SYNMORPH: LEARNING TO BUILD TASK-ADAPTIVE SPATIOTEMPORAL MODELS FROM SYNAPTIC PRIMITIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

In real-world spatiotemporal learning applications, a single observation system often involves diverse learning tasks with heterogeneous objectives and data patterns. Researchers still need to manually design task-specific models according to data characteristics, even when tasks share the same spatiotemporal observations but require different input-output mappings, resulting in considerable engineering efforts. Existing architectures addressing task diversity mainly rely on fixed-structure adaptation, such as transfer learning, or architecture search strategies that lack structural inheritance and composition. Inspired by synaptic diversity in biological brains, we propose Synaptic Morphogenesis (SynMorph), a task-adaptive framework that learns to construct spatiotemporal models from heterogeneous synaptic primitives according to task-specific data patterns. Specifically, SynMorph first generates task encoding by jointly characterizing task types and spatiotemporal patterns. It then constructs a neuron-level meta-synaptic pool inspired by diverse synaptic mechanisms and adaptively selects synaptic primitives to compose task-specific networks through a shared Deep Q-Network (DQN) controller. The decision process is guided by network states and an overall reward function balancing model complexity and prediction performance. Finally, the composed architecture is trained end-to-end for the target task. Extensive experiments on diverse datasets involving regression and classification tasks demonstrate the effectiveness and interpretability of SynMorph, achieving up to 7.0% improvement over the strongest existing approaches on environment datasets. SynMorph provides a neuro-inspired perspective for task-adaptive spatiotemporal learning by uncovering the potential dependencies among data patterns, task requirements, and model architectures, which can be viewed as 'the model of model'.

1 INTRODUCTION

In real-world spatiotemporal learning applications, a single observation system often supports diverse learning tasks, each characterized by different input–output relationships and underlying data patterns (Yi et al., 2024; Jiang et al., 2023). Even when sharing the same spatiotemporal observations, variations in task objectives or output semantics frequently demand task-specific model designs. Consequently, existing systems rely heavily on manual, task-specific model engineering for specific prediction objectives, making it hard to reuse or transfer architectural knowledge across related tasks (Bühlmann, 2020). This limitation raises a fundamental challenge, i.e., how to construct neural architectures according to task-specific spatiotemporal patterns while enabling effective reuse and transfer of structural knowledge among related tasks. These challenges highlight the need for a more flexible and autonomous approach to model construction, where architectures can be dynamically adapted to task requirements for global optimization.

A variety of learning paradigms have been proposed to address task variations and distribution shifts in spatiotemporal learning, e.g., continual learning, transfer learning, and neural architecture search (NAS). Continual learning and transfer learning focus on preserving knowledge and transferring representations across tasks or domains, but they generally rely on predefined network architectures and achieve adaptation mainly through parameter-level optimization, limiting their ability to reorganize computational structures according to task-specific spatiotemporal patterns (Liu et al., 2025;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Zhuang et al., 2021; Wang et al., 2024). NAS aims to automate architecture design by searching for high-performing structures. Cao et al. (2026) further explored automated architecture design for time-series forecasting by searching composite architectures from predefined temporal modules and balancing prediction performance with model complexity, yet such approaches rely on predefined components and optimize architecture selection instead of learning task-conditioned structural composition. Generally, NAS primarily performs structural reselection instead of enabling structural reuse, inheritance and composition, which are essential for adapting architectures to diverse spatiotemporal tasks (Heuillet et al., 2024; Kang et al., 2023). Moreover, these approaches provide limited mechanistic interpretability regarding how architectural decisions correspond to task-specific data patterns and learning dynamics. These limitations highlight the need for an autonomous and interpretable architecture design based on task-dependent spatiotemporal characteristics.

Fortunately, cognitive neuroscience provides a novel perspective on adaptive learning. Human intelligence emerges from autonomous processes that explore and assemble combinations of synaptic components based on experience, rather than relying on fixed computational pathways (Lake et al., 2017; Hassabis et al., 2017; Kasai et al., 2021). Meanwhile, humans tend to utilize more concise and efficient methods to solve problems (Callaway et al., 2022; Correa et al., 2023). Crucially, synapses exhibit rich morphological diversity, which underlies their functional heterogeneity (Maynard et al., 2023; van Oostrum et al., 2023). Such diversity can be broadly characterized by four classes of synaptic mechanisms, i.e., spatial integration (Gidon et al., 2020), temporal dynamics (Murray et al., 2014; Zheng et al., 2024), information projection (Markov et al., 2014; Roberts et al., 2022), and modulation (Grossman & Cohen, 2022). These morphologically distinct synaptic forms enable flexible routing and transformation of information, allowing neural circuits to adapt data flow patterns to dynamic spatiotemporal data patterns (Rabinowitch et al., 2024). From a machine learning perspective, this implies that task diversity can be naturally captured by composing heterogeneous structural primitives, rather than scaling a homogeneous architecture. But how to provide a computational abstraction inspired by this human autonomous learning process that utilizes synaptic diversity to design task-adaptive neural model still remain unresolved. However, how to computationally emulate such synaptic diversity to autonomously construct task-adaptive spatiotemporal neural architectures remains unexplored. More analyses of related works can be found in Appendix A.1.

To this end, we propose Synaptic Morphogenesis (SynMorph), a flexible framework learning to build task-adaptive spatiotemporal model. Unlike conventional NAS, SynMorph learns a shared architecture-construction policy that transfers reusable structural decision knowledge across heterogeneous tasks. SynMorph first encodes task-specific information and data patterns. Inspired by synaptic morphological diversity, we then construct a neuron-level meta-synaptic pool composed of four heterogeneous synaptic primitives, which provides flexible computational structures for modeling diverse spatiotemporal data flows. To enable autonomous architecture design, we formulate synaptic composition as a reinforcement-based decision process, where a Deep Q-Network (DQN) selects appropriate synaptic units to optimize overall rewards balancing prediction performance and model complexity. Through adaptive selection and composition of meta-synaptic primitives, SynMorph learns task-adaptive architectures in an end-to-end manner so as to provide a computational abstraction inspired by the human autonomous learning process. We further validate SynMorph by designing distinct architectures for regression and classification tasks under identical spatiotemporal inputs, demonstrating its flexibility and task-adaptive capability. The contributions are as follows:

- We propose SynMorph, which learns to build task-adaptive spatiotemporal models from heterogeneous synaptic primitives through a shared DQN controller that transfers reusable architecture-selection knowledge across tasks.
- We introduce a spatiotemporal structural task encoder that captures task type, spatial topology, and temporal periodicity, together with a function-informed meta-synaptic pool composed of four categories of synaptic primitives. A reinforcement-based task-conditioned structural decision process then performs task-conditioned primitive selection and sequential composition, enabling flexible modeling of diverse spatiotemporal information.
- We extensively evaluate SynMorph on three spatiotemporal datasets including environment, traffic and brain signal. The results demonstrate its effectiveness in constructing task-specific architectures, achieving up to 7.0% improvement over the strongest existing approaches on environment datasets and revealing the potential dependencies among data, tasks and model structures, which can be viewed as 'the model of model'.

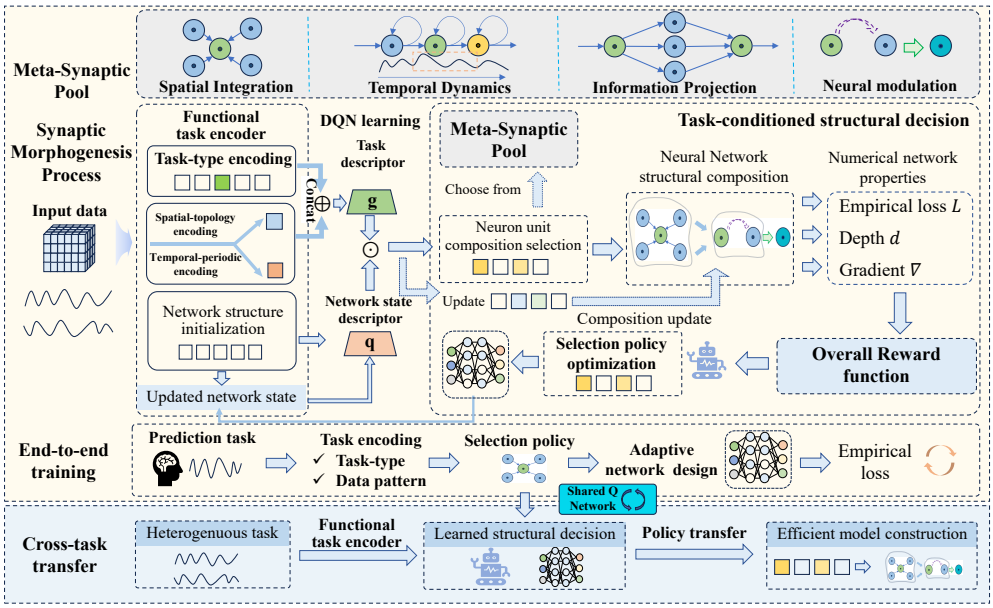


Figure 1: Framework overview of SynMorph.

2 PRELIMINARIES

2.1 TASK HETEROGENEITY UNDER SHARED INPUT SPACE

Consider a learning setting where multiple heterogeneous tasks are defined over a shared input space $\boldsymbol{X} \in \mathbb{R}^{N \times T \times D}$, where T, N, and D denote temporal length, spatial entities, and feature dimensions, respectively. Let $\mathcal{T} = \{\tau_t\}_{t=1}^{T_{task}}$ denotes the set of tasks, where each task τ_t has a task-specific output space $\boldsymbol{Y}_t \in \mathbb{R}^{d_t}$ and loss function $\mathcal{L}_t$. Despite identical inputs, different tasks may impose distinct inductive biases on representations, i.e., $d_1 \neq d_2 \neq ... \neq d_{T_{task}}$. Specifically, for a given input $x \in \mathcal{X}$, different tasks induce heterogeneous prediction targets $y_t \in \mathcal{Y}_t$ and optimization objectives $\mathbb{E}[\mathcal{L}_t(f_t(x; \boldsymbol{\theta}_t), y_t)]$.

2.2 SYNAPTIC DIVERSITY AS DEGREES OF FREEDOM

In biological neural systems, computation arises not only from neurons, but from diverse synaptic and local circuit mechanisms. Different synapse types exhibit distinct structural and dynamical properties, shaping how information is processed. We formalize synaptic diversity as a minimal set of computational degrees of freedom supporting heterogeneous task demands. Let $\mathcal{S} = \{s_k\}_{k=1}^{K}$ denote the synaptic pool where each synaptic unit s_k implements a mapping $f_k : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$. For task τ_t, we construct a network by sequentially selecting synaptic units from $\mathcal{S}$, i.e.,

$$\boldsymbol{\theta}_t = \{s_{t_0}, s_{t_1}, ..., s_{t_{N-1}}\}, s_{t_n} \in \mathcal{S} \tag{1}$$

which induces a composed hypothesis

$$f_t(x; \boldsymbol{\theta_t}) = (s_{t_{N-1}} \circ s_{t_{N-2}} \circ ... \circ s_{t_1} \circ s_{t_0})(x) \tag{2}$$

where N denotes maximum number of layers of $\boldsymbol{\theta}_t$. The formulation provides a bridge from biological synaptic diversity to task-conditioned network composition, supporting heterogeneous tasks under shared input space while grounding the design in neural computation principles.

3 METHODOLOGY

3.1 FRAMEWORK OVERVIEW

SynMorph is a task-adaptive spatiotemporal model construction framework that learns to compose task-specific architectures from heterogeneous synaptic primitives. As illustrated in Figure 1, SynMorph consists of three major components. First, a spatiotemporal structural task encoder characterizes task-specific properties, including task type, spatial topology, and temporal periodicity. Second, a function-informed meta-synaptic pool provides heterogeneous computational primitives for modeling diverse spatiotemporal dependencies. Third, a shared DQN controller performs task-conditioned primitive selection and sequential architecture composition according to the encoded task characteristics and current network states. Here, the controller is shared across heterogeneous tasks. Its architecture-selection policy learned from previous tasks is retained and progressively adapted to subsequent heterogeneous tasks, enabling structural knowledge to be transferred across tasks. Through this process, SynMorph learns not only task-specific architectures, but also a reusable model-construction policy for heterogeneous spatiotemporal learning.

3.2 TASK ENCODING AS SPATIOTEMPORAL STRUCTURAL MODULATORY SIGNAL

Inspired by top-down modulation in human brain (Miller & Cohen, 2001; Duncan, 2010), SynMorph treats task information as a functional signal to guide architecture construction. Specifically, we introduce task encoding to jointly characterize task types and the structural properties of spatiotemporal learning data, including spatial topology and temporal periodicity. These descriptors provide architecture-relevant information for subsequent synaptic selection. Formally, for each task $\tau_t \in \mathcal{T}$ with input observations $\boldsymbol{X} \in \mathbb{R}^{N\times T\times D}$, we construct the task encoding as,

$$e_t = e_t^{type} \oplus e_t^{spatial} \oplus e_t^{temporal} \tag{3}$$

where e_t^{type} denotes the one-hot encoding for τ_t (i.e., $[0, ..., 1, ..., 0]$ and 1 is on the t-th position), and $\oplus$ denotes the concatenation of vectors.

Spatial-topology encoding. To characterize spatial dependencies, we first aggregate the temporal observations as $\bar{\boldsymbol{X}} = \frac{1}{T}\sum_{i=1}^{T} \boldsymbol{X}_i$. Given an adjacency matrix $\boldsymbol{A}$, the spatial topology descriptor is obtained through lightweight topology-aware propagation as,

$$\boldsymbol{H}^{spatial} = \text{ReLU}(\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\boldsymbol{D}^{-1/2}\bar{\boldsymbol{X}}\boldsymbol{W}_s) \tag{4}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ and $\boldsymbol{D}$ is its degree matrix. Then the resulting topology signature is

$$e_t^{spatial} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{H}_i^{spatial} \tag{5}$$

For observations without explicit topology like some EEG signals, $\boldsymbol{A}$ can be estimated from pairwise correlations among spatial entities as $A_{ij} = Cov(X_i, X_j)/(\sigma_i\sigma_j + \epsilon_A)$, where σ denotes the standard deviation and ϵ_A is a small positive number to avoid denominator set zero.

Temporal-periodicity encoding. To explicitly capture periodic dynamics, we transform the temporal observations into the frequency domain as $\boldsymbol{F}(\omega) = |\text{FFT}(\boldsymbol{X})|$ where FFT denotes the Fast Fourier Transform. We then average the spectral responses over all spatial entities and feature dimensions as,

$$\bar{\boldsymbol{F}}(\omega) = \frac{1}{ND}\sum_{i=1}^{N}\sum_{j=1}^{D} \boldsymbol{F}(\omega) \tag{6}$$

where $\bar{\boldsymbol{F}}(\omega)$ represents the average spectral strength at frequency ω. Finally, the temporal-periodicity descriptor is constructed as $e_t^{temporal} = [\omega_{max}, \bar{\boldsymbol{F}}_{max}]$, where $\omega_{max} = \arg\max_{\omega\neq 0} \bar{\boldsymbol{F}}_\omega$ and $\bar{\boldsymbol{F}}_{max} = \bar{\boldsymbol{F}}(\omega_{max})$. The resulting descriptor characterizes both the dominant temporal periods and their corresponding periodic strengths, providing an explicit representation of task-specific temporal regularities.

Then each synaptic primitive s_k is conditionally chosen given the task encoding e_t and current state of the constructed network $S(\boldsymbol{\theta}_t)$. To be specific, the task-conditioned action values over candidate

synaptic primitives are represented by z_t, defined as,

$$\boldsymbol{z}_t(S(\boldsymbol{\theta}_t), e_t) = q_t(S(\boldsymbol{\theta}_t)) \odot g_t(e_t) \tag{7}$$

where $S(\boldsymbol{\theta}_t) = [t_0, t_1, ..., t_n, 0, 0, ..., 0] \in \mathbb{R}^N$ encodes current state of the constructed network $\boldsymbol{\theta}_t$ at step n, $q_t(\cdot), g_t(\cdot)$ are learnable network state and task descriptors that maps into space $\mathbb{R}^k$ respectively, and $\odot$ is the Hadamard product. Each element $\boldsymbol{z}_{t,k}$ of the selection vector $\boldsymbol{z}_t \in \mathbb{R}^k$ represents the task-conditioned Q-value for selecting primitive s_k under the current construction state. The primitive with the maximum Q-value is selected as $k^* = \arg\max_k \boldsymbol{z}_{t,k}$ and s_{k^*} is appended to the current architecture. In SynMorph, e_t serves as a conditioning signal to modulate the selection and combination process of subsequent meta-synaptic primitives. This design corresponds to the phenomenon in biological neural systems where 'the same input leads to circuit reconfiguration under different task conditions', enabling the model to construct task-specific computational structures while sharing the input space. To this end, the resulting network structure $f_t(\boldsymbol{\theta}_t)$ for task τ_t is,

$$f_t(x; \boldsymbol{\theta_t}) = (s_{t_{N-1}} \circ s_{t_{N-2}} \circ ... \circ s_{t_1} \circ s_{t_0})(x) \tag{8}$$

where $s_{t_n} \in \boldsymbol{\theta}_t$ is added based on the selection vector $\boldsymbol{z}_t$. By placing task encoding at the entry point of structural generation, spatiotemporal task heterogeneity and data patterns are explicitly translated into conditional constraints over structural composition, providing a biologically grounded foundation for synaptic diversity and subsequent dynamic structure search.

3.3 Function-informed meta-synaptic pool construction

Inspired by morphological diversity of biological synapses, we construct a function-informed meta-synaptic pool, where heterogeneous synaptic primitives provide complementary computational functions for spatiotemporal information processing (Marder, 2012; Yuste, 2015). The proposed pool consists of following four types of synaptic primitives:

(1) Spatial integration ($\mathcal{S}^{spatial}$). Spatial integration synapses capture spatial dependencies by aggregating localized or topology-aware information, including receptive-field synapses implemented by convolution operations for local feature aggregation and graph-aware synapses that incorporate adjacency information to model structured spatial interactions. Formally, let x_i denote presynaptic inputs distributed over a receptive field $\mathcal{R}_k$, a spatial integration synapse computes $s_k^{\text{spatial}}(x) = \phi(\sum_{i \in \mathcal{R}_k} w_{ki} x_i)$, encoding task-dependent aggregation locality and invariance.

(2) Temporal dynamics ($\mathcal{S}^{temporal}$). Temporal dynamics synapses model sequential evolution and temporal dependencies, including memory-based synapses for adaptive temporal integration and delay synapses that introduce configurable transmission delays to capture multi-timescale dynamics. They encode history-dependent transformations as $h^{(l)} = s_k^{\text{temporal}}(x^{(l)}, h^{(l-1)})$, where $h^{(l)}$ parameterizes intrinsic time constants on the l-th time step.

(3) Information projection ($\mathcal{S}^{proj}$). Information projection synapses regulate feature transformation during information propagation, including dimensional expansion, identity mapping, and dimensional compression units for flexible representation scaling, together with skip and attention units for efficient feature interaction. Formally, they determine inter-module routing as $x^{(l)} = s_k^{\text{proj}}(x^{(l-1)}; A_k)$ where A_k defines directed connectivity patterns across layers or regions.

(4) Neural modulation ($\mathcal{S}^{mod}$). Neural modulation synapses adjust the stability of neural pathways, including stochastic masking units for adaptive regularization and leaky integration units for information decay and plasticity enhancement. Formally, they exert influence on other units as $\widetilde{s}_{k'} = s_k^{\text{mod}}(s_{k'})$, controlling plasticity or excitability to ensure stable task-conditioned computation.

During the subsequent architecture construction process, selected synaptic units are locally parameterized through few-epoch optimization to maintain task-data adaptivity. After unit composition, the final architecture is parameterized with dynamically aligned feature dimensions, while shared architectural hyperparameters constrain the search space. By structuring the meta-synaptic pool as $\mathcal{S} = \mathcal{S}^{spatial} \cup \mathcal{S}^{temporal} \cup \mathcal{S}^{proj} \cup \mathcal{S}^{mod} \cup \textbf{STOP}$, SynMorph exposes a task-conditioned structural search space reflecting how biological systems exploit synaptic diversity to meet heterogeneous demands beyond a single universal architecture. More details can be found in Appendix A.4.

3.4 TASK-CONDITIONED STRUCTURAL DECISION

In cognitive neuroscience, task execution emerges from goal-driven selection and assembly of synaptic components rather than fixed neural architectures. This process can be viewed as constrained structural decision-making guided by task goals and feedback signals (Miller & Cohen, 2001; Duncan, 2010; Botvinick, 2012). Motivated by this principle, SynMorph formulates network construction as a task-conditioned sequential decision process, which is realized by reinforcement learning, where the core challenge is to design a reasonable reward function. Here we design the reward function based on numerical property $\mathcal{E}$ of the constructed network including training loss $\mathcal{L}_n$ and structural efficiency $\mathcal{C}(\boldsymbol{\theta}_t)$ as,

$$R_n = \mathcal{E}(\boldsymbol{\theta}_t^{(n+1)}) - \mathcal{E}(\boldsymbol{\theta}_t^{(n)}) \tag{9}$$

where (n) denotes the n-th step and $\mathcal{E}$ encodes the numerical properties of $\boldsymbol{\theta}_t^{(n)}$ as,

$$\mathcal{E}(\boldsymbol{\theta}_t^{(n)}) = \frac{\mathcal{C}(\boldsymbol{\theta}_t^{(n)})}{1 + \mathcal{L}_t^{(n)}/\sigma_{out}} \tag{10}$$

where σ_{out} denotes the standard deviation of output to normalize the task loss by the output scale. In fact, the structural efficiency is determined by the model depth as well as the gradient of model. To this end, we further define the function $\mathcal{C}(\boldsymbol{\theta}_t^{(n)})$ as,

$$\mathcal{C}(\boldsymbol{\theta}_t^{(n)}) = \eta^{d(\boldsymbol{\theta}_t^{(n)})} \lambda^{\nabla \boldsymbol{\theta}_t^{(n)}} \tag{11}$$

where $0 < \eta \leq 1, 0 < \lambda \leq 1$ are hyperparameters which control the extent of the impact of $d(\boldsymbol{\theta}_t)$ and $\nabla \boldsymbol{\theta}_t$, where $d(\boldsymbol{\theta}_t)$ denotes the depth of $\boldsymbol{\theta}_t$, $\nabla \boldsymbol{\theta}_t$ denotes the average gradient of $\boldsymbol{\theta}_t$ computed as,

$$\nabla \boldsymbol{\theta}_t = \frac{1}{d(\boldsymbol{\theta}_t)} \sum_{n=1}^{d(\boldsymbol{\theta}_t)} ||\nabla s_{t_n}||_2^2 \tag{12}$$

where $||\nabla s_{t_n}||_2$ denotes the L2-norm of the gradient of s_{t_n}. Here, since the decision is based on the relative rewards, and different gradient calculation methods like loss scaling do not affect the relative complexity of the model, they do not obviously affect the final decision. To conclude, the final reward function R_n is denoted as,

$$R_n = \frac{\eta^{d(\boldsymbol{\theta}_t^{(n+1)})} \lambda^{\nabla \boldsymbol{\theta}_t^{(n+1)}}}{1 + \mathcal{L}_t^{(n+1)}/\sigma_{out}} - \frac{\eta^{d(\boldsymbol{\theta}_t^{(n)})} \lambda^{\nabla \boldsymbol{\theta}_t^{(n)}}}{1 + \mathcal{L}_t^{(n)}/\sigma_{out}} \tag{13}$$

Algorithm 1 in Appendix summarizes the task-conditioned structural decision process in SynMorph. Consequently, the policy learns task-dependent synaptic selection strategies, resulting in different structural compositions under different tasks. Finally, the algorithm will return a well-learned model with maximum overall reward and we will train the return model end-to-end based on the given data. Consequently, our SynMorph not only allows flexible design, but captures the potential dependencies among data, task and models for design policy, which can be viewed as 'the model of model'.

4 EXPERIMENTS

4.1 DATASETS AND IMPLEMENTATION DETAILS

Our experiments cover three categorical areas including environment, traffic and brain signals, which simultaneously have both regression and classification tasks under shared input spaces, implementing the neuro-inspired architecture to facilitate task-adaptive model design. To verify the task adaptiveness, we design the learning on different tasks but with shared inputs. The datasets are as follows:

1) **Environment dataset**. We conduct experiments on **Knowair** dataset (Wang et al., 2020), which includes PM2.5 data from 2015 to 2018 for forecasting air quality.

2) **Traffic datasets**. We conduct experiments on I) **CHI** (CHICAGO, 2023), including Risk, TaxiPick and TaxiDrop collected in the second half of 2023 from Chicago. II) **SIP**, including three months of traffic data consisting of Flow and Speed collected in Suzhou Industrial Park.

Table 1: Comparison on Knowair. Best results are **bold**, second best results are underlined.

Methods	Regression			Classification	AvgRank
	MAE	RMSE	MAPE(%)	ACC(%)	
Adashare	35.02±0.32	43.91±0.36	53.23±0.51	38.99±0.34	4.0
LearnToBranch	34.69±0.29	44.17±0.60	52.09±0.64	37.60±0.23	3.5
einspace	34.95±0.43	43.80±0.46	52.04±0.53	38.17±0.24	4.0
GRU	35.33±0.38	44.83±0.46	54.25±0.43	34.02±0.32	8.5
GC-LSTM	35.27±0.29	44.62±0.49	56.97±0.39	34.85±0.36	6.5
GAGNN	39.65±0.29	65.63±0.42	65.72±0.38	30.77±0.27	11.5
PM2.5-GNN	34.92±0.27	**43.24±0.64**	**51.54±0.43**	34.17±0.32	6.0
AirFormer	36.29±0.28	50.55±0.53	57.36±0.60	34.47±0.39	8.0
AirPhyNet	36.69±0.20	57.52±0.57	61.56±0.30	33.31±0.25	10.0
DeepAir	**29.61±0.25**	47.94±0.47	52.61±0.44	35.09±0.30	3.0
MGSFormer	41.55±0.32	51.18±0.52	59.24±0.51	33.03±0.31	11.5
SynMorph	34.28±0.39	43.87±0.40	51.66±0.46	**41.70±0.19**	**1.5**

Table 2: Performance comparison on traffic datasets. Best results are **bold** and the second best results are underlined. 'R' means regression and 'C' means classification.

Methods	CHI R			CHI C	AvgRank	SIP R			SIP C	AvgRank
	MAE	RMSE	MAPE(%)	ACC(%)		MAE	RMSE	MAPE(%)	ACC(%)	
Adashare	1.123	2.051	25.13	94.93	6.0	12.805	23.759	26.90	96.03	2.5
LearnToBranch	1.122	2.048	25.17	95.88	3.5	12.991	23.653	25.87	95.78	4.0
einspace	1.123	2.050	25.20	94.70	6.5	14.007	25.134	26.14	96.19	4.5
DCRNN	1.119	**2.046**	**25.10**	91.77	5.0	**12.723**	**22.814**	25.59	92.29	5.0
AGCRN	1.127	2.051	25.83	87.04	9.0	15.783	24.724	29.38	92.69	8.5
GWNET	1.129	2.049	25.59	86.76	10.0	13.958	23.498	**25.37**	91.95	8.0
ASTGCN	**1.118**	2.058	25.09	95.33	3.0	16.511	26.121	30.77	92.79	8.5
STTN	1.125	2.050	25.21	95.71	5.5	14.646	25.057	28.45	94.70	6.5
STDN	1.121	2.047	25.18	95.49	4.0	13.689	24.063	25.91	94.67	5.5
SynMorph	1.119	2.047	25.12	**97.50**	**1.5**	12.878	22.904	25.85	**96.89**	**2.0**

3) **Brain signal datasets**. We conduct experiments on I) **HGD** (High Gamma Dataset) (Schirrmeister et al., 2017a), which is a publicly available dataset containing 128-channel EEG recordings of fourteen healthy subjects executing three different movement classes and a resting condition. II) **2A** (BCI Competition IV Dataset 2A) (Brunner et al., 2008), which comprised EEG data collected from nine individuals. The dataset consisted of four MI tasks: imagining the movements of the left hand, right hand, legs, and tongue. The implementation details can be found in Appendix A.2.

4.2 Evaluation Metrics and Baselines

We apply four metrics: Mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) for regression tasks, and accuracy (ACC) for classification tasks.

On environment data, we exploit eleven baselines, including task-adaptive model Adashare (Sun et al., 2020), LearnToBranch (Guo et al., 2020), NAS-based einspace (Ericsson et al., 2024), GRU (Dey & Salem, 2017), GC-LSTM (Zhang et al., 2019), GAGNN (Chen et al., 2023), PM2.5-GNN (Wang et al., 2020), AirFormer (Liang et al., 2023), AirPhyNet (Hettige et al., 2024), DeepAir (Yi et al., 2018), MGSFormer (Yu et al., 2025).

On traffic datasets, we exploit nine prevalent baselines for evaluations, including Adashare, LearnToBranch, einspace, DCRNN (Zhou & Yu, 2023) , AGCRN (Bai et al., 2020), GWNET (Wu et al., 2019), ASTGCN (Guo et al., 2019), STTN (Xu et al., 2020), and STDN (Cao et al., 2025).

On brain signal datasets, we exploit nine baselines, including Adashare, LearnToBranch, einspace, EEGNet (Lawhern et al., 2018), TCN (Hewage et al., 2020), CompactCNN (Izzuddin et al., 2021), Deep4Net, ShallowNet (Schirrmeister et al., 2017b) and CBraMod (Wang et al., 2025).

Table 3: Performance comparison on brain signal datasets. Best results are **bold** and the second best results are underlined. 'R' means regression and 'C' means classification.

Methods	HGD R			HGD C	AvgRank	2A R			2A C	AvgRank
	MAE	RMSE	MAPE	ACC(%)		MAE	RMSE	MAPE	ACC(%)	
Adashare	2.99	4.24	1.54	79.15	3.0	0.747	0.969	1.30	73.06	4.5
LearnToBranch	3.10	4.28	1.45	69.93	5.5	0.750	0.966	1.34	73.17	4.5
einspace	3.07	4.35	1.41	78.67	4.0	0.742	0.965	1.37	74.16	3.0
EEGNet	3.44	4.87	1.67	68.75	7.5	0.774	0.983	1.42	72.60	6.5
TCN	3.28	4.79	1.75	61.88	7.5	0.777	0.984	1.40	75.44	5.5
CompactCNN	3.92	5.29	1.68	66.25	9.5	0.776	0.984	1.46	47.33	9.0
Deep4Net	3.64	4.32	1.53	**87.62**	5.0	0.775	0.983	1.78	53.02	7.5
ShallowNet	3.53	4.61	1.61	83.21	5.5	0.778	0.986	1.63	**81.49**	5.5
CBraMod	3.31	4.34	1.52	70.06	6.0	0.769	0.972	1.37	51.38	7.0
SynMorph	**2.95**	**4.04**	**1.34**	83.27	**1.5**	**0.740**	**0.961**	**1.22**	75.09	**2.0**

Table 4: Ablation studies of SynMorph on five datasets. 'R' means regression and 'C' means classification. We report MAE results for regression tasks and ACC(%) results for classification tasks.

Methods	Knowair R	Knowair C	CHI R	CHI C	SIP R	SIP C	HGD R	HGD C	2A R	2A C
SynMorph	34.28	41.70	1.12	97.50	12.88	96.89	2.95	83.27	0.74	75.09
SynMorph-TE	35.39	34.87	1.13	91.83	13.61	92.98	3.36	76.18	0.77	70.19
SynMorph-spatial	38.93	35.06	1.18	91.59	15.73	92.44	3.26	71.32	0.78	70.02
SynMorph-temporal	40.74	34.31	1.14	91.94	15.68	93.00	3.87	64.11	0.77	61.45
SynMorph-proj	46.65	32.30	1.15	92.01	16.90	89.85	4.05	61.08	0.79	67.94
SynMorph-modu	34.78	38.58	1.13	95.89	13.65	95.97	3.16	78.39	0.76	73.46
SynMorph-depth	36.39	37.96	1.14	93.06	14.13	96.01	3.12	79.40	0.75	73.10
SynMorph-grad	37.16	40.35	1.17	94.54	13.29	94.58	3.04	80.25	0.79	71.73
SynMorph-select	50.98	27.76	1.25	85.61	18.10	85.24	4.95	59.11	0.82	58.03
SynMorph-shared	35.34	38.90	1.16	95.76	13.12	91.52	3.48	80.29	0.78	73.60
SynMorph-DQN	36.05	39.62	1.14	95.91	13.20	92.82	3.17	79.03	0.79	72.95

4.3 Performance Comparison

Comparison results are summarized in Tables 1–3. We add metrics AvgRank $= \frac{1}{T_{task}} \sum_{t=1}^{T_{task}} \text{Rank}_t$ to evaluate global performance across different tasks, where MAE on regression and Accuracy on classification tasks is ranked. SynMorph achieves competitive performance across heterogeneous regression and classification tasks. Specifically, it improves classification performance on environment datasets by 7.0%–35.5%, classification performance on traffic datasets by 0.7%–12.4%, and regression performance on brain signal datasets by 0.3%–31.5% over baselines. Considering that SynMorph aims to learn a unified architecture construction policy that adapts to different task objectives, although task-specific baselines may achieve better performance on individual objectives for which their architectures are explicitly designed, such as environmental and traffic regression or brain-signal classification, SynMorph consistently maintains competitive performance across heterogeneous tasks and achieves substantial improvements on complementary objectives and achieves the best average rank. This demonstrates that SynMorph provides stronger cross-task adaptability. Detailed robustness evaluations are provided in Appendix A.5.

4.4 Ablation Study

The ablated variants are: 1) Task encoding (**TE**); 2) Spatial integration units (**spatial**); 3) Temporal dynamic units (**temporal**); 4) Information projection units (**proj**); 5) Neural modulation units (**modu**); 6) Depth d in reward function (**depth**); 7) Gradient ∇ in reward function (**grad**); 8) Random select primitives from the meta-synaptic pool (**select**); 9) Shared controller, i.e., reinitialize controller for each task (**shared**); 10) **DQN**, i.e., fully greedy selection for maximum immediate reward. Table 4 summarizes ablation results, where removing any component degrades performance. Without TE and shared encoder, performance drops significantly, especially for classification tasks on environmental and traffic datasets and regression tasks on brain signals, highlighting the importance of task-aware signals in guiding synaptic selection. Removing spatial-integration and temporal-dynamic units impairs spatial pattern extraction and sequential dependency modeling, respectively. Large degradation occurs without information projection units and process of selection, demonstrating their importance in adapting across diverse computational processes. Removing modulation

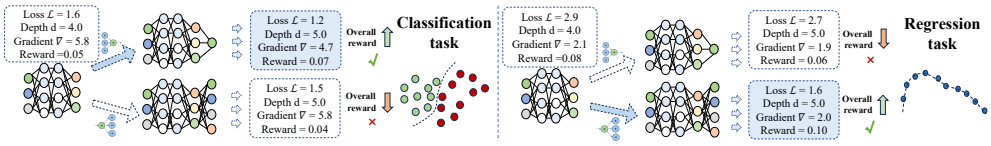


Figure 2: Different structural preference under heterogeneous tasks

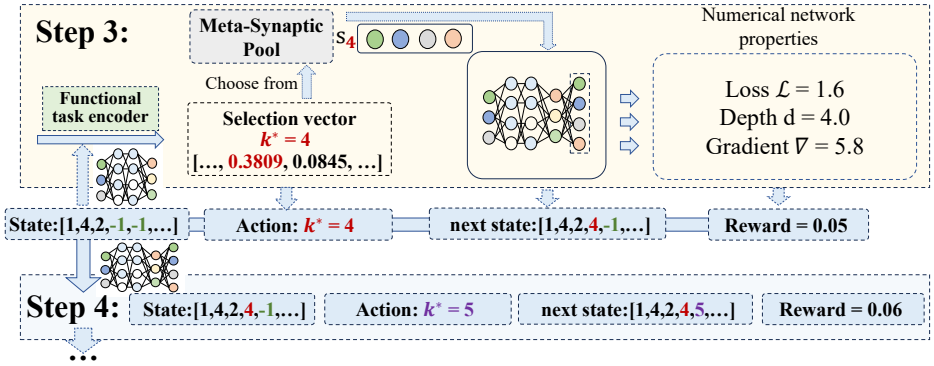


Figure 3: Influence of reward function on the selection vector

units leads to less stable training and weaker generalization, confirming their contribution to robust architecture adaptation. Removing depth and gradient factor of reward function leads to moderate performance drops, confirming their contribution to appropriate design of reward function. Finally, performance drops without DQN, verifying the importance of long-term sequential decision.

4.5 Detailed Analysis and Hyperparameter Analysis

Different structural preference under heterogeneous tasks. SynMorph exhibits task-dependent structural preferences. As shown in Figure 2, SynMorph tends to select divergent synaptic units to enrich intermediate representations for continuous function approximation under regression tasks while prefer convergent units for progressive feature compression and decision boundary learning under classification tasks. The results demonstrate that SynMorph can adaptively translate task demands into distinct structural configurations rather than relying on a fixed architecture.

The influence of the overall reward function on the selection vector. Figure 3 shows how the overall reward guides the selection vector. At step 3 of Algorithm 1, the state vector $S_3(\boldsymbol{\theta}_t^{(3)})$ preserves the state of selected synaptic units. The functional task encoder then maps $S_3(\boldsymbol{\theta}_t^{(3)})$ into a selection vector $\boldsymbol{z}_t$, which identifies the synaptic unit with the highest overall reward (index 4, reward 0.3809). Such unit of index 4 is added to the network, and its numerical properties are computed to yield the reward and next state. The functional task encoder is then updated by sampling (state, action, next state, reward) tuples via DQN. This process iterates until SynMorph selects STOP or reaches the maximum layer depth. The hyperparameter analysis can be found in Appendix A.3.

5 Conclusion

In this paper, we propose SynMorph, a neuroscience-inspired framework which learns to construct spatiotemporal models from heterogeneous synaptic primitives according to task-specific data patterns. SynMorph introduces a task encoder to characterize task-specific spatiotemporal patterns and a function-informed Meta-Synaptic Pool composed of four categories of synaptic primitives to provide flexible structural choices. A reinforcement-based DQN further learns the composition policy by balancing prediction performance and structural efficiency, enabling autonomous architecture construction under different task requirements. Extensive experiments on environmental, urban traffic, and brain signal datasets demonstrate the effectiveness and flexibility of SynMorph in addressing regression and classification tasks, achieving up to 7.0% improvement over the strongest existing approaches on environment datasets. This work provides a new perspective for neuroscience-inspired adaptive spatiotemporal learning beyond manually designed homogeneous architectures.

AI USE STATEMENT

In this work, we used generative AI tools for help develop theoretical models or conceptual frameworks, and implement methods. We have not used generative AI tools for formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, support qualitative and thematic data analysis, interpret results, and Generate synthetic data sets, clean and reformat dataset are not applicable to this work. Additionally, we used generative AI tools for summarize or analyze existing literature, create or edit software code, edit a research paper to improve readability. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature survey. LLM-generated code was verified and tested for correctness . We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work focuses on the task-adaptive model design. Inspired by theoretical neuroscience principles without collecting individual data, the data for experiments is publicly available and does not involve private information from any human, animal, or individuals, posing no ethical concerns.

REPRODUCIBILITY STATEMENT

The main implementation code can be found in `https://anonymous.4open.science/r/SynMorph-ICLR-8025/`.

REFERENCES

Sasan Salmani Pour Avval, Nathan D Eskue, Roger M Groves, and Vahid Yaghoubi. Systematic review on neural architecture search. *Artif. Intell. Rev.*, 58(3):73, 2025.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Matthew Michael Botvinick. Hierarchical reinforcement learning and decision making. *Current opinion in neurobiology*, 22(6):956–962, 2012.

Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. Bci competition 2008–graz data set a. *Institute for knowledge discovery (laboratory of brain-computer interfaces), Graz University of Technology*, 16(1-6):34, 2008.

Peter Bühlmann. Invariance, causality and robustness. *Statistical Science*, 35(3):404–426, 2020.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once for all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020.

Frederick Callaway, Bas Van Opheusden, Sayan Gul, Priyam Das, Paul M Krueger, Thomas L Griffiths, and Falk Lieder. Rational use of cognitive resources in human planning. *Nature human behaviour*, 6(8):1112–1125, 2022.

Lingxiao Cao, Bin Wang, Guiyuan Jiang, Yanwei Yu, and Junyu Dong. Spatiotemporal-aware trend-seasonality decomposition network for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11463–11471, 2025.

Qianying Cao, Shanqing Liu, Alan John Varghese, Jérôme Darbon, Michael S Triantafyllou, and George Em Karniadakis. Automatic selection of the best neural architecture for time series forecasting. *Nature Communications*, 2026.

Ling Chen, Jiahui Xu, Binqing Wu, and Jianlong Huang. Group-aware graph neural network for nationwide city air quality forecasting. *ACM Transactions on Knowledge Discovery from Data*, 18(3):1–20, 2023.

CHICAGO. Chi dataset. Website, 2023. https://data.cityofchicago.org/browse.

David G Clark and Larry F Abbott. Theory of coupled neuronal-synaptic dynamics. *Physical Review X*, 14(2):021001, 2024.

Carlos G Correa, Mark K Ho, Frederick Callaway, Nathaniel D Daw, and Thomas L Griffiths. Humans decompose tasks by trading off utility and computational cost. *PLoS computational biology*, 19(6):e1011087, 2023.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pp. 1597–1600. IEEE, 2017.

John Duncan. The multiple-demand (md) system of the primate brain: mental programs for intelligent behaviour. *Trends in cognitive sciences*, 14(4):172–179, 2010.

Linus Ericsson, Miguel Espinosa, Chenhongyi Yang, Antreas Antoniou, Amos Storkey, Shay B Cohen, Steven McDonagh, and Elliot J Crowley. einspace: Searching for neural architectures from fundamental operations. *Advances in Neural Information Processing Systems*, 37:1919–1953, 2024.

Shangce Gao, Mengchu Zhou, Yirui Wang, Jiujun Cheng, Hanaki Yachi, and Jiahai Wang. Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. *IEEE transactions on neural networks and learning systems*, 30(2):601–614, 2018.

Albert Gidon, Timothy Adam Zolnik, Pawel Fidzinski, Felix Bolduan, Athanasia Papoutsi, Panayiota Poirazi, Martin Holtkamp, Imre Vida, and Matthew Evan Larkum. Dendritic action potentials and computation in human layer 2/3 cortical neurons. *Science*, 367(6473):83–87, 2020.

Cooper D Grossman and Jeremiah Y Cohen. Neuromodulation and neurophysiology on the timescale of learning and decision-making. *Annual review of neuroscience*, 45:317–337, 2022.

Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *International conference on machine learning*, pp. 3854–3863. PMLR, 2020.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 922–929, 2019.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kethmi Hirushini Hettige, Jiahao Ji, Shili Xiang, Cheng Long, Gao Cong, and Jingyuan Wang. Airphynet: Harnessing physics-guided neural networks for air quality prediction. In *The Twelfth International Conference on Learning Representations*, 2024.

Alexandre Heuillet, Ahmad Nasser, Hichem Arioui, and Hedi Tabia. Efficient automation of neural network design: A survey on differentiable neural architecture search. *ACM Computing Surveys*, 56(11):1–36, 2024.

Pradeep Hewage, Ardhendu Behera, Marcello Trovati, Ella Pereira, Morteza Ghahremani, Francesco Palmieri, and Yonghuai Liu. Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24 (21):16453–16482, 2020.

Tarmizi Ahmad Izzuddin, Norlaili Mat Safri, and Mohd Afzan Othman. Compact convolutional neural network (cnn) based on sincnet for end-to-end motor imagery decoding and analysis. *Biocybernetics and Biomedical Engineering*, 41(4):1629–1645, 2021.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 4365–4373, 2023.

Jeon-Seong Kang, JinKyu Kang, Jung-Jun Kim, Kwang-Woo Jeon, Hyun-Joon Chung, and Byung-Hoon Park. Neural architecture search survey: A computer vision perspective. *Sensors*, 23(3): 1713, 2023.

Haruo Kasai, Noam E Ziv, Hitoshi Okazaki, Sho Yagishita, and Taro Toyoizumi. Spine dynamics in the brain, mental disorders and artificial neural networks. *Nature Reviews Neuroscience*, 22(7): 407–422, 2021.

DP Kingma. Adam: a method for stochastic optimization. In *Int Conf Learn Represent*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.

Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

Yuxuan Liang, Yutong Xia, Songyu Ke, Yiwei Wang, Qingsong Wen, Junbo Zhang, Yu Zheng, and Roger Zimmermann. Airformer: Predicting nationwide air quality in china with transformers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 14329–14337, 2023.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2018.

Jiayue Liu, Zhongchao Yi, Zhengyang Zhou, Qihe Huang, Kuo Yang, Xu Wang, and Yang Wang. Synevo: A neuro-inspired spatiotemporal evolutional framework for cross-domain adaptation. In *Forty-second International Conference on Machine Learning*, 2025.

Eve Marder. Neuromodulation of neuronal circuits: back to the future. *Neuron*, 76(1):1–11, 2012.

Nikola T Markov, Julien Vezoli, Pascal Chameau, Arnaud Falchier, René Quilodran, Cyril Huissoud, Camille Lamy, Pierre Misery, Pascale Giroud, Shimon Ullman, et al. Anatomy of hierarchy: feedforward and feedback pathways in macaque visual cortex. *Journal of comparative neurology*, 522(1):225–259, 2014.

Stephanie A Maynard, Jonas Ranft, and Antoine Triller. Quantifying postsynaptic receptor dynamics: insights into synaptic function. *Nature Reviews Neuroscience*, 24(1):4–22, 2023.

Earl K Miller and Jonathan D Cohen. An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, 24(1):167–202, 2001.

John D Murray, Alberto Bernacchia, David J Freedman, Ranulfo Romo, Jonathan D Wallis, Xinying Cai, Camillo Padoa-Schioppa, Tatiana Pasternak, Hyojung Seo, Daeyeol Lee, et al. A hierarchy of intrinsic timescales across primate cortex. *Nature neuroscience*, 17(12):1661–1663, 2014.

Michalis Pagkalos, Spyridon Chavlis, and Panayiota Poirazi. Introducing the dendrify framework for incorporating dendrites to spiking neural networks. *Nature Communications*, 14(1):131, 2023.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Mu-ming Poo, Michele Pignatelli, Tomás J Ryan, Susumu Tonegawa, Tobias Bonhoeffer, Kelsey C Martin, Andrii Rudenko, Li-Huei Tsai, Richard W Tsien, Gord Fishell, et al. What is memory? the present state of the engram. *BMC biology*, 14(1):40, 2016.

Ithai Rabinowitch, Daniel A Colón-Ramos, and Michael Krieg. Understanding neural circuit function through synaptic engineering. *Nature Reviews Neuroscience*, 25(2):131–139, 2024.

Ruairí JV Roberts, Sinziana Pop, and Lucia L Prieto-Godino. Evolution of central neural circuits: state of the art and perspectives. *Nature Reviews Neuroscience*, 23(12):725–743, 2022.

Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human Brain Mapping*, aug 2017a. ISSN 1097-0193. doi: 10.1002/hbm.23730. URL http://dx.doi.org/10.1002/hbm.23730.

Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11):5391–5420, 2017b.

Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33: 8728–8740, 2020.

Marc van Oostrum and Erin M Schuman. Understanding the molecular diversity of synapses. *Nature Reviews Neuroscience*, 26(2):65–81, 2025.

Marc van Oostrum, Thomas M Blok, Stefano L Giandomenico, Susanne Tom Dieck, Georgi Tushev, Nicole Fürst, Julian D Langer, and Erin M Schuman. The proteomic landscape of synaptic diversity across brain regions and cell types. *Cell*, 186(24):5411–5427, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Haiteng Jiang, Shijian Li, Tao Li, and Gang Pan. Cbramod: A criss-cross brain foundation model for eeg decoding. In *International conference on learning representations*, volume 2025, pp. 75310–75346, 2025.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383, 2024.

Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. Pm2.5-gnn: A domain knowledge enhanced graph neural network for pm2.5 forecasting. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '20, pp. 163–166, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380195. doi: 10.1145/3397536.3422208. URL https://doi.org/10.1145/3397536.3422208.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *International Joint Conference on Artificial Intelligence 2019*, pp. 1907–1913. Association for the Advancement of Artificial Intelligence (AAAI), 2019.

Yu Xing, Chunfang Zan, and Lu Liu. Recent advances in understanding neuronal diversity and neural circuit complexity across different brain regions using single-cell sequencing. *Frontiers in Neural Circuits*, 17:1007755, 2023.

Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. 2020.

Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.

Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. Deep distributed fusion network for air quality prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 965–973, 2018.

Zhongchao Yi, Zhengyang Zhou, Qihe Huang, Yanjiang Chen, Liheng Yu, Xu Wang, and Yang Wang. Get rid of isolation: A continuous multi-task spatio-temporal learning framework. *Advances in Neural Information Processing Systems*, 37:136701–136726, 2024.

Chengqing Yu, Fei Wang, Yilun Wang, Zezhi Shao, Tao Sun, Di Yao, and Yongjun Xu. Mgsfformer: A multi-granularity spatiotemporal fusion transformer for air quality prediction. *Information Fusion*, 113:102607, 2025.

Rafael Yuste. From the neuron doctrine to neural networks. *Nature reviews neuroscience*, 16(8): 487–497, 2015.

Han Zhang, Yonghong Song, and Yuanlin Zhang. Graph convolutional lstm model for skeleton-based action recognition. In *2019 IEEE International conference on multimedia and expo (ICME)*, pp. 412–417. IEEE, 2019.

Hanle Zheng, Zhong Zheng, Rui Hu, Bo Xiao, Yujie Wu, Fangwen Yu, Xue Liu, Guoqi Li, and Lei Deng. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15(1):277, 2024.

Jie Zhou and Qian Yu. Dcrnn: A deep cross approach based on rnn for partial parameter sharing in multi-task learning, 2023.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1): 43–76, 2021.

Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.

A Appendix

A.1 Related works

Task-adaptive architectures for spatiotemporal learning. Continual learning and transfer learning have been widely studied to address evolving tasks and distribution shifts in spatiotemporal learning. Continual learning focuses on acquiring new knowledge while mitigating catastrophic forgetting through approaches like parameter regularization, experience replay and parameter isolation (Parisi et al., 2019; De Lange et al., 2021; Wang et al., 2024). However, these methods mainly adapt fixed architectures through parameter updates or representation reuse, limiting structural reorganization for task-specific spatiotemporal patterns. Transfer learning reuse pretrained representations to transfer knowledge across tasks or domains, while maintaining largely invariant model structures (Zhuang et al., 2021; Wang et al., 2024; Liu et al., 2025). Such strategies are effective for shared representations but lack the flexibility to adapt architectures to heterogeneous spatiotemporal dependencies. Neural Architecture Search (NAS) further attempts to automate architecture design by searching for high-performing structures under predefined objectives. Representative approaches, including reinforcement learning-based NAS, ENAS, and DARTS, optimize architectures within predefined search spaces (Zoph & Le, 2017; Liu et al., 2018; Cai et al., 2020; Avval et al., 2025). Cao et al. proposed a multi-objective framework that automatically searches composite forecasting architectures from predefined temporal modules, achieving a balance between prediction performance and model complexity (Cao et al., 2026). However, existing NAS methods mainly optimize static architectures for specific objectives, lacking task-aware structural reuse and adaptive composition according to diverse spatiotemporal patterns. Overall, existing approaches mainly adapt tasks within fixed structural templates, motivating learning frameworks that can autonomously construct task-specific architectures according to spatiotemporal patterns.

Synaptic diversity inspired neural architecture design. Biological neural systems exhibit remarkable flexibility in processing diverse spatiotemporal signals and supporting complex behaviors, widely attributed to the heterogeneity of synaptic and circuit-level computations rather than a uniform processing mechanism (Poo et al., 2016; van Oostrum & Schuman, 2025). Neuroscientific studies have shown that synapses exhibit systematic diversity in their computational functions, including local receptive-field integration, divergent and convergent connectivity patterns, multi-timescale temporal dynamics, and topology-constrained information propagation (Pagkalos et al., 2023; Clark & Abbott, 2024; Xing et al., 2023). Such heterogeneous mechanisms allow neural circuits to flexibly organize information pathways and adapt computation according to task demands. Inspired by these biological principles, several machine learning studies have attempted to incorporate neural computation mechanisms into artificial architectures. Conventional neural architectures exploit individual computational motifs, such as local receptive fields in convolutional networks (Krizhevsky et al., 2012), residual information propagation through skip connections (He et al., 2016), attention-based modulation mechanisms (Vaswani et al., 2017), and graph-structured information propagation (Yamins & DiCarlo, 2016). Meanwhile, dendritic neural networks explicitly model dendritic computation within artificial neurons, demonstrating that individual neurons can perform nonlinear feature integration beyond conventional weighted summation (Gao et al., 2018). These studies demonstrate the potential of biologically inspired structures for enhancing neural computation. However, existing approaches mainly improve individual modules, while the selection and composition of heterogeneous primitives remain manually predefined rather than adaptively optimized for task-specific requirements. In contrast, SynMorph treats synaptic diversity as a principle for task-adaptive spatiotemporal architecture construction. It explicitly models heterogeneous synaptic primitives and enables their task-conditioned composition, transforming structural diversity from a fixed design choice into a learnable mechanism.

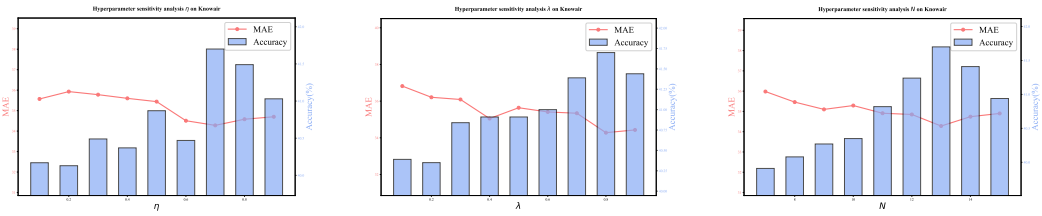
A.2 Implementation Experimental Details

We conduct experiments on regression and classification tasks, i.e., τ_1 = Regression and τ_2 = Classification. The average results of five independent experiments are reported. Standard deviation over multiple runs on traffic and brain signal datasets is reported in Table 7-8. All relevant experiments have passed the hypothetical tests with p¡0.05. On dataset Knowair, we conduct experiments on Sub-dataset 3. To evaluate classification task, $PM_{2.5}$ data is split into 8 classes, whose standard corresponds to *HJ 633—2012 Technical Regulation on Ambient Air Quality Index (on trial)* [1] of People's Republic of China. On traffic datasets, data is split into training, validation and testing sets with the ratio of 7:1:2. To evaluate classification task, the Risk data of CHI and Flow data of SIP are split into 3 classes, i.e. high (top 50%), middle (25%-50%) and low (bottom 25%). On brain signal datasets, we conduct experiments on subject 1. We split the datasets into training, validation and testing sets with the ratio of 6:2:2. To evaluate regression task, we utilize the former 100 time steps to forecast the future 50 time steps. We apply CrossEntropyLoss for classification tasks, while MSELoss for regression tasks. We apply over-sampling to avoid imbalance among different classes. The optimizer is Adam (Kingma, 2014), the learning rate is 5×10^{-3} and the weight decay is 5×10^{-4}. For fair comparison, all methods are evaluated under the same data splits, training protocol, and evaluation metrics. As detailed in Algorithm 1 Line 16, we reduce extra costs via proxy training, i.e., evaluating candidates with $<$6 epochs (particularly, 3) and small batches. Empirically, SynMorph increases training time by only about 21% while reduces space complexity by about 30%, fully justifying the cost for automated, task-adaptive architecture design. Within each shared-input system, a single DQN controller is reused across heterogeneous tasks. For DQN, its Q-network is a fully connected network with the hidden states 32, 64 and ReLU activations in the hidden layers. The online Q-network is optimized using Adam, while the target network is synchronized every five episodes. The discount factor is set to 0.9. Experience replay uses a buffer capacity of 5,000 transitions and a mini-batch size of 32. An epsilon-greedy policy is adopted, with an initial epsilon of 0.95, a minimum epsilon of 0.1, and a decay factor of 0.995; epsilon is decayed only after a successful episode. The controller is trained for 500 episodes. We conduct experiments on Tesla V100-PCIE-16GB.

[1] https://www.mee.gov.cn/ywgz/fgbz/bz/bzwb/jcffbz/201203/W020250407406362045613.pdf

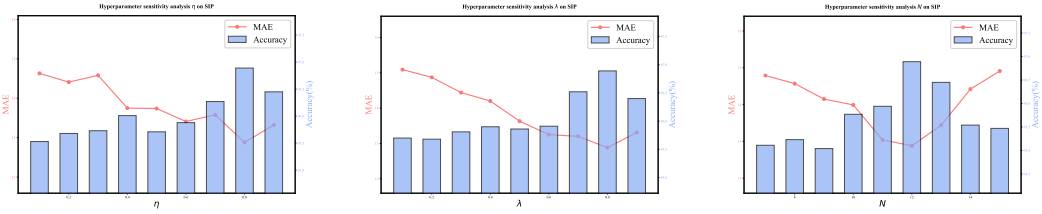
A.3 HYPERPARAMETER SENSITIVITY ANALYSIS

We varied parameter η from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, λ from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and N from $\{7, 8, 9, 10, 11, 12, 13, 14, 15\}$ on Knowair, SIP and HGD. Results shown in Figure 4 to Figure 6 indicate that the optimal settings are $\eta = 0.7$, $\lambda = 0.8$ and $N = 13$ on Knowair, $\eta = 0.8$, $\lambda = 0.8$ and $N = 12$ on SIP, and $\eta = 0.6$, $\lambda = 0.7$ and $N = 11$ on HGD. When hyperparameters are small, the architecture will focus too much on model complexity, thus sacrificing the performance. When hyperparameters exceed the optimal value, the architecture will be too complex so as to result in over-fitting on test sets. Consequently, the balance between the model complexity and training performance should be taken into consideration.



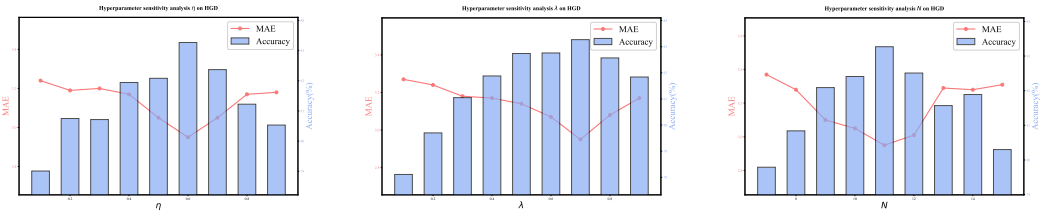
((a)) η sensitivity on Knowair ((b)) λ sensitivity on Knowair ((c)) N sensitivity on Knowair

Figure 4: Hyper sensitivity analysis on Knowair



((a)) η sensitivity on SIP ((b)) λ sensitivity on SIP ((c)) N sensitivity on SIP

Figure 5: Hyper sensitivity analysis on SIP



((a)) η sensitivity on HGD ((b)) λ sensitivity on HGD ((c)) N sensitivity on HGD

Figure 6: Hyper sensitivity analysis on HGD

A.4 IMPLEMENTATION DETAILS OF META-SYNAPTIC POOL

A.4.1 DEFINITION OF META-SYNAPTIC POOL

Inspired by the morphological and functional diversity of biological synapses, SynMorph constructs a Meta-Synaptic Pool consisting of heterogeneous computational primitives. Different from conventional neural architectures that stack homogeneous layers or manually designed modules, each

meta-synaptic unit represents a minimal functional transformation mechanism responsible for a specific information processing pattern. Formally, given an input representation,

$$\boldsymbol{X} \in \mathbb{R}^{T \times N \times D} \tag{14}$$

where T, N, D denotes the temporal length, spatial dimension, and feature dimension respectively, the meta-synaptic pool is defined as,

$$\mathcal{S} = \{\mathcal{S}_{sp}, \mathcal{S}_{tm}, \mathcal{S}_{pr}, \mathcal{S}_{md}\} \tag{15}$$

where $\mathcal{S}_{sp}, \mathcal{S}_{tm}, \mathcal{S}_{pr}, \mathcal{S}_{md}$ represents spatial integration, temporal dynamics, information projection, and neural modulation synapses, respectively. Each synaptic group contains multiple functional units, i.e.,

$$\mathcal{S}_i = \{s_i^1, s_i^2, ..., s_i^{M_i}\} \tag{16}$$

where $i \in \{sp, tm, pr, md\}$, and M_i represents the number of candidate units within each functional category. Each synaptic unit $s_k \in \mathcal{S}$ performs a parameterized transformation,

$$\boldsymbol{Y} = s_k(\boldsymbol{X}; \boldsymbol{\theta}_k) \tag{17}$$

where $\boldsymbol{\theta}_k$ denotes learnable parameters. According to their functional roles, different synaptic units operate on different aspects of information flow, i.e.,

$$s_k : \underbrace{\boldsymbol{X}}_{\text{input representation}} \rightarrow \underbrace{\boldsymbol{Y}}_{\text{transformed representation}} \tag{18}$$

where spatial integration units determine where information is aggregated, temporal dynamic units determine how information evolves over time, projection units determine how representations are transformed and propagated, and modulation units determine how information pathways are regulated and stabilized. During architecture construction, the reinforcement-based controller selects and composes different synaptic units to form task-adaptive neural architectures. Therefore, meta-synaptic units serve as reusable functional primitives rather than complete neural architectures, enabling SynMorph to construct diverse task-specific models through compositional structure learning. Detailed information of the four types of synapses are as follows.

A.4.2 Spatial Integration Synapses

Spatial Integration synapses aim to capture spatial dependencies by aggregating information from different locations or structural neighborhoods. Inspired by dendritic local integration mechanisms, these units provide different strategies for spatial information aggregation.

(1) Receptive-field Synapse

The receptive-field synapse models localized information integration through learnable spatial kernels as,

$$\boldsymbol{Y} = \sigma(\boldsymbol{W} * \boldsymbol{X}) \tag{19}$$

where $*$ denotes spatial convolution and σ is the nonlinear activation function. This unit enables the model to extract local spatial patterns by defining adaptive receptive fields, analogous to how biological neurons integrate signals from spatially distributed synaptic inputs.

(2) Topology-aware Synapse

For graph-structured spatiotemporal systems, spatial dependencies are not solely determined by physical proximity but also by underlying topology. Therefore, the topology-aware synapse incorporates structural relationships as,

$$\boldsymbol{Y} = f(\boldsymbol{A}, \boldsymbol{X}) \tag{20}$$

where $\boldsymbol{A}$ denotes the adjacency structure. This unit allows information propagation along topology-constrained pathways, mimicking connectivity-dependent information integration in neural circuits.

A.4.3 Temporal Dynamics Synapses

Temporal Dynamics synapses model the evolution of sequential patterns and capture dependencies across multiple temporal scales. Inspired by dynamic synaptic transmission and temporal filtering mechanisms, these units provide flexible temporal information processing.

(1) Dynamic Integration Synapse

The dynamic integration synapse performs adaptive temporal aggregation as,

$$\boldsymbol{Y} = f(\boldsymbol{X}_{1:T}; \boldsymbol{\theta}) \tag{21}$$

where temporal parameters determine the integration of historical information. This unit captures intrinsic temporal dependencies and enables the model to adaptively learn different temporal dynamics from various tasks.

(2) Delay Synapse

Biological neural communication involves non-negligible transmission latency. Motivated by synaptic delay mechanisms, the delay synapse introduces configurable temporal shifts as,

$$\boldsymbol{Y}_t = \boldsymbol{X}_{t-\tau} \tag{22}$$

where τ represents the transmission delay. This unit enables the architecture to model delayed effects and multi-timescale temporal interactions.

A.4.4 Information Projection Synapses

Information Projection synapses regulate feature transformation and information flow throughout the network. Different from conventional fixed-dimensional layers, these units provide flexible mechanisms for expanding, preserving, compressing, and selectively routing representations.

(1) Divergent Projection Synapse

The divergent projection expands the representation space as,

$$\boldsymbol{Y} = \boldsymbol{W}_e \boldsymbol{X}, \quad D_{out} > D_{in} \tag{23}$$

It enables richer feature representations by increasing the dimensional capacity of intermediate representations.

(2) Identity Projection Synapse The identity projection maintains the original feature dimension as,

$$\boldsymbol{Y} = \boldsymbol{W}_i \boldsymbol{X}, \quad D_{out} = D_{in} \tag{24}$$

It preserves information flow without additional transformation and provides a direct pathway for stable information propagation.

(3) Convergent Projection Synapse

The convergent projection compresses feature representations as,

$$\boldsymbol{Y} = \boldsymbol{W}_c \boldsymbol{X}, \quad D_{out} < D_{in} \tag{25}$$

It extracts task-relevant information by reducing redundant dimensions.

(4) Skip Projection Synapse

The skip projection enables shortcut information propagation as,

$$\boldsymbol{Y} = \boldsymbol{X} + F(\boldsymbol{X}) \tag{26}$$

This mechanism facilitates efficient gradient propagation and long-range information transmission.

(5) Attention Projection Synapse

The attention projection performs adaptive feature interaction. Given input $\boldsymbol{X}$, it computes

$$\boldsymbol{Q} = \boldsymbol{W}_Q \boldsymbol{X}, \boldsymbol{K} = \boldsymbol{W}_K \boldsymbol{X}, \boldsymbol{V} = \boldsymbol{W}_V \boldsymbol{X} \tag{27}$$

and aggregates information through

$$\boldsymbol{Y} = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V} \tag{28}$$

By dynamically weighting feature dependencies, this synapse enables selective information propagation and context-aware representation transformation.

A.4.5 NEURAL MODULATION SYNAPSES

Neural Modulation synapses regulate the activity and stability of other computational pathways. Inspired by biological neuromodulation mechanisms, these units control information retention and pathway activation.

(1) Stochastic Modulation Synapse

The stochastic modulation synapse introduces adaptive pathway suppression as,

$$\boldsymbol{Y} = \boldsymbol{M} \odot \boldsymbol{X} \tag{29}$$

where $\boldsymbol{M}$ is a stochastic mask. By selectively reducing redundant information flow, this unit improves model robustness and prevents overfitting.

(2) Leaky Integration Synapse

The leaky integration synapse controls information persistence as,

$$\boldsymbol{Y}_t = \alpha \boldsymbol{Y}_{t-1} + \boldsymbol{X}_t \tag{30}$$

where α controls the retention ratio of previous states. This mechanism provides adaptive memory regulation and improves model plasticity under changing task conditions.

A.4.6 SUMMARY OF META-SYNAPTIC UNITS

Table 5 summarizes all meta-synaptic units. Each unit corresponds to a distinct computational function rather than a predefined neural architecture. By composing these heterogeneous primitives, SynMorph transforms structural diversity from a manually designed architectural property into a task-conditioned and learnable mechanism.

A.4.7 INTERPRETATION OF TASK-CONDITIONED SYNAPTIC COMPOSITION

We further visualize the selection proportions of different synaptic units in Fig. 7 to interpret the learned architecture construction policy. SynMorph exhibits modality-dependent structural preferences. Temporal dynamic units are predominantly selected for EEG signals to capture intrinsic neural temporal dynamics, whereas spatial integration units are more frequently activated for city traffic data to characterize spatial dependencies and network topology. Such functional alignment indicates that the learned synaptic composition is not an arbitrary search process, but an interpretable adaptation mechanism that maps task-specific spatiotemporal characteristics to appropriate computational structures.

A.5 ROBUSTNESS OF SYNMORPH UNDER DIFFERENT TRAINING CONFIGURATIONS

To further evaluate the robustness of SynMorph under different training configurations, we conduct additional experiments by varying training epochs, random seeds, optimizers, learning rate schedulers, loss scaling strategies, and task order on SIP dataset. Table 6 summarizes the empirical results, demonstrating that SynMorph maintains consistent performance across different training settings.

Table 5: Detailed specification of meta-synaptic units in SynMorph. Each unit represents a functional computational primitive inspired by heterogeneous synaptic mechanisms.

Category	Synaptic Unit	Formulation	Functional Role	Biological Inspiration
Spatial Integration	Receptive-field	$\mathbf{Y} = \sigma(\mathbf{W} * \mathbf{X})$	Local spatial aggregation through learnable receptive fields	Dendritic local integration of spatially distributed inputs
	Topology-aware	$\mathbf{Y} = f(\mathbf{A}, \mathbf{X})$	Topology-guided information propagation for structured data	Connectivity-dependent spatial integration in neural circuits
Temporal Dynamics	Dynamic Integration	$\mathbf{Y} = f(\mathbf{X}_{1:T}; \theta)$	Adaptive temporal integration for sequential dependencies	Multi-timescale synaptic temporal dynamics
	Delay	$\mathbf{Y}_t = \mathbf{X}_{t-\tau}$	Temporal latency modeling through configurable delays	Synaptic transmission delay mechanisms
Information Projection	Divergent Projection	$\mathbf{Y} = W_e\mathbf{X}, D_{out} > D_{in}$	Expands representation space for richer feature encoding	Divergent neural projections for distributed representations
	Identity Projection	$\mathbf{Y} = \mathbf{X}$	Preserves information flow without transformation	Stable neural communication pathways
	Convergent Projection	$\mathbf{Y} = W_c\mathbf{X}, D_{out} < D_{in}$	Compresses representations and extracts task-relevant features	Convergent integration of multiple neural inputs
	Skip Projection	$\mathbf{Y} = \mathbf{X} + F(\mathbf{X})$	Enhances long-range information propagation	Shortcut communication pathways in neural circuits
	Attention Projection	$\mathbf{Y} = \mathrm{softmax}(\frac{\mathbf{QK}^T}{\sqrt{d_k}})\mathbf{V}$	Adaptive feature interaction and selective information routing	Context-dependent selective communication
Neural Modulation	Stochastic Modulation	$\mathbf{Y} = \mathbf{M} \odot \mathbf{X}$	Adaptive pathway regulation and robustness enhancement	Stochastic synaptic regulation
	Leaky Integration	$\mathbf{Y}_t = \alpha\mathbf{Y}_{t-1} + \mathbf{X}_t$	Controls information retention and temporal plasticity	Persistent neural activity and synaptic plasticity

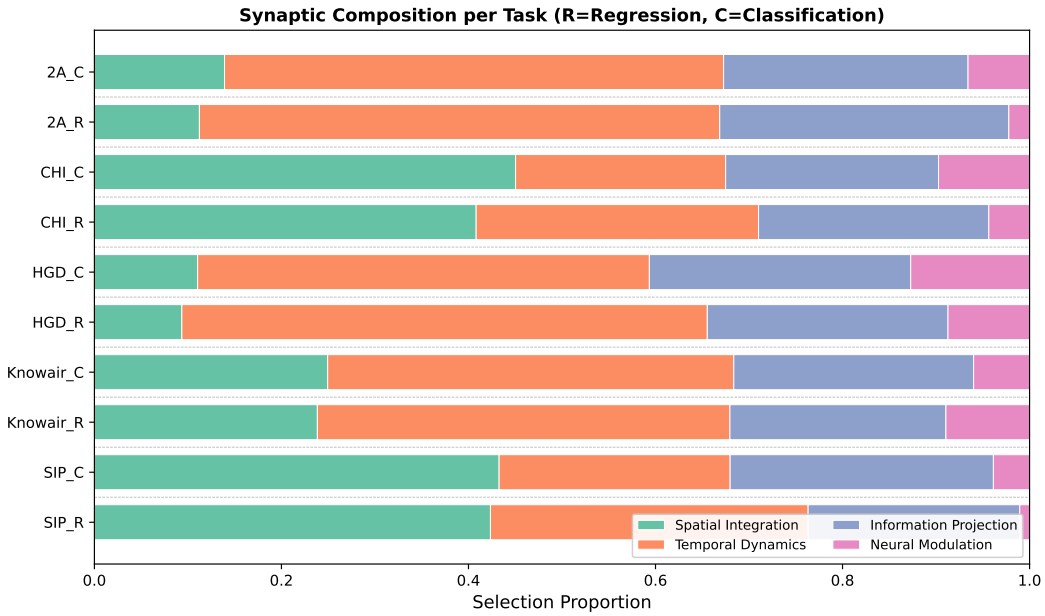


Figure 7: Task-dependent synaptic selection patterns learned by SynMorph

Algorithm 1 Task-conditioned structural decision

Require: Meta-synaptic pool $\mathcal{S} = \{s_1, s_2, \ldots, s_K\}$, task set $\mathcal{T}$, maximum layer depth N, maximum episode E

Initialize Task Encoder
Initialize experience pool P
Initialize shared controller $Q(\boldsymbol{\Theta}_c)$
for $t = 1$ to T_{task} **do**
  for $ep = 1$ to E **do**
    Initialize empty structure $\boldsymbol{\theta}_t^{(0)} \leftarrow \emptyset$
    Initialize $\mathcal{E}(\boldsymbol{\theta}_t^{(0)})$ as 0
    Initialize state $S_0(\boldsymbol{\theta}_t^{(0)})$
    for $n = 0$ to $N - 1$ **do**
      Sample action $k^* = \arg\max_k \boldsymbol{z}_t(S_n, e_t)$
      if k^* is **STOP then**
        break
      end if
      Append selected synapse s_{k*} to structure:

$$\boldsymbol{\theta}_t^{(n+1)} \leftarrow s_{k*} \circ \boldsymbol{\theta}_t^{(n)}$$

      Update state $S_{n+1}(\boldsymbol{\theta}_t^{(n+1)})$
      Train $\boldsymbol{\theta}_t$ for few epochs on small batches of τ_t
      Compute $\mathcal{L}_t^{(n+1)}$, $\nabla\boldsymbol{\theta}_t^{(n+1)}$ and $d(\boldsymbol{\theta}_t^{(n+1)})$ to get $\mathcal{E}(\boldsymbol{\theta}_t^{(n+1)})$
      Compute overall reward R_n according to Eq. 13.
      Store the tuple (S_n, k^*, R_n, S_{n+1}) into P
      Update the shared DQN controller $Q(\boldsymbol{\Theta}_c)$ and task-conditioned descriptors using sampled transitions from P
    end for
  end for
end for

Table 6: More empirical performance comparison on SIP dataset. MAE denotes the metrics for regression tasks and ACC denotes the accuracy metrics for classification tasks.

Configuration	**Concrete value**	MAE	ACC
Epochs	2	13.019	96.36
	3	12.878	96.89
	4	12.905	96.87
	5	12.860	96.90
Random seeds	1	13.013	96.74
	42	12.895	96.81
	1234	12.915	96.83
	2026	12.878	96.89
Optimizer	SGD	12.908	96.81
	Adagrad	13.006	96.84
	Adam	12.878	96.87
Scheduler	Exponential Decay	12.902	96.79
	MultiStep LR	12.878	96.87
	Constant LR	12.916	96.83
Loss scale	0.8	12.929	96.80
	1	12.878	96.87
	1.2	12.994	96.85
Task order	Classification→Regression	12.891	96.45
	Regression→Classification	12.878	96.87

Table 7: Standard deviation over 5 independent experiments on traffic datasets. 'R' means regression and 'C' means classification.

Methods	CHI R			CHI C	SIP R			SIP C
	MAE	**RMSE**	**MAPE(%)**	**ACC(%)**	**MAE**	**RMSE**	**MAPE(%)**	**ACC(%)**
Adashare	0.08	0.25	0.31	0.90	1.16	1.56	2.15	1.40
LearnToBranch	0.11	0.19	0.37	0.89	1.00	1.84	1.93	3.33
einspace	0.11	0.20	0.13	0.92	1.16	1.93	2.56	2.97
DCRNN	0.10	0.22	0.15	0.51	1.30	2.00	1.63	3.14
AGCRN	0.03	0.19	0.32	1.03	1.25	2.30	2.32	1.95
GWNET	0.06	0.07	0.11	1.06	1.31	1.76	2.24	2.79
ASTGCN	0.10	0.17	0.32	0.91	1.17	1.93	2.11	3.50
STTN	0.04	0.16	0.21	0.57	0.58	1.21	1.79	3.40
STDN	0.09	0.15	0.23	0.71	0.97	2.29	2.08	2.13
SynMorph	0.08	0.22	0.18	0.89	0.80	1.41	2.30	2.04

Table 8: Standard deviation over 5 independent experiments on brain signal datasets. 'R' means regression and 'C' means classification.

Methods	HGD R			HGD C	2A R			2A C
	MAE	**RMSE**	**MAPE**	**ACC(%)**	**MAE**	**RMSE**	**MAPE**	**ACC(%)**
Adashare	0.21	0.31	0.08	0.93	0.08	0.12	0.09	0.50
LearnToBranch	0.14	0.19	0.08	0.51	0.08	0.09	0.10	0.37
einspace	0.21	0.16	0.06	2.01	0.08	0.11	0.01	0.41
EEGNet	0.24	0.21	0.08	0.89	0.12	0.04	0.06	0.24
TCN	0.17	0.36	0.11	0.75	0.01	0.06	0.04	0.09
CompactCNN	0.23	0.14	0.13	2.10	0.04	0.06	0.09	0.37
Deep4Net	0.21	0.22	0.02	1.03	0.03	0.04	0.02	0.46
ShallowNet	0.18	0.15	0.04	1.00	0.09	0.11	0.05	0.44
CBraMod	0.31	0.17	0.10	2.23	0.01	0.07	0.07	0.38
SynMorph	0.17	0.12	0.05	1.09	0.03	0.04	0.01	0.49