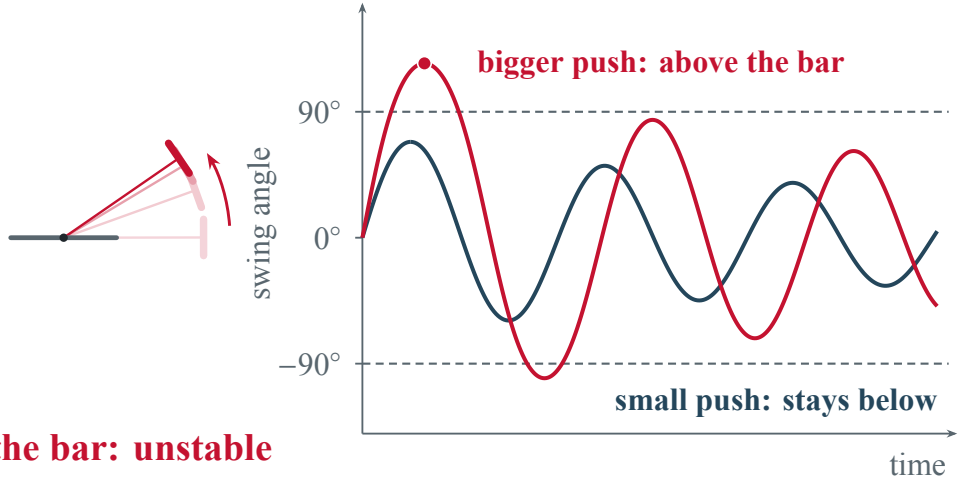
Annual Meeting 2026 · San Francisco · November 3
Session TE46: Decision-Focused Learning

Learning to Test:

Latent Representation Learning for Dynamic Instability Detection

Minxing Zheng[1], Zewei Deng[2], Liyan Xie[2], Shixiang (Woody) Zhu[1]
[1]Carnegie Mellon University [2]University of Minnesota

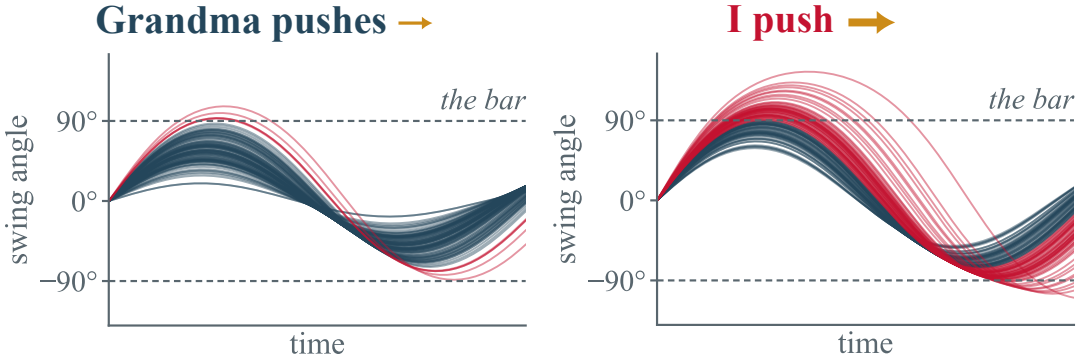
A toy example: the swing



above the bar: unstable

TOY EXAMPLE

Same swing, a different pusher

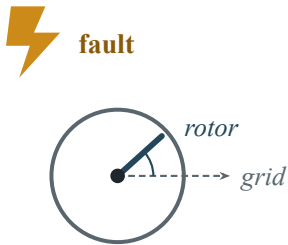


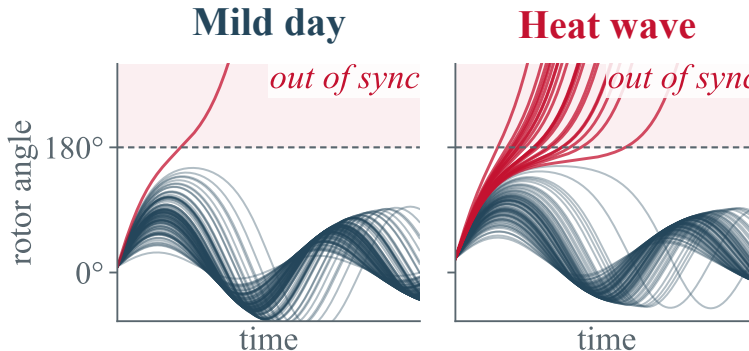
95% stay below the bar

only 50% stay below

When the external environment changes, does the system stay stable?

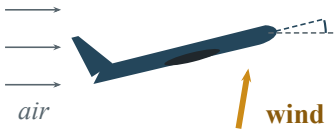
Power grid: a heat wave (control settings)

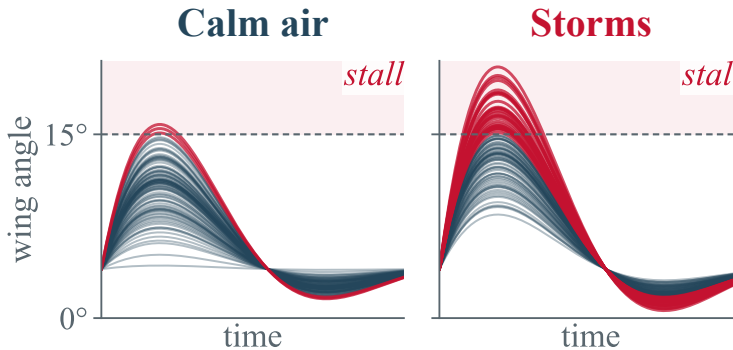




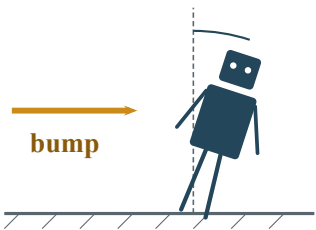
99% stable **78% stable**

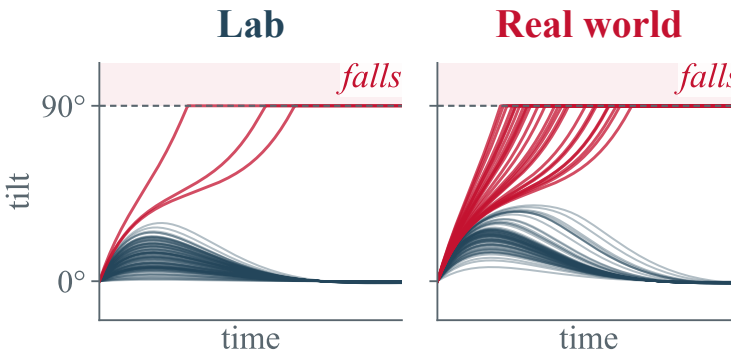
Aircraft: storm season (autopilot)





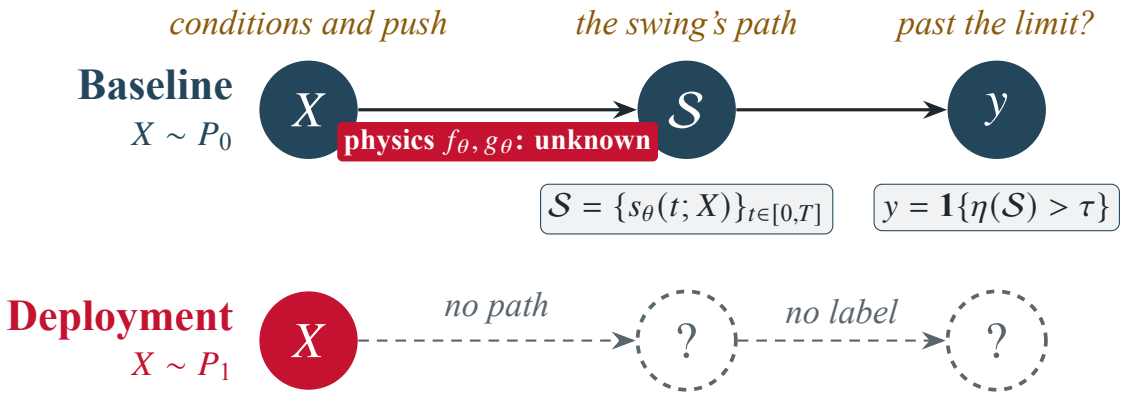
Robot: out of the lab (policy)





97% stable **74% stable**

Baseline and deployment



Stability as a chance constraint

- **System:** a DAE whose physics θ we do not know exactly,

$$\dot{s} = f_\theta(s, a; X) \qquad \textit{dynamics}$$
$$0 = g_\theta(s, a; X) \qquad \textit{constraints}$$

- **Stable regime:** unstable with probability at most α,

$$\mathcal{P}_0 = \left\{P : \mathbb{P}_{X\sim P}[y(X) = 1] \le \alpha\right\}, \qquad P_0 \in \mathcal{P}_0.$$

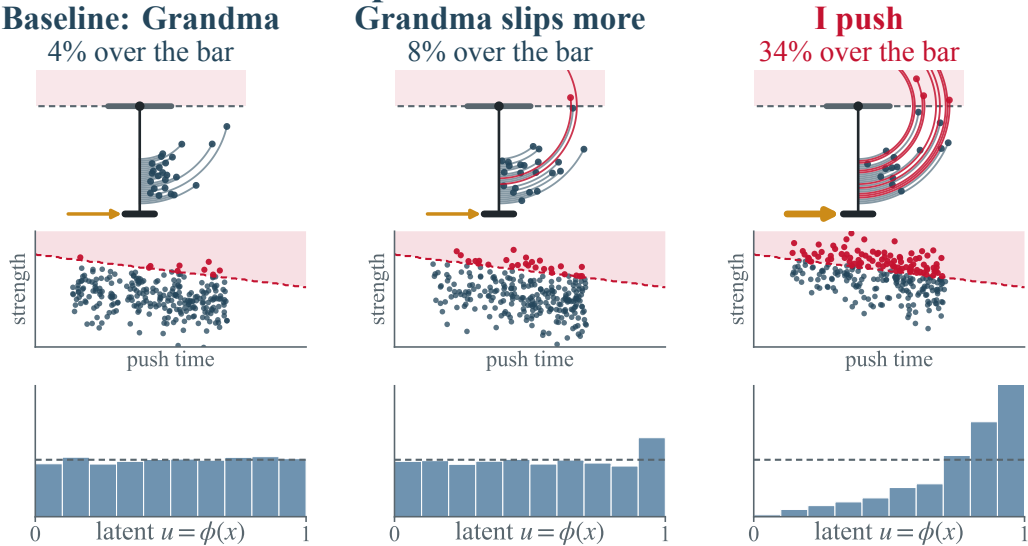
Our question of interest: is the deployment stable?

$$H_0 : P_1 \in \mathcal{P}_0 \qquad \text{vs.} \qquad H_1 : P_1 \notin \mathcal{P}_0 \qquad \text{(a composite null)}$$

Why the standard routes fall short

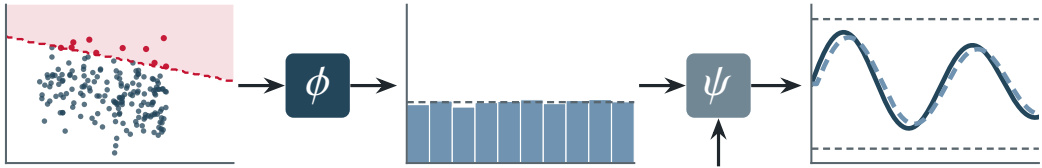
1. **Simulate every trajectory.**
   - *Requires the exact physics law, and is computationally slow.*
2. **Two-sample test on X.**
   - *Detects all changes: raises false alarms.*
3. **Classify $X \mapsto y$.**
   - *Not dynamics-aware: sees only a 0/1 label.*

Learn a test-focused representation

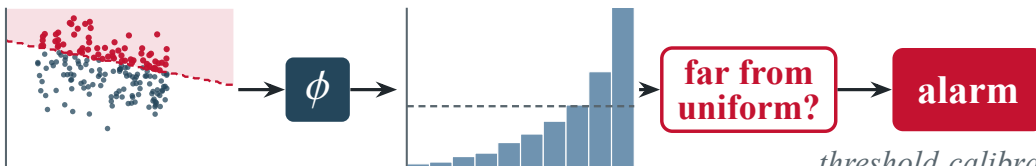


Train once, test any deployment

Train on baseline runs *(contexts and their trajectories)*



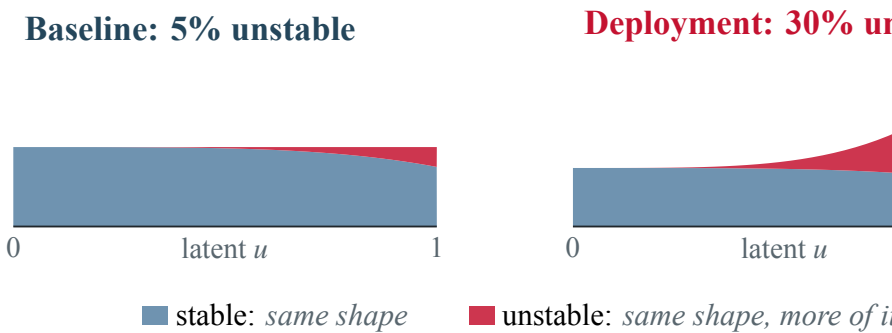
Test a new deployment *(contexts only)*



One assumption: only the mix changes

Assumption. $\mathrm{Law}_{P_1}(u \mid y) = \mathrm{Law}_{P_0}(u \mid y)$

Label distribution changes; the conditional latent distribution remains unchanged.



Theoretical guarantees (informal)

Original test

$H_0 : \mathbb{P}_{X \sim P_1}[y(X) = 1] \leq \alpha$

needs trajectories and labels

Theorem 1

exact

Latent test

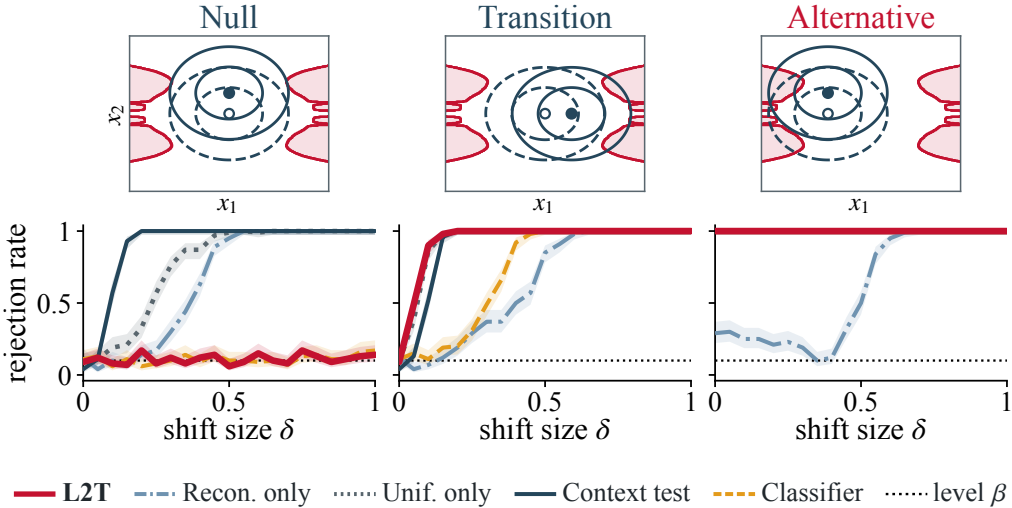
$H_0 : D(Q_{P_1}, Q^*) \leq r_\alpha$

needs only contexts, $u = \phi(x)$

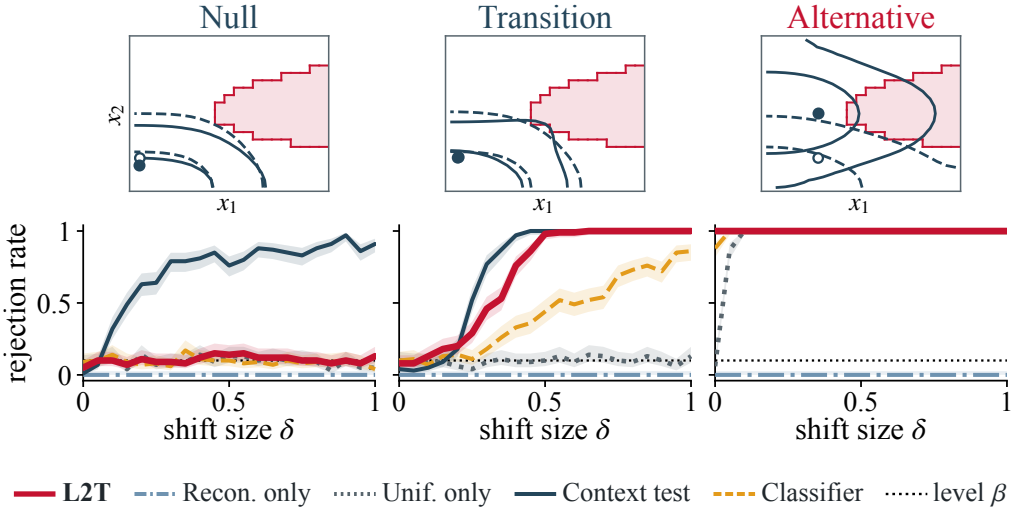
Theorem 2 (finite-sample error calibration)

Calibrate r_α on stable re-mixes of baseline data:
false alarms $\leq \beta$ for every stable deployment, and power $\to 1$.

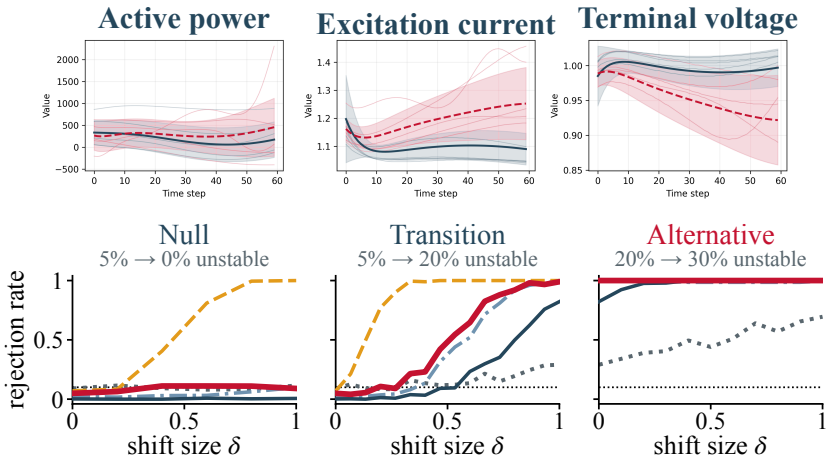
Mass–spring–damper



Cartesian pendulum



Power grid: faults on the IEEE 39-bus system



TAKEAWAYS

Test-focused learning for stability

1. **Test-focused representation learning.**
2. **Exact reformulation as a latent-space test.**

Limits and next steps

- Broader shifts → add physics (energy functions).
- Sequential monitoring.
- More applications: aircraft, robots.

Thank you

Questions and feedback welcome

Website



minxing-zheng.github.io

Paper



arxiv.org/abs/2604.10967

Learning to Test: Latent Representation Learning for Dynamic Instability Detection

Minxing Zheng, Zewei Deng,
Liyan Xie, Shixiang (Woody) Zhu
minxingz@andrew.cmu.edu