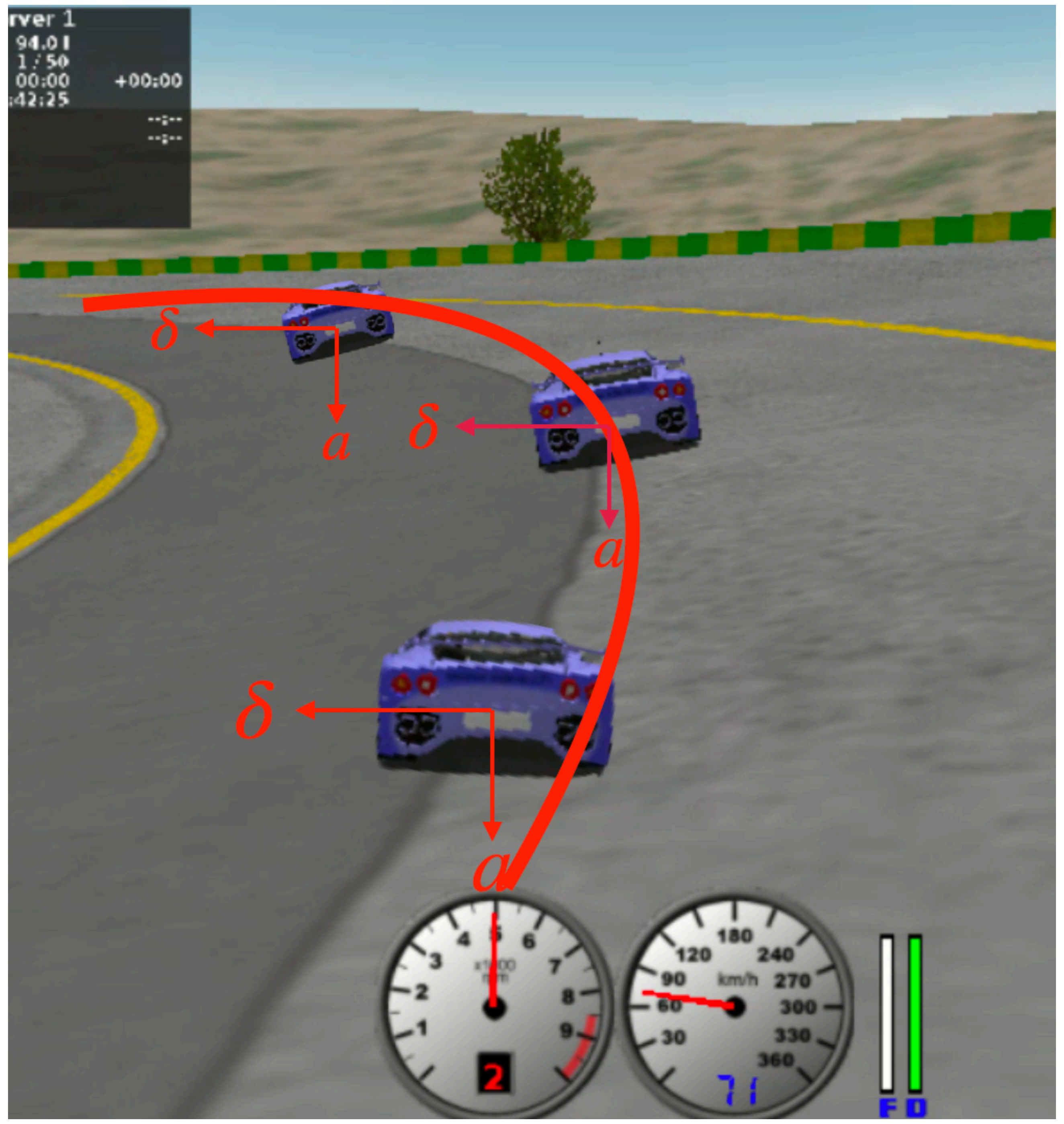
PPO with Lyapunov critics
in trained vs. untrained env

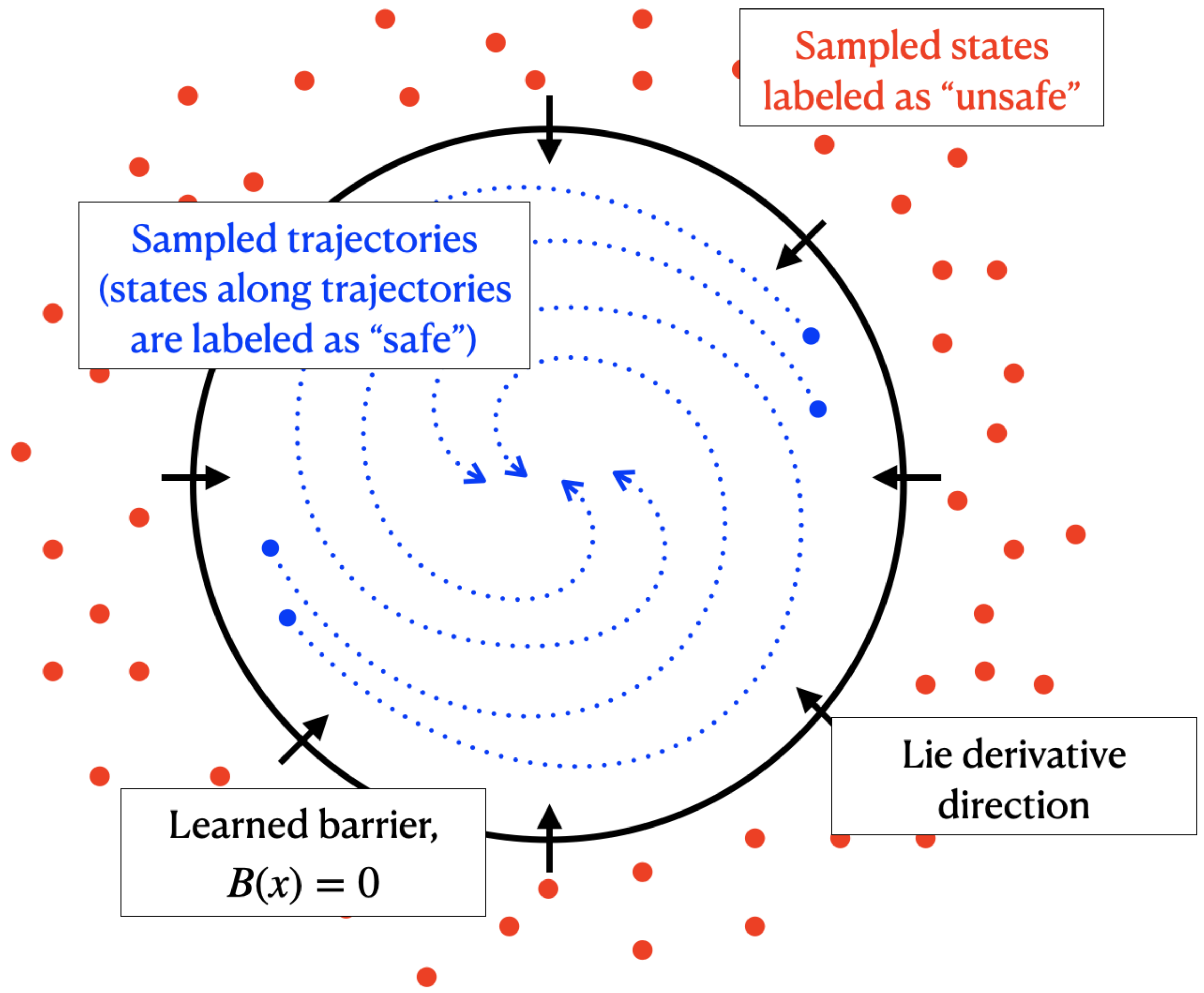


Quantifying Safety of Neural Controllers

Learn a barrier function for the trained policy to identify potential forward invariance sets. Assume model−free setting with only blackbox simulators.



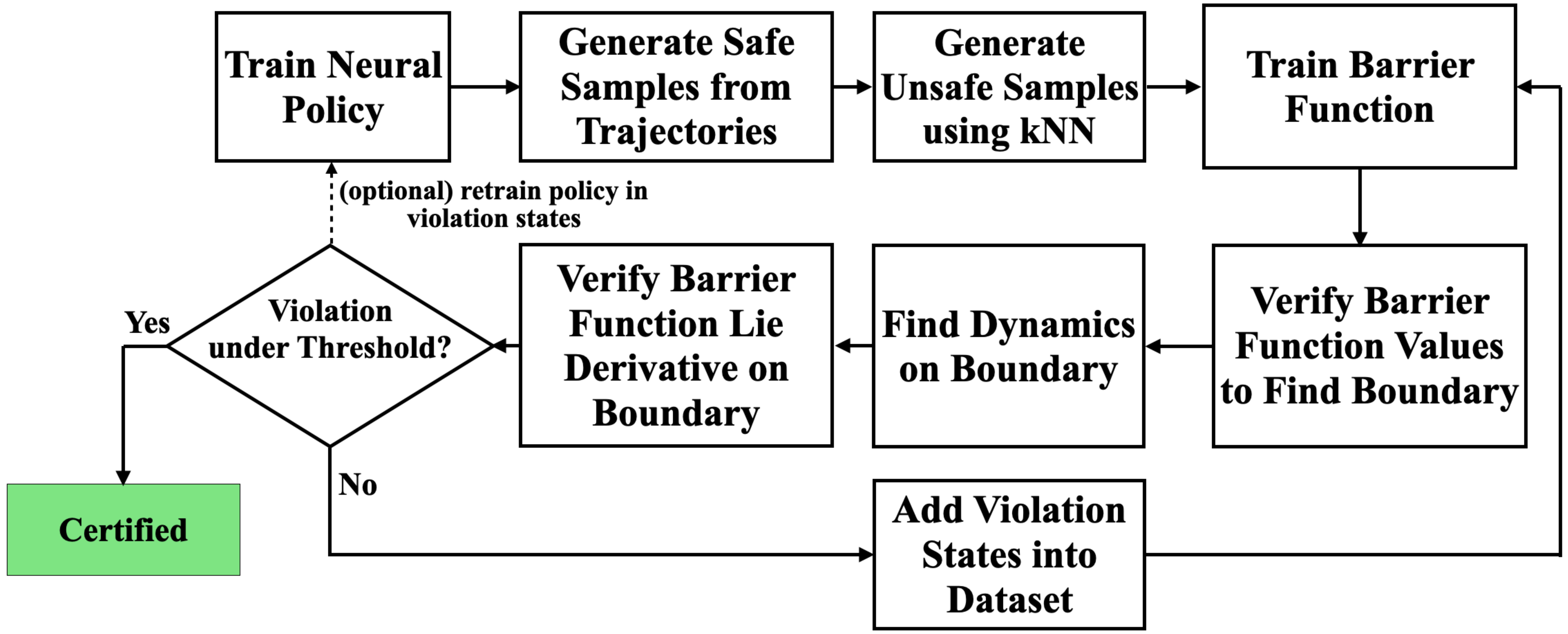




Quantifying Safety of Neural Controllers

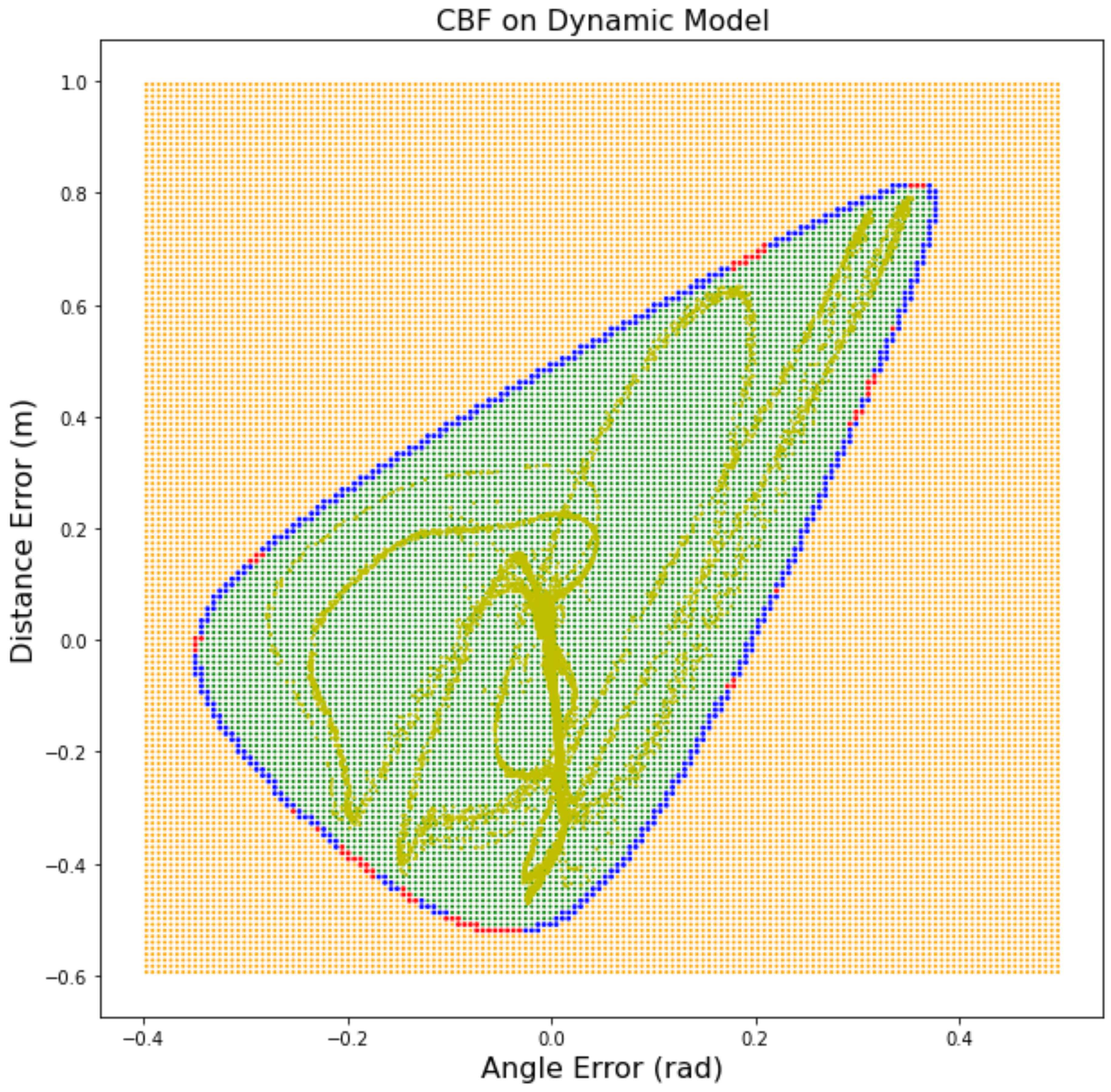
– The learning process turns the barrier conditions into loss functions and uses a counterexample-guided loop.

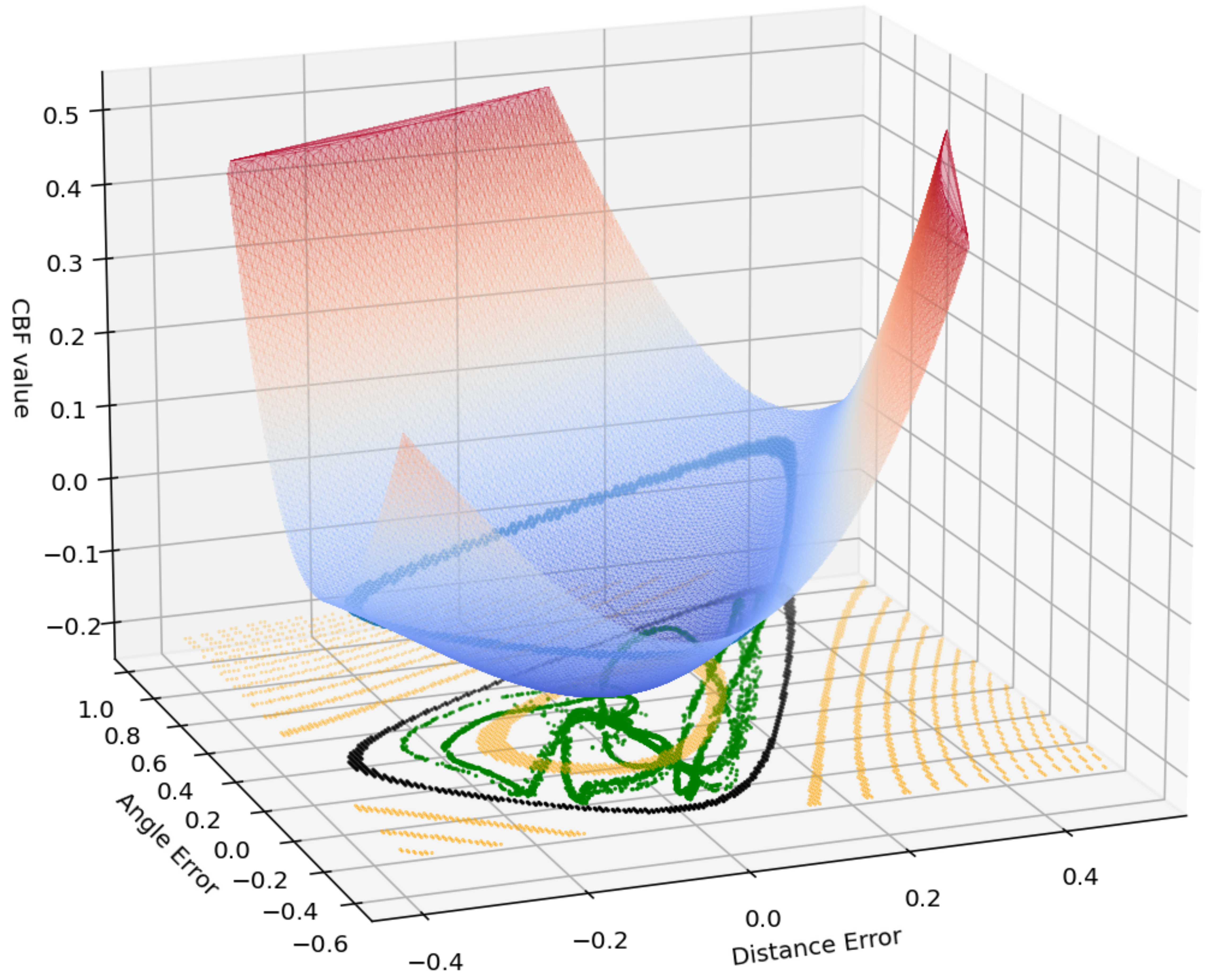
– The certification part is sampling-based and uses robustness analysis of neural networks.



$$L(\theta) = w_s \frac{1}{N_s} \sum_{i=0}^{N_s} \phi(B(x_s^i)) + w_u \frac{1}{N_u} \sum_{i=0}^{N_u} \phi(-B(x_u^i)) + w_l \frac{1}{N_s} \sum_{i=0}^{N_s} \phi(\mathsf{L}_f B(x_s^i) + \gamma B(x_s^i)))$$

Quantifying Safety of Neural Controllers





Towards rigorous neural control

- More simulation and sampling (data)
  - blackbox high–fidelity simulators
  - Analytic methods still important
  - RL is not the only way

Towards rigorous neural control

- More demand for scalable optimization (algorithms)
  - Optimization will take care of the details of design
  - Nonconvex optimization will become mainstream
  - Division of labor between human insight and algorithmic automation (learning and optimization will become part of the "compilers")

Towards rigorous neural control

- More demand for certification (proofs)
  - Controllers are learned, so designers can no longer say "because I just know"!
  - Mindset from formal methods will be widely adopted

Towards rigorous neural control

- How to make neural controllers generalize (sim to real)?
- How to scale to higher dimensions and eventually allow it to be based on raw perception inputs alone? (but with rigorous certification)
- How to improve interpretability of neural controllers' behaviors?
- How to only use real-world data such as from human interaction?
- How to build reliable end-to-end system stacks for neural controllers?

Conclusion

- Neural network controllers are opening up exciting new fronts of practical nonlinear control and formal methods
  - More simulation and sampling (data)
  - More demand for scalable optimization (algorithms)
  - More demand for certification (proofs)
- Convergence of many different areas