

Lightweight Physics-Informed Reservoir Computing for Battery Health Prediction

The 6th Modeling, Estimation and Control Conference

Kumar Anurag* Yanwen Xu[†] Wenbin Wan*

* *University of New Mexico, Albuquerque, NM, USA*

[†] *University of Texas at Dallas, Richardson, TX, USA*

Oct 25-28, 2026

📍 Phoenix, AZ, USA

Why Battery Health Prediction Matters

- Electric vehicles are rapidly growing



Image Source: Unsplash / Research Database

Why Battery Health Prediction Matters

- ▶ Electric vehicles are rapidly growing
- ▶ State-of-Health (SOH) estimation is critical for:
 - Safety and reliability
 - Remaining capacity tracking
 - Failure prediction



Image Source: Vecteezy

Why Battery Health Prediction Matters

- ▶ Electric vehicles are rapidly growing
- ▶ State-of-Health (SOH) estimation is critical for:
 - Safety and reliability
 - Remaining capacity tracking
 - Failure prediction
- ▶ **Challenge:** Battery Management System (BMS) runs on resource- constrained microcontrollers

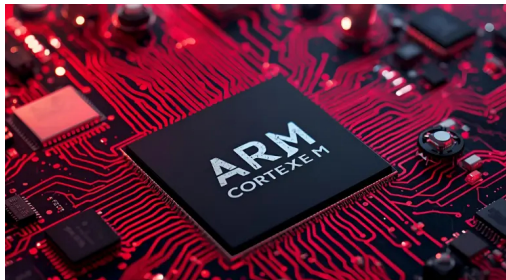


Image Source: Indian Institute of Embedded Systems

Why Battery Health Prediction Matters

- ▶ Electric vehicles are rapidly growing
- ▶ State-of-Health (SOH) estimation is critical for:
 - Safety and reliability
 - Remaining capacity tracking
 - Failure prediction
- ▶ **Challenge:** Battery Management System (BMS) runs on resource- constrained microcontrollers

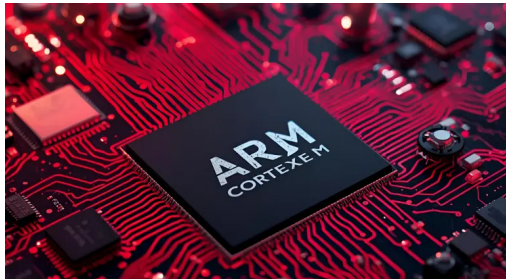
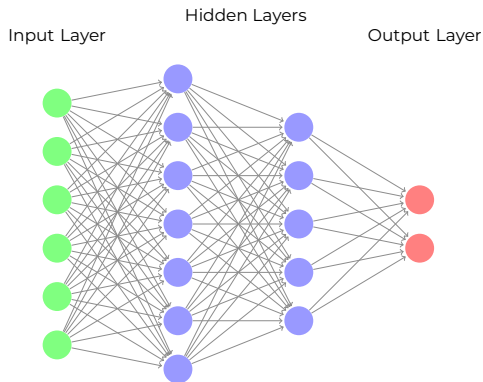


Image Source: Indian Institute of Embedded Systems

➡ **We need accurate models that fit on embedded hardware.**

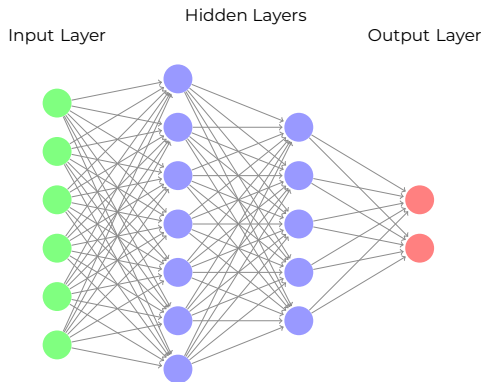
Limitations of Pure Data-Driven Models

- ▶ Traditional data-driven approaches act as “black-box” mappings.



Limitations of Pure Data-Driven Models

- ▶ Traditional data-driven approaches act as “black-box” mappings.
- ▶ They rely *entirely* on historical data distributions.



Limitations of Pure Data-Driven Models

- ▶ Traditional data-driven approaches act as “black-box” mappings.
- ▶ They rely *entirely* on historical data distributions.
- ▶ **Critical Failure Mode:**
 - Poor generalization to unseen operating conditions.
 - Examples: Extreme temperatures, irregular charging protocols.

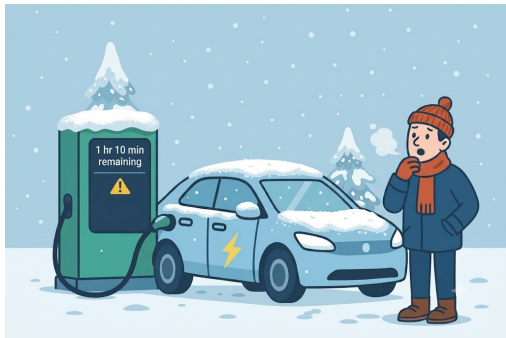


Image Source: ARP AE

Limitations of Pure Data-Driven Models

- ▶ Traditional data-driven approaches act as “black-box” mappings.
- ▶ They rely *entirely* on historical data distributions.
- ▶ **Critical Failure Mode:**
 - Poor generalization to unseen operating conditions.
 - Examples: Extreme temperatures, irregular charging protocols.

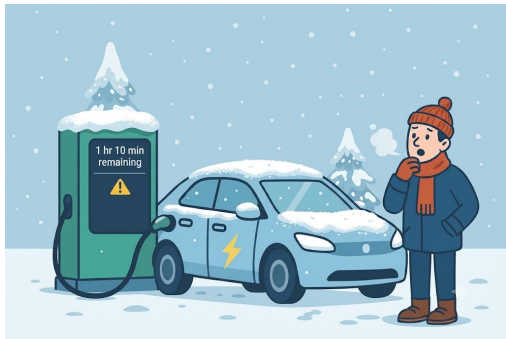


Image Source: ARPAE

➡ **Without physical constraints, data-driven models can produce unsafe predictions.**

Physics-Informed Neural Networks (PINNs)

- **Idea:** Embed governing physical laws directly into the training objective.

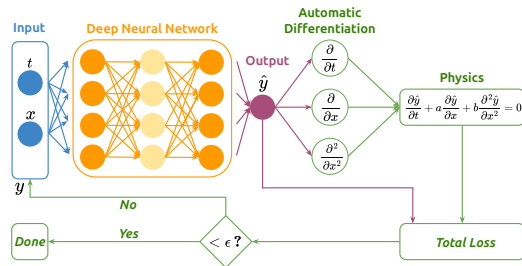


Figure 1: PINN Architecture, [Ref: Trahan, C., Loveland, M., & Dent, S. (2024).

Quantum physics-informed neural networks. Entropy, 26(8), 649.]

Physics-Informed Neural Networks (PINNs)

- **Idea:** Embed governing physical laws directly into the training objective.

- **Loss Function:**

$$\mathcal{L}(\theta) = \underbrace{\frac{1}{T} \sum \|\hat{y}_\theta - y\|^2}_{\text{Data Residual}} + \lambda_p \underbrace{\frac{1}{T} \sum \|\dot{\hat{y}}_\theta - \tilde{f}(\hat{y}_\theta)\|^2}_{\text{Physics Residual}}$$

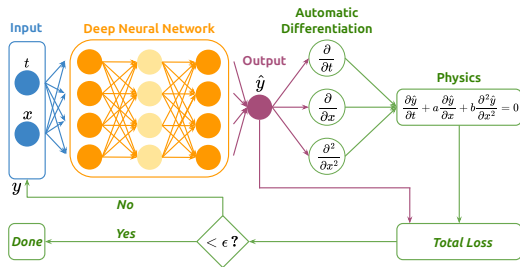


Figure 1: PINN Architecture, [Ref: Trahan, C., Loveland, M., & Dent, S. (2024).

Quantum physics-informed neural networks. Entropy, 26(8), 649.]

Physics-Informed Neural Networks (PINNs)

- **Idea:** Embed governing physical laws directly into the training objective.

- **Loss Function:**

$$\mathcal{L}(\theta) = \underbrace{\frac{1}{T} \sum \|\hat{y}_\theta - y\|^2}_{\text{Data Residual}} + \lambda_p \underbrace{\frac{1}{T} \sum \|\dot{\hat{y}}_\theta - \tilde{f}(\hat{y}_\theta)\|^2}_{\text{Physics Residual}}$$

- **Limitations:**

- Requires iterative, gradient-based optimization (backpropagation).
- Thousands of trainable parameters.
- **Result:** Prohibitive memory and compute cost for edge devices.

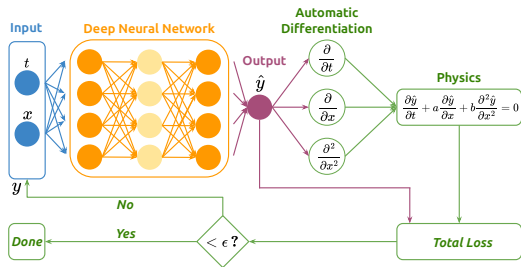


Figure 1: PINN Architecture, [Ref: Trahan, C., Loveland, M., & Dent, S. (2024).

Quantum physics-informed neural networks. Entropy, 26(8), 649.]

Physics-Informed Neural Networks (PINNs)

- **Idea:** Embed governing physical laws directly into the training objective.

- **Loss Function:**

$$\mathcal{L}(\theta) = \underbrace{\frac{1}{T} \sum \|\hat{y}_\theta - y\|^2}_{\text{Data Residual}} + \lambda_p \underbrace{\frac{1}{T} \sum \|\dot{\hat{y}}_\theta - \tilde{f}(\hat{y}_\theta)\|^2}_{\text{Physics Residual}}$$

- **Limitations:**

- Requires iterative, gradient-based optimization (backpropagation).
- Thousands of trainable parameters.
- **Result:** Prohibitive memory and compute cost for edge devices.

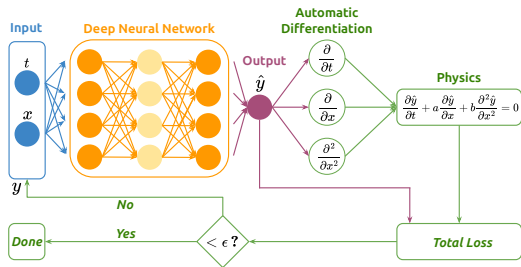


Figure 1: PINN Architecture, [Ref: Trahan, C., Loveland, M., & Dent, S. (2024).

Quantum physics-informed neural networks. Entropy, 26(8), 649.]

➡ **PINNs respect physical laws, but iterative training is costly.**

Reservoir Computing (RC)

- **Idea:** Avoid backpropagation by using a *fixed*, randomly initialized recurrent network.

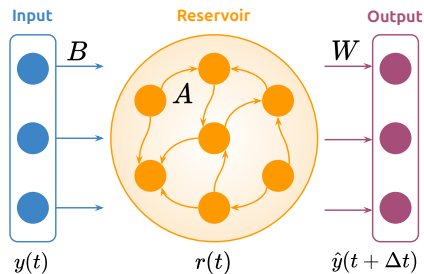


Figure 2: RC Architecture

Reservoir Computing (RC)

- **Idea:** Avoid backpropagation by using a *fixed*, randomly initialized recurrent network.
- **State Update:**

$$r(t) = (1 - \alpha)r(t - \Delta t) + \alpha \tanh(Ar(t - \Delta t) + By(t) + c)$$

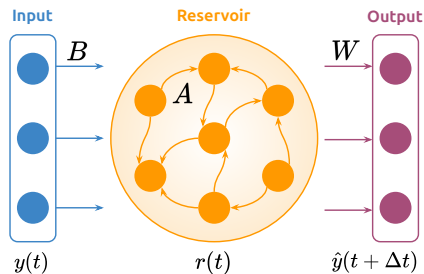


Figure 2: RC Architecture

Reservoir Computing (RC)

- **Idea:** Avoid backpropagation by using a *fixed*, randomly initialized recurrent network.

- **State Update:**

$$r(t) = (1 - \alpha)r(t - \Delta t) + \alpha \tanh(Ar(t - \Delta t) + By(t) + c)$$

- **Closed-Form Readout:**

$$\hat{y}(t + \Delta t) = W_{out}^T r(t), \text{ where } W_{out} \text{ is solved via one-shot ridge regression.}$$

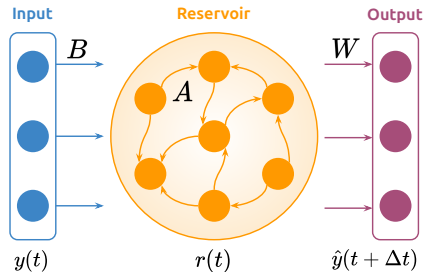


Figure 2: RC Architecture

Reservoir Computing (RC)

- **Idea:** Avoid backpropagation by using a *fixed*, randomly initialized recurrent network.

- **State Update:**

$$r(t) = (1 - \alpha)r(t - \Delta t) + \alpha \tanh(Ar(t - \Delta t) + By(t) + c)$$

- **Closed-Form Readout:**

$\hat{y}(t + \Delta t) = W_{out}^T r(t)$, where W_{out} is solved via *one-shot ridge regression*.

- **Limitations:** Requires large N ($N \gg d$) for expressiveness; Over-parameterization causes instability in autonomous rollout; Still lacks physical constraints.

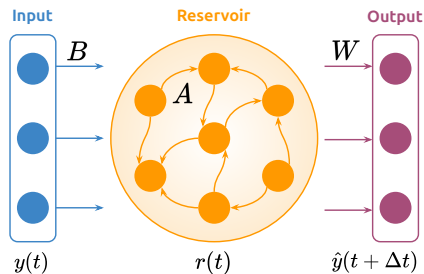


Figure 2: RC Architecture

Reservoir Computing (RC)

- **Idea:** Avoid backpropagation by using a *fixed*, randomly initialized recurrent network.
- **State Update:**

$$r(t) = (1 - \alpha)r(t - \Delta t) + \alpha \tanh(Ar(t - \Delta t) + By(t) + c)$$

- **Closed-Form Readout:**

$\hat{y}(t + \Delta t) = W_{out}^T r(t)$, where W_{out} is solved via *one-shot ridge regression*.

- **Limitations:** Requires large N ($N \gg d$) for expressiveness; Over-parameterization causes instability in autonomous rollout; Still lacks physical constraints.

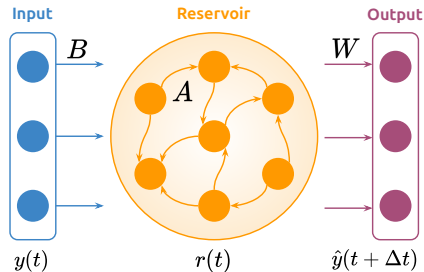


Figure 2: RC Architecture

➡ **RC enables lightweight training, but lacks physics and can suffer from over-parameterization.**

Next-Generation RC (NGRC) & Hybrid Models

- **NGRC:** Replaces the random reservoir with deterministic Nonlinear Vector Autoregression (NVAR).

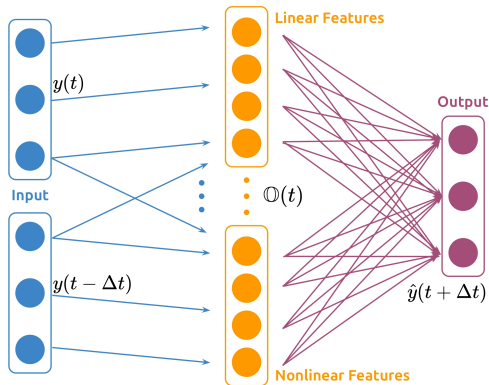


Figure 3: NGRC Architecture, [Ref: Wang, H., Hu, J., Baek, Y., Tsuchiyama, K., Joly, M., Liu, Q., & Gigan, S. (2025). Optical next generation reservoir computing. Light, 14(1), 245.]

Next-Generation RC (NGRC) & Hybrid Models

- **NGRC:** Replaces the random reservoir with deterministic Nonlinear Vector Autoregression (NVAR).

- **NVAR Features:**

$$\mathbb{O}(t) = 1 \oplus \mathbb{O}_{lin}(t) \oplus P^p(\mathbb{O}_{lin}(t))$$

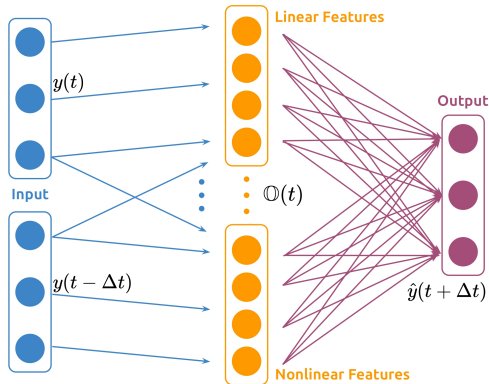


Figure 3: NGRC Architecture, [Ref: Wang, H., Hu, J., Baek, Y., Tsuchiyama, K., Joly, M., Liu, Q., & Gigan, S. (2025). Optical next generation reservoir computing. *Light*, 14(1), 245.]

Next-Generation RC (NGRC) & Hybrid Models

- **NGRC:** Replaces the random reservoir with deterministic Nonlinear Vector Autoregression (NVAR).

- **NVAR Features:**

$$\mathbb{O}(t) = 1 \oplus \mathbb{O}_{lin}(t) \oplus P^p(\mathbb{O}_{lin}(t))$$

- **Hybrid RC + NGRC:** Concatenates random reservoir states with NVAR features.

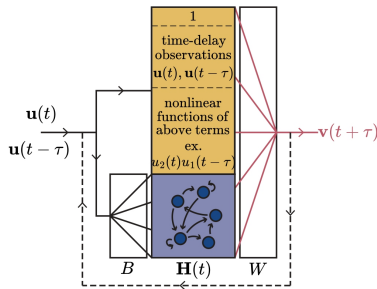


Figure 4: Hybrid Architecture, [Ref: Chepuri, R., Amzalag, D., Antonsen, T. M., & Girvan, M. (2024). Hybridizing traditional and next-generation reservoir computing to accurately and efficiently forecast dynamical systems. *Chaos*, 34(6).]

Next-Generation RC (NGRC) & Hybrid Models

- ▶ **NGRC:** Replaces the random reservoir with deterministic Nonlinear Vector Autoregression (NVAR).
- ▶ **NVAR Features:**
$$\mathbb{O}(t) = 1 \oplus \mathbb{O}_{lin}(t) \oplus P^p(\mathbb{O}_{lin}(t))$$
- ▶ **Hybrid RC + NGRC:** Concatenates random reservoir states with NVAR features.
- ▶ **Limitations:** Excess degrees of freedom amplify small errors; Can lead to *catastrophic divergence* during autonomous rollout.

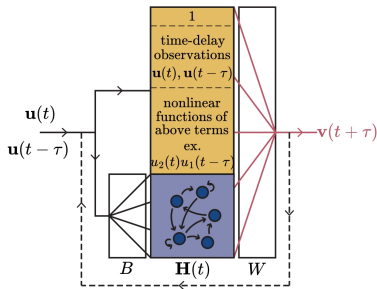


Figure 4: Hybrid Architecture, [Ref: Chepuri, R., Amzalag, D., Antonsen, T. M., & Girvan, M. (2024). Hybridizing traditional and next-generation reservoir computing to accurately and efficiently forecast dynamical systems. *Chaos*, 34(6).]

Next-Generation RC (NGRC) & Hybrid Models

- ▶ **NGRC:** Replaces the random reservoir with deterministic Nonlinear Vector Autoregression (NVAR).
- ▶ **NVAR Features:**
$$\mathbb{O}(t) = 1 \oplus \mathbb{O}_{lin}(t) \oplus P^p(\mathbb{O}_{lin}(t))$$
- ▶ **Hybrid RC + NGRC:** Concatenates random reservoir states with NVAR features.
- ▶ **Limitations:** Excess degrees of freedom amplify small errors; Can lead to *catastrophic divergence* during autonomous rollout.

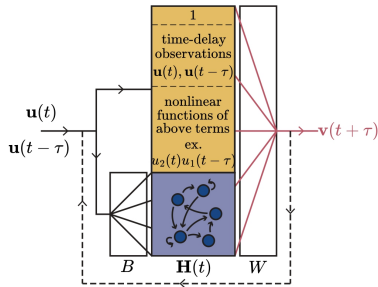
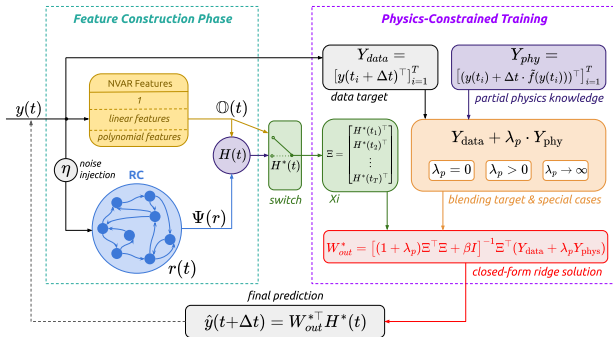


Figure 4: Hybrid Architecture

➡ **Hybrids improve expressiveness, but structural instability makes them unreliable for long - term prediction.**

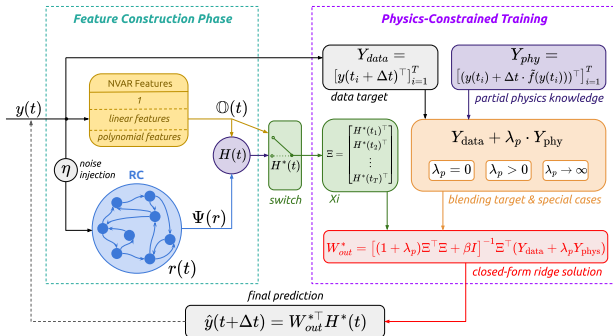
Physics-Informed Reservoir Computing (PIRC)

- **Idea:** Embed physical priors directly into a *closed-form* ridge regression readout.



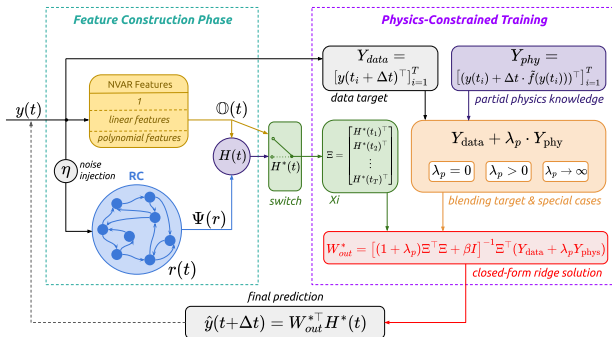
Physics-Informed Reservoir Computing (PIRC)

- **Idea:** Embed physical priors directly into a *closed-form* ridge regression readout.
- Replaces thousands of iterative backpropagation steps with a **single-pass matrix solve**.



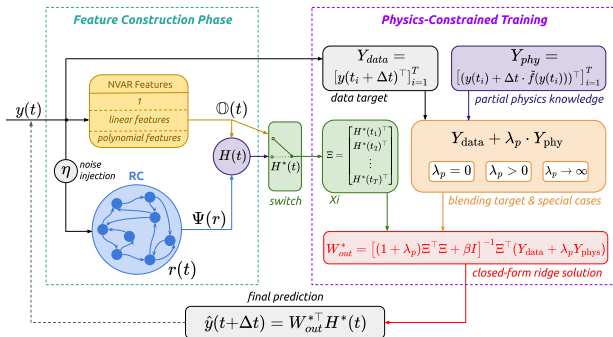
Physics-Informed Reservoir Computing (PIRC)

- **Idea:** Embed physical priors directly into a *closed-form* ridge regression readout.
- Replaces thousands of iterative backpropagation steps with a **single-pass matrix solve**.
- **Three Sequential Phases:** Feature construction phase; Adaptive switching mechanism; Physics-constrained training



Physics-Informed Reservoir Computing (PIRC)

- **Idea:** Embed physical priors directly into a *closed-form* ridge regression readout.
- Replaces thousands of iterative backpropagation steps with a **single-pass matrix solve**.
- **Three Sequential Phases:** Feature construction phase; Adaptive switching mechanism; Physics-constrained training

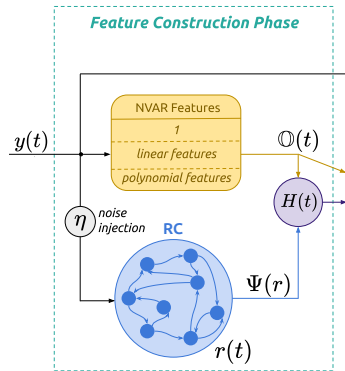


➡ **PIRC delivers the physical consistency of PINNs with the closed-form and lightweight training of RC.**

PIRC Phase 1: Feature Construction

- **NVAR Branch (Deterministic):** Interpretable polynomial basis over time-delayed observations

$$\mathbb{O}(t) = 1 \oplus \mathbb{O}_{lin}(t) \oplus P^p(\mathbb{O}_{lin}(t))$$



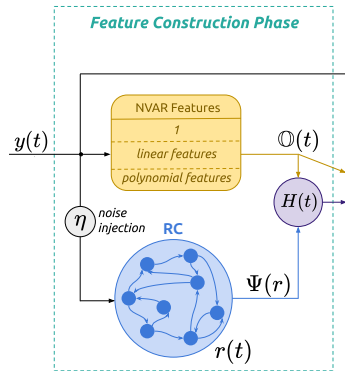
PIRC Phase 1: Feature Construction

- **NVAR Branch (Deterministic):** Interpretable polynomial basis over time-delayed observations

$$\mathbb{O}(t) = 1 \oplus \mathbb{O}_{lin}(t) \oplus P^p(\mathbb{O}_{lin}(t))$$

- **Reservoir Branch (Stochastic):** Infinite-impulse-response (IIR) fading memory of input history.

$$r(t) = (1 - \alpha)r(t - \Delta t) + \alpha \tanh(Ar(t - \Delta t) + By(t) + c)$$



PIRC Phase 1: Feature Construction

- **NVAR Branch (Deterministic):** Interpretable polynomial basis over time-delayed observations

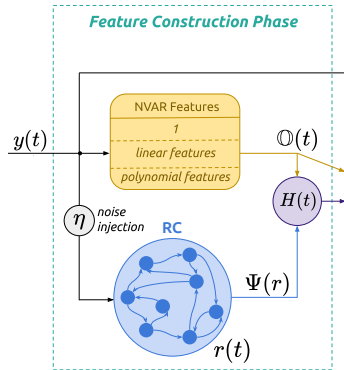
$$\mathbb{O}(t) = 1 \oplus \mathbb{O}_{lin}(t) \oplus P^p(\mathbb{O}_{lin}(t))$$

- **Reservoir Branch (Stochastic):** Infinite-impulse-response (IIR) fading memory of input history.

$$r(t) = (1 - \alpha)r(t - \Delta t) + \alpha \tanh(Ar(t - \Delta t) + By(t) + c)$$

- **Hybrid Representation:** To break symmetry

$$H(t) = \Psi(r(t)) \oplus \mathbb{O}(t); \quad \Psi(r(t))_j = \begin{cases} r_j(t), & \text{if } j \text{ is odd,} \\ r_j(t)^2, & \text{if } j \text{ is even.} \end{cases}$$



PIRC Phase 1: Feature Construction

- **NVAR Branch (Deterministic):** Interpretable polynomial basis over time-delayed observations

$$\mathbb{O}(t) = 1 \oplus \mathbb{O}_{lin}(t) \oplus P^p(\mathbb{O}_{lin}(t))$$

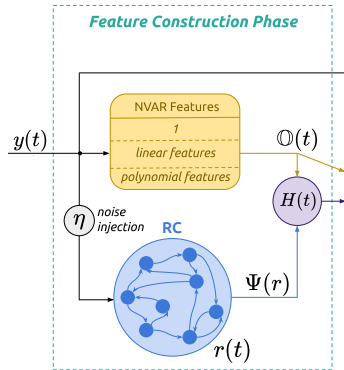
- **Reservoir Branch (Stochastic):** Infinite-impulse-response (IIR) fading memory of input history.

$$r(t) = (1 - \alpha)r(t - \Delta t) + \alpha \tanh(Ar(t - \Delta t) + By(t) + c)$$

- **Hybrid Representation:** To break symmetry

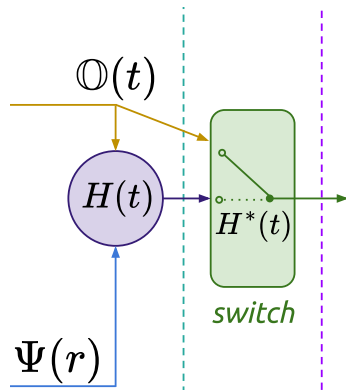
$$H(t) = \Psi(r(t)) \oplus \mathbb{O}(t); \quad \Psi(r(t))_j = \begin{cases} r_j(t), & \text{if } j \text{ is odd,} \\ r_j(t)^2, & \text{if } j \text{ is even.} \end{cases}$$

- ➡ **Parallel branches capture both explicit polynomial structure and long-term nonlinear memory.**



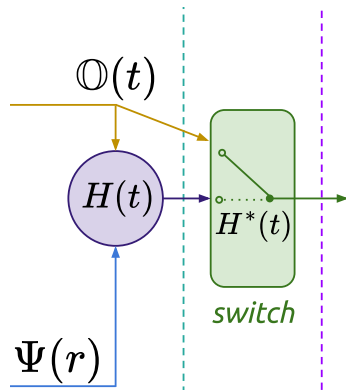
PIRC Phase 2: Adaptive Switching Mechanism

- **Problem:** Blindly using the full hybrid $H(t)$ causes over-parameterization.
 - In low-dimensional or data-scarce regimes, excess degrees of freedom amplify errors.
 - Leads to *catastrophic divergence* during autonomous rollout.



PIRC Phase 2: Adaptive Switching Mechanism

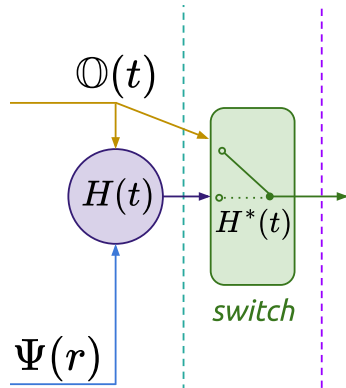
- **Problem:** Blindly using the full hybrid $H(t)$ causes over-parameterization.
 - In low-dimensional or data-scarce regimes, excess degrees of freedom amplify errors.
 - Leads to *catastrophic divergence* during autonomous rollout.
- **Solution:** Treat the feature representation as a discrete hyperparameter.



PIRC Phase 2: Adaptive Switching Mechanism

- **Problem:** Blindly using the full hybrid $H(t)$ causes over-parameterization.
 - In low-dimensional or data-scarce regimes, excess degrees of freedom amplify errors.
 - Leads to *catastrophic divergence* during autonomous rollout.
- **Solution:** Treat the feature representation as a discrete hyperparameter.
- **Mechanism:** Joint grid search.

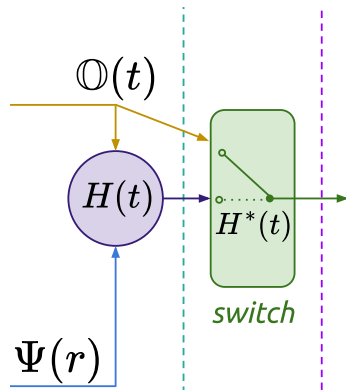
$$H^*(t) = \begin{cases} H(t), & \text{if } E_H \leq E_{\mathbb{O}} \\ \mathbb{O}(t), & \text{otherwise} \end{cases}$$



PIRC Phase 2: Adaptive Switching Mechanism

- **Problem:** Blindly using the full hybrid $H(t)$ causes over-parameterization.
 - In low-dimensional or data-scarce regimes, excess degrees of freedom amplify errors.
 - Leads to *catastrophic divergence* during autonomous rollout.
- **Solution:** Treat the feature representation as a discrete hyperparameter.
- **Mechanism:** Joint grid search.

$$H^*(t) = \begin{cases} H(t), & \text{if } E_H \leq E_{\mathbb{O}} \\ \mathbb{O}(t), & \text{otherwise} \end{cases}$$

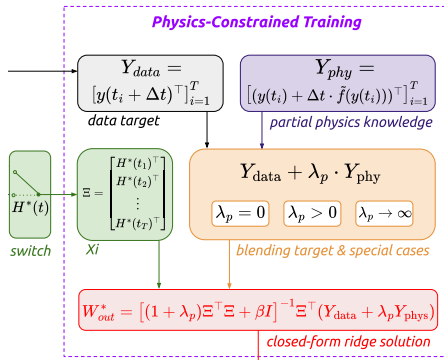


- ➡ **Autonomously selects the minimal sufficient feature set to prevent divergence and minimize parameters.**

PIRC Phase 3: Physics-Constrained Training

► Two Training Targets:

- *Data Target* (Y_{data}): Next observed state.
- *Physics Target* (Y_{phy}): State predicted by governing equation $\tilde{f}(y)$ (discretized).



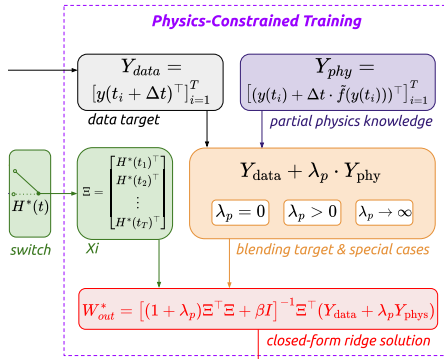
PIRC Phase 3: Physics-Constrained Training

► Two Training Targets:

- *Data Target* (Y_{data}): Next observed state.
- *Physics Target* (Y_{phy}): State predicted by governing equation $\tilde{f}(y)$ (discretized).

► Combined Loss Function:

$$\mathcal{L}(W) = \|\Xi W - Y_{data}\|_F^2 + \lambda_p \|\Xi W - Y_{phy}\|_F^2 + \beta \|W\|_F^2$$



PIRC Phase 3: Physics-Constrained Training

► Two Training Targets:

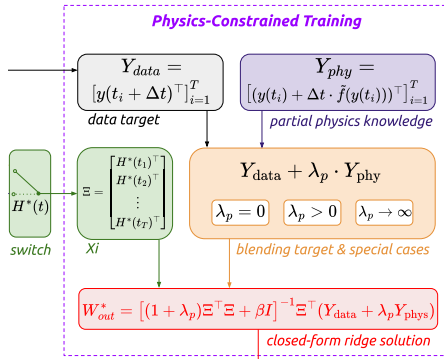
- *Data Target* (Y_{data}): Next observed state.
- *Physics Target* (Y_{phy}): State predicted by governing equation $\tilde{f}(y)$ (discretized).

► Combined Loss Function:

$$\mathcal{L}(W) = \|\Xi W - Y_{data}\|_F^2 + \lambda_p \|\Xi W - Y_{phy}\|_F^2 + \beta \|W\|_F^2$$

► Closed-Form Solution: Since $\mathcal{L}(W)$ is convex, set $\nabla_W \mathcal{L} = 0$:

$$W_{out}^* = [(1 + \lambda_p)\Xi^T \Xi + \beta I]^{-1} \Xi^T (Y_{data} + \lambda_p Y_{phy})$$



PIRC Phase 3: Physics-Constrained Training

► Two Training Targets:

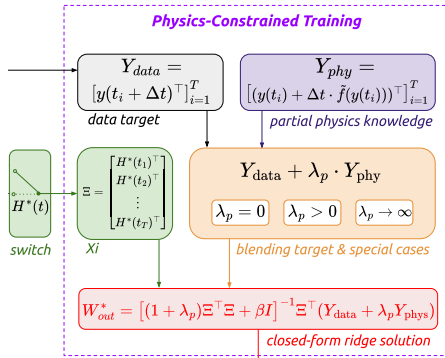
- *Data Target* (Y_{data}): Next observed state.
- *Physics Target* (Y_{phy}): State predicted by governing equation $\tilde{f}(y)$ (discretized).

► Combined Loss Function:

$$\mathcal{L}(W) = \|\Xi W - Y_{data}\|_F^2 + \lambda_p \|\Xi W - Y_{phy}\|_F^2 + \beta \|W\|_F^2$$

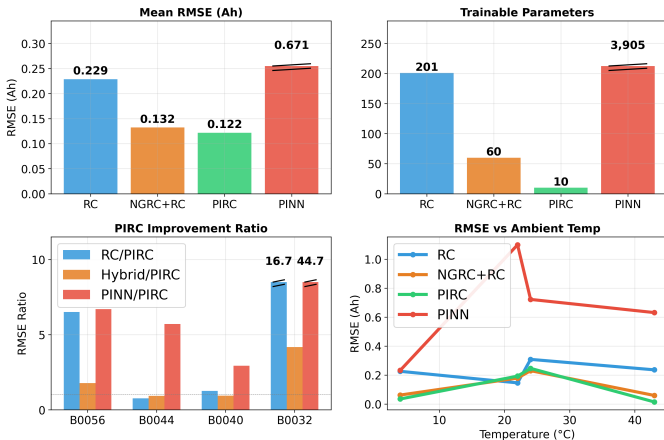
► Closed-Form Solution: Since $\mathcal{L}(W)$ is convex, set $\nabla_W \mathcal{L} = 0$:

$$W_{out}^* = [(1 + \lambda_p)\Xi^T \Xi + \beta I]^{-1} \Xi^T (Y_{data} + \lambda_p Y_{phy})$$



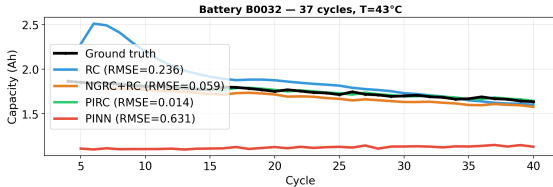
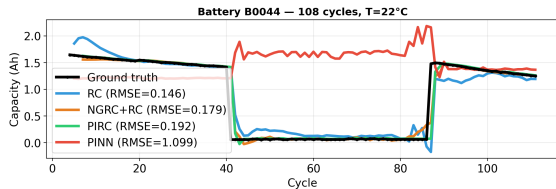
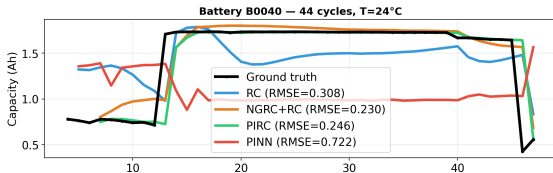
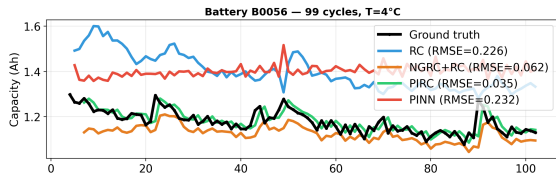
➡ **A single-pass matrix solve replaces thousands of iterative backpropagation steps.**

NASA PCoE Battery SOH Estimation

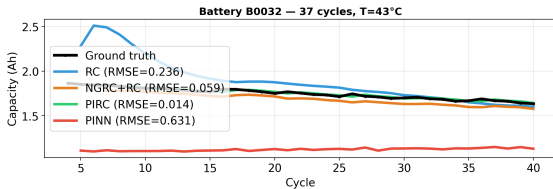
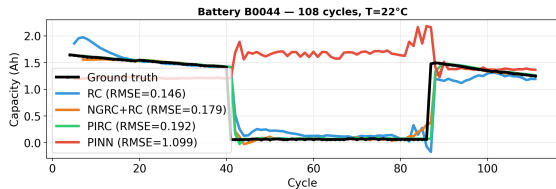
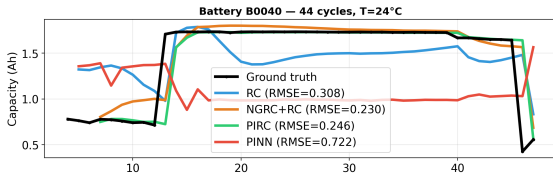
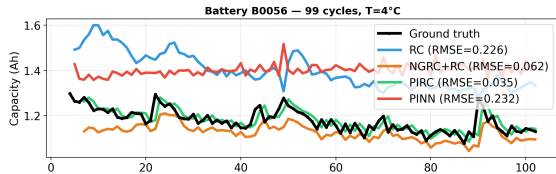


➡ **PIRC achieves the lowest RMSE with only 10 trainable parameters—a 390× reduction compared to PINNs.**

Per-Battery SOH Capacity Tracking



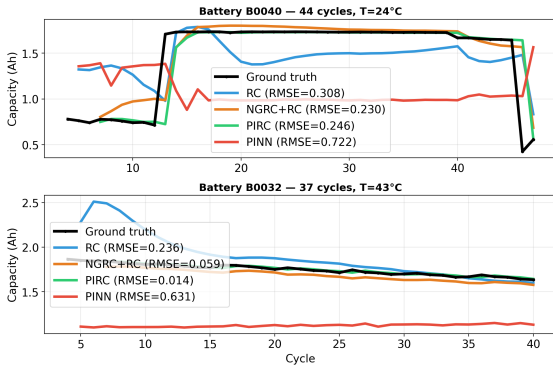
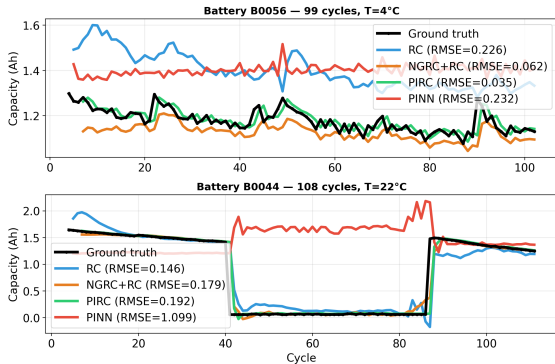
Per-Battery SOH Capacity Tracking



MAE	RC	NGRC+RC	PINN	PIRC
Mean	0.188	0.067	0.597	0.036

Table 2: Mean MAE across different Temperatures

Per-Battery SOH Capacity Tracking



MAE	RC	NGRC+RC	PINN	PIRC
Mean	0.188	0.067	0.597	0.036

Table 2: Mean MAE across different Temperatures

➡ **PIRC (green) closely follows the ground-truth trajectory across all temperature regimes without oscillatory errors.**

Chaotic Systems

System	Model	RMSE	Corr.	Params	Train
Lorenz	RC	0.443	0.884	303	0.007s
	NGRC+RC	0.235	0.948	384	0.041s
	PINN	1.754	0.072	17,155	5.138s
	PIRC	0.007	1.000	84	0.038s
Rössler	RC	3.949	0.679	303	0.027s
	NGRC+RC	874.027	0.781	384	0.146s
	PINN	0.969	-0.015	17,155	19.440s
	PIRC	0.002	1.000	84	0.136s

Table 3: Autonomous Prediction Results on Chaotic Systems

Chaotic Systems

System	Model	RMSE	Corr.	Params	Train
Lorenz	RC	0.443	0.884	303	0.007s
	NGRC+RC	0.235	0.948	384	0.041s
	PINN	1.754	0.072	17,155	5.138s
	PIRC	0.007	1.000	84	0.038s
Rössler	RC	3.949	0.679	303	0.027s
	NGRC+RC	874.027	0.781	384	0.146s
	PINN	0.969	-0.015	17,155	19.440s
	PIRC	0.002	1.000	84	0.136s

Table 3: Autonomous Prediction Results on Chaotic Systems

➡ **PIRC maintains stable autonomous rollout where standard hybrid models fail catastrophically.**

Summary

- ▶ **Lightweight & Closed-Form:** Replaced iterative backpropagation with a single-pass matrix solve, reducing trainable parameters by up to $390\times$ compared to PINNs.

Summary

- ▶ **Lightweight & Closed-Form:** Replaced iterative backpropagation with a single-pass matrix solve, reducing trainable parameters by up to $390\times$ compared to PINNs.
- ▶ **Physics-Informed:** Embedded physical priors directly into the ridge regression readout, ensuring physically consistent predictions without heavy computational overhead.

Summary

- ▶ **Lightweight & Closed-Form:** Replaced iterative backpropagation with a single-pass matrix solve, reducing trainable parameters by up to $390\times$ compared to PINNs.
- ▶ **Physics-Informed:** Embedded physical priors directly into the ridge regression readout, ensuring physically consistent predictions without heavy computational overhead.
- ▶ **Adaptive & Stable:** Introduced a switching mechanism that autonomously prunes unnecessary reservoir features, preventing the catastrophic divergence seen in standard hybrid models.

Summary

- ▶ **Lightweight & Closed-Form:** Replaced iterative backpropagation with a single-pass matrix solve, reducing trainable parameters by up to $390\times$ compared to PINNs.
- ▶ **Physics-Informed:** Embedded physical priors directly into the ridge regression readout, ensuring physically consistent predictions without heavy computational overhead.
- ▶ **Adaptive & Stable:** Introduced a switching mechanism that autonomously prunes unnecessary reservoir features, preventing the catastrophic divergence seen in standard hybrid models.
- ▶ **Future Work:** Extending the framework with formal stability guarantees and a Bayesian filtering extension for uncertainty quantification.

➡ **PIRC enables high-fidelity, physically consistent battery health prediction on resource-constrained edge devices.**

Thank you for listening!

Questions & Discussion



OPTIMIZATION AND ESTIMATION LAB

ONE.UNM.EDU