Finetune Chiplet Design Floorplan via KuanNet

Xiaomeng Wang, Zhen Zhou, *Senior Member, IEEE*, and Yang Yi, *Senior Member, IEEE*

***Abstract*—Chiplet-based architectures require efficient through-silicon via (TSV) assignment to optimize interconnect performance and system integration. Unlike traditional 3-D integrated circuits, heterogeneous chiplet systems demand coordination across dies with varying sizes and functionalities, creating exponentially complex solution spaces that challenge existing optimization methods. This article introduces knowledge-unified attention neural network (KuanNet), a multiagent reinforcement learning (RL) framework integrating echo state networks (ESNs) with attention mechanisms for chiplet TSV assignment. The key innovation is a knowledge-unified architecture with temporal-static decomposition: temporal features shared across agents are processed through both ESN reservoirs and skip connections, while static features remain agent-private, enabling coordinated decisions with temporal memory and spatial awareness. Building on multiagent deep deterministic policy gradient (MADDPG) with K-head attention critics, KuanNet demonstrates superior optimization performance over the state-of-the-art baseline across standard benchmark circuits of varying scale and complexity. Ablation studies validate individual component contributions of the KuanNet architecture.**

***Index Terms*—Attention mechanism, chiplet design, echo state network (ESN), heterogeneous integration, multiagent reinforcement learning (RL), placement optimization, through-silicon via (TSV), TSV assignment.**

I. Introduction

THE semiconductor industry's transition to chiplet-based architectures represents one of the most significant paradigm shifts in integrated circuit design since the advent of system-on-chip (SoC) integration. Major products, including AMD's MI300 series accelerators, Intel's Ponte Vecchio GPUs, and NVIDIA's GB200 Grace Blackwell Superchip, have demonstrated that disaggregated multidie designs can achieve superior performance, yield, and cost-effectiveness compared to monolithic alternatives [1], [2], [3]. Central to these architectures are through-silicon vias (TSVs), which provide vertical interconnections enabling efficient connections between heterogeneous dies in advanced packaging technologies [4], [5]. However, TSV placement optimization directly determines key design metrics including interconnect performance, routing efficiency, thermal distribution, and signal integrity—making it a critical bottleneck in chiplet design flows.

Received 4 January 2026; revised 13 March 2026; accepted 30 March 2026. This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1750450, Grant ECCS-1731928, Grant ECCS-2128594, Grant ECCS-2314813, and Grant CCF-1937487. *(Corresponding author: Xiaomeng Wang.)*

Xiaomeng Wang and Yang Yi are with the Bradley Department of Electrical and Computer Engineering and the Institute for Advanced Computing, Virginia Tech, Alexandria, VA 22305 USA (e-mail: scholar@wangxm.com).

Zhen Zhou is with Intel Corporation, Chandler, AZ 85248 USA.

Data is available on-line at www.wangxm.com/research/KuanNet

Digital Object Identifier 10.1109/TVLSI.2026.3681746

While TSV assignment has been extensively studied for 3-D integrated circuits (3-D ICs) using classical optimization techniques such as simulated annealing [6], genetic algorithms [7], and thermal-aware heuristics [8], chiplet systems introduce fundamentally different challenges. Unlike vertically stacked homogeneous 3-D ICs where all dies share identical layouts and TSVs connect corresponding locations across layers, chiplet architectures feature: 1) heterogeneous dies with vastly different sizes, functionalities, and layouts on the same tier; 2) lateral die-to-die connections requiring complex horizontal routing through interposers or bridges; and 3) asymmetric TSV assignment patterns where different dies may require different numbers and placements of vertical interconnects. These characteristics create a significantly larger and more complex optimization space that challenges traditional optimization methods' assumptions of structural regularity.

Reinforcement learning (RL) offers a promising alternative, with recent successes in electronic design automation [9], [10], [11] showing that deep RL methods [12], [13] can learn effective heuristics from environment interaction. ATTA [14] pioneered multiagent RL for 3-D IC TSV assignment, but existing approaches face critical limitations for chiplet systems: 1) lack of coordination mechanisms across heterogeneous dies, as existing methods target homogeneous 3-D ICs exclusively and produce locally optimal but globally suboptimal placements; and 2) inability to capture temporal optimization dynamics—placement quality depends on trajectory history, which standard feedforward networks discard.

This article introduces knowledge-unified attention neural network (KuanNet), a multiagent RL framework addressing these challenges through three innovations. Chiplet designs induce combinatorial complexity from heterogeneous integration—we address this with attention-based multiagent deep deterministic policy gradient (MADDPG) [15], [16], [17] with K-head attention critics, where each interface operates as an independent agent under centralized training and decentralized execution. Multiagent cooperation in iterative placement demands trajectory awareness; we integrate ESN reservoirs [18], [19], [20] that maintain optimization history at a fraction of long short-term memory (LSTM) [21] computational cost. Finally, effective decentralized execution requires preserving each agent's spatial locality; our **knowledge-unified architecture with temporal-static decomposition** selectively shares temporal features (wirelength evolution and cost trajectories) across agents for coordination while keeping spatial features agent-private (detailed in Section III).

A. Contributions

By building a multiagent RL framework tailored to chiplet-based TSV assignment, integrating attention-based coordination, temporal trajectory modeling via reservoir

computing, and spatial-aware information decomposition, we make the following contributions.

1) *First Chiplet-Specific RL Framework:* We present the first multiagent RL approach designed for chiplet heterogeneous systems, addressing heterogeneous die placement, lateral connections, and coordination complexity through attention-based MADDPG [15], [16] with K-head critics, together with a refined action space tailored for discrete TSV candidate selection at chiplet interfaces.
2) *Knowledge-Unified Architecture:* We introduce a dual-pathway architecture with temporal-static decomposition, where temporal features (shared across agents) are processed through both ESN reservoirs and skip connections, while static features (agent-private) are processed directly. This is the first work to systematically decompose and selectively share state information in multiagent RL for semiconductor design.
3) *Expanded Action Space:* A composite action space combining neighboring positions with randomly selected distant empty spaces, balancing local refinement with global exploration across heterogeneous chiplet configurations.

Experiments on MCNC [22], [23] and GSRC [24] benchmarks demonstrate significant wirelength reduction over ATT-TA [14] on 3D ICs, with consistent improvements across both homogeneous (3-tier, 4-tier) and heterogeneous chiplet configurations. Ablation studies validate individual component contributions across benchmark sizes from ami33 to n300.

B. Article Organization

The remainder of this article is organized as follows. Section II provides background on multiagent RL and echo state networks (ESNs), then formalizes the chiplet TSV assignment problem and establishes the optimization objectives. Section III presents the KuanNet architecture, including the knowledge-unified dual-pathway design, ESN integration, and attention-based coordination mechanisms. Section IV describes the experimental setup, presents baseline comparison and ablation studies across MCNC and GSRC benchmarks, and discusses architectural insights. Section V offers concluding remarks.

II. Background and Problem Formulation

We map TSV assignment to a multiagent decision process before delving into algorithmic choices. Each interchiplet interface becomes an independent agent, creating a natural MARL formulation where agent observations mix global placement progress (e.g., wirelength trajectory) with local geometry and context. Joint actions consist of simultaneous TSV relocations across all interfaces, environment transitions correspond to floorplan updates via placement engines, and rewards measure instantaneous wirelength reduction. This formulation clarifies why MADDPG with centralized critics (for coordination) and decentralized actors (for scalable deployment) is well-suited for chiplet TSV assignment, and motivates the ESN-based temporal modeling that follows.

A. Multiagent RL

Multiagent RL extends single-agent RL to scenarios where multiple agents interact in a shared environment. We formulate the problem as a multiagent Markov decision process (MA-MDP).

An MA-MDP for N agents is defined by the tuple $\langle \mathcal{S}, \{\mathcal{O}_i\}, \{\mathcal{A}_i\}, \mathcal{T}, \{\mathcal{R}_i\}, \gamma \rangle$, where $\mathcal{S}$ is the global state space, $\mathcal{O}_i$ and $\mathcal{A}_i$ are the observation and action spaces for agent i, $\mathcal{T}$ is the state transition function, $\mathcal{R}_i$ is the reward function for agent i, and γ is the discount factor. We adopt MADDPG [15], which extends DDPG [16] to multiagent settings using centralized training with decentralized execution (CTDE): each agent learns a deterministic actor π_i using only local observations and a centralized critic Q_i with access to global state and all actions. Target networks are soft-updated as $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$ with $\tau \ll 1$.

B. Echo State Networks

ESNs [18], [19], [20] provide an efficient approach to temporal sequence modeling for RL applications. ESNs belong to the reservoir computing paradigm and offer a computationally efficient alternative to traditional recurrent neural networks (RNNs) such as LSTM [21] or GRU [25]. Unlike these architectures, ESNs decouple temporal feature extraction from gradient-based learning by using a fixed, randomly initialized reservoir, eliminating the need for backpropagation through time (BPTT).

The ESN architecture consists of an input layer, a recurrent reservoir layer, and a trainable readout layer. The reservoir dynamics are given by

$$\mathbf{h}^t = \tanh\left(\mathbf{W}_{\text{in}}\mathbf{x}^t + \mathbf{W}_{\text{res}}\mathbf{h}^{t-1} + \mathbf{b}\right) \tag{1}$$

$$\mathbf{y}^t = \mathbf{W}_{\text{out}}\left[\mathbf{h}^t; \mathbf{x}^t\right] \tag{2}$$

where $\mathbf{x}^t$ is the input, $\mathbf{h}^t$ is the reservoir state, and $\mathbf{y}^t$ is the output. Crucially, $\mathbf{W}_{\text{in}}$ (input weights), $\mathbf{W}_{\text{res}}$ (reservoir weights), and $\mathbf{b}$ (bias) are *randomly initialized once and never updated*, and $\tanh(\cdot)$ is the activation function with output range $[-1, 1]$. Only the readout layer $\mathbf{W}_{\text{out}}$ is trained. The concatenation $[\mathbf{h}^t; \mathbf{x}^t]$ in (2) represents a skip connection where the original input bypasses the reservoir and is directly fed to the readout layer, enabling the network to learn both direct input–output mappings and complex temporal dynamics.

The spectral radius $\rho(\mathbf{W}_{\text{res}}) = \max_i |\lambda_i(\mathbf{W}_{\text{res}})|$ controls memory retention. A larger spectral radius (closer to 1.0) enables longer memory retention, while smaller values cause faster forgetting. Values are typically scaled to 0.9–0.99 to ensure the echo state property—a condition guaranteeing that the reservoir state asymptotically depends only on the input history, not on initial conditions.

For additional temporal control, we employ leaky integration

$$\mathbf{h}^t = (1 - \alpha)\,\mathbf{h}^{t-1} + \alpha \tanh\left(\mathbf{W}_{\text{in}}\mathbf{x}^t + \mathbf{W}_{\text{res}}\mathbf{h}^{t-1} + \mathbf{b}\right) \tag{3}$$

where $\alpha \in (0, 1]$ is the leaky rate controlling the balance between retaining previous state and incorporating new activations. When $\alpha = 1$, this reduces to the standard ESN; smaller values (0.1–0.3) produce slower dynamics suited for long-term

dependencies, while larger values (0.7–1.0) yield faster, more responsive updates. The leaky rate provides complementary control to spectral radius: while spectral radius affects information propagation through recurrent connections, the leaky rate controls how quickly the reservoir responds to new inputs.

Advantages for RL: The key motivation for choosing ESNs over trainable RNNs is their fixed reservoir weights, which allow ESN modules to be plugged into any RL architecture with minimal changes to the training pipeline. This yields several practical advantages as follows.

1) *No BPTT:* Since reservoir weights are fixed, gradients only flowthrough the readout layer, eliminating vanishing/exploding gradient problems and significantly improving training stability. Unlike LSTM [21] and GRU [25], which require BPTT to learn gating, ESN's echo state property [18] provides rich temporal features without gradient-based temporal learning—critical in off-policy RL where single-step replay [26] prevents meaningful BPTT.
2) *Simple Replay Buffer Integration:* We store reservoir states $\mathbf{h}_i^t$ with experience transitions $(s_i^t, a_i^t, r_i^t, s_i^{t+1}, \mathbf{h}_i^t)$. During minibatch sampling, stored reservoir states can be directly used without recomputing temporal trajectories, enabling efficient off-policy learning. Because reservoir dynamics are invariant, stored states remain valid across training—unlike trainable RNNs, which suffer from *stale state* mismatch as weights evolve. Sequence replay with burn-in [27] mitigates this but adds complexity impractical for multiagent shared buffers.
3) *Parameter Efficiency:* ESN's reservoir consists of fixed, randomly initialized weights with only a linear readout layer trained, whereas LSTM and GRU require all recurrent gate parameters to be learned. For equivalent hidden dimensions, this yields 4–6× fewer trainable parameters in the temporal layer alone, accelerating learning and reducing overfitting risk.

C. TSV Assignment Problem Formulation

The TSV assignment problem is formalized below—placing TSVs to connect electrical nets spanning multiple dies while satisfying connectivity and placement constraints to optimize design objectives.

1) Mathematical Formulation: We formulate the TSV assignment problem in terms of electrical nets crossing chiplet interfaces. Consider a chiplet system with N interchiplet interfaces and a set of electrical nets $\mathcal{N} = \{n_1, n_2, \ldots, n_J\}$ where each net n_j may have terminals distributed across multiple dies. For each interface $i \in \{1, \ldots, N\}$, let $\mathcal{N}_i \subseteq \mathcal{N}$ denote the subset of nets that cross interface i (i.e., nets with terminals on both dies connected by interface i). Given L_i candidate TSV locations $\mathcal{L}_i = \{\ell_1, \ell_2, \ldots, \ell_{L_i}\}$ at interface i, the goal is to assign TSV locations to nets. While in practice some nets (e.g., power/ground, high-current signals) may require multiple TSVs per interface for redundancy or current capacity, we simplify the formulation by assigning exactly one TSV per net per interface, which is standard for most signal nets and enables tractable optimization. For nets requiring redundant TSVs, each additional TSV constitutes a separate assignment decision within the same agent's action sequence—scalability is linear in TSV count, not exponential. The multiagent framework naturally accommodates variable TSV counts per interface without architectural changes to KuanNet.

Let $\mathbf{A}_i \in \{0,1\}^{|\mathcal{N}_i| \times L_i}$ denote the assignment matrix for interface i, where $A_{i,j,k} = 1$ if net $n_j \in \mathcal{N}_i$ is assigned a TSV at location ℓ_k at interface i, and $A_{i,j,k} = 0$ otherwise. Note that a single net may cross multiple interfaces and thus require TSV assignments at each interface it traverses. In particular, nets connecting nonadjacent dies traverse intermediate interfaces as pass-throughs, requiring TSV assignments at each intermediate interface and contributing horizontal routing distance on intermediate dies to connect successive TSV locations. The problem is subject to the following constraints:

$$\sum_{k=1}^{L_i} A_{i,j,k} = 1 \quad \forall j \in \mathcal{N}_i \ \forall i \in \{1, \ldots, N\} \tag{4}$$

$$\sum_{j \in \mathcal{N}_i} A_{i,j,k} \leqslant 1 \quad \forall k \in \{1, \ldots, L_i\} \ \forall i \in \{1, \ldots, N\} \tag{5}$$

where (4) ensures each net n_j crossing an interface is assigned exactly one TSV location at that interface, and (5) prevents multiple nets from being assigned to the same TSV location ℓ_k at the same interface.

2) Objective Functions: While TSV assignment optimization can be formulated as a multiobjective problem incorporating metrics such as wirelength, signal integrity, thermal distribution, and routing congestion, multiobjective approaches require manually tuning weights among objectives—a process that is problem-dependent and lacks principled guidelines for weight selection [28]. Moreover, different weight configurations can lead to vastly different Pareto-optimal solutions, making fair performance comparison across methods difficult. To enable direct, reproducible evaluation and avoid the confounding effects of subjective weight tuning, this article focuses on wirelength minimization as the primary optimization objective. Wirelength serves as a fundamental metric for interconnect quality and directly impacts both performance (latency) and power consumption.

Wirelength Objective: The total wirelength is the sum of routing distances across all nets and all interfaces

$$f_{\text{wire}}(\mathbf{A}_1, \ldots, \mathbf{A}_N) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} d_{j,i}(\mathbf{A}_i) \tag{6}$$

where N is the number of interfaces (one between each adjacent die pair), $\mathcal{N}_i$ is the subset of nets crossing interface i, and $d_{j,i}(\mathbf{A}_i)$ is the routing distance contribution for net n_j at interface i under TSV assignment $\mathbf{A}_i$. For each net n_j at each interface i, the routing distance $d_{j,i}$ includes as follows.

1) *Horizontal Routing Within Each Die:* Minimum spanning tree (MST) is used to connect all terminals of n_j on each die adjacent to interface i.
2) *Vertical TSV Connections:* TSV routing distance at interface i if n_j crosses this interface (i.e., when $n_j \in \mathcal{N}_i$).

The outer sum over interfaces $\sum_{i=1}^{N}$ aggregates contributions across all die-to-die interfaces, while the inner sum $\sum_{j \in \mathcal{N}_i}$

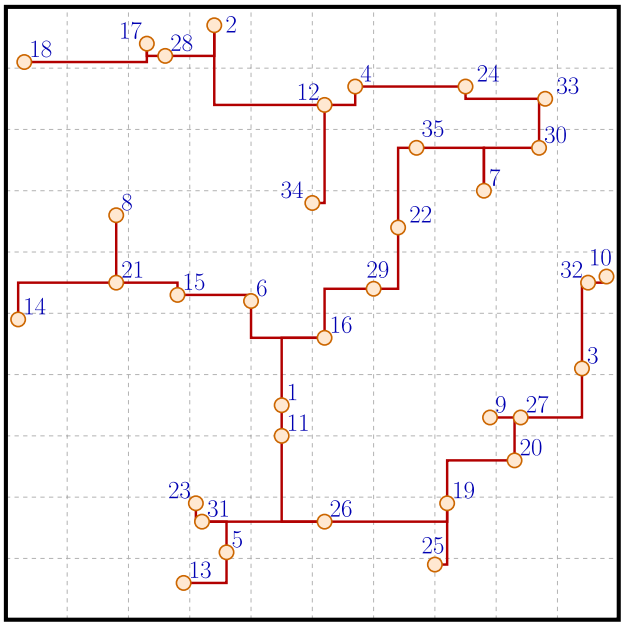


Fig. 1. MST for TSV routing distance computation showing 35 TSV locations connected via MST structure (red edges) to minimize total wirelength.

totals the wirelength over nets crossing each interface. In practice, we use MST heuristics to compute $d_{j,i}$ for each net-interface pair, as illustrated in Fig. 1.

3) Challenges in Chiplet TSV Assignment: Existing optimization approaches for chiplet TSV assignment face several fundamental challenges as follows.

1) *Computational Complexity:* The TSV assignment problem is combinatorially complex, with solution space growing exponentially with the number of nets and interfaces. This combinatorial explosion makes exhaustive search intractable for realistic problem sizes with hundreds of nets and multiple chiplet interfaces.
2) *Coordination:* TSV placement decisions are interdependent, requiring coordinated optimization across multiple dies and layers.
3) *Scalability:* Solution quality degrades rapidly with increasing system size and number of chiplets.
4) *Temporal Dynamics:* Although the target is a static placement, the optimization *process* exhibits temporal structure as follows.
    a) *Path dependence:* Each TSV perturbation alters the wirelength landscape for subsequent decisions, requiring trajectory-aware strategy adaptation.
    b) *Self-modifying environment:* Agents reduce placement disorder over time, shifting from exploratory to conservative regimes indistinguishable from single-step observations alone.
    c) *MARL nonstationarity* [17]*:* Other agents' evolving policies continuously shift each agent's perceived dynamics, necessitating temporal memory for implicit coordination.

These challenges motivate our multiagent RL approach, which can naturally handle temporal dynamics and coordination between multiple decision variables through learned policies.

III. Proposed KuanNet Framework

The KuanNet framework is designed from the ground up around the physical characteristics of chiplet TSV assignment. Each architectural choice maps to a specific design characteristic: the multiagent decomposition mirrors the natural partitioning of die interfaces, each with distinct spatial constraints and module densities; the observation space encodes physical layout features such as net lengths, empty space distributions, and connectivity patterns; and the knowledge-unified architecture integrates spatial awareness with temporal reasoning, enabling agents to adapt strategies based on both where they are on the die and how the optimization has progressed.

A. Multiagent RL Framework for TSV Assignment

Our approach models the TSV assignment problem as an MA-MDP where each **interchiplet interface** requiring TSV connectivity is modeled as an autonomous agent that selects optimal TSV placements from available locations within the chiplet. This formulation enables decentralized decision-making while maintaining global coordination through shared environmental feedback.

This multiagent formulation is necessitated by the combinatorial structure of the problem and provides four advantages specific to chiplet TSV assignment as follows.

1) *Action Space Tractability:* A single-agent controlling all N interfaces faces a joint action space growing as $|\mathcal{A}|^N$, whereas multiagent decomposition reduces this to N independent agents each with $|\mathcal{A}_i|$.
2) *Per-Interface Specialization:* Heterogeneous chiplet architectures feature dies with varying sizes, module densities, and connectivity patterns. Each agent develops die-specific policies attuned to its local characteristics—for instance, a dense compute die requires different TSV relocation strategies than a sparse I/O die.
3) *Per-Agent Credit Attribution:* The centralized critic decomposes the global objective into per-agent contributions, enabling each agent to learn how its local actions affect overall solution quality.
4) *Natural Scalability:* Varying chiplet counts are accommodated without retraining, as each new interface simply adds an agent with the same architecture while the attention critic dynamically adjusts its heads.

1) Agent Formulation: Each TSV agent i is defined by the tuple $\langle \mathcal{O}_i, \mathcal{A}_i, \mathcal{R}_i, \pi_i \rangle$ where the following aspects include.

Observation Space $\mathcal{O}_i$: Following the CTDE paradigm, each agent's actor observes a combination of temporal and static features at time t:

$$o_i^t = \left[\mathbf{s}_T^t; s_{i,S}^t\right] \tag{7}$$

where $\mathbf{s}_T^t$ represents sharable temporal features tracking optimization progress (cost trajectories, episode progress), and $s_{i,S}^t$ denotes agent-specific static features (spatial position). The

TABLE I
STATE FEATURES FOR RL FRAMEWORK

Feature	Range	T*	S†	Description
Cost Trajectory Features (Temporal)				
$c(s)$	$[-1,1]$	✓	✓	Current state cost
$c^\star$	$[-1,1]$	✓	✓	Best achieved cost
$\bar{c}$	$[-1,1]$	✓	✓	Average cost over episode
$\bar{c}^\star$	$[-1,1]$	✓	✓	Average cost since best
Temporal Control Features				
t	$[0,1]$	✓	✗	Normalized episode timestep
$\Delta t^\star$	$[0,1]$	✓	✗	Normalized episode timestep since best
Network & Spatial Features (Non-temporal)				
n	$[0,1]$	✗	✗	Normalized number of affected nets
l	$[0,1]$	✗	✗	Net length percentage
$\bar{w}$	$[0,1]$	✗	✗	Average net complexity
P_A	$[0,1]$	✗	✗	Empty space area percentile
P_E	$[0,1]$	✗	✗	Empty space edges count percentile
P_{wlr}	$[0,1]$	✗	✗	Empty space width vs. length ratio percentile
P_{AR}	$[0,1]$	✗	✗	Empty space area vs. boundary ratio percentile
Action & Location Features (Non-temporal)				
δ	$\{0,1\}$	✗	✗	Perturbation type indicator
v	$\{0,1\}$	✗	✗	Action validity indicator
$\hat{c}_i(s')$	$[-1,1]$	✗	✗	Action i induced normalized cost

*T: Temporal feature (✓) or non-temporal (✗).
†S: Sharable with other agents' actors. Critics have access to all features.

detailed feature composition is presented in Table I, which categorizes features by their temporal nature and shareability across agents.

Table I presents a comprehensive overview of the state features used in our RL framework for TSV assignment optimization. The features are carefully designed to capture both temporal dynamics and spatial characteristics essential for effective decision-making. We categorize features into temporal and nontemporal types, where temporal features (marked with ✓) track the evolution of optimization metrics over time, including cost trajectories (current cost $c(s)$, best achieved cost $c^\star$, average cost $\bar{c}$, and average cost since best $\bar{c}^\star$) and episode progress. These temporal features provide agents with a high-level, bird's-eye view of the optimization process, enabling proactive rather than reactive placement strategies. Nontemporal features (marked with ✗) capture spatial and structural properties.

Feature sharing across agents' actors is determined by the nature of the information: global metrics like best achieved cost and average cost are sharable to facilitate coordination, while agent-specific observations, such as sampled neighbor costs and local perturbation information, can remain private to maintain decentralized execution. All cost-based features are normalized following the approach in ATT-TA [14] to enable fair comparison with the baseline. This feature set enables agents to balance immediate local optimization decisions with long-term global objectives while maintaining computational efficiency through selective information sharing.

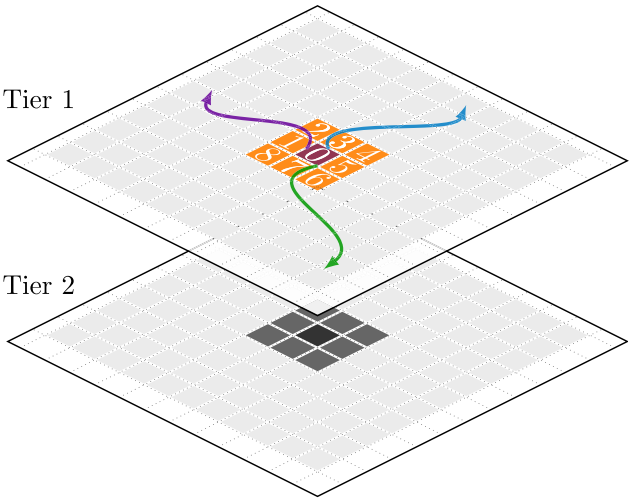


Fig. 2. Illustration of the action space design for TSV assignment. The action space consists of the current TSV location (center), $|\mathcal{A}_i^{\text{nb}}|$ neighboring positions (8-connected neighborhood shown), and $|\mathcal{A}_i^{\text{es}}|$ randomly selected locations from distant empty spaces to encourage exploration. The neighborhood size and distant candidate count are tunable; sensitivity analysis is provided in Section IV.

Action Space $\mathcal{A}_i$: The action space for each TSV assignment consists of multiple candidate locations designed to balance local refinement with global exploration. Let $\mathcal{A}_i^0$ denote the current TSV location. The action space is defined as

$$\mathcal{A}_i = \mathcal{A}_i^0 \cup \mathcal{A}_i^{\text{nb}} \cup \mathcal{A}_i^{\text{es}} \tag{8}$$

where $\mathcal{A}_i^0$ is the current TSV location (enabling the agent to maintain its existing placement if optimal), $\mathcal{A}_i^{\text{nb}}$ represents the neighborhood candidates for local perturbation, and $\mathcal{A}_i^{\text{es}}$ denotes randomly selected locations from other available empty spaces on the same chiplet for global exploration. As illustrated in Fig. 2, this design provides $|\mathcal{A}_i| = 1 + |\mathcal{A}_i^{\text{nb}}| + |\mathcal{A}_i^{\text{es}}|$ total candidate locations. Random locations are independently sampled at each step, ensuring diverse coverage of the solution space while maintaining a tractable action set. The default values ($|\mathcal{A}_i^{\text{nb}}| = 8$, $|\mathcal{A}_i^{\text{es}}| = 3$) were determined via sensitivity analysis (see Section IV).

Reward Function $\mathcal{R}_i$: Following our focus on wirelength optimization (see Section II), we employ a direct reward signal based on the raw change in total wirelength without additional normalization or penalty terms. Since our framework adopts a collaborative multiagent paradigm, all agents receive the same shared reward based on the global optimization objective. The reward for each agent i at timestep t is defined as

$$r_i^t = -\left(\mathfrak{W}^t - \mathfrak{W}^{t-1}\right) = \mathfrak{W}^{t-1} - \mathfrak{W}^t \tag{9}$$

where $\mathfrak{W}^t$ represents the global design wirelength after the action at timestep t, and $\mathfrak{W}^{t-1}$ is the wirelength before the action. A positive reward indicates wirelength reduction (improvement), while a negative reward represents wirelength increase (degradation). Unlike ATT-TA [14], which incorporates adversarial reward shaping with hand-crafted penalties to discourage inefficient exploration, our approach relies solely on the raw wirelength change signal, enabling unbiased multiagent coordination.

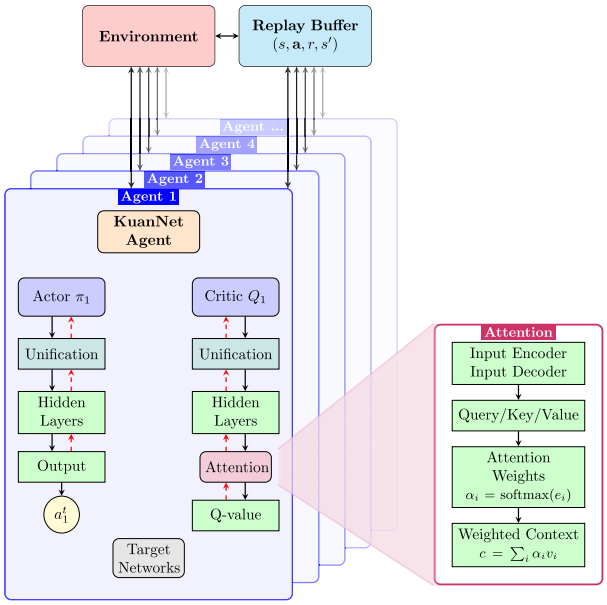


Fig. 3. KuanNet multiagent architecture with knowledge-unified processing. Each agent maintains an actor–critic pair with ESN+MLP dual-pathway input layers. The attention mechanism (zoomed-in view) processes all agents' actions for coordination.

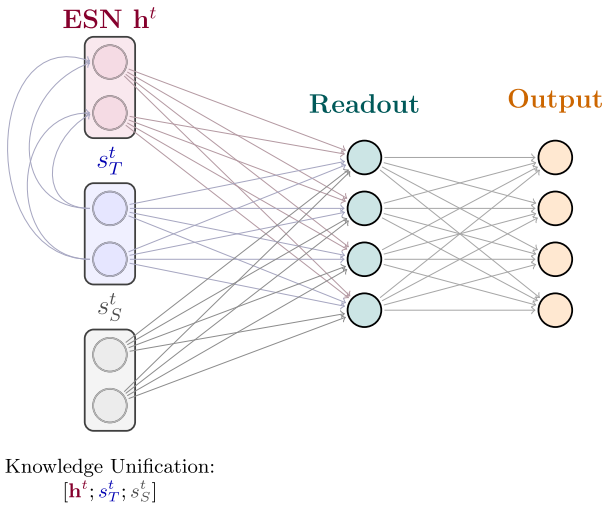


Fig. 4. Knowledge-unified neural network architecture showing the hybrid processing pathway. Temporal features connect to the ESN reservoir via curved pathways (visualized from the left) while also providing skip connections directly to the readout layer. The ESN reservoir state, temporal features, and static features are unified through concatenation before the final readout transformation.

2) Coordination Mechanisms: To address the coordination challenges inherent in multiagent TSV assignment, we implement several mechanisms as follows.

CTDE: Each agent i maintains its own actor π_i and critic Q_i. During training, each agent's actor receives shared temporal features $\mathbf{s}_T^t$ and its own private static features $s_{i,S}^t$, while its critic has access to all agents' temporal features, static features, and actions, enabling better value estimation. During deployment, agents execute their policies independently using only their observations, ensuring scalability.

Selective Information Sharing: Temporal features tracking optimization progress (cost trajectories and episode metrics) can be selectively shared among agents' actors to facilitate coordination. This sharing is configurable, enabling ablation studies to evaluate the impact of different information sharing strategies on multiagent cooperation and final performance.

Shared Experience Replay: Critical experiences that lead to significant improvements in global objectives are shared across all agents to accelerate learning.

B. Algorithm Architecture

Our KuanNet framework integrates ESNs (see Section II) with attention MADDPG (ATT-MADDPG) to enable effective coordination and temporal reasoning for TSV assignment. The framework follows the CTDE paradigm while introducing a novel *knowledge-unified architecture with selective temporal information sharing*: temporal features (e.g., wirelength evolution, optimization trajectories) can be shared among agents and processed through both ESN reservoir and direct skip connections, while static features (e.g., spatial positions and connectivity structure) remain private and are processed directly. All three components—ESN reservoir state, temporal skip features, and static features—are unified through concatenation before the readout layer. This knowledge unification enables agents to leverage complementary information: temporal memory from ESN, original temporal patterns from skip connections, and spatial structure from static features. Fig. 3 illustrates the overall architecture of our attention-based MARL framework with knowledge-unified temporal information sharing.

1) Knowledge-Unified Architecture With Temporal Information Sharing: Information Decomposition: Each agent's state observation s_i^t is decomposed into two components as follows.

1) *Temporal Features $s_{i,T}^t$:* Time-varying features including wirelength evolution, optimization trajectory, and temporal TSV assignment patterns. This information is *sharable* among agents to enable coordinated temporal reasoning through ESN processing.
2) *Static Features $s_{i,S}^t$:* Spatial positions, connectivity structure, and current assignment state. This information remains *private* to each agent and is processed directly.

Fig. 4 illustrates this knowledge-unified architecture, showing the three parallel processing streams—ESN reservoir, temporal skip connection, and static pathway—unified through concatenation before the readout layer.

2) Actor Network Formulation: For each agent i, the actor network processes temporal features (sharable across agents) through ESN and static features (private) directly, following the knowledge unification architecture:

$$\mathbf{s}_T^t = \bigcup_{i=1}^{N} s_{i,T}^t \tag{10}$$

$$\mathbf{h}_i^{\pi,t} = \tanh\left(\mathbf{W}_{\text{in}}^{\pi}\mathbf{s}_T^t + \mathbf{W}_{\text{res}}^{\pi}\mathbf{h}_i^{\pi,t-1}\right) \tag{11}$$

$$a_i^t = \pi_i\left(\mathbf{u}_i^{\pi,t}; \theta_i^{\pi}\right) = \pi_i\left(\left[\mathbf{h}_i^{\pi,t}; \mathbf{s}_T^t; s_{i,S}^t\right]; \theta_i^{\pi}\right) \tag{12}$$

where $\mathbf{s}_T^t$ takes the union of each agent's temporal feature set, forming a fixed-dimension vector where overlapping features are not duplicated, tracking optimization progress metrics (cost

trajectories, episode progress, improvement indicators)—the specific sharing configuration is adjustable, allowing agents to receive either their own temporal features, aggregated features from all agents, or selectively shared subsets depending on the experimental configuration; $\mathbf{W}_{\text{in}}^{\pi}$ and $\mathbf{W}_{\text{res}}^{\pi}$ are *randomly initialized, fixed* matrices for the ESN reservoir (never trained); $[\cdot;\cdot]$ denotes concatenation, and θ_i^{π} represents the trainable actor readout parameters. The knowledge unification produces the actor unified representation $\mathbf{u}_i^{\pi,t} = [\mathbf{h}_i^{\pi,t}; \mathbf{s}_T^t; s_{i,S}^t]$ by concatenating three components: ESN reservoir state $\mathbf{h}_i^{\pi,t}$ (capturing temporal memory and nonlinear dynamics), temporal skip features $\mathbf{s}_T^t$ (preserving original time-varying patterns), and static features $s_{i,S}^t$ (maintaining spatial and structural information). The ESN reservoir processes sharable temporal features to enable coordinated temporal dynamics with fixed weights, while the direct pathways preserve complementary information for decentralized execution.

3) Critic Network Formulation: The critic network follows the same knowledge unification architecture, processing each agent's state-action information individually. For agent i's critic, the representation of each agent $j \in \{1,\ldots,N\}$ is computed as

$$\mathbf{h}_{i,j}^{Q,t} = \tanh\left(\mathbf{W}_{\text{in}}^{Q}\left[s_{j,T}^t; a_j^t\right] + \mathbf{W}_{\text{res}}^{Q}\mathbf{h}_{i,j}^{Q,t-1}\right) \tag{13}$$

$$\mathbf{u}_{i,j}^{Q,t} = \left[\mathbf{h}_{i,j}^{Q,t}; s_{j,T}^t; s_{j,S}^t\right] \tag{14}$$

where each agent j's temporal features and action are processed through agent i's critic ESN reservoir with *fixed random weights* ($\mathbf{W}_{\text{in}}^{Q}$, $\mathbf{W}_{\text{res}}^{Q}$), and the knowledge-unified representations $\mathbf{u}_{i,j}^{Q,t}$ concatenate ESN reservoir state, temporal skip features, and static features. The unified representations $\{\mathbf{u}_{i,1}^{Q,t}, \mathbf{u}_{i,2}^{Q,t}, \ldots, \mathbf{u}_{i,N}^{Q,t}\}$ serve as input to the attention mechanism.

Following ATT-MADDPG's architecture [17], the critic uses a K-head attention mechanism to model the dynamic joint policy of teammates. For agent i, the critic first generates K action-conditional Q value heads, where each head k represents a potential coordination pattern

$$Q_i^k\left(\mathbf{s}, a_i; \theta_i^{Q,k}\right) = f_k\left(\left[\mathbf{u}_{i,i}^{Q,t}; a_i\right]; \theta_i^{Q,k}\right), \quad k \in \{1,\ldots,K\} \tag{15}$$

where $f_k(\cdot;\theta_i^{Q,k})$ is a feedforward neural network for head k producing an embedding $Q_i^k \in \mathbb{R}^{d_k}$ instead of a scalar value, and $\mathbf{u}_{i,i}^{Q,t}$ is agent i's own knowledge-unified representation from (14).

A hidden vector $\mathbf{z}_i^t$ is computed from teammates' knowledge-unified representations to capture the joint policy information

$$\mathbf{z}_i^t = g\left(\left[\mathbf{u}_{i,1}^{Q,t}; \ldots; \mathbf{u}_{i,i-1}^{Q,t}; \mathbf{u}_{i,i+1}^{Q,t}; \ldots; \mathbf{u}_{i,N}^{Q,t}\right]; \theta_i^{Q,g}\right) \tag{16}$$

where $g(\cdot;\theta_i^{Q,g})$ is a feedforward neural network that encodes all teammates' representations (excluding agent i's own representation).

4) K-Head Attention Mechanism: Following ATT-MADDPG [17], the K-head attention mechanism computes scaled dot-product attention weights over the K Q value heads using the teammates' hidden vector $\mathbf{z}_i^t$ [29]

$$\alpha_i^k = \text{softmax}\left(\frac{\left(\mathbf{z}_i^t\right)^T Q_i^k}{\sqrt{d_k}}\right) \tag{17}$$

where d_k is the dimension of Q value head vectors. The $1/\sqrt{d_k}$ scaling factor (absent in original ATT-MADDPG [17]) prevents gradient instability. The final Q value is the attention-weighted sum passed through an output network $h(\cdot;\theta_i^{Q,h})$

$$Q_i\left(\mathbf{s}, \mathbf{a}; \theta_i^{Q}\right) = h\left(\sum_{k=1}^{K} \alpha_i^k \cdot Q_i^k; \theta_i^{Q,h}\right). \tag{18}$$

5) Training and Backpropagation: The KuanNet framework follows the standard MADDPG training procedure with centralized training and decentralized execution. The training process alternates between critic updates using temporal difference (TD) learning and actor updates using deterministic policy gradients.

a) Critic loss and backpropagation: For each agent i, the critic network is trained to minimize the TD error. The critic loss is computed as

$$\mathcal{L}\left(\theta_i^{Q}\right) = \mathbb{E}_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')}\left[\left(Q_i\left(\mathbf{s}, \mathbf{a}; \theta_i^{Q}\right) - y_i\right)^2\right] \tag{19}$$

where the TD target y_i is computed using target networks

$$y_i = r_i + \gamma Q_i'\left(\mathbf{s}', \mathbf{a}'; \theta_i^{Q'}\right) \tag{20}$$

with $\mathbf{a}' = [\pi_1'(\mathbf{u}_1^{\pi,t+1}; \theta_1^{\pi'}), \ldots, \pi_N'(\mathbf{u}_N^{\pi,t+1}; \theta_N^{\pi'})]$ being the target actions from next-state representations.

b) Actor policy gradient: Each actor network is updated to maximize the expected return by following the deterministic policy gradient. The actor's objective for agent i is

$$J\left(\theta_i^{\pi}\right) = \mathbb{E}_{\mathbf{s}}\left[Q_i\left(\mathbf{s}, \mathbf{a}; \theta_i^{Q}\right)\Big|_{a_i=\pi_i\left(\mathbf{u}_i^{\pi,t};\theta_i^{\pi}\right)}\right]. \tag{21}$$

The policy gradient is computed by backpropagating through the critic to the actor

$$\nabla_{\theta_i^{\pi}} J = \mathbb{E}_{\mathbf{s}}\left[\nabla_{\theta_i^{\pi}}\pi_i\left(\mathbf{u}_i^{\pi,t}; \theta_i^{\pi}\right) \cdot \nabla_{a_i} Q_i\left(\mathbf{s}, \mathbf{a}; \theta_i^{Q}\right)\Big|_{a_i=\pi_i\left(\mathbf{u}_i^{\pi,t};\theta_i^{\pi}\right)}\right]. \tag{22}$$

In both critic and actor updates, only the readout layers (output MLPs, K-head attention parameters) receive gradient updates. The ESN reservoir weights ($\mathbf{W}_{\text{in}}$, $\mathbf{W}_{\text{res}}$) remain fixed throughout training, leveraging the Echo State Property as a rich temporal feature extractor while eliminating vanishing/exploding gradient issues common in trained RNNs. Target networks are soft-updated as

$$\theta_i^{Q'} \leftarrow \tau\theta_i^{Q} + (1-\tau)\,\theta_i^{Q'} \tag{23}$$

$$\theta_i^{\pi'} \leftarrow \tau\theta_i^{\pi} + (1-\tau)\,\theta_i^{\pi'} \tag{24}$$

with $\tau \ll 1$ to stabilize training.

Algorithm 1 provides a complete specification of the KuanNet training procedure, integrating the key innovations of our approach: knowledge-unified architecture with temporal-static feature decomposition, fixed ESN reservoirs for efficient temporal modeling, and K-head attention-enhanced CTDE. The algorithm explicitly shows how ESN reservoir weights remain fixed throughout training (line 7), how temporal information is shared among agents while static features remain private (lines 16-17), and how the K-head attention mechanism models teammates' joint policy during critic updates (line 24).

Algorithm 1 KuanNet: Knowledge-Unified Attention Network for Chiplet Design TSV Assignment

Input: Heterogeneous chiplets design with N TSV agents; State space $\mathcal{S}$ (temporal s_T, static s_S), action space $\mathcal{A}$ (8)
Input: Hyperparameters: α_π (actor lr), α_Q (critic lr), γ (discount), τ (target update), B (batch), K (attention heads); ESN: N_{res}, ρ, σ_{in}
Output: Decentralized actor policies $\{\pi_1, \ldots, \pi_N\}$ for optimization on defined metric
// *Initialization*
for each agent $i \in \{1, \ldots, N\}$ **do**
  Init actor θ_i^π, critic θ_i^Q (incl. K-head MLPs, hidden MLP); Copy to targets: $\theta_i^{\pi'}, \theta_i^{Q'}$
  Init **fixed** ESN: $\mathbf{W}_{\text{in}}^\pi, \mathbf{W}_{\text{res}}^\pi, \mathbf{W}_{\text{in}}^Q, \mathbf{W}_{\text{res}}^Q \sim \mathcal{U}(-\sigma_{\text{in}}, \sigma_{\text{in}})$; Scale $\mathbf{W}_{\text{res}}$ to ρ; **Fix**; $\mathbf{h}_i^0 = \mathbf{0}$
end for
Init replay buffer $\mathcal{D} \leftarrow \emptyset$
// *Training Loop*
for episode $e = 1$ to M **do**
  Reset env, observe $\mathbf{s}^0$; Reset ESN: $\mathbf{h}_i^0 = \mathbf{0}\ \forall i$
  for timestep $t = 0$ to $T-1$ **do**
    // *Decentralized Execution*
    for each agent $i \in \{1, \ldots, N\}$ **do**
      Decompose: $s_{i,T}^t$ (temporal, shared), $s_{i,S}^t$ (static, local); Aggregate: $\mathbf{s}_T^t = \bigcup_{i=1}^N s_{i,T}^t$ (10)
      ESN: $\mathbf{h}_i^{\pi,t} = \tanh(\mathbf{W}_{\text{in}}^\pi \mathbf{s}_T^t + \mathbf{W}_{\text{res}}^\pi \mathbf{h}_i^{\pi,t-1})$ (11); Unify: $\mathbf{u}_i^{\pi,t} = [\mathbf{h}_i^{\pi,t}; \mathbf{s}_T^t; s_{i,S}^t]$ (12); Select: $a_i^t = \pi_i(\mathbf{u}_i^{\pi,t}; \theta_i^\pi)$
    end for
    Execute $\mathbf{a}^t$; Observe $\mathbf{r}^t$, $\mathbf{s}^{t+1}$ (9); Store in $\mathcal{D}$
    // *Centralized Training*
    if $|\mathcal{D}| \geq B$ **then**
      Sample minibatch of B transitions from $\mathcal{D}$
      for each agent $i \in \{1, \ldots, N\}$ **do**
        Compute target $a_j' = \pi_j'(\mathbf{u}_j^{\pi,t+1}; \theta_j^{\pi'})\ \forall j$; knowledge-unified $\mathbf{u}_{i,j}^{Q,t}$, $\mathbf{u}_{i,j}^{Q,t+1}$ (13)–(14); K-head attention Q_i^k, weights α_i^k (15)–(18)
        Critic: $y_i = r_i + \gamma Q_i'(\mathbf{s}', \mathbf{a}'; \theta_i^{Q'})$ (20); $\mathcal{L} = \frac{1}{B}\sum_{b=1}^B (Q_i(\mathbf{s}^b, \mathbf{a}^b) - y_i^b)^2$ (19); Update: $\theta_i^Q \leftarrow \theta_i^Q - \alpha_Q \nabla \mathcal{L}$
        Actor: $\nabla_{\theta_i^\pi} J = \mathbb{E}_{\mathbf{s}}[\nabla_{\theta_i^\pi} \pi_i(\cdot) \cdot \nabla_{a_i} Q_i(\cdot)|_{a_i=\pi_i(\cdot)}]$ (22); Update: $\theta_i^\pi \leftarrow \theta_i^\pi + \alpha_\pi \nabla J$
        Targets: $\theta_i^{Q'} \leftarrow \tau\theta_i^Q + (1-\tau)\theta_i^{Q'}$, $\theta_i^{\pi'} \leftarrow \tau\theta_i^\pi + (1-\tau)\theta_i^{\pi'}$ (23) ad (24)
      end for
    end if
  end for
end for
return Trained policies $\{\pi_1(\cdot; \theta_1^\pi), \ldots, \pi_N(\cdot; \theta_N^\pi)\}$

IV. Experimental Evaluation

This section presents a comprehensive experimental evaluation of the proposed KuanNet framework for chiplet TSV assignment optimization. We first describe our implementation and experimental configuration, then evaluate KuanNet through baseline comparison against ATT-TA, ablation studies isolating key architectural components (ESN integration, temporal modeling, and information sharing), and sensitivity analysis of the action space design. Experiments span industry-standard MCNC and GSRC benchmarks across both homogeneous 3-D IC and heterogeneous chiplet configurations, with wirelength reduction as the primary optimization metric.

A. Implementation and Experimental Setup

1) System Architecture: The implementation decouples the RL optimization engine from the physical design environment: the environment exposes a standard `step`/`reset` interface that computes placement cost from the current TSV configuration, while the agents interact solely through observations, actions, and scalar rewards. This separation allows new optimization objectives—thermal, congestion, signal integrity—to be incorporated by extending the environment's reward computation without modifying the agent architecture or training pipeline. The multiagent framework scales linearly with interface count: each die boundary is an independent agent instance sharing the same network architecture, so adding dies or assigning multiple TSVs per net requires only instantiating additional agents.

2) Multiagent Environment: The chiplet TSV assignment environment provides multiagent coordination capabilities as follows.

State Representation: Each agent receives spatial position information, wirelength metrics for current placement, local neighborhood information, and historical assignment decisions (via ESN memory).

Action Processing: Actions are processed through conflict resolution for simultaneous assignments, constraint checking for spatial and electrical feasibility, and wirelength calculation based on Manhattan distance.

Reward Computation: Rewards are computed based on wirelength changes as defined in (9), providing agents with direct feedback on placement quality without additional normalization or penalty terms.

TSV Initialization: All experiments use a greedy centroid-based workflow that places each TSV at the closest valid empty grid location to its net's pin centroid, falling back to random placement when no greedy position is available. This provides a consistent initial environment $\mathfrak{W}_0$ shared across all methods.

B. Experimental Configuration

1) Platform and Software: Experiments are conducted on multiple Apple Mac mini units (M4 chip, 16 GB unified memory) using PyTorch with metal performance shaders (MPS) backend, Python 3.12, Hydra [30] for configuration management, and [31] for model design and training guidelines.

2) Training Hyperparameters: Table II summarizes the key hyperparameters used for training KuanNet. Parameters marked with † are shared with the ATT-TA baseline for fair comparison, while KuanNet-specific parameters like ESN configuration reflect our architectural innovations. These values were selected based on preliminary experiments and held fixed across all benchmarks.

TABLE II
TRAINING HYPERPARAMETERS FOR KUANNET

Parameter	Value	Parameter	Value
RL Training†		*Exploration*†	
Episodes	20000	Type	Gumbel
Steps/episode	50	Init Temp	3
γ	0.99	Final Temp	0.01
τ	0.001		
Buffer size	1M		
Batch size	128		
Actor Network†		*Actor ESN*	
Hidden layers	[32, 32]	Reservoir size	36
Learning rate	0.0001	ρ (spectral)	0.99
Optimizer	AdamW	α (leaky)	0.8
Critic Network†		*Critic ESN*	
Hidden layers	[32, 32]	Reservoir size	40
K-head heads	4	ρ (spectral)	0.99
Learning rate	0.001	α (leaky)	0.6
Optimizer	AdamW		

† Parameters shared with ATT-TA baseline for fair comparison.

3) Benchmarks and Experimental Configurations: We evaluate our approach using industry-standard MCNC [22], [23] (ami33 with 33 blocks and ami49 with 49 blocks) and GSRC [24] (n100, n200, and n300) benchmarks across two architectural scenarios. For 3-D IC architectures with homogeneous stacking (3-tier and 4-tier configurations), we compare KuanNet against the ATT-TA baseline. For heterogeneous Chiplet architectures featuring a horizontally split top layer (resulting in 4 and 5 total chiplets for 3-tier and 4-tier base configurations, respectively), we conduct KuanNet-only ablation studies to isolate the contributions of our architectural innovations. In total, this yields 20 benchmark-design configurations (5 benchmarks $\times$4 designs). Input floorplans are generated using FlexPlanner [32], a RL-based 3-D floorplanner that produces block-to-die partitioning and within-die placement for each benchmark configuration. Notably, these benchmarks differ by an order of magnitude in die dimensions (under 400 μm for GSRC versus several thousand for MCNC); we intentionally do not normalize across benchmark families, directly testing algorithmic generalization across diverse spatial scales.

4) Evaluation Metrics: We evaluate algorithm performance using the percentage wirelength reduction $\Delta\mathfrak{W}\%$, which normalizes across benchmarks of different scales

$$\Delta\mathfrak{W}\% = \frac{\Delta\mathfrak{W}}{\mathfrak{W}_0} \times 100 = \frac{\mathfrak{W}_0 - \mathfrak{W}^\star}{\mathfrak{W}_0} \times 100 \tag{25}$$

where $\mathfrak{W}_0$ is the total wirelength of the initial greedy TSV placement and $\mathfrak{W}^\star$ is the total wirelength after optimization. Higher $\Delta\mathfrak{W}\%$ indicates greater optimization. We also report the final optimized wirelength $\mathfrak{W}^\star$ and initial wirelength $\mathfrak{W}_0$ where applicable.

Each configuration is evaluated across multiple independent training runs with different random seeds. We report the best test performance per configuration, assessed using an average rank methodology: for each (dataset, design) pair, variants are ranked by $\Delta\mathfrak{W}\%$; ranks are then averaged across all pairs to produce a single scalar, where a lower rank indicates better performance. KuanNet ablation studies use 20 ranking pairs (5 datasets $\times$4 designs), while ATT-TA comparisons use ten pairs (5 datasets $\times$2 3-D IC designs). This ranking approach provides robustness against scale differences across benchmarks and prevents individual outliers from dominating the comparison.

TABLE III
KUANNET VERSUS ATT-TA WIRELENGTH ON 3-D IC BENCHMARKS
($\mathfrak{W}_0$, $\Delta\mathfrak{W}$, $\Delta\mathfrak{W}\%$, $\mathfrak{W}^\star$ IN $\times 10^3\,\mu$M)

Benchmark	$\mathfrak{W}_0$	ATT-TA			KuanNet		
		$\Delta\mathfrak{W}$	$\Delta\mathfrak{W}\%$	$\mathfrak{W}^\star$	$\Delta\mathfrak{W}$	$\Delta\mathfrak{W}\%$	$\mathfrak{W}^\star$
3-Tier 3D IC							
ami33	139.7	1.7	1.2	138.0	**35.2**	**25.2**	**104.5**
ami49	1,999.7	2.8	0.1	1996.9	**625.9**	**31.3**	**1373.8**
n100	234.4	2.9	1.2	231.5	**86.7**	**37.0**	**147.7**
n200	546.9	3.5	0.6	543.4	**185.1**	**33.9**	**361.8**
n300	693.0	3.4	0.5	689.6	**179.7**	**25.9**	**513.3**
4-Tier 3D IC							
ami33	134.3	2.5	1.9	131.8	**31.4**	**23.4**	**102.9**
ami49	2,041.4	3.4	0.2	2038.0	**905.9**	**44.4**	**1135.5**
n100	207.0	2.6	1.3	204.4	**84.7**	**40.9**	**122.3**
n200	448.2	4.3	1.0	443.9	**181.2**	**40.4**	**267.0**
n300	716.2	4.8	0.7	711.4	**269.1**	**37.6**	**447.1**

$\mathfrak{W}_0$: initial wirelength (greedy initialization). $\mathfrak{W}^\star$: final optimized wirelength. $\Delta\mathfrak{W}\%$: percentage reduction $= \Delta\mathfrak{W}/\mathfrak{W}_0 \times 100$. All values in $\times 10^3\,\mu$m. All methods share identical $\mathfrak{W}_0$.

C. Algorithm Comparison Results

We present experimental results in five parts: First, we compare KuanNet against the state-of-the-art ATT-TA baseline on 3-D IC benchmarks to establish competitive performance on standard homogeneous stacking configurations. Second, we conduct ablation studies on both 3-D IC and heterogeneous chiplet architectures to isolate the contributions of ESN integration across actor and critic networks. Third, we compare ESN against alternative temporal architectures, including LSTM, GRU, state history concatenation, and exponential moving average (EMA) to validate our choice of reservoir computing. Fourth, we analyze the impact of different information-sharing configurations between agents to validate our knowledge-unified architecture design. Fifth, we perform action space sensitivity analysis on neighborhood size and distant candidate count. Together, these results demonstrate that KuanNet achieves substantial wirelength reduction improvements while maintaining architectural flexibility across diverse chiplet configurations.

1) Comparison of KuanNet and ATT-TA on 3-D IC Benchmarks: Table III compares KuanNet against the ATT-TA baseline on homogeneous 3-D IC configurations (3-tier and 4-tier stacking) across all five benchmarks. Both algorithms use identical RL training hyperparameters (see Table II) and reward function (9) for fair comparison.

KuanNet substantially outperforms ATT-TA across all benchmarks. On 3-tier 3-D ICs, KuanNet achieves 20.7 – −223.5$\times$ larger wirelength reductions, averaging 76$\times$. For 4-tier configurations, gains range from 12.6$\times$ to 266.4$\times$,

TABLE IV
IMPACT OF NETWORK ARCHITECTURE ON KUANNET
($\Delta\mathfrak{W}\%$, $\mathfrak{W}^\star$ IN $\times 10^3\,\mu$M)

Benchmark	FC†		A-ESN‡		C-ESN§		Both*		Full¶	
3-Tier 3D IC	$\mathfrak{W}_0$: *ami33=139.7, ami49=1999.7, n100=234.4, n200=546.9, n300=693.0*									
ami33	18.9%	113.3	19.2%	112.9	21.6%	109.6	24.2%	105.9	**25.2%**	**104.5**
ami49	27.4%	1451.0	26.4%	1472.3	29.5%	1409.4	29.1%	1418.1	**31.3%**	**1373.8**
n100	32.9%	157.3	35.6%	151.1	36.6%	148.6	34.9%	152.6	**37.0%**	**147.7**
n200	32.0%	371.8	31.1%	376.7	**34.0%**	**361.0**	33.9%	361.4	33.9%	361.8
n300	21.8%	541.7	20.1%	553.6	**26.4%**	**510.4**	25.6%	515.4	25.9%	513.3
Avg. Rank		4.4		4.4		1.8		2.8		**1.6**
4-Tier 3D IC	$\mathfrak{W}_0$: *ami33=134.3, ami49=2041.4, n100=207.0, n200=448.2, n300=716.2*									
ami33	15.9%	113.0	22.0%	104.8	19.1%	108.7	**24.7%**	**101.2**	23.4%	102.9
ami49	36.4%	1298.0	44.1%	1140.9	36.0%	1305.8	**44.5%**	**1132.3**	44.4%	1135.5
n100	30.6%	143.6	39.1%	126.1	35.2%	134.2	38.7%	126.8	**40.9%**	**122.3**
n200	28.4%	320.9	39.3%	271.9	31.5%	306.8	**41.2%**	**263.4**	40.4%	267.0
n300	25.9%	530.4	34.1%	472.3	27.6%	518.6	**37.9%**	**444.5**	37.6%	447.1
Avg. Rank		4.8		2.8		4.2		**1.4**		1.8
3-Tier Chiplets	$\mathfrak{W}_0$: *ami33=128.9, ami49=1865.0, n100=219.9, n200=485.3, n300=668.9*									
ami33	13.0%	112.2	5.4%	121.9	12.4%	113.0	12.4%	113.0	**13.9%**	**111.0**
ami49	26.4%	1373.6	**26.9%**	**1363.7**	20.3%	1486.2	25.7%	1384.9	25.5%	1389.3
n100	34.4%	144.3	28.5%	157.2	23.9%	167.3	34.4%	144.2	**35.2%**	**142.6**
n200	30.7%	336.2	29.1%	344.2	21.2%	382.4	**31.1%**	**334.2**	30.9%	335.4
n300	26.7%	490.0	**27.1%**	**487.6**	24.8%	502.9	25.6%	497.6	24.9%	502.1
Avg. Rank		**2.4**		3.0		4.6		2.6		**2.4**
4-Tier Chiplets	$\mathfrak{W}_0$: *ami33=145.9, ami49=1950.6, n100=197.9, n200=427.7, n300=676.2*									
ami33	21.5%	114.5	16.9%	121.3	20.9%	115.5	20.8%	115.6	**26.1%**	**107.8**
ami49	40.0%	1171.0	32.3%	1321.3	35.6%	1257.0	41.9%	1133.4	**43.4%**	**1103.1**
n100	33.4%	131.8	24.2%	149.9	31.6%	135.3	19.6%	159.2	**34.7%**	**129.3**
n200	38.3%	264.0	30.4%	297.7	29.0%	303.5	28.4%	306.2	**40.0%**	**256.7**
n300	35.3%	437.5	29.6%	476.3	24.2%	512.8	36.1%	432.3	**37.2%**	**424.7**
Avg. Rank		2.4		4.2		3.8		3.6		**1.0**
Overall		3.50		3.60		3.60		2.60		**1.70**

$\mathfrak{W}^\star$: final total wirelength ($\times 10^3\,\mu$m); $\Delta\mathfrak{W}\%$: reduction from initial placement. Best per benchmark in **bold**. †FC: Both use fully connected. ‡A-ESN: Actor uses ESN. §C-ESN: Critic uses ESN. *Both: Both use ESN. ¶Full: Both use ESN with sharing.

TABLE V
COMPARISON OF VARIOUS TEMPORAL ARCHITECTURES
($\Delta\mathfrak{W}\%$, $\mathfrak{W}^\star$ IN $\times 10^3\,\mu$M)

Benchmark	Hist.†		EMA‡		GRU§		LSTM#		ESN¶	
3-Tier 3D IC	$\mathfrak{W}_0$: *ami33=139.7, ami49=1999.7, n100=234.4, n200=546.9, n300=693.0*									
ami33	19.7%	112.2	8.6%	127.8	23.7%	106.6	20.2%	111.6	**25.2%**	**104.5**
ami49	22.6%	1548.2	19.9%	1601.5	22.9%	1541.6	23.2%	1536.2	**31.3%**	**1373.8**
n100	26.4%	172.6	18.8%	190.4	33.4%	156.1	26.0%	173.5	**37.0%**	**147.7**
n200	22.0%	426.8	19.4%	440.6	30.7%	379.0	23.8%	416.7	**33.9%**	**361.8**
n300	22.9%	534.4	16.6%	578.1	20.5%	551.0	23.3%	531.4	**25.9%**	**513.3**
Avg. Rank		3.6		5.0		2.6		2.8		**1.0**
4-Tier 3D IC	$\mathfrak{W}_0$: *ami33=134.3, ami49=2041.4, n100=207.0, n200=448.2, n300=716.2*									
ami33	17.3%	111.1	16.2%	112.5	23.6%	102.7	**23.6%**	**102.7**	23.4%	102.9
ami49	33.8%	1351.8	32.9%	1370.6	42.6%	1172.1	41.5%	1193.3	**44.4%**	**1135.5**
n100	32.4%	139.8	27.6%	149.9	39.0%	126.3	38.9%	126.5	**40.9%**	**122.3**
n200	29.1%	317.9	30.5%	311.6	39.4%	271.7	**41.0%**	**264.6**	40.4%	267.0
n300	26.3%	528.1	28.4%	512.6	34.2%	470.9	**38.2%**	**442.7**	37.6%	447.1
Avg. Rank		4.4		4.6		2.4		**1.8**		**1.8**
3-Tier Chiplets	$\mathfrak{W}_0$: *ami33=128.9, ami49=1865.0, n100=219.9, n200=485.3, n300=668.9*									
ami33	6.5%	120.5	3.7%	124.1	11.2%	114.4	10.7%	115.2	**13.9%**	**111.0**
ami49	17.9%	1531.3	20.2%	1488.0	19.1%	1509.3	23.8%	1420.5	**25.5%**	**1389.3**
n100	20.8%	174.2	25.1%	164.8	22.7%	169.9	25.2%	164.5	**35.2%**	**142.6**
n200	18.6%	395.0	24.5%	366.6	21.7%	380.0	25.3%	362.7	**30.9%**	**335.4**
n300	16.2%	560.5	18.6%	544.5	17.8%	549.6	19.6%	537.6	**24.9%**	**502.1**
Avg. Rank		4.8		3.4		3.6		2.2		**1.0**
4-Tier Chiplets	$\mathfrak{W}_0$: *ami33=145.9, ami49=1950.6, n100=197.9, n200=427.7, n300=676.2*									
ami33	21.3%	114.9	26.3%	107.6	18.1%	119.5	**26.4%**	**107.4**	26.1%	107.8
ami49	36.3%	1241.6	41.5%	1140.9	32.4%	1319.4	**44.4%**	**1084.0**	43.4%	1103.1
n100	32.5%	133.7	34.5%	129.6	27.2%	144.0	**37.8%**	**123.0**	34.7%	129.3
n200	36.1%	273.3	**40.1%**	**256.0**	28.2%	307.3	39.8%	257.6	40.0%	256.7
n300	31.9%	460.3	36.3%	430.4	24.8%	508.8	**37.7%**	**421.1**	37.2%	424.7
Avg. Rank		4.0		2.4		5.0		**1.4**		2.2
Overall		4.20		3.85		3.40		2.05		**1.50**

$\mathfrak{W}^\star$: final total wirelength ($\times 10^3\,\mu$m); $\Delta\mathfrak{W}\%$: reduction from initial placement. Best per benchmark in **bold**. †Hist.: state history concatenation (K=10). ‡EMA: exponential moving average (α=0.3). §GRU: Gated Recurrent Unit (single-step, no BPTT). #LSTM: Long Short-Term Memory (single-step, no BPTT). ¶ESN: Echo State Network (fixed reservoir, trainable readout).

averaging 82×. These results validate our architectural innovations: ESN-based temporal processing and K-head attention provide superior multiagent coordination compared to ATT-TA's feedforward networks. Notably, while ATT-TA employs attention mechanisms, its lack of temporal memory and reliance on reward shaping penalties encourage local greedy improvements rather than global coordination, limiting scalability to complex multichiplet configurations.

2) Impact of ESN on KuanNet Performance: Table IV demonstrates the performance impact of incorporating ESNs into the KuanNet framework. This ablation systematically isolates ESN contributions: FC uses standard fully connected networks; A-ESN and C-ESN employ ESN in actor or critic only; both use ESN in both networks; only Full implements complete ESN-based temporal sharing among agents.

The full configuration with complete ESN-based knowledge sharing achieves the best performance across most benchmarks. Against the FC baseline, Full delivers substantial improvements on 3-tier ami33 and 4-tier chiplet ami33. Intermediate configurations (A-ESN, C-ESN, Both) show mixed results, with some benchmarks improving while others degrade. Notably, the both configuration (ESN in both networks but no interagent sharing) sometimes underperforms FC, indicating that ESN requires coordinated temporal information sharing to be effective. These findings validate our core principle: ESN-based temporal memory provides benefits only when combined with the complete knowledge-unified sharing mechanism.

3) Temporal Architecture Ablation: Table V compares ESN against alternative temporal models: LSTM, GRU, state history concatenation (K = 10), and EMA (α = 0.3). All models use full interagent collaboration and share the same MADDPG training pipeline with single-step state updates (no BPTT). ESN uses 4×–6× fewer trainable parameters in the temporal layer, since its reservoir is fixed and randomly initialized while only a linear readout is trained—unlike LSTM and GRU, which require learning all recurrent gate weights.

ESN achieves the best overall average rank (1.50) across all four design configurations, with LSTM as a competitive second (2.05). ESN's advantage is most pronounced on 3-tier configurations (average rank 1.0 for both 3-D IC and chiplets), while on 4-tier designs LSTM ties ESN on 3-D IC (both 1.8) and surpasses ESN on chiplets (1.4 versus 2.2), suggesting that the increased optimization complexity of additional tiers provides richer per-step learning signals that partially compensate for the lack of BPTT. Simpler temporal approaches—EMA (3.85) and state history concatenation (4.20)—consistently trail, confirming that effective temporal modeling requires structured recurrent processing rather than simple aggregation of past states. While sequence-based BPTT training is possible in off-policy RL, it requires additional infrastructure (stored trajectories, truncated BPTT windows, and careful sequence sampling); ESN sidesteps this entirely as a drop-in temporal module requiring no changes to the training pipeline or replay buffer.

4) Impact of Information Sharing Configurations: A critical design decision in multiagent RL is determining what information should be shared between agents and how it should be processed. Our architecture provides flexible configuration options for information sharing, enabling systematic ablation studies to understand the contribution of different sharing strategies. We investigate sharing configurations across two dimensions: 1) which temporal features are shared between actors and 2) how shared information is processed (ESN pathway, skip connection, or both).

The KuanNet architecture supports four information-sharing configurations as follows.

FC (Fully Connected): Both actors and critics use standard feedforward networks without ESN-based temporal sharing. Coordination relies on attention mechanisms during centralized training.

A-ESN (Actor ESN Sharing): Actors share temporal features through ESN reservoirs: $\mathbf{h}_i^{\pi,t} = \tanh(\mathbf{W}_{\text{in}}^{\pi}\mathbf{s}_T^t + \mathbf{W}_{\text{res}}^{\pi}\mathbf{h}_i^{\pi,t-1})$, with output $a_i^t = \pi_i([\mathbf{h}_i^{\pi,t}; \mathbf{s}_T^t; s_{i,S}^t]; \theta_i^{\pi})$. Critics remain fully connected.

C-ESN (Critic ESN Sharing): Critics share temporal features via ESN while actors use fully connected networks. This configuration explores whether temporal memory benefits centralized value estimation.

Full (Complete ESN Sharing): Both actors and critics employ ESN-based temporal sharing as described in Section III, providing comprehensive temporal coordination across all network components.

Table VI presents the performance of different information sharing configurations across our benchmark suite. The results reveal important insights about the interplay between architecture design, problem characteristics, and information-sharing strategies.

The sharing configuration results demonstrate that optimal strategies are architecture-dependent. For both 3-tier and 4-tier 3-D ICs, Full achieves the best performance across all benchmarks. Heterogeneous chiplet configurations exhibit more complex patterns: 4-tier chiplets consistently favor Full sharing with dramatic improvements, while 3-tier chiplets show variable optimal strategies across benchmarks. These results reveal that homogeneous 3-D IC architectures benefit from balanced actor–critic coordination, whereas heterogeneous chiplet topologies require the full knowledge-unified architecture for complex interchiplet dependencies.

TABLE VI

ROLE OF INTERAGENT INFORMATION SHARING ($\Delta\mathfrak{W}\%$, $\mathfrak{W}^\star$ IN $\times 10^3\,\mu$M)

Benchmark	FC†		A-ESN‡		C-ESN§		Full¶	
3-Tier 3D IC	$\mathfrak{W}_0$: *ami33=139.7, ami49=1999.7, n100=234.4, n200=546.9, n300=693.0*							
ami33	10.9%	124.6	18.6%	113.7	15.8%	117.7	**25.2%**	**104.5**
ami49	20.0%	1600.3	22.3%	1553.3	26.7%	1466.5	**31.3%**	**1373.8**
n100	19.1%	189.6	24.3%	177.4	36.1%	149.8	**37.0%**	**147.7**
n200	19.9%	438.0	20.6%	434.1	30.9%	377.9	**33.9%**	**361.8**
n300	14.0%	596.0	16.2%	580.6	13.2%	601.7	**25.9%**	**513.3**
Avg. Rank		3.8		2.6		2.6		**1.0**
4-Tier 3D IC	$\mathfrak{W}_0$: *ami33=134.3, ami49=2041.4, n100=207.0, n200=448.2, n300=716.2*							
ami33	19.9%	107.6	13.3%	116.5	22.5%	104.0	**23.4%**	**102.9**
ami49	35.8%	1311.5	33.8%	1351.9	**45.3%**	**1117.7**	44.4%	1135.5
n100	32.0%	140.8	29.4%	146.2	**41.2%**	**121.7**	40.9%	122.3
n200	30.8%	310.1	27.2%	326.2	39.6%	270.6	**40.4%**	**267.0**
n300	29.0%	508.7	27.2%	521.3	30.2%	499.8	**37.6%**	**447.1**
Avg. Rank		3.0		4.0		1.6		**1.4**
3-Tier Chiplets	$\mathfrak{W}_0$: *ami33=128.9, ami49=1865.0, n100=219.9, n200=485.3, n300=668.9*							
ami33	**13.9%**	**111.0**	8.2%	118.4	11.3%	114.4	13.9%	111.0
ami49	**28.4%**	**1335.9**	22.4%	1446.4	25.0%	1398.0	25.5%	1389.3
n100	34.0%	145.2	24.2%	166.6	33.1%	147.1	**35.2%**	**142.6**
n200	29.7%	341.1	20.3%	386.8	29.8%	340.6	**30.9%**	**335.4**
n300	**26.5%**	**491.5**	24.2%	507.4	23.5%	511.8	24.9%	502.1
Avg. Rank		**1.6**		3.8		3.0		**1.6**
4-Tier Chiplets	$\mathfrak{W}_0$: *ami33=145.9, ami49=1950.6, n100=197.9, n200=427.7, n300=676.2*							
ami33	14.2%	125.2	**27.3%**	**106.1**	24.4%	110.4	26.1%	107.8
ami49	32.5%	1317.0	38.4%	1202.2	26.1%	1441.4	**43.4%**	**1103.1**
n100	23.2%	152.0	**36.8%**	**125.1**	31.9%	134.7	34.7%	129.3
n200	30.4%	297.6	37.5%	267.3	25.1%	320.3	**40.0%**	**256.7**
n300	28.1%	486.3	31.3%	464.8	21.0%	534.3	**37.2%**	**424.7**
Avg. Rank		3.4		1.6		3.6		**1.4**
Overall		2.95		3.00		2.70		**1.35**

$\mathfrak{W}^\star$: final total wirelength ($\times 10^3\,\mu$m); $\Delta\mathfrak{W}\%$: reduction from initial placement. Best per benchmark in **bold**. †FC: No ESN sharing. ‡A-ESN: Actor shares via ESN. §C-ESN: Critic shares via ESN. ¶Full: Both share via ESN.

5) Action Space Sensitivity Analysis: Tables VII and VIII present one-at-a-time sensitivity sweeps of the two action space parameters. Table VII varies the neighborhood size $|\mathcal{A}_i^{\text{nb}}|$ from 4 to 48 while fixing distant candidates at 3. Table VIII sweeps distant candidates from 1 to 9 with $|\mathcal{A}_i^{\text{nb}}| = 8$ fixed. The 8-neighborhood achieves the best overall rank (1.45), consistently outperforming both smaller neighborhoods that lack diagonal coverage and larger ones where the expanded action space and observation input dimension (see Table I) increase training complexity without proportional benefit. Distant candidate count is robust, with 3 candidates achieving the best overall rank (1.20) and diminishing returns beyond 3–5. The optimal combination may depend on network hyperparameters and chiplet design complexity; we determine these values via a two-stage sweep (neighborhood first, then distant candidates) and report a well-rounded configuration across our benchmark suite.

TABLE VII

NEIGHBORHOOD SIZE $|\mathcal{A}_i^{\mathrm{NB}}|$ ABLATION STUDY ($\Delta\mathfrak{W}\%$, $\mathfrak{W}^\star$ IN $\times 10^3\,\mu$M)

Benchmark	4		8		12		24		48	
3-Tier 3D IC	$\mathfrak{W}_0$: *ami33=139.7, ami49=1999.7, n100=234.4, n200=546.9, n300=693.0*									
ami33	8.3%	128.1	**25.2%**	**104.5**	20.4%	111.2	17.2%	115.7	23.2%	107.3
ami49	17.3%	1654.5	**31.3%**	**1373.8**	22.7%	1545.2	26.3%	1473.5	23.1%	1538.5
n100	16.3%	196.3	**37.0%**	**147.7**	26.8%	171.6	29.5%	165.2	29.6%	165.1
n200	15.9%	460.2	**33.9%**	**361.8**	17.9%	449.3	29.7%	384.6	28.8%	389.3
n300	11.4%	614.2	**25.9%**	**513.3**	19.6%	557.4	13.6%	598.5	10.2%	622.1
Avg. Rank	4.8		**1.0**		3.4		2.8		3.0	
4-Tier 3D IC	$\mathfrak{W}_0$: *ami33=134.3, ami49=2041.4, n100=207.0, n200=448.2, n300=716.2*									
ami33	17.1%	111.4	**23.4%**	**102.9**	19.3%	108.3	16.3%	112.4	11.9%	118.4
ami49	29.7%	1435.8	**44.4%**	**1135.5**	29.4%	1441.2	29.7%	1436.1	42.0%	1183.9
n100	25.5%	154.2	**40.9%**	**122.3**	25.7%	153.8	18.2%	169.3	38.5%	127.4
n200	29.3%	317.0	**40.4%**	**267.0**	24.1%	340.3	30.3%	312.4	32.3%	303.6
n300	24.3%	541.8	**37.6%**	**447.1**	24.5%	541.0	21.3%	563.9	31.2%	493.0
Avg. Rank	3.6		**1.0**		3.6		4.2		2.6	
3-Tier Chiplets	$\mathfrak{W}_0$: *ami33=128.9, ami49=1865.0, n100=219.9, n200=485.3, n300=668.9*									
ami33	10.7%	115.1	**13.9%**	**111.0**	8.3%	118.2	6.1%	121.0	9.9%	116.2
ami49	25.7%	1384.9	25.5%	1389.3	25.8%	1384.1	18.0%	1529.3	**28.2%**	**1339.0**
n100	**36.4%**	**139.9**	35.2%	142.6	35.1%	142.8	27.0%	160.6	34.0%	145.2
n200	29.6%	341.9	30.9%	335.4	**31.5%**	**332.4**	21.6%	380.5	24.8%	365.1
n300	**25.4%**	**499.0**	24.9%	502.1	24.2%	507.2	11.2%	593.7	22.8%	516.2
Avg. Rank	**2.0**		2.2		2.6		5.0		3.2	
4-Tier Chiplets	$\mathfrak{W}_0$: *ami33=145.9, ami49=1950.6, n100=197.9, n200=427.7, n300=676.2*									
ami33	26.0%	108.0	**26.1%**	**107.8**	26.1%	107.9	24.6%	110.0	17.2%	120.8
ami49	37.3%	1222.3	**43.4%**	**1103.1**	39.2%	1185.0	41.3%	1144.4	33.0%	1306.4
n100	34.2%	130.1	34.7%	129.3	**37.3%**	**124.1**	35.3%	128.0	24.5%	149.4
n200	37.1%	269.0	40.0%	256.7	37.5%	267.2	**40.2%**	**255.7**	26.6%	314.1
n300	34.2%	444.9	**37.2%**	**424.7**	35.9%	433.2	37.0%	425.9	16.5%	564.4
Avg. Rank	3.8		**1.6**		2.4		2.2		5.0	
Overall	3.55		**1.45**		3.00		3.55		3.45	

$\mathfrak{W}^\star$: final total wirelength ($\times 10^3\,\mu$m); $\Delta\mathfrak{W}\%$: reduction from initial placement. Best per benchmark in **bold**. $|\mathcal{A}_i^{\mathrm{es}}|$=3 fixed.

TABLE VIII

EMPTY SPACE ACTIONS $|\mathcal{A}_i^{\mathrm{ES}}|$ ABLATION STUDY ($\Delta\mathfrak{W}\%$, $\mathfrak{W}^\star$ IN $\times 10^3\,\mu$M)

Benchmark	1		3		5		7		9	
3-Tier 3D IC	$\mathfrak{W}_0$: *ami33=139.7, ami49=1999.7, n100=234.4, n200=546.9, n300=693.0*									
ami33	2.1%	136.9	**25.2%**	**104.5**	22.0%	109.0	17.4%	115.5	17.1%	115.8
ami49	12.2%	1754.9	**31.3%**	**1373.8**	29.4%	1411.5	14.2%	1715.3	25.1%	1497.5
n100	15.2%	198.7	**37.0%**	**147.7**	34.1%	154.5	34.4%	153.7	25.9%	173.8
n200	10.6%	488.9	**33.9%**	**361.8**	30.5%	380.1	32.1%	371.2	28.9%	388.8
n300	13.3%	600.6	**25.9%**	**513.3**	19.8%	556.1	19.1%	560.9	20.5%	551.3
Avg. Rank	5.0		**1.0**		2.6		3.0		3.4	
4-Tier 3D IC	$\mathfrak{W}_0$: *ami33=134.3, ami49=2041.4, n100=207.0, n200=448.2, n300=716.2*									
ami33	17.7%	110.5	23.4%	102.9	**25.0%**	**100.8**	21.6%	105.3	16.8%	111.8
ami49	33.7%	1352.6	**44.4%**	**1135.5**	40.8%	1207.9	39.2%	1240.3	36.9%	1288.1
n100	35.4%	133.7	**40.9%**	**122.3**	39.7%	124.7	38.0%	128.2	38.5%	127.4
n200	30.0%	313.9	**40.4%**	**267.0**	40.4%	267.1	39.9%	269.4	30.5%	311.4
n300	27.5%	519.3	**37.6%**	**447.1**	34.1%	471.8	28.1%	514.8	31.5%	491.0
Avg. Rank	4.8		**1.2**		1.8		3.4		3.8	
3-Tier Chiplets	$\mathfrak{W}_0$: *ami33=128.9, ami49=1865.0, n100=219.9, n200=485.3, n300=668.9*									
ami33	2.0%	126.3	**13.9%**	**111.0**	9.9%	116.1	9.1%	117.2	7.0%	119.9
ami49	**27.5%**	**1352.0**	25.5%	1389.3	25.0%	1399.6	25.8%	1383.6	17.8%	1532.8
n100	34.9%	143.2	**35.2%**	**142.6**	32.1%	149.4	32.3%	148.8	33.5%	146.3
n200	30.0%	339.9	**30.9%**	**335.4**	28.1%	349.0	28.8%	345.5	27.4%	352.1
n300	**26.4%**	**492.6**	24.9%	502.1	22.6%	517.8	23.1%	514.1	24.2%	507.1
Avg. Rank	2.2		**1.6**		4.0		3.2		4.0	
4-Tier Chiplets	$\mathfrak{W}_0$: *ami33=145.9, ami49=1950.6, n100=197.9, n200=427.7, n300=676.2*									
ami33	17.4%	120.6	**26.1%**	**107.8**	23.2%	112.0	11.8%	128.8	14.1%	125.3
ami49	33.0%	1306.4	**43.4%**	**1103.1**	40.9%	1153.6	29.7%	1371.0	32.4%	1319.2
n100	29.5%	139.4	**34.7%**	**129.3**	30.0%	138.6	25.9%	146.7	23.4%	151.6
n200	30.0%	299.6	**40.0%**	**256.7**	37.2%	268.5	26.1%	316.2	24.3%	323.8
n300	26.5%	497.2	**37.2%**	**424.7**	36.7%	428.1	22.8%	522.1	22.2%	526.1
Avg. Rank	3.0		**1.0**		2.0		4.4		4.6	
Overall	3.75		**1.20**		2.60		3.50		3.95	

$\mathfrak{W}^\star$: final total wirelength ($\times 10^3\,\mu$m); $\Delta\mathfrak{W}\%$: reduction from initial placement. Best per benchmark in **bold**. $|\mathcal{A}_i^{\mathrm{nb}}|$=8 fixed. *Overall rank for 9 excludes 4-Tier 3D IC (not tested).

D. Discussion

Our experiments reveal key insights about the knowledge-unified ESN architecture for multiagent TSV optimization. We first analyze the large performance gap with ATT-TA, then examine how ESN's fixed reservoir provides temporal memory without BPTT overhead, why sharing strategies must adapt to physical design structure, how CTDE enables scalable coordination, and finally outline directions for future work.

1) Performance Gap Analysis With ATT-TA: The large performance gap between KuanNet and ATT-TA (see Table III) reflects fundamental architectural differences rather than incomplete convergence. Three factors contribute: 1) ATT-TA's adversarial reward shaping with hand-crafted penalty terms encourages local greedy descent, trapping agents in local optima—an effect amplified in heterogeneous chiplet environments with complex optimization landscapes; 2) ATT-TA implements its policy with feedforward networks that discard observation history, despite its theoretical foundation ATT-MADDPG [17] explicitly requiring history-dependent reasoning; and 3) ATT-TA's action space is restricted to binary accept/reject decisions within a small perturbation radius, whereas KuanNet's action space (8) includes both local neighborhood and randomly sampled distant locations for global exploration.

2) ESN Temporal Memory Benefits: Fixed ESN reservoirs provide computational efficiency while capturing temporal dependencies in wirelength evolution. The temporal-static feature decomposition enables effective coordination—temporal features provide global optimization context through sharing, while static features preserve agent-specific spatial information. However, ESN benefits emerge only when combined with complete interagent sharing; partial ESN integration without the full knowledge-unified architecture fails to consistently improve performance.

3) Off-Policy Training Compatibility: All temporal models use single-step hidden state updates in the replay buffer following the DRQN approach [26], without BPTT. While R2D2 [27] proposes sequence replay as an improvement, this adds complexity impractical for multiagent shared replay. This regime inherently favors ESN: its fixed reservoir provides stable temporal features without BPTT, whereas LSTM [21] and GRU [25] cannot leverage learned gating without temporal gradient flow.

More broadly, beyond ATT-MADDPG [17], ESN reservoirs can be integrated into other cooperative multiagent RL frameworks not originally designed for recurrent temporal processing, with minimal pipeline changes and overhead. Since ESN weights are fixed and only the readout layer is trained, adding an ESN module requires no modifications to the existing training loop, loss functions, or replay buffer structure. This makes ESN a practical drop-in temporal module for any feedforward-based multiagent architecture where history-dependent reasoning could benefit coordination, without introducing the complexity of BPTT or sequence-based replay.

4) Architecture-Dependent Sharing Strategies: Homogeneous 3-D IC configurations show consistent improvements with Full sharing across benchmarks, while heterogeneous chiplet topologies exhibit more complex patterns. When computational resources are constrained, prioritizing critic-side ESN sharing (C-ESN) often provides stronger gains than actor-only sharing, aligning with the CTDE paradigm where centralized value estimation benefits from rich temporal information. This performance asymmetry between homogeneous and heterogeneous configurations confirms that architectural choices respond to physical design structure: uniform die stacks benefit uniformly from shared temporal context, whereas chiplet assemblies with varying die sizes and module densities require more nuanced sharing strategies to avoid conflating distinct spatial characteristics across dissimilar interfaces. Notably, agents sharing the same neural network architecture produce distinct optimization trajectories when assigned to chiplets of different complexity—agents on intricate, densely packed chiplets converge more slowly and face harder early-stage adjustments than agents on simpler chiplets. ESN's temporal memory encodes these trajectory differences (convergence rate, reward trends, and optimization phase), enabling each agent to adapt its policy to the difficulty of its assigned chiplet without requiring separate network configurations.

5) Multiagent Coordination: The CTDE paradigm successfully addresses TSV optimization challenges by enabling global coordination during training while maintaining scalable decentralized execution. Temporal feature sharing among agents enables coordinated wirelength optimization without explicit communication during deployment. The K-head attention mechanism in centralized critics effectively aggregates multiagent information, allowing agents to learn which teammates' actions are most relevant for coordination.

6) Scalability: KuanNet maintains consistent performance across benchmarks spanning diverse scales—from small MCNC circuits (ami33 and ami49) to larger GSRC designs (n100–n300)—despite order-of-magnitude differences in die dimensions and net counts, demonstrating that the multiagent coordination framework generalizes across problem sizes without architecture or hyperparameter changes.

7) Opportunities for Future Work: Several promising directions build on the current framework as follows.

a) *Multiobjective Optimization:* The reward function naturally extends to $r_t = w_1 \cdot \Delta\mathfrak{W} + w_2 \cdot \Delta T_{\max} + \cdots$ to incorporate thermal, congestion, and signal integrity objectives. ESN's temporal features are particularly suited for these metrics, as changes to one net's TSV placement affect thermal distribution and routing congestion in neighboring areas both within and across dies—cascading effects that unfold over sequential moves and benefit from temporal memory.

b) *Industrial-Scale Validation:* The multiagent architecture scales linearly with interface count, providing a foundation for designs with thousands of nets and proprietary chiplet configurations.

c) *Transfer Learning:* Pretraining on diverse chiplet layouts and fine-tuning on target configurations could substantially reduce per-design training cost, enabling rapid deployment across chiplet families.

V. Conclusion

This article presents KuanNet, a multiagent RL framework for chiplet TSV assignment that addresses the unique challenges of heterogeneous die integration. Our key innovation is a knowledge-unified architecture with temporal-static decomposition, where ESNs provide temporal memory for optimization trajectory learning while maintaining computational efficiency through fixed reservoir weights. The dual-pathway design—processing temporal features through both ESN reservoirs and skip connections while keeping static features agent-private—enables coordinated decisions with spatial awareness. Building on MADDPG with K-head attention critics, this architecture achieves effective multiagent coordination as policies evolve during training.

Experimental results demonstrate substantial wirelength reduction over the state-of-the-art ATT-TA baseline across MCNC and GSRC benchmarks. Ablation studies validate individual contributions of ESN integration, temporal architecture choice, and information sharing strategies across benchmark sizes from ami33 to n300 on both homogeneous 3-D IC and heterogeneous chiplet configurations. These results establish the viability of reservoir computing-enhanced multiagent RL for semiconductor design optimization.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] X. Wang, Z. Zhou, and Y. Yi, "Chapter 13-transforming ai landscape with neuromorphic computing and chiplets," in *Energy-Efficient Devices and Circuits for Neuromorphic Computing*. Amsterdam, The Netherlands: Elsevier, 2026, ch. 1, pp. 405–428. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780443299810000045

[2] S. Chen, H. Zhang, Z. Ling, J. Zhai, and B. Yu, "The survey of chiplet-based integrated architecture: An EDA perspective," 2024, *arXiv:2411.04410*.

[3] G. Shan, Y. Zheng, C. Xing, D. Chen, G. Li, and Y. Yang, "Architecture of computing system based on chiplet," *Micromachines*, vol. 13, no. 2, p. 205, Jan. 2022.

[4] J. H. Lau, *Chiplet Design and Heterogeneous Integration Packaging*. Cham, Switzerland: Springer, 2023.

[5] J. Knechtel, "Interconnect planning for physical design of 3D integrated circuits," Ph.D. dissertation, Fac. Elect. Eng. Inf. Technol., Dresden Univ. Technol., Dresden, Germany, 2014. [Online]. Available: https://tu-dresden.de/ing/elektrotechnik

[6] J. Ao, S. Dong, S. Chen, and S. Goto, "Through-silicon-via assignment for 3D ICs," in *Proc. 9th IEEE Int. Conf. ASIC*, Xiamen, China, Oct. 2011, pp. 353–356, doi: 10.1109/ASICON.2011.6157194.

[7] D. Saha and S. Sur-Kolay, "Guided GA-based multiobjective optimization of placement and assignment of TSVs in 3-D ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 8, pp. 1742–1750, Aug. 2019.

[8] Y. Zhao, C. Hao, and T. Yoshimura, "Thermal and wirelength optimization with TSV assignment for 3-D-IC," *IEEE Trans. Electron Devices*, vol. 66, no. 1, pp. 625–632, Jan. 2019.

[9] G. Huang et al., "Machine learning for electronic design automation: A survey," *ACM Trans. Design Autom. Electron. Syst. (TODAES)*, vol. 26, no. 5, pp. 1–46, 2021.

[10] D. Vashisht et al., "Placement in integrated circuits using cyclic reinforcement learning and simulated annealing," 2020, *arXiv:2011.07577*.

[11] V. B. Pawar, "Application of machine learning to physical design," M.S. thesis, School Eng., San Francisco State Univ., San Francisco, CA, USA, 2022. [Online]. Available: https://scholarworks.calstate.edu/downloads/vx021n292

[12] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[13] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] W. Guan, X. Tang, H. Lu, Y. Zhang, and Y. Zhang, "ATT-TA: A cooperative multiagent deep reinforcement learning approach for TSV assignment in 3-D ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 12, pp. 1905–1917, Dec. 2023.

[15] R. Lowe, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6379–6390.

[16] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[17] H. Mao, Z. Zhang, Z. Xiao, and Z. Gong, "Modelling the dynamic joint policy of teammates with attention multi-agent DDPG," 2018, *arXiv:1811.07029*.

[18] H. Jaeger, "Echo state network," *Scholarpedia*, vol. 2, no. 9, p. 2330, 2007.

[19] B. Schrauwen, D. Verstraeten, and J. M. V. Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *Proc. 15th Eur. Symp. Artif. Neural Netw.*, 2007, pp. 471–482.

[20] G. Tanaka et al., "Recent advances in physical reservoir computing: A review," *Neural Netw.*, vol. 115, pp. 100–123, Jul. 2019.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[22] University of Michigan. (2024).*Mcnc Benchmark Circuits*. [Online]. Available: http://vlsicad.eecs.umich.edu/BK/MCNCbench/

[23] K. Koźmiński, "Benchmarks for layout synthesis–evolution and current status," in *Proc. 28th Conf. ACM/IEEE Des. Autom. Conf. (DAC)*, San Francisco, CA, USA, Jun. 1991, pp. 265–270. [Online]. Available: https://ieeexplore.ieee.org/document/979726

[24] University of Michigan. (2024). *Gsrc Benchmark Circuits*. [Online]. Available: http://vlsicad.eecs.umich.edu/BK/GSRCbench/

[25] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," 2014, *arXiv:1409.1259*.

[26] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," in *Proc. AAAI Fall Symp. Sequential Decis. Making Intell. Agents (FS-15-06)*, Arlington, VA, USA, Nov. 2015, pp. 29–37. [Online]. Available: https://cdn.aaai.org/ocs/11673/11673-51288-1-PB.pdf

[27] S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney, "Recurrent experience replay in distributed reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–19.

[28] Y. Ma, L. Delshadtehrani, C. Demirkiran, J. L. Abellán, and A. Joshi, "TAP-2.5D: A thermally-aware chiplet placement methodology for 2.5D systems," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2021, pp. 1246–1251.

[29] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017.

[30] O. Yadan. (2019). *Hydra—A Framework for Elegantly Configuring Complex Applications*. [Online]. Available: https://github.com/facebookresearch/hydra

[31] X. Wang. (2025). *Practical Tips for Machine Learning Research and Development*. [Online]. Available: https://blog.wangxm.com/2025/02/practical-tips-for-machine-learning-research-and-development/

[32] R. Zhong et al., "Flexplanner: Flexible 3D floorplanning via deep reinforcement learning in hybrid action space with multi-modality representation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 37, 2024, pp. 49252–49278.

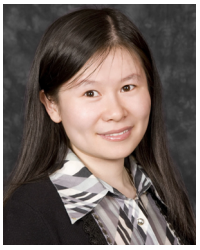
Xiaomeng Wang received the B.S. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2012, and the M.S. degree in electrical engineering from The University of Hong Kong, Hong Kong, China, in 2014. He is currently working toward the Ph.D. degree in computer engineering at the Bradley Department of Electrical and Computer Engineering and the Institute for Advanced Computing, Virginia Tech, Alexandria, VA, USA.

His research interests include machine learning for electronic design automation, VLSI design optimization, and hardware acceleration for generative AI training and inference. More information is available at www.wangxm.com

Zhen Zhou (Senior Member, IEEE) received the M.S.E.E. and Ph.D. degrees in electrical engineering from The University of Arizona, Tucson, AZ, USA, in 2006 and 2009, respectively.

She is currently a Senior Staff Research Scientist with Intel, Chandler, AZ, USA. She has authored over 45 peer-reviewed journal and conference publications and holds more than 50 patents filed with the United States Patent and Trademark Office and international authorities. Her research interests include ultrahigh-speed interconnect technologies, including wireless and optical solutions, and extreme-scale computing systems.

Dr. Zhou was a recipient of the John L. Prince Fellowship from the Semiconductor Research Corporation, IBM Innovative Research Award, and multiple Intel achievement and invention awards. She was an Invited Speaker at the International Symposium on VLSI Design Automation and Test 2017 and has been serving on the Technical Program Committee of the International Symposium on Quality Electronic Design since 2021.



Yang Yi (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2003 and 2005, respectively, and the Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX, USA, in 2009.

She is currently a Tenured Professor with the Bradley Department of Electrical and Computer Engineering and the Institute for Advanced Computing, Virginia Tech, Alexandria, VA, USA. She leads the BRain-Inspired Computing, Communication, and Security Laboratory (BRICCS@VT). Her research interests include VLSI circuits and systems, computer-aided design, neuromorphic architecture for brain-inspired computing systems, and low-power circuits design with advanced nano-technologies for high-speed wireless systems.