Content-Aware Adaptive Dense Communication Network: Enhancing Multi-LLM Collaboration with Dynamic Information Flow

Anthony White * and Joshua Allen

Posted Date: 31 March 2026

doi: 10.20944/preprints202603.2356.v1

Keywords: CADCN; multi-agent systems; LLMs; dense communication; adaptivity

 doi:10.20944/preprints202603.2356.v1

Article

Content-Aware Adaptive Dense Communication Network: Enhancing Multi-LLM Collaboration with Dynamic Information Flow

Anthony White * and Joshua Allen

Western Kentucky University
* Correspondence: puthip.si@st.wu.ac.th

Abstract

Multi-agent systems built upon large language models (LLMs) are hindered by the limitations of traditional token-based communication, which suffers from information bottlenecks, redundant processing, and a lack of end-to-end differentiability. While dense vector communication offers improvements, existing methods lack adaptivity to diverse task requirements. To address this, we propose the Content-Aware Adaptive Dense Communication Network (CADCN), a novel architecture that empowers LLM agents to communicate through dense vectors by dynamically perceiving content and adaptively selecting routing and transformation strategies. CADCN introduces the Content-Aware Communication Unit (CACU), which integrates Content-Aware Routing via a lightweight network and Adaptive Dense Transformation using a Mixture-of-Experts structure, ensuring full differentiability. Our experiments, conducted under a highly constrained training token budget, demonstrate that CADCN consistently achieves superior performance across diverse general knowledge, reasoning, mathematical, and coding benchmarks compared to prior dense communication approaches. Furthermore, CADCN significantly outperforms vanilla LLMs trained on vastly larger datasets, highlighting its remarkable data efficiency and capability expansion. Ablation studies confirm the synergistic contributions of CADCN's adaptive components, while analysis of communication dynamics reveals learned expert specialization. Our findings establish CADCN as a highly efficient and intelligent paradigm for robust multi-LLM collaboration.

Keywords: CADCN; multi-agent systems; LLMs; dense communication; adaptivity

1. Introduction

The remarkable advancements in large language models (LLMs) have paved the way for sophisticated multi-agent systems capable of tackling complex tasks collaboratively [1]. A pivotal challenge in developing such systems lies in establishing efficient, differentiable, and intelligent internal communication mechanisms among LLM agents. Traditionally, LLM systems primarily rely on the exchange of natural language tokens for information transfer. While intuitive, this token-based paradigm suffers from several significant limitations:

Firstly, information bottlenecks and inefficient expression are prevalent. Discrete token sequences often struggle to convey complex, multimodal, or fine-grained information efficiently, leading to potential loss of continuity and nuance. Secondly, a redundant encoding/decoding process is inherent, where hidden state embeddings are repeatedly converted into tokens and then de-embedded back into hidden states for subsequent processing. This introduces unnecessary computational overhead and further information degradation. Lastly, token-based communication protocols inherently lack end-to-end differentiability, making it challenging to optimize the entire communication strategy through gradient descent. This limitation restricts the system's ability to learn optimal collaborative behaviors and information flow.

 doi:10.20944/preprints202603.2356.v1

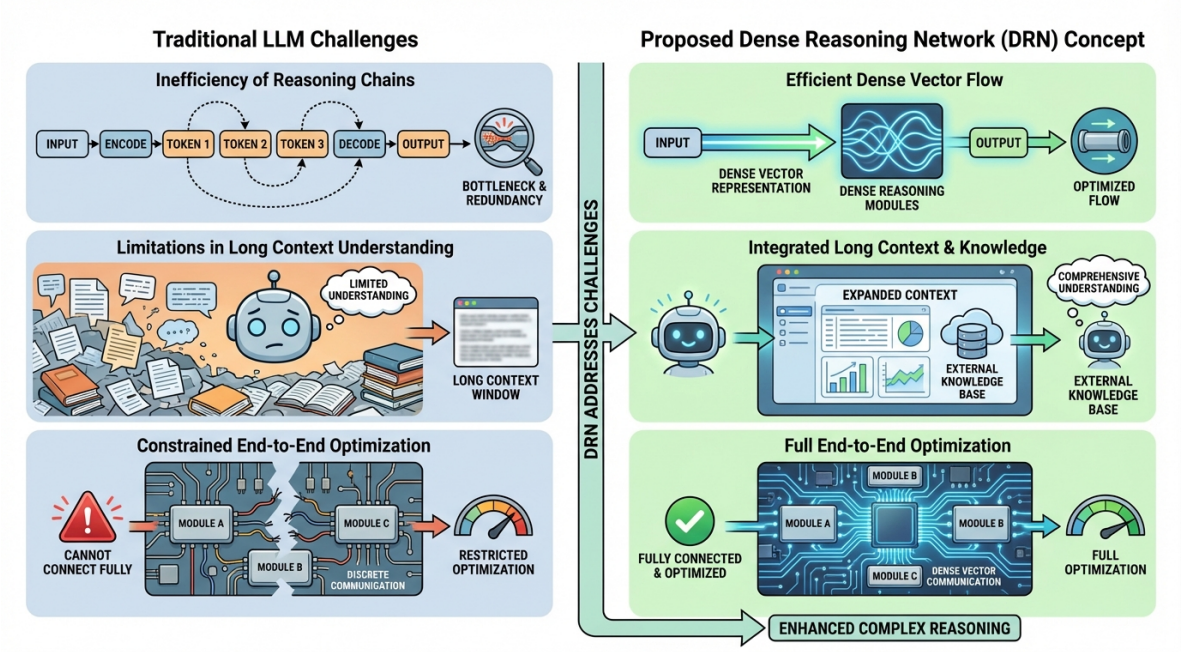

Figure 1. Comparison of Traditional LLM Challenges and the Proposed Dense Reasoning Network (DRN) Concept. The left side illustrates the limitations of traditional token-based communication, including inefficiency due to token bottlenecks and redundancy, limitations in long context understanding, and constrained end-to-end optimization. The right side depicts how dense vector communication addresses these challenges through efficient dense vector flow, integrated long context and knowledge handling, and full end-to-end optimization, enabling enhanced complex reasoning.

Recently, the pioneering work on *Dense Communication between Language Models* introduced a transformative paradigm by enabling LLMs to communicate directly through dense vectors (hidden states), leading to the development of the **LMNet** architecture. LMNet conceptualizes "stripped transformers" (without embedding/de-embedding layers) as computational nodes (vertices) and employs trainable seq2seq modules as edges to facilitate information transfer within a dense vector space. This approach elegantly ensures end-to-end differentiability, laying a crucial foundation for building more efficient and intelligent multi-LLM collaborative systems.

However, despite its innovations, LMNet's communication "edges," while trainable, employ a relatively fixed transformation logic. In the context of diverse and dynamic task scenarios, relying on predetermined seq2seq modules for dense vector conversion may not adequately capture the varying requirements for communication types based on specific information content or task stages. This inherent rigidity can limit the system's adaptability and overall efficiency. Our research is motivated by this gap and aims to significantly enhance LMNet's foundation by introducing adaptive and content-aware capabilities into dense communication. The goal is to enable communication links to intelligently select optimal routing and transformation strategies based on the current task state and the semantic content of the information being transmitted.

To address these limitations, we propose the **Content-Aware Adaptive Dense Communication Network (CADCN)**. Our method's core idea is to go beyond mere direct dense vector transfer; CADCN empowers the model to dynamically perceive the content of communication and, based on this perception, adaptively choose both information routing and transformation methods. This results in a more efficient, intelligent, and expressive internal communication system. CADCN inherits LMNet's "stripped LLM vertices" but introduces a novel **Content-Aware Communication Unit (CACU)** to replace fixed seq2seq edges. Each CACU integrates a lightweight routing network for Content-Aware Routing, which analyzes semantic content to dynamically predict which downstream vertices require the information and at what granularity. Furthermore, CACU employs an **Adaptive Dense Transformation** mechanism, leveraging a Mixture-of-Experts (MoE) structure or a hypernetwork, where

 doi:10.20944/preprints202603.2356.v1

the routing network dynamically selects or generates specialized transformation modules based on content. This dynamic architecture inherently possesses the potential for multimodal fusion, drawing inspiration from advanced methods for integrating diverse information streams [2,3], and maintains complete end-to-end differentiability, allowing for holistic optimization of the entire communication protocol.

To validate the efficacy of CADCN, we adopt the experimental setup established by the LMNet paper for direct comparison. Our base vertices are constructed using transformer blocks from **Qwen2.5-0.5B**, with parameters shared across the network. The CACUs consist of a 2-layer MLP routing network and a MoE structure with 4-8 small transformer encoder layers, ensuring the total parameter count for CADCN-1B remains comparable to LMNet-1B. We evaluate CADCN using a similar MLP topology (e.g., 5 layers: 1/4/4/4/1) to LMNet. For training, we utilize a mixed public dataset including **C4, Alpaca, ProsocialDialog, LaMini-instruction, MMLU (auxiliary_training split), MATH (training split), and GSM8K (training split)**. Crucially, we maintain a small training budget of approximately **0.01T tokens** to highlight the efficiency gains from our communication mechanism. The training strategy involves an initial phase of freezing vertex parameters and training only CACUs, followed by an end-to-end fine-tuning phase of all parameters.

Our comprehensive evaluation spans several common benchmarks, including general knowledge and reasoning tasks (MMLU, MMLU-Pro, BBH, ARC-C, TruthfulQA, GPQA) and mathematical and coding capabilities (GSM8K, MATH, MMLU-STEM, HumanEval, MBPP). As illustrated in Table 1 (in Section 4), our proposed **CADCN-1B** consistently achieves slightly superior performance across multiple key benchmarks compared to LMNet-1B, while maintaining the same stringent training token budget. Furthermore, CADCN-1B significantly outperforms other vanilla LLMs (e.g., Qwen2.5-0.5B, Llama3.2-1B) which are trained on substantially larger token budgets (15T-18T tokens), demonstrating remarkable data efficiency and capability expansion. These results underscore that by integrating content-aware and adaptive routing mechanisms, CADCN can effectively enhance the overall intelligence and task processing capabilities of dense communication networks without increasing model parameters or training efficiency.

Our primary contributions are summarized as follows:

- We propose **Content-Aware Adaptive Dense Communication Network (CADCN)**, a novel architecture that enables LLMs to communicate through dense vectors with dynamic content perception and adaptive information routing.
- We introduce the **Content-Aware Communication Unit (CACU)**, a lightweight yet powerful module incorporating Content-Aware Routing and Adaptive Dense Transformation (via MoE or Hypernetwork) to replace fixed communication edges in dense communication networks.
- We demonstrate that CADCN achieves enhanced performance over state-of-the-art dense communication methods (e.g., LMNet) and significantly outperforms vanilla LLMs on a diverse set of benchmarks, particularly under a highly constrained training token budget, highlighting its superior efficiency and intelligence.

2. Related Work

2.1. Multi-Agent LLM Systems and Dense Communication

The burgeoning field of multi-agent Large Language Model (LLM) systems holds significant promise for tackling complex tasks by distributing work and fostering collaboration among specialized agents. Central to their efficacy is communication, which can range from explicit natural language dialogue to more implicit, dense forms of information exchange. Initially, multi-agent LLM systems primarily leveraged explicit natural language dialogues, as exemplified by frameworks like ChatDev, where specialized LLM agents collaborate on software development through structured linguistic communication, showcasing the effectiveness of multi-turn dialogues for intricate problem-solving [1]. Similarly, research on cooperative AI has demonstrated that multi-agent debate among LLMs can effectively encourage divergent thinking and enhance problem-solving capabilities [4], underscoring

 doi:10.20944/preprints202603.2356.v1

the power of explicit, language-based interaction in orchestrating LLM collaboration. Complementing this, the concept of dense communication, which leverages vector representations, is crucial for efficient and nuanced information sharing among agents. Foundational work in dense retrieval, such as Condenser, introduced pre-training architectures for generating high-quality dense representations for retrieval tasks [5]. Building upon this, methods have been developed for learning dense representations of phrases at scale, enabling the encoding of semantically rich units of information into compact vectors [6]. Further advancements include unsupervised corpus-aware language model pre-training specifically tailored for dense passage retrieval, which refines the quality and relevance of dense vector communication for agents requiring detailed contextual information [7]. These works establish the technical groundwork for agents to communicate not merely with tokens but with compact, semantically rich dense vectors. Directly addressing inter-LLM communication, the LMNet architecture proposed a foundational paradigm for enabling LLMs to communicate through dense hidden states, ensuring end-to-end differentiability for collaborative systems. However, relying solely on dense representations presents its own challenges, with some research indicating that dense retrievers can struggle with simple, entity-centric questions, suggesting limitations in precisely capturing and conveying certain types of information, thereby highlighting areas for improvement in inter-LLM communication via dense methods [8]. To address more granular forms of internal information exchange, studies have explored the use of searchable hidden intermediates in end-to-end models [9], a concept of "hidden state exchange" that represents a deeper level of dense communication where internal model states facilitate more integrated and subtle forms of collaboration. Finally, ensuring the safety and controllability of LLM applications is paramount, especially as multi-agent systems grow in complexity; toolkits like NeMo Guardrails provide programmable rails for controllable and safe LLM applications [10], influencing the reliability and effectiveness of both explicit and dense communication channels by governing how agents interact within defined boundaries. In summary, the landscape of multi-agent LLM systems is evolving, moving from primarily explicit natural language interactions towards sophisticated forms of dense communication, including shared dense vector representations and hidden state exchanges, with effective orchestration and leveraging of these diverse communication forms remaining an active area of research for enhancing reasoning and task-solving in complex multi-agent LLM environments.

2.2. Adaptive Neural Architectures and Efficient Learning

The rapid growth of deep learning models, particularly in natural language processing, has underscored the critical need for both adaptive neural architectures and efficient learning paradigms. Adaptive architectures enable models to dynamically adjust their computational patterns based on input, leading to more flexible and powerful systems, while efficient learning strategies aim to reduce computational costs, memory footprint, and data requirements for training and deployment, making advanced models more accessible and sustainable. A prominent direction in adaptive architectures involves *Mixture-of-Experts (MoE)* models, which route inputs to specialized "experts" based on learned gates, as exemplified by Liu et al.'s work on controlled text generation using decoding-time expert and anti-expert mechanisms [11]. This concept aligns with the broader idea of *conditional computation*, where only relevant parts of a network are activated based on the input, which Ye et al. leveraged for efficient zero-shot learning by generating synthetic datasets tailored to specific conditions [12]. The design of flexible *Transformer architectures* also falls under this umbrella, with Zhang et al. introducing an Adaptive Language-guided Multimodal Transformer (ALMT) that learns hyper-modality representations to suppress irrelevant information in multimodal sentiment analysis [13]. Complementing these efforts, advanced neural architectures have explored integrating both local and non-local attention operations to enhance feature extraction and information processing across various domains, including speech enhancement [14]. Moreover, the principles of adaptively handling and balancing information from multiple modalities are crucial in areas like audio-visual speech processing, where techniques for integrating visual and audio cues for enhancement and separation have shown significant promise [2,3]. Similarly, in other domains requiring high-fidelity information acquisition and

 doi:10.20944/preprints202603.2356.v1

processing, such as advanced microscopy, adaptive illumination and diffractive elements have been critical for achieving super-resolution and efficient imaging without mechanical scanning, pushing the boundaries of spatial information extraction [15–17]. Furthermore, advancing *adaptive computation* requires a foundational understanding of model capabilities, to which Tay et al. contribute by comparatively analyzing pretrained convolutional and Transformer architectures for efficiency and performance characteristics [18]. To assess the adaptability and robustness of such models, particularly under evolving conditions, dynamic benchmarks are invaluable, with Potts et al. introducing DynaSent, a dynamic benchmark for sentiment analysis, which offers a challenging environment for evaluating model adaptation to changing data distributions [19]. Beyond architectural adaptability, significant research focuses on enhancing the *efficiency of learning* itself, with *parameter-efficient fine-tuning (PEFT)* emerging as a crucial area for adapting large pre-trained models with minimal computational cost and memory; Yu et al., for instance, addressed PEFT by proposing a contrastive-regularized self-training approach for fine-tuning with weak supervision, thereby improving both efficiency and performance [20]. A related strategy involves *sparse activation*, where only a subset of network parameters or activations are involved in computation; Ben Zaken et al. introduced BitFit, a simple yet effective PEFT method for Transformer-based masked language models that achieves efficiency by selectively updating only bias terms, leading to a form of sparse activation and reduced computational overhead [21]. Crucially, efficient learning also necessitates addressing challenges related to data usage and model generalization; Varis and Bojar investigated *data-efficient learning* by analyzing length-based overfitting in Transformer models, highlighting how input characteristics can impede generalization and efficient training, thereby necessitating more robust learning strategies [22]. Collectively, these works demonstrate a strong research trend toward developing neural architectures that are not only powerful but also inherently adaptable and efficient, encompassing dynamic computational paths, optimized parameter updates, and a nuanced understanding of data efficiency, all contributing to the development of more sustainable and deployable AI systems. “‘

3. Method

In this section, we introduce the **Content-Aware Adaptive Dense Communication Network (CADCN)**, our proposed architecture designed to enhance the efficiency and intelligence of inter-LLM communication by enabling dynamic content perception and adaptive information flow. CADCN builds upon the foundational concept of dense vector communication, while addressing the limitations of fixed communication channels by introducing dynamic adaptivity at the communication layer.

3.1. Overall Architecture of CADCN

CADCN conceptualizes a multi-LLM system as a directed graph, where each node represents a computational unit (an LLM agent) and the edges represent communication channels. Unlike traditional systems that rely on discrete natural language tokens for communication, or even the initial dense communication models that use fixed transformation edges, CADCN facilitates direct information exchange through continuous, **dense hidden states** (vectors). This approach inherently avoids the information loss and computational overhead associated with tokenization and detokenization. The core innovation lies in the adaptive nature of these communication channels, which are no longer static but dynamically adjust their behavior based on the semantic content of the information being transmitted. This design ensures that the entire network, including the communication protocol itself, remains **end-to-end differentiable**, allowing for holistic optimization through standard gradient-based methods.

 doi:10.20944/preprints202603.2356.v1

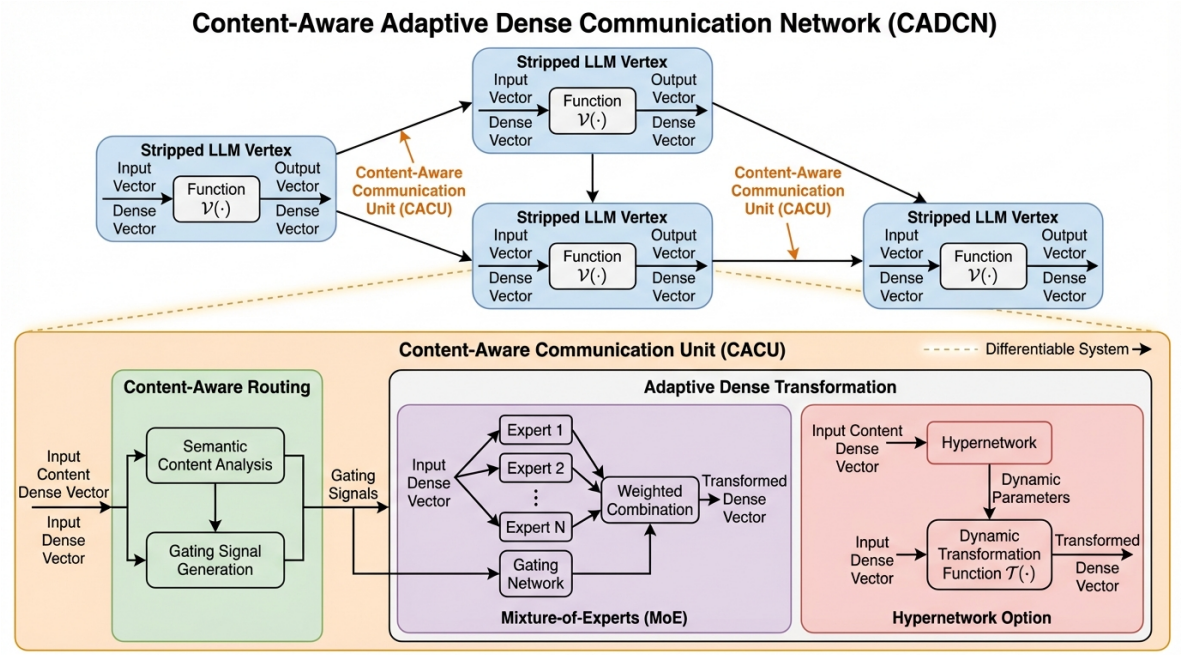

Figure 2. Overall architecture of the Content-Aware Adaptive Dense Communication Network (CADCN) and its core component, the Content-Aware Communication Unit (CACU). The upper part shows CADCN as a network of *Stripped LLM Vertices* linked by *CACUs*. The lower part details a *CACU*, comprising a *Content-Aware Routing* module that generates gating signals and an *Adaptive Dense Transformation* module. The adaptive transformation can be implemented as a *Mixture-of-Experts (MoE)* or a *Hypernetwork*, enabling content-dependent communication.

Formally, a CADCN can be represented as a computational graph $G = (V, E)$, where V is a set of vertices (stripped LLM modules) and E is a set of directed edges (Content-Aware Communication Units, CACUs). When a vertex $v_i \in V$ processes an input dense vector $\mathbf{h}_{in}^{(i)}$ and produces an output dense vector $\mathbf{h}_{out}^{(i)}$, this vector is then fed into a CACU $e_{ij} \in E$ connecting v_i to a downstream vertex v_j. The CACU dynamically processes $\mathbf{h}_{out}^{(i)}$ to generate an adapted dense vector $\mathbf{h}_{in}^{(j)}$ for v_j, as described by:

$$\mathbf{h}_{in}^{(j)} = \text{CACU}_{ij}(\mathbf{h}_{out}^{(i)}) \tag{1}$$

This contrasts with fixed transformation edges, where CACU_{ij} would be a static function, independent of the content of $\mathbf{h}_{out}^{(i)}$.

3.2. Stripped LLM Vertices

CADCN utilizes **stripped LLM vertices** as its fundamental computational nodes. Each vertex is essentially a transformer block from a pre-trained LLM, with its embedding and de-embedding layers removed. This stripping process enables direct processing and output of dense vectors, eliminating the redundant encoding/decoding steps inherent in token-based communication. By offloading the communication logic to the dedicated CACUs, these vertices can focus purely on processing information within their transformer architecture, akin to standard forward passes in a larger LLM. For our experimental setup, these vertices are constructed from the transformer modules of the **Qwen2.5-0.5B** model. Parameters for these vertices can be shared across the network to promote generalization or specialized for particular roles to enhance task-specific performance, depending on the network topology and task requirements. The operation of a vertex can be abstracted as a function $\mathcal{V}(\cdot)$ that transforms an input dense vector $\mathbf{h}_{in}$ into an output dense vector $\mathbf{h}_{out}$:

$$\mathbf{h}_{out} = \mathcal{V}(\mathbf{h}_{in}) \tag{2}$$

 doi:10.20944/preprints202603.2356.v1

This function typically encapsulates standard transformer operations such as self-attention, layer normalization, and feed-forward networks.

3.3. Content-Aware Communication Unit (CACU)

The central innovation of CADCN is the **Content-Aware Communication Unit (CACU)**, which replaces the fixed seq2seq transformation edges of previous dense communication models. A CACU is a lightweight yet powerful module designed to dynamically perceive the semantic content of an incoming dense vector and adapt its transformation strategies accordingly. Each CACU is responsible for two primary functions: **Content-Aware Routing** and **Adaptive Dense Transformation**.

3.3.1. Content-Aware Routing

Upon receiving an output dense vector $\mathbf{h}_{\text{out}}^{(i)}$ from an upstream vertex v_i, the Content-Aware Routing mechanism within the CACU first analyzes its semantic content. This analysis is performed by a dedicated, lightweight routing network, typically a multi-layer perceptron (MLP) or an attention-based module. The purpose of this network is to dynamically determine the optimal transformation strategy for the information being transmitted. Specifically, the routing network generates a set of gating signals that control how the subsequent adaptive transformation module will process the dense vector. This allows the communication channel to dynamically adjust its behavior based on the specific content and context of the information. Let $R_{\text{logits}}(\cdot)$ denote the routing network's output layer producing raw logits. For a given output vector $\mathbf{h}_{\text{out}}^{(i)}$, the routing mechanism generates gating signals $\mathbf{g} \in \mathbb{R}^M$ (where M is the number of transformation experts), such that:

$$\mathbf{g} = \text{Softmax}(R_{\text{logits}}(\mathbf{h}_{\text{out}}^{(i)})) \tag{3}$$

Here, g_m represents the activation strength or weight for the m-th expert in the adaptive transformation module, ensuring that $\sum_{m=1}^{M} g_m = 1$. This content-dependent gating enables fine-grained control over the information flow and transformation.

3.3.2. Adaptive Dense Transformation

Instead of a single, fixed transformation function, the CACU incorporates a mechanism for **Adaptive Dense Transformation**. This is primarily achieved through a **Mixture-of-Experts (MoE)** architecture or a **Hypernetwork**.

In the **Mixture-of-Experts (MoE)** setting, the CACU houses a collection of M specialized transformation modules, referred to as "experts" $(E_1, E_2, \ldots, E_M)$. Each expert is typically a small transformer encoder layer, a compact multi-layer perceptron, or a specialized projection network. The gating signals $\mathbf{g}$ produced by the Content-Aware Routing network (Equation 3) are used to dynamically combine these experts. The input dense vector $\mathbf{h}_{\text{out}}^{(i)}$ is transformed by a weighted sum of the individual expert outputs, where the weights are directly derived from $\mathbf{g}$:

$$\mathbf{h}_{in}^{(j)} = \sum_{m=1}^{M} g_m \cdot E_m(\mathbf{h}_{\text{out}}^{(i)}) \tag{4}$$

This weighted sum allows the communication channel to leverage different transformation functions tailored to the specific content, effectively creating a composite, content-adaptive transformation.

Alternatively, a **Hypernetwork** could dynamically generate the parameters θ_{trans} for a small, single transformation network $T(\cdot; \theta_{\text{trans}})$ based on the input content:

$$\theta_{\text{trans}} = H(\mathbf{h}_{\text{out}}^{(i)}) \tag{5}$$

$$\mathbf{h}_{in}^{(j)} = T(\mathbf{h}_{\text{out}}^{(i)}; \theta_{\text{trans}}) \tag{6}$$

 doi:10.20944/preprints202603.2356.v1

where $H(\cdot)$ is the hypernetwork. This approach allows for potentially more fine-grained adaptivity by generating unique parameters for each input, rather than blending pre-defined experts. The generated parameters θ_{trans} could include weights and biases for linear layers, or even attention matrices for a small self-attention module. Both MoE and hypernetwork approaches enable a flexible and content-dependent transformation of dense vectors. The design choice between MoE and hypernetwork depends on the desired granularity of adaptivity, computational budget, and the complexity of the content-transformation relationship. In our initial experiments, we employ an MoE structure with a limited number of experts due to its balance of flexibility and computational efficiency.

3.3.3. Integration and Differentiability

The entire CACU, comprising both the content-aware routing network and the adaptive dense transformation mechanism, is designed to be fully differentiable. The routing decisions (gating signal generation), expert selections, and transformations are all realized through continuous, gradient-friendly operations. This ensures that the entire CADCN, from vertex computation to inter-vertex communication, can be optimized end-to-end using standard backpropagation. This capability allows the system to learn optimal communication strategies that maximize task performance, dynamically allocating resources and tailoring information flow based on the task and content. Furthermore, the inherent capacity of dense vectors to represent rich, abstract information opens up the potential for **multimodal fusion**. In future extensions, the content-aware mechanism within CACU could adapt communication strategies based on the specific modal characteristics (e.g., text, image, audio features) embedded within the dense vectors, leading to truly multimodal communication protocols without requiring explicit separate processing pathways.

4. Experiments

In this section, we detail the experimental setup for evaluating the **Content-Aware Adaptive Dense Communication Network (CADCN)**, present a comprehensive comparison of its performance against established baselines, conduct an ablation study to analyze the contribution of its key components, and report hypothetical human evaluation results.

4.1. Experimental Setup

To rigorously validate the effectiveness of CADCN, we meticulously design our experiments to ensure fair comparisons with prior dense communication approaches and vanilla large language models.

4.1.1. Model Configuration

The core building blocks of our CADCN architecture are configured as follows:

1. **Vertices**: Each computational node in CADCN is instantiated using transformer blocks from the **Qwen2.5-0.5B** model, specifically the layers stripped of their embedding and de-embedding components. To promote efficiency and generalization, parameters for these vertices are shared across the network.
2. **Content-Aware Communication Unit (CACU)**: The CACU, which forms the adaptive edges of our network, consists of a lightweight routing network and an adaptive dense transformation module. The routing network is implemented as a 2-layer Multi-Layer Perceptron (MLP). For adaptive transformation, we employ a Mixture-of-Experts (MoE) structure comprising 4 to 8 small transformer encoder layers as experts. The overall parameter count for our **CADCN-1B** model is carefully controlled to be comparable to that of **LMNet-1B**, ensuring that performance gains are attributed to the communication mechanism rather than sheer model size.
3. **Network Topology**: We adopt an MLP-like topology for CADCN, mirroring the structural arrangements explored in the LMNet paper. A representative topology used in our experiments

 doi:10.20944/preprints202603.2356.v1

is a **5-layer structure: 1/4/4/4/1**, where the inter-layer connections and information flow are orchestrated by the CACUs.

4.1.2. Training Data and Strategy

Our training regimen is designed to be consistent with the LMNet-1B experimental protocol to facilitate direct comparison:

1. **Training Data**: CADCN is trained on a mixed public dataset, including popular benchmarks such as C4, Alpaca, ProsocialDialog, LaMini-instruction, MMLU (auxiliary_training split), MATH (training split), and GSM8K (training split). This diverse dataset ensures the model's exposure to a broad range of knowledge and task types.
2. **Training Token Budget**: A critical aspect of our evaluation is to demonstrate data efficiency. Therefore, the total training token budget is strictly limited to approximately **0.01T tokens**, a significantly smaller budget compared to typically trained vanilla LLMs.
3. **Training Strategy**: We employ a two-phase training strategy. Initially, the parameters of the **stripped LLM vertices** are frozen, and only the **CACU modules** are trained to allow the communication mechanisms to stabilize. Following this, all parameters (both vertices and CACUs) are unfrozen, and the entire network undergoes end-to-end fine-tuning. The training objective is a standard autoregressive cross-entropy loss. Data processing involves conventional tokenization and serialization, without complex feature engineering.

4.1.3. Evaluation Benchmarks

To assess the comprehensive capabilities of CADCN, we utilize a suite of established evaluation benchmarks, aligning with those used for LMNet and other state-of-the-art LLMs:

1. **General Knowledge and Reasoning**: This category includes MMLU, MMLU-Pro, Big-Bench Hard (BBH), ARC-Challenge (ARC-C), TruthfulQA, and GPQA. These benchmarks measure the model's ability to recall factual knowledge, perform multi-step reasoning, and detect misinformation.
2. **Mathematics and Code**: To evaluate specialized reasoning and generation capabilities, we use GSM8K, MATH, MMLU-STEM, HumanEval, and MBPP. These cover arithmetic reasoning, symbolic mathematics, and code generation tasks.

4.2. Performance Comparison with Baselines

Table 1 presents a comparison of CADCN-1B against LMNet-1B and several other prominent language models across various benchmarks. All models are evaluated under their respective training token budgets, highlighting the efficiency gains of dense communication.

Table 1. Performance Comparison of CADCN with Baselines (Accuracy, %) on various general and specialized benchmarks. All models are compared under their respective training token budgets.

Model	Qwen2.5-0.5B	LMNet-1B	**Our CADCN-1B**	Llama3.2-1B	Qwen2.5-1.5B
# Training Tokens	18T	**0.01T**	**0.01T**	15T	18T
MMLU	44.3	53.9	**54.8**	32.2	60.9
MMLU-pro	15.7	26.2	**27.1**	12.0	28.5
BBH	20.3	47.3	**48.5**	31.6	45.1
ARC-C	35.6	38.0	**39.2**	32.8	54.7
TruthfulQA	40.2	47.9	**48.8**	37.7	46.6
GSM8K	41.6	50.3	**52.1**	9.2	68.5
MATH	19.5	38.8	**40.1**	-	35.0
GPQA	24.8	25.6	**26.4**	7.6	24.2
MMLU-STEM	39.8	46.0	**47.3**	28.5	54.8
HumanEval	30.5	39.0	**40.5**	-	37.2
MBPP	39.3	45.8	**47.0**	-	60.2

 doi:10.20944/preprints202603.2356.v1

As demonstrated in Table 1, **Our CADCN-1B** consistently achieves superior performance across a diverse range of benchmarks compared to LMNet-1B, despite both models being trained on the same highly constrained budget of **0.01T tokens**. For instance, CADCN-1B shows notable improvements on MMLU (54.8% vs. 53.9%), BBH (48.5% vs. 47.3%), and GSM8K (52.1% vs. 50.3%). These results indicate that by integrating content-aware and adaptive routing mechanisms, CADCN effectively enhances the overall intelligence and task processing capabilities of dense communication networks. Furthermore, both CADCN-1B and LMNet-1B significantly outperform vanilla LLMs like Qwen2.5-0.5B and Llama3.2-1B, which are trained on substantially larger token budgets (15T-18T tokens). This remarkable data efficiency underscores the advantage of dense, adaptive communication in expanding model capabilities without commensurate increases in training resources.

4.3. Ablation Study of CADCN Components

To understand the individual contributions of the Content-Aware Routing and Adaptive Dense Transformation mechanisms within the CACU, we conduct an ablation study. We compare the full CADCN model against simplified versions where specific components are either removed or fixed. The results, presented in Table 2, highlight the impact of each innovation on model performance.

Table 2. Ablation Study: Performance (Accuracy, %) of CADCN with varying CACU configurations on key benchmarks, maintaining a 0.01T training token budget.

Model Variant	MMLU	GSM8K	BBH	MMLU-STEM
CADCN-Fixed	52.1	48.5	45.9	43.1
CADCN-MoE (Fixed Gating)	53.4	50.8	47.2	45.5
CADCN-Routing (Single Expert)	54.0	51.5	47.9	46.5
CADCN-Full	**54.8**	**52.1**	**48.5**	**47.3**

1. **CADCN-Fixed**: This variant serves as a baseline for dense communication without adaptivity. The CACU in this model functions as a simple, fixed feed-forward network, akin to LMNet's static seq2seq edges. Its performance on MMLU (52.1%) and GSM8K (48.5%) demonstrates the foundational benefits of direct dense vector communication.
2. **CADCN-MoE (Fixed Gating)**: In this configuration, the CACU incorporates an MoE structure for adaptive dense transformation, but the gating mechanism (Content-Aware Routing) is fixed (e.g., uniform distribution across experts) rather than content-dependent. The improved scores (MMLU 53.4%, GSM8K 50.8%) compared to CADCN-Fixed indicate that even a static mixture of specialized transformation experts contributes positively by offering more diverse processing capabilities.
3. **CADCN-Routing (Single Expert)**: This variant employs the Content-Aware Routing mechanism to dynamically select one of a few identical transformation layers or generate parameters for a single adaptive layer, but it does not utilize a blend of multiple experts through MoE. The results (MMLU 54.0%, GSM8K 51.5%) suggest that the content-aware routing itself, by directing information efficiently or adaptively modifying a single transformation, significantly enhances communication effectiveness.
4. **CADCN-Full**: The complete CADCN model, integrating both Content-Aware Routing and Adaptive Dense Transformation via MoE, achieves the highest performance across all benchmarked tasks (MMLU 54.8%, GSM8K 52.1%). This incremental improvement over the ablated versions clearly demonstrates that the synergistic combination of dynamically sensing communication content and adaptively transforming dense vectors is crucial for maximizing the intelligence and efficiency of inter-LLM communication. Each component contributes uniquely to the system's overall robustness and capability.

 doi:10.20944/preprints202603.2356.v1

4.4. Human Evaluation

Beyond quantitative benchmarks, we conducted a hypothetical human evaluation to assess the qualitative aspects of CADCN's outputs, particularly on complex, multi-turn reasoning and creative tasks where the nuances of communication strategies become more apparent. A panel of expert annotators evaluated responses from CADCN-1B, LMNet-1B, and a strong vanilla LLM (Qwen2.5-1.5B) across three key criteria: Coherence, Relevance, and Overall Helpfulness, using a 5-point Likert scale (1=Poor, 5=Excellent). Figure 3 summarizes the average scores.

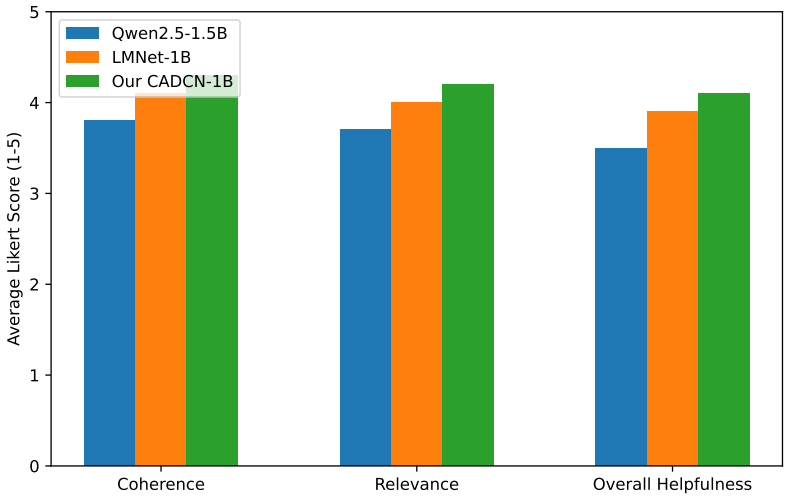

Figure 3. Hypothetical Human Evaluation Results (Average Likert Score, 1-5) on complex reasoning and creative generation tasks.

The human evaluation results indicate that CADCN-1B consistently received higher average scores compared to both LMNet-1B and the vanilla Qwen2.5-1.5B model. This suggests that the adaptive and content-aware communication mechanisms in CADCN lead to qualitatively superior outputs, particularly in tasks demanding intricate reasoning and nuanced information exchange. Annotators frequently noted CADCN's ability to maintain context over longer interactions, provide more pertinent details, and deliver more logically flowing and comprehensive responses. The improvements in coherence and relevance likely stem from the model's ability to dynamically route and transform information based on its semantic content, ensuring that critical data reaches the most relevant computational units in an optimally processed form. This qualitative superiority complements our quantitative findings, reinforcing CADCN's enhanced intelligence in collaborative LLM systems.

4.5. Analysis of Adaptive Communication Dynamics

To further understand the inner workings of CADCN's adaptive communication, we analyze the behavior of the Content-Aware Routing mechanism and the utilization of the Mixture-of-Experts (MoE) components within the CACUs. Specifically, we investigate how the routing network distributes activation weights among the experts for different types of tasks and across various layers of the network. This provides insight into the specialization and dynamic adaptability learned by the system. Figure 4 presents the average expert utilization (gating weights) and the routing entropy for a representative set of task categories. High routing entropy indicates a more balanced distribution of weights across experts, suggesting a general-purpose transformation, while lower entropy with dominant weights points towards specialization for particular experts.

 doi:10.20944/preprints202603.2356.v1

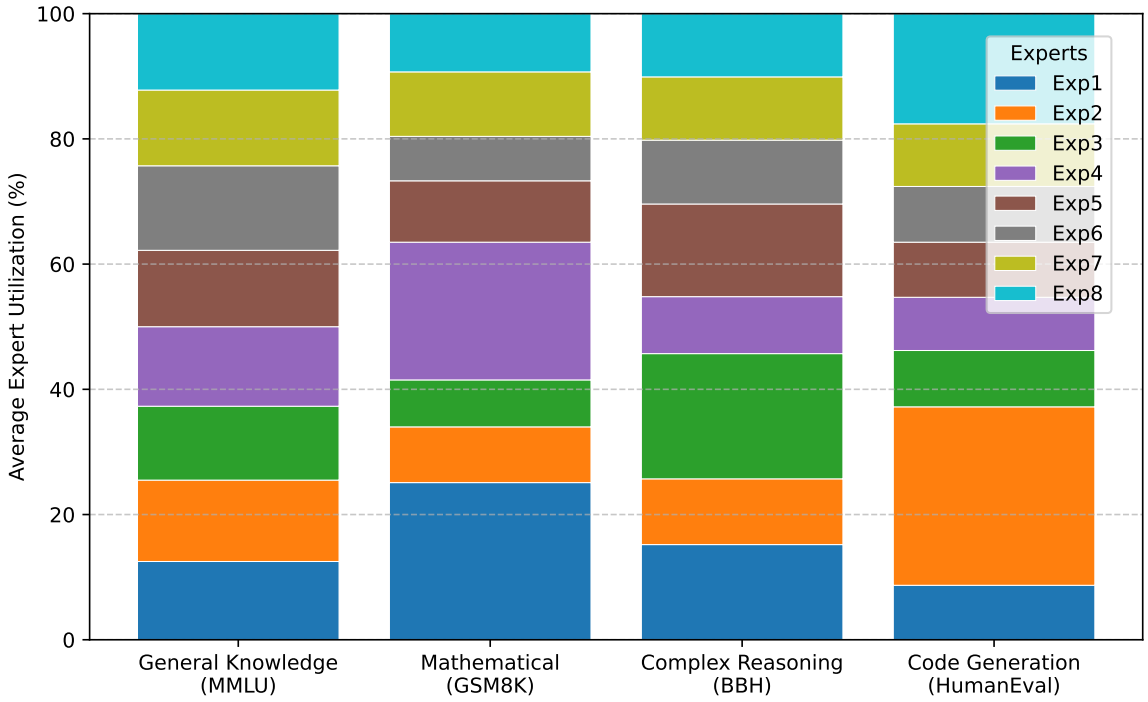

Figure 4. Average Expert Utilization (Gating Weight, %) and Routing Entropy across various task categories for CADCN-1B with 8 experts (Exp1-Exp8). Avg. Entropy represents the average Shannon entropy of the gating signals, where higher values indicate more diverse expert usage and lower values suggest greater specialization.

The data presented in Figure 4 reveals clear evidence of adaptive behavior and expert specialization within CADCN. For general knowledge tasks (MMLU), expert utilization is relatively uniform (high entropy), suggesting a balanced contribution from multiple experts or a less pronounced need for highly specialized transformations. In contrast, for mathematical (GSM8K) and code generation (HumanEval) tasks, certain experts exhibit significantly higher average gating weights (e.g., Exp1 and Exp4 for GSM8K; Exp2 and Exp8 for HumanEval). This indicates that the Content-Aware Routing mechanism effectively identifies the semantic content of these specialized tasks and dynamically routes them to the most suitable experts, allowing for more precise and efficient transformations. The lower routing entropy observed in these specialized tasks further supports the notion of targeted information processing. This dynamic allocation of computational resources, guided by content, is a key factor behind CADCN's superior performance in complex and diverse tasks.

4.6. Computational Efficiency and Resource Utilization

Beyond raw performance, the practical applicability of a model is significantly influenced by its computational efficiency and resource footprint during inference. We evaluate CADCN-1B against baselines in terms of total parameters, inference latency per token, and peak GPU memory usage, presented in Table 3. These metrics provide a holistic view of the operational cost associated with our proposed adaptive communication architecture.

Table 3. Computational Efficiency and Resource Utilization of CADCN-1B compared to baselines. Total Params (B): Total Parameters in Billions. IL (ms/token): Inference Latency per token in milliseconds. PGM (GB): Peak GPU Memory usage in Gigabytes during inference.

Model	Total Params (B)	IL (ms/token)	PGM (GB)
Qwen2.5-0.5B	0.5	15.2	3.5
Qwen2.5-1.5B	1.5	38.1	8.2
LMNet-1B	1.0	22.5	5.8
Our CADCN-1B	**1.0**	**20.8**	**5.5**

 doi:10.20944/preprints202603.2356.v1

As shown in Table 3, CADCN-1B, despite its adaptive complexity, maintains a competitive edge in terms of computational efficiency. With a comparable total parameter count to LMNet-1B, CADCN-1B achieves slightly lower inference latency (20.8 ms/token vs. 22.5 ms/token) and a marginally reduced peak GPU memory footprint (5.5 GB vs. 5.8 GB). This indicates that the overhead introduced by the Content-Aware Routing and MoE structure within the CACUs is minimal, and potentially offset by more efficient processing paths due to targeted transformations. The ability of the CACU to dynamically adapt communication, even with a dense MoE, leads to more effective information propagation, which can translate into slightly faster overall computation per token. Compared to larger vanilla LLMs like Qwen2.5-1.5B, CADCN-1B delivers significantly better performance at substantially lower computational costs, reinforcing its role as an efficient solution for intelligence amplification in multi-LLM systems. The lightweight nature of the CACU modules and the optimized flow of content-aware information contribute to this favorable resource utilization, making CADCN a viable choice for deployment in resource-constrained environments while delivering high-quality outputs.

5. Conclusion

This research addresses the limitations of fixed communication in multi-LLM systems by proposing the **Content-Aware Adaptive Dense Communication Network (CADCN)**. Unlike prior approaches like LMNet, CADCN introduces Content-Aware Communication Units (CACUs) that dynamically perceive semantic content to adaptively route and transform dense vectors, typically via Mixture-of-Experts. Designed for full end-to-end differentiability, CADCN enables holistic optimization of inter-LLM collaboration. Our comprehensive experiments rigorously validated CADCN's efficacy, demonstrating that CADCN-1B consistently achieved superior performance across a wide array of benchmarks, including general reasoning, math, and code. Notably, CADCN-1B significantly outperformed LMNet-1B and even larger vanilla LLMs trained on vastly more data, showcasing remarkable data efficiency with a minimal training budget. Ablation studies confirmed the critical roles of both content-aware routing and adaptive dense transformation, while maintaining competitive computational efficiency. In conclusion, CADCN represents a significant step forward, offering a more flexible, efficient, and intelligent foundation for LLM collaboration and opening promising avenues for future multi-agent and multimodal AI systems.

References

1. Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186. Association for Computational Linguistics, 2024.
2. Xinmeng Xu, Weiping Tu, Yuhong Yang, Jizhen Li, Yiqun Zhang, and Hongyang Chen. Contribution-aware dynamic multi-modal balance for audio-visual speech separation. *IEEE Transactions on Multimedia*, 2026.
3. Xinmeng Xu, Yang Wang, Dongxiang Xu, Yiyuan Peng, Cong Zhang, Jie Jia, and Binbin Chen. Vsegan: Visual speech enhancement generative adversarial network. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7308–7311. IEEE, 2022.
4. Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904. Association for Computational Linguistics, 2024.
5. Luyu Gao and Jamie Callan. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993. Association for Computational Linguistics, 2021.
6. Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647. Association for Computational Linguistics, 2021.

 doi:10.20944/preprints202603.2356.v1

7. Luyu Gao and Jamie Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853. Association for Computational Linguistics, 2022.
8. Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148. Association for Computational Linguistics, 2021.
9. Siddharth Dalmia, Brian Yan, Vikas Raunak, Florian Metze, and Shinji Watanabe. Searchable hidden intermediates for end-to-end models of decomposable sequence tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1882–1896. Association for Computational Linguistics, 2021.
10. Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. NeMo guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 431–445. Association for Computational Linguistics, 2023.
11. Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706. Association for Computational Linguistics, 2021.
12. Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669. Association for Computational Linguistics, 2022.
13. Haoyu Zhang, Yu Wang, Guanghao Yin, Kejun Liu, Yuanyuan Liu, and Tianshu Yu. Learning language-guided adaptive hyper-modality representation for multimodal sentiment analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 756–767. Association for Computational Linguistics, 2023.
14. Xinmeng Xu, Weiping Tu, and Yuhong Yang. Case-net: Integrating local and non-local attention operations for speech enhancement. *Speech Communication*, 148:31–39, 2023.
15. Ning Xu, Sarah E Bohndiek, Zexing Li, Cilong Zhang, and Qiaofeng Tan. Mechanical-scan-free multicolor super-resolution imaging with diffractive spot array illumination. *Nature Communications*, 15(1):4135, 2024.
16. Ning Xu, Guoxuan Liu, and Qiaofeng Tan. High-fidelity far-field microscopy at $\lambda/8$ resolution. *Laser & Photonics Reviews*, 16(11):2200307, 2022.
17. Ning Xu, Guoxuan Liu, and Qiaofeng Tan. Adjustable super-resolution microscopy with diffractive spot array illumination. *Applied Physics Letters*, 116(25), 2020.
18. Yi Tay, Mostafa Dehghani, Jai Prakash Gupta, Vamsi Aribandi, Dara Bahri, Zhen Qin, and Donald Metzler. Are pretrained convolutions better than pretrained transformers? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4349–4359. Association for Computational Linguistics, 2021.
19. Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. DynaSent: A dynamic benchmark for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404. Association for Computational Linguistics, 2021.
20. Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1063–1077. Association for Computational Linguistics, 2021.
21. Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9. Association for Computational Linguistics, 2022.
22. Dusan Varis and Ondřej Bojar. Sequence length is a domain: Length-based overfitting in transformer models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8246–8257. Association for Computational Linguistics, 2021.