NEURAL ARCHITECTURE SEARCH WITH LLM-DISCOVERED FUSION OPERATORS FOR MULTIMODAL CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fusion operators play an important role in multimodal classification (MMC) by determining how representations from different modalities interact and are fused, thereby affecting the achievable performance. However, the fusion operators used in existing MMC methods are largely human-designed, limiting the diversity of fusion mechanisms that can be explored. We propose LLMDFO-NAS, a neural architecture search framework that automatically constructs and explores the fusion operator space using Large Language Models (LLMs) and evolutionary search. Specifically, it leverages LLMs to generate executable and validated fusion primitives and employs evolutionary search to discover effective primitive compositions as candidate fusion operators. Candidate fusion operators are efficiently evaluated across multiple architecture contexts using an architecture-aggregated zero-cost proxy with Pareto selection, and the high quality ones among them are selected as the discovered operators. These discovered operators then define the operator library for downstream multimodal architecture search, forming a two-level automated design process. Experiments on five representative benchmarks show that the discovered operator spaces consistently improve the resulting fusion architectures across different modality configurations. Our work extends automated design from architecture optimization within a fixed functional space to the joint design of functional and structural spaces.

1 INTRODUCTION

MMC exploits complementary information from multiple modalities to obtain more comprehensive representations than those derived from a single modality (Wang et al., 2020; Liang et al., 2022; Yu et al., 2026). A central component of MMC is fusion operator, which governs how representations from different modalities interact and are fused (Joze et al., 2020; Guo et al., 2024). However, the fusion operators used in existing MMC methods are largely human-designed, limiting the diversity of fusion mechanisms that can be explored (Li & Tang, 2026). Moreover, since different tasks may favor different forms of cross-modal interaction, a fixed library of predefined fusion operators may lack the flexibility needed to meet task-specific fusion requirements (Peng et al., 2022; Wei et al., 2024). To alleviate the limitations of manual fusion design, Neural Architecture Search (NAS) has been widely adopted in MMC due to its ability to automatically optimize multimodal fusion architectures by searching various structural configurations (Yin et al., 2022; Liang et al., 2024; Fu et al., 2024a). However, as illustrated in Fig. 1, existing multimodal NAS methods still largely optimize architectures over predefined fusion operators. Although they can adapt where and in what order modality interactions occur, the operators used to compute these interactions remain fixed in advance. Consequently, the fusion operator space itself remains manually specified, constraining the range of fusion mechanisms that multimodal NAS can explore.

The emergence of LLMs has brought new possibilities for automatically constructing fusion operator spaces. These methods have demonstrated strong capabilities in code generation, program synthesis, and automated design (Seo et al., 2026; Chen et al., 2021; 2023), while LLM-guided iterative and evolutionary methods have shown effectiveness in discovering executable programs, heuristics, and algorithms (Romera-Paredes et al., 2024; Zhang & Lu, 2026; Novikov et al., 2025). Fusion operators can be represented as executable programs that map multiple modality representations to fused

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

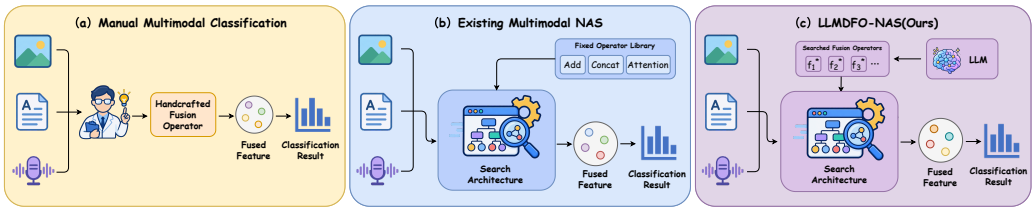


Figure 1: Comparison of fusion operator design paradigms in MMC. (a) Multimodal fusion relies on handcrafted fusion operators; (b) existing multimodal NAS searches for architectures within a fixed fusion operator library; and (c) LLMDFO-NAS discovers fusion operators with LLMs and uses them for architecture search.

representations, enabling their automatic discovery and optimization through LLM-based generation and evolutionary search. This makes it possible to move beyond selecting from a fixed operator library and instead automatically construct and optimize the fusion operator library itself. However, automatically exploring such an fusion operator introduces three challenges: (1) generated operators must satisfy execution and differentiability requirements, (2) the compositional search space grows rapidly, and (3) operator quality may depend on the architecture in which an operator is instantiated.

To address these issues, we propose LLMDFO-NAS, which combines LLM-based primitive generation with population-based evolutionary composition search. It views a fusion operator as a program composed of executable primitives, with LLMs generating diverse primitives and evolutionary search identifying effective compositions to form candidate fusion operators. Candidate fusion operators are evaluated across multiple architecture contexts using an architecture-aggregated zero-cost proxy and Pareto selection, avoiding full training for every candidate while supporting effective fusion operator space exploration. The selected high-performing operators constitute the discovered operators, which define the functional building blocks for multimodal architecture search, forming a two-level automated design process. *Fusion operator search determines how modality representations interact, whereas architecture search determines where and in what order these interactions occur*. The framework explores previously unspecified fusion compositions and provides a more flexible search space for downstream NAS. Experiments on five representative benchmarks demonstrate that the discovered operator spaces consistently improve the resulting fusion architectures across different modality configurations. Our main contributions are summarized as follows:

- LLMDFO-NAS extends multimodal NAS from architecture search over predefined operators to treating fusion operator space construction itself as a searchable design problem.
- An LLM-guided evolutionary framework generates validated executable fusion primitives and discovers effective operator compositions beyond human-designed operator libraries.
- Architecture-aggregated zero-cost evaluation with Pareto selection enables efficient operator search across multiple architecture contexts, yielding discovered operator spaces that benefit downstream architecture search across five MMC benchmarks.

2 Related Work

Multimodal fusion: Multimodal fusion aims to integrates complementary information from different modalities or views to learn comprehensive representations. Existing studies have explored various mechanisms to improve fusion robustness. For example, adaptive weighting methods dynamically adjust the contributions of different modalities, while uncertainty-aware and trusted learning explicitly model modality reliability (Jiang et al., 2021; Geng et al., 2021). Other studies incorporate opinion aggregation, dynamic evidential fusion, and safety-aware multi-view learning to alleviate the influence of unreliable or conflicting modality information (Liu et al., 2022; Han et al., 2023; Liu et al., 2023). Recent work has further investigated modality competition, modality imbalance, and adaptive multimodal learning (Yu et al., 2026; Chen et al., 2026). Despite these advances, the underlying fusion mechanisms and operators remain largely human-designed and predefined. In

contrast, LLMDFO-NAS treats fusion operators as searchable objects and automatically discovers effective compositional fusion mechanisms.

Multimodal NAS: NAS automates architecture design by searching predefined spaces, with recent multimodal studies exploring modality selection, fusion order, connectivity, and multi-scale representations. CoMO-NAS formulates multimodal fusion as a multi-objective architecture search problem, while MS-NAS and subsequent multi-scale approaches further exploit hierarchical feature representations to improve fusion effectiveness (Fu et al., 2024b; Lv et al., 2024; Fu et al., 2025). In parallel, training-free and zero-cost estimators have been developed to reduce the computational cost of architecture evaluation by ranking candidate networks without full training (Lee & Ham, 2024; Tybl & Neumann, 2025). Despite these advances, existing multimodal NAS methods generally search over manually predefined fusion operators, meaning that the search process mainly determines where and in what order modalities interact rather than how the fusion operation itself should be constructed. LLMDFO-NAS extends this paradigm by making the fusion operator itself searchable.

LLM-driven automated design: LLMs have increasingly been integrated into automated design frameworks, where they are used not only to generate executable solutions but also to participate in iterative search and optimization. Recent studies have extended this paradigm toward automatic metaheuristic generation, structured exploration of algorithmic search spaces, and population-level heuristic evolution. LLaMEA combines LLM-based generation with evolutionary optimization to automatically construct metaheuristics (van Stein & Bäck, 2025), while MCTS-AHD introduces Monte Carlo tree search to improve the exploration of candidate heuristic designs (Zheng et al., 2025). EoH-S further evolves sets of complementary heuristics to enhance population diversity and search effectiveness (Liu et al., 2026). These methods demonstrate the potential of LLM-guided search for discovering executable algorithms beyond manually specified candidates. However, existing studies primarily focus on general-purpose algorithms and heuristics. LLMDFO-NAS applies this paradigm to multimodal learning by treating executable fusion operators as searchable programs and integrating the discovered operators into downstream multimodal architecture search.

3 METHOD

LLMDFO-NAS consists of three stages: (1) multimodal feature extraction, (2) discovering fusion operators through LLM-guided evolutionary search, and (3) searching multimodal architectures over the discovered fusion operator space. An overview is shown in Fig. 2.

3.1 MULTIMODAL FEATURE EXTRACTION

Let $\mathcal{D} = \left\{\left(x_i^1, x_i^2, \ldots, x_i^V, y_i\right)\right\}_{i=1}^N$ denote a multimodal dataset containing V modalities, where x_i^v is the input of the v-th modality and y_i is its label. Each modality is first processed by its corresponding feature extractor:

$$m_i^v = E_v\left(x_i^v\right), \tag{1}$$

where $E_v(\cdot)$ denotes the feature extractor for modality v. Since features from different modalities may have different dimensions, they are projected into a common latent space:

$$z_i^v = \alpha_v\, \sigma\left(W_v m_i^v + b_v\right), \qquad \alpha_v = \frac{\exp(w_v)}{\sum_{u=1}^V \exp(w_u)}, \tag{2}$$

where $\sigma(\cdot)$ is a nonlinear activation function, W_v and b_v are the learnable projection parameters for modality v, and w_v is a learnable scalar logit that determines the relative contribution of modality v. The normalized modality weight α_v is obtained by applying the softmax function. The aligned feature set is denoted by $\mathcal{M}_i = \{z_i^1, z_i^2, \ldots, z_i^V\}$. The common feature dimension provides a unified input and output interface for the subsequent fusion operator and architecture searches, allowing the same discovered operator to be reused at different positions in a fusion architecture.

3.2 LLM-GUIDED FUSION OPERATOR DISCOVERY

The fusion operator search contains two stages: primitive discovery and fusion composition search. The former constructs a set of executable primitives, whereas the latter composes these primitives into more expressive fusion operators.

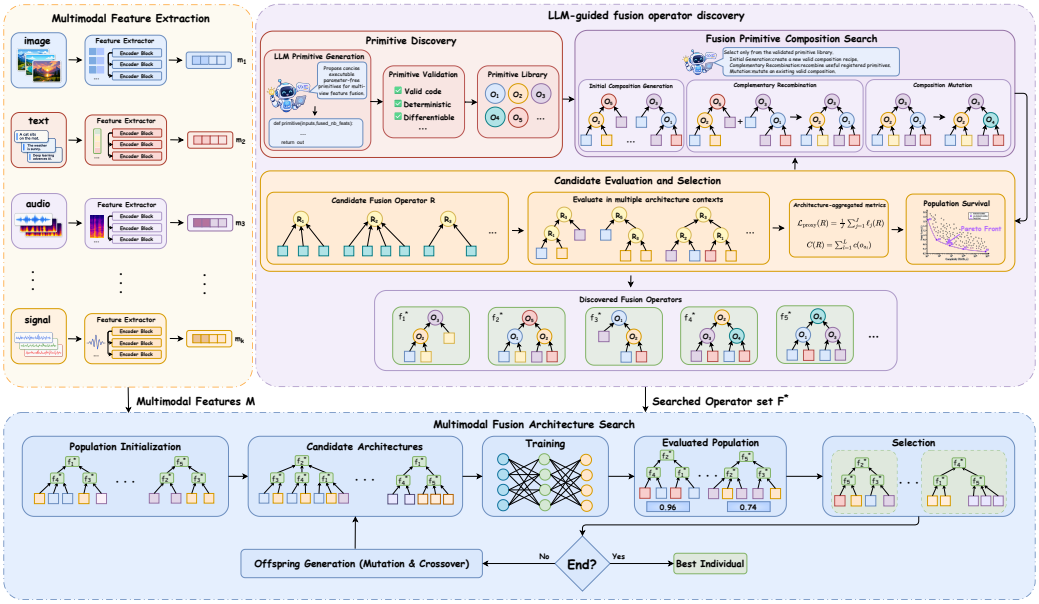


Figure 2: Overview of the proposed LLMDFO-NAS framework.

3.2.1 Primitive Discovery

A primitive is defined as a stateless, shape-preserving function with zero operator-specific trainable parameters:

$$o : \left(\mathbb{R}^d\right)^r \to \mathbb{R}^d, \tag{3}$$

where r is the number of input features and d is the common feature dimension. We restrict each primitive to be stateless, shape-preserving, differentiable, and free of operator-specific trainable parameters. We decouple primitive discovery from architecture-specific training, allowing the same primitive to be instantiated and compared across different candidate architectures. Accordingly, the current search focuses on reusable feature transformations and excludes parameterized modules such as attention, gating, and bilinear interactions.

We prompt an LLM to generate executable primitives based on the fusion objective, input and output requirements, and previously accepted primitives. Each primitive is then checked for execution validity, gradient propagation, and functional duplication. Invalid or duplicate candidates are discarded, yielding the validated primitive library

$$\mathcal{O} = \{o_1, o_2, \ldots, o_B\}, \tag{4}$$

which provides the basic components for the subsequent composition search.

3.2.2 Fusion Primitive Composition Search

A fusion primitive composition specifies how multiple primitives are connected to form an executable fusion operator and is represented as a directed acyclic computation graph:

$$R = \{n_1, n_2, \ldots, n_L\}, \qquad L_{\min} \le L \le L_{\max}, \tag{5}$$

where each node specifies a primitive from $\mathcal{O}$ and a set of references to the original inputs or preceding nodes. Let h_q denote the feature representation associated with reference q, which is either an original input feature or the output of a preceding node. For the l-th node,

$$h_l = o_{a_l}\left(\{h_q \mid q \in \mathcal{I}_l\}\right), \tag{6}$$

where $o_{a_l} \in \mathcal{O}$ and $\mathcal{I}_l$ contains references to the original modality inputs or preceding nodes. The output of the last node defines the resulting fusion operator. Invalid computation graphs are discarded before evaluation.

Inspired by population based LLM driven algorithm design (Liu et al., 2024; Hu et al., 2026; Sadikov, 2026), we optimize fusion primitive compositions through an evolutionary process consisting of initialization, recombination, mutation, and selection. The LLM generates the initial population and proposes offspring through complementary recombination and composition mutation, while population selection retains promising candidates according to the evaluation criteria described below. Detailed descriptions are provided in Appendices A.1–A.3.

3.2.3 CANDIDATE EVALUATION AND SELECTION

The candidate evaluation strategy aims to estimate the quality of a fusion primitive composition beyond any particular architecture context. A composition that performs well only within a specific architecture may not provide a generally useful operator for downstream NAS. We therefore regard operator quality as an architecture-marginalized property and evaluate each LLM-generated composition across multiple architecture contexts, while jointly considering its functional quality and structural complexity (Liu et al., 2024; Ruan et al., 2026).

Fully training a multimodal model for every generated composition would incur a prohibitive computational cost. We therefore adopt the expressivity component of the Zero-Cost Fitness Screening mechanism in MBTF-NAS (Guo et al., 2026) as the quality proxy. Each candidate fusion primitive composition is instantiated in J randomly sampled multimodal fusion architectures with randomly initialized parameters. Gaussian random inputs are propagated through each architecture, and the spectral entropy of the resulting feature covariance matrix is used to quantify representational expressivity. Let $E_j(R)$ denote the expressivity score obtained by composition R in the j-th evaluation architecture. We convert this score into a reciprocal proxy loss:

$$\ell_j(R) = \frac{1}{\max(E_j(R), \epsilon)}, \tag{7}$$

where ϵ is a small positive constant introduced for numerical stability. The proxy losses across sampled architectures are then aggregated as

$$\mathcal{L}_{\text{proxy}}(R) = \frac{1}{J} \sum_{j=1}^{J} \ell_j(R), \tag{8}$$

where a lower value indicates consistently higher expressivity across different structural contexts. This aggregation provides an architecture-marginalized estimate of operator quality, reducing the risk of selecting compositions that are effective only under a particular architecture.

In addition to functional quality, we explicitly quantify structural complexity to discourage unnecessarily elaborate compositions:

$$C(R) = \sum_{l=1}^{L} c(o_{a_l}), \tag{9}$$

where $c(o_{a_l})$ denotes the complexity descriptor of the primitive assigned to node n_l. Population survival jointly considers the aggregated proxy loss and structural complexity through empirical Pareto selection, favoring compositions that achieve a better balance between representational expressivity and structural simplicity.

Finally, the discovered fusion operator library is selected from all evaluated compositions by prioritizing candidates on the empirical Pareto front while preserving diversity in primitive signatures and functional categories:

$$\mathcal{F}^* = \{f_1^*, f_2^*, \ldots, f_K^*\}. \tag{10}$$

We set $K = 5$ in our implementation.

3.3 MULTIMODAL FUSION ARCHITECTURE SEARCH

After fusion operator discovery, $\mathcal{F}^*$ defines the functional dimension of the multimodal search space, while the subsequent architecture search optimizes its structural dimension. Specifically, the discovered fusion operators determine how modality representations interact, whereas the architecture determines where and in what order these interactions occur. For downstream architecture search,

each candidate architecture is represented as a multi-branch tree, whose leaf nodes correspond to modality features in $\mathcal{M}$ and internal nodes are assigned fusion operators from $\mathcal{F}^*$. For an internal node e with child set $\mathrm{ch}(e)$, its output is

$$h_e = f_e^* \left(\{ h_q : q \in \mathrm{ch}(e) \} \right), \qquad f_e^* \in \mathcal{F}^*. \tag{11}$$

Thus, the tree determines which modality representations interact and in what order, whereas the discovered operator determines how the representations are aggregated.

Each candidate fusion tree is decoded through post-order traversal. Starting from the weighted modality features in Eq. 2, the decoder recursively evaluates the internal nodes until obtaining the root representation c. Following the normalization strategy commonly adopted in evolutionary multimodal fusion (Liang et al., 2021), c is normalized to $\bar{c}$ before classification:

$$\bar{c} = \frac{\mathrm{sign}(c) \odot \sqrt{|c|}}{\left\| \mathrm{sign}(c) \odot \sqrt{|c|} \right\|_2}, \qquad \hat{y} = \mathrm{Softmax}\left(W_c \bar{c} + b_c \right). \tag{12}$$

The decoded network is optimized using the cross-entropy loss.

The architecture population is initialized with randomly generated multi-branch trees. Each individual is decoded, trained, and evaluated on the held-out evaluation split, with classification performance serving as its fitness. At each generation, tournament selection chooses parent architectures. Subtree crossover exchanges fusion substructures between parents, while mutation modifies either an input modality or the operator assigned to an internal node. The next population is selected from the union of the parent and offspring populations, and the best individual is retained throughout evolution. The final output is the highest fitness multimodal architecture together with its learned modality weights.

4 Experiments

In this section, we evaluate LLMDFO-NAS on five representative benchmarks and investigate the effectiveness of replacing manually predefined fusion operators with LLM-discovered ones. We report comparisons with representative multimodal NAS methods in terms of classification performance, model complexity, and search time, followed by further analyses of the fusion operator discovery process.

4.1 Experimental settings

Datasets. We conduct experiments on five benchmarks: MM-IMDB (Arevalo et al., 2017), EgoGesture (Zhang et al., 2018), NUS-WIDE-128 (Tang et al., 2017), NTU RGB-D (Shahroudy et al., 2016), and MV-RML2016.10A (O'Shea & West, 2016). These datasets cover different modality configurations and application scenarios. MM-IMDB consists of image and text modalities for movie genre prediction. EgoGesture provides RGB and depth information for gesture recognition. NUS-WIDE-128 contains multiple visual feature views for image classification, while NTU RGB-D contains video and skeleton representations for human action recognition. MV-RML2016.10A is used for modulation classification under different SNR conditions. We follow the same data preprocessing, feature representations, and dataset partitions as MBTF-NAS (Guo et al., 2026). Detailed descriptions are provided in Appendix A.5.1.

Evaluation metrics. Following baseline evaluation protocols, we report weighted F1-score for MM-IMDB and classification accuracy for EgoGesture, NUS-WIDE-128, NTU RGB-D, and MV-RML2016.10A. We additionally report the number of trainable parameters and search time to evaluate model complexity and search efficiency. The reported performance corresponds to the final fusion architecture obtained through the complete architecture search and evaluation procedure.

Compared methods. We compare LLMDFO-NAS with representative multimodal NAS methods, including EDF (Liang et al., 2021), MFAS (Perez-Rua et al., 2019), BM-NAS (Yin et al., 2022), MS-NAS (Lv et al., 2024), DC-NAS (Liang et al., 2024), CoMO-NAS (Fu et al., 2024b), CSG-NAS (Fu et al., 2024a), and MBTF-NAS (Guo et al., 2026). These methods cover a broad range of approaches to automated multimodal fusion architecture design. We follow the corresponding data processing and evaluation protocols for each benchmark.

Implementation details. All modality features are projected to a 128-dimensional space. The population size and number of generations are set to 28 and 20, with crossover and mutation probabilities of 0.9 and 0.2. Candidate networks are trained using Adam with a batch size of 256 for up to 100 epochs and an early-stopping patience of 20. DeepSeek-V4-Flash is used for fusion operator discovery, and five operators are retained for architecture search. All experiments use identical settings and are conducted with TensorFlow 2.17.0 on four NVIDIA Tesla V100 GPUs.

4.2 Experimental results

This section evaluates LLMDFO-NAS from eight aspects: (1) comparison with representative multimodal NAS methods, (2) impact analysis of different LLM backbones, (3) impact analysis of the evolutionary composition strategies, (4) effectiveness analysis of the zero-cost evaluation, (5) impact analysis of the searched fusion operators, (6) impact analysis of LLM-discovered operators on architecture search dynamics, (7) search efficiency, and (8) generalization to wireless signal processing under negative SNR conditions. The main benchmark results are obtained from the final architectures produced by the complete fusion operator discovery and architecture search procedure, while the subsequent analyses examine specific components and behaviors of the proposed framework.

Comparison with multimodal NAS methods. As shown in Table 1, LLMDFO-NAS achieves the best classification performance on all four benchmarks. Compared with MBTF-NAS, it improves performance by 0.27, 0.08, 1.10, and 0.81 percentage points on MM-IMDB, EgoGesture, NUS-WIDE-128, and NTU RGB-D, respectively. Since LLMDFO-NAS follows the same downstream architecture search and evaluation procedure as MBTF-NAS while replacing its predefined operator library with the discovered operators, the consistent gains demonstrate the effectiveness of expanding the fusion operator space. Notably, the improvements on MM-IMDB and EgoGesture are also obtained with fewer trainable parameters than MBTF-NAS, indicating that better fusion performance does not necessarily require larger architectures.

Table 1: Comparison with representative multimodal NAS methods on four benchmark datasets. The best and second-best results are highlighted in **boldface** and underlining, respectively. "–" indicates that the corresponding result is unavailable.

	MM-IMDB			EgoGesture			NUS-WIDE-128			NTU RGB-D		
Method	Params	Time (h)	F1-W (%)	Params	Time (h)	Acc. (%)	Params	Time (h)	Acc. (%)	Params	Time (h)	Acc. (%)
EDF	–	–	–	–	–	–	0.31M	11.43	73.67	–	–	–
MFAS	–	–	–	–	–	–	–	–	–	2.16M	603.64	89.50
BM-NAS	0.65M	1.24	62.94	0.61M	20.67	94.96	–	–	–	0.98M	53.68	90.48
MS-NAS	–	1.75	64.01	–	14.64	95.24	–	–	–	–	10.37	88.99
DC-NAS	0.42M	1.19	63.70	0.19M	4.57	95.22	0.53M	4.61	74.20	0.26M	13.63	90.85
CoMO-NAS	0.42M	0.68	63.84	0.26M	3.53	95.25	0.29M	2.19	74.24	0.42M	8.86	90.94
CSG-NAS	0.56M	0.98	64.12	0.20M	3.27	95.25	0.37M	2.71	74.52	0.19M	5.19	91.12
MBTF-NAS	0.37M	0.41	64.53	3.04M	1.32	95.29	0.59M	1.15	75.59	3.57M	4.12	91.25
LLMDFO-NAS	0.25M	2.33	**64.80**	1.16M	4.09	**95.37**	0.80M	2.68	**76.69**	4.40M	13.87	**92.06**

Impact analysis of different LLMs. As shown in Table 2, LLMDFO-NAS achieves comparable performance with different LLM backbones on NUS-WIDE-128. GPT-5.5 obtains the highest average accuracy of 76.76%, while Claude Opus 4.8 and Gemini 3.6 Flash achieve 76.60% and 76.61%, respectively, resulting in only a 0.16 percentage-point gap between the best and worst results. This small variation indicates that the proposed fusion operator discovery framework is relatively robust to the choice of LLM and that its effectiveness does not rely on a specific backbone.

Table 2: Impact analysis of different LLMs on LLMDFO-NAS over NUS-WIDE-128.

LLM	Run 1	Run 2	Run 3	Avg.
Claude Opus 4.8	76.66	76.49	76.66	76.60
GPT-5.5	76.75	76.88	76.64	**76.76**
Gemini 3.6 Flash	76.72	76.70	76.40	76.61

Table 3: Impact analysis of evolutionary composition strategies on NUS-WIDE-128.

Method	Accuracy (%)	Change (%)
w/o Recombination	76.40	-0.29
w/o Mutation	76.21	-0.48
LLMDFO-NAS	**76.69**	–

Impact analysis of the evolutionary composition strategies. We evaluate the contributions of complementary recombination and composition mutation on NUS-WIDE-128. As shown in Table 3, removing complementary recombination and composition mutation reduces the accuracy by 0.29 and 0.48 percentage points, respectively. These results confirm the complementary roles of both strategies, with composition mutation showing a larger contribution under this setting.

Effectiveness analysis of the zero-cost evaluation. To evaluate whether the architecture-aggregated zero-cost expressivity score can reliably rank fusion compositions, we measure its correlation with downstream trained performance on 200 held-out compositions on EgoGesture. As shown in Fig. 3, the proxy achieves a Spearman correlation of $\rho = 0.9103$ and a Kendall correlation of $\tau_b = 0.7413$, demonstrating a strong positive association between the proxy ranking and the ranking obtained from trained validation performance. These results indicate that the proposed zero-cost proxy can effectively distinguish high-quality fusion compositions without requiring full training, supporting its use for efficient operator-level evaluation.

Impact analysis of the searched fusion operators. Table 4 summarizes the five fusion operators searched on NTU RGB-D. The fusion composition search allows each candidate to contain two to four primitives. It can be observed that the searched operators exhibit different yet complementary fusion behaviors. F1 and F4 mainly exploit max-based aggregation, F3 introduces robust trimmed-mean aggregation, while F2, F3, and F5 further incorporate feature normalization. These results show that the proposed search can flexibly combine different primitives rather than relying on a single predefined fusion rule. A concrete implementation of a discovered fusion operator is provided in Appendix A.6.

Table 4: Compositional characteristics of the searched fusion operators on NTU RGB-D.

Operator	Primitive Sequence	Functional Characteristic
F1	Max interaction → Mean aggregation	Max-enhanced feature averaging
F2	Max interaction → Min-max scaling	Normalized max interaction
F3	Trimmed mean → ℓ_2 normalization	Robust normalized aggregation
F4	Max aggregation → Max aggregation	Repeated max aggregation
F5	Max aggregation → Z-score standardization	Standardized max aggregation

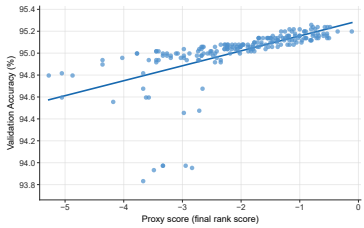


Figure 3: Correlation between proxy scores and trained validation performance on 200 held-out fusion compositions on EgoGesture.

We further report the usage frequency of each operator over all 531 valid architectures. As shown in Table 5, the complete architecture population contains $2{,}609$ fusion node occurrences, and different searched operators exhibit clear differences in usage frequency. F3 and F5 are more frequently selected on NTU RGB-D, indicating that downstream architecture search tends to favor effective operator compositions. These findings highlight the benefit of constructing a task-adaptive fusion operator space for multimodal NAS.

Table 5: Usage frequency of the searched fusion operators over all 531 valid architectures on NTU RGB-D.

Operator	Count	Share	Coverage
F1	198	7.6%	180/531 (33.9%)
F2	218	8.4%	177/531 (33.3%)
F3	440	16.9%	433/531 (81.5%)
F4	20	0.8%	18/531 (3.4%)
F5	**1,733**	**66.4%**	**484/531 (91.1%)**
Total	2,609	100.0%	–

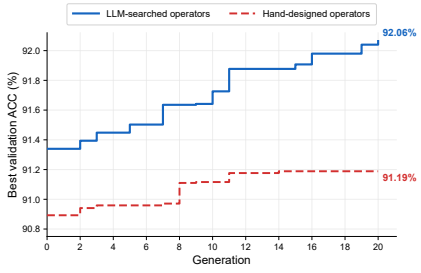


Figure 4: Search dynamics on NTU RGB-D with LLM-discovered and hand-designed fusion operators.

Impact analysis of LLM-discovered operators on search dynamics. Fig 4 compares the architecture search trajectories using LLM-discovered and manually designed operators on NTU RGB-D. The LLM-discovered operator space consistently achieves higher validation accuracy throughout the 20-generation search and reaches a final accuracy of 92.06%, compared with 91.19% for the hand-designed operator space. These results indicate that the discovered operators provide a more effective search space, improving both the search trajectory and the quality of the final architecture.

Search efficiency. Table 1 further demonstrates the practical efficiency of LLMDFO-NAS. On MM-IMDB and EgoGesture, LLMDFO-NAS achieves the best classification performance with compact architectures of only 0.25M and 1.16M parameters, respectively, while maintaining manageable search costs. On NUS-WIDE-128, although a slightly larger model is obtained, the search requires only 2.68 hours and yields the highest accuracy. A similar advantage is observed on NTU RGB-D, where LLMDFO-NAS achieves the best performance with an acceptable search cost compared with several NAS baselines. These results indicate that the additional operator-space exploration introduces a reasonable computational cost in exchange for consistently improved fusion architectures.

Generalization to Wireless Signal Processing.

To further evaluate the generalization of LLMDFO-NAS, we extend it to the challenging MV-RML2016.10A wireless signal classification task under negative SNR conditions. As shown in Table 6, LLMDFO-NAS substantially outperforms representative unimodal methods, including AWN Zhang et al. (2023), MCLDNN Xu et al. (2020), ResNet He et al. (2016), LSTM Graves (2012), and GRU Chung et al. (2014), with gains of 8.32, 5.50, and 6.11 percentage points over the strongest unimodal baselines at -12, -16, and -20 dB, respectively. It also surpasses representative multimodal methods, including ETMC Han et al. (2023), AssoDMVC Liang et al. (2025), and IBML Qin et al. (2026), by up to about 2.5 percentage points, and achieves the highest accuracy at -12 and -16 dB among all compared methods, while remaining only 0.02 percentage points below MBTF-NAS at -20 dB. These results demonstrate that the discovered fusion operators remain effective under severe noise and generalize well to challenging wireless signal classification.

Table 6: Accuracy comparison on MV-RML2016.10A under different SNR levels (dB).

Method	Accuracy (%) at SNR (dB)		
	-12	-16	-20
Unimodal Methods			
AWN	28.09	15.82	9.36
MCLDNN	27.82	14.77	8.59
ResNet	23.23	9.05	9.18
LSTM	14.32	10.27	10.23
GRU	28.05	15.95	8.77
Multi-modal Methods			
ETMC	32.16	18.61	14.45
AssoDMVC	33.89	18.66	14.45
IBML	33.56	18.95	13.87
EDF	34.97	19.16	15.59
CSG-NAS	35.59	19.79	15.86
DC-NAS	35.09	19.77	14.86
EFB-EMVC	35.56	19.83	15.75
MBTF-NAS	36.09	21.32	**16.36**
LLMDFO-NAS	**36.41**	**21.45**	16.34

5 Conclusion

In this paper, we propose LLMDFO-NAS for automated multimodal fusion design. Unlike existing multimodal NAS methods that treat the fusion operator library as fixed, LLMDFO-NAS makes both the functional and structural dimensions of fusion searchable. LLM-guided evolutionary search constructs a task-adaptive operator space, while downstream NAS organizes the discovered interactions into fusion architectures. Experiments on five benchmarks consistently demonstrate the effectiveness of replacing predefined operators with discovered ones. More broadly, our results suggest that the operator space itself can be treated as an optimization object rather than a fixed human prior, advancing multimodal NAS toward more fully automated search-space design.

AI use statement

Large language models were used as part of the proposed LLMDFO-NAS framework to generate and search candidate fusion primitives and operators. Generative AI tools were also used for language polishing of the manuscript. All experiments, result analysis, interpretation, and scientific conclusions were conducted by the authors.

Ethics statement

This work uses existing multimodal benchmark datasets and does not involve new data collection or direct interaction with human participants. Large language models are used only within the proposed framework to generate candidate fusion primitives and operators, which are subsequently subject to predefined validation and evaluation procedures. We do not identify any specific ethical concerns beyond the general limitations and potential biases associated with the benchmark datasets and pretrained language models used in this study.

Reproducibility statement

We provide detailed descriptions of the proposed method, search procedure, datasets, evaluation protocols, and implementation settings in Sections 3 and 4. The main hyperparameters, training configuration, LLM backbone, search budget, software environment, and hardware setup are explicitly reported. All controlled comparisons use identical data partitions, input features, training configurations, and search budgets.

References

John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A González. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017.

Angelica Chen, David Dohan, and David So. Evoprompting: Language models for code-level neural architecture search. *Advances in neural information processing systems*, 36:7787–7817, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Zijie Chen, Kejing Yin, Wenfang Yao, William Kwok-wai Cheung, and Jing Qin. MASAM: Multimodal adaptive sharpness-aware minimization for heterogeneous data fusion. In *International Conference on Learning Representations*, volume 2026, pp. 92384–92411, 2026.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Pinhan Fu, Xinyan Liang, Tingjin Luo, Qian Guo, Yayu Zhang, and Yuhua Qian. Core-structures-guided multi-modal classification neural architecture search. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 3980–3988, 2024a.

Pinhan Fu, Xinyan Liang, Yuhua Qian, Qian Guo, Zhifang Wei, and Wen Li. CoMO-NAS: Core-structures-guided multi-objective neural architecture search for multi-modal classification. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 9126–9135, 2024b.

Pinhan Fu, Xinyan Liang, Yuhua Qian, Qian Guo, Yayu Zhang, Qin Huang, and Ke Tang. Multi-scale features are effective for multi-modal classification: An architecture search viewpoint. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(2):1070–1083, 2025.

Yu Geng, Zongbo Han, Changqing Zhang, and Qinghua Hu. Uncertainty-aware multi-view representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7545–7553, 2021.

Alex Graves. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pp. 37–45. 2012.

Qian Guo, Xinyan Liang, Yuhua Qian, Zhihua Cui, and Jie Wen. A progressive skip reasoning fusion method for multi-modal classification. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 429–437, 2024.

Qian Guo, Quanchen Su, Xinyan Liang, Yuhua Qian, Nan Li, and Zhihua Cui. Multi-branch tree-based fusion neural architecture search with zero-cost screen for multi-modal classification. *IEEE Transactions on Image Processing*, 35:7192–7205, 2026.

Zongbo Han, Changqing Zhang, Huazhu Fu, and Joey Tianyi Zhou. Trusted multi-view classification with dynamic evidential fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2551–2566, 2023.

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

Qinglong Hu, Tong Xialiang, Mingxuan Yuan, Fei Liu, Zhichao Lu, and Qingfu Zhang. Multimodal llm-assisted evolutionary search for programmatic control policies. In *International Conference on Learning Representations*, volume 2026, pp. 137725–137762, 2026.

Bingbing Jiang, Junhao Xiang, Xingyu Wu, Wenda He, Libin Hong, and Weiguo Sheng. Robust adaptive-weighting multi-view classification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3117–3121, 2021.

Hamid Reza Vaezi Joze, Amirreza Shaban, Michael L Iuzzolino, and Kazuhito Koishida. MMTM: Multimodal transfer module for cnn fusion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13286–13296, 2020.

Junghyup Lee and Bumsub Ham. AZ-NAS: Assembling zero-cost proxies for network architecture search. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5893–5903, 2024.

Songtao Li and Hao Tang. Multimodal alignment and fusion: A survey. *International Journal of Computer Vision*, 134(3):1–31, 2026.

Xinyan Liang, Qian Guo, Yuhua Qian, Weiping Ding, and Qingfu Zhang. Evolutionary deep fusion method and its application in chemical structure recognition. *IEEE Transactions on Evolutionary Computation*, 25(5):883–893, 2021.

Xinyan Liang, Yuhua Qian, Qian Guo, Honghong Cheng, and Jiye Liang. AF: An association-based fusion method for multi-modal classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9236–9254, 2022.

Xinyan Liang, Pinhan Fu, Qian Guo, Keyin Zheng, and Yuhua Qian. DC-NAS: Divide-and-conquer neural architecture search for multi-modal classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13754–13762, 2024.

Xinyan Liang, Li Lv, Qian Guo, Bingbing Jiang, Feijiang Li, Liang Du, and Lu Chen. View-association-guided dynamic multi-view classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 5680–5688, 2025.

Fei Liu, Xialiang Tong, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. Evolution of Heuristics: Towards efficient automatic algorithm design using large language model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32201–32223, 2024.

Fei Liu, Yilu Liu, Qingfu Zhang, Xialiang Tong, and Mingxuan Yuan. EoH-S: Evolution of heuristic set using LLMs for automated heuristic design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37090–37098, 2026.

Wei Liu, Xiaodong Yue, Yufei Chen, and Thierry Denoeux. Trusted multi-view deep learning with opinion aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7585–7593, 2022.

Wei Liu, Yufei Chen, Xiaodong Yue, Changqing Zhang, and Shaorong Xie. Safe multi-view deep classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8870–8878, 2023.

Jindi Lv, Yanan Sun, Qing Ye, Wentao Feng, and Jiancheng Lv. A multiscale neural architecture search framework for multimodal fusion. *Information Sciences*, 679:121005, 2024.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Timothy J. O'Shea and Nathan West. Radio machine learning dataset generation with gnu radio. *Proceedings of the GNU Radio Conference*, 1(1), 2016.

Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8237, 2022.

Juan-Manuel Perez-Rua, Valentin Vielzeuf, Stephane Pateux, Moez Baccouche, and Fredericc Jurie. MFAS: Multimodal fusion architecture search. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6959–6968, 2019.

Yang Qin, Yanglin Feng, Yuan Sun, Dezhong Peng, Xi Peng, and Peng Hu. Deep information-balanced multimodal learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(8):9384–9396, 2026.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Kai Ruan, Yilong Xu, Ze-Feng Gao, Yang Liu, Yike Guo, Ji-Rong Wen, and Hao Sun. Discovering physical laws with parallel symbolic enumeration. *Nature Computational Science*, 6(1):53–66, 2026.

Amir Sadikov. Llm-guided evolutionary program synthesis for quasi-monte carlo design. In *International Conference on Learning Representations*, volume 2026, pp. 107605–107634, 2026.

Minju Seo, Jinheon Baek, Seongyun Lee, and Sung Ju Hwang. Paper2Code: Automating code generation from scientific papers in machine learning. In *International Conference on Learning Representations*, volume 2026, pp. 38867–38932, 2026.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1010–1019, 2016.

Jinhui Tang, Xiangbo Shu, Guo-Jun Qi, Zechao Li, Meng Wang, Shuicheng Yan, and Ramesh Jain. Tri-clustered tensor completion for social-aware image tag refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1662–1674, 2017.

Ondrej Tybl and Lukas Neumann. Training-free neural architecture search through variance of knowledge of deep network weights. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14881–14890, 2025.

Niki van Stein and Thomas Bäck. LLaMEA: A large language model evolutionary algorithm for automatically generating metaheuristics. *IEEE Transactions on Evolutionary Computation*, 29 (2):331–345, 2025.

Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12692–12702, 2020.

Yake Wei, Ruoxuan Feng, Zihe Wang, and Di Hu. Enhancing multimodal cooperation via sample-level modality valuation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27328–27337, 2024.

Jialang Xu, Chunbo Luo, Gerard Parr, and Yang Luo. A spatiotemporal multi-channel learning framework for automatic modulation recognition. *IEEE Wireless Communications Letters*, 9(10): 1629–1632, 2020.

Yihang Yin, Siyu Huang, and Xiang Zhang. BM-NAS: Bilevel multimodal neural architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8901–8909, 2022.

Feng Yu, Xiangyu Wu, Yang Yang, and Jianfeng Lu. Multimodal classification via total correlation maximization. In *International Conference on Learning Representations*, volume 2026, pp. 45101–45117, 2026.

Jiawei Zhang, Tiantian Wang, Zhixi Feng, and Shuyuan Yang. Toward the automatic modulation classification with adaptive wavelet network. *IEEE Transactions on Cognitive Communications and Networking*, 9(3):549–563, 2023.

Rui Zhang and Zhichao Lu. Rethinking code similarity for automated algorithm design with llms. In *International Conference on Learning Representations*, volume 2026, pp. 88723–88745, 2026.

Yifan Zhang, Congqi Cao, Jian Cheng, and Hanqing Lu. EgoGesture: A new dataset and benchmark for egocentric hand gesture recognition. *IEEE Transactions on Multimedia*, 20(5):1038–1050, 2018.

Zhi Zheng, Zhuoliang Xie, Zhenkun Wang, and Bryan Hooi. Monte carlo tree search for comprehensive exploration in LLM-based automatic heuristic design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 78338–78373, 2025.

A APPENDIX

This appendix provides the prompts used for fusion primitive composition generation, the detailed search procedure of LLMDFO-NAS, additional dataset and implementation details, and an example of a discovered fusion operator. Specifically:

- **A.1:** Prompt for initial primitive composition generation.
- **A.2:** Prompt for complementary recombination.
- **A.3:** Prompt for composition mutation.
- **A.4:** Detailed search procedure of LLMDFO-NAS.
- **A.5:** Datasets, evaluation protocols, and implementation details.
- **A.6:** Example of a fusion operator discovered on NTU RGB-D.

A.1 INITIAL PRIMITIVE COMPOSITION GENERATION

Initial primitive composition generation constructs fusion operators without using existing candidates. The LLM selects primitives from the validated library and organizes them into a valid computation graph. The corresponding prompt is presented in Fig. 5.

Prompt for Initial Composition Generation

I need help designing a new fusion composition for multimodal classification on {`dataset_name`}.
The validated fusion primitives are:
{`primitive_library`}
Please create a new composition using these primitives.
Return only one JSON object:
{`"nodes":[...],"output":"n_last"`}
The composition should contain 2–4 valid nodes. Each node may use the original inputs or the outputs of preceding nodes.
Do not use unregistered primitives or give additional explanations.

Figure 5: Prompt used for initial primitive composition generation.

The prompt provides the fusion task, validated primitive library, graph-size constraint, and required output format. Since no parent composition is supplied, the generated candidates provide the initial structural diversity for population-based search. Each returned composition is validated and compiled before evaluation.

A.2 COMPLEMENTARY RECOMBINATION

Complementary recombination generates a new candidate from two existing fusion compositions. Their structures, architecture-wise proxy losses, and structural complexities are provided to the LLM, as illustrated in Fig. 6.

The prompt encourages the LLM to preserve useful components from both parents while generating a different computation graph. Unlike conventional crossover, the recombination is performed at the semantic level of primitives and connections. The resulting composition must still satisfy the graph and execution constraints imposed by the framework.

A.3 COMPOSITION MUTATION

Composition mutation modifies one existing fusion composition. The parent composition and its evaluation information are used as the search context, as presented in Fig. 7.

Compared with complementary recombination, mutation performs a localized search around one parent. It may replace a primitive, modify an input reference, or reorganize a local computation path. The generated composition is subsequently validated, compiled, and evaluated by the framework.

Prompt for Complementary Recombination

I have two existing fusion compositions:
Composition 1:
{parent_composition_1}
Proxy Loss: {parent_loss_1}; Complexity: {parent_complexity_1}
Composition 2:
{parent_composition_2}
Proxy Loss: {parent_loss_2}; Complexity: {parent_complexity_2}
The validated fusion primitives are:
{primitive_library}
Please create a new complementary composition by recombining useful primitives and connections from the two compositions.
Return only one valid JSON composition:
{"nodes":[...],"output":"n_last"}
The new composition should be different from both parents. Do not give additional explanations.

Figure 6: Prompt used for complementary recombination.

Prompt for Composition Mutation

I have one existing fusion composition:
{parent_composition}
Proxy Loss: {parent_loss}; Complexity: {parent_complexity}
The validated fusion primitives are:
{primitive_library}
Please create an improved composition by modifying its primitive selection or connection structure.
Return only one valid JSON composition:
{"nodes":[...],"output":"n_last"}
The new composition must differ from the existing one and contain 2–4 valid nodes.
Do not give additional explanations.

Figure 7: Prompt used for composition mutation.

A.4 Detailed search procedure of LLMDFO-NAS

LLMDFO-NAS performs two consecutive search processes. fusion operator discovery first constructs a validated primitive library and searches for executable primitive compositions. Multimodal architecture search then uses the discovered operators to determine the modality-interaction structure. Algorithm 1 summarizes the complete procedure.

A.4.1 Primitive discovery and validation

Primitive discovery begins with an empty library. In each round, the LLM receives the fusion objective and an executable function template. It also receives the permitted tensor operations and previously accepted primitives. For each candidate, the LLM generates a name and a short description. It also assigns a functional category and a complexity descriptor. An executable implementation is returned with this metadata.

Each candidate first undergoes static validation. The framework checks its function signature and declared constraints. It then inspects the external dependencies and tensor operations. The abstract syntax tree is examined for unsupported or unsafe code. Candidates are rejected if they introduce trainable layers or stateful variables. Random operations and duplicate implementations are also prohibited.

Candidates that pass static validation are instantiated with different input cardinalities. Runtime tests verify the output shape and check for non-finite values. They also examine gradient propagation and deterministic execution. Finally, the framework checks the number of trainable parameters. Only candidates that pass all checks are added to $\mathcal{O}$. Rejected candidates and their failure reasons

Algorithm 1 Search procedure of LLMDFO-NAS

Require: LLM G, multimodal data $\mathcal{D}$, primitive target B, composition budget M, operator library size K, architecture generations T
Ensure: Operator library $\mathcal{F}^*$ and fusion architecture A^*

fusion operator discovery
$\mathcal{O} \leftarrow \textsc{DiscoverAndValidatePrimitives}(G, B)$
$\mathcal{P}_o \leftarrow \textsc{InitializeCompositions}(G, \mathcal{O})$
$\mathcal{P}_o \leftarrow \textsc{ValidateAndEvaluate}(\mathcal{P}_o)$
$\mathcal{A}_o \leftarrow \mathcal{P}_o$
while $|\mathcal{A}_o| < M$ **do**
  $(R_a, R_b) \leftarrow \textsc{SelectComplementaryPair}(\mathcal{P}_o)$
  $R_c \leftarrow \textsc{Recombine}(G, R_a, R_b, \mathcal{O})$
  $R_p \leftarrow \textsc{SelectParent}(\mathcal{P}_o)$
  $R_m \leftarrow \textsc{Mutate}(G, R_p, \mathcal{O})$
  $\mathcal{Q}_o \leftarrow \textsc{ValidateAndEvaluate}(\{R_c, R_m\})$
  $\mathcal{A}_o \leftarrow \mathcal{A}_o \cup \mathcal{Q}_o$
  $\mathcal{P}_o \leftarrow \textsc{ParetoSurvival}(\mathcal{P}_o \cup \mathcal{Q}_o)$
end while
$\mathcal{F}^* \leftarrow \textsc{DiversityAwareSelection}(\mathcal{A}_o, K)$
Multimodal architecture search
$\mathcal{P}_a \leftarrow \textsc{InitializeFusionTrees}(\mathcal{F}^*)$
$\mathcal{P}_a \leftarrow \textsc{TrainAndEvaluate}(\mathcal{P}_a, \mathcal{D})$
for $t = 1$ to T **do**
  $\mathcal{S}_a \leftarrow \textsc{TournamentSelect}(\mathcal{P}_a)$
  $\mathcal{Q}_a \leftarrow \textsc{GenerateOffspring}(\mathcal{S}_a)$
  $\mathcal{Q}_a \leftarrow \textsc{TrainAndEvaluate}(\mathcal{Q}_a, \mathcal{D})$
  $\mathcal{P}_a \leftarrow \textsc{ElitistSurvival}(\mathcal{P}_a \cup \mathcal{Q}_a)$
end for
$A^* \leftarrow \arg\max_{A \in \mathcal{P}_a} \mathrm{Fit}(A)$
return $\mathcal{F}^*, A^*$

are recorded. This feedback may be included in subsequent prompts to reduce repeated invalid proposals.

A.4.2 Composition encoding and compilation

Each fusion composition is represented as an ordered directed acyclic graph containing two to four primitive nodes. A node records a primitive identifier and a non-empty set of input references. An input reference may point to the original modality inputs or to the output of a preceding node.

The framework verifies that all primitives belong to $\mathcal{O}$, node identifiers follow the execution order, and every reference is available when the corresponding node is evaluated. The output must be produced by the last node, and every intermediate node must contribute to the final output.

A valid composition is compiled into an executable fusion function. The framework, rather than the LLM, performs this compilation. Invalid graphs, unavailable references, unused intermediate nodes, duplicate compositions, and functions that violate the input–output contract are rejected before proxy evaluation.

A.4.3 fusion operator population evolution

The initial operator population is produced by repeatedly applying the prompt in Appendix A.1. Each valid composition is evaluated within the same set of randomly sampled architecture contexts. Its architecture-wise proxy-loss vector and structural complexity are stored as its search attributes.

During complementary recombination, two parents are selected according to differences in their architecture-wise proxy losses and structural complexities. The prompt in Appendix A.2 instructs the LLM to reorganize useful primitives and connections from both parents.

Mutation selects one parent using rank-based sampling. The prompt in Appendix A.3 then modifies its primitive assignments or connection structure. Recombination promotes exploration between candidates with different behaviors, whereas mutation performs local refinement around an existing composition.

After evaluation, the parent and offspring compositions are merged. Empirical Pareto survival retains non-dominated candidates with respect to aggregated proxy loss and structural complexity. If the Pareto front does not fill the population, the remaining candidates are ordered by proxy loss and complexity.

The operator search terminates when the composition-evaluation budget M is exhausted. The final set is selected from the complete search archive. Candidates on the empirical Pareto front are considered first, duplicate primitive signatures are excluded, and diversity across functional categories is preferred. The top K candidates selected by this procedure form the discovered operator library $\mathcal{F}^*$.

A.4.4 MULTIMODAL ARCHITECTURE EVOLUTION

Each candidate multimodal architecture is represented as a fusion tree. Its leaf nodes refer to modality features, while its internal nodes are assigned operators from $\mathcal{F}^*$. The tree is decoded through post-order traversal so that every internal node is evaluated after its child representations become available.

The architecture population is initialized with randomly generated fusion trees. Each tree is decoded into a multimodal network and optimized on the training split. Its performance on the held-out evaluation split is used as fitness.

At each generation, binary tournament selection chooses parent architectures. Subtree crossover exchanges complete fusion substructures between two parents. Mutation modifies either the modality assigned to a leaf node or the discovered operator assigned to an internal node.

The offspring are decoded, trained, and evaluated using the same protocol as their parents. The parent and offspring populations are then merged, and higher-fitness architectures are retained. Elitist preservation prevents the best architecture found so far from being lost. After the final generation, the architecture with the highest held-out fitness is returned as A^*.

A.5 DATASETS AND IMPLEMENTATION DETAILS

This section provides additional information about the datasets, evaluation protocols, and fusion operator discovery configuration. The same preprocessing procedures and data partitions are used for LLMDFO-NAS and the corresponding controlled baselines.

A.5.1 DATASETS

We evaluate LLMDFO-NAS on five multimodal classification benchmarks. Their main statistics are summarized in Table 7. The reported sample and view counts correspond to the preprocessed data used in our experiments.

Table 7: Statistics of the datasets used in our experiments.

Dataset	#Samples	#Views	#Classes/labels	View or modality configuration
MM-IMDB	23,351	6	23	Four image views and two text views
EgoGesture	24,161	8	83	Four RGB views and four depth views
NUS-WIDE-128	23,438	7	10	Six visual descriptors and one tag view
NTU RGB-D	56,877	8	60	Four video views and four skeleton views
MV-RML2016.10A	220,000	5	11	Five deep signal feature views

MM-IMDB (Arevalo et al., 2017) is a multi-label movie-genre classification dataset containing image and text modalities. The preprocessed version used in our experiments contains 23,351 movies and 23 genre labels. Each movie is represented by four image feature views with dimensions

512, 512, 512, and 2,048, together with two text feature views with dimensions 300 and 600. This dataset evaluates fusion across modalities with substantially different semantic structures. We report the weighted F1-score as the evaluation metric.

EgoGesture (Zhang et al., 2018) is an egocentric gesture recognition dataset containing synchronized RGB and depth modalities. The preprocessed version used in our experiments contains 24,161 gesture samples from 83 classes. Each sample is represented by four RGB feature views and four depth feature views, with dimensions 512, 1,024, 2,048, and 2,048 for each modality. The complementary modalities describe the visual appearance and spatial structure of hand movements. Classification accuracy is used as the evaluation metric.

NUS-WIDE-128 (Tang et al., 2017) is a multi-view image classification benchmark containing heterogeneous visual and textual feature views. Following the subset used by MBTF-NAS (Guo et al., 2026), we use 23,438 single-label images from 10 categories. Each image is represented by seven feature views: color histogram (64 dimensions), color moments (225 dimensions), color correlogram (144 dimensions), edge-direction histogram (73 dimensions), wavelet texture (128 dimensions), bag-of-words features (500 dimensions), and tag features (1,000 dimensions). Classification accuracy is used as the evaluation metric.

NTU RGB-D (Shahroudy et al., 2016) is a large-scale human-action recognition dataset containing visual and skeleton-based representations. The preprocessed version used in our experiments contains 56,877 action samples from 60 classes. Each sample is represented by four video feature views with dimensions 512, 1,024, 2,048, and 2,048, and four skeleton feature views with dimensions 128, 256, 1,024, and 512. The dataset evaluates fusion between appearance, motion, and body-joint information. Classification accuracy is used as the evaluation metric.

MV-RML2016.10A is a multi-view modulation classification benchmark adopted following MBTF-NAS (Guo et al., 2026). It is derived from the RML2016.10A automatic modulation classification dataset (O'Shea & West, 2016), which contains 220,000 simulated I/Q signal samples from 11 modulation classes under different SNR conditions. Each sample is represented by five pre-extracted deep feature views obtained from AWN, MCLDNN, ResNet, LSTM, and GRU. Classification accuracy is used to evaluate performance under different SNR levels.

A.5.2 FUSION OPERATOR DISCOVERY CONFIGURATION

The primitive-library target B, composition-size range $[L_{\min}, L_{\max}]$, operator-population size P, number of evaluation architecture contexts J, zero-cost proxy batch size b_{zc}, composition-evaluation budget M, and retained operator library size K are treated as configurable hyperparameters. The values adopted in this work are summarized in Table 8.

The same evaluation architecture contexts are used across candidate compositions within one search run. This controlled evaluation reduces variation caused by changing architecture contexts. Once the budget M is exhausted, the search archive is ranked to select the final operator library of size K.

Invalid LLM outputs, unavailable node references, duplicate compositions, and compositions that fail runtime validation do not enter the population. These candidates are rejected and resampled before being counted toward the composition-evaluation budget M.

A.5.3 ARCHITECTURE SEARCH AND TRAINING CONFIGURATION

The architecture search and network-training settings follow the implementation described in Section 4.1. The same data partitions, input features, training settings, and search budgets are used for all candidate architectures and controlled comparisons. The main hyperparameters are summarized in Table 8.

A.6 EXAMPLE OF A DISCOVERED FUSION OPERATOR ON NTU RGB-D

We select F_5 as a representative fusion operator because it is the most frequently used operator in the downstream architecture search on NTU RGB-D. Figure 8 presents its equivalent implementation.

Table 8: Project-specific hyperparameter settings used in our experiments.

Configuration	Value
Common feature dimension	128
Primitive-library target (B)	15
Composition-size range ($L_{\min}$–$L_{\max}$)	2–4
Operator-population size (P)	5
Evaluation architecture contexts (J)	32
Zero-cost proxy batch size (b_{zc})	256
Composition-evaluation budget (M)	500
Retained operator library size (K)	5
Architecture population size	28
Architecture generations	20
Crossover probability	0.9
Mutation probability	0.2
Training batch size	256
Maximum training epochs	100
Early-stopping patience	20
Optimizer	Adam

```python
import tensorflow as tf

def fusion_operator(inputs, fused_nb_feats):
    # Node n0: max aggregation across modalities
    stacked = tf.stack(inputs, axis=0)
    n0 = tf.reduce_max(stacked, axis=0)

    # Node n1: Z-score standardization
    mean = tf.reduce_mean(n0, axis=0, keepdims=True)
    variance = tf.reduce_mean(
        tf.square(n0 - mean),
        axis=0,
        keepdims=True
    )
    n1 = (n0 - mean) / tf.sqrt(variance + 1e-7)

    return n1
```

Figure 8: Representative fusion operator F_5 discovered on NTU RGB-D. The operator combines max aggregation with feature standardization without introducing operator-specific trainable parameters.

The max-aggregation node preserves the strongest response in each feature dimension. The subsequent standardization node controls the scale of the fused representation and reduces differences between feature dimensions. The two primitives therefore provide complementary feature selection and normalization while preserving the common feature dimension.