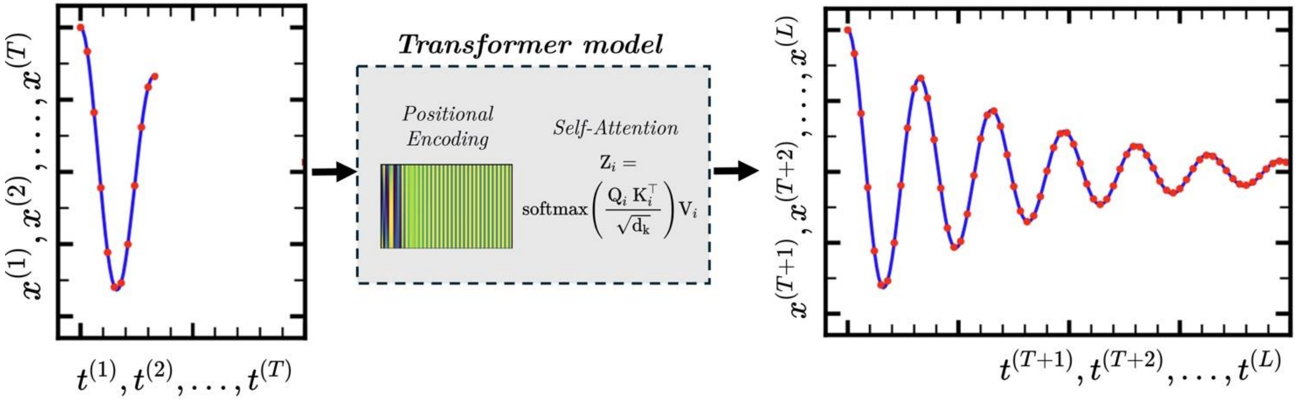
Accelerating quantum dynamics simulations with machine learning



Alexei Kananenka

(akanane@udel.edu)

University of Delaware

MolSSI workshop “Machine-Learning in Quantum and Nonadiabatic Dynamics”

Buffalo, August 16, 2024

Acknowledgments

Kananenka group

- *Luis Herrera Rodriguez*
- Kennet Espinosa
- Kyle Billings
- Aarti
- Nikhil Maroli
- Aritri Biswas
- Natalia Munoz
- James Fitzerald

Collaborators

- Pavlo Dral (Xiamen)
- Arif Ullah (Anhui U.)

U.S. DEPARTMENT OF ENERGY

NVIDIA

ORAU

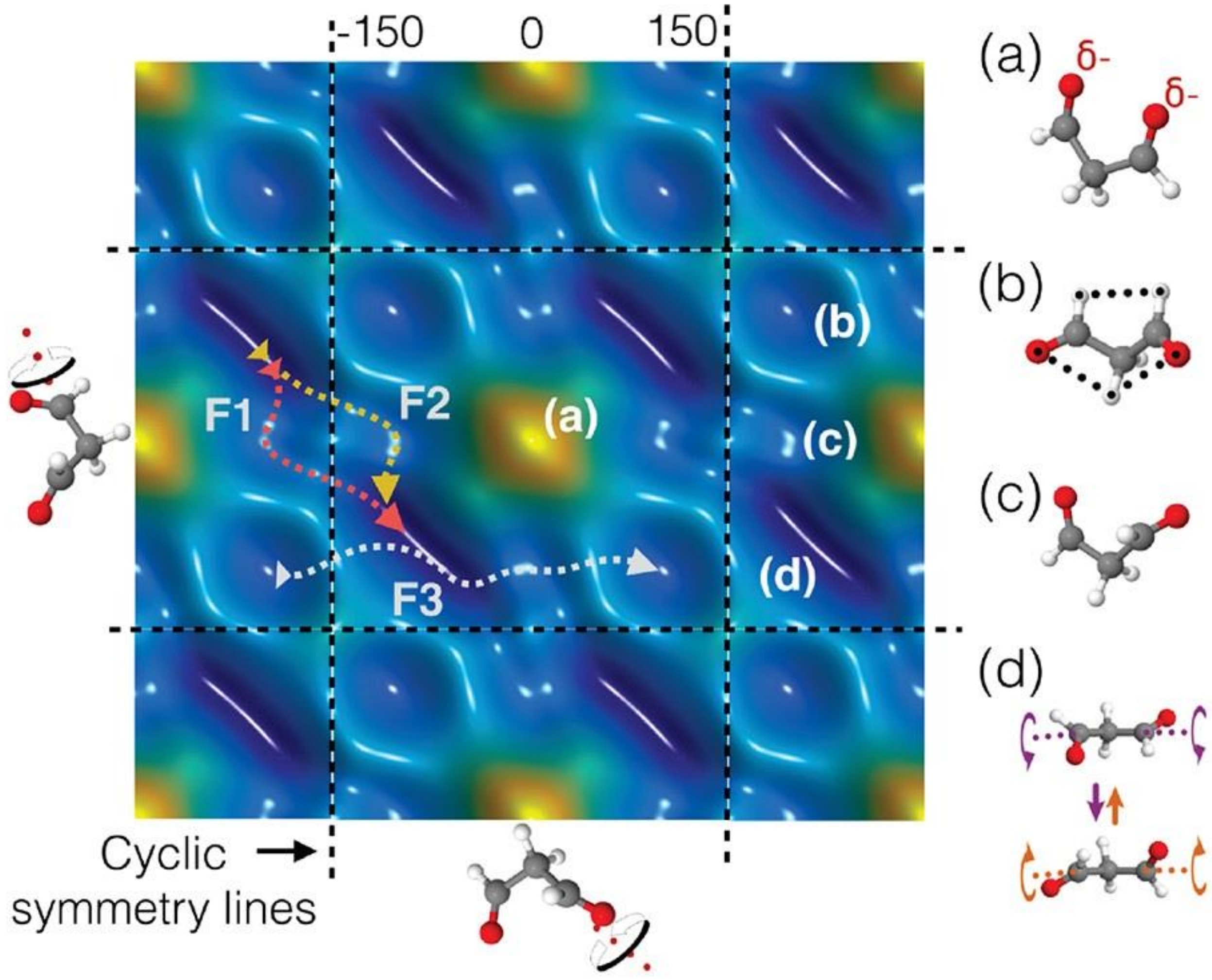
UNIVERSITY OF DELAWARE

Machine learning for molecular dynamics

Most common:
Use ML to get PES (U) faster than solving ESP
Use in classical or AIMD

$$-\frac{dU}{dx} = F = ma$$

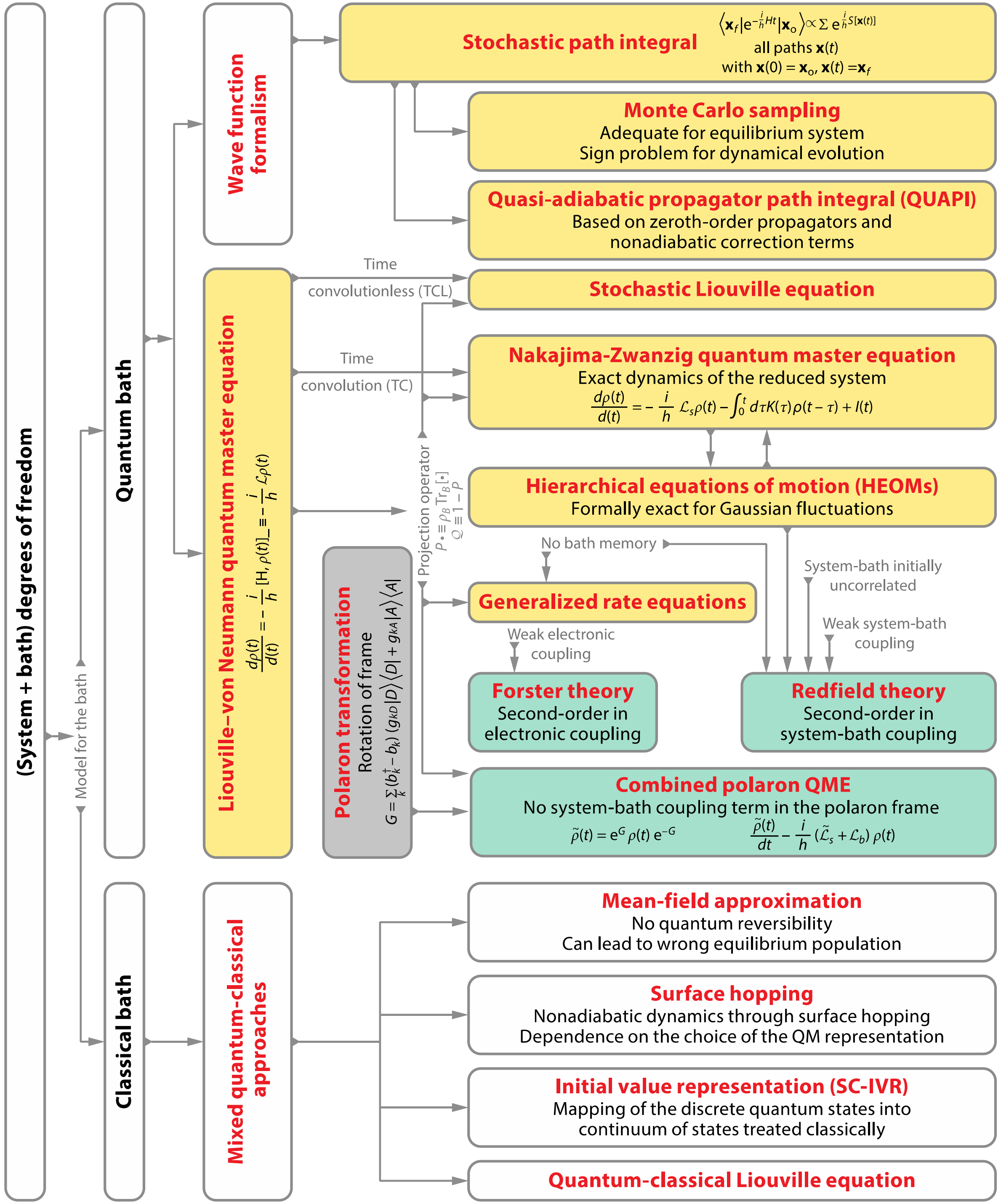
PES



Our work:
Assume U is known
solve QD problem

$$i\hbar\frac{d}{dt}\Psi = (T+U)\Psi$$

Unke et al., *Chem. Rev. 121, 10142-10186 (2021)*



There is no shortage of methods for quantum dynamics simulations

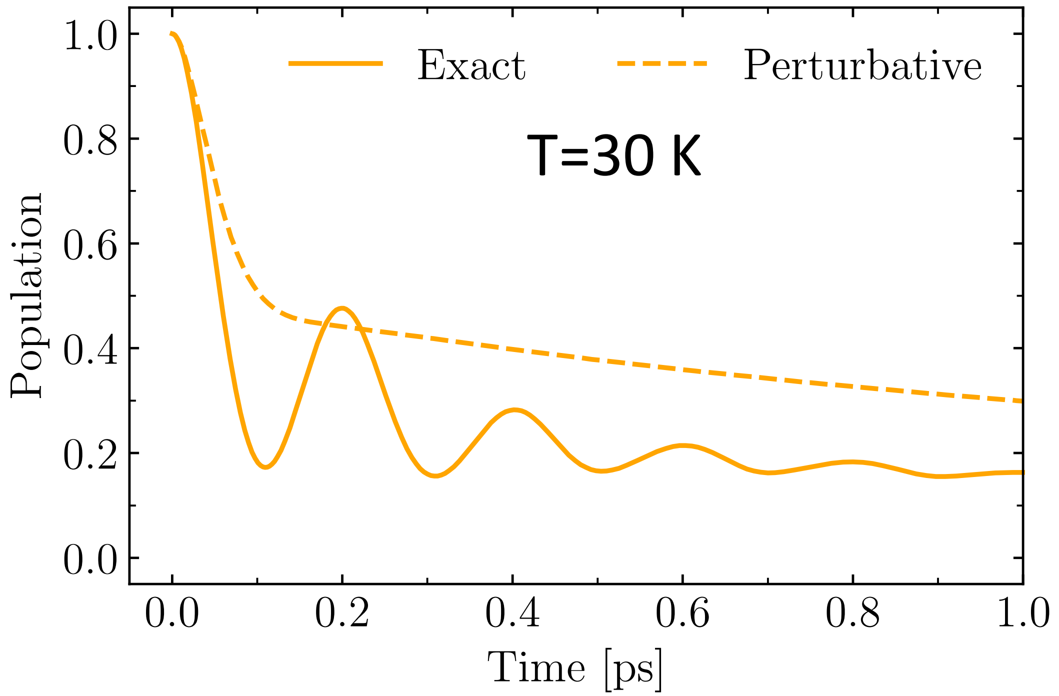
Numerically exact methods are computationally expensive

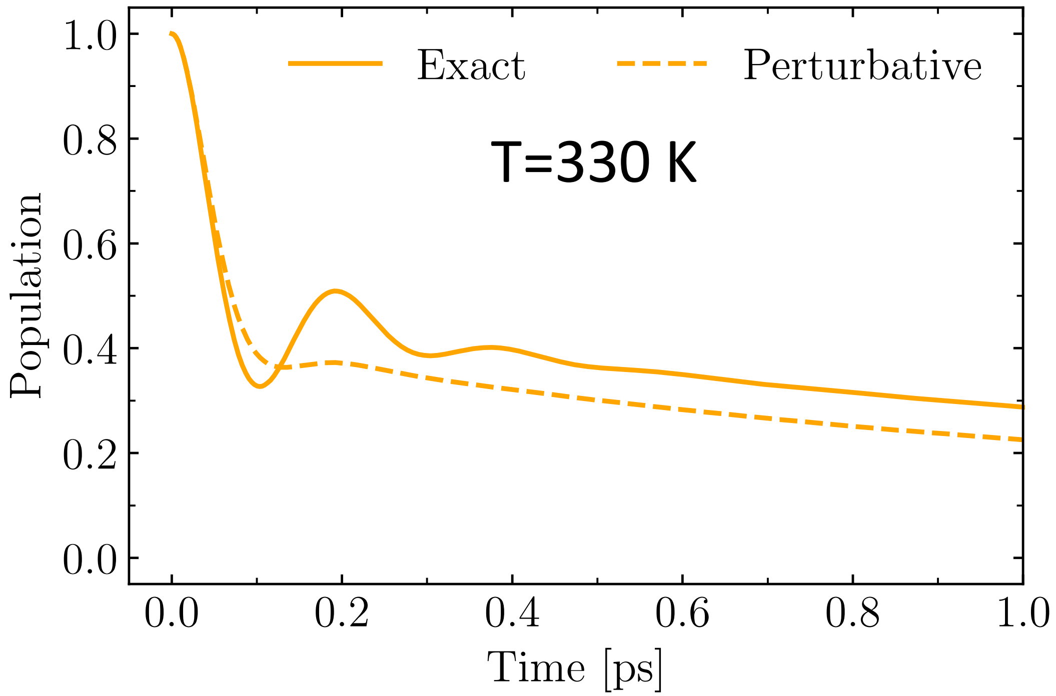
Perturbative methods limited to certain regimes

Quantum-classical often inaccurate *long-time*

Approximate methods are reasonably accurate for short-time dynamics

Fenna-Matthews-Olson population dynamics





- Exact: 5.5 h and 23 Gb of RAM
- Approx.: 30 sec and <0.1 Mb of RAM

- Exact: 2 min and 5 Mb of RAM
- Approx.: 30 sec and <0.1 Mb of RAM

Can we use accurate short-time information to “extrapolate” to longer times?

Physics-based methods: GQME

Time evolution of the reduced density operator

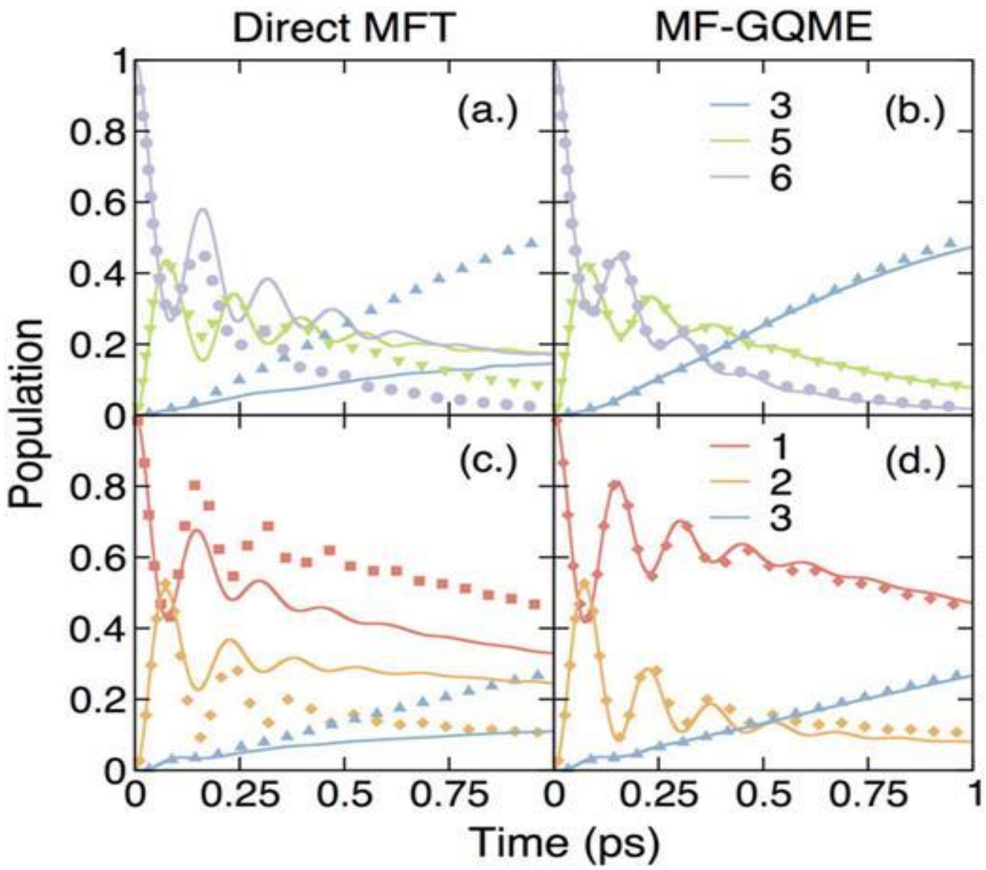
$$\frac{d\rho_s(t)}{dt} = \frac{i}{\hbar} L_s \rho_s(t) - \int_0^t K(\tau)\rho_s(t-\tau)d\tau$$

Memory **kernel K(t):**

- Environmental effects
- $N^2 \cdot N^2$ dimension
- Very difficult (impossible) to obtain in the exact form for realistic (anharmonic) systems
- If known long-time dynamics can be obtained by solving the GQME

The spin-boson model

$$H = \epsilon\sigma_z + \Delta\sigma_x + \sigma_z \sum_\alpha g_\alpha (b_\alpha^\dagger + b_\alpha) + \sum_\alpha \omega_\alpha b_\alpha^\dagger b_\alpha$$



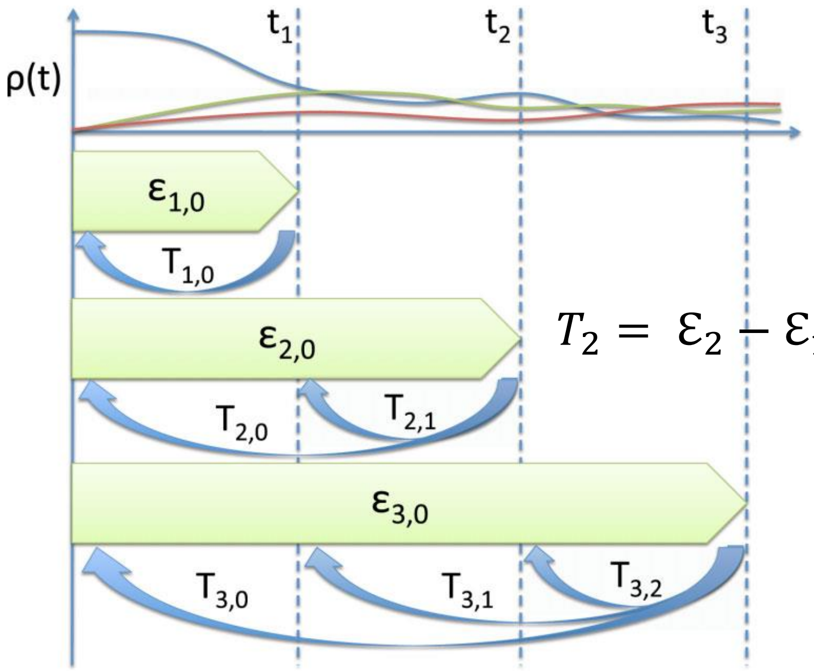
J. Chem. Phys. 150, 244109 (2019)

Physics-based methods: transfer tensor method

Dynamical map: $\rho(t_k) = \mathcal{E}_k \rho(0)$

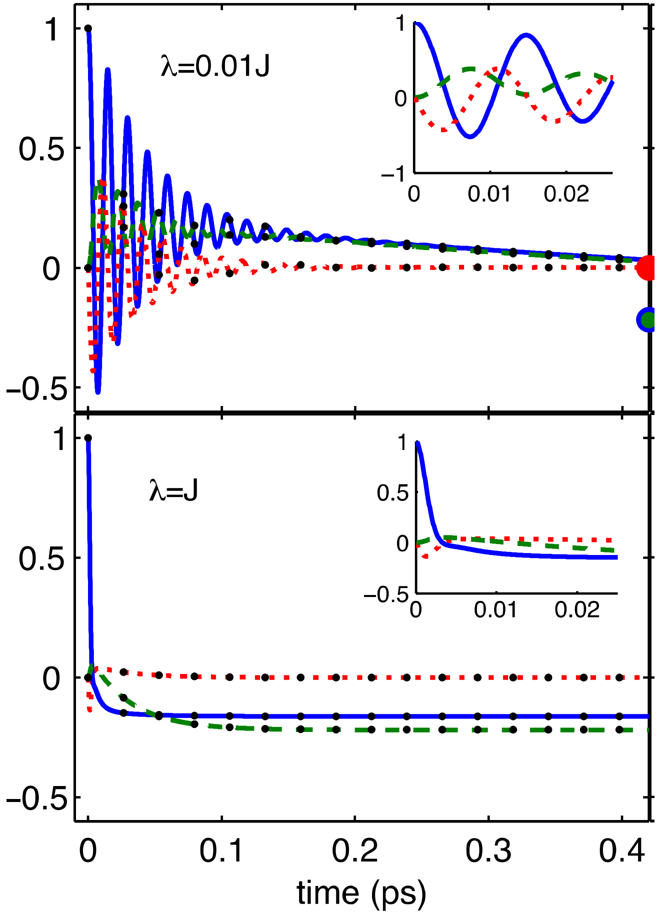
Transfer tensor: $T_k = \mathcal{E}_k - \sum_{m=1}^{k-1} T_{k-m}\mathcal{E}_m$

Propagation: $\rho(t_m) = \boldsymbol{T} \otimes [\rho(t_{m-1}) \ldots \rho(t_{m-k})]$



$T_2 = \mathcal{E}_2 - \mathcal{E}_1\mathcal{E}_1$

Spin-boson model

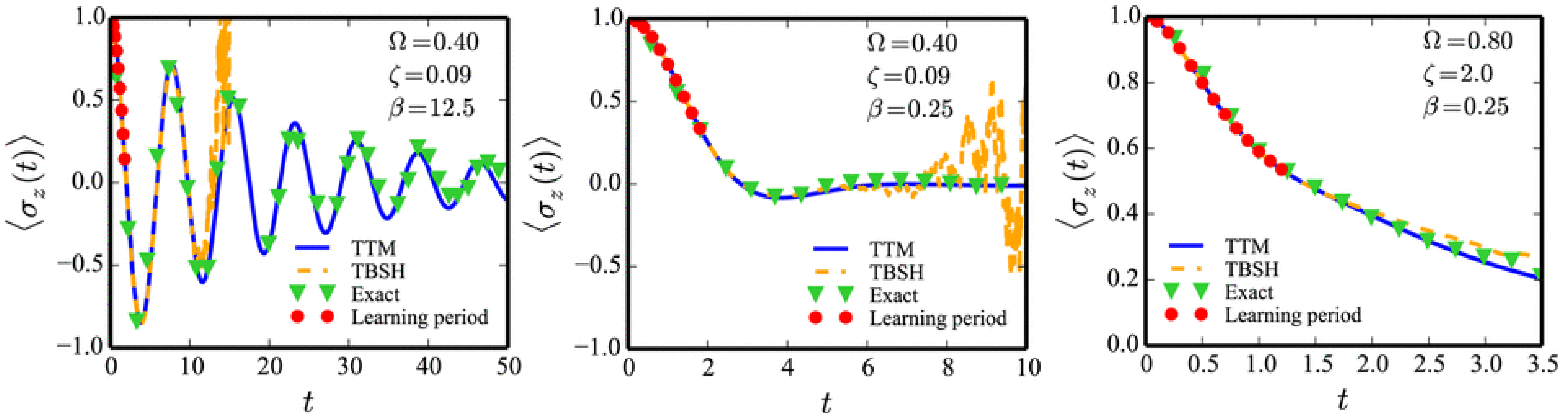


Requires N^4 calculations to produce dynamical maps

Phys. Rev. Lett. 112, 110401 (2014)

TTM with approximate quantum-classical input

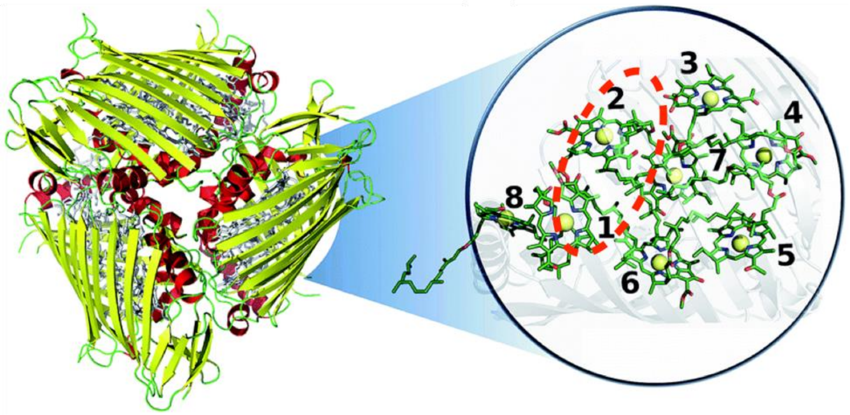
Population difference for two-level system coupled to dissipative environment



TTM works well even when input is not from the exact method

__AAK__, C.-Y. Hsieh, J. Cao, E. Geva, J. Phys. Chem. Lett. 7, 4809 (2016)

Where is memory coming from?



Reduction to relevant DOFs

Generalized Quantum Master Equation

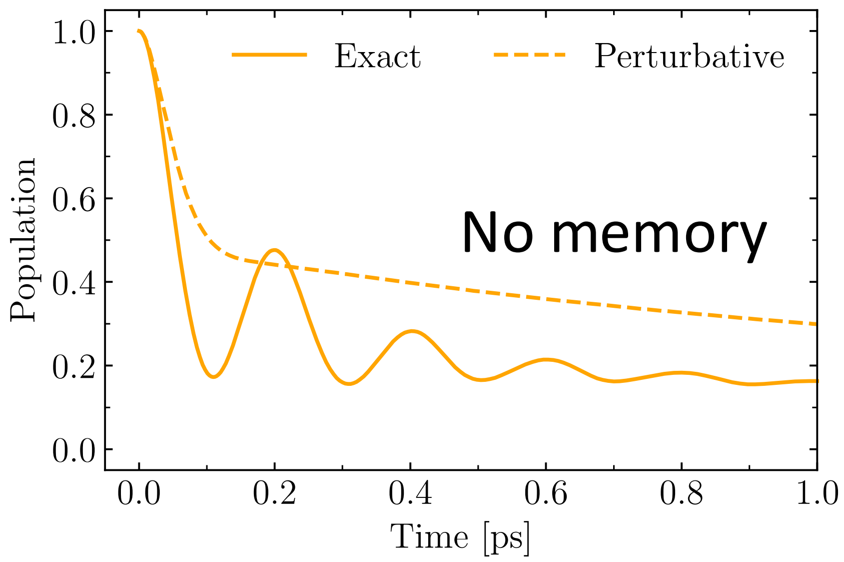
$$\frac{d\rho_s(t)}{dt} = \frac{i}{\hbar} L_s \rho_s(t) - \int_0^t K(\tau)\rho_s(t)(t-\tau)d\tau$$

non-Markovian dynamics is a result of reduction used to focus on “relevant” subsystem

Schrödinger equation

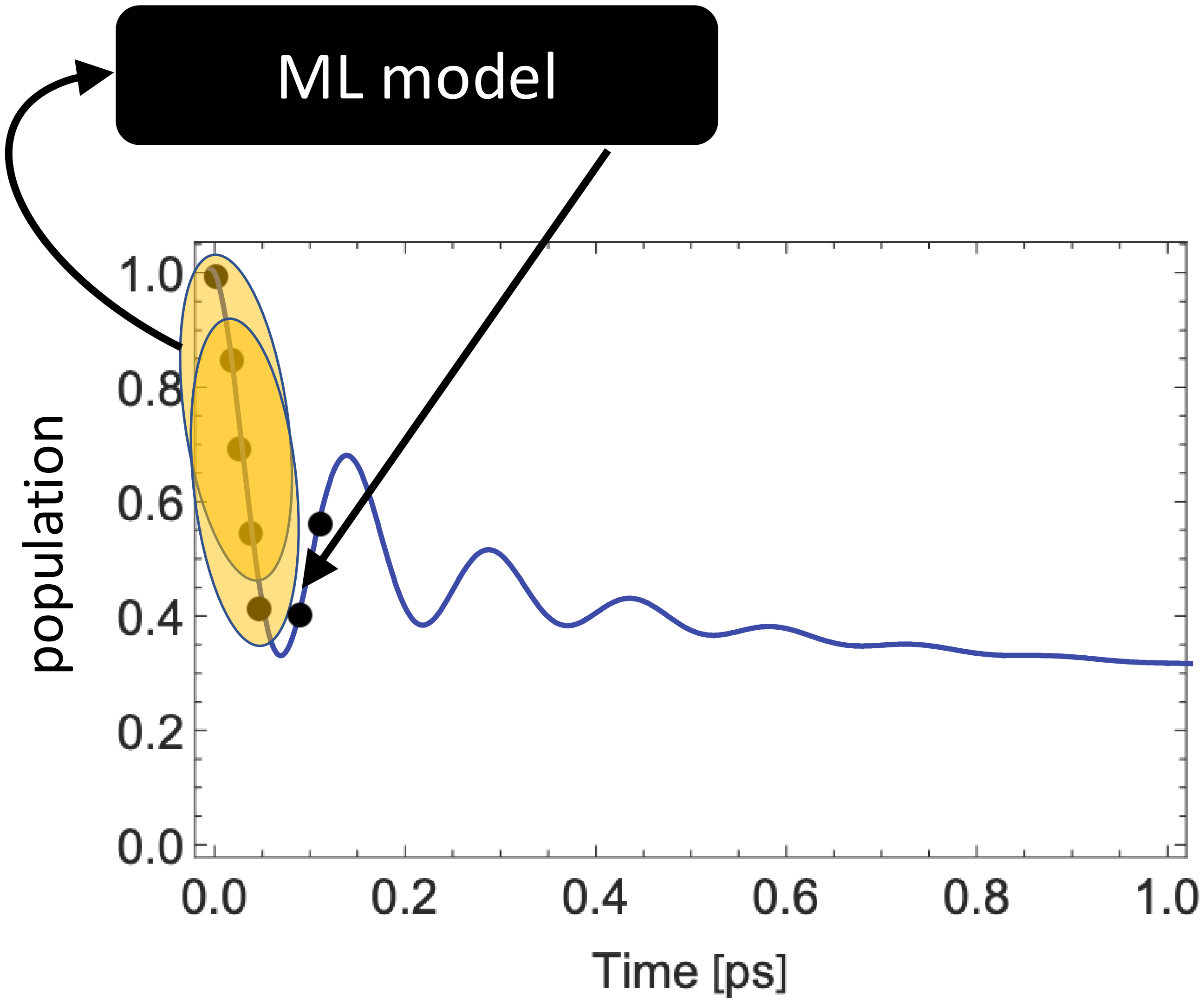
$$i\hbar\frac{d}{dt}\Psi = H_{total}\Psi$$

Markovian: future time-evolution is fully determined by the present state of the system



Memory is a property of environment which determines the physical behavior of the subsystem

Our approach: time-series forecasting with data-driven models

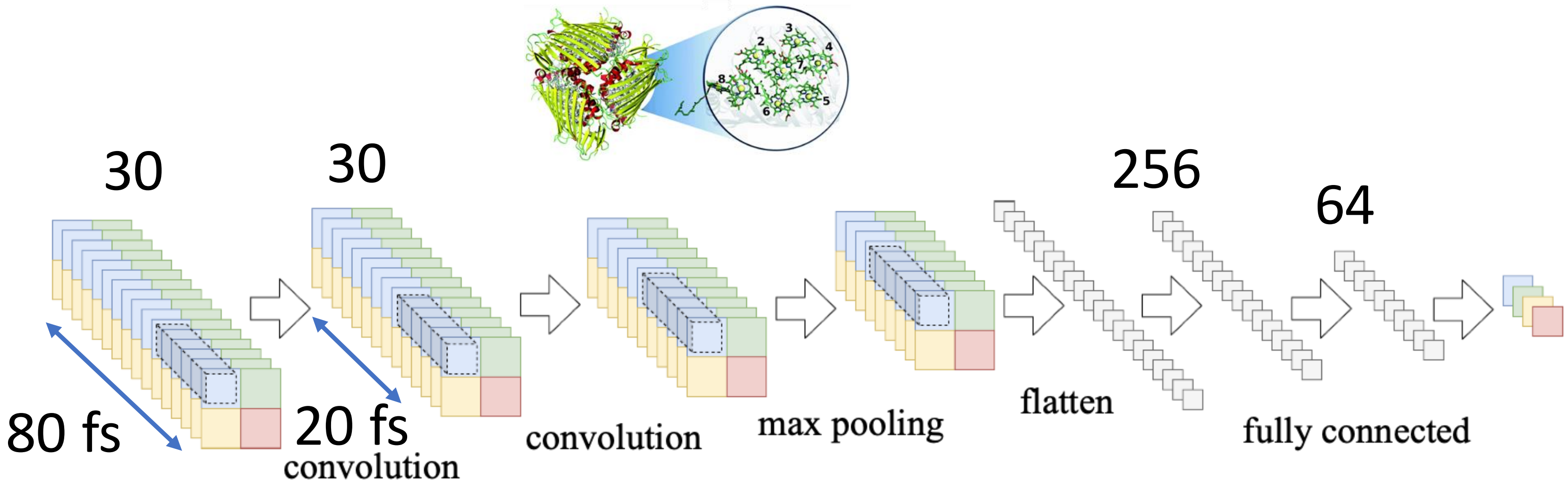


What we want:

- Must be single-step accurate
- Must be faster than direct "exact" QD calculation (e.g., HEOM)
- "Short times" should be short
- Minimum work to get the input

*L. Rodriguez and **AAK**, J. Phys. Chem. Lett. 12, 2476 (2021)*

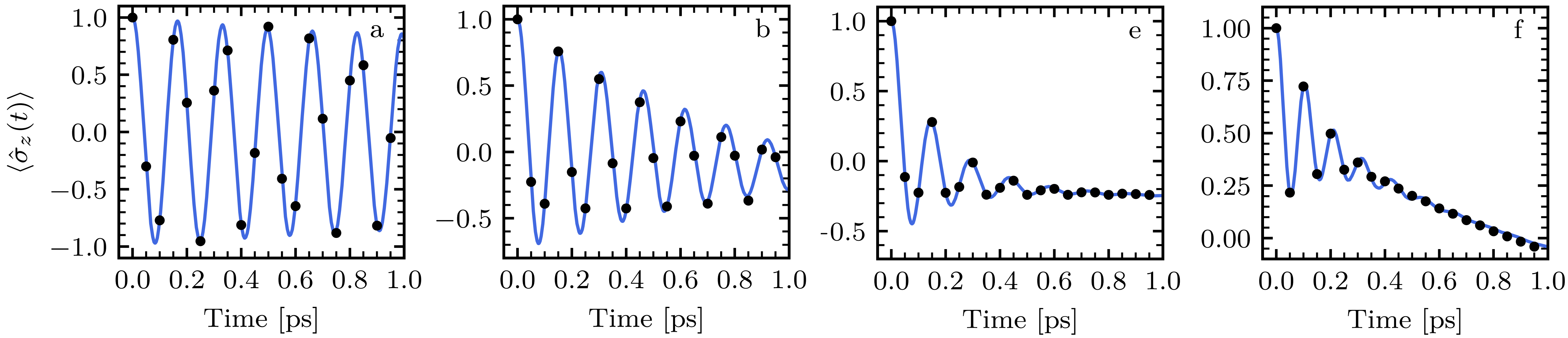
FMO dimer dynamics with convolutional neural network



2,500,000 trainable parameters

Trained on ~5,000,000 0.2 ps trajectories

1->2 population difference in molecular dimer (from FMO): input 2x2 density matrix



*L. Rodriguez and **AAK**, J. Phys. Chem. Lett. 12, 2476 (2021)*

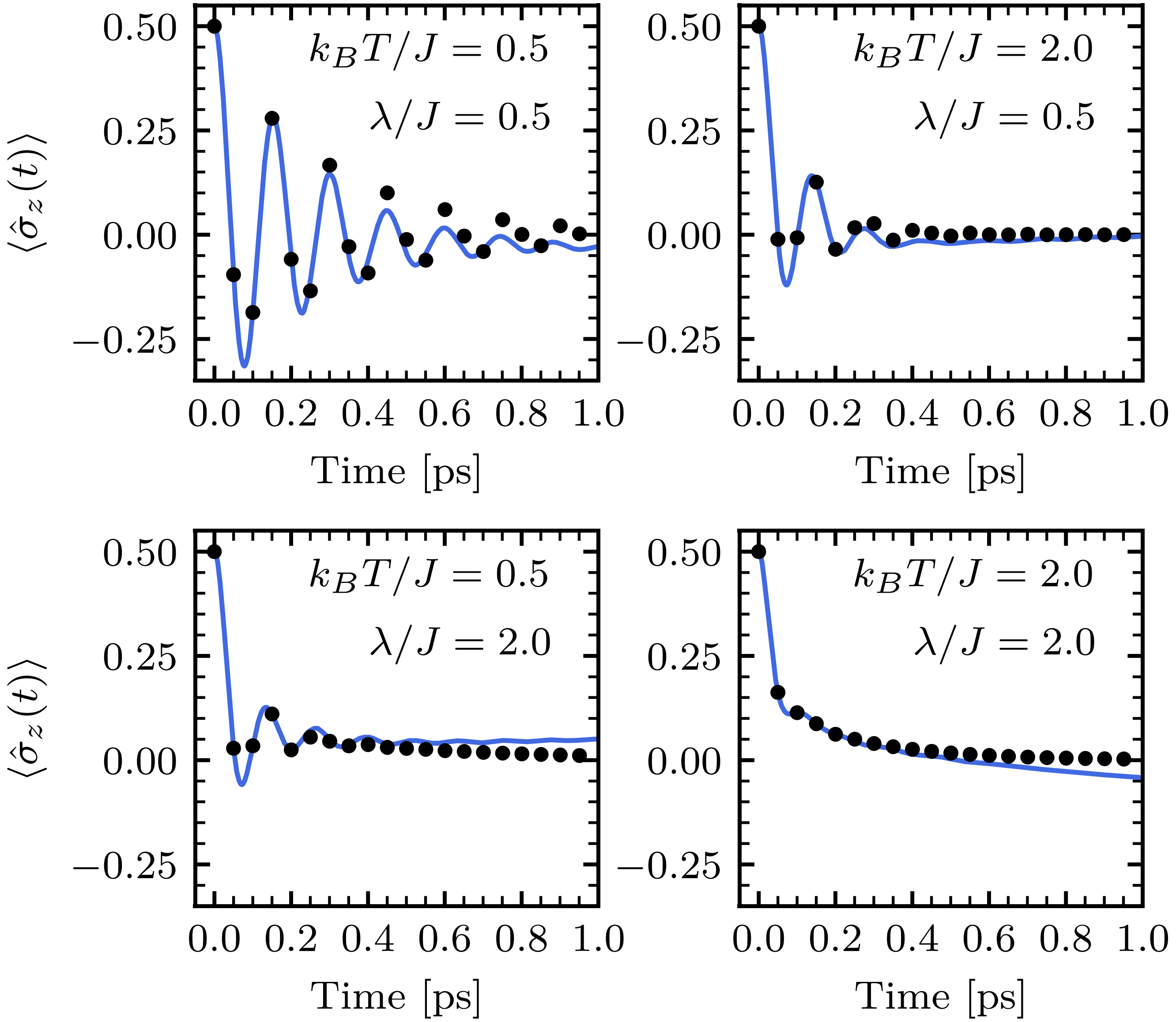
Transferability

Input data consisted of dynamics data starting from the excited state

$$\rho_s = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

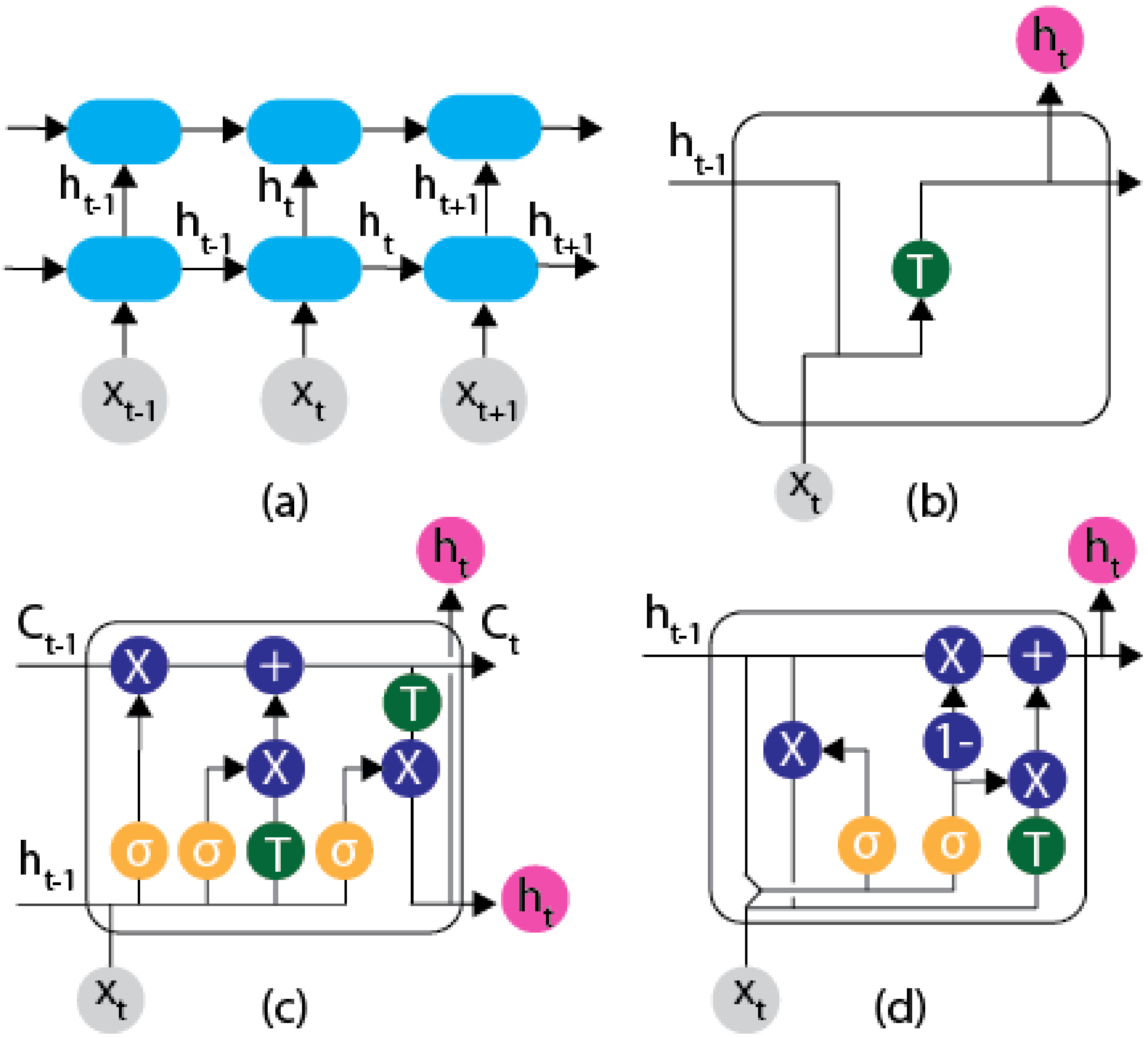
The trained CNN model also works for the initial mixed state

$$\rho_s = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$



*L. Rodriguez and **AAK**, J. Phys. Chem. Lett. 12, 2476 (2021)*

More machine learning models



Benchmarking 22 most popular ML models:

- Feed-forward NNs
- Convolutional NNs
- Recurrent NNs (a): simple RNN (b), LSTM (b), GRU (c)
- Bidirectional RNNs
- Convolutional Recurrent NNs
- Kernel ridge regression models with kernels:
  - Gaussian
  - Matern (n=1-4)
  - Exponential
  - Periodic-decaying

Fixed number of trainable parameters (NN): 500,000-530,000

*L. Rodriguez,…, **AAK**, Mach. Learn.: Sci. Technol. 3 045016 (2022)*

Kernel ridge regression

Prediction for input x' $$f(x') = \sum_{i=1}^{N_{train}} \alpha_i K(x', x_i)$$

Regression coefficient

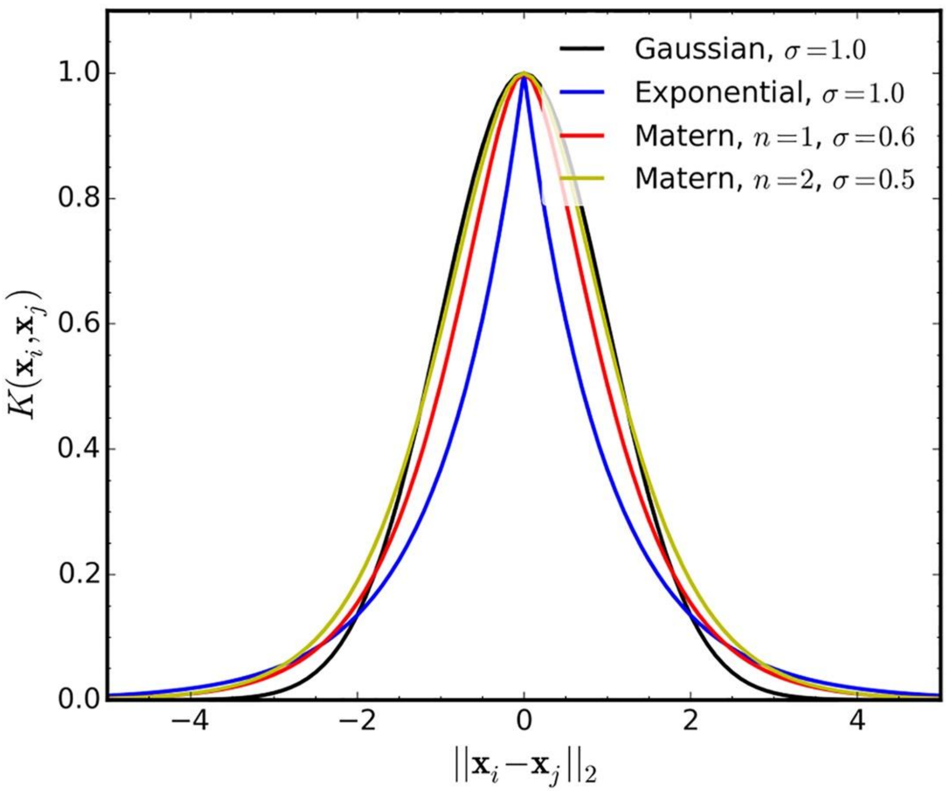
Kernel function (Kernel)

Regression coefficients are determined by "training"

$$min_{\alpha} \sum_{i=1}^{N_{train}} (f(x_i) - y_i)^2 + \lambda \alpha^T K \alpha$$

- Training is expensive for large data sets, N^3
- Fixed size input (unlike RNNs)
- Few kernels exist for time-series data

Kernel functions



Spin-boson data set

$H = \epsilon\sigma_z + \Delta\sigma_x + \sigma_z \sum_\alpha g_\alpha (b_\alpha^\dagger + b_\alpha) + \sum_\alpha \omega_\alpha b_\alpha^\dagger b_\alpha$

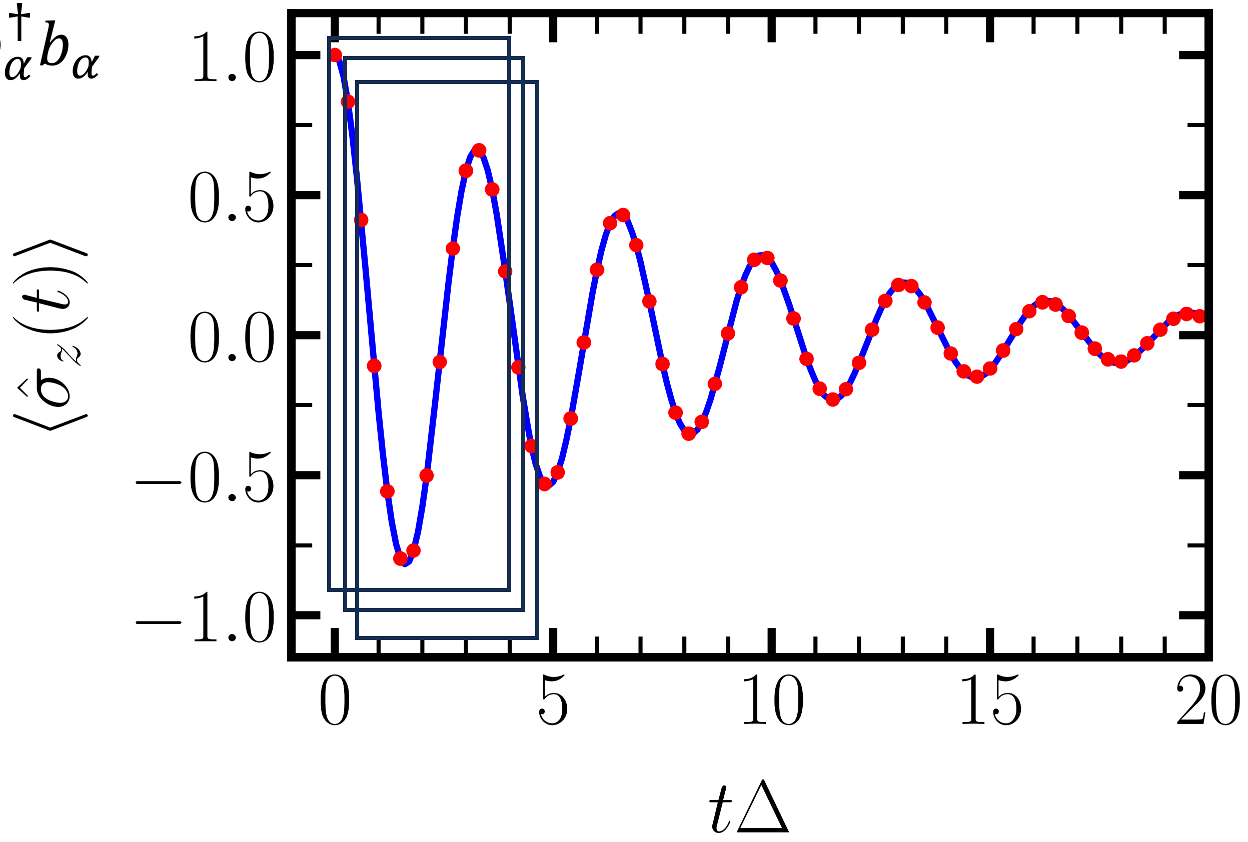
Ohmic spectral density:

$$J(\omega) = 2\lambda \frac{\omega\omega_c}{\omega^2 - \omega_c{}^2}$$

Use HEOM to generate RDMs for:

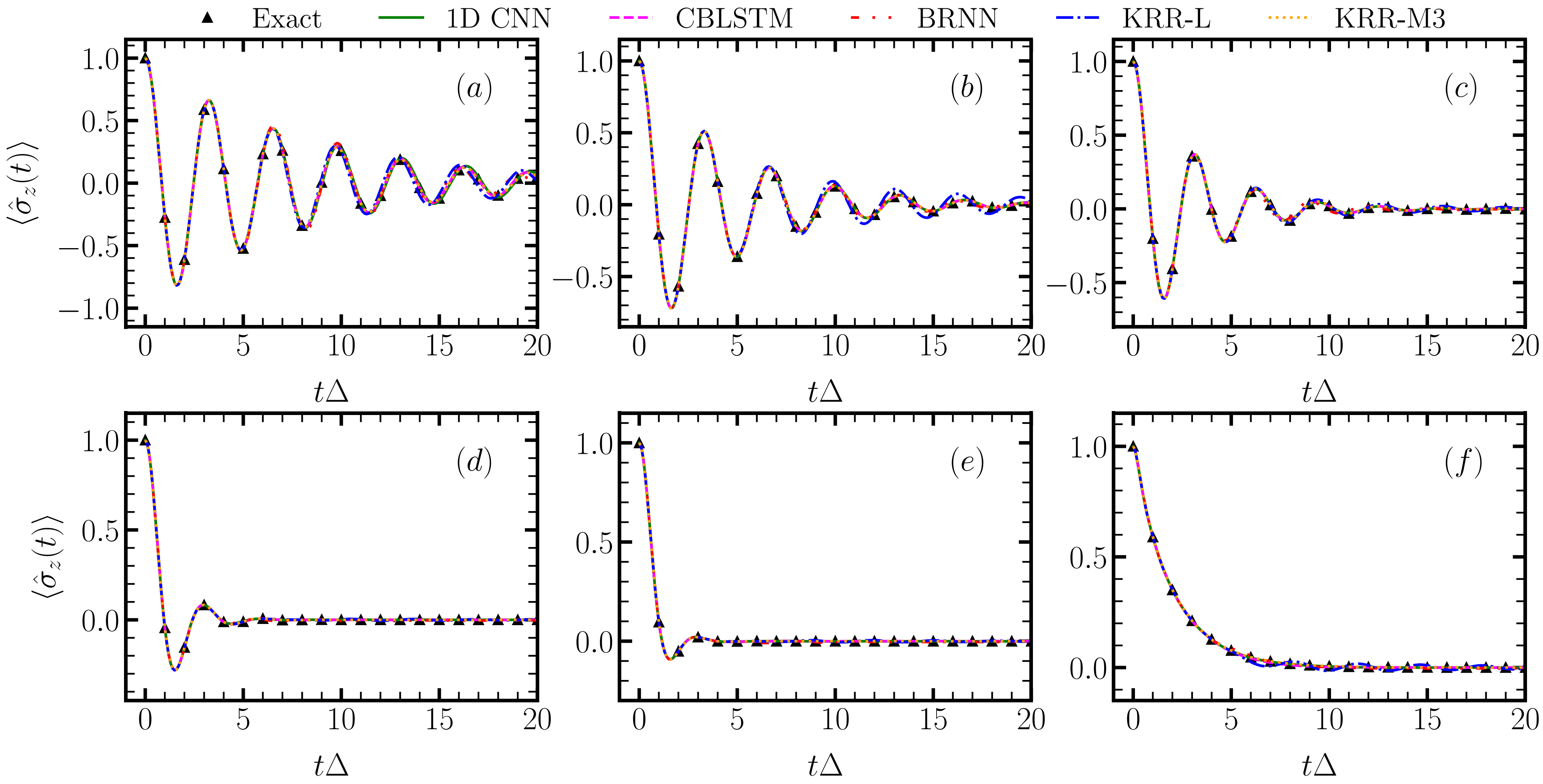
$\frac{\epsilon}{\Delta} = \{0,1\}$ $\frac{\omega_c}{\Delta} = \{1,2,3,\ldots,10\}$

$\beta\Delta = \{0.1,0.25,0.5,0.75,1.0\}$ $\frac{\lambda}{\Delta} = \{0.1,0.2,0.3,\ldots,1.0\}$



*A. Ullah, L. Rodriguez, P. O. Dral, and **AAK**, Front. Physics 11, 1223973 (2023)*

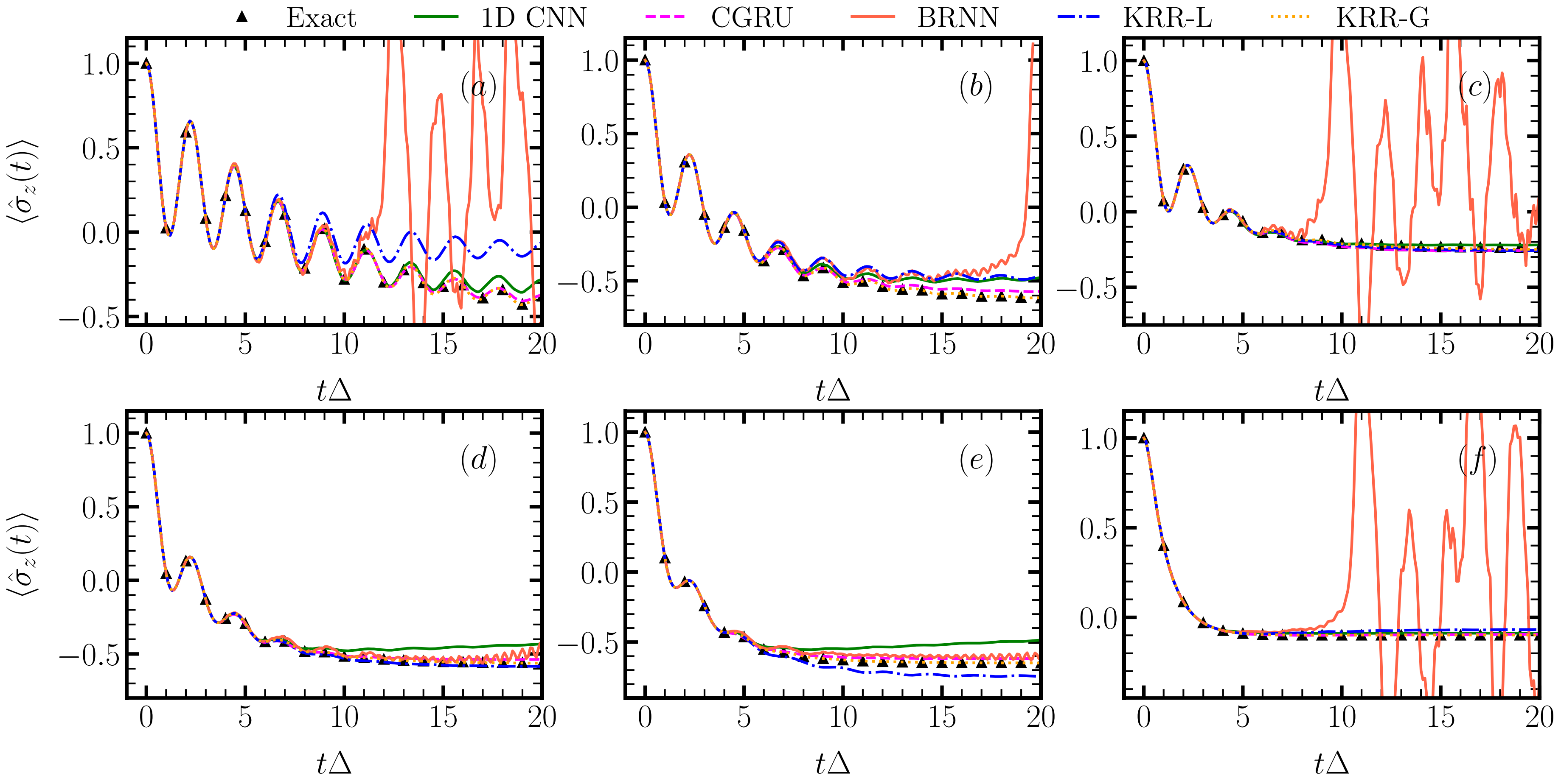
Symmetric spin-boson system



Lowest single-time step prediction error $2 \cdot 10^{-4}$ for KRR with Matern-4 kernel

*L. Rodriguez,…, **AAK**, Mach. Learn.: Sci. Technol. 3 045016 (2022)*

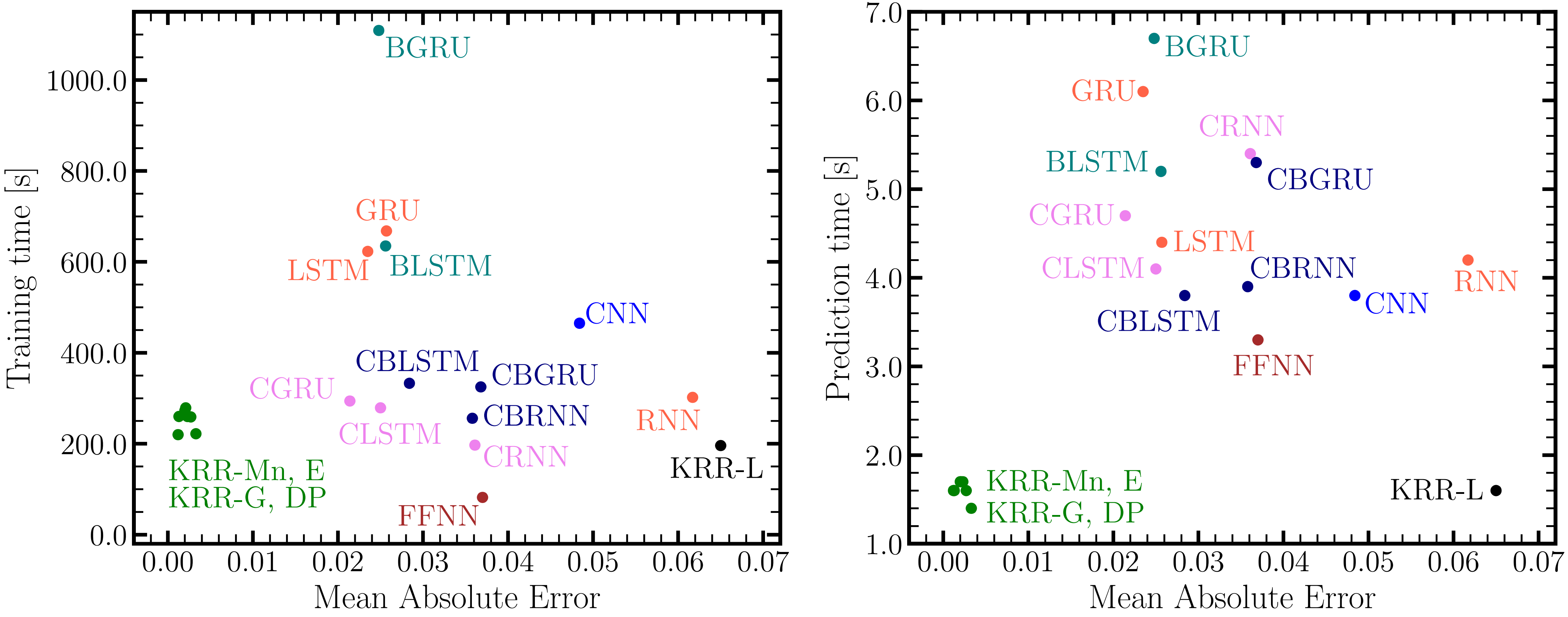
Asymmetric spin-boson system



Lowest single-time step prediction error $1.2 \cdot 10^{-3}$ for KRR with the Gaussian kernel

*L. Rodriguez,…, **AAK**, Mach. Learn.: Sci. Technol. 3 045016 (2022)*

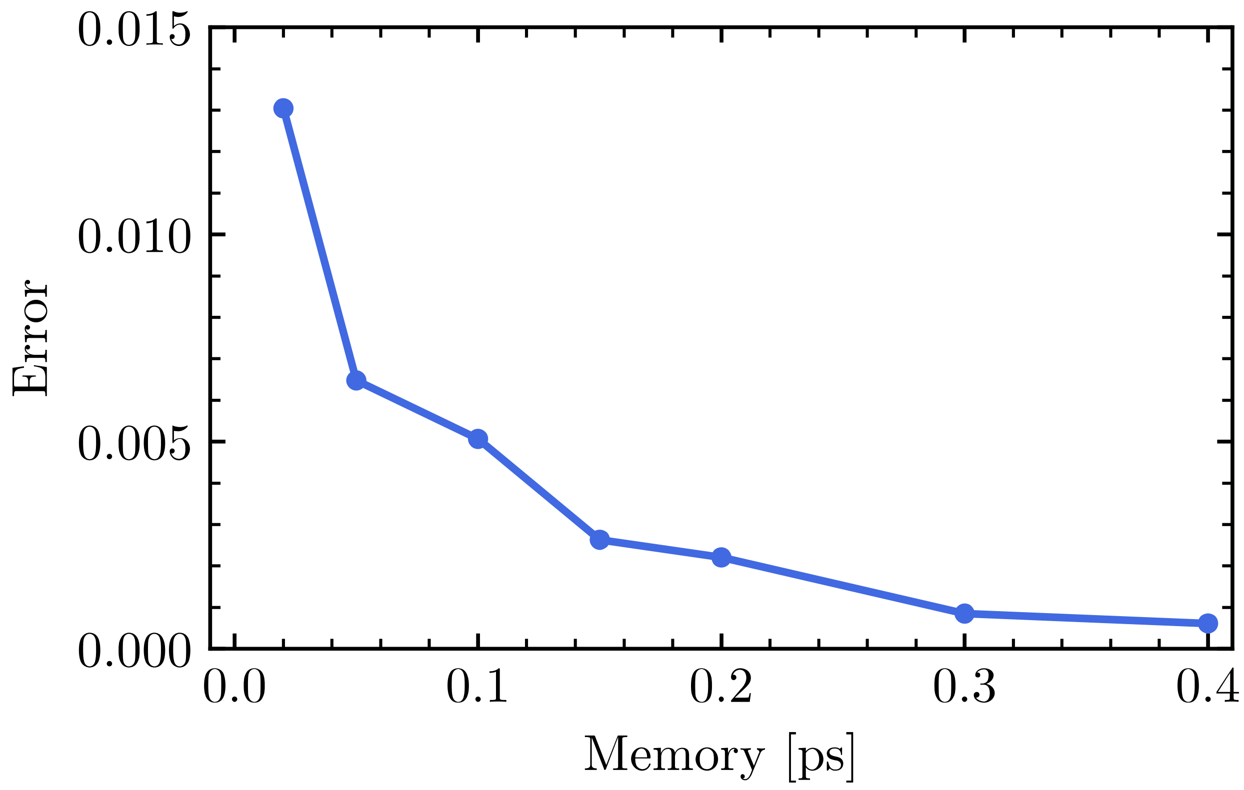
Accuracy vs running time



*L. Rodriguez,…, **AAK**, Mach. Learn.: Sci. Technol. 3 045016 (2022)*

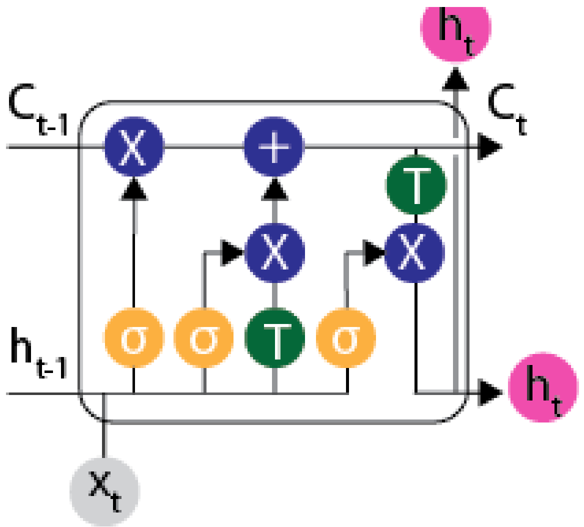
Choosing the memory: accuracy vs cost tradeoff

Convolutional neural network model



How to choose memory?

- Too short memory leads to sizable errors
- Too long memory requires more costly input generation
- Future work: Extract from RNNs



*L. Rodriguez and **AAK**, J. Phys. Chem. Lett. 12, 2476 (2021)*

Transformers

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

A. Viswani et al., NIPS 2017

Driving paradigm shift in AI

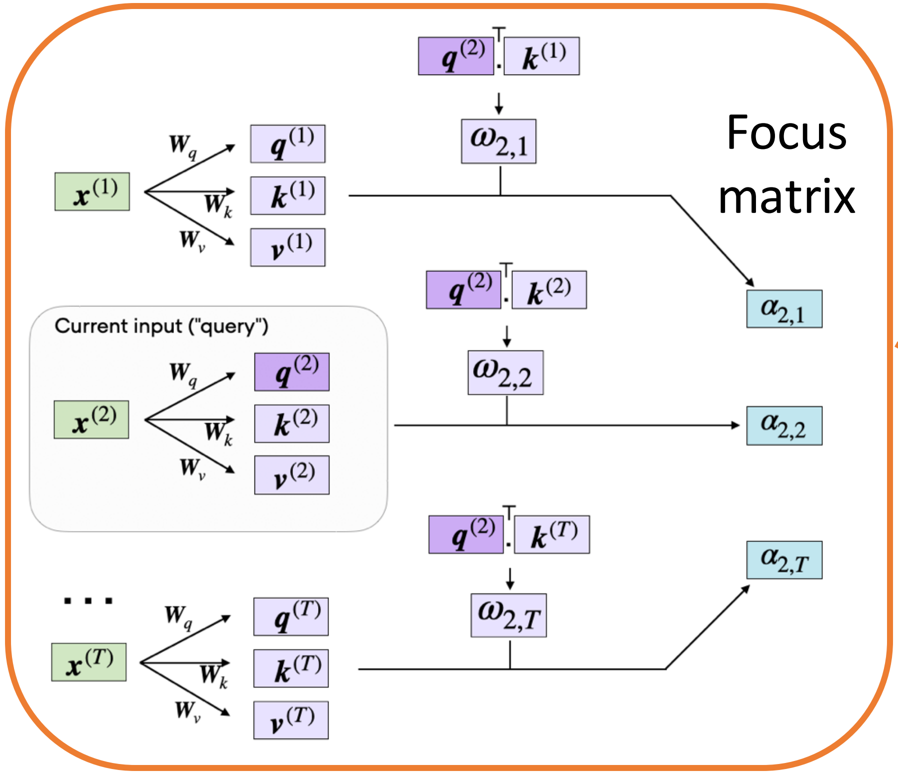
- RNNs are sequential (slow), short-term memory
- Introduced to improve/accelerate processing of long sequences (can do infinite in principle)
- 70% ArXiv papers on AI last 2 years
- conversational chat boxes, search engines
- Transformers use attention mechanism for context (parallel processing)

"General-purpose computer that is also trainable and efficient to run on our computer hardware…"
A. Karpathy

(https://youtu.be/9uw3F6rndnA?si=3lctTgxKDzjFpKSV)

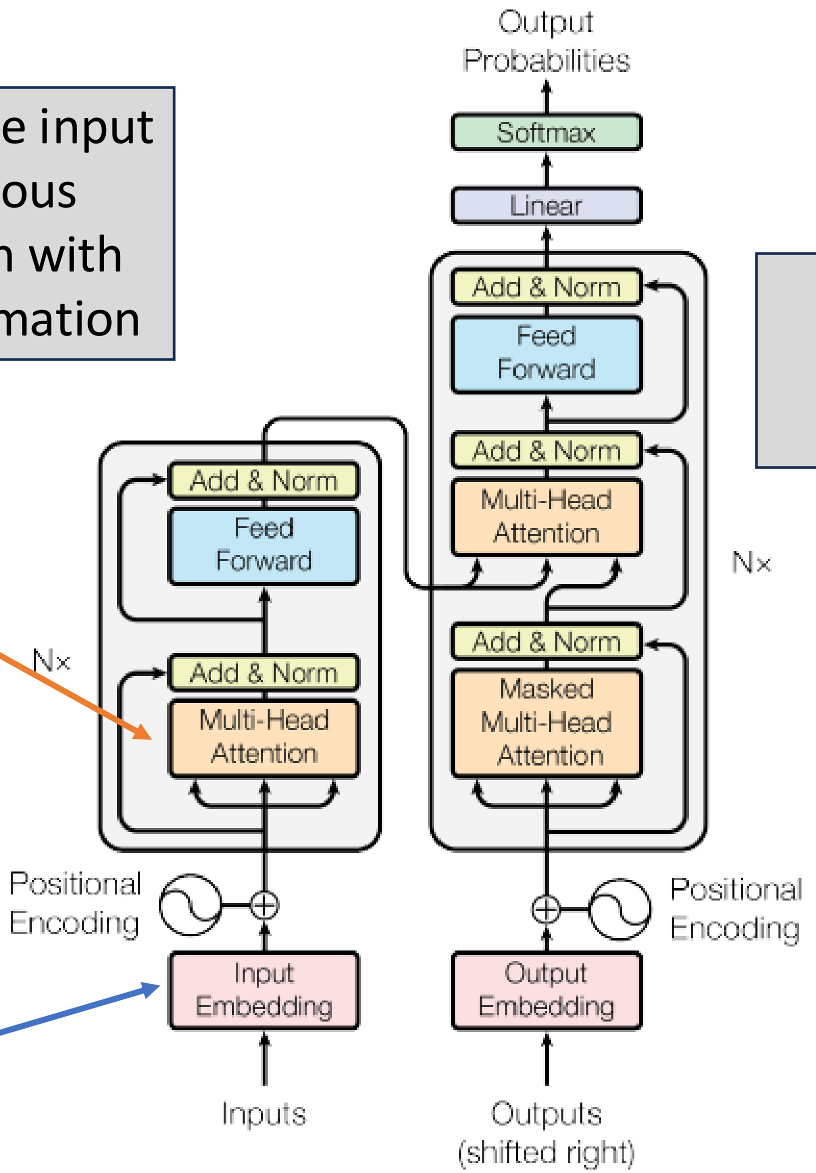
How transformers work: attention mechanism

Query, key, value



Encoder: Encode input into continuous representation with attention information

Decoder: generate sequencies

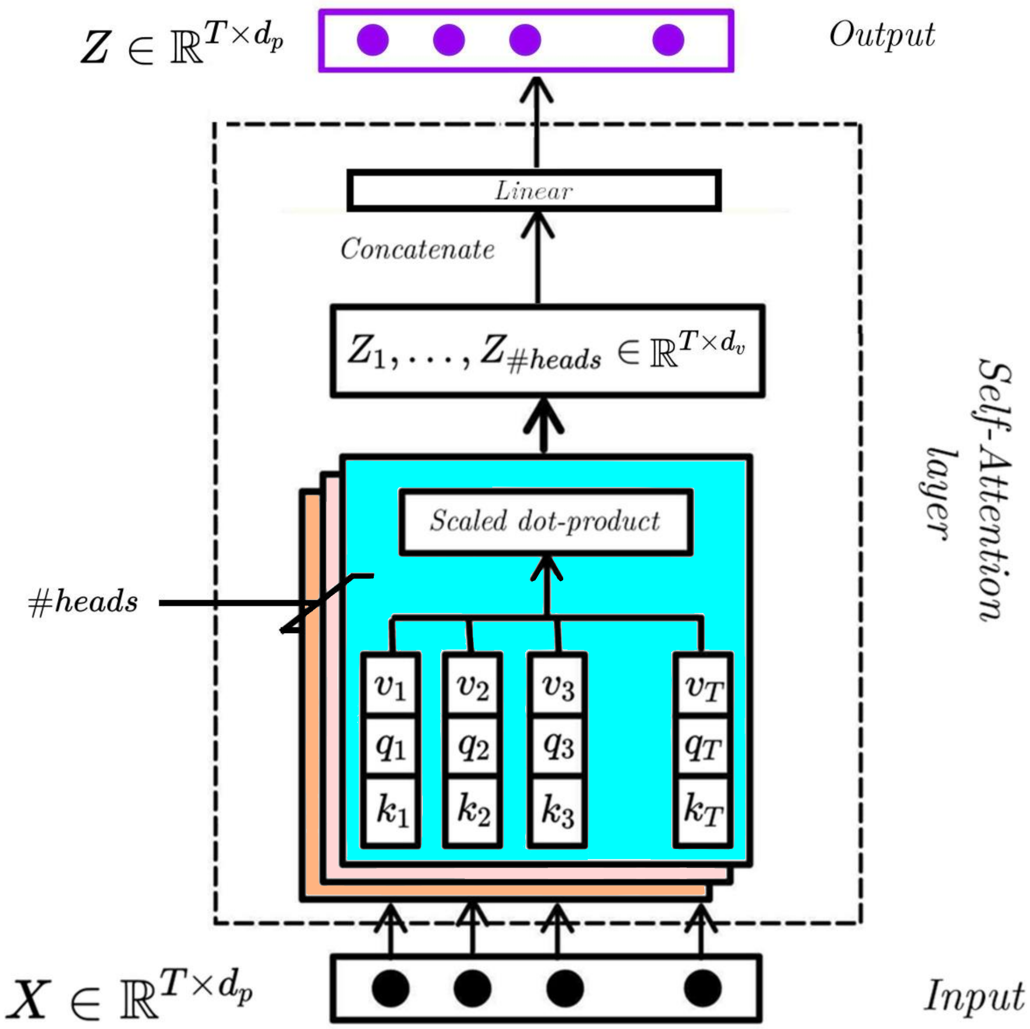


add positional information (PE)

map input (words) into a vector

A. Viswani et al., NIPS 2017

Self-attention layer



Concatenates output of each head

$$Z_1, Z_2, \ldots, Z_{Nheads}$$

generate self-attention matrix as weighted sum over input values

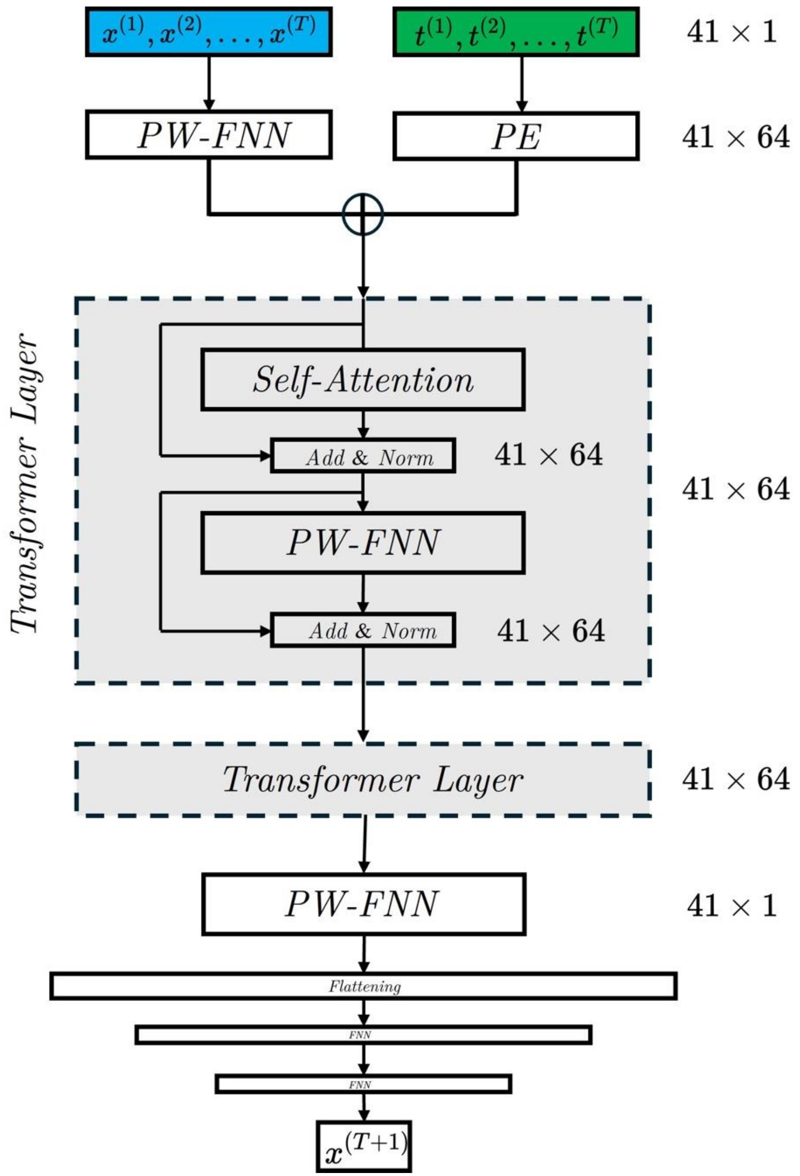
$$Z_i = softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i$$

Each self-attention head generates queries, keys, values

$$Q_i = XW_i^q, K_i = XW_i^k, V_i = XW_i^v$$

*L. Rodriguez and **AAK**, under review*

Transformer neural network for quantum dynamics



L. Rodriguez and ***AAK****, under review*

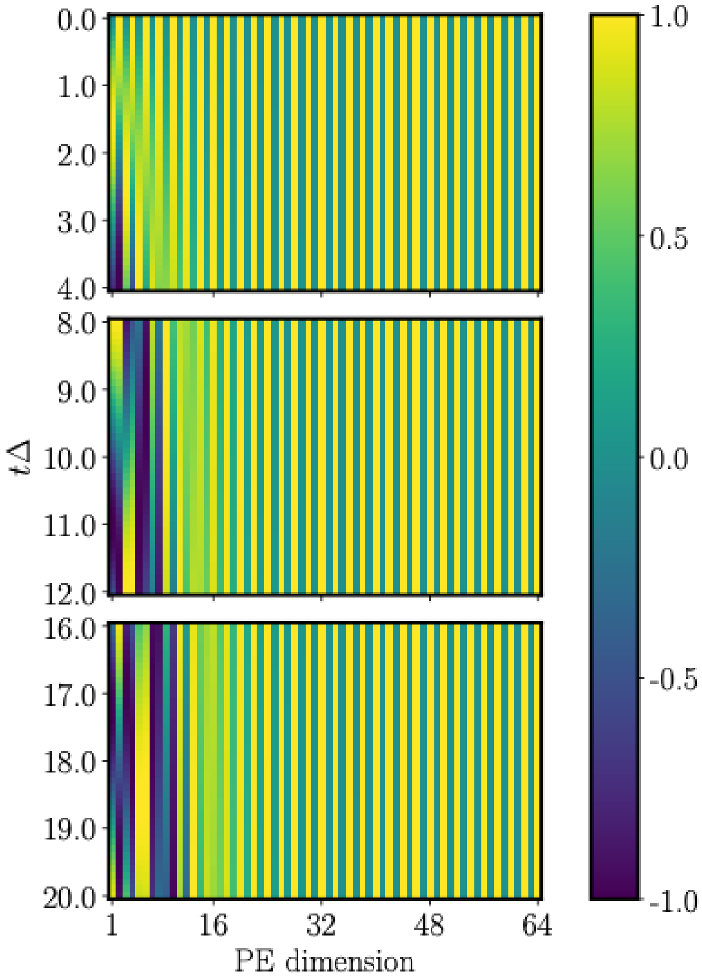
Positional encoding

- No explicit time information in the self-attention layer
- PE creates representation of time

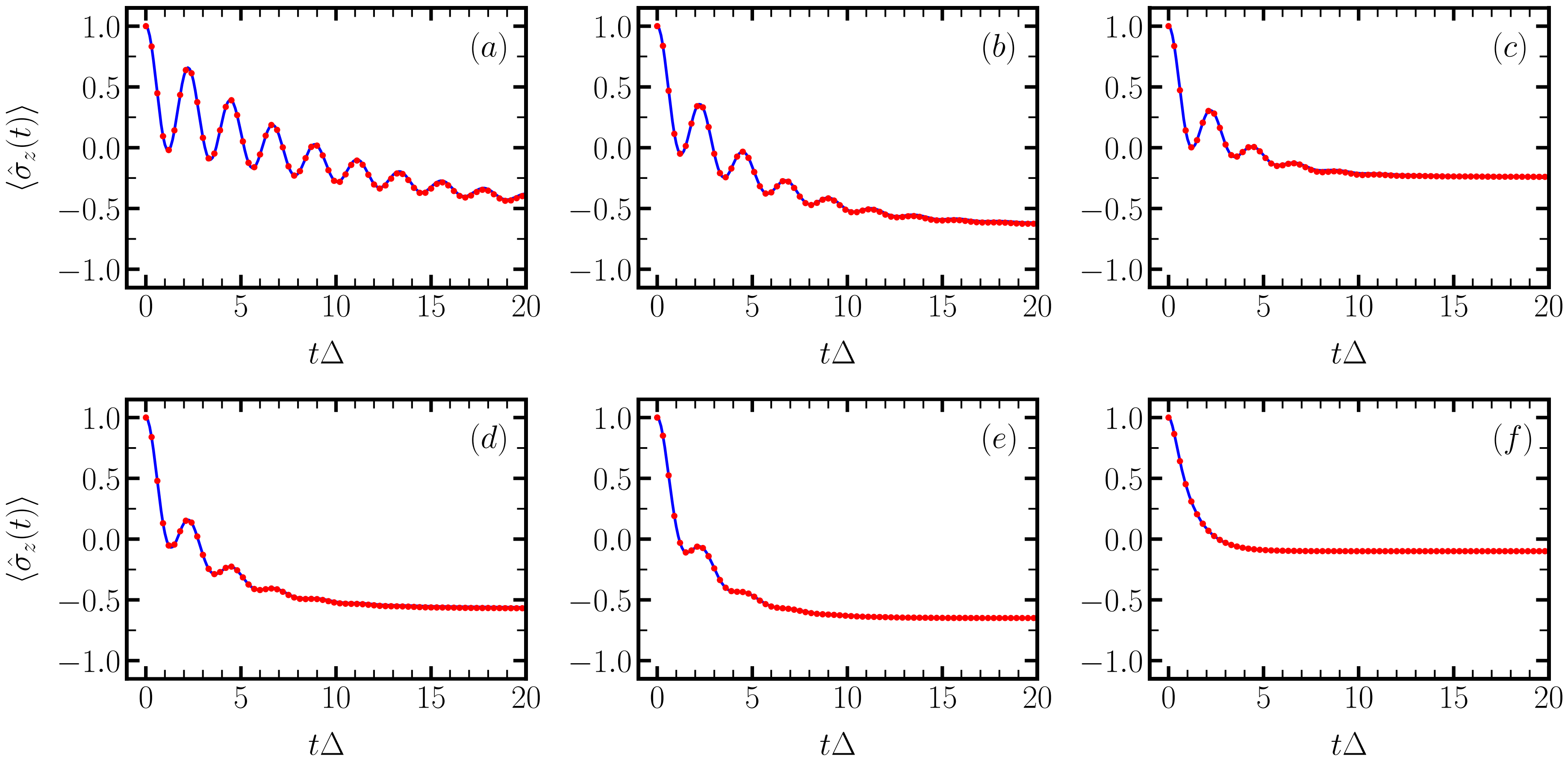
$$PE_{j,k} = \begin{cases} \sin(t_j \omega_k), k \text{ is even} \\ \cos(t_j \omega_k), k \text{ is odd} \end{cases}$$

$$\omega_k = \frac{1}{1000^{2k}/d_p}$$

Non-trainable PE
(extensions to trainable PE exist)



Asymmetric spin-boson system

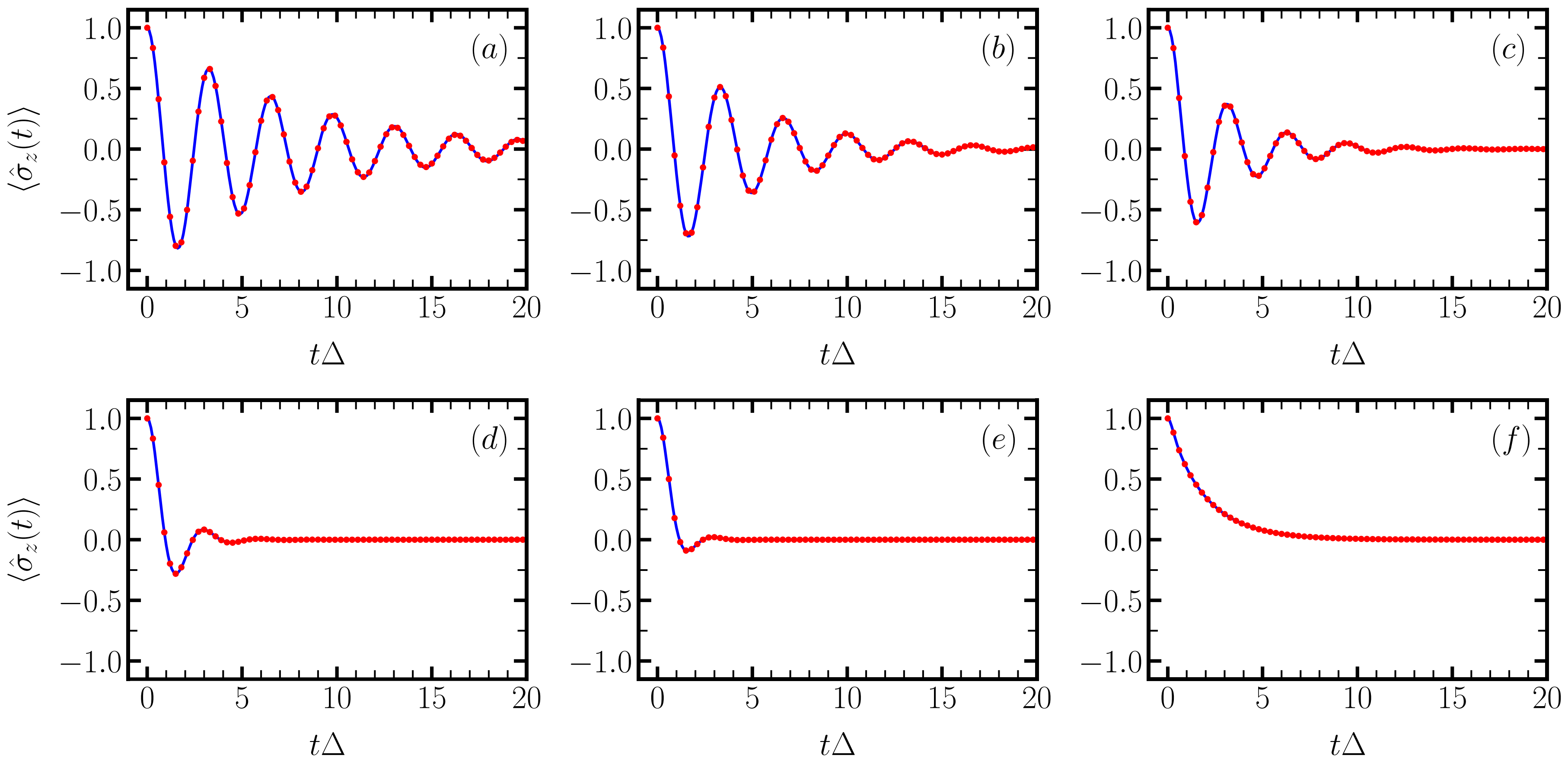


Lowest single-time step prediction error $7.5 \cdot 10^{-3}$

Number of trainable parameters: 1,918,018

*L. Rodriguez and **AAK**, under review*

Symmetric spin-boson system

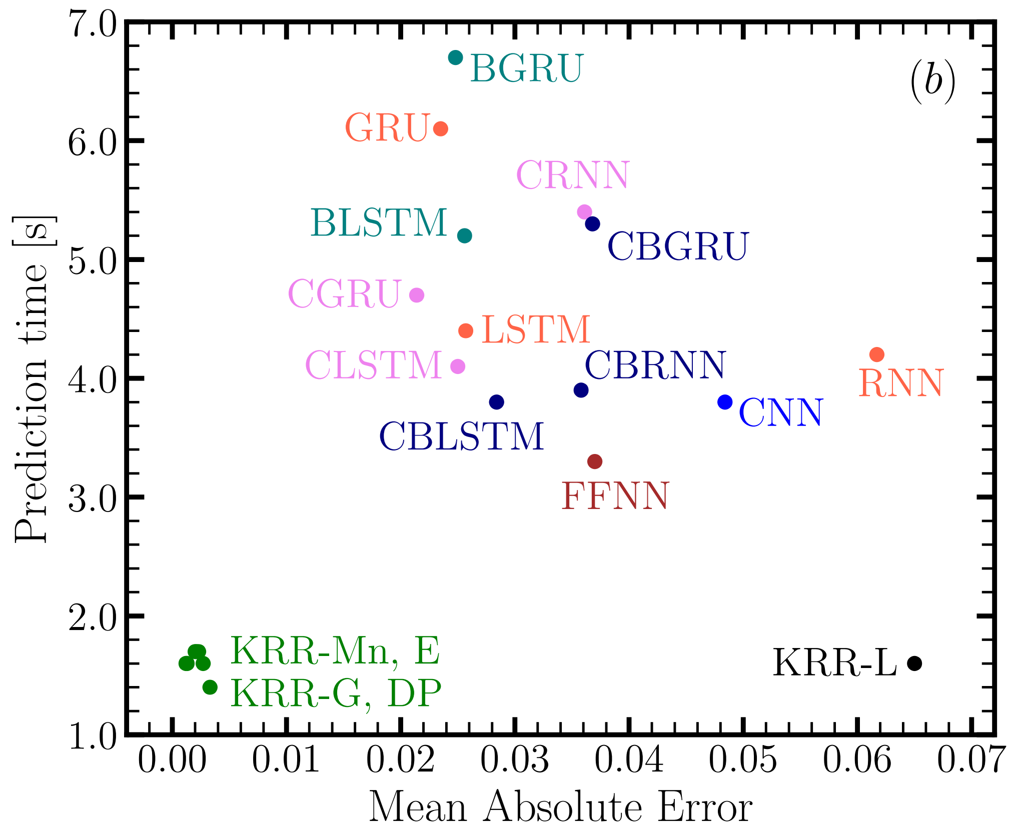


Lowest single-time step prediction error $4.3 \cdot 10^{-4}$

*L. Rodriguez and **AAK**, under review*

Conclusions

- ML can be an efficient way to simulate long-time quantum dynamics
- KRR is the fastest most accurate method for this, but they are restricted to fixed-size input (need to know the memory)
- RNNs work with input of different size: CGRU is the best of them
- Transformers can reach accuracy of best KRR models



Papers, codes, and data sets

*L. Rodriguez and **AAK**, J. Phys. Chem. Lett. 12, 2476 (2021)*

*L. Rodriguez, A. Ullah, P. O. Dral, **AAK**, Mach. Learn.: Sci. Technol. 3 045016 (2022)*

*L. Rodriguez and **AAK**, (under review)*

*A. Ullah, L. Rodriguez, P. O. Dral, and **AAK**, Front. Physics 11, 1223973 (2023)*

TLS and FMO dynamics data sets: https://doi.org/10.25452/figshare.plus.c.6389553

Github: https://github.com/kananenka-group

Transformers tutorial (Luis):
https://github.com/leherrer/Transformer_QD/tree/main