Fourier Learning with Cyclical Data

Presenter: Yingxiang Yang

A joint work with: Zhihan Xiong, Tianyi Liu, Taiqing Wang, Chong Wang

Why are Cyclical Data Interesting?

- Many predictive models are trained on data with cyclical properties.

Recommender Systems

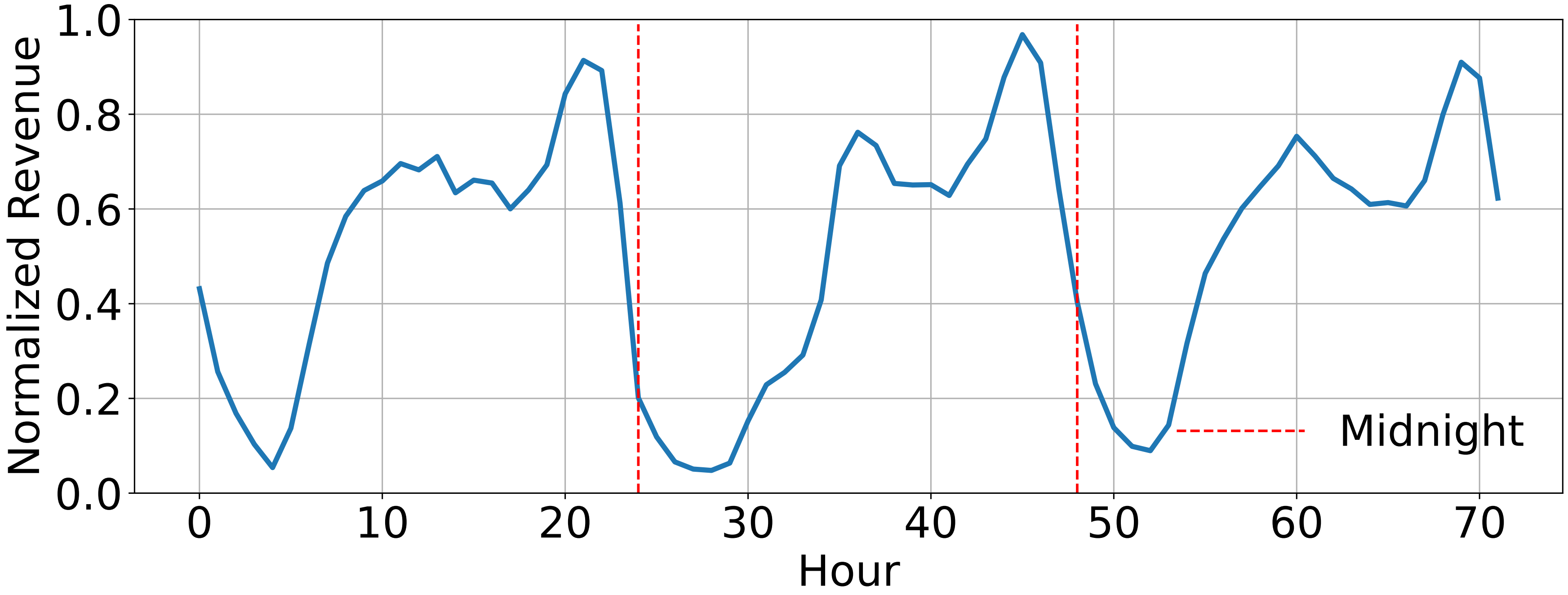
Financial Markets

Search Engines

Why are Cyclical Data Interesting?

- Mechanisms that routinely repeat themselves result in periodicity.

  Recommender systems: user lifestyle.



- **Research question:**

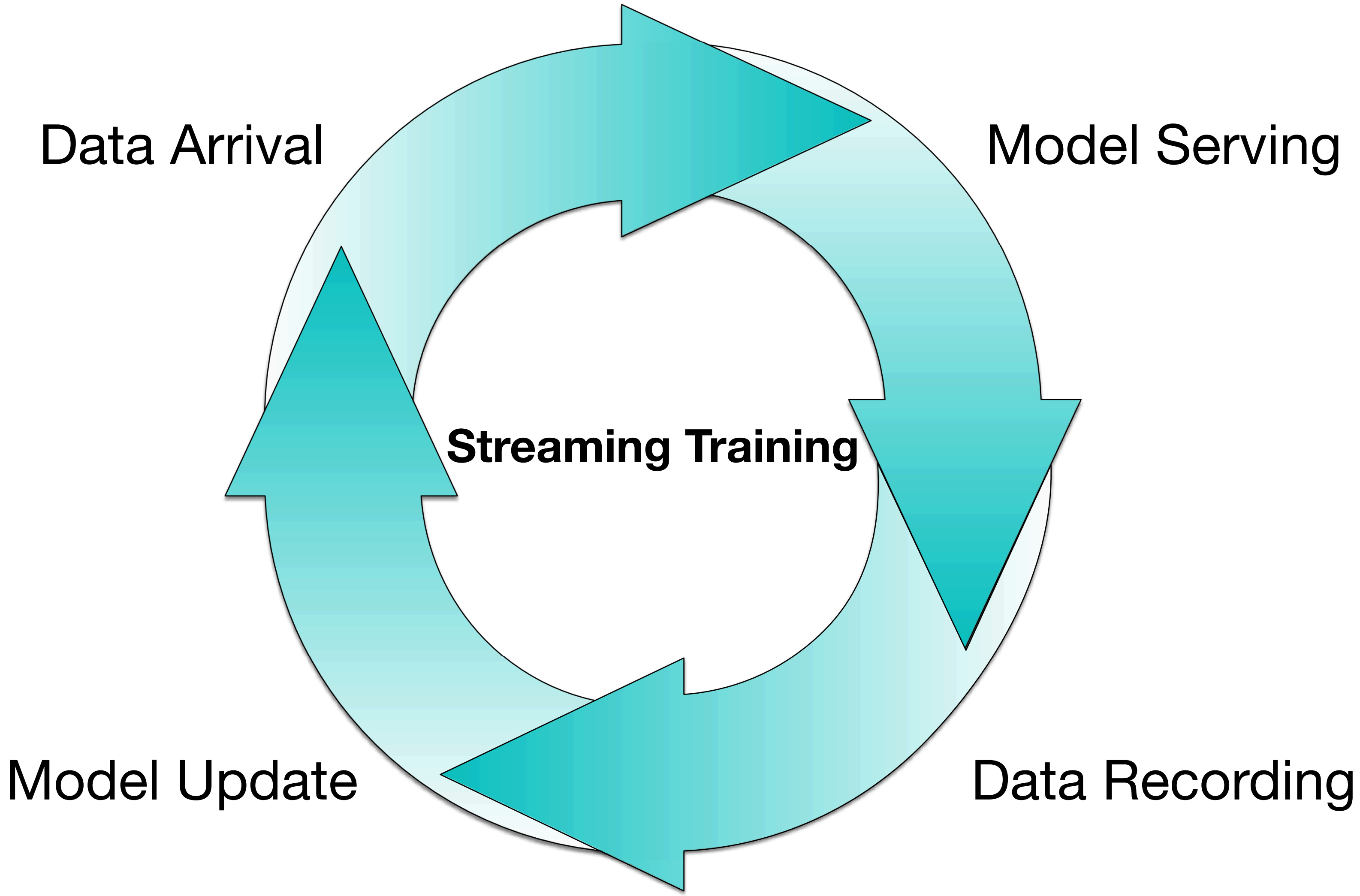
Can we exploit periodicity to improve prediction performance?

- **Research question:**

  Can we exploit periodicity to improve prediction performance?

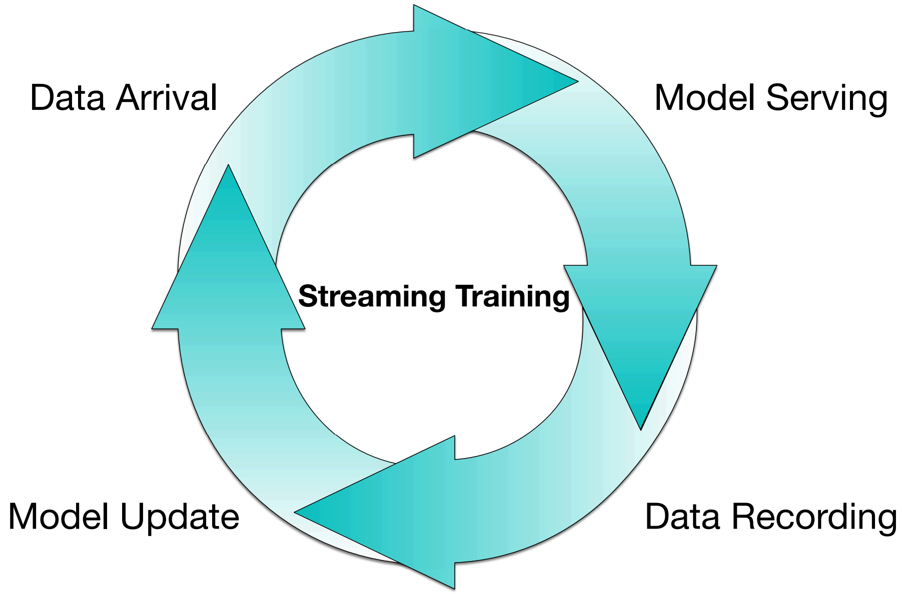
- **Focus: large models, streaming training.**

Problem Setup: Streaming Training



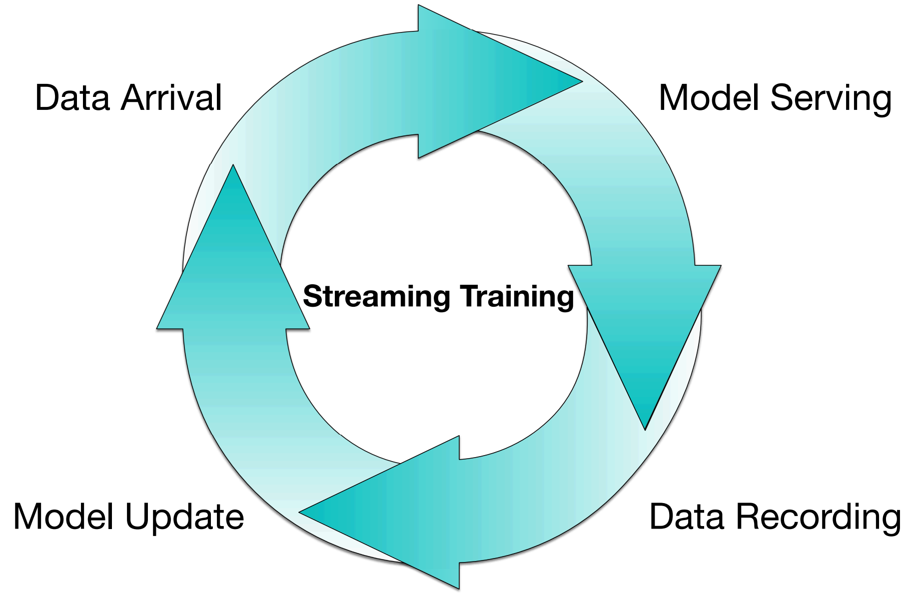
Problem Setup: Streaming Training

- Procedure: a 4 step cycle.



Problem Setup: Streaming Training

- Procedure: a 4 step cycle.
- Goal: serve with the best model for every t.

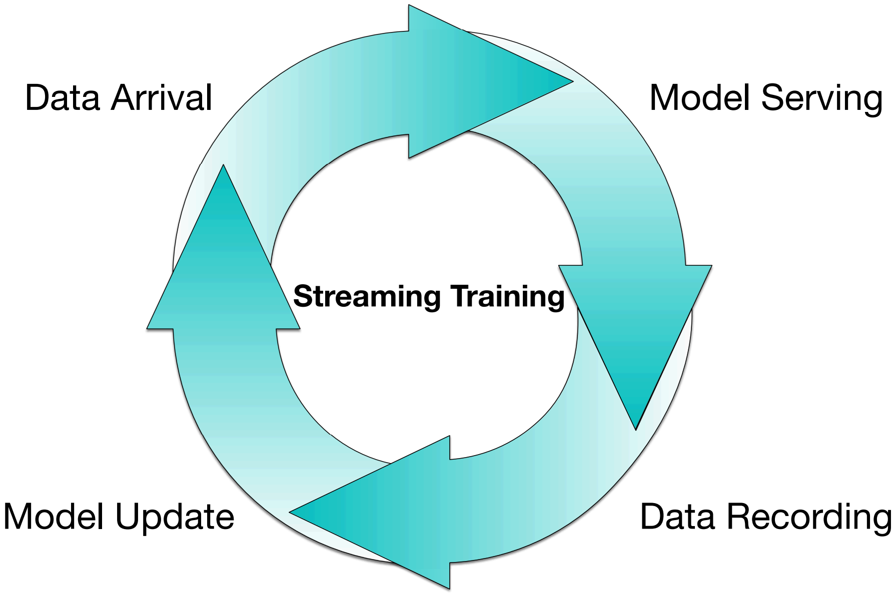


Solve $f_t^*(x) \in \underset{f}{\operatorname{argmin}} \mathbb{E}_{(x,y)\sim\mathcal{D}_t} [\ell(f(x), y)] \quad \forall t \in \mathbb{R}$

- x, feature; y, label; t, time.
- f, model; ℓ, loss function.
- $\mathcal{D}_t$, time-dependent data distribution.
- T, **periodicity**, i.e. $\mathcal{D}_t = \mathcal{D}_{t-T}$.

Problem Setup: Streaming Training

- Procedure: a 4 step cycle.
- Goal: serve with the best model for every t.



Solve $f_t^*(x) \in \underset{f}{\operatorname{argmin}} \mathbb{E}_{(x,y)\sim\mathcal{D}_t}\left[\ell(f(x),y)\right] \quad \forall t \in \mathbb{R}$

- x, feature; y, label; t, time.
- f, model; ℓ, loss function.
- $\mathcal{D}_t$, time-dependent data distribution.
- T, **periodicity**, i.e. $\mathcal{D}_t = \mathcal{D}_{t-T}$.

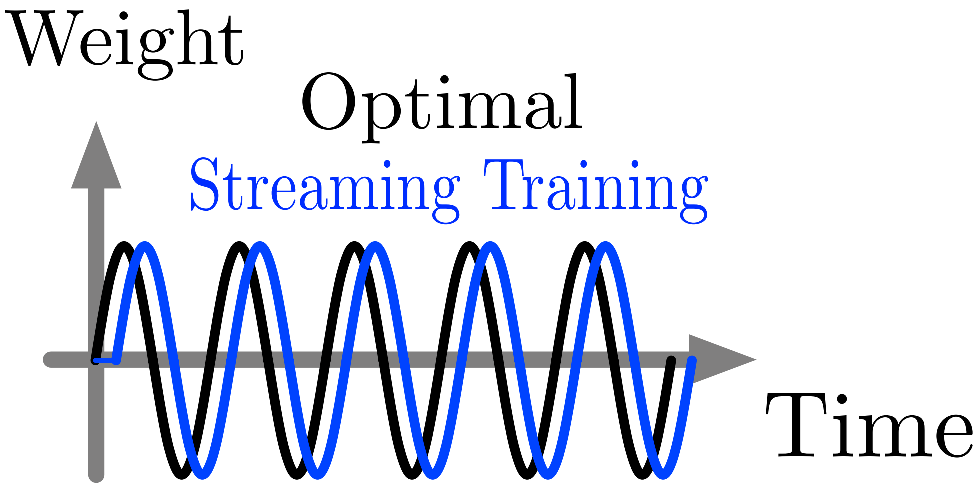
Challenge: data distribution continuously evolves over time.

Good news: the data arriving at time t come from the same distribution as t-T.

Question: how to access past information in streaming training?

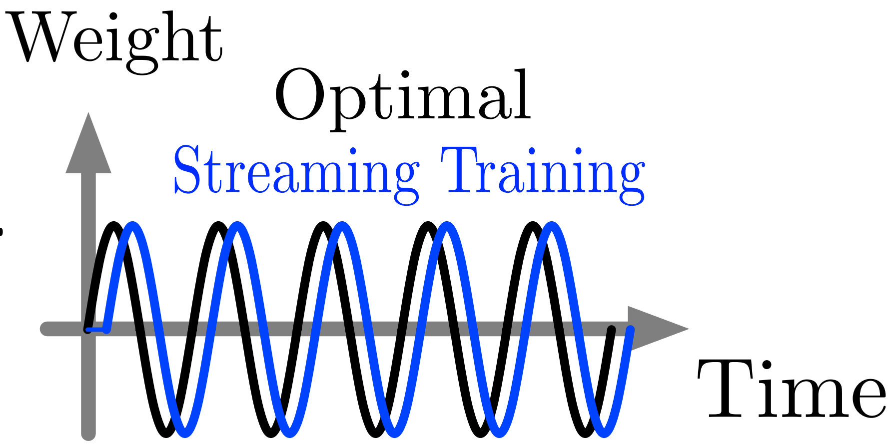
Existing Approaches

- **Streaming training: does not exploit periodicity.**
  - **Learned model has a lag.**



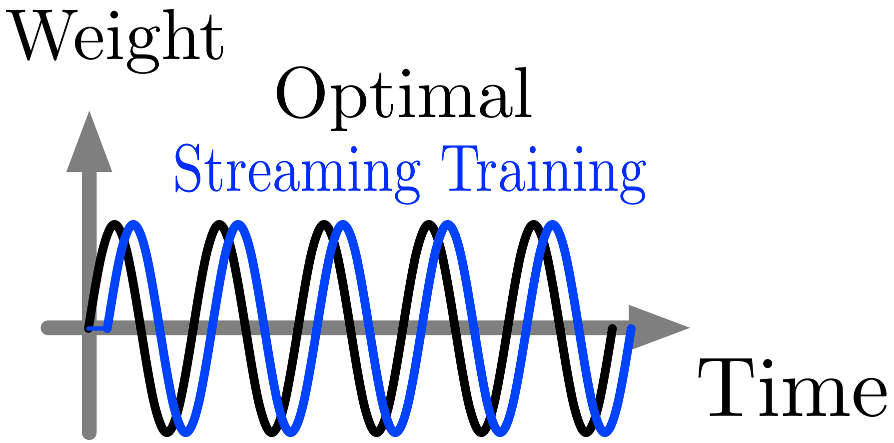
Existing Approaches

- Streaming training: does not exploit periodicity.
  - Learns a set of models with a lag.

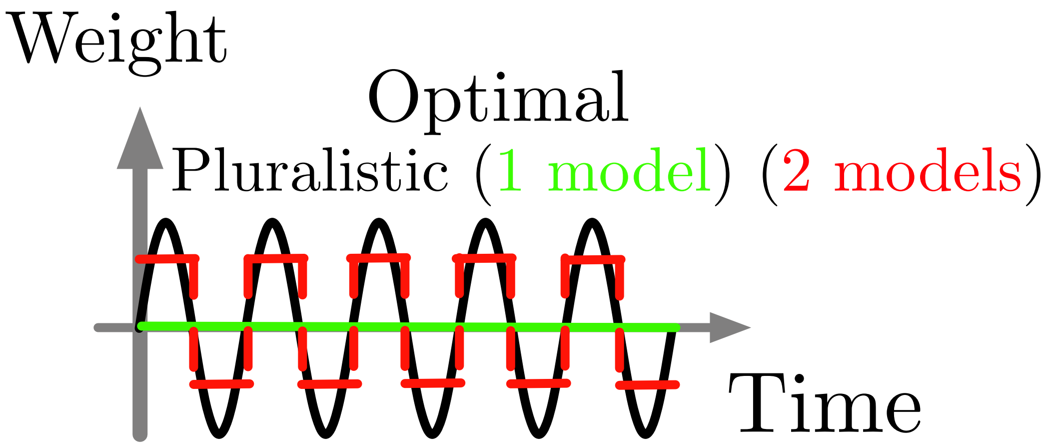


- **Increasing model capacity (adding a time feature).**
  - **Requires feature engineering.**
  - **Hard to adapt to other models.**

Existing Approaches



- Streaming training: does not exploit periodicity.
  - Learns a set of models with a lag.
- Increasing model capacity (adding a time feature).
  - Requires feature engineering.
  - Hard to adapt to other models.
- **Improving streaming training framework.**
  - **Pluralistic [1]: learns a model for every t.**
    - **Hard to scale, approximation error.**



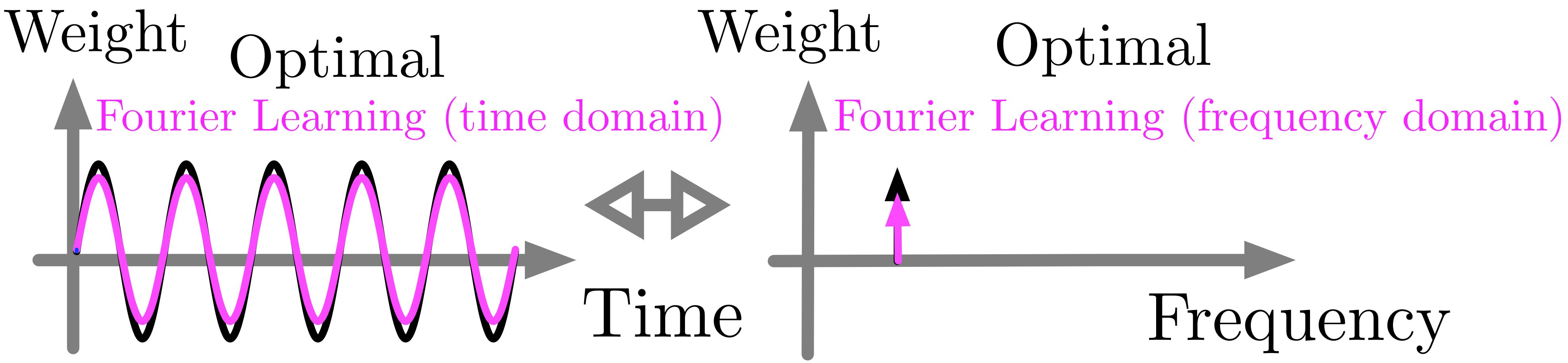
[1] Eichner, Hubert, et al. "Semi-cyclic stochastic gradient descent." ICML, 2019.

Our Approach: Fourier Learning

- **Learns a frequency representation instead.**

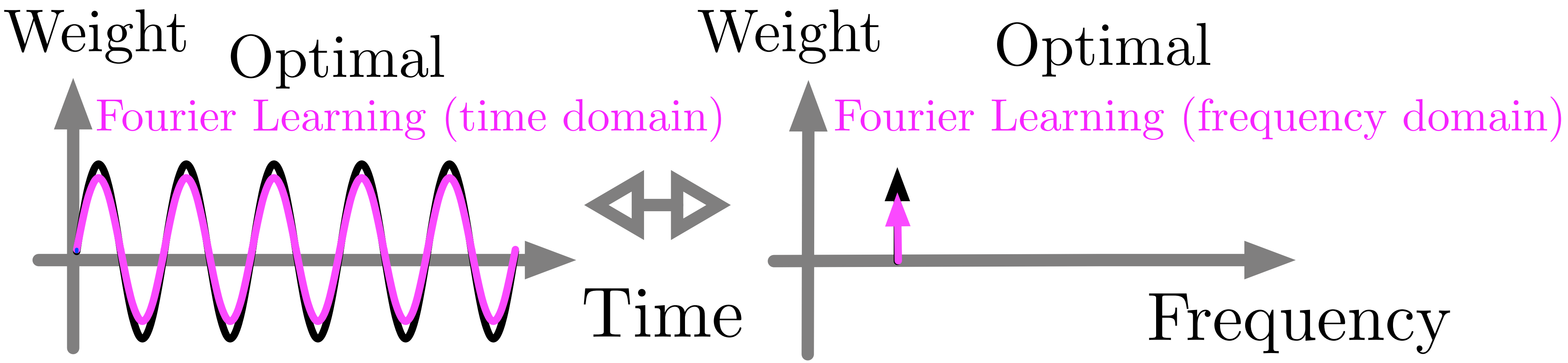
Our Approach: Fourier Learning

- **Learns a frequency representation instead.**



Our Approach: Fourier Learning

- **Learns a frequency representation instead.**

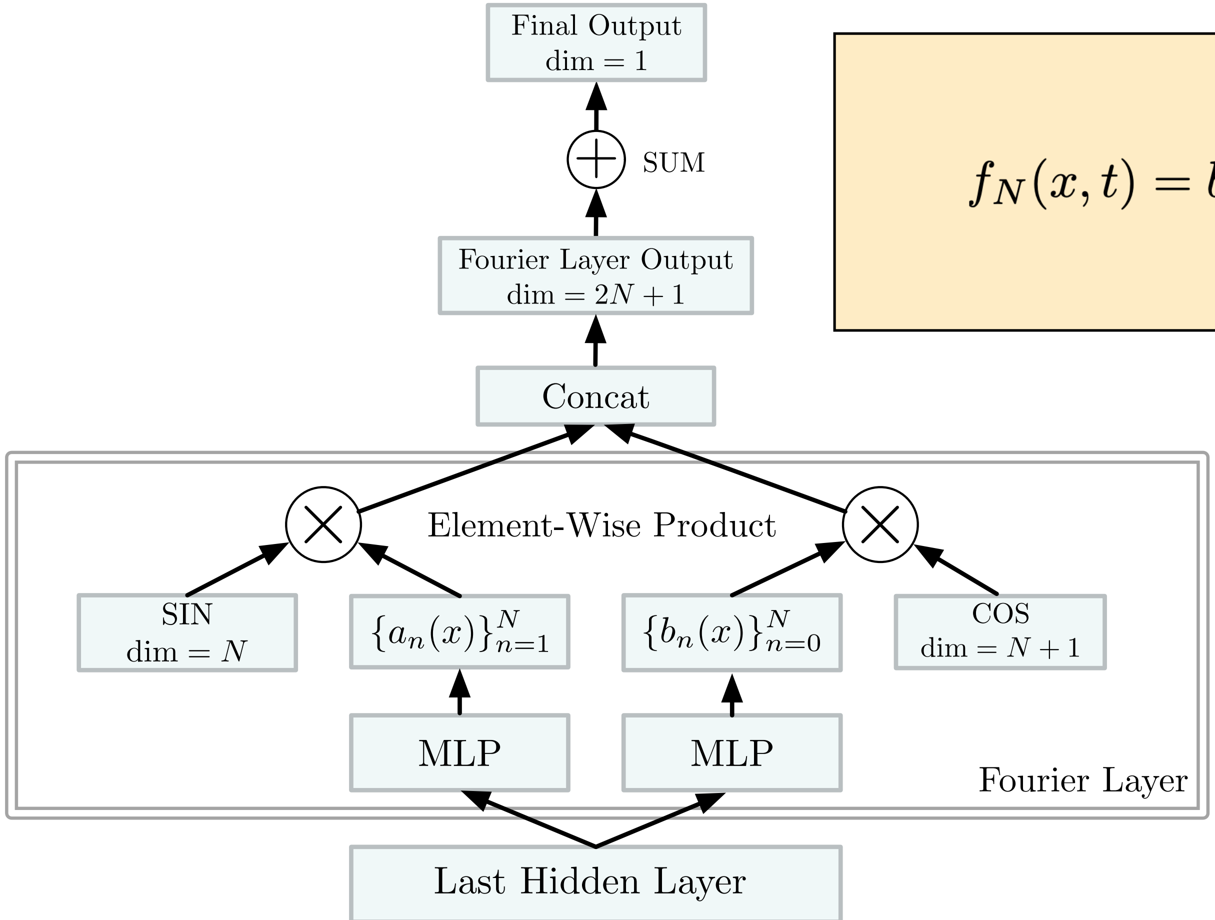


$$f_N(x,t) = b_0(x) + \sum_{n=1}^{N}\left[a_n(x)\sin\left(\frac{2\pi nt}{T}\right) + b_n(x)\cos\left(\frac{2\pi nt}{T}\right)\right]$$

- Learning goal: learn $a_n(x)$ and $b_n(x)$.

Our Approach: Fourier Learning

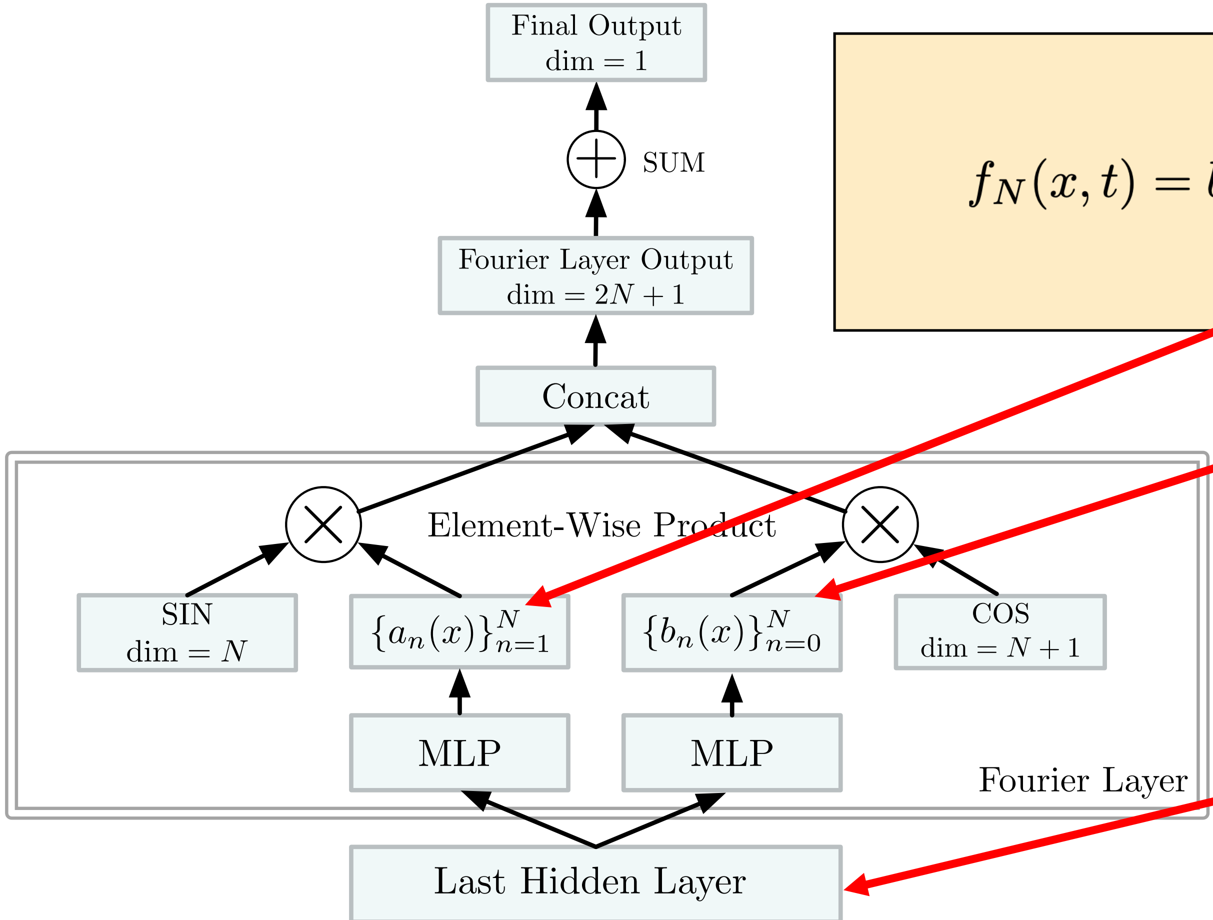
- Integration with deep learning: Fourier-MLP.



$$f_N(x,t) = b_0(x) + \sum_{n=1}^{N} \left[a_n(x) \sin\left(\frac{2\pi nt}{T}\right) + b_n(x) \cos\left(\frac{2\pi nt}{T}\right) \right]$$

Our Approach: Fourier Learning

- Integration with deep learning: Fourier-MLP.

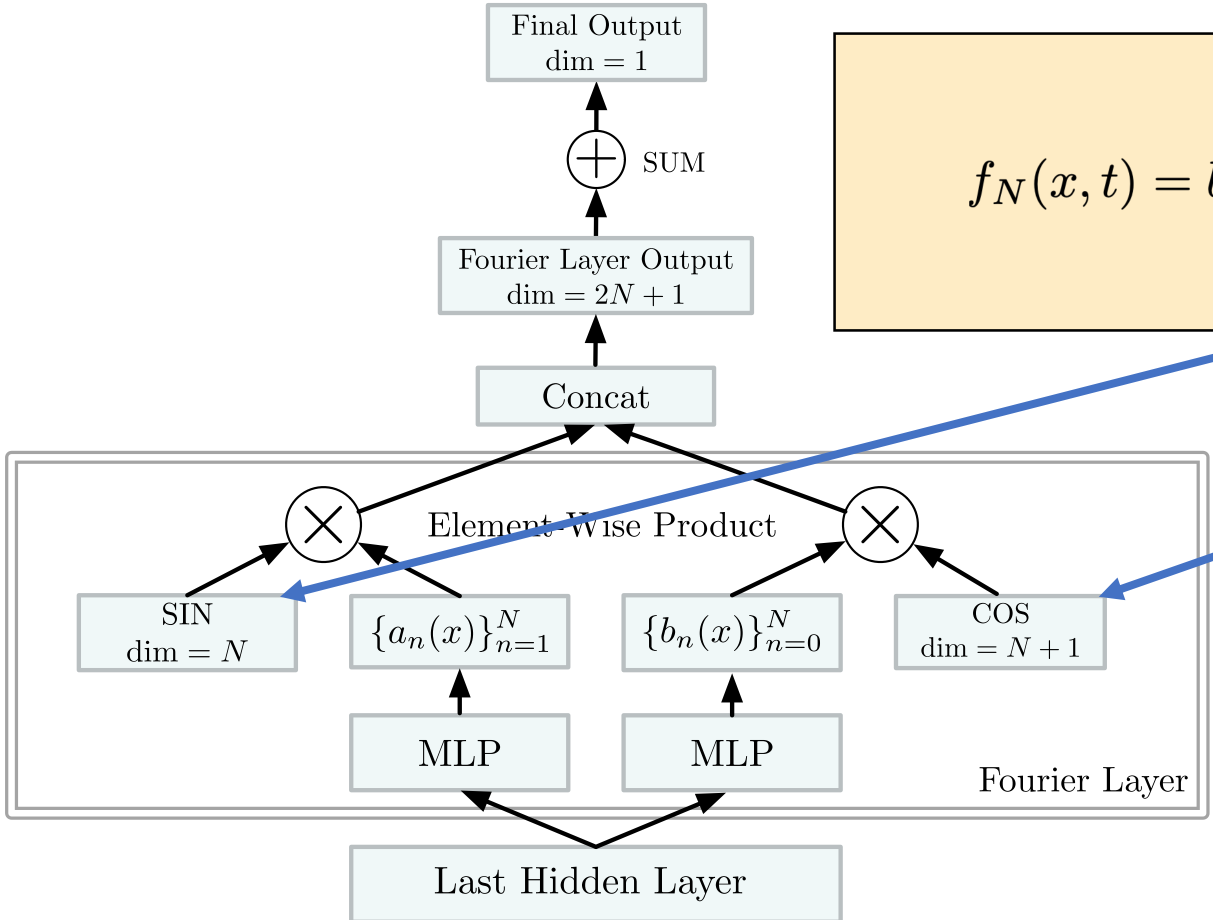


$$f_N(x,t) = b_0(x) + \sum_{n=1}^{N} \left[a_n(x) \sin\left(\frac{2\pi n t}{T}\right) + b_n(x) \cos\left(\frac{2\pi n t}{T}\right) \right]$$

$a_n(x)$ and $b_n(x)$ come from the last layer of MLP.

Our Approach: Fourier Learning

- Integration with deep learning: Fourier-MLP.



$$f_N(x,t) = b_0(x) + \sum_{n=1}^{N} \left[a_n(x) \sin\left(\frac{2\pi n t}{T}\right) + b_n(x) \cos\left(\frac{2\pi n t}{T}\right) \right]$$

$a_n(x)$ and $b_n(x)$ come from the last layer of MLP.

They are then combined with time-dependent weights.

Our Approach: Fourier Learning

- Advantages:
  - Provably converges to a fixed optimal under streaming training.
  - Easily integrable with neural networks and adapts to a wide variety of models.
- Disadvantage:
  - Requires data distribution to be strictly periodic. [2]

[2] Fan, Wei, et al. “DEPTS: Deep expansion learning for periodic time series forecasting.” ICLR, 2022.

More details at our poster session.