Router-Suggest: Dynamic Routing for Multimodal Auto-Completion in Visually-Grounded Dialogs

Sandeep Mishra[1], **Devichand Budagam**[1], **Anubhab Mandal**[1], **Bishal Santra**[2], **Pawan Goyal**[1], **Manish Gupta**[2]
[1]IIT Kharagpur, India [2]Microsoft, India
sandeepmishraismyname@gmail.com, devichand579@gmail.com, anubhab.saie@gmail.com, bishalsantra@microsoft.com, pawangiitk@gmail.com, gmanish@microsoft.com

Abstract

Real-time multimodal auto-completion is essential for digital assistants, chatbots, design tools, and healthcare consultations, where user inputs rely on shared visual context. We introduce Multimodal Auto-Completion (MAC), a task that predicts upcoming characters in live chats using partially typed text and visual cues. Unlike traditional text-only auto-completion (TAC), MAC grounds predictions in multimodal context to better capture user intent. To enable this task, we adapt MMDialog and ImageChat to create benchmark datasets. We evaluate leading vision-language models (VLMs) against strong textual baselines, highlighting trade-offs in accuracy and efficiency. We present *Router-Suggest*, a router framework that dynamically selects between textual models and VLMs based on dialog context, along with a lightweight variant for resource-constrained environments. Router-Suggest achieves a 2.3× to 10× speedup over the best-performing VLM. A user study shows that VLMs significantly excel over textual models on user satisfaction, notably saving user typing effort and improving the quality of completions in multi-turn conversations. These findings underscore the need for multimodal context in auto-completions, leading to smarter, user-aware assistants. We make our code and benchmarks publicly available[1].

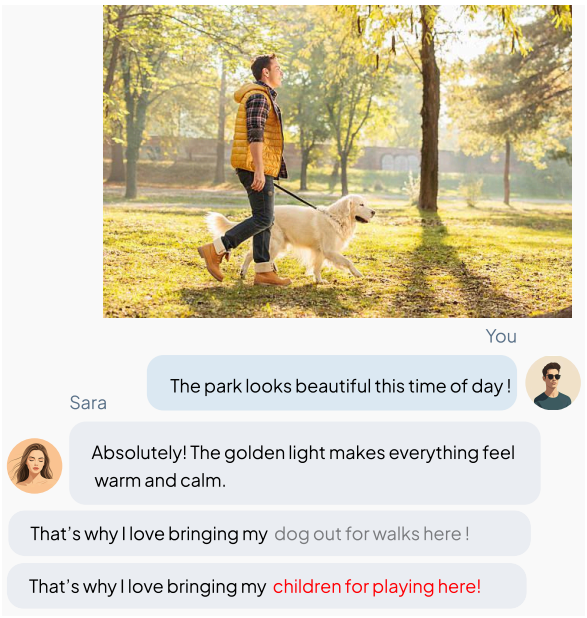


Figure 1: Example of multimodal auto-completion. Given the image context (*a man walking a golden retriever in a sunlit park*) and the partial user input *"That's why I love bringing my "*, the MAC model predicts "dog out for walks here!", while a text-based TAC model incorrectly predicts "children for playing here!". The MAC model prediction leverages both the textual prefix and visual context for a grounded completion.

1 Introduction

As conversations become increasingly multimodal, the ability to predict what users will type next, while understanding both text and visuals, can transform digital assistants from reactive tools into truly intuitive partners. Conversational systems are increasingly used in both consumer and enterprise contexts through digital assistants, service bots, AI tools, and productivity copilots, where efficient and contextually relevant interactions are critical. Systems like ChatGPT (OpenAI, 2022) and Microsoft Copilot[2] exemplify this trend by offering intelligent, context-aware responses. Yet, as these systems evolve, user interactions increasingly include images to clarify intent, share visuals, or seek help, such as screenshots for tech support, product photos in e-commerce, design drafts in collaboration, or medical scans in telehealth. These raise new opportunities and challenges for predictive text technologies.

To streamline such interactions, inline text auto-completion (TAC) predicts user inputs in real-time

[1]https://github.com/devichand579/MAC

[2]https://copilot.microsoft.com/

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track), pages 145–156

using typed prefixes and dialog context. Unlike traditional query auto-completion (QAC) (Bar-Yossef and Kraus, 2011), which presents a (dropdown) ranked list of full query suggestions, TAC offers a single completion as part of the input text field, thereby minimizing cognitive and interaction costs. However, TAC remains underdeveloped for conversational systems requiring real-time predictions in multi-turn dialogs, as most existing solutions focus on list-based QAC. For multimodal dialogues, where intent depends on both text and visuals, there exists no inline auto-completion system. Hence, we introduce Multimodal Auto-Completion (MAC), which extends TAC by using both linguistic and visual contexts to predict user input.

MAC poses distinct challenges: (i) *disambiguation under partial input*, where similar textual prefixes can warrant different completions conditioned on the image; (ii) *modality alignment*, requiring the model to ground predictions in visually salient cues; and (iii) *latency-efficiency trade-offs*, since vision-language inference can be substantially slower than text-only models in interactive systems.

For instance (see Figure 1, with an image of a man and a 'golden retriever' in a park, if a user types *"That's why I love bringing my "* a TAC model might suggest *"children here"* or *"wife here"* ignoring the visual cue. Conversely, MAC uses the image to complete the input as *"dog here"* illustrating the effectiveness of multimodal grounding.

Our key contributions are as follows:

- **Task Definition and Benchmarking:** We define MAC as predicting inline user input from partially typed text and multimodal dialog history. To support systematic evaluation, we construct standardized benchmarks by adapting two widely used multimodal dialog datasets: MMDialog (Feng et al., 2023) and ImageChat (Shuster et al., 2020), with rigorous filtering to ensure strong visual relevance.
- **Model Benchmarking:** We conduct a comprehensive evaluation of recent vision-language models (VLMs) like MiniCPM-V (Yao et al., 2024), PaliGemma (Beyer et al., 2024), Qwen2-VL (Yang et al., 2024) alongside textual baselines like Most Popular Completion (MPC) (Bar-Yossef and Kraus, 2011) and Query Blazer (QB) (Kang et al., 2021) on the MAC task, highlighting key trade-offs in multimodal understanding and completion quality.
- **Router-Suggest:** We present a dynamic routing framework that decides, at each character, whether to use a lightweight textual model or one of the more expressive VLMs, based on the visual significance of the dialog context.
- **User Study:** We perform a user study to evaluate the MAC's practical effectiveness by quantifying Typing Effort Saved (TES) and user satisfaction. Results demonstrate substantial gains over text-only methods. We release our code and benchmarks[1].

2 Related work

Query Auto-Completion (QAC): QAC has long been a core component of search systems, improving efficiency and reducing query formulation effort (Bast and Weber, 2006). Traditional approaches exploit signals such as popularity-based rankings (Whiting et al., 2013), spatial and temporal patterns (Backstrom et al., 2008), and session-level co-occurrence statistics (Bar-Yossef and Kraus, 2011). Implementations range from classical machine learning (Di Santo et al., 2015; Sordoni et al., 2015) to modern neural architectures, including LSTMs (Wang et al., 2020) and transformer-based models like BERT and BART (Mustar et al., 2020).

Text-only Auto-Completion (TAC): TAC, or *inline auto-completion*, also called *ghosting* (Ramachandran and Murthy, 2019), offers a single continuation within the input field, unlike QAC's ranked suggestions. This design suits conversational contexts where dropdowns disrupt flow. Early neural methods used subword language models (Kim, 2019) for token-level efficiency, while transformer models such as GPT-2 have been fine-tuned for next-phrase prediction in structured domains (Lee et al., 2021). More recently, reinforcement learning approaches (Chitnis et al., 2024; Li et al., 2024) have emerged for TAC. Additional literature is provided in Appendix A.

Research on dialog systems largely focuses on next-utterance prediction, whereas inline auto-completion, i.e., predicting user input mid-turn, remains underexplored. This challenge intensifies in multimodal contexts where images influence intent. Existing models prioritize full-turn responses, neglecting real-time mid-turn predictions. We introduce MAC to bridge this gap, gen-

erating grounded continuations of partial inputs using dialog history and visual context, linking full-turn response generation with real-time typing assistance in vision-language interfaces.

3 Methods for MAC

3.1 The MAC Task Definition

The MAC task aims to generate a contextually appropriate continuation of a user's partially typed input by leveraging both textual and visual dialog history. The model input comprises: (1) a textual prefix $p \in \mathcal{V}^{\leq T}$, representing the user's partially typed message, where $\mathcal{V}$ is the vocabulary and T is the maximum prefix length; and (2) a multimodal dialog history of k previous utterances, $\mathcal{H}_{\text{mm}} = \{(u_1, m_1), (u_2, m_2), \ldots, (u_k, m_k)\}$, where $u_i \in \mathcal{V}^{l_i}$ is a prior utterance of length $l_i \leq T$ and $m_i \in \mathcal{M}$ is an optional associated modality such as an image.

The model outputs a textual continuation c such that the concatenated sequence $[p; c]$ forms a fluent and contextually coherent message with respect to the multimodal dialog context $\mathcal{H}_{\text{mm}}$. We learn model parameters θ that maximize the conditional likelihood of c given the prefix and multimodal context:

$$\theta^* = \arg\max_{\theta} P(c \mid p, \mathcal{H}_{\text{mm}}; \theta)$$

At inference, given a new prefix p and context $\mathcal{H}_{\text{mm}}$, the model generates a prediction $\hat{c}$ via:

$$\hat{c} = \arg\max_{c} P(c \mid p, \mathcal{H}_{\text{mm}}; \theta^*)$$

This formulation enables real-time autocompletion during multimodal interactions, improving typing efficiency and coherence in visually grounded conversations.

3.2 Benchmark Construction for MAC Evaluation

Progress on multimodal auto-completion has been limited by the absence of standardized benchmarks. Existing multimodal dialog datasets rarely emphasize visual context as a key driver of user intent. To address this, we adapt two prominent multimodal dialog datasets: MMDialog (Feng et al., 2023) and ImageChat (Shuster et al., 2020) for the MAC task.

We utilize GPT-4V (OpenAI, 2023) to filter datasets, selecting dialogs where images are essential for predicting the user's next input, ensuring visual grounding. We focus on single-image conversations to allow accurate visual relevance assessment without hallucinations. MMDialog (MMDD) (Feng et al., 2023) includes domain-specific conversations enhanced with visuals like movie posters and scene stills; we select cases where images significantly influence dialog flow. ImageChat (Shuster et al., 2020) offers open-domain conversations linked to images.

Dataset	Split	# Dialogs	Avg Uttr Len	Avg # Uttr
MMDD	Train	13,182	51.81	11.97
	Test	893	50.96	12.80
ImageChat	Train	186,724	49.32	1.91
	Test	9,994	49.44	3.00

Table 1: MAC Benchmark Dataset statistics after preprocessing. Length is measured in characters.

Following the filtering and formatting steps, the curated versions of MMDialog and ImageChat form robust MAC benchmarks. Table 1 summarizes the key statistics: MMDialog features longer dialogs with more utterances per conversation, while ImageChat contains shorter, image-grounded exchanges. Additional details appear in Appendix B.

3.3 Models for the MAC Task

We benchmark both textual models and VLMs, ranging from traditional retrieval-based approaches to modern VLMs, for the MAC task. Appendix C.1 lists additional information about these models.

Textual Models: These include trie-based methods such as *Most Popular Completion (MPC)* (Bar-Yossef and Kraus, 2011), *MPC++* (Bar-Yossef and Kraus, 2011) and n-gram based model *QueryBlazer (QB)* (Kang et al., 2021).

Vision Language Models (VLMs): These include MiniCPM-V (Yao et al., 2024), PaliGemma (3B) (Beyer et al., 2024) and Qwen2-VL (Wang et al., 2024).

3.4 The Proposed Router-Suggest Framework

Textual models and VLMs vary significantly in terms of their latency and accuracy. To balance these trade-offs, we present *Router-Suggest*, which adaptively selects the optimal model per prefix. we frame routing as a classification problem, where a lightweight neural router predicts the best model based on input complexity. The average system latency with a router configuration

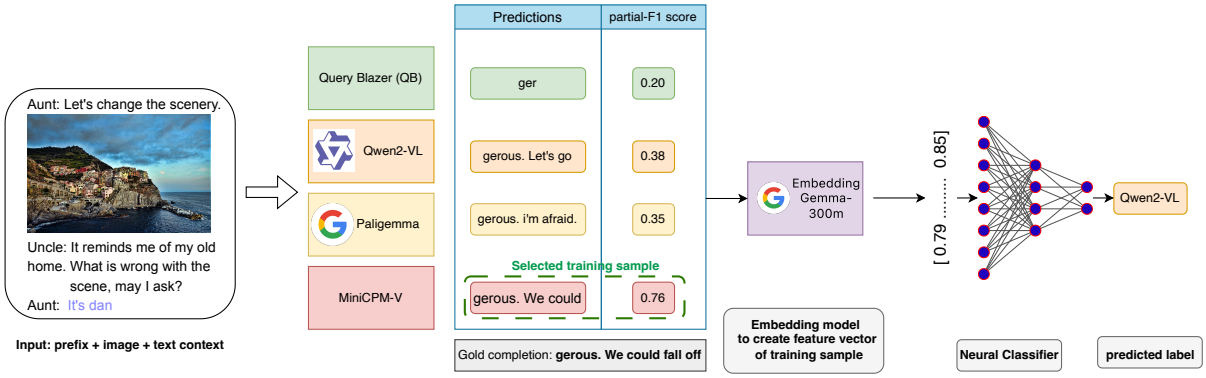


Figure 2: During router training, VLMs receive the entire input context, while the textual QB model only uses the prefix. We calculate partial-F1 scores of predictions to determine the gold label. Further, we generate a feature vector for the input prefix of the training sample using EmbeddingGemma-300m for training the neural classifier.

with n MAC models can be computed as:

$$L_{\text{total}} = L_{\text{Router}} + \sum_{i=1}^{n} p_i \cdot L_i$$

where, p_i is the probability of triggering the i-th MAC model. We employ a lightweight neural classifier as a router to minimize the router's latency overhead, i.e., $L_{\text{Router}} \approx 0$. For router training (See Fig. 2), for each training (prefix, completion) sample, we use 768D EmbeddingGemma-300m (Vera et al., 2025) representations of input prefixes as features. To train the router, we obtain the ground truth optimal model for each sample as follows. First, we generate completions for an input prefix using all the models. The model with the highest partial-F1 score against the true completion is selected as the ground truth optimal model.

To incorporate latency-awareness, we perform cost-sensitive training of the router. For C candidate models (and hence number of classes for router) and a batch of N samples, let p_s^m denote the predicted probability for model $m \in [1, c]$ and sample $s \in [1, N]$, and c^m its cost proportional to its average latency. Let y_s denote the true class label for sample s. Then we compute the cross entropy loss for the batch as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \log p_s^{y_s}$$

For each sample s, the expected cost is the probability-weighted average of per-class costs:

$$\mathbb{E}_{p_s}[\text{cost}] = \sum_{m=1}^{C} p_s^m c^m$$

Averaged across the batch:

$$\mathcal{L}_{\text{Cost}} = \frac{1}{N} \sum_{s=1}^{N} \sum_{m=1}^{C} p_s^m c^m$$

The overall loss $\mathcal{L}$ combines accuracy and cost-awareness in a single objective. $\mathcal{L}_{\text{CE}}$ encourages correct classification, while $\mathcal{L}_{\text{Cost}}$ penalizes predictions with higher expected costs.

$$\mathcal{L} = (1 - \lambda)\, \mathcal{L}_{\text{CE}} + \lambda\, \mathcal{L}_{\text{Cost}}$$

The trade-off parameter λ enables a controlled compromise between accuracy and cost efficiency. The routing framework is model-agnostic, integrating the text-based TAC and MAC models with different latency-accuracy trade-offs. This ensures efficient, real-time deployment of multimodal completion systems with high completion quality. At test time, we select the model having the highest probability predicted by the router.

4 Experiments and Results

4.1 Evaluation Metrics

Standard NLG metrics like BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) are unsuitable for MAC tasks, which require inline continuation of user input. These metrics focus on sequence overlap, but MAC needs accuracy in continuing user input to avoid cognitive load and ensure acceptance. Traditional QAC metrics such as top-k accuracy or Mean Reciprocal Rank (MRR) assume a ranked list of suggestions, which is incompatible with the single, inline nature of MAC. These approaches also fail to account for the real-time aspect of interaction, when and how often suggestions are triggered.

Type	Model	TR	SM	PR-P	PR-R	PR-F1	|Pred|	TES
MMDD								
Textual	MPC	0.1991	0.0000	0.0782	0.0676	0.0725	**40.6**	0.0015
	MPC++	0.5651	0.0332	0.1831	0.1303	0.1525	29.4	0.0430
	QB	0.9220	0.0426	**0.3498**	0.1287	0.1892	8.9	0.1724
VLMs	MiniCPM-V	**0.9898**	**0.1182**	0.3362	**0.2423**	**0.2800**	21.1	**0.2136**
	PaliGemma	0.9880	0.0972	0.2896	0.2145	0.2470	20.3	0.2030
	Qwen2-VL	0.9891	0.1034	0.2950	0.2223	0.2532	18.8	0.1844
ImageChat								
Textual	MPC	0.2749	0.0007	0.1120	0.0685	0.0845	**27.7**	0.0030
	MPC++	0.6728	0.0341	0.2080	0.1202	0.1523	17.3	0.0371
	QB	0.9604	0.0373	0.3065	0.1225	0.1755	5.9	0.0955
VLMs	MiniCPM-V	**0.9892**	**0.0715**	**0.3128**	**0.2205**	**0.2586**	16.1	0.1246
	PaliGemma	0.9881	0.0616	0.2850	0.1996	0.2348	16.7	0.1148
	Qwen2-VL	0.9889	0.0577	0.2931	0.1971	0.2356	16.2	**0.1422**

Table 2: Performance metrics on **unseen prefixes** of the MMDD (top) and ImageChat (bottom), organized by type (Textual vs. VLMs). |Pred|=Avg Pred Len. TES is calculated relative to ground truth completions.

To address these limitations, we utilize a set of MAC-specific metrics from (Mishra et al., 2025), including Trigger Rate (TR), Syntactic Match (SM), Partial Recall (PR-R), Partial Precision (PR-P), Partial F1 (PR-F1), and Typing Effort Saved (TES). These metrics provide a precise assessment of the usability, accuracy, and efficiency of real-time multimodal chat system completions.

Let s_i be the model's suggestion for instance i, g_i be the ground truth continuation for instance i and N denote the number of utterances in the evaluation dataset.

- **Syntactic Match (SM):** SM measures the percentage of model-generated completions that exactly match the ground truth continuation. A completion is considered a syntactic match if it is identical to the reference output when suggestions are shown.

$$\text{SM} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(s_i = g_i)$$

  where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the condition is true, and 0 otherwise.

- **Partial Recall (PR-R):** PR-R quantifies the average percentage of ground truth characters that overlap with the predicted completion, starting from the beginning. It reflects how much of the true continuation the model successfully recovered as a prefix.

$$\text{Recall}_p = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{len}(\text{prefix\_match}(s_i, g_i))}{\text{len}(g_i)}$$

  where $\text{prefix\_match}(s_i, g_i)$ returns the longest common prefix between s_i and g_i.

- **Partial Precision (PR-P):** PR-P quantifies the average percentage of predicted characters that overlap with the ground truth continuation, starting from the beginning. It reflects how much of the predicted completion is actually correct as a prefix.

$$\text{Precision}_p = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{len}(\text{prefix\_match}(s_i, g_i))}{\text{len}(s_i)}$$

- **Trigger Rate (TR)**: TR measures how frequently a suggestion is shown to the user, based on a predefined confidence threshold. It is calculated as the ratio of the number of times a suggestion was triggered to the total number of characters typed by the user.

$$\text{TR} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{\# suggestions triggered}_i}{\text{\# total characters typed}_i}$$

- **Typing Effort Saved (TES)**: TES measures the proportion of ground truth characters saved, i.e., the overlap between prediction and target continuation. TES can be interpreted as a normalized keystroke saving rate across the entire dataset.

$$\text{TES} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\text{\# characters actually typed}_i}{\text{total utterance length}_i}\right)$$

These metrics assess several aspects of the MAC task: *accuracy* (assessed through PR-P, PR-R and the partial-F1, which represents the harmonic mean of PR-P and PR-R), *usability* (via TES and TR), and *syntactic fluency* (via SM). Collectively, they enable a more comprehensive understanding of model behavior than traditional metrics and are essential for benchmarking MAC systems.

4.2 Finetuning Setup

We perform two pre-processing steps (unrolling and splitting) on the dialog datasets to format them into the standard structure desired: *context + image + prefix + completion*. In the unrolling step, the dialog is progressively built by appending each utterance one at a time, resulting in an increasingly rich context. In the splitting step, the entire conversation is preserved up to the penultimate utterance. The last utterance is then randomly divided into two segments: the first serves as the prefix, and the second becomes the target completion to be predicted.

We trained our text models using default settings, closely following QB (Kang et al., 2021), which includes a 4,096-token vocabulary that covers 99.95% of characters. Subsequently, an 8-gram language model was constructed with pruning. Models utilizing both MPC (Bar-Yossef and Kraus, 2011) and MPC++ (Bar-Yossef and Kraus, 2011) were implemented with their standard configurations. For the VLM-based models, we conducted training over 5 epochs, using a batch size of 8 per device and a learning rate of 0.0001. This process employed mixed-precision (`FP16`) training. LoRA adapters, with a rank of 8, were incorporated into all linear layers and subjected to a 0.05 dropout rate. Throughout this, we maintained the base model in a frozen state, updating only the LoRA parameters.

4.3 Performance on MAC Benchmarks

Table 2 reveals a clear performance gap between text models and VLMs on unseen prefixes across both MMDD and ImageChat datasets. Text models collapse in MMDD, with MPC showing nearly zero Syntactic Match ($SM = 0$) and TES (0.0015), indicating severe overfitting. Even the enhanced MPC++ offers limited gains, while QB generalizes modestly but still deteriorates in multimodal contexts. In contrast, VLMs maintain consistently high Trigger Rates ($TR \approx 0.99$) and stable PR metrics, leveraging multimodal grounding for robust contextual completions. MiniCPM-V achieves the best overall TES (0.2136) and balanced PR scores while generating shorter, more efficient completions ($\approx$ 18-22 characters) compared to verbose outputs from text models (e.g., MPC $|Pred| = 40.6$).

On ImageChat, the gap narrows as text models degrade less sharply, but VLMs still outperform, sustaining higher TES and smoother precision–recall trade-offs. Overall, VLMs demonstrate superior generalization and adaptability in unseen multimodal scenarios. Please see Appendix C.2 for results on seen prefixes on both benchmarks.

4.4 Evaluation of Router-Suggest

Table 3 presents the latency-performance tradeoff of individual models alongside Router-Suggest. The absolute latencies for all VLMs are determined through inference using vLLM (Kwon et al., 2023) as the inference engine, applied to a representative dataset consisting of prefixes from both MMDD and ImageChat. We conducted a joint hyperparameter and architectural search for router configurations across various λ (See Fig. 3) to optimize performance and latencies, as detailed in Appendix C.3.

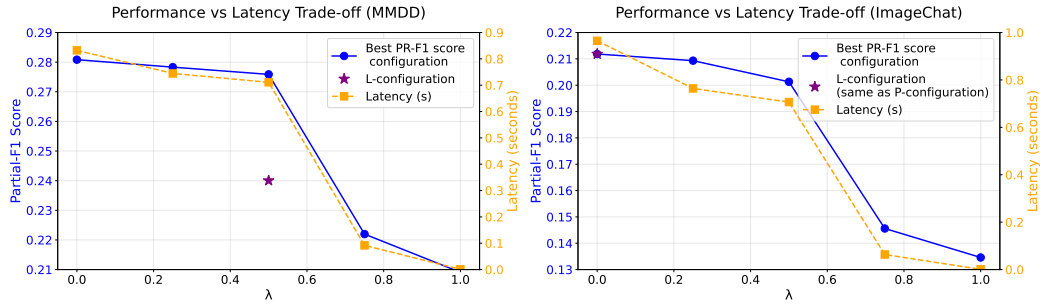
Router-Suggest with 4 models (QB, Qwen2-VL, PaliGemma and MiniCPM-V) needs $\sim$25GB memory on an Nvidia L40 GPU for inference. For constrained environments, we also experiment with a router configuration with just 2 models (QB, Qwen2-VL), requiring only $4GB$ GPU memory. We refer to router configurations as Router-4 and Router-2, respectively. Further, after joint hyperparameter and architecture search, we choose 2 configurations: L and P. Router-L corresponds to the hyperparameter configuration that leads to minimum latency with performance (PR-F1) close to the best model. Router-P corresponds to the hyperparameter configuration that leads to maximum performance (PR-F1). We also compute the oracle performance of the Router-4 configuration, where the best perfroming model is always chosen for every prefix.

Router-4-L achieves near-competitive performance of the best-performing individual model with minimal latency, while Router-4-P offers the highest PR-F1 score. Thus, Router-Suggest models improve PR-F1 and syntactic match, reducing latency compared to high-capacity models, showcasing lightweight routing's efficiency. On MMDD, Router-4-L matched MiniCPM-V's PR-F1 score at $5\times$ faster response time. Router-4-P achieved a PR-F1 of 0.281, close to the 0.356 upper bound at one-third the latency of MiniCPM-V. On ImageChat, routing maintains accuracy with minimal time overhead, highlighting scalability and practical benefits.

Router-2-L achieves near-optimal PR-F1 compared to Qwen2-VL (0.248 on MMDD, 0.192 on ImageChat) with substantially reduced latency

Model	MMDD			ImageChat		
	PR-F1	SM	Time (s)↓	PR-F1	SM	Time (s)↓
Individual Models						
MiniCPM-V	0.247	0.116	2.080	**0.223**	**0.067**	2.080
PaliGemma	0.216	0.097	1.490	0.199	0.057	1.490
QB	0.209	0.102	**0.001**	0.135	0.036	**0.001**
Qwen2-VL	0.222	0.101	0.733	0.197	0.053	0.733
Router-Suggest						
Router-4-L	0.240	0.110	0.351	0.212	0.056	0.966
Router-4-P	**0.281**	**0.135**	0.832	0.212	0.056	0.966
Router-2-L	0.240	0.109	0.170	0.196	0.053	0.288
Router-2-P	0.261	0.122	0.271	0.196	0.053	0.288
Router-4-Max (Oracle)	0.356	0.195	–	0.281	0.090	–

Table 3: Performance and latency comparison of individual models and Router-Suggest configurations across MMDD and ImageChat.



(a) (b)

Figure 3: Different router configurations for Router-4 at different λ and their latency vs PR-F1 score tradeoff for (a) MMDD and (b) ImageChat.

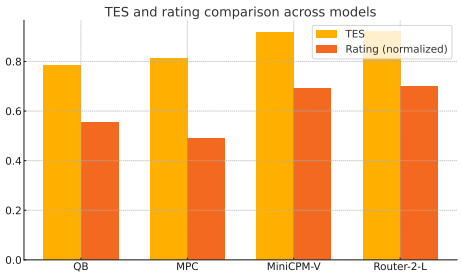


Figure 4: Comparison of mean TES and user ratings (normalized) for various models. TES is calculated relative to the final text approved by the user at the moment the rating is submitted.

compared to Qwen2-VL and a speedup $10\times$ compared to the best-performing model (MiniCPM-V), demonstrating effective lightweight routing.

5 User Study

We developed a platform where anonymous users can participate in completing conversations initialized from randomly selected samples of the MMDD and ImageChat datasets. During interactions, users engage with a randomly selected model (QB, MPC, or MiniCPM-V) without knowing the specific model, thus minimizing bias. Users assess the system's completion on a scale from 0 to 9, where 9 represents the most satisfactory and well-aligned completion and 0 indicates a completely unaligned, poor, or absent completion. TES calculation is based on the final user query at the moment the rating is submitted. Our study encompasses 190 sessions, distributed as follows: 53 with MPC, 47 with QB, 45 with MiniCPM-V and 45 with Router-2-L.

Figure 4 illustrates a strong positive relationship between TES and user ratings across models. The visual trend confirms that as TES increases, user ratings also rise. These TES scores are significantly higher than the offline TES scores (Table 2). This is expected because, in interactive settings, users often adapt their typed continuations based on the system's suggestions. As a result, the 'ground truth' becomes partially influenced by the model itself, naturally inflating agreement metrics such as TES. MiniCPM-V consistently outperforms the text models, achieving the highest TES and an unnormalized user rating and router-2-L also achieved similar scores. This demonstrates that VLMs not only achieve higher TES but also deliver a more stable and satisfying user experience than the textual counterparts.

6 Conclusion

We propose Multimodal Auto Completion (MAC), a novel task for predicting user input in visually grounded conversations, along with standardized benchmarks from MMDialog and ImageChat and an evaluation protocol designed for inline auto-completion. Experiments reveal textual models excel with known prefixes but struggle with new ones, whereas VLMs maintain high trigger rates and better TES and robustness in new conditions. Router-Suggest selectively engages VLMs, providing competitive partial-F1 as the best models with 2.3-10$\times$ speedup. We also provide a low-resource setup for Router-Suggest. A user study confirms TES as a reliable user satisfaction measure, aligning with subjective ratings and shows that VLM completions better meet user expectations compared to outputs from textual models. Overall, the results show that visually grounded completions can greatly reduce typing effort and improve perceived usefulness in interactive settings.

7 Limitations

The MAC benchmarks, adapted from MMDialog and ImageChat using GPT-4V filtering, may introduce selection bias toward visually explicit cases and lack linguistic diversity. Current datasets only cover single-image contexts, limiting generalization to real-world multimodal settings with evolving or multiple visuals. Router-Suggest, though effective in reducing latency, relies on embedding-based heuristics that may degrade under domain shift and lacks interpretability in its routing choices.

8 Ethical Considerations

The MAC benchmark is built using automated relevance filtering (GPT-4V) and curated public corpora, which may introduce noisy labels, annotation biases, privacy concerns, and hallucination risks. The user study relies primarily on TES and a small user pool, which may overlook key factors: TES can fail to capture subtle misinformation, cultural or demographic mismatches, and sampling choices can introduce biases that limit generalizability. Additionally, the router's invocation patterns raise fairness and cost-allocation concerns, as it may disproportionately route certain input types or user groups to more compute-intensive MAC models, leading to unequal latency, computational cost, or quality of experience.

References

Lars Backstrom, Jon Kleinberg, Ravi Kumar, and Jasmine Novak. 2008. Spatial variation in search engine queries. In *Proceedings of the 17th international conference on World Wide Web*, pages 357–366.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Ziv Bar-Yossef and Naama Kraus. 2011. Context-sensitive query auto-completion. In *Proceedings of the 20th international conference on World wide web*, pages 107–116.

Holger Bast and Ingmar Weber. 2006. Type less, find more: fast autocompletion search with a succinct index. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 364–371.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, and 16 others. 2024. Paligemma: A versatile 3b vlm for transfer. *Preprint*, arXiv:2407.07726.

Rohan Chitnis, Shentao Yang, and Alborz Geramifard. 2024. Sequential decision-making for inline text autocomplete. *arXiv preprint arXiv:2403.15502*.

Giovanni Di Santo, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2015. Comparing approaches for query autocompletion. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 775–778.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Jiazhan Feng, Qingfeng Sun, Can Xu, Pu Zhao, Yaming Yang, Chongyang Tao, Dongyan Zhao, and Qingwei Lin. 2023. MMDialog: A large-scale multi-turn dialogue dataset towards multi-modal open-domain conversation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7348–7363, Toronto, Canada. Association for Computational Linguistics.

Young Mo Kang, Wenhao Liu, and Yingbo Zhou. 2021. Queryblazer: Efficient query autocompletion framework. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1020–1028.

Gyuwan Kim. 2019. Subword language model for query auto-completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5022–5032, Hong Kong, China. Association for Computational Linguistics.

Fanheng Kong, Peidong Wang, Shi Feng, Daling Wang, and Yifei Zhang. 2024. TIGER: A unified generative model framework for multimodal dialogue response generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16135–16141, Torino, Italia. ELRA and ICCL.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model

serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Dong-Ho Lee, Zhiqiang Hu, and Roy Ka-Wei Lee. 2021. Improving text auto-completion with next phrase prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4434–4438, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bolun Li, Zhihong Sun, Tao Huang, Hongyu Zhang, Yao Wan, Ge Li, Zhi Jin, and Chen Lyu. 2024. Ircoco: Immediate rewards-guided deep reinforcement learning for code completion. *Proceedings of the ACM on Software Engineering*, 1(FSE):182–203.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. Workshop on Text Summariation Branches Out, Post-Conference Workshop of ACL 2004*, pages 74–81.

Bo Liu, Lejian He, Yafei Liu, Tianyao Yu, Yuejia Xiang, Li Zhu, and Weijian Ruan. 2022. Transformer-based multimodal infusion dialogue systems. *Electronics*, 11(20).

Sandeep Mishra, Anubhab Mandal, Bishal Santra, Tushar Abhishek, Pawan Goyal, and Manish Gupta. 2025. Chat-ghosting: A comparative study of methods for auto-completion in dialog systems. *Preprint*, arXiv:2507.05940.

Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. 2020. Using bert and bart for query suggestion. In *Joint Conference of the Information Retrieval Communities in Europe*, volume 2621. CEUR-WS. org.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt. Accessed July 2025.

OpenAI. 2023. Gpt-4v(ision) technical work and authors. https://openai.com/contributions/gpt-4v/.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Lakshmi Ramachandran and Uma Murthy. 2019. Ghosting: contextualized query auto-completion on amazon search. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 1377–1378.

Kurt Shuster, Samuel Humeau, Antoine Bordes, and Jason Weston. 2020. Image-chat: Engaging grounded conversations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2414–2429, Online. Association for Computational Linguistics.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international on conference on information and knowledge management*, pages 553–562.

Qingfeng Sun, Yujing Wang, Can Xu, Kai Zheng, Yaming Yang, Huang Hu, Fei Xu, Jessica Zhang, Xiubo Geng, and Daxin Jiang. 2022. Multimodal dialogue response generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2854–2866, Dublin, Ireland. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, and 70 others. 2025. Embeddinggemma: Powerful and lightweight text representations. *Preprint*, arXiv:2509.20354.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *Preprint*, arXiv:2409.12191.

Sida Wang, Weiwei Guo, Huiji Gao, and Bo Long. 2020. Efficient neural query auto completion. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2797–2804.

Stewart Whiting, Andrew James McMinn, and Joemon M Jose. 2013. Exploring real-time temporal query auto-completion. In *DIR*, pages 12–15.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Hee Suk Yoon, Eunseop Yoon, Joshua Tian Jin Tee, Kang Zhang, Yu-Jung Heo, Du-Seong Chang, and Chang D Yoo. 2024. Bi-mdrg: Bridging image history in multimodal dialogue response generation. In *European Conference on Computer Vision*, pages 378–396. Springer.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986.

A Additional Related Work

Recent work in multimodal dialog systems has focused on generating context-aware responses by integrating both visual and textual dialog history inputs. Sun et al. (2022) proposed DIVTER, a dual-channel model that enables text or image response generation under low-resource conditions by decoupling textual and visual training. Kong et al. (2024) introduced TIGER, a unified transformer-based framework capable of producing text, image, or mixed-modal responses by dynamically selecting the output modality. Yoon et al. (2024) presented BI-MDRG, which incorporates visual history across dialog turns to maintain object consistency and support grounded response generation. Earlier approaches, such as MAGIC and MATE, applied transformer-based cross-modal attention mechanisms (Liu et al., 2022) to generate visually coherent textual responses, highlighting the role of structural alignment between modalities.

B Benchmark Construction

B.1 Relevance filtering using GPT-4V

To ensure that images meaningfully contribute to the dialog, we employ GPT-4V (OpenAI, 2023) as an automatic discriminator to assess the relevance of each image-dialog pair, using the prompt template illustrated in Figure 5. Each sample is rated on a standardized 5-point scale: **1** = Contradictory, **2** = Ignored, **3** = Marginally relevant, **4** = Clearly useful, **5** = Critical for understanding.

Only samples receiving a relevance score of **4** or **5** are retained in the final benchmark to ensure strong visual grounding and eliminate noisy or irrelevant pairs. Figure 6 illustrates examples identified as highly image-relevant by GPT-4V, highlighting the kinds of interactions that demand grounded multimodal understanding, central to the challenge of MAC. Following the filtration process, over 66% of the samples were removed from the datasets.

Prompt Template

You are a discriminator model who will decide if the following hold:

1. The dialog is relevant to the image.
2. The image fits the context and is accounted for in the following utterances.
3. The image and the dialog are coherent.
4. The image can be used for autocompletion of following utterances.
5. The image should not be the last utterance because it is of no use then.

The user will provide the dialog starting from when the image was shared and including up to 3 subsequent utterances. Carefully assess how much the image contributes to the conversation. Think through the following questions step by step before assigning a score:
Step-by-step Analysis:

1. Provide a caption for the image (regardless of the conversation).
2. Is the image misleading? Does it contradict or confuse the dialog?
   If yes, rate it lower.
3. Is the image completely ignored? Do the following messages continue without acknowledging it at all?
   If yes, rate it low.
4. Does the image add some relevance? Do the next messages mention something loosely connected to it, even if the dialog still makes sense without it?
   If yes, give a mid-range score.
5. Is the image clearly useful? Do the messages directly reference the image, making the conversation easier to understand?
   If yes, score it higher.
6. Is the image essential? Would the dialog be incomplete, confusing, or meaningless without it?
   If yes, give the highest score.

Your Task: Provide your response in valid JSON format:

```
<results>
{
  "caption": "<caption>",
  "answer": <score between 1-5>,
  "explanation": "<Step-by-step reasoning
                  for the score>"
}
</results>
```

Scoring Scale:

- **1** → The image contradicts or misleads the dialog.
- **2** → The image is ignored and not acknowledged at all.
- **3** → The image is loosely relevant, but the dialog makes sense without it.
- **4** → The image adds context and is referenced, but isn't crucial.
- **5** → The image is critical, and the dialog wouldn't make sense without it.

Important: Justify your score with logical reasoning before assigning it.

Figure 5: Prompt template for relevance filtering using GPT-4V.

MMDialog Dataset

Alex: hey there buddy boy boyo
Sara: hello , you have any hobbies ?
Alex:: i can listen to britney spears all day
Sara: awesome i like listening to it while i play tennis .
Alex:: i love to spend money that i did not earn
Sara: oh , i see that a lot in my insurance office .
Alex:: what do you do for a living ?
Sara: since i was fired i found a job in insurance .
Alex:: what is the pay like ?
Sara: it is ok , but my dad made a ton before he passed away .
Alex:: i am sorry . at least he is in a better place now .
Sara: it is ok , i was pretty young when it happened .
Alex:: do you like to tan ?
Sara:

Alex: I am too lazy to play sports.

ImageChat Dataset

Aunt: Let's change the scenery.

Uncle: It reminds me of my old home. What is wrong with the scene, may I ask?
Aunt: It's dangerous. We could fall off.

Figure 6: Two illustrative examples of MAC from the MMDialog and ImageChat datasets, where the image context significantly influences the prediction. Blue indicates the input prefix provided to the MAC model, while Green highlights the text characters that the model is expected to predict.

B.2 Formatting interleaved inputs

For models that do not natively support interleaved image-text inputs, we restructure the input to explicitly encode the position of visual content. Image embeddings are prepended to the input sequence, and a special token such as `<IMAGE>` is inserted at the corresponding turn in the dialog where the image appeared. This approach enables the model to attend to both the image features and their temporal alignment within the dialog. For example, a turn originally written as: "*User: That looks amazing!*" would be transformed into "*User: `<IMAGE>` That looks amazing!*"

C Additional Details for Experiments

C.1 Baseline Models

Textual Models: These models operate solely on textual input, without access to any visual modality. Trie-based methods such as *Most Popular Completion (MPC)* (Bar-Yossef and Kraus, 2011) construct a character-level trie from historical user utterances to suggest completions based on frequency, while its extension *MPC++* (Bar-Yossef and Kraus, 2011) uses a suffix trie to offer better coverage for previously unseen prefixes. N-gram-based methods like *QueryBlazer (QB)* (Kang et al., 2021) rely on subword tokenization and n-gram language modeling to retrieve completions from historical logs and synthesize novel predictions.

Vision Language Models: Recent advances in VLMs enable the processing of both textual and visual modalities. The models we explored include MiniCPM-V (Yao et al., 2024), a powerful 8B parameter VLM that integrates a SigLIP (Zhai et al., 2023) vision encoder with a Qwen2.5-7B language decoder. PaliGemma (3B) (Beyer et al., 2024) also employs a SigLIP vision encoder, coupled with the Gemma 2 (Team et al., 2024) language model for text generation. Lastly, Qwen2-VL (Wang et al., 2024) is a vision-language instruction-tuned variant from the Qwen2 series (Yang et al., 2024), combining a Vision Transformer (ViT) (Dosovitskiy et al., 2020) encoder with the Qwen2 decoder to enable fine-grained, instruction-following capabilities across vision and text modalities.

C.2 Performance of MAC Benchmarks on Seen prefixes

On seen prefixes (See Table 4), textual models achieve their strongest performance, with MPC and MPC++ reaching very high syntactic and semantic alignment on MMDD (SM $=$ 0.79, F1 $=$ 0.81, TES $=$ 0.72), indicating strong memorization and a close fit to training distributions. VLMs, while showing lower syntactic precision (F1 $\approx$ 0.27–0.30), maintain consistent trigger rates (TR $\approx$ 0.99) and balanced completion lengths, reflecting stable yet less overfitted behavior. In ImageChat, both model families perform comparably, with VLMs (MiniCPM-V, PaliGemma) matching or slightly surpassing textual models in Partial-F1 ($\approx$ 0.48). Overall, textual models dominate on seen data through memorization, whereas VLMs achieve similar precision with greater contextual grounding.

C.3 Additional Details of Router-Suggest

We performed joint hyperparameter and architecture search using random sampling over a struc-

Method	Model	TR	Syntactic Match	PR-Precision	PR-Recall	Partial-F1	Avg Pred Len	TES
				MMDD				
Text	MPC	0.9679	**0.7902**	**0.8066**	**0.8060**	**0.8063**	**27.5**	**0.7153**
	MPC++	0.9679	**0.7902**	**0.8066**	**0.8060**	**0.8063**	**27.5**	**0.7153**
	QB	0.9474	0.2355	0.5508	0.3213	0.4064	12.1	0.3725
VLMs	MiniCPM-V	0.9898	0.1349	0.3505	0.2632	0.3007	22.3	0.2352
	PaliGemma	0.9880	0.1179	0.3138	0.2381	0.2707	20.0	0.2357
	Qwen2-VL	**0.9902**	0.1112	0.3016	0.2279	0.2596	19.9	0.2097
				ImageChat				
Text	MPC	0.9497	**0.2892**	0.4559	0.4723	0.4639	13.7	**0.2688**
	MPC++	0.9497	**0.2892**	0.4559	0.4723	0.4639	13.7	**0.2688**
	QB	0.9741	0.2094	**0.5053**	0.4404	0.4708	8.2	0.2444
VLMs	MiniCPM-V	**0.9958**	0.2100	0.4611	**0.5010**	0.4802	14.4	0.2552
	PaliGemma	0.9875	0.2020	0.4694	0.4924	**0.4806**	**14.7**	0.3021
	Qwen2-VL	0.9945	0.1699	0.4323	0.4617	0.4465	**14.7**	0.2464

Table 4: Performance metrics on **seen prefixes** of the MMDD (top) and ImageChat (bottom) test sets, organized by model type (Text vs. VLMs).

tured search space, combining both network topology and training parameters. Each configuration was trained using a fixed batch size of 256 and dropout rate of 0.2. For every trade-off parameter $\lambda \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$, we executed 50 random trials, totaling 250 experiments for each dataset.

Parameter	Search Space
Hidden dimensions	$[128]$, $[256]$, $[128, 64]$, $[256, 128]$, $[512, 256]$, $[64, 32]$, $[256, 128, 64]$, $[512, 256, 128]$
Epochs	{50, 100}
Learning rate	$\{1e^{-4}, 5e^{-4}, 1e^{-3}\}$
λ	{0.0, 0.25, 0.5, 0.75, 1.0}
Batch size	256 (fixed)
Dropout	0.2 (fixed)

Table 5: Search space for architecture and hyperparameter tuning. Each λ setting was tuned independently using random search.

The scoring function balanced accuracy and latency using a weighted objective:

$$\text{Score} = (1 - \lambda) \times \text{Accuracy} + \lambda \times \text{Cost},$$

where cost values were normalized by the maximum observed latency ($\max \text{cost} = 2.0891$ for *MiniCPM-V*). This formulation ensured fair comparison across trade-off settings, allowing selection of the highest-scoring model overall.