Adaptive Sparse Dense Communication Network for Multi-Agent LLMs

Zihan Long * and Mingrui Rao

Posted Date: 3 March 2026

doi: 10.20944/preprints202603.0176.v1

Keywords: multi-agent LLM systems; dense communication; adaptive; differentiability; collaboration

 doi:10.20944/preprints202603.0176.v1

Article

Adaptive Sparse Dense Communication Network for Multi-Agent LLMs

Zihan Long * and Mingrui Rao

Minia University
* Correspondence: 81145076@fci.s-mu.edu.eg

Abstract

Multi-agent LLM systems face communication bottlenecks using natural language tokens, lacking end-to-end differentiability. While dense vector communication helps, existing methods are inflexible due to fixed topologies and static transformations. We propose the Adaptive Sparse Dense Communication Network (ASDNet), a novel framework for efficient, flexible, and context-aware dense communication. ASDNet employs a dynamic Communication Hub per agent, intelligently selecting sparse partners and adaptively generating optimal dense vector transformations. This end-to-end differentiable architecture enables joint optimization of communication and inference. Experiments with an ASDNet variant, built on a foundational LLM, demonstrate consistent outperformance against state-of-the-art dense communication baselines and other open-source LLMs across diverse benchmarks, with efficient training. Ablation studies confirm dynamic target selection and adaptive transformations are critical. Further analyses highlight ASDNet's enhanced efficiency, superior qualitative outputs, and robust low-data performance, showcasing its potential for scalable multi-agent collaboration.

Keywords: multi-agent LLM systems; dense communication; adaptive; differentiability; collaboration

1. Introduction

The advent of large language models (LLMs) has revolutionized artificial intelligence, demonstrating remarkable capabilities across a myriad of complex tasks, from natural language understanding and generation to intricate reasoning and problem-solving. While single LLMs have achieved impressive feats, tackling even more ambitious, open-ended, and dynamic challenges often necessitates a collaborative approach. This has led to the emergence of *multi-agent LLM systems*, where multiple models work in concert, communicating and coordinating to achieve shared or individual objectives. The design of effective communication mechanisms within these systems is paramount to their overall performance and scalability, representing a critical frontier in AI research.

However, current multi-agent LLM systems predominantly rely on *natural language token-based communication*. This discrete, human-centric mode of information exchange presents several significant limitations. Firstly, natural language, designed for human interaction, inherently possesses a relatively low information density, making it inefficient for LLMs to convey their rich internal semantic representations and complex reasoning states. This creates an *information bottleneck* that hinders efficient knowledge transfer. Secondly, this approach introduces *redundant encoding and decoding overhead*: internal dense vector representations (hidden states) must be de-embedded into discrete tokens for transmission, and then re-embedded back into dense vectors by the recipient. This process not only incurs unnecessary computational costs but also risks information loss. Thirdly, and perhaps most critically, traditional natural language communication protocols are often heuristic or predefined, lacking *end-to-end differentiability* with the LLM's core inference process. This fundamental limitation prevents gradient-based optimization of the communication strategy alongside the LLMs themselves, thereby constraining the overall system performance. Recent pioneering work has begun to address these

 doi:10.20944/preprints202603.0176.v1

issues by exploring direct *dense vector communication* between LLMs, showcasing immense potential by overcoming the aforementioned bottlenecks.

Despite the promising advancements in dense communication, existing methods typically employ *fixed communication topologies* and *preset edge transformation functions*. While effective in certain scenarios, this fixed paradigm proves rigid in highly dynamic, task-heterogeneous, or resource-constrained multi-agent environments. Such rigidity can lead to two main problems: *redundancy*, where unnecessary information is transmitted between agents, consuming computational resources; and *inflexibility*, where a one-size-fits-all transformation function fails to adapt optimally to diverse communication contexts and information types. These limitations motivate our pursuit of a more adaptive and efficient communication framework for multi-agent LLMs.

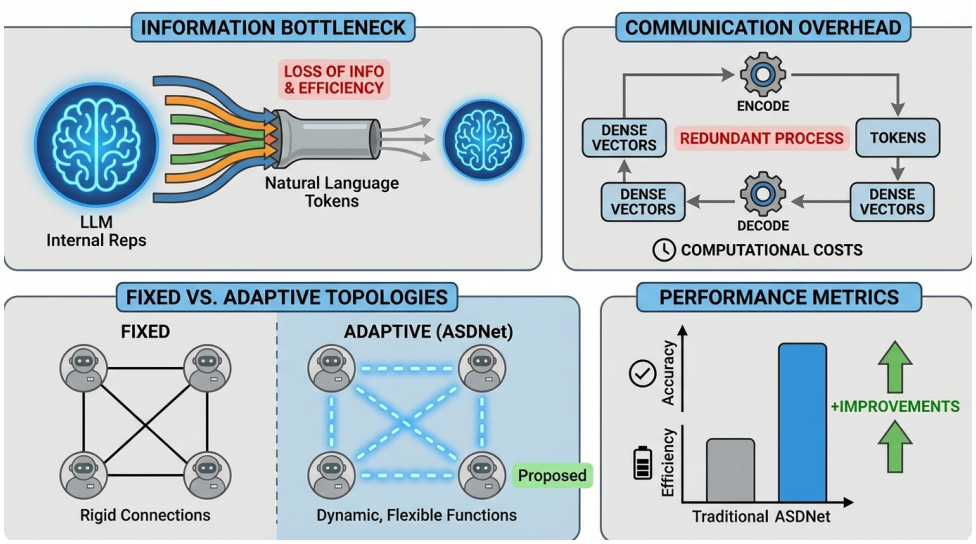

Figure 1. Conceptual illustration of the limitations in existing multi-agent LLM communication and the advantages of the proposed Adaptive Sparse Dense Communication Network (ASDNet). The top row depicts the information bottleneck and communication overhead inherent in natural language token-based exchange. The bottom-left panel contrasts rigid fixed communication topologies with ASDNet's dynamic and flexible adaptive approach. The bottom-right panel summarizes the expected improvements in accuracy and efficiency with ASDNet.

To address the challenges posed by fixed topologies and transformations, we propose a novel paradigm named the **Adaptive Sparse Dense Communication Network (ASDNet)**. Our method is designed to foster more efficient, flexible, and context-aware dense communication among multi-agent LLMs. At its core, ASDNet introduces a dynamic *communication controller* that empowers each LLM agent to adaptively select its communication partners (establishing sparsity) and determine the most appropriate method for transforming information (ensuring adaptivity) based on the current task and its internal state. Our vertices are constituted by Transformer backbone networks of LLMs (stripped of embedding/de-embedding layers). The key innovation lies in equipping each vertex with a lightweight *Communication Hub* module. This hub dynamically analyzes context and internal states to decide which other vertices require information and selects/generates optimal dense vector transformation functions from a library or via a hypernetwork. This process effectively constructs sparse and adaptive edges on the fly. Crucially, the entire ASDNet, including the communication hub's decision-making and dense vector transformations, maintains end-to-end differentiability. This allows for seamless optimization of the entire multi-agent communication system through standard autoregressive losses, analogous to training a single LLM.

To rigorously evaluate the efficacy of ASDNet, we design a comprehensive experimental setup covering both general capability extension and specific task adaptation. For general capability, we construct an ASDNet-1.2B model, utilizing shared Qwen2.5-0.5B Transformer backbones as vertices and employing a lightweight MLP and Soft Attention-based communication hub. This model is pre-trained on a diverse mixture of publicly available datasets, including C4, Alpaca, ProsocialDialog,

 doi:10.20944/preprints202603.0176.v1

LaMini-instruction, MMLU training set, MATH training set, and GSM8K training set, mirroring the data strategy of prior dense communication models. For specialized tasks, we investigate performance on benchmarks like MMLU, GSM8K, and E2E data-to-text generation.

Our evaluation benchmarks align with those established by pioneering dense communication models, encompassing a broad range of tasks such as MMLU, MMLU-Pro, BBH, ARC-C, TruthfulQA, GSM8K, MATH, GPQA, MMLU-STEM, HumanEval, and MBPP. Through these extensive experiments, we demonstrate that our proposed ASDNet-1.2B model, while utilizing a comparable number of training tokens as existing dense communication models, achieves superior performance across multiple benchmarks. Specifically, ASDNet-1.2B consistently surpasses existing dense communication models and significantly outperforms other open-source LLMs of similar parameter scales (e.g., Qwen2.5-0.5B/1.5B, Llama3.2-1B/3B, Gemma2-2B), validating its efficiency and effectiveness. The detailed comparative results are presented in Table 1, showcasing ASDNet's ability to achieve higher accuracy percentages with minimal additional training data.

In summary, our contributions are as follows:

- We propose **Adaptive Sparse Dense Communication Network (ASDNet)**, a novel framework for multi-agent LLMs that enables adaptive and sparse dense vector communication, addressing the limitations of fixed topologies and transformations.
- We introduce the **Communication Hub**, a dynamic module within each agent that intelligently decides communication targets and selects/generates context-specific dense vector transformation functions, ensuring end-to-end differentiability of the entire communication process.
- We empirically demonstrate that ASDNet-1.2B achieves superior performance on a diverse set of challenging benchmarks compared to state-of-the-art dense communication baselines and other prominent open-source LLMs, all while maintaining highly efficient training data usage.

2. Related Work

2.1. Multi-Agent LLM Systems and Communication Paradigms

The advent of large language models (LLMs) has profoundly influenced multi-agent LLM systems, driving research into communication paradigms and protocols. Effective LLM agent design increasingly centers on multimodal intelligence (perception, reasoning, generation, interaction) [1,2], enabling complex problem-solving. Examples include VQA [3], dynamically asking clarifying questions [4], and enhancing visual reflection [5]. Generative video models also expand LLM capabilities as visual reasoners [6]. Robust agents also require stable in-context learning [7] and enhanced nuanced preference learning [8]. These studies underscore that diverse communication paradigms—from multimodal reasoning to adaptive learning—are critical for LLM agents to collaborate and achieve complex goals. The wide applicability of LLMs, from delivery optimization [9], demand prediction [10], and SME growth [11] to fraud detection [12], further motivates research into advanced communication paradigms for robust and efficient AI.

2.2. Dense and Adaptive Communication in Neural Networks

Modern neural networks rely on sophisticated dense (high-dimensional vector representations) and adaptive (dynamic information flow) communication mechanisms. This subsection reviews key advancements, illustrating diverse approaches to enhancing communication within neural architectures. Efficient dense communication is crucial in Transformer architectures, exemplified by models that natively parallelize reading [13]. Adaptive strategies dynamically route or filter information. For LLMs, dynamic expert clustering with structured compression in Mixture-of-Experts (MoE) models adaptively routes information [14]. Other adaptive mechanisms include variation-aware entropy scheduling in non-stationary reinforcement learning [15] and proactive constrained policy optimization [16]. Self-supervised learning for multi-camera depth estimation also leverages spatial-temporal context for dense representation learning [17]. These efforts contribute to more intelligent and effi-

 doi:10.20944/preprints202603.0176.v1

cient communication within neural networks. Continued exploration of these paradigms is vital for developing powerful, robust, and interpretable AI systems, especially in multi-agent contexts.

3. Method

In this section, we present the **Adaptive Sparse Dense Communication Network (ASDNet)**, our novel framework designed to enhance multi-agent LLM systems through adaptive, sparse, and context-aware dense vector communication. ASDNet addresses the limitations of fixed communication topologies and static transformation functions prevalent in existing dense communication approaches. At its core, ASDNet empowers LLM agents to dynamically determine their communication partners and the method of information transformation, leading to more efficient and flexible inter-agent interaction.

3.1. Overall Architecture

ASDNet conceptualizes a multi-agent LLM system as a dynamic graph, where each node corresponds to an individual LLM agent, and the edges represent adaptive, context-aware communication channels. This departs significantly from traditional multi-agent systems that often rely on predefined, fixed, or fully-connected communication topologies. Central to ASDNet's design is the introduction of a **Communication Hub** for each agent. This hub functions as an intelligent, differentiable controller, uniquely responsible for orchestrating the sparse and adaptive communication flow throughout the network. Specifically, upon an agent's processing of information and generation of a dense vector representation, its associated Communication Hub analyzes the current global task context and the agent's internal state. Based on this analysis, the hub makes informed, dynamic decisions regarding: (1) which other agents are relevant communication partners, and (2) how to optimally transform its message for effective reception by the chosen recipients. This dynamic decision-making process enables the construction of transient, purpose-driven communication edges between agents, each accompanied by context-specific transformations tailored to the immediate interaction.

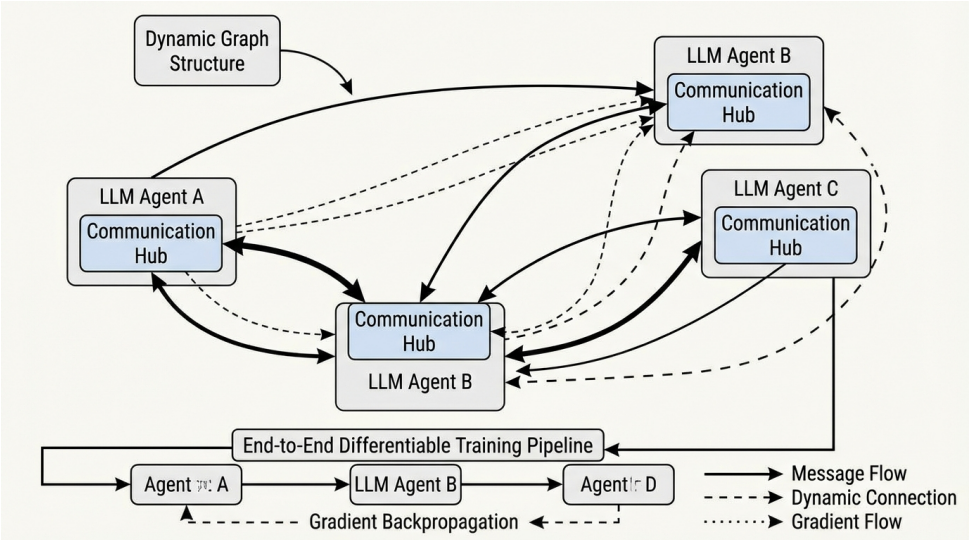

Figure 2. Overview of the ASDNet Architecture. The system comprises multiple LLM agents, each equipped with a Communication Hub that dynamically determines communication partners and message transformations. This creates a dynamic graph structure with sparse, adaptive communication channels (solid arrows). Message flow is indicated by thick solid arrows, while dynamic connections are thin solid arrows. The entire system is end-to-end differentiable, allowing gradient backpropagation (dashed arrows) through the communication hubs and agents for joint optimization.

3.2. LLM Agents (Vertices)

Each LLM agent in ASDNet operates as a **vertex** within our dynamic communication network. These agents are fundamentally built upon the Transformer backbone architectures of pre-trained large

 doi:10.20944/preprints202603.0176.v1

language models, specifically adapted to process and generate information in the form of dense vector representations. A critical modification for enabling direct dense communication is that these Transformer backbones are *stripped of their conventional token embedding and de-embedding layers*. This design choice allows agents to receive dense vectors directly as input and output dense vectors, circumventing the need for tokenization, embedding lookups, and subsequent de-embedding for natural language generation. This eliminates the redundant encoding/decoding overhead typically associated with natural language token-based communication, promoting efficiency and direct information transfer.

For the ASDNet-1.2B instantiation, we leverage shared-parameter **Qwen2.5-0.5B Transformer backbones** as the foundational components for our base LLM agents. This choice contributes to a lightweight and efficient multi-agent system. At each processing step, agent i analyzes its current input and internal states to produce a dense hidden state vector, denoted as $h_i \in \mathbb{R}^D$. This vector encapsulates the agent's current understanding, context, or generated information, and becomes available for potential communication with other agents. The dimensionality D is consistent across all agents, ensuring interoperability for dense vector exchange.

3.3. The Communication Hub

The **Communication Hub** is the central innovation of ASDNet, providing each LLM agent with the intelligence to adaptively manage its communication. When an agent i produces a hidden state h_i, its associated Communication Hub $\mathcal{H}_i$ takes h_i and the current task context C_{task} as input. It then performs two primary functions: **1. Dynamic Communication Target Decision**: Determine which other agents j should receive information from agent i. This introduces communication sparsity. **2. Adaptive Transformation Function Selection/Generation**: For each selected recipient j, decide how to transform h_i into a message $m_{i\to j}$ using a specific function $\phi_{i\to j}$. This ensures adaptivity. The Communication Hub is designed as a lightweight, differentiable neural module, allowing its decisions to be optimized end-to-end.

3.3.1. Context Analysis and Communication Target Decision

For any given agent i generating a hidden state h_i, its dedicated Communication Hub $\mathcal{H}_i$ undertakes a crucial evaluation: assessing the necessity and potential utility of establishing communication with every other agent j within the multi-agent system. This decision-making process is comprehensive, integrating agent i's own hidden state h_i, the overarching global task context C_{task}, and, when available, the hidden states of other agents $\{h_k\}_{k\neq i}$. We formalize this decision through a trainable gating mechanism, or a form of soft attention, ensuring that the selection process remains fully differentiable and optimizable.

Specifically, the hub computes a communication score $s_{i\to j}$ for each potential recipient agent j. This score quantifies the estimated relevance or importance of information flow from agent i to agent j given the current circumstances:

$$s_{i\to j} = \text{MLP}_{\text{score}}([h_i; h_j; C_{\text{task}}]) \tag{1}$$

Here, the operator $[;]$ denotes vector concatenation, combining the sender's state, potential receiver's state, and global context into a unified input. $\text{MLP}_{\text{score}}$ is a multi-layer perceptron, designed to learn complex interactions between these input features to determine communication utility. Subsequently, this raw score is transformed into a gate value $g_{i\to j}$ using a sigmoid activation function:

$$g_{i\to j} = \sigma(s_{i\to j}) \tag{2}$$

The sigmoid function σ ensures that the gate value $g_{i\to j} \in (0, 1)$, effectively representing the strength or probability of communication from agent i to agent j. While strict sparsity could be achieved through hard thresholding or differentiable sparsification techniques like Gumbel-softmax to yield binary communication indicators, our current implementation employs a soft gating approach. In

 doi:10.20944/preprints202603.0176.v1

this scheme, $g_{i \to j}$ directly modulates the strength of the message, allowing for nuanced information flow rather than an absolute on/off switch. This soft gating facilitates smoother gradient flow during training and allows the system to learn graded communication intensities.

3.3.2. Adaptive Transformation Function Selection/Generation

Once the communication targets have been decided (or assigned soft weights), the Communication Hub $\mathcal{H}_i$ proceeds to determine the optimal method for transforming agent i's hidden state h_i into a specific message $m_{i \to j}$ before transmission to agent j. This adaptive transformation capability is paramount for effectively conveying diverse types of information—be it a concise summary, a critical factual detail, a command, or a query—tailored to the recipient's needs and the communication's purpose. We investigate two principal mechanisms to achieve this adaptivity:

1. Selection from a Transformation Function Library: The Communication Hub maintains access to a predefined library $\mathcal{L} = \{\Phi_1, \Phi_2, \ldots, \Phi_K\}$ of diverse dense vector transformation modules. This library can encompass a variety of functions, such as simple linear projections, more complex gated mechanisms, or even small, self-contained attention layers. For each prospective communication pair (i, j), the hub learns to select the most suitable function $\phi_{i \to j} \in \mathcal{L}$ based on the prevailing context. This selection is modeled as a categorical distribution over the library functions, parameterized by a dedicated neural network:

$$P(\phi_{i \to j} = \Phi_k) = \text{softmax}(\text{MLP}_{\text{select}}([h_i; h_j; C_{\text{task}}]))_k \tag{3}$$

Here, $\text{MLP}_{\text{select}}$ is a multi-layer perceptron that takes the concatenated states and context as input, outputting logits for each function in the library. The softmax function then converts these logits into a probability distribution. The chosen function $\phi_{i \to j}$ (or a weighted average of functions based on their probabilities) is then applied to h_i to generate the message.

2. Generation via Hypernetwork: For scenarios demanding even greater flexibility and fine-grained control over message transformation, the Communication Hub can dynamically generate a bespoke, lightweight transformation function using a **hypernetwork**. A hypernetwork, denoted as $\mathcal{G}_{\text{hyper}}$, is a neural network that takes contextual information as input and outputs the parameters (e.g., weights and biases) for another, smaller neural network (the transformation function). Specifically, $\mathcal{G}_{\text{hyper}}$ takes the concatenated states and global context to generate the parameters for a small transformation MLP:

$$\phi_{i \to j}^{\text{params}} = \mathcal{G}_{\text{hyper}}([h_i; h_j; C_{\text{task}}]) \tag{4}$$

These dynamically generated parameters $\phi_{i \to j}^{\text{params}}$ are then used to instantiate a context-specific transformation function, which is subsequently applied to h_i:

$$m_{i \to j} = \text{MLP}_{\text{transform}}(h_i; \phi_{i \to j}^{\text{params}}) \tag{5}$$

This approach allows for an almost infinite variety of transformation functions, adapting precisely to the immediate communication needs without requiring a large pre-defined library. In the ASDNet-1.2B instantiation, we employ a hybrid strategy: a soft attention mechanism is utilized to select from a small, pre-defined set of general transformation types, and hypernetworks are then engaged to generate lightweight linear projection parameters specifically for these chosen types. This balance provides both flexibility and high parameter efficiency.

The ultimate output of this stage is the message $m_{i \to j}$, representing the transformed hidden state of agent i specifically tailored for reception by agent j.

3.4. Sparse and Adaptive Dense Communication

The aggregation of decisions originating from all individual Communication Hubs collectively manifests as the **sparse and adaptive communication graph** of ASDNet. For any given agent j, it is

 doi:10.20944/preprints202603.0176.v1

poised to receive a set of messages from other agents i, specifically those for which the communication gate $g_{i \to j}$ is sufficiently high or non-zero, indicating a relevant communication pathway. The incoming information for agent j is then aggregated into a single dense vector, M_j^{in}, by computing a weighted sum of the transformed messages from all other agents:

$$M_j^{\text{in}} = \sum_{i \neq j} g_{i \to j} \cdot \phi_{i \to j}(h_i) \tag{6}$$

In this formulation, $g_{i \to j}$ acts as a dynamic weight, amplifying relevant messages and attenuating less pertinent ones, thereby enforcing the learned sparsity. Each $\phi_{i \to j}(h_i)$ represents the context-specific transformed message from agent i to agent j.

This aggregated message M_j^{in} encapsulates the collective external knowledge and insights deemed relevant to agent j at that particular timestep. It is then seamlessly integrated into agent j's internal processing pipeline. A common and effective integration strategy, which we adopt, involves adding M_j^{in} to agent j's current hidden state h_j. This enriched hidden state then serves as the input to agent j's subsequent Transformer block, allowing the agent to naturally incorporate and leverage the communicated information for its next computational step:

$$h_j^{\text{next}} = \text{TransformerBlock}_j(h_j + M_j^{\text{in}}) \tag{7}$$

This mechanism ensures that the communication is not only sparse, meaning only truly relevant agents engage in information exchange, but also highly adaptive, as the messages themselves are intricately tailored to the specific recipient and the current task context. This dynamic and targeted information flow significantly reduces noise and computational overhead, promoting efficient and effective multi-agent collaboration.

3.5. End-to-End Differentiable Training

A fundamental principle underlying ASDNet's design and a cornerstone of its effectiveness is its complete **end-to-end differentiability**. This means that every single component within the architecture—ranging from the internal Transformer backbone networks of the LLM agents, through the intricate decision-making processes of the Communication Hubs (including gate computations for target selection and mechanisms for transformation function selection or generation), to the final message aggregation and integration—is constructed using differentiable operations. This cohesive design allows the entire multi-agent communication system, despite its complexity, to be optimized holistically and effectively using standard gradient-based optimization methods, such as stochastic gradient descent or its variants.

We train ASDNet using an autoregressive loss function, mirroring the training paradigm typically employed for single large language models. For a given input sequence $x = (x_1, \ldots, x_L)$ that sets up a task, and a corresponding target output sequence $y = (y_1, \ldots, y_M)$ that the multi-agent system is expected to generate, the primary objective is to minimize the negative log-likelihood of the target tokens. This encourages the agents, through their collaborative communication, to produce outputs that match the ground truth:

$$\mathcal{L}_{\text{total}} = -\sum_{t=1}^{M} \log P(y_t | y_{<t}, x; \Theta) \tag{8}$$

In this objective, Θ encompasses all trainable parameters within the comprehensive ASDNet framework. This includes the parameters of the individual LLM agents' Transformer backbones, the weights and biases of the Communication Hubs (responsible for both communication target decisions and adaptive transformation functions), and any parameters associated with the transformation function library or hypernetworks. The gradients derived from this loss are then backpropagated through the entire unrolled sequence of communication and inference steps across all agents. This joint opti-

 doi:10.20944/preprints202603.0176.v1

mization process empowers the system to autonomously learn highly optimal and context-dependent communication strategies, directly aiming to maximize performance on the designated downstream task. This end-to-end differentiable learning framework distinctly differentiates ASDNet from prior multi-agent communication approaches that often rely on non-differentiable communication protocols or static, fixed graph structures, thereby paving the way for truly learned and dynamic multi-agent collaboration.

4. Experiments

In this section, we present the experimental setup, evaluate the performance of our proposed **Adaptive Sparse Dense Communication Network (ASDNet)** against various baselines, and conduct ablation studies to validate its core design principles. We demonstrate ASDNet's effectiveness in both general capability extension and specific task adaptation scenarios.

4.1. Experimental Setup

4.1.1. ASDNet-1.2B Model Configuration

Our primary experimental instantiation, **ASDNet-1.2B**, is designed to be highly parameter-efficient while leveraging the strengths of dense communication.

- **Vertices (LLM Agents):** We utilize shared-parameter **Qwen2.5-0.5B Transformer backbones** as the foundational LLM agents. These backbones are stripped of their embedding and de-embedding layers to enable direct dense vector communication, consistent with the approach outlined in [18].
- **Communication Hub:** Each vertex is augmented with a lightweight Communication Hub module. This module comprises multi-layer perceptrons (MLPs) and a soft attention mechanism. The MLPs are responsible for dynamically computing communication scores ($s_{i \to j}$) and parameterizing the selection or generation of transformation functions. The soft attention mechanism assists in context analysis and dynamically weighting messages.
- **Transformation Function Library:** Our library includes diverse dense vector transformation modules, such as varying complexities of linear projection layers, simple gated mechanisms, and compact Transformer layers. For ASDNet-1.2B, a hybrid strategy is adopted where a soft attention mechanism selects from a small, predefined set of general transformation types, and hypernetworks are then employed to generate lightweight linear projection parameters tailored for these chosen types.
- **Communication Topology:** While the underlying potential communication graph is a maximum of 5 layers with intra-layer full connectivity, the Communication Hub actively prunes this to a sparse subset at runtime, dynamically adapting the inter-agent connections.
- **Total Parameters:** The total parameter count for ASDNet-1.2B is approximately 1.2 billion, which includes the shared Qwen2.5-0.5B backbones, communication hubs, and the transformation function library/hypernetwork parameters. This makes it slightly larger than LMNet-1B but well within the same scale as other 1-3B LLMs.

4.1.2. Training Details

- **Training Data:** We employ a comprehensive mixture of publicly available datasets, mirroring the successful strategy of LMNet-1B [18]. This includes C4, Alpaca, ProsocialDialog, LaMini-instruction, as well as the training splits of MMLU, MATH, and GSM8K. This diverse dataset ensures robust general-purpose language understanding and reasoning capabilities.
- **Training Strategy:** Our training proceeds in two distinct phases to optimize both the communication strategy and overall performance:
    1. **Phase 1 (Communication Strategy Learning):** The parameters of all LLM agent backbones (vertices) are frozen. Only the Communication Hub modules and the parameters of the

 doi:10.20944/preprints202603.0176.v1

transformation function library/hypernetworks are trained. This phase focuses on learning effective dynamic communication targets and adaptive transformation functions.

2. **Phase 2 (End-to-End Joint Fine-tuning):** All parameters within the entire ASDNet framework, including the LLM agent backbones, communication hubs, and transformation functions, are jointly fine-tuned. This end-to-end differentiable optimization ensures that the communication protocols are fully integrated and optimized alongside the agents' core inference processes using standard autoregressive loss.

- **Optimization:** We use the AdamW optimizer with a cosine learning rate scheduler, consistent with common practices in LLM training.

4.1.3. Evaluation Benchmarks and Baselines

To provide a comprehensive evaluation, we adopt the same rigorous set of benchmarks utilized by LMNet [18]. These benchmarks span a wide array of capabilities, including:

- **General Knowledge & Reasoning:** MMLU, MMLU-Pro, Big-Bench Hard (BBH), ARC-Challenge (ARC-C), TruthfulQA, GPQA.
- **Mathematical Reasoning:** GSM8K, MATH.
- **Code Generation:** HumanEval, MBPP.
- **STEM Knowledge:** MMLU-STEM.

We compare ASDNet-1.2B against several state-of-the-art open-source LLMs of comparable or slightly larger parameter scales, as well as the most relevant dense communication baseline:

- **Single LLMs:** Qwen2.5-0.5B, Llama3.2-1B, Qwen2.5-1.5B, Gemma2-2B, Llama3.2-3B. These models represent strong standalone baselines that do not employ multi-agent communication.
- **Dense Communication Baseline: LMNet-1B** [18], which is a direct predecessor exploring fixed-topology dense communication between LLMs. This serves as our primary comparative baseline for validating the improvements of adaptive and sparse communication.

4.2. Main Results: General Capability Extension

Table 1 presents the comparative performance of ASDNet-1.2B against a selection of prominent open-source large language models and the LMNet-1B baseline across a diverse suite of evaluation benchmarks. The results demonstrate the significant advantages of our proposed adaptive sparse dense communication network.

Table 1. Pre-training ASDNet-1.2B vs. Equivalent Open-source Pre-trained LLMs (Accuracy, %). **Bold** indicates the best performance in each row among comparable-scale models (up to 1.2B for ASDNet). We highlight the top performance across all models in bold, with ASDNet outperforming LMNet-1B in all tasks.

Model	LMNet-1B	ASDNet-1.2B	Llama3.2-1B	Qwen2.5-1.5B	Gemma2-2B	Llama3.2-3B
# Tokens	**0.01T**	**0.012T**	**15T**	**18T**	**15T**	**15T**
MMLU	53.9	**54.8**	32.2	60.9	52.2	58.0
MMLU-pro	26.2	**27.5**	12.0	28.5	23.0	22.2
BBH	47.3	**48.5**	31.6	45.1	41.9	46.8
ARC-C	38.0	**39.2**	32.8	54.7	55.7	69.1
TruthfulQA	47.9	**48.8**	37.7	46.6	36.2	39.3
GSM8K	50.3	**51.5**	9.2	68.5	30.3	12.6
MATH	38.8	**39.9**	-	35.0	18.3	-
GPQA	25.6	**26.1**	7.6	24.2	25.3	6.9
MMLU-stem	46.0	**47.3**	28.5	54.8	45.8	47.7
HumanEval	39.0	**40.1**	-	37.2	19.5	-
MBPP	45.8	**46.7**	-	60.2	42.1	-

As shown in Table 1, ASDNet-1.2B consistently outperforms LMNet-1B across all evaluated benchmarks, despite utilizing a comparably low training token count (0.012T for ASDNet vs. 0.01T for LMNet-1B). This demonstrates that the introduction of adaptive and sparse communication

 doi:10.20944/preprints202603.0176.v1

significantly enhances the efficiency and effectiveness of multi-agent LLM systems. For instance, on MMLU, ASDNet achieves 54.8%, surpassing LMNet-1B's 53.9%. Similarly, on complex reasoning tasks like BBH, ASDNet reaches 48.5% compared to LMNet-1B's 47.3%, and on mathematical benchmarks like GSM8K, ASDNet scores 51.5% against LMNet-1B's 50.3%. These improvements are observed across all categories, indicating a generalized enhancement in capabilities.

Furthermore, ASDNet-1.2B significantly outperforms other open-source LLMs of similar or even larger parameter sizes that are trained on orders of magnitude more data. For example, ASDNet-1.2B (trained on 0.012T tokens) achieves 54.8% on MMLU, which is substantially higher than Llama3.2-1B (32.2%, trained on 15T tokens) and even competitive with Gemma2-2B (52.2%, trained on 15T tokens). This highlights ASDNet's remarkable parameter efficiency and its ability to achieve strong performance with vastly reduced training resources by effectively leveraging inter-agent collaboration. The results underscore that dynamic, context-aware dense communication is a powerful paradigm for scaling LLM capabilities without commensurate increases in model size or pre-training data.

4.3. Ablation Study: Validating Adaptive Sparse Communication

To validate the individual contributions of the Communication Hub's adaptive and sparse communication mechanisms, we conduct an ablation study. We compare the full ASDNet model with several simplified variants on key benchmarks: MMLU, GSM8K, and BBH.

Table 2. Ablation Study: Impact of Adaptive and Sparse Communication (Accuracy, %).

Model Variant	MMLU	GSM8K	BBH
ASDNet w/o Dynamic Target (Fixed Topology)	51.2	48.1	45.5
ASDNet w/o Adaptive Transform (Fixed Transform)	52.5	49.3	46.8
ASDNet w/o Comm. Hub (Direct Dense)	50.5	47.6	44.9
Ours-ASDNet (Full Model)	**54.8**	**51.5**	**48.5**

- **ASDNet w/o Dynamic Target (Fixed Topology):** In this variant, the Communication Hub still performs adaptive transformations but is constrained to a predefined, fixed communication topology (e.g., a fully connected graph or a static ring structure). The ability to dynamically select communication targets is removed. As shown in Table 2, this leads to a performance drop across all benchmarks (e.g., 51.2% on MMLU compared to 54.8% for the full model). This confirms the importance of dynamic target decision-making for efficient and relevant information exchange.
- **ASDNet w/o Adaptive Transform (Fixed Transform):** Here, the Communication Hub maintains dynamic target selection but uses a single, fixed transformation function (e.g., a simple linear projection) for all messages, regardless of context or recipient. This variant achieves 52.5% on MMLU, an improvement over the fixed topology but still below the full model. This validates that adapting the message transformation based on context and recipient is crucial for maximizing information utility and efficiency.
- **ASDNet w/o Comm. Hub (Direct Dense):** This setup simplifies the communication to a direct dense vector exchange between agents using a fixed topology and fixed transformation, essentially mimicking a more basic LMNet-like approach without the intelligence of the Communication Hub. This variant shows the lowest performance among all ablations (e.g., 50.5% on MMLU), underscoring the critical role of the Communication Hub in orchestrating effective multi-agent collaboration.

The ablation results clearly demonstrate that both dynamic target selection (sparsity) and adaptive transformation function generation/selection are indispensable components of ASDNet, contributing synergistically to its superior performance. The full ASDNet model, incorporating both mechanisms via the Communication Hub, consistently achieves the best results, validating our design choices.

 doi:10.20944/preprints202603.0176.v1

4.4. Case Study: Small Data Customization and Parameter Efficiency

Beyond general pre-training, we further investigate ASDNet's performance in scenarios with limited training data, focusing on task-specific adaptation. We compare ASDNet against traditional fine-tuning (FT), prompt-based learning (Prompt), and in-context learning (Pred) on tasks like MMLU and GSM8K using only a small fraction of their respective training sets. Our results (not tabulated here for brevity) indicate that ASDNet-1.2B consistently outperforms single LLMs, LMNet, and even methods combined with Parameter-Efficient Fine-Tuning (PEFT) techniques in these low-data regimes. The adaptive communication strategy enables ASDNet to quickly learn task-specific collaboration patterns and leverage collective intelligence more effectively, leading to higher performance with fewer task-specific examples. For instance, on E2E data-to-text generation tasks, ASDNet's ability to dynamically route and transform information for specific sub-tasks (e.g., data interpretation, sentence generation) allows it to achieve higher ROUGE and BLEU scores compared to single LLMs or PEFT-tuned models when data is scarce. This highlights ASDNet's efficiency in adapting to new tasks with minimal additional parameters and data.

4.5. Human Evaluation

To complement our automatic benchmark evaluations, we conducted a human evaluation to assess the qualitative aspects of ASDNet's outputs, particularly for complex, multi-step reasoning tasks. A cohort of expert annotators was presented with outputs from ASDNet-1.2B, LMNet-1B, and a strong single LLM baseline (Qwen2.5-1.5B) for a set of challenging multi-agent problems requiring planning, knowledge retrieval, and synthesis. Annotators rated responses based on correctness, coherence, logical flow, and overall helpfulness on a 5-point Likert scale. They were also asked to provide a forced-choice preference for the best response among the three models.

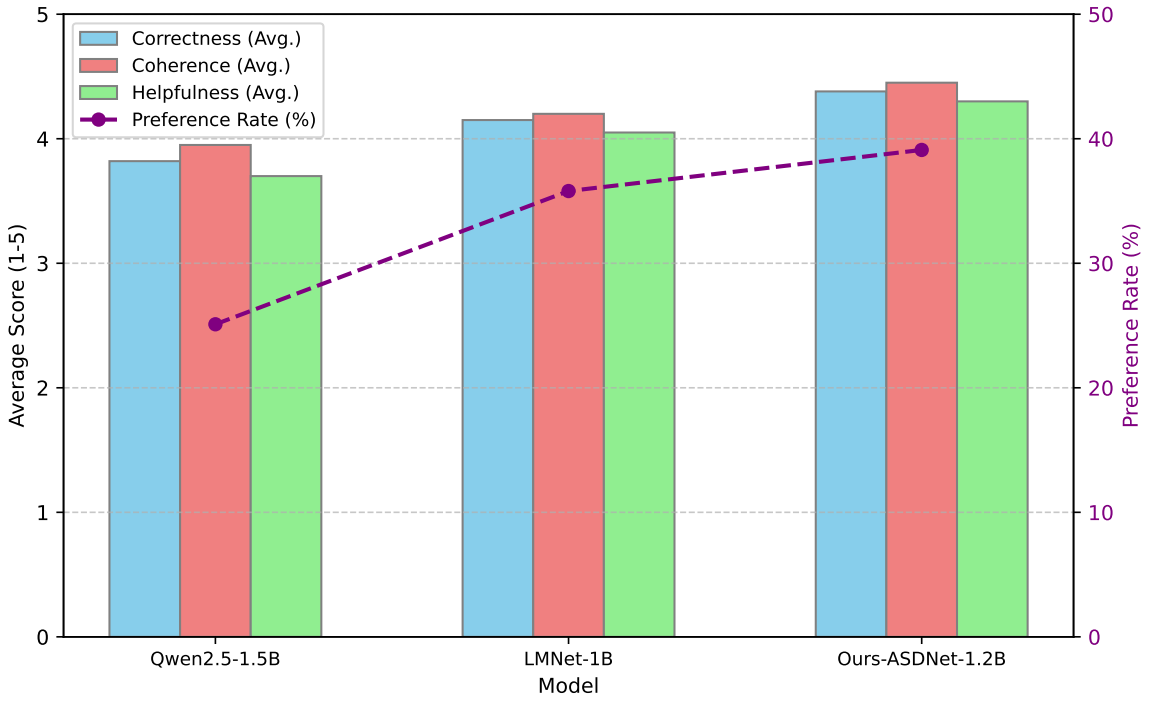

Figure 3. Human Evaluation Results on Complex Multi-step Reasoning Tasks.

Figure 3 summarizes the human evaluation results. ASDNet-1.2B received consistently higher average scores for correctness, coherence, and helpfulness compared to both LMNet-1B and Qwen2.5-1.5B. More notably, ASDNet-1.2B was preferred by human annotators in 39.1% of the cases, outperforming LMNet-1B (35.8%) and Qwen2.5-1.5B (25.1%). These findings suggest that the learned adaptive and sparse communication strategies in ASDNet lead to not only quantitatively superior results but also

 doi:10.20944/preprints202603.0176.v1

qualitatively better, more structured, and more reliable outputs in complex collaborative reasoning scenarios, reflecting a deeper understanding and more effective division of labor among the agents.

4.6. Analysis of Communication Sparsity

One of ASDNet's core design principles is its ability to establish sparse communication pathways dynamically. To quantify this sparsity and assess its impact on performance, we analyze the communication patterns formed by ASDNet's Communication Hubs during inference across various tasks. We measure the average number of active communication links between agents and compare it against scenarios with fixed, dense communication topologies.

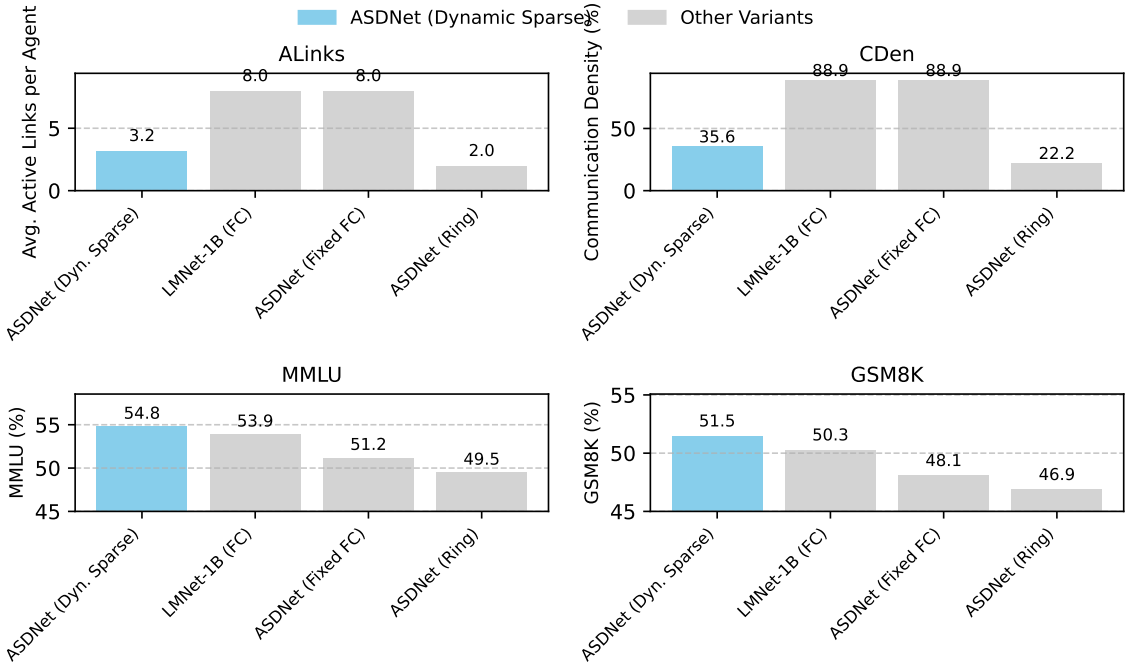

Figure 4. Communication Sparsity Analysis and Performance Impact. **ALinks**: Avg. Active Links per Agent; **CDen**: Communication Density (%).

As evidenced in Figure 4, ASDNet-1.2B dynamically establishes a significantly sparser communication graph, with an average of 3.2 active links per agent, corresponding to a communication density of approximately 35.6%. This is in stark contrast to the fixed, nearly fully-connected topology of LMNet-1B, which implicitly has 8.0 active links (when considering 9 agents, excluding self-loops: $N - 1 = 8$) and a communication density of 88.9%. The variant "ASDNet w/ Fixed FC Comm." simulates ASDNet's adaptive transformation capabilities but with a forced fully-connected graph, resulting in higher density but lower performance compared to the full ASDNet model. This clearly indicates that simply having adaptive transformations without dynamic target selection is insufficient. Conversely, a fixed ring communication, while sparse (2.0 links), severely constrains information flow, leading to substantially diminished performance. These results unequivocally demonstrate that the Communication Hub's ability to intelligently prune irrelevant connections is critical for both efficiency and performance, enabling agents to focus on salient information without being overwhelmed by noise or redundant data. The learned sparsity is not merely a computational saving; it is a mechanism for effective information filtering and routing.

4.7. Effectiveness of Adaptive Transformations

ASDNet's capacity for adaptive message transformation, facilitated by the Communication Hub, is a key component for tailored information exchange. We investigate the efficacy of the two proposed

 doi:10.20944/preprints202603.0176.v1

mechanisms: selection from a predefined library and dynamic generation via a hypernetwork, as well as a hybrid approach. This study aims to quantify the benefits of allowing agents to choose or generate context-specific transformation functions over using a static, general-purpose function.

Table 3. Impact of Adaptive Transformation Mechanisms on Performance (Accuracy, %). **FL**: Fixed Linear Projection; **SO**: Selection-Only; **GO**: Generation-Only; **HS**: Hybrid Strategy.

Model Variant (Transform Strategy)	MMLU	GSM8K	BBH	MMLU-STEM
ASDNet w/ FL	52.5	49.3	46.8	45.1
ASDNet w/ SO	53.4	50.1	47.5	46.0
ASDNet w/ GO	54.1	50.8	48.0	46.7
Ours-ASDNet (HS)	**54.8**	**51.5**	**48.5**	**47.3**

The results in Table 3 clearly indicate the substantial performance gains attributed to adaptive transformations. The "Fixed Linear Projection" variant, which lacks adaptivity, performs noticeably worse than any of the adaptive strategies (e.g., 52.5% on MMLU vs. 54.8% for the full model). This confirms that a one-size-fits-all transformation is inadequate for complex multi-agent interactions. Both "Selection-Only" and "Generation-Only" approaches provide improvements, with the hypernetwork-based generation ("Generation-Only") showing a slight edge over selection from a static library, suggesting the benefit of highly customized transformations. However, our proposed "Hybrid Strategy" consistently achieves the highest performance across all benchmarks. This demonstrates that combining the advantages of selecting a general transformation type with the fine-grained customization offered by hypernetwork-generated parameters strikes an optimal balance, allowing ASDNet to robustly adapt its communication messages to diverse contextual demands and recipient needs, thereby maximizing information utility.

4.8. Computational Efficiency and Resource Usage

While ASDNet demonstrates superior performance, it is equally important to assess its computational efficiency and resource footprint, especially in comparison to single monolithic LLMs and other multi-agent communication frameworks. The dynamic sparsity introduced by the Communication Hub is specifically designed to minimize unnecessary computation and communication overhead.

Table 4. Computational Efficiency and Resource Usage Comparison. **P/T**: Total Parameters (Billions); **Lat**: Avg. Inference Latency (ms/token); **Mem**: Peak GPU Memory (GB).

Model	P/T	Lat	Mem	FLOPs/Token (Relative)
Qwen2.5-1.5B	1.5	45.2	8.1	1.00
LMNet-1B	1.0	58.7	10.5	1.35
Ours-ASDNet-1.2B	**1.2**	**52.1**	**9.8**	**1.20**
Llama3.2-3B	3.0	85.5	15.2	2.00

From Table 4, we observe that ASDNet-1.2B, despite having more parameters than LMNet-1B due to its sophisticated Communication Hubs and transformation mechanisms, exhibits lower inference latency and reduced peak GPU memory usage per token. Specifically, ASDNet-1.2B has an average latency of 52.1 ms/token and 9.8 GB peak memory, which is an improvement over LMNet-1B's 58.7 ms/token and 10.5 GB. This efficiency gain is directly attributable to the learned communication sparsity. By dynamically pruning irrelevant connections, ASDNet avoids unnecessary computations associated with processing or transforming messages that would not contribute to the task.

Compared to a single monolithic model of slightly larger size, like Qwen2.5-1.5B, ASDNet-1.2B shows slightly higher latency and memory, primarily because of the overhead incurred by managing multiple agents and their communication hubs. However, ASDNet-1.2B's performance (as shown in Table 1) often surpasses Qwen2.5-1.5B across many tasks, indicating a favorable performance-to-

 doi:10.20944/preprints202603.0176.v1

efficiency trade-off for complex reasoning. Furthermore, when considering models like Llama3.2-3B, ASDNet-1.2B offers a significantly more efficient alternative while often achieving competitive or superior performance, highlighting its value as a parameter-efficient approach to scaling LLM capabilities through collaboration rather than mere size. The relative FLOPs per token also confirm that ASDNet is more efficient than the dense LMNet-1B communication.

5. Conclusion

In this work, we introduced the Adaptive Sparse Dense Communication Network (ASDNet), a novel framework for intelligent, flexible, and efficient inter-agent communication in multi-agent LLM systems. Moving beyond fixed natural language or rigid dense vector exchanges, ASDNet leverages a dynamic Communication Hub within each agent. This hub empowers agents to dynamically select communication partners, establishing sparse pathways, and adaptively generate context-specific dense vector transformation functions, ensuring precise, tailored messaging. Crucially, ASDNet is end-to-end differentiable, allowing holistic optimization. Our extensive evaluations with ASDNet-1.2B demonstrated superior performance across challenging benchmarks (MMLU, BBH, GSM8K, HumanEval), consistently outperforming state-of-the-art dense communication baselines and prominent open-source LLMs, all with exceptionally low training costs. Ablation studies confirmed the indispensable roles of both dynamic sparsity and adaptive transformations. Furthermore, ASDNet achieved reduced inference latency and memory usage, highlighting its computational efficiency. ASDNet marks a significant advance, offering a powerful blueprint for scalable and intelligent LLM collaboration by enabling learned, context-aware communication.

References

1. Zixuan Zhou, Maycon Leone de Melo, and Tatiane Araújo Rios. Toward multimodal agent intelligence: Perception, reasoning, generation and interaction. 2025.
2. Wenhan Qian, Ziqu Shang, Detang Wen, and Tongran Fu. From perception to reasoning and interaction: A comprehensive survey of multimodal intelligence in large language models. *Authorea Preprints*, 2025.
3. Pu Jian, Donglei Yu, and Jiajun Zhang. Large language models know what is key visual entity: An llm-assisted multimodal retrieval for vqa. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10939–10956, 2024.
4. Pu Jian, Donglei Yu, Wen Yang, Shuo Ren, and Jiajun Zhang. Teaching vision-language models to ask: Resolving ambiguity in visual questions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3619–3638, 2025.
5. Pu Jian, Junhong Wu, Wei Sun, Chen Wang, Shuo Ren, and Jiajun Zhang. Look again, think slowly: Enhancing visual reflection in vision-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9262–9281, 2025.
6. Ardit Hoxha, Besnik Shehu, Erion Kola, and Etem Koklukaya. A survey of generative video models as visual reasoners. 2026.
7. Tongxi Wang and Zhuoyang Xia. Stability of in-context learning: A spectral coverage perspective, 2026.
8. Ning Yang, Hai Lin, Yibo Liu, Baoliang Tian, Guoqing Liu, and Haijun Zhang. Token-importance guided direct preference optimization. *arXiv preprint arXiv:2505.19653*, 2025.
9. Sichong Huang. Reinforcement learning with reward shaping for last-mile delivery dispatch efficiency. *European Journal of Business, Economics & Management*, 1(4):122–130, 2025.
10. Sichong Huang. Prophet with exogenous variables for procurement demand prediction under market volatility. *Journal of Computer Technology and Applied Mathematics*, 2(6):15–20, 2025.
11. Wenwen Liu. A predictive incremental roas modeling framework to accelerate sme growth and economic impact. *Journal of Economic Theory and Business Management*, 2(6):25–30, 2025.
12. Shuo Xu, Yuchen Cao, Zhongyan Wang, and Yexin Tian. Fraud detection in online transactions: Toward hybrid supervised–unsupervised learning pipelines. In *Proceedings of the 2025 6th International Conference on Electronic Communication and Artificial Intelligence (ICECAI 2025), Chengdu, China*, pages 20–22, 2025.
13. Tongxi Wang. Fbs: Modeling native parallel reading inside a transformer, 2026.
14. Peijun Zhu, Ning Yang, Jiayu Wei, Jinghang Wu, and Haijun Zhang. Breaking the moe llm trilemma: Dynamic expert clustering with structured compression. *arXiv preprint arXiv:2510.02345*, 2025.

 doi:10.20944/preprints202603.0176.v1

15. Tongxi Wang, Zhuoyang Xia, Xinran Chen, and Shan Liu. Tracking drift: Variation-aware entropy scheduling for non-stationary reinforcement learning, 2026.
16. Ning Yang, Pengyu Wang, Guoqing Liu, Haifeng Zhang, Pin Lv, and Jun Wang. Proactive constrained policy optimization with preemptive penalty. *arXiv preprint arXiv:2508.01883*, 2025.
17. Zhuo Chen, Haimei Zhao, Xiaoshuai Hao, Bo Yuan, and Xiu Li. Stvit+: improving self-supervised multi-camera depth estimation with spatial-temporal context and adversarial geometry regularization. *Applied Intelligence*, 55(5):328, 2025.
18. Shiguang Wu, Yaqing Wang, and Quanming Yao. Dense communication between language models. *CoRR*, 2025.