ZHEJIANG UNIVERSITY

Tracking Functional Changes in Nonstationary Signals with Evolutionary Ensemble Bayesian Model for Robust Neural Decoding

Xinyun Zhu, Yu Qi*, Kedi Xu, Junming Zhu, Jianmin Zhang, Yueming Wang*

Presented by 祝歆韵 (Ph. D Student)

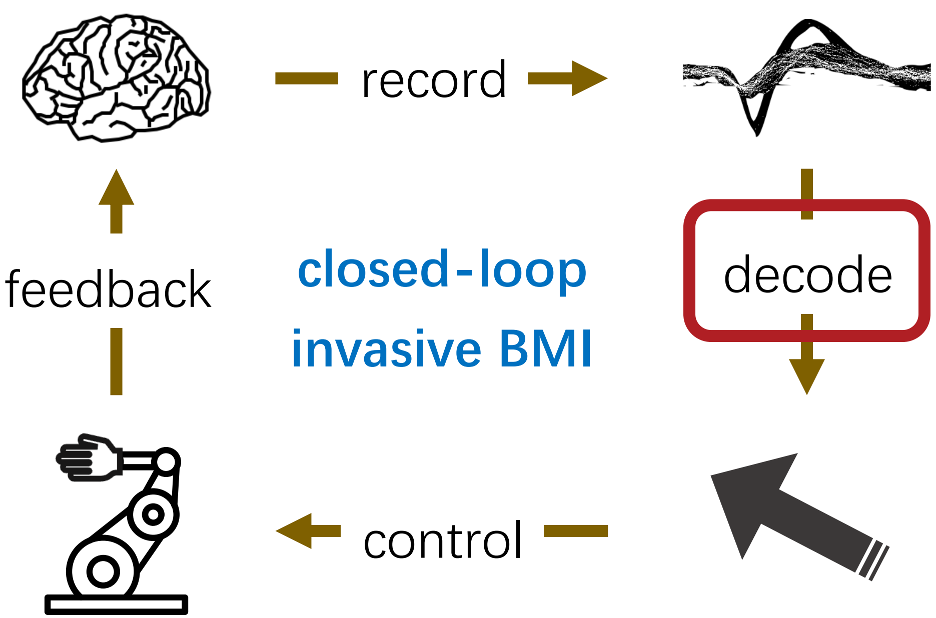
Zhejiang University

Invasive **Brain-machine Interface** (iBMI)

Brain ⟵ control & communication pathway ⟶ **External Device**

such as：computer cursor、robotic arms、exoskeleton

- **Online、closed-loop control**



- **System Process：**

① **Collect** neural signals from the brain

② **Decode** kinematics from neural signals through **decoding algorithm**

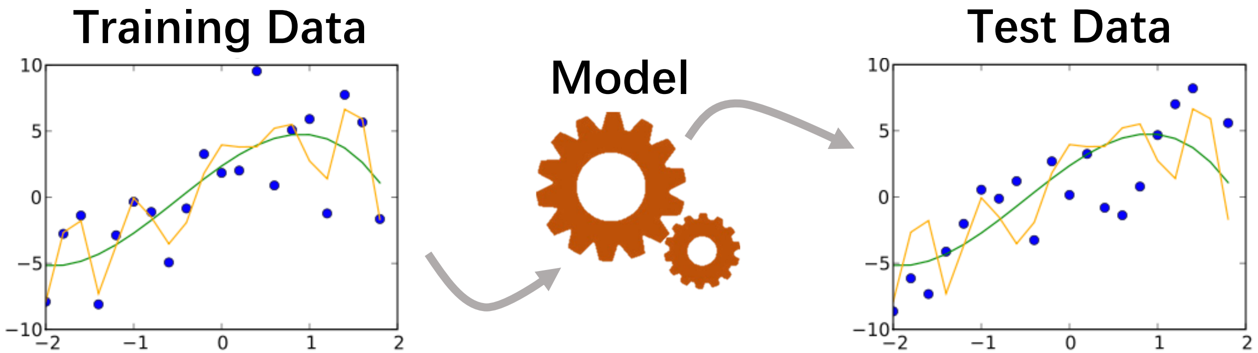
③ **Control** external devices by decoded kinematics

④ Participant adjusts behaviour according to the **feedback**

- **Key：decoding algorithm**

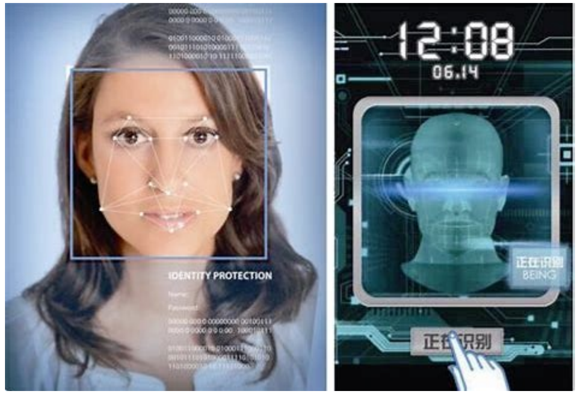
Dynamic World and Static Models

- **Typical model training pipeline**
  - ➢ use a static model
  - ➢ assumes the data distribution is fixed and stable in time

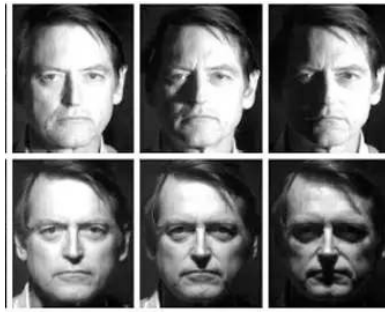


- **It would fail if the assumption is not satisfied ...**

For example, in face recognition：



illumination



expression

ages

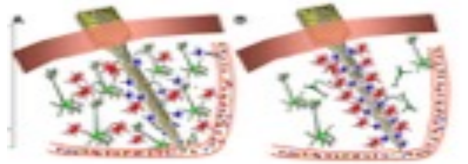
A fixed model can not be optimized for every situation.

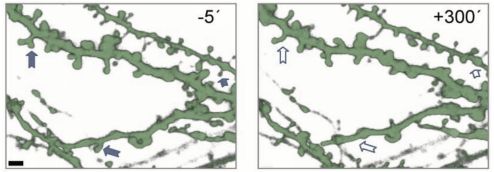
Brain Signals are Typical Nonstationary Data

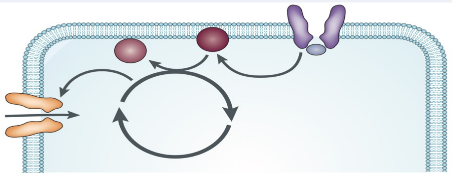
- **Cause ：**

① neuron-related factors

scar tissue build-up	neuroplasticity	nervous system dynamics

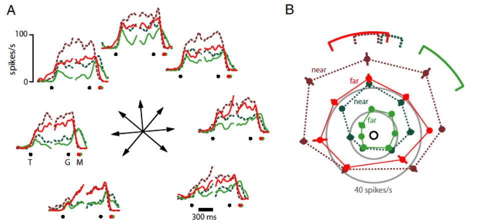


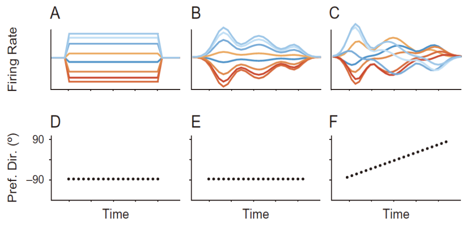


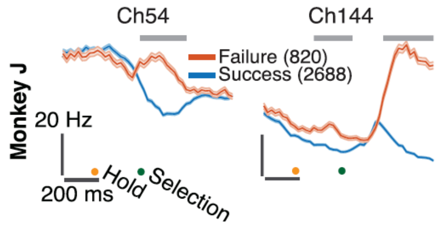


② task-related factors

speed[1][2]	period[3][4]	error[5]





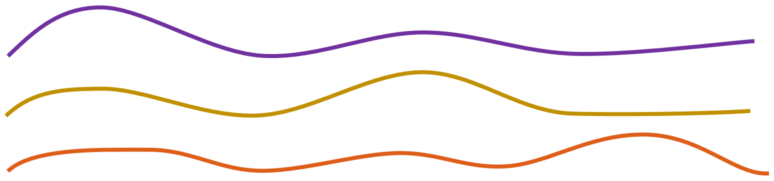


③ external noise

[1] Schwartz, Nature Comm 2018, Decoding arm speed during reaching.
[2] Shenoy, JNP 2006, Preparatory activity in premotor and motor cortex reflects the speed of the upcoming reach.
[3] Schwartz, Cerebral Cortex 2018, Temporally Segmented Directionality in the Motor Cortex.
[4] Flash, Cerebral Cortex 2019, Movement Decomposition in the Primary Motor Cortex.
[5] Shenoy, JNE 2017, Augmenting intracortical brain-machine interface with neurally driven error detectors.

- **Example ：**

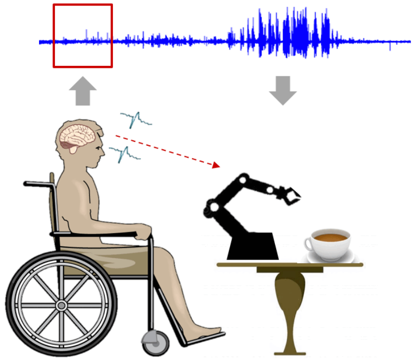
variability mode 1 | variability mode 2 | variability mode 3



Different mode dominate at different times.

- **Challenge ：**

Single & static decoder leads to **unstable** BMI control

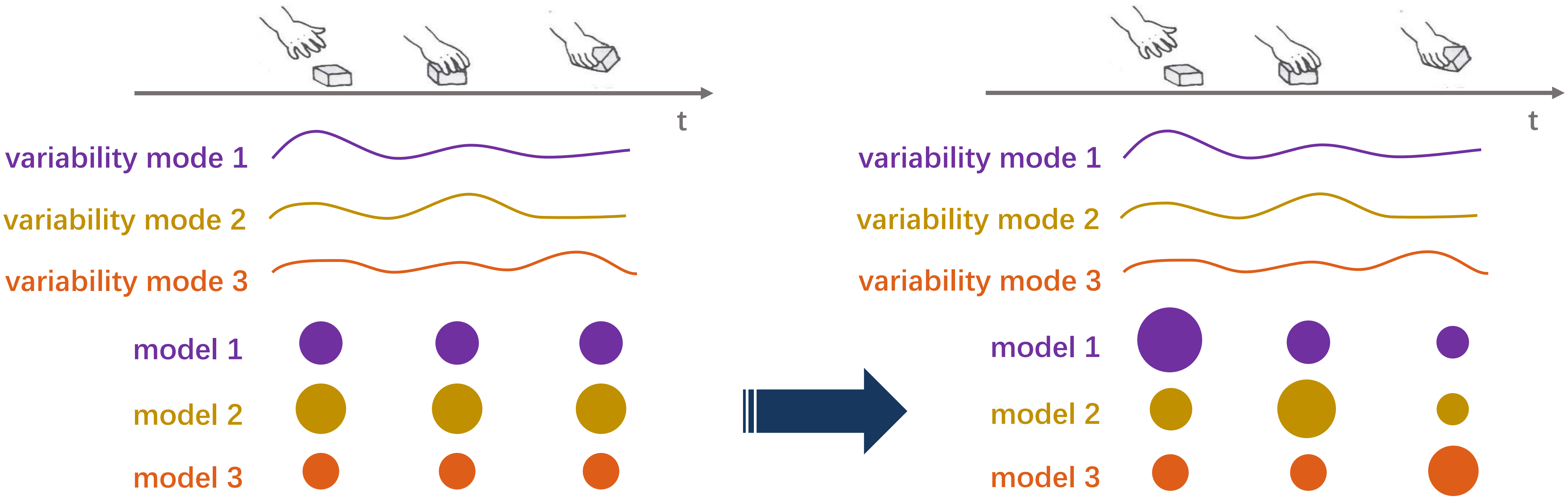


How to Dynamically Cope with Nonstationary Data?

- **Main insights** ：

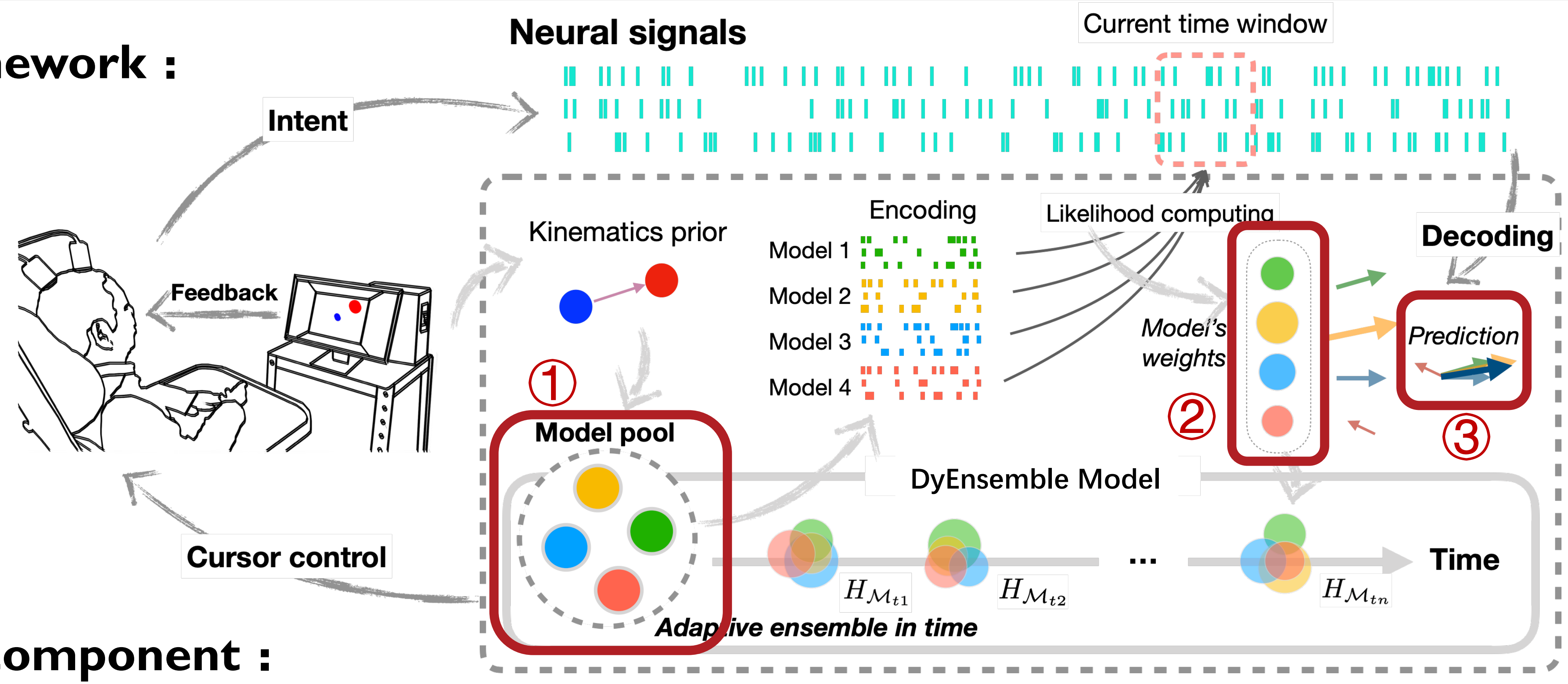
  single model is not enough → assembling **multiple** model

  static model weights → **dynamic** model weights changing with neural signals



Previous Work : DyEnsemble Framework

- **Framework :**



- **Key component :**

① construct model pool ② calculate model weights ③ assemble models

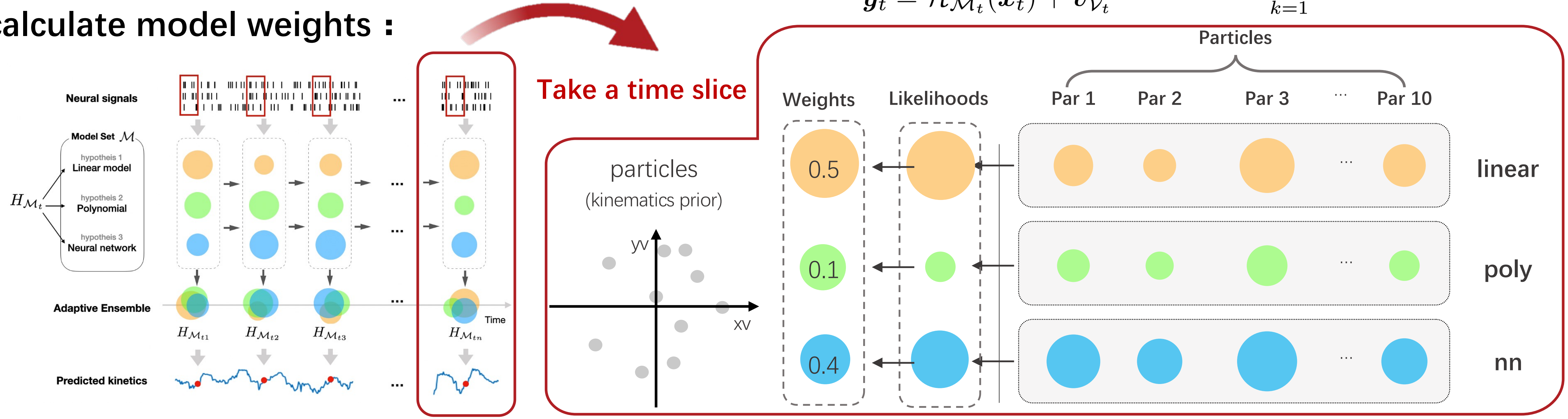
Previous Work : DyEnsemble Framework

① construct model pool : linear | poly | nn1 | nn2

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + \boldsymbol{b} + \boldsymbol{u}_{t-1}$$
$$\boldsymbol{y}_t = \mathcal{H}_{\mathcal{M}_t}(\boldsymbol{x}_t) + \boldsymbol{v}_{\mathcal{V}_t}$$

$$\mathcal{H}_{\mathcal{M}_t} = \sum_{k=1}^{q} w_{k,t} m_k(.)$$

② calculate model weights :



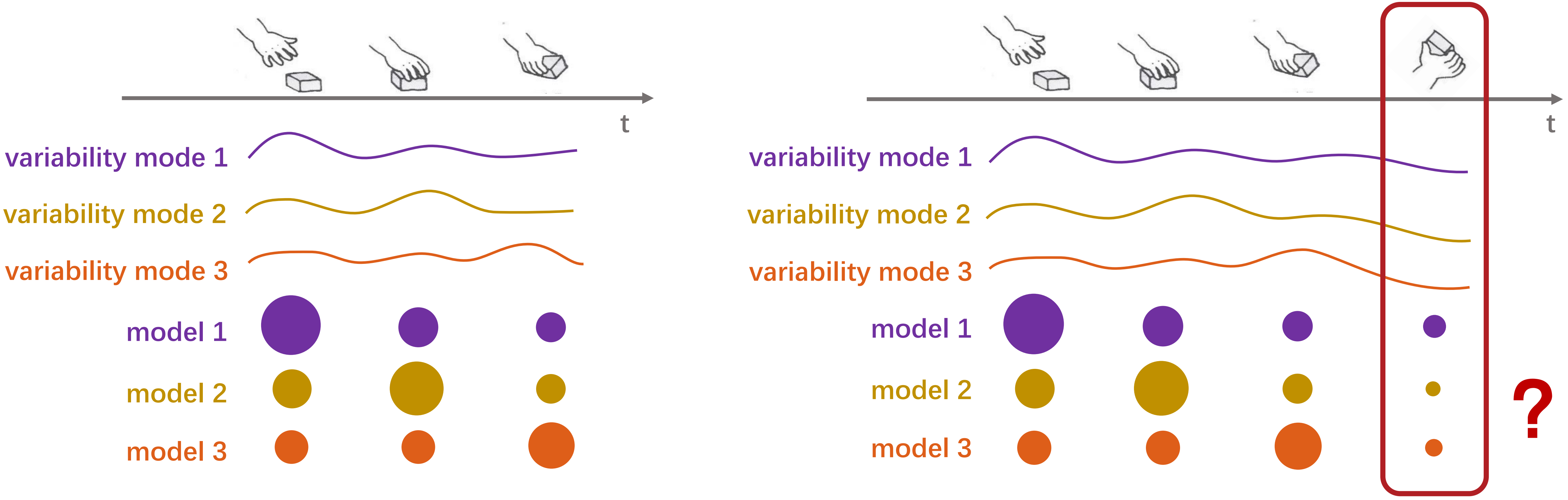
③ **assemble models :** single model's estimated velocity = **the weighted sum** of all single model's particles

all model's estimated velocity = **the weighted sum** of all single model's estimated velocity

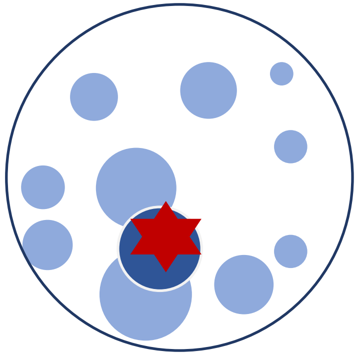
$$p_k(\boldsymbol{y}_t|\boldsymbol{y}_{0:t-1}) \approx \sum_{i=1}^{N_s} \omega_{t-1}^i p_m(\boldsymbol{y}_t|\boldsymbol{x}_t^i) \qquad p_k(\boldsymbol{x}_t|\mathcal{H}_{\mathcal{M}_t} = m_k, \boldsymbol{y}_{0:t}) \approx \sum_{i=1}^{N_s} \omega_{k,t}^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i)$$

DyEnsemble : Dynamic Model Weights but Static Model Pool

- **What if the likelihood of all models in the model pool is small ?**

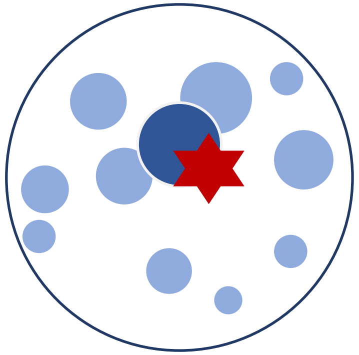


DyEnsemble : Dynamic Model Weights but Static Model Pool



- **a static model pool**

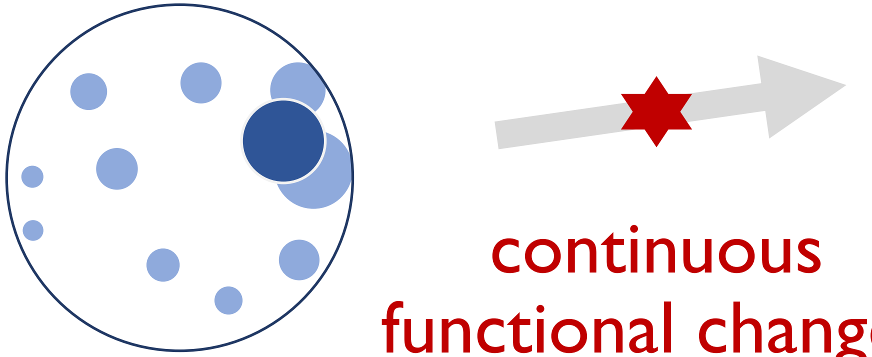
DyEnsemble : Dynamic Model Weights but Static Model Pool



- **only addressing neural changes in a certain range**

DyEnsemble : Dynamic Model Weights but Static Model Pool

- Model **weights** change
- Model **parameter** change

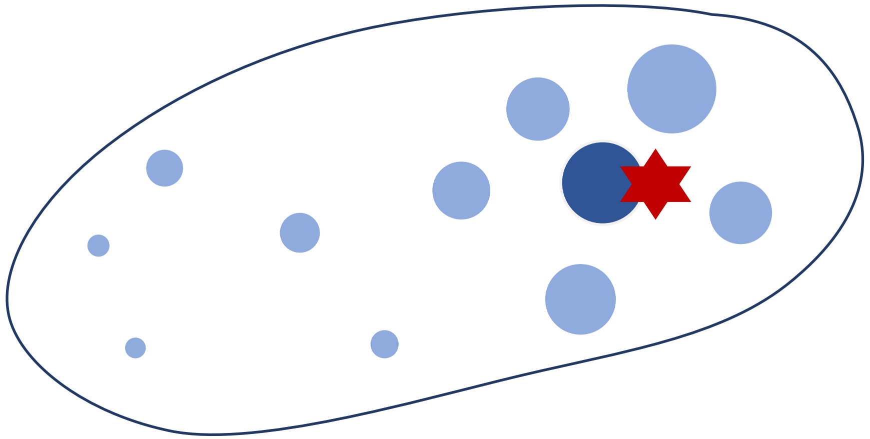


continuous functional changes

How to cope with the functional changes in the neural system?

EvoEnsemble : Dynamic Weights and Dynamic Model Pool

- Model **weights** change ✓
- Model **parameter** change ✓



Evolutionary Ensemble Bayesian Filter (EvoEnsemble)

EvoEnsemble : Differential Evolution Algorithm

- **Main idea :**

  Evolve individuals towards best individual with a certain randomness

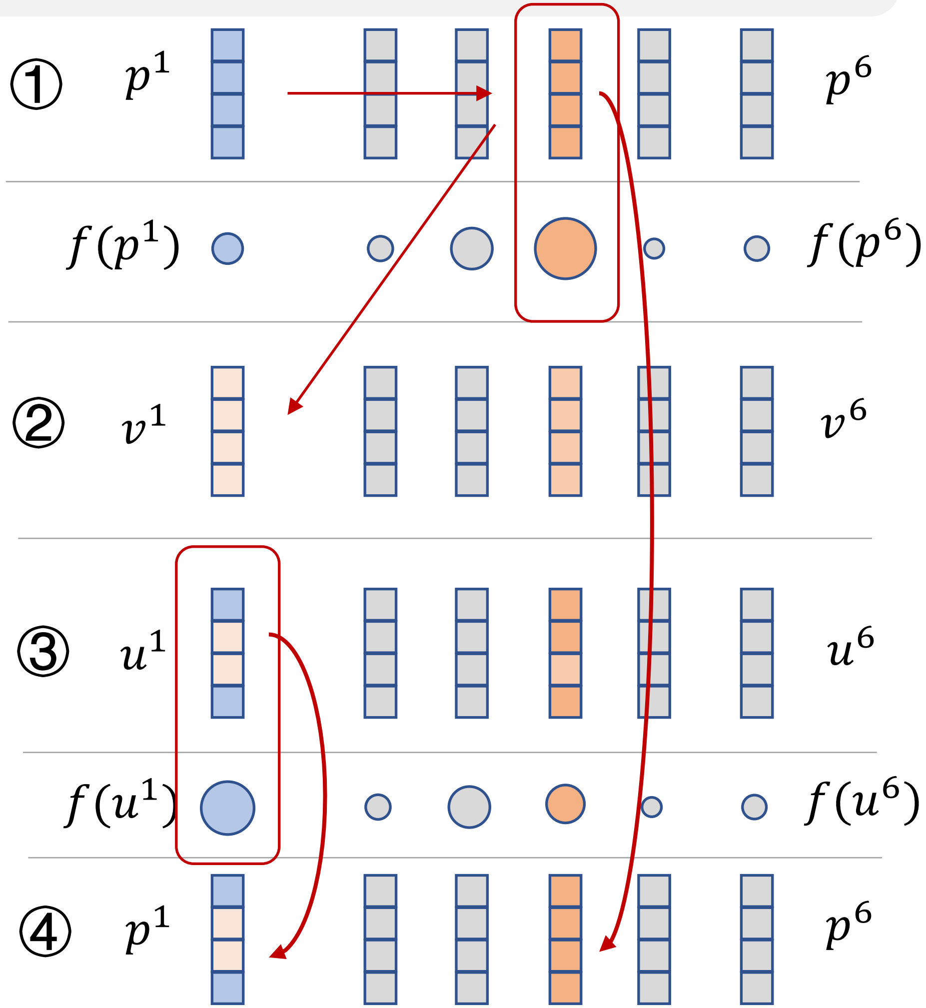
- **Steps :**

  ① **Initialization :** $\mathcal{M} = \{\mathbf{p}^k | k = 1, 2, ..., N\}$

  ② **Mutation :** $\mathbf{v}_G^k = \mathbf{p}_G^k + F_k \cdot (\mathbf{p}_G^{\text{pbest}} - \mathbf{p}_G^k) + F_k \cdot (\mathbf{p}_G^{r_k^1} - \tilde{\mathbf{p}}_G^{r_k^2})$

  ③ **Crossover :** $\boldsymbol{u}_{j,G}^k = \begin{cases} \boldsymbol{v}_{j,G}^k, & \text{if rand}(0,1) \leq CR_k \text{ or } j = j_{\text{rand}} \\ \boldsymbol{p}_{j,G}^k, & \text{otherwise} \end{cases}$

  ④ **Selection :** $\mathbf{p}_{G+1}^k = \begin{cases} \mathbf{u}_G^k, & \text{if } f(\mathbf{u}_G^k) > f(\mathbf{p}_G^k) \\ \mathbf{p}_G^k, & \text{otherwise} \end{cases}$

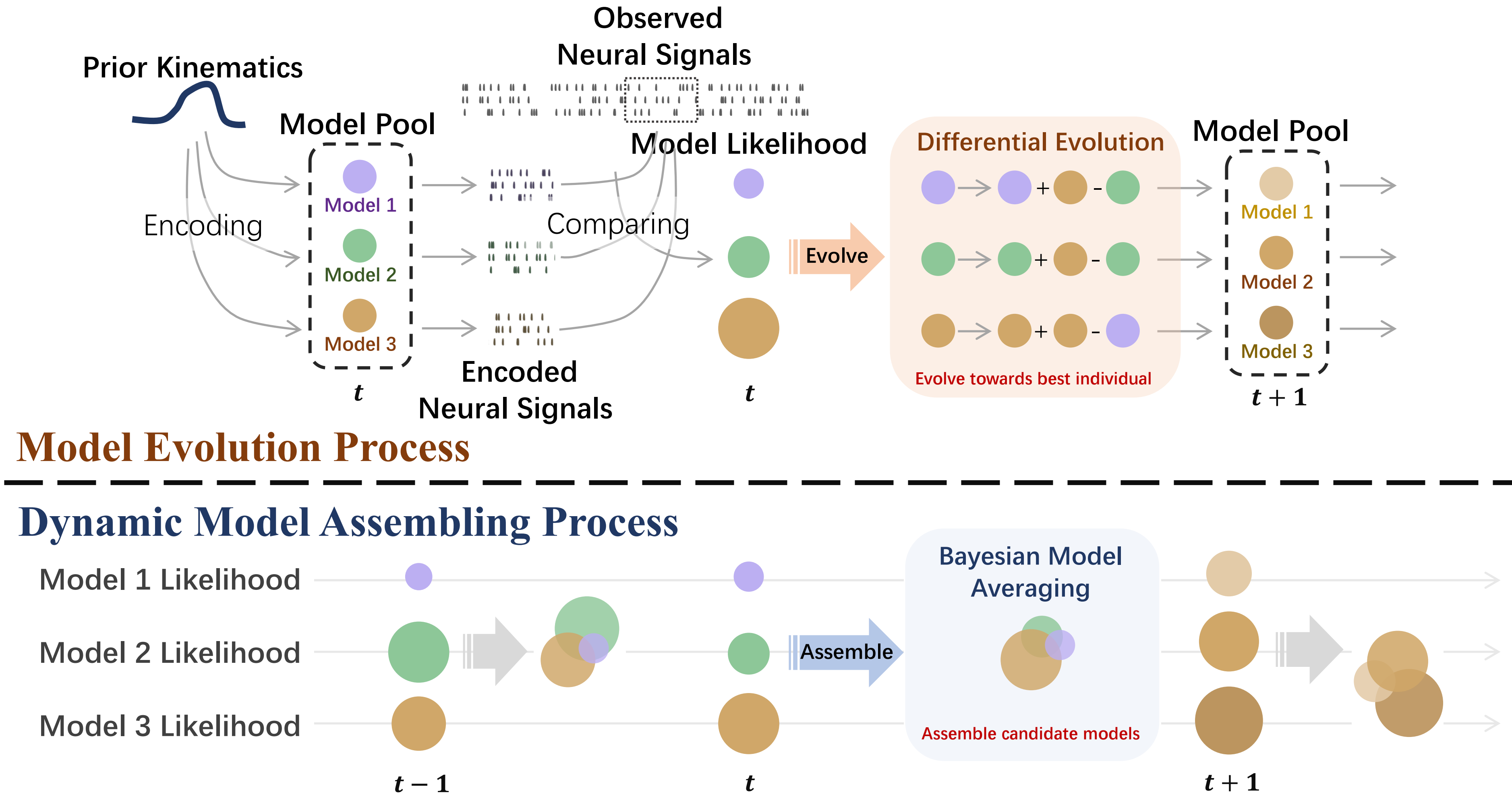


EvoEnsemble : Evolution Fitness = Bayesian Likelihood

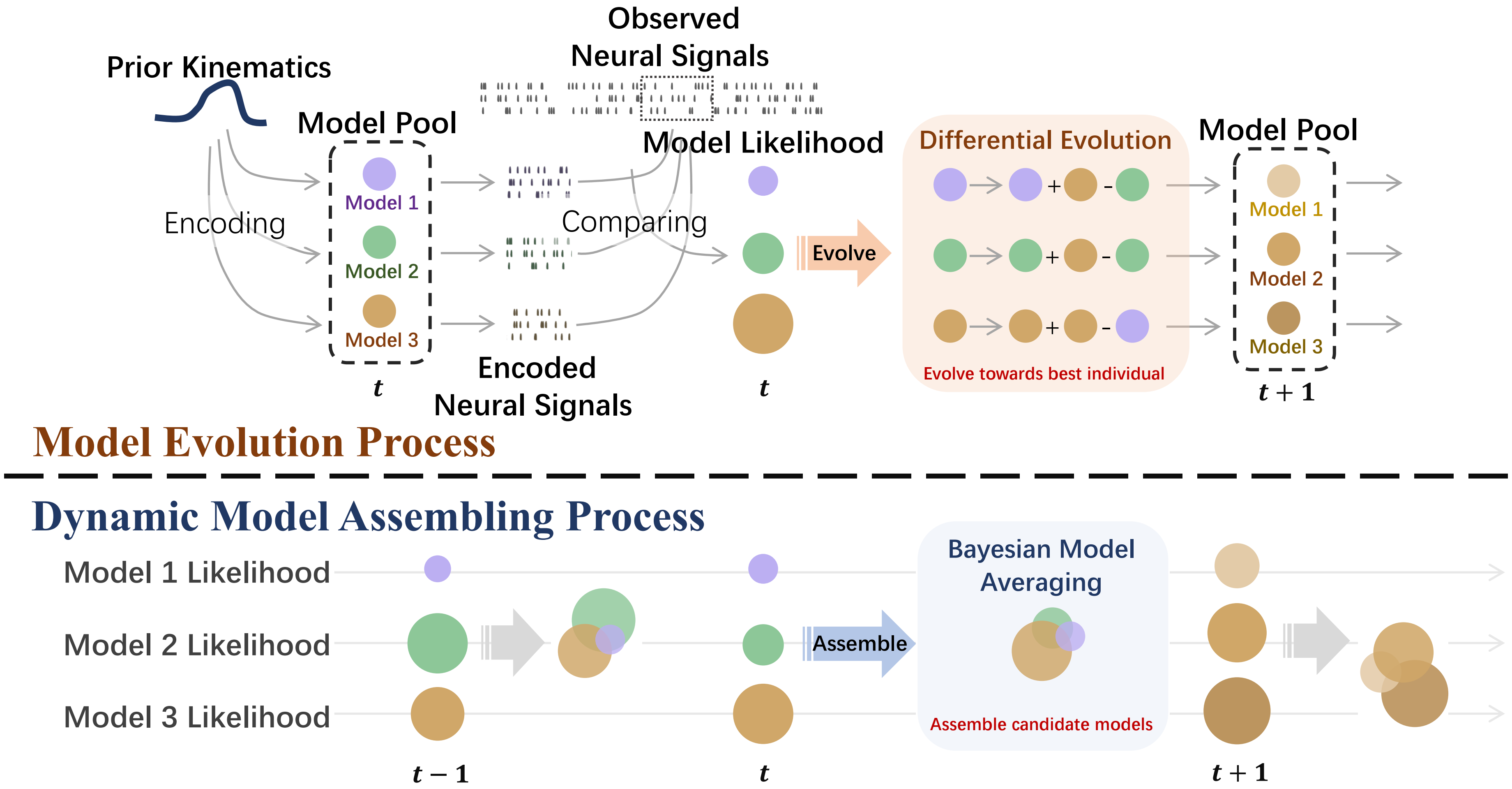
Model pool ⬇ **Population**

Model parameter ⬇ **Individual**

Likelihood ⬇ **Fitness**



EvoEnsemble : Two Model Pool Update Strategies



- **Two Strategies :**

① When to update ?

Evolve-at-changes

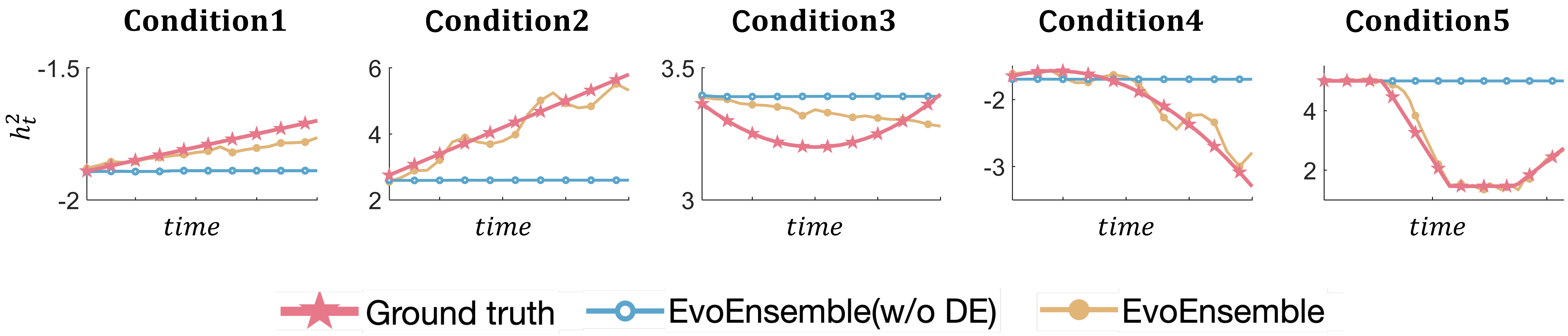
Computational efficiency

② How to update ?

History-model-archive

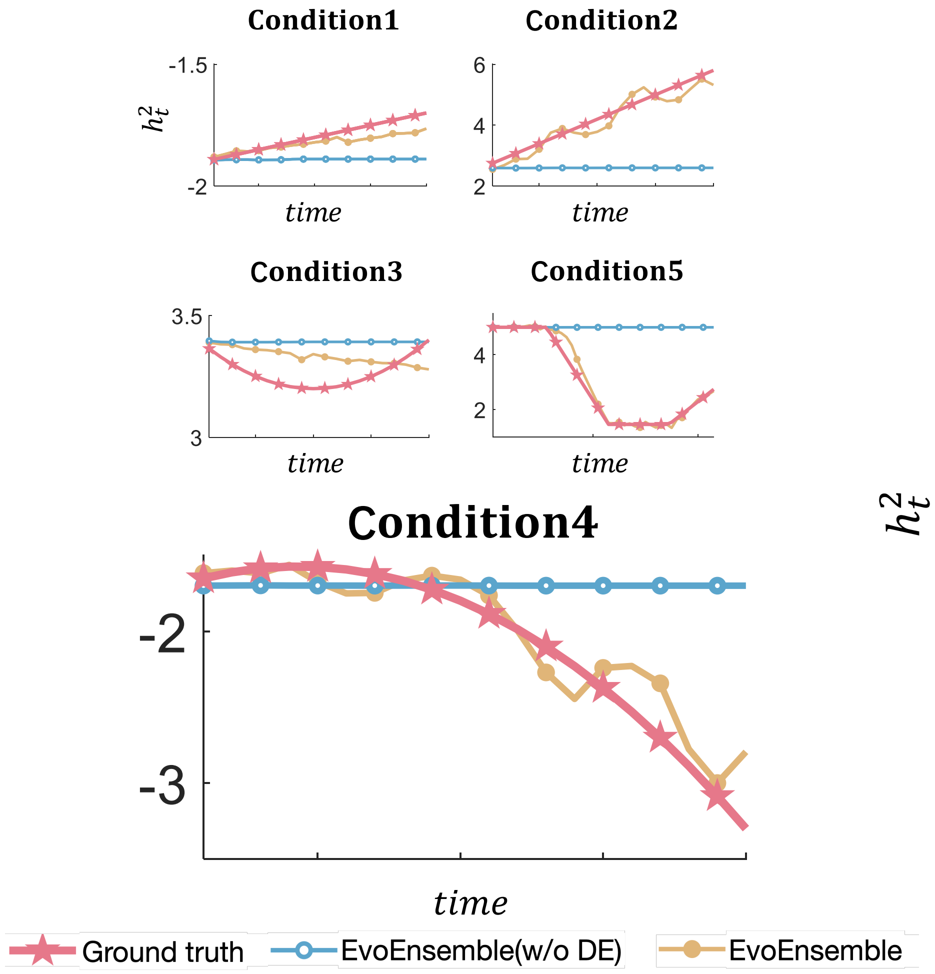
estimation stability

EvoEnsemble : Five Condition Simulation Experiments

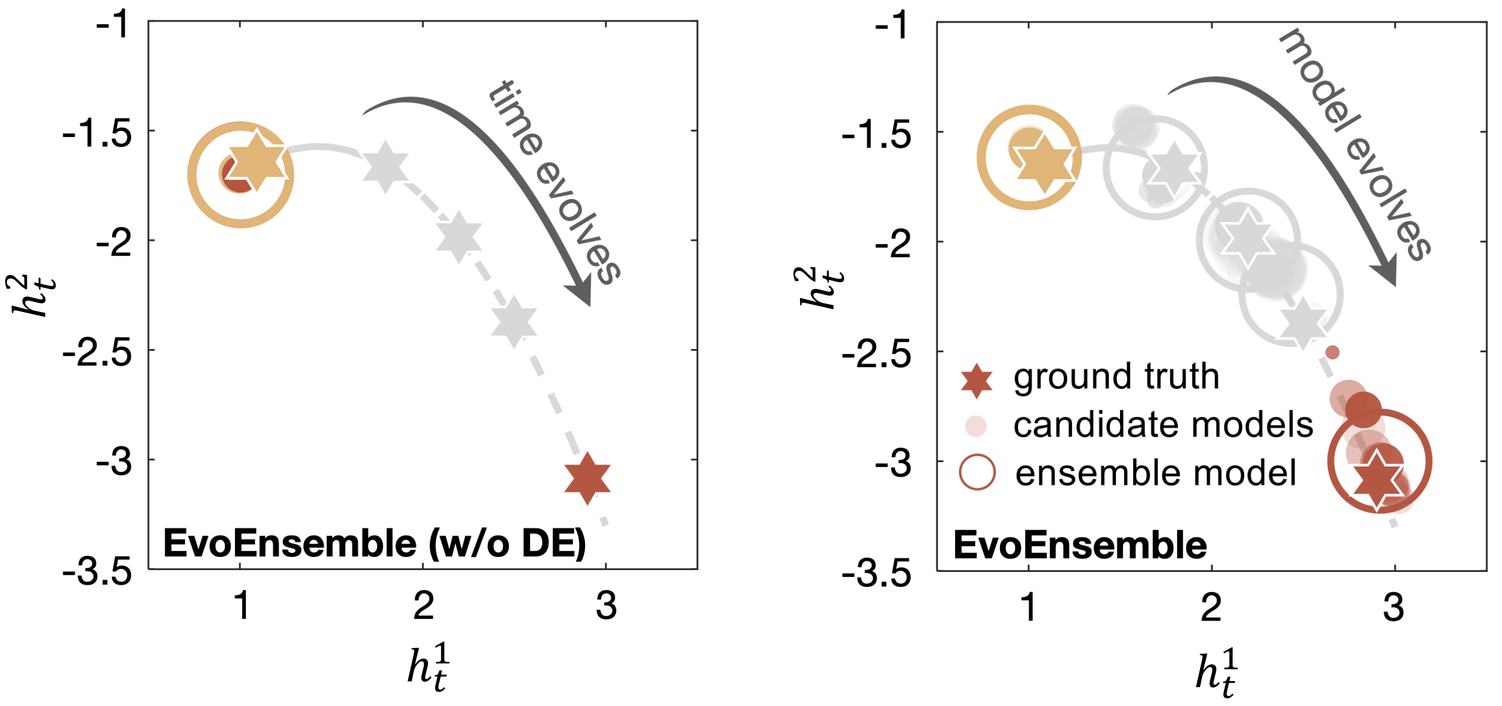


- **EvoEnsemble tracks the changes in functions closely over time for all five conditions.**

EvoEnsemble : Function Tracking Process Visualization



- **internal tracking process** of Condition 4



EvoEnsemble : Effectiveness of Two Strategies

- **Two Strategies :**

① When to update ?

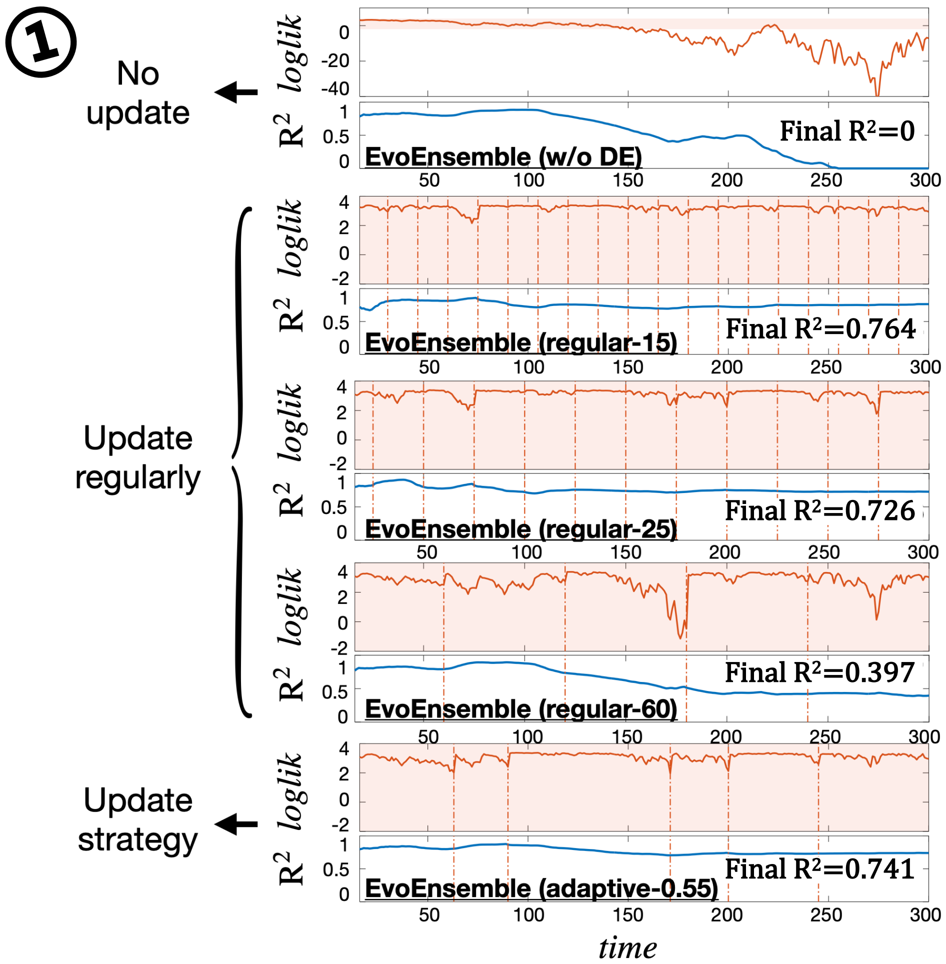
Evolve-at-changes

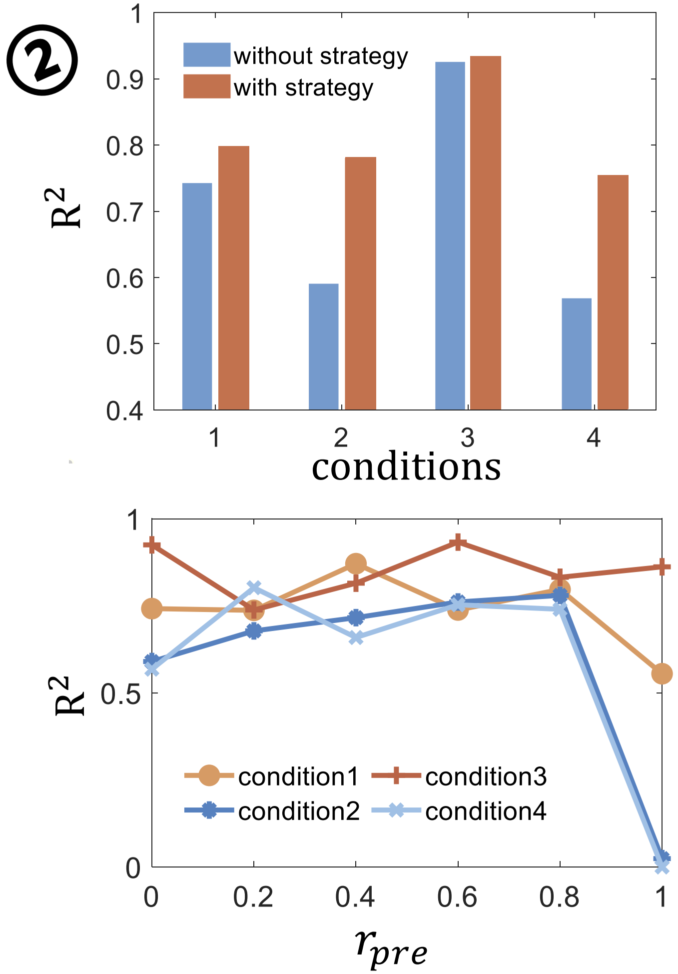
Computational efficiency

② How to update ?

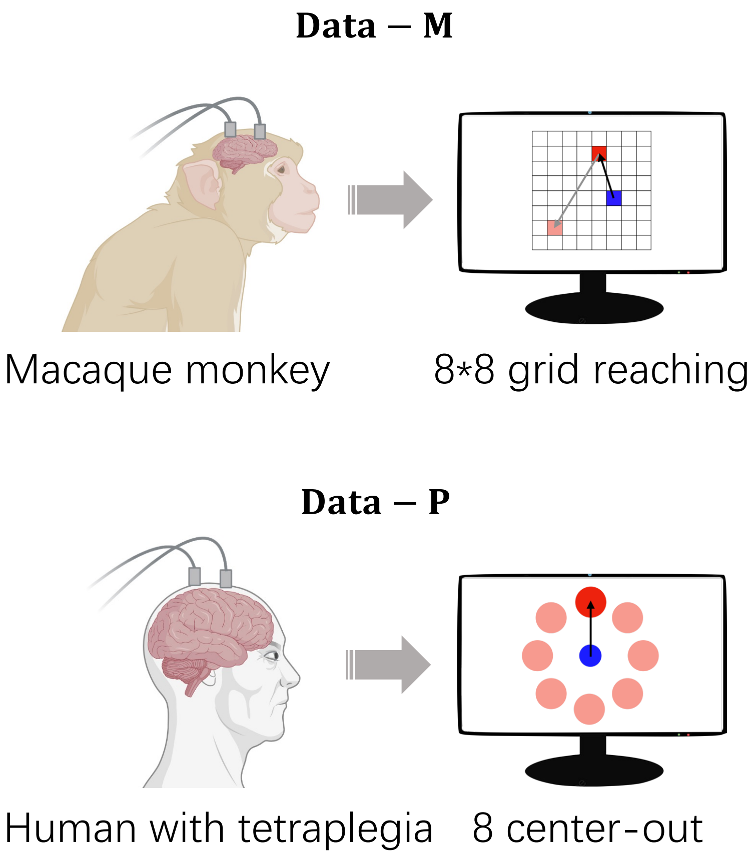
History-model-archive

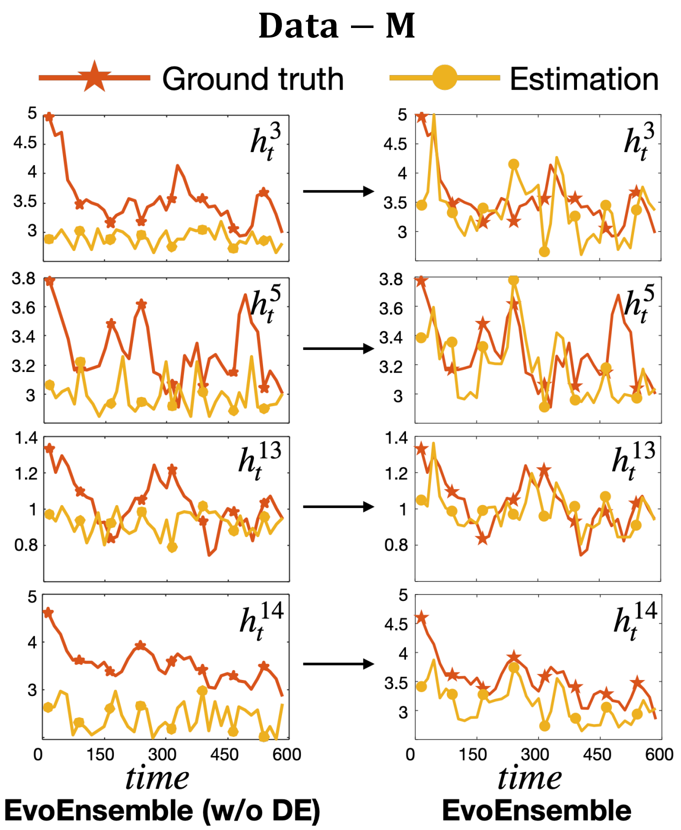
estimation stability

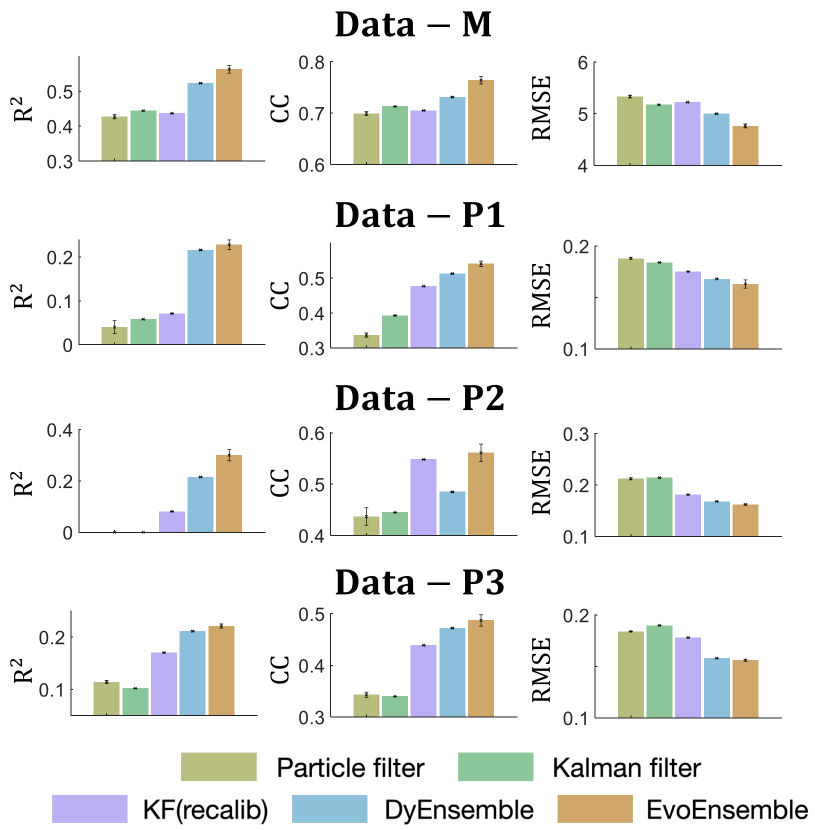




EvoEnsemble : Better Tracking and Decoding Performance







Conclusion : Dynamic Ensemble Modeling Approach

- **DyEnsemble → DyEnsemble-II → EvoEnsemble**

① Linear → Nonlinear

② Offline → Online

③ Fixed model pool → Evolvable model pool

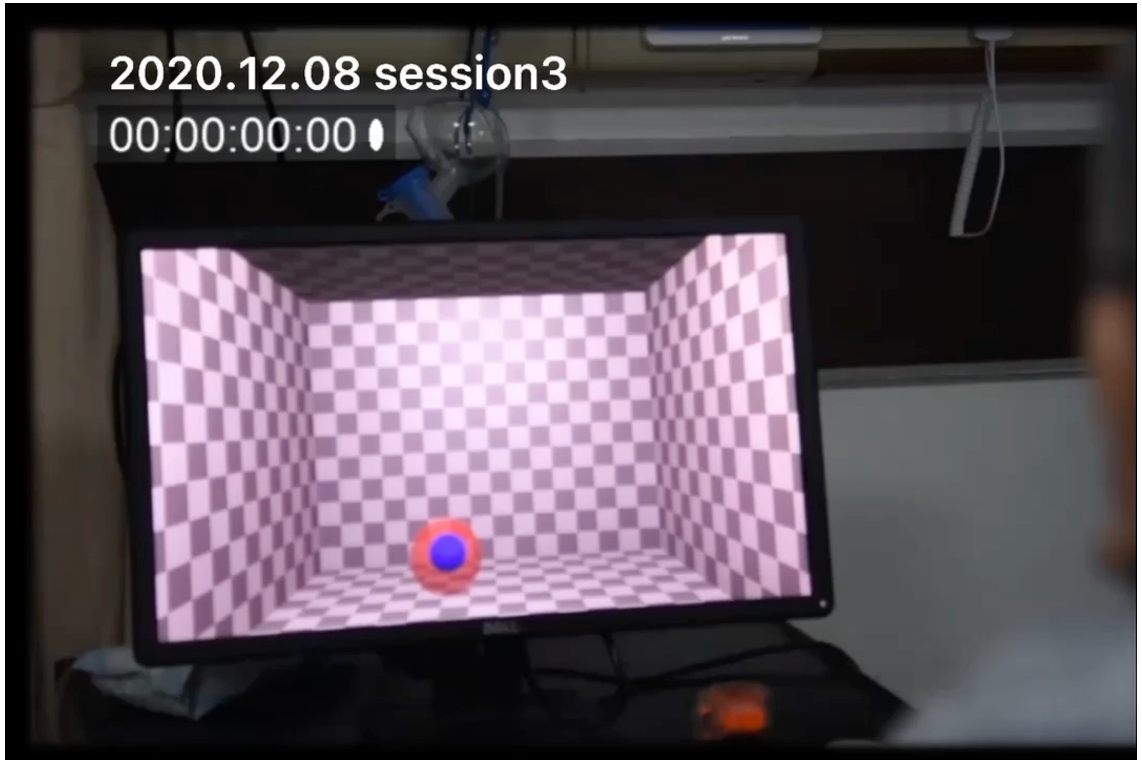
✓ provides a **flexible** adaptive neural decoding **framework**.

✓ can dynamically **change model weights** and **evolve model parameters** to cope with the nonstationary neural signals.

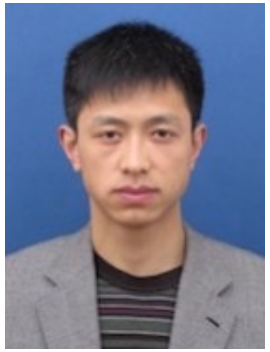
✓ achieves **higher** and **more robust** decoding performance.

Demos : Online Computer Cursor Control

- Computer cursor control



- Play Mahjong game



Yueming Wang

Yu Qi

Thanks!

Scan me to download the papers:

DyEnsemble

EvoEnsemble

Please find details in our papers:

[1] Tracking Functional Changes in Nonstationary Signals with Evolutionary Ensemble Bayesian Model for Robust Neural Decoding, ***NeurIPS*** 2022

[2] Dynamic Ensemble Bayesian Filter for Robust Control of a Human Brain-machine Interface, ***IEEE Trans. BME*** 2022