IEEE/ACM ICCAD 2024 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN

43rd Edition

Pseudo Adjoint Optimization: Harnessing the Solution Curve for SPICE Acceleration

Jiatai Sun[1], Xiaru Zha[1], Chao Wang[2], Xiao Wu[3], Dan Niu[2], Wei Xing[4] and Zhou Jin[1]

[1]Super Scientific Software Laboratory, China University of Petroleum-Beijing

[2]School of Automation, Southeast University

[3]Huada Empyrean Software Co. Ltd

[4]SoMaS, The University of Sheffield

jinzhou@cup.edu.cn

Outline

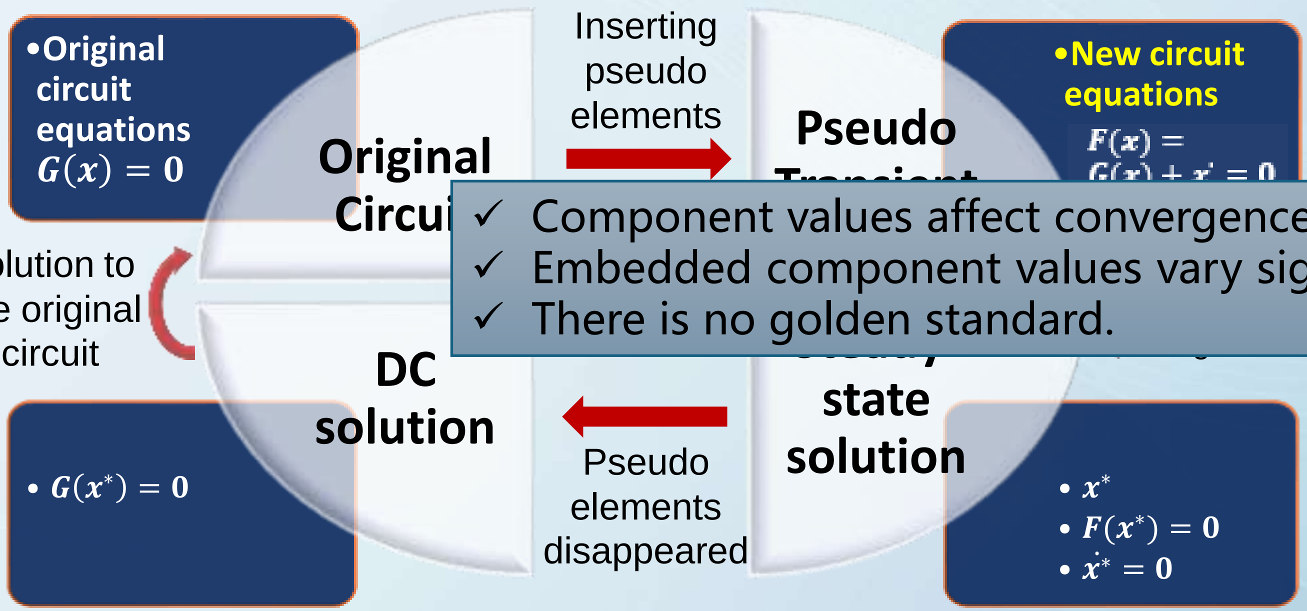
- **Background and related work**
- **Soda-PTA: Harnessing the Solution Curve for SPICE Acceleration**
  - Soda-PTA Framework
  - Forward Process Design
  - Backward Process Design
  - GCN Design
- **Experiment**
  - Fitting Solution Curve Effect of Neural ODE
  - Optimization Performance of Soda-PTA
  - Convergence Performance of Soda-PTA
  - Optimization Performance of Soda-PTA with GCN
  - Optimization Performance of Soda-PTA with RL-S

Outline

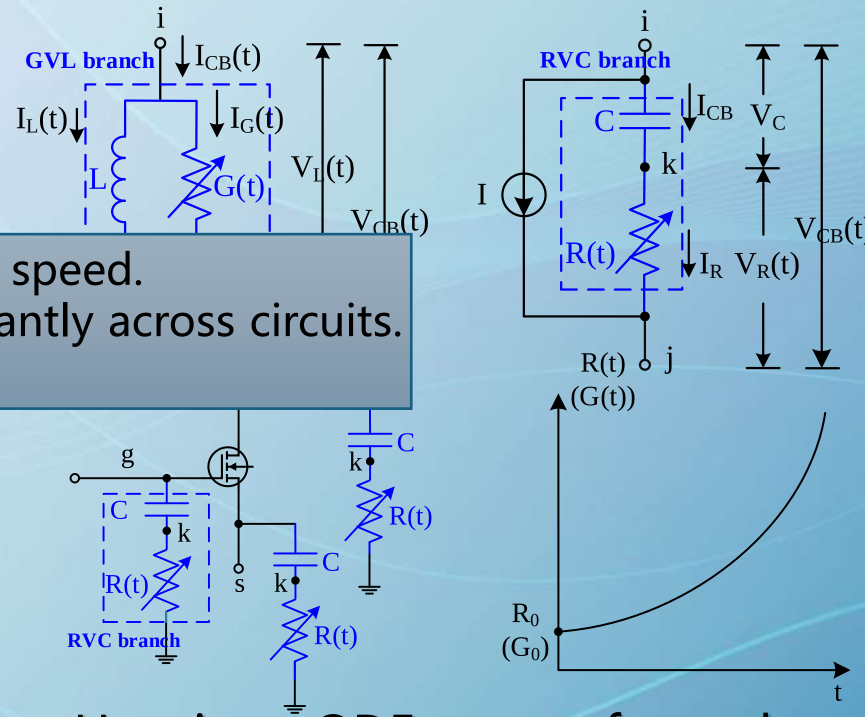
- **Background and related work**
- **Soda-PTA: Harnessing the Solution Curve for SPICE Acceleration**
  - Soda-PTA Framework
  - Forward Process Design
  - Backward Process Design
  - GCN Design
- **Experiment**
  - Fitting Solution Curve Effect of Neural ODE
  - Optimization Performance of Soda-PTA
  - Convergence Performance of Soda-PTA
  - Optimization Performance of Soda-PTA with GCN
  - Optimization Performance of Soda-PTA with RL-S

Background: PTA Methods

Pseudo Transient Analysis (PTA) is currently the **most powerful and promising** numerical solving algorithm in SPICE circuit simulation for DC analysis, as it is easy to implement and has **good continuity and convergence**.



- ✓ Component values affect convergence and speed.
- ✓ Embedded component values vary significantly across circuits.
- ✓ There is no golden standard.

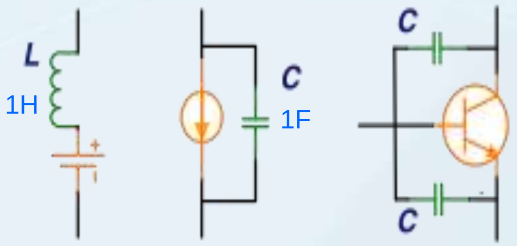


- Inserting capacitors can effectively address discontinuity issues, but it introduces oscillation problems and increases computation time.
- How is an ODE system formed under the PTA method?

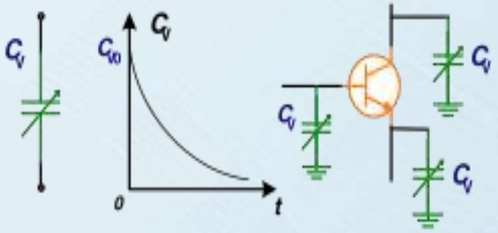
Related Work: More Easily Solvable System

What components should be embedded?

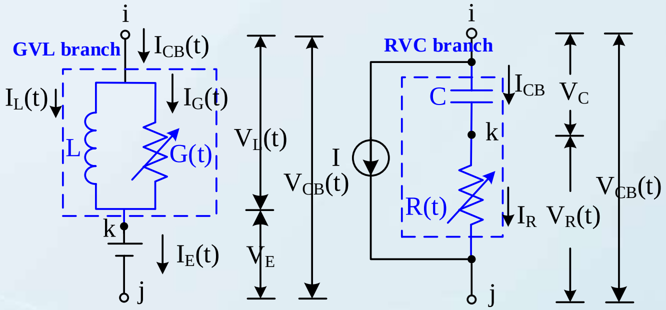
- **Pure PTA/DPTA[1][2]**



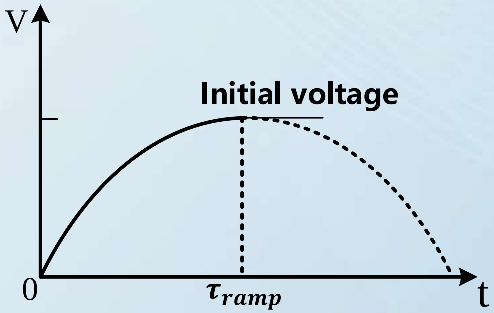
- **Time-varying PTA[3]**



- **CEPTA[4]**



- **Ramping PTA[5]**



What parameters should be embedded

- An adaptive dynamic-element PTA method for solving nonlinear DC operating point of transistor circuits[6]

  The algorithm inserts dynamic pseudo-elements for each transistor, with values that change independently and automatically based on the simulation state.

- BoA-PTA: A Bayesian Optimization Accelerated PTA Solver for SPICE Simulation[7]

  The PTA algorithm based on Bayesian optimization is the first application of machine learning in SPICE Solvers.

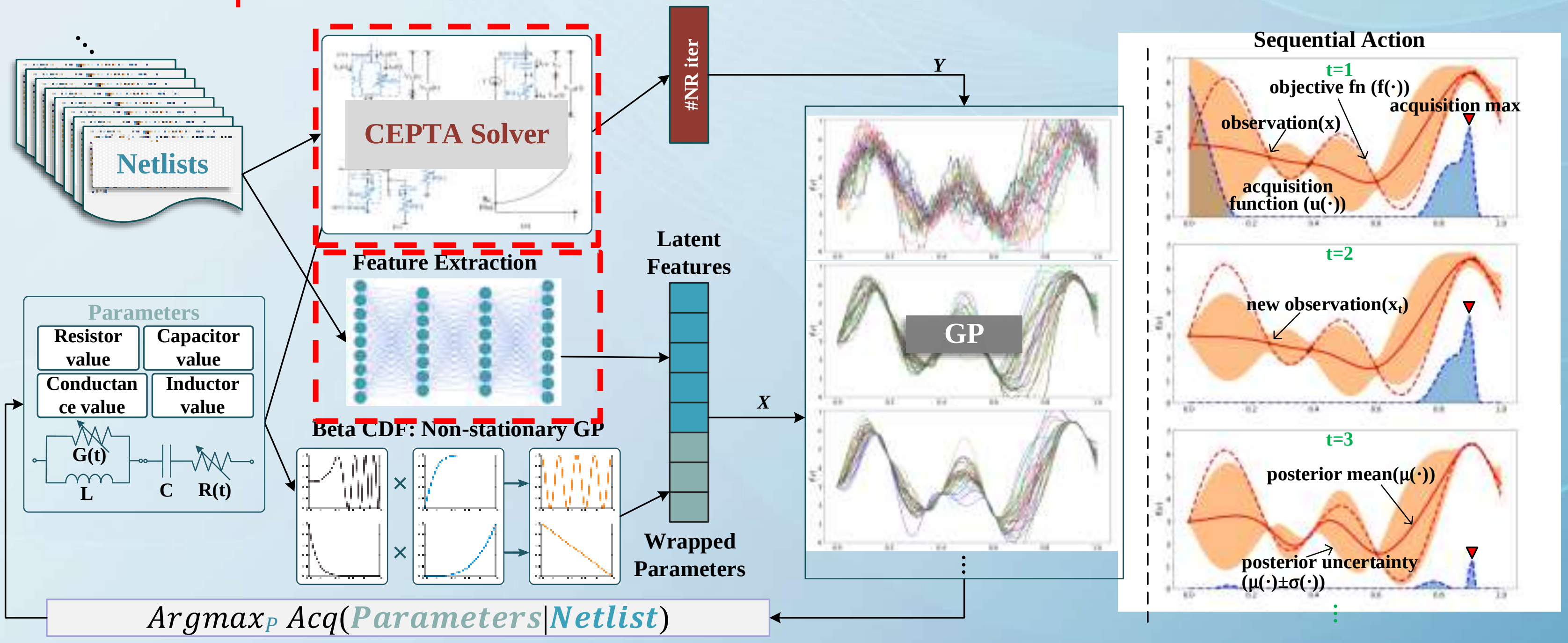
[1] W. Weeks, A. Jimenez, G. Mahoney, D. Mehta, H. Qassemzadeh and T. Scott, Algorithms for ASTAP--A network-analysis program, IEEE Trans. Circuits Theory, 1973.
[2] X. Wu, Z. Jin, and Y. Inoue. Numerical integration algorithms with artificial damping for the pta method applied to dc analysis of nonlinear circuits. In ICCCAS, 2013.
[3] R. Wilton, Supplementary algorithms for DC convergence, IEE Colloquium, SPICE: Surviving Problems in Circuit Evaluation, 1993.
[4] H. Yu, Y. Inoue, K. Sako, X. Hu, and Z. Huang. An effective spice3 implementation of the compound element pseudo-transient algorithm. IEICE Trans. Fundam. Electron. Commun. Comput. Sci, 2007.
[5] Z. Jin, X. Wu, Y. Inoue, and N. Dan. A ramping method combined with the damped pta algorithm to find the dc operating points for nonlinear circuits. In ISIC, 2014.
[6] Z. Jin, M. Liu, and X. Wu. An adaptive dynamic-element pta method for solving nonlinear dc operating point of transistor circuits. In MWSCAS, 2018.
[7] W. W. Xing, X. Jin, T. Feng, D. Niu, W. Zhao, and Z. Jin. Boa-pta: A Bayesian optimization accelerated pta solver for spice simulation. ACM TODATES, 2022.

Related Work: BOA-PTA

- **BoA-PTA[1]: A Bayesian Optimization Accelerated PTA Solver for SPICE Simulation**

  Drawbacks:

  - Treat the SPICE simulation process as a black box, without utilizing key information from the simulation process.
  - Only a simple neural network was used for circuit feature extraction, and the circuit topology information was not captured.



[1] W. W. Xing, X. Jin, T. Feng, D. Niu, W. Zhao, and Z. Jin. Boa-pta: A Bayesian optimization accelerated pta solver for spice simulation. ACM TODATES, 2022.

Outline

- **Background and related work**
- **Soda-PTA: Harnessing the Solution Curve for SPICE Acceleration**
  - Soda-PTA Framework
  - Forward Process Design
  - Backward Process Design
  - GCN Design
- **Experiment**
  - Fitting Solution Curve Effect of Neural ODE
  - Optimization Performance of Soda-PTA
  - Convergence Performance of Soda-PTA
  - Optimization Performance of Soda-PTA with GCN
  - Optimization Performance of Soda-PTA with RL-S

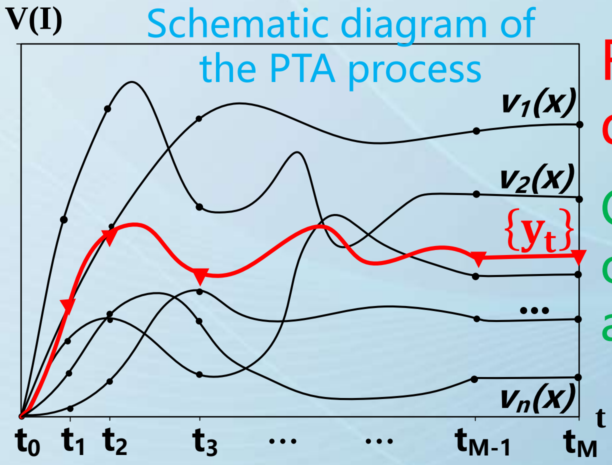
Soda-PTA: Core Idea

Soda-PTA ***uses Neural ODE to model the PTA process, thereby approximating it as a surrogate to obtain gradient information on the simulation performance with respect to the PTA parameters*****.**

In SPICE circuit simulation, we can describe the PTA method as:

$$(NR\_iters; M; \{x_t\}_{t=1}^{M}) = PTA(\xi, \theta)$$

Character term	Meaning
PTA(·)	PTA solver, PTA execution process
ξ	circuit netlist
$\boldsymbol{\theta}$	Parameters inserted in the PTA process, PTA hyperparameters
M	Total number of steps for discrete numerical integration in PTA
x	Solution vector of the ODE system at each time point during the PTA
NR_iters	Total number of NR iterations during the PTA, the key performance metric



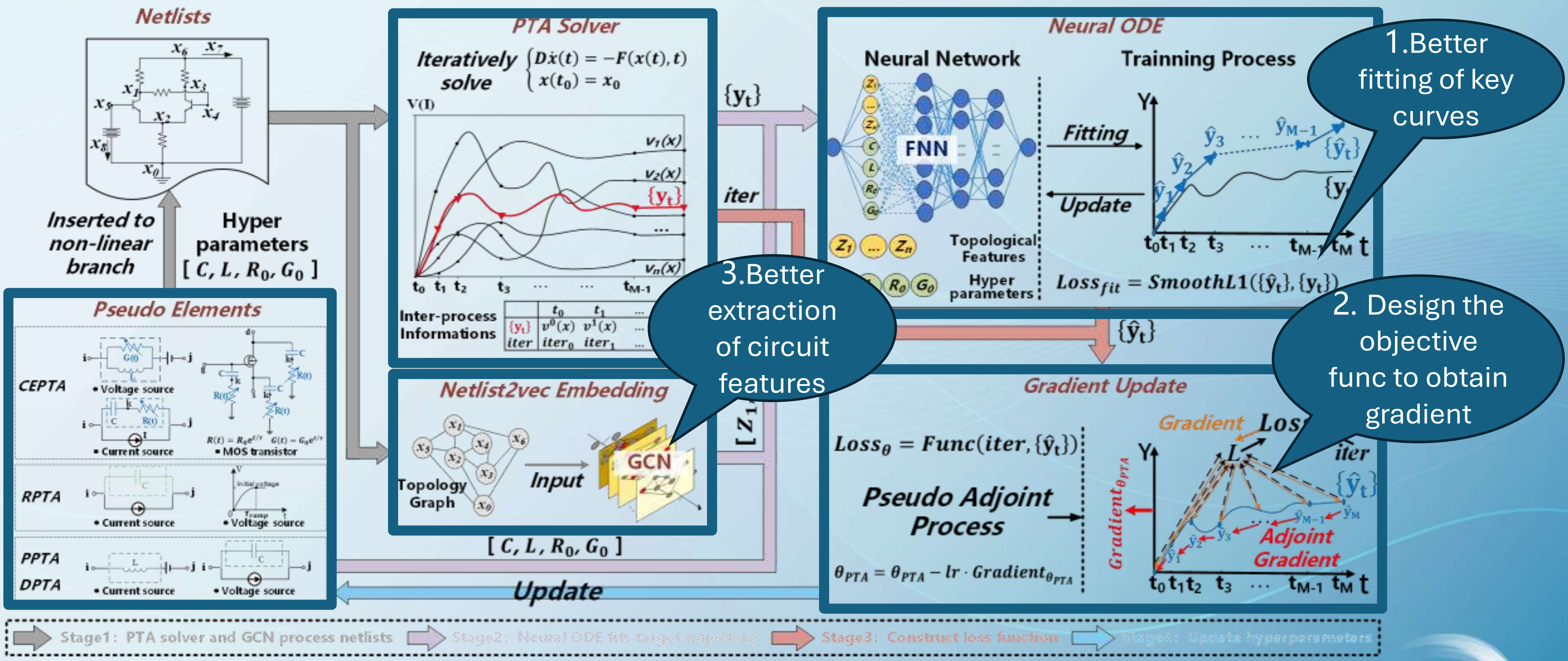
For any circuit ξ, how to adjust θ to minimize NR_iters?

Can we use gradient information to guide the PTA hyperparameter updates to minimize NR_iters?

The PTA(·) process is difficult to trace, and deriving gradient information with-in it is a challenge.

- BoA-PTA Bayesian optimization uses Gaussian processes as a surrogate model to learn the relationship between $\boldsymbol{\theta}$ and $\boldsymbol{NR\_iters}$. **Useful information from $\{x_t\}$ is ignored.**
- Soda-PTA Use Neural ODE as a surrogate model to approximate $PTA(\cdot)$.

Soda-PTA: Framework



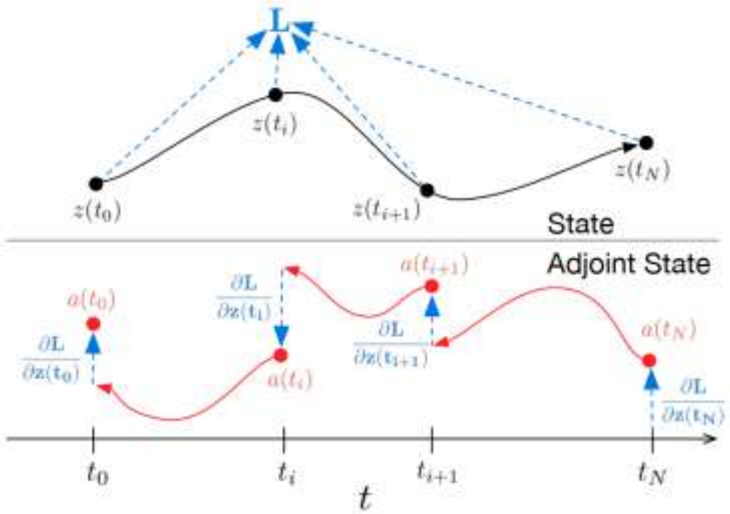
Soda-PTA: Forward Process Design

Process Modeling and Target Curve Fitting

Neural ODE[1] is a neural network that learns the derivative of the hidden layer state with respect to time.

$$\dot{z}(t) = f_w(z(t), t)$$

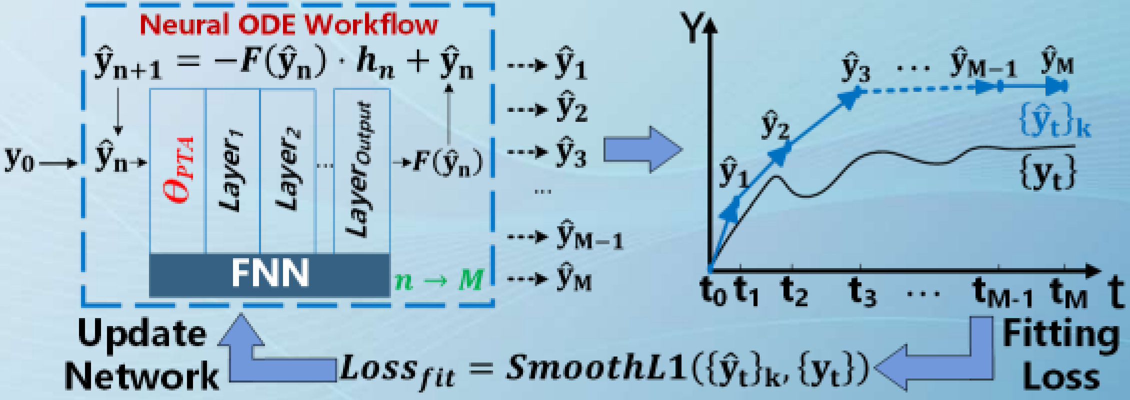
Where, $z(t)$ is the hidden layer state, $\dot{z}$ is the derivative of z with respect to time, and f_w is a neural network model parameterized by w.



Neural ODE Forward

Neural ODE Backward

The key to forward design is fitting the target simulation curve to imitate the behavior of the PTA solver, thereby approximating it as a surrogate.



Objective ODE System

$$G(x) + D\dot{x} = 0$$

Equivalent Form

$$F(x) + \dot{x} = 0$$
$$x = x(t)$$

Map the solution vector to the state vector of the NODE, state evolution process of the NODE is equivalent to the PTA process.

[1] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In NIPS, 2018.

Soda-PTA: Backward Process Design

Gradient Calculation and Pseudo-Adjoint Optimization

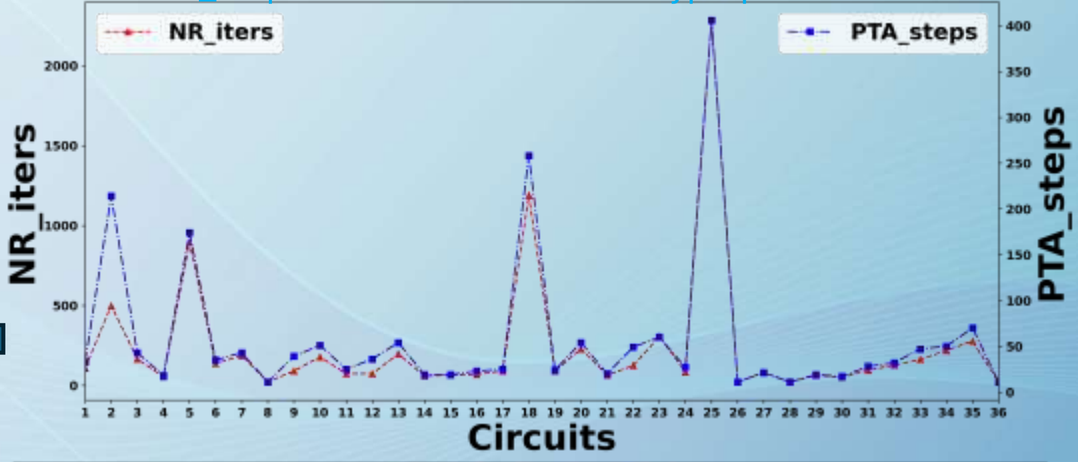
$$NR\_iters; M; \{\mathbf{x_t}\}_{t=1}^{M} = PTA(\xi, \boldsymbol{\theta})$$

$$\{\hat{y}_t\} = NODE(FNN, x\_0, t, eluer)$$

Based on the output of the surrogate NODE, $\{\hat{y}_t\}$, how can we design the objective function?

Empirically, there is a strong correlation between PTA_steps and NR_iters

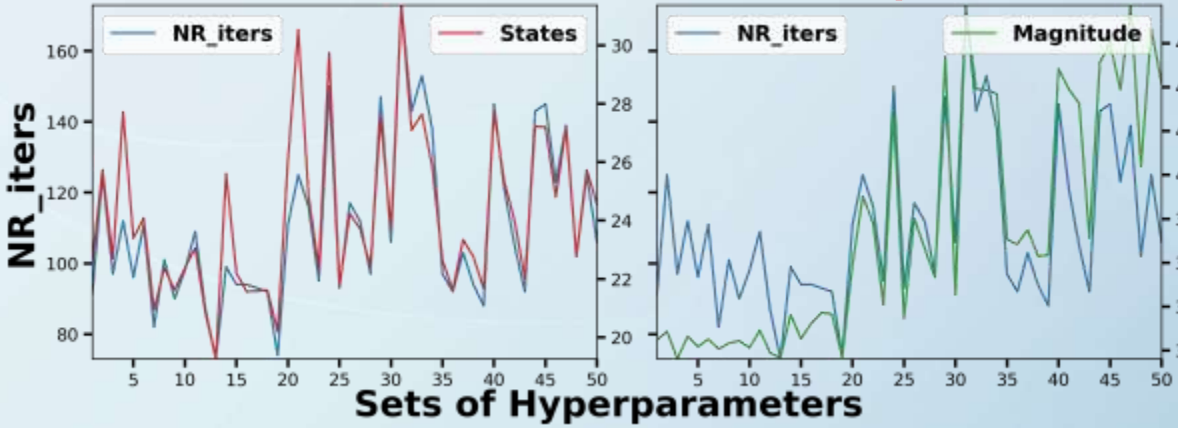
On the benchmark, the relationship between NR_iters and PTA_steps under default CEPTA hyperparameters.

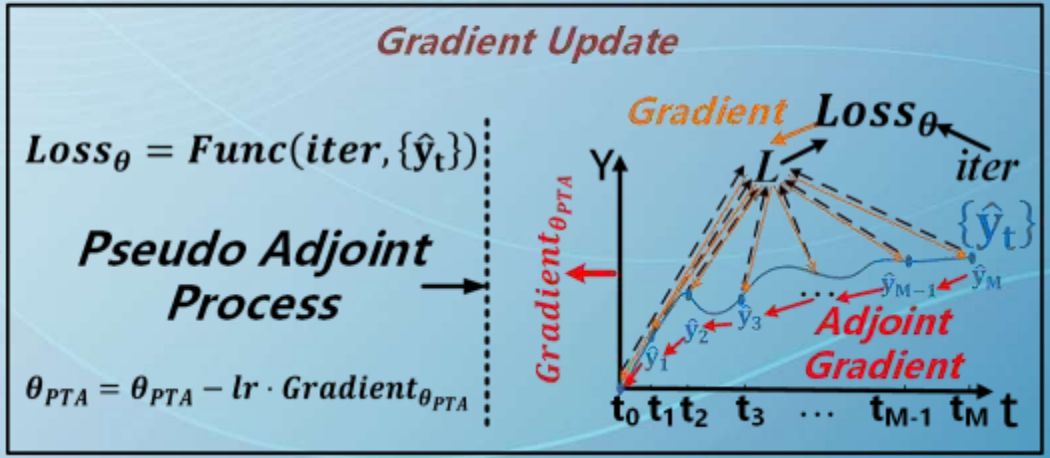


$$loss_{\boldsymbol{\theta}}(\{\hat{\mathbf{y}}_t\}) = NR\_iters \cdot (|\hat{\mathbf{y}}_t|_1 + |(\hat{\mathbf{y}}_t - \widehat{\mathbf{y}_{t-1}})|_1)$$

state cumulative magnitude

NR_iters relationship with state variables and cumulative magnilitude under 'hussamp' circuit

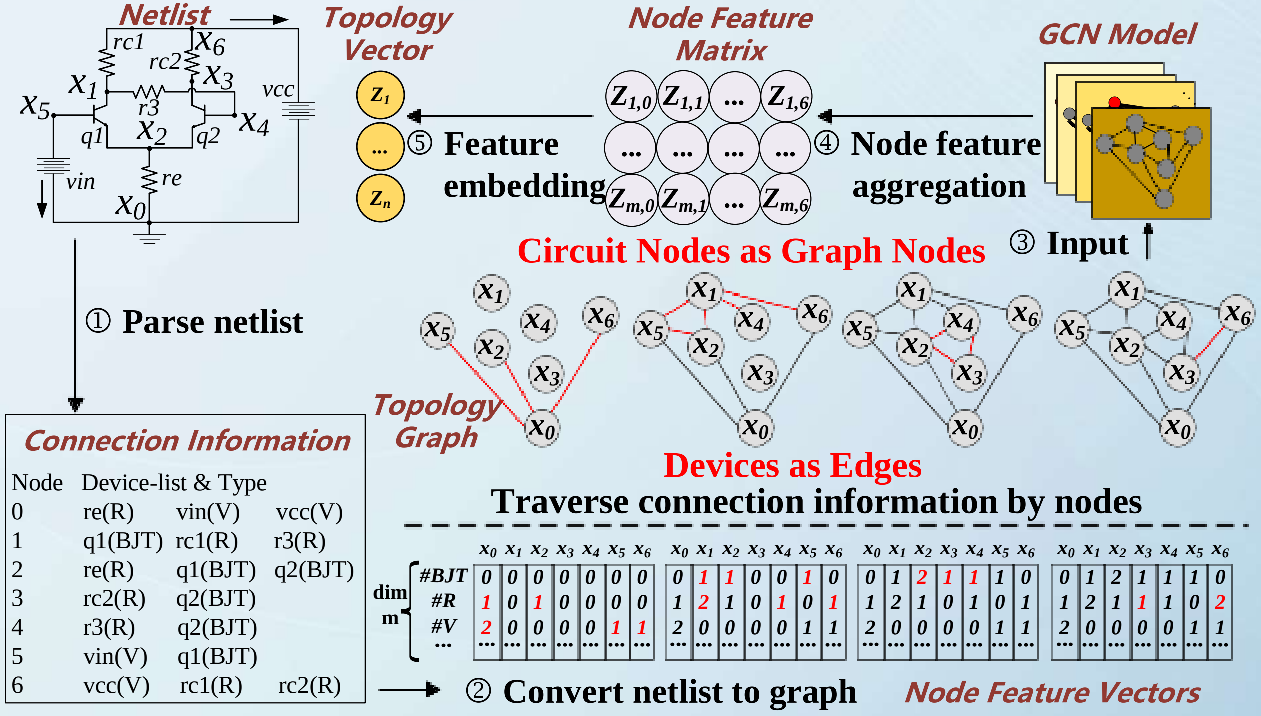




Soda-PTA: GCN Design

Circuit Feature Recognition and Accelerated Parameter Selection

- For any circuit, Soda-PTA_noGNN always starts optimizing PTA hyperparameters from scratch.
- Soda-PTA_noGNN cannot utilize historical experience from similar circuit types.
- **Graph Convolutional Networks (GCNs) can enable better results.**
  - Circuit nodes as vertices.
  - The feature vector represents a node's connected devices.
  - Devices as edges.



Algorithm 2 Soda-PTA for unseen circuit with GCN

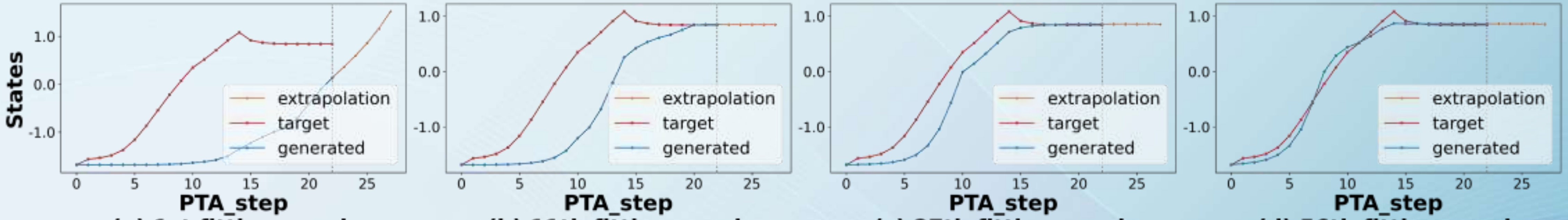
Input: Soda-PTA_noGNN, $GCN(\cdot)$, test set Te, train set Tr, N_{epoch}^{Te}, train iteration N_{epoch}^{Tr}

1: Initilize Neural ODE $f_w(\cdot)$
2: **for** μ in Tr **do**
3: $\quad \xi = \mathrm{GCN}(\mu)$
4: $\quad \theta^*(\xi)$ = Soda-PTA_noGNN($f_w(\cdot), \theta_0, \xi, N_{epoch}^{Tr}$)
5: **end for**
6: Update GCN by the total loss of all generated trajectories
7: **for** μ in Te **do**
8: $\quad \xi = \mathrm{GCN}(\mu)$
9: $\quad \theta^*$ = Soda-PTA_noGNN($f_w(\cdot), \theta_0, \xi, N_{epoch}^{Te}$)
10: **end for**

Outline

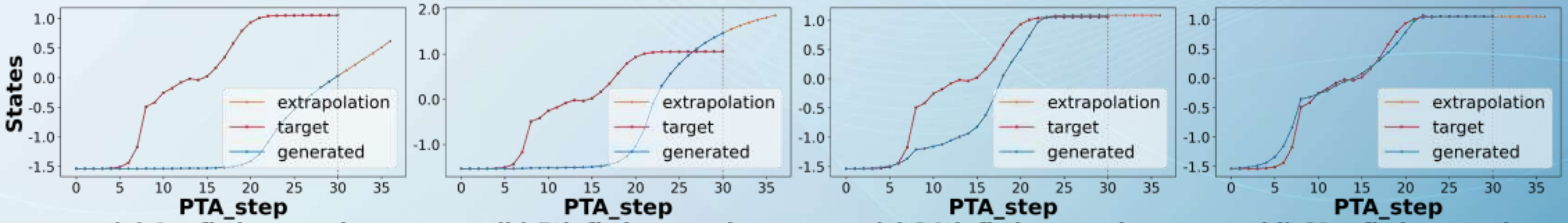
- **Background and related work**
- **Soda-PTA: Harnessing the Solution Curve for SPICE Acceleration**
  - Soda-PTA Framework
  - Forward Process Design
  - Backward Process Design
  - GCN Design
- **Experiment**
  - Fitting Solution Curve Effect of Neural ODE
  - Optimization Performance of Soda-PTA
  - Convergence Performance of Soda-PTA
  - Optimization Performance of Soda-PTA with GCN
  - Optimization Performance of Soda-PTA with RL-S

Experiment: Fitting Solution Curve Effect



(a) 1st fitting epoch (b) 11th fitting epoch (c) 27th fitting epoch (d) 56th fitting epoch

The Neural ODE fitting process of the 'hussamp' circuit under CEPTA



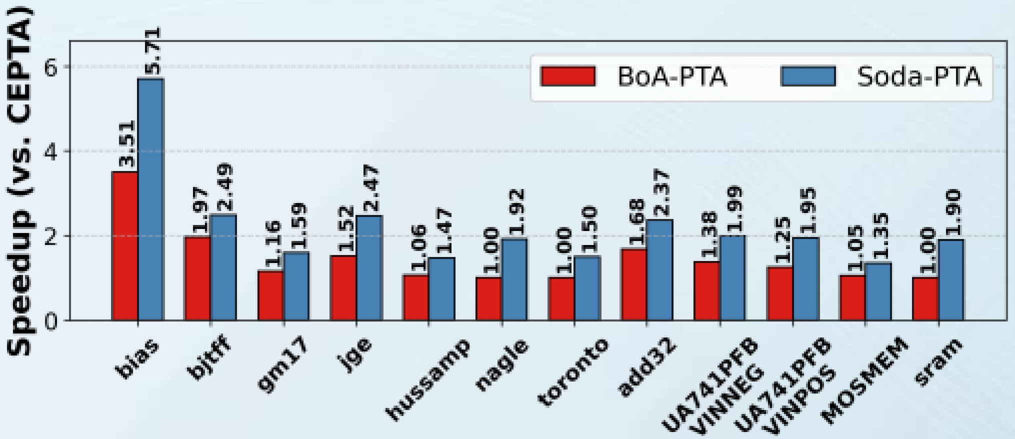
(a) 1st fitting epoch (b) 5th fitting epoch (c) 34th fitting epoch (d) 61st fitting epoch

The Neural ODE fitting process of the '6stageLimAmp' circuit under DPTA

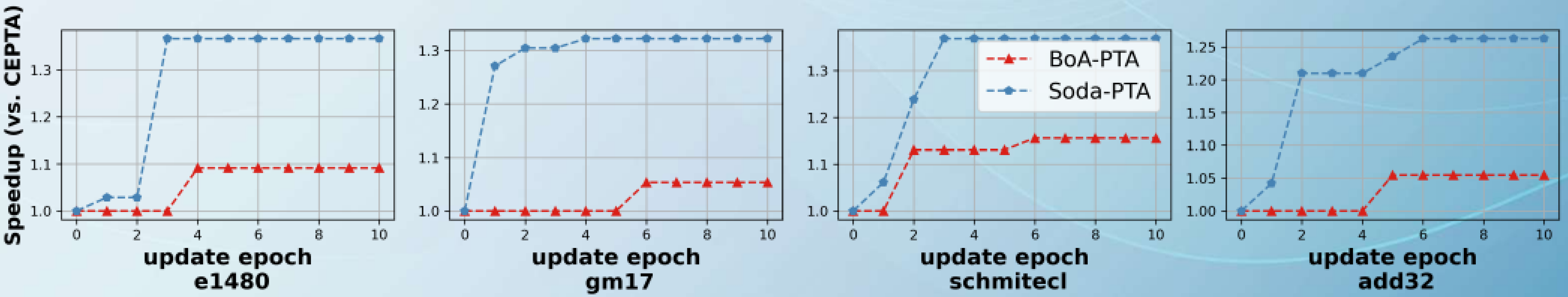
The red curve represents the solution curve of the PTA process in the SPICE simulator, the blue line represents the curve obtained after training the NODE, and the yellow represents the extrapolated results from the NODE.

Experiment: Optimization Performance

- Comparison with BOA-PTA under CEPTA



- Compared to the SOTA method BoA-PTA, it achieves an average speedup of 1.53x across 12 test circuits, with a maximum speedup of 1.9x.
- In the four types of circuits in the benchmark (with four representative circuits shown in the figure), Soda-PTA consistently achieves a better speedup with fewer hyperparameter optimization iterations compared to BoA-PTA.



Experiment: Optimization Performance

- The performance of Soda-PTA under four different PTA methods

circuits	NR_iters								Speedup			
	CEPTA		PPTA		DPTA		RPTA					
	navie	Soda-PTA	navie	Soda-PTA	navie	Soda-PTA	navie	Soda-PTA	v.CEPTA	v.PPTA	v.DPTA	v.RPTA
ab_opamp	150	***110***	—	—	2417	146	2408	127	1.36x	—	16.55x	18.96x
astabl	55	45	108	64	81	43	75	***41***	1.22x	1.69x	1.88x	1.83x
bias	839	147	—	899	755	607	498	***110***	5.71x	—	1.24x	4.53x
bjtinv	186	53	125	77	155	***51***	101	101	3.51x	1.62x	3.04x	1.00x
cram	91	88	—	—	130	100	128	***81***	1.03x	—	1.30x	1.58x
gm6	69	42	—	—	110	55	107	***38***	1.64x	—	2.00x	2.82x
hussamp	91	***62***	—	—	209	87	240	71	1.47x	—	2.40x	3.38x
mosrect	65	***51***	251	53	838	63	837	55	1.27x	4.74x	13.30x	15.22x
nand	83	53	—	***32***	—	142	—	76	1.57x	—	—	—
schmitfast	82	59	71	***30***	5681	106	5678	92	1.39x	2.37x	53.59x	61.72x
6stageLimAmp	137	51	69	***38***	135	73	137	51	2.69x	1.82x	1.85x	2.69x
add32	173	73	—	—	1765	234	1970	***70***	2.37x	—	7.54x	28.14x
DCOSC	126	***78***	108	91	116	98	136	100	1.62x	1.19x	1.18x	1.36x
DIFFPAIR	148	57	101	71	114	109	137	***47***	2.60x	1.42x	1.05x	2.91x
MOSAMP1	122	82	—	139	158	96	162	***69***	1.49x	—	1.65x	2.35x
MOSBandgap	153	***85***	—	—	342	113	341	104	1.80x	—	3.03x	3.28x
MOSMEM	127	***94***	253	98	26029	171	26037	101	1.35x	2.58x	152.22x	257.79x
TADEGLOW	103	63	151	***51***	164	66	86	60	1.63x	2.96x	2.48x	1.43x
UA709	407	***110***	311	143	2985	219	3270	887	3.70x	2.17x	13.63x	3.69x
Multiplier	—	105	—	—	232	***92***	225	94	—	—	2.52x	2.39x
Average									**2.11x**	**2.26x**	**14.77x**	**22.12x**

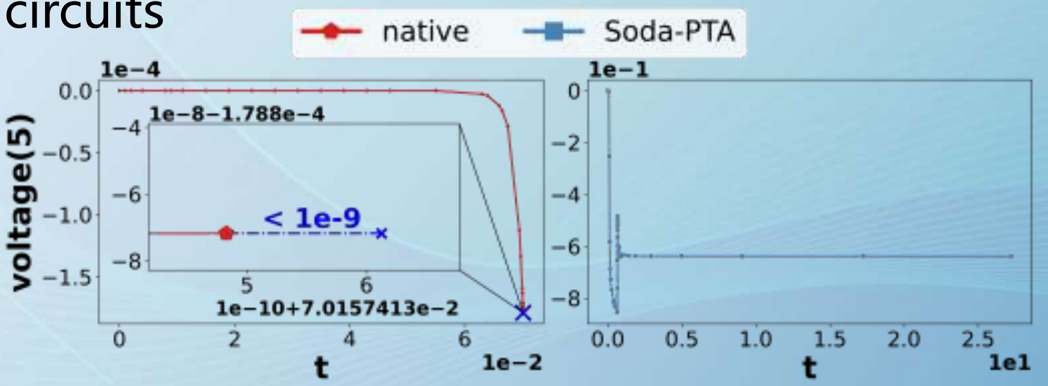
In the table, ***bold italics*** indicate the best results among the four PTA algorithms after Soda-PTA optimization, **all outperforming the original CEPTA**. This shows that Soda-PTA consistently guides optimal parameter selection, regardless of the PTA algorithm chosen.

Experiment: Convergence Performance

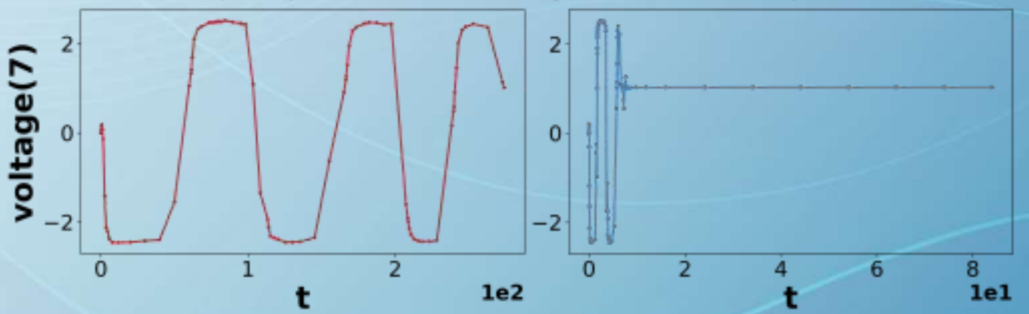
- Convergence tests of Soda-PTA and BoA-PTA in three PTA algorithms

PTA Methods	Circuits	NR_iters		
		navie	BoA-PTA	Soda-PTA
CEPTA	opampal	time-step too small	635	317
	D10		65	60
	loc	timeout	—	328
	ram2k		188	158
DPTA	gm17	timeout	N/A	304
	gm19		N/A	160
	REGULATOR		N/A	644
	Divider		N/A	511
RPTA	Schmitslow	timeout	N/A	4507
	bjtff		N/A	1458
	toronto		N/A	1484
	sram		N/A	2341

- Comparison of simulation curves before and after Soda-PTA optimization for two non-convergent circuits



(a) circuit "opampal" (Non convergence, time-step too small)

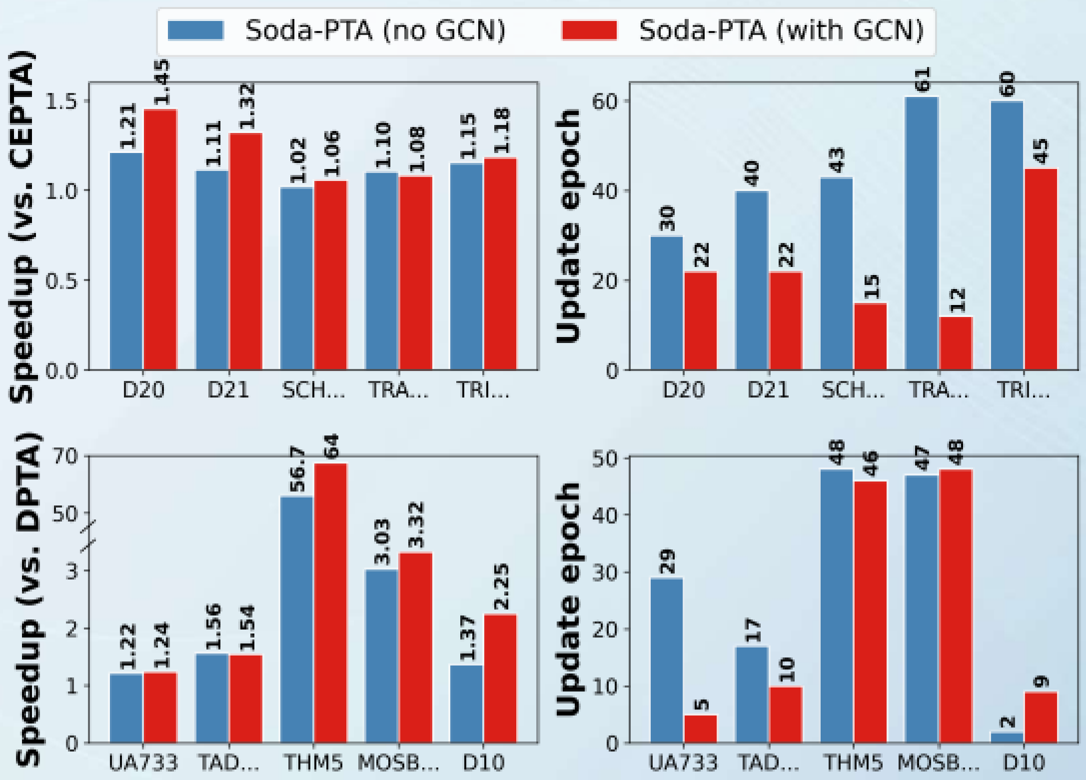


(b) circuit "Divider" (Non convergence, oscillation)

Soda-PTA offers consistently superior convergence capability compared to BoA-PTA.

Experiment: Optimization with GCN

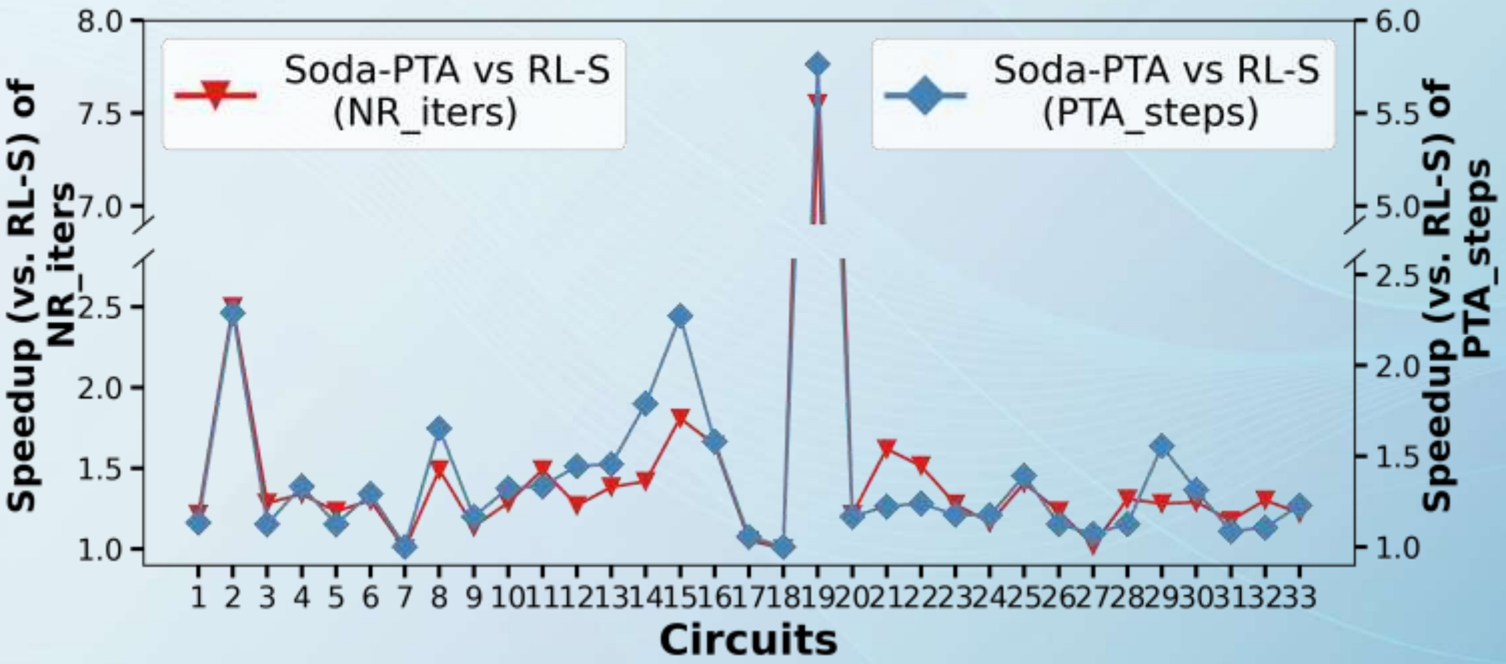
➢ Convergence tests of Soda-PTA and BoA-PTA in three PTA algorithms



- The left subfigure shows tests on CEPTA. Circuits 'D20' and 'D21' indicate that GCN enhances optimization results while reducing parameter update iterations. For the 'TRACKTorig' circuit, fewer iterations are needed without significantly changing the optimization results. The same conclusion is observed under DPTA.
- GCN provides a stable mapping from circuit descriptions to vector space, guiding parameter gradient updates and ensuring optimization quality.

Experiment: Optimization with RL-S

➢ Experimental comparison under the SOTA time step control strategy RL-S[1]



- RL-S is the SOTA time step control optimization strategy in PTA methods. The above figure shows that applying Soda-PTA to RL-S results in average improvements of 1.53x (max 7.55x) in NR_iters and 1.46x (max 5.76x) in PTA_steps.
- The synergistic use of both offers potential value for further optimizing PTA methods.

[1] Z. Jin, H. Pei, Y. Dong, X. Jin, X. Wu, W. W. Xing, and D. Niu. Accelerating nonlinear dc circuit simulation with reinforcement learning. In DAC, 2022.

Summary

- This paper proposes a parameter optimization framework that maps the PTA solving process to the Neural ODE training process, deriving effective gradient information for PTA hyperparameters to address the performance dependence on hyperparameters in PTA algorithms.

- The framework is equipped with a GCN model to capture circuit topology features, enhancing the quality of the parameter optimization process.

- Compared to the SOTA method BoA-PTA, Soda-PTA achieves an average improvement of 1.53x and a maximum of 1.90x under CEPTA, while also ensuring better convergence capability. Similarly, significant performance improvements are observed in other PTA algorithms, with an average of 14.77x under the DPTA algorithm.

https://github.com/SuperScientificSoftwareLaboratory/Soda-PTA

WELCOME TO COOPERATE WITH US!

Email: jinzhou@cup.edu.cn

IEEE/ACM 2024 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN ICCAD 43rd Edition