MICA: Multi-Agent Industrial Coordination Assistant

Di Wen[1], Kunyu Peng[1,2,*], Junwei Zheng[1], Yufan Chen[1], Yitian Shi[1], Jiale Wei[1], Ruiping Liu[1], Kailun Yang[3], and Rainer Stiefelhagen[1]

***Abstract*— Industrial workflows demand adaptive and trustworthy assistance that can operate under limited computing, connectivity, and strict privacy constraints. In this work, we present MICA (Multi-Agent Industrial Coordination Assistant), a perception-grounded and speech-interactive system that delivers real-time guidance for assembly, troubleshooting, part queries, and maintenance. MICA coordinates five role-specialized language agents, audited by a safety checker, to ensure accurate and compliant support. To achieve robust step understanding, we introduce Adaptive Step Fusion (ASF), which dynamically blends expert reasoning with online adaptation from natural speech feedback. Furthermore, we establish a new multi-agent coordination benchmark across representative task categories and propose evaluation metrics tailored to industrial assistance, enabling systematic comparison of different coordination topologies. Our experiments demonstrate that MICA consistently improves task success, reliability, and responsiveness over baseline structures, while remaining deployable on practical offline hardware. Together, these contributions highlight MICA as a step toward deployable, privacy-preserving multi-agent assistants for dynamic factory environments. The source code will be made publicly available at https://github.com/Kratos-Wen/MICA.**

I. Introduction

Modern manufacturing increasingly operates under rapid line reconfiguration, product variants, and strict safety and compliance requirements. Assembly procedures are long-horizon and interdependent, with tool–part constraints and exception handling that challenge non-expert and rotating workers; mistakes incur time, quality, and safety costs [1]. At the same time, privacy and connectivity constraints often preclude cloud offloading, and confidentiality limits the collection of large annotated datasets. Although vision-based assistance improves stepwise guidance in realistic settings [2], reliable on-device deployment under limited data remains difficult.

Large language models have strong general reasoning ability [3], [4], and multi-agent formulations promise structured problem solving [5], [6], [7], [8], [9], [10]. Existing multi-agent evaluations are largely text-centric or simulated, with limited grounding in sensed factory state or speech interaction; coordination reliability degrades under partial or asynchronous observations, conflicting with cycle-time and safety requirements on the shop floor. This gap motivates perception-grounded and budget-aware multi-agent assistance. To reduce data and privacy barriers, recent work shows that a small capture of part photos or short multi-view videos, together with manuals, can bootstrap an image and text knowledge base for local, privacy-preserving assistance [11].

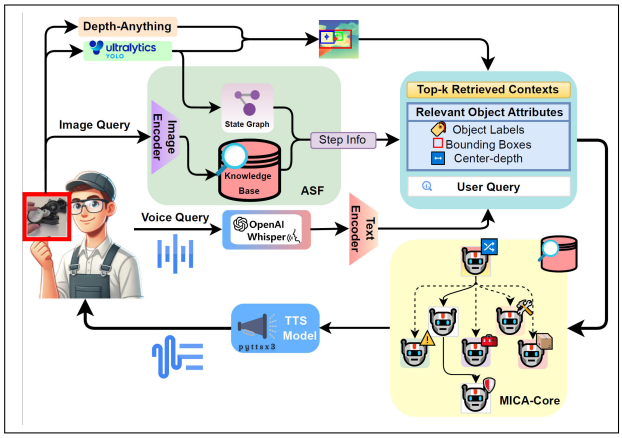

Fig. 1. Overview of the proposed MICA system. Egocentric vision and speech queries are processed into structured object contexts via YOLO-based detection and depth estimation. These contexts, together with state-graph priors and knowledge base information, support Adaptive Step Fusion (ASF) for robust step recognition. The MICA-core then integrates perception and reasoning to deliver safety-audited, speech-based guidance in real time.

We present **MICA** (Multi-Agent Industrial Coordination Assistant), a perception-grounded and speech-interactive industrial assistant that runs entirely on edge hardware. MICA couples egocentric vision with multi-agent language reasoning to deliver real-time assembly, troubleshooting, part queries, and maintenance support. The system comprises three integrated modules: ❶ Depth-guided Object Context Extraction for stable, view-aligned part context; ❷ Adaptive Assembly Step Recognition that blends a state-graph expert with an image-retrieval expert; and ❸ *MICA-core*, a modular reasoning layer that routes queries to role-specialized agents under safety auditing. Built on a lightweight image–text knowledge base derived from assembly manuals and a small set of component captures, our system avoids large-scale annotation while remaining adaptable to new assembly procedures. To enable rigorous comparison under

This work was supported in part by the SmartAge project sponsored by the Carl Zeiss Stiftung (P2019-01-003; 2021-2026), the University of Excellence through the "KIT Future Fields" project, in part by the Helmholtz Association Initiative and Networking Fund on the HoreKA@KIT partition and the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG. This work was also supported in part by the National Natural Science Foundation of China (Grant No. 62473139), in part by the Hunan Provincial Research and Development Project (Grant No. 2025QK3019), and in part by the State Key Laboratory of Autonomous Intelligent Unmanned Systems (the opening project number ZZKF2025-2-10).

[1]The authors are with Karlsruhe Institute of Technology, Germany.
[2]The author is also with INSAIT, Sofia University "St. Kliment Ohridski", Bulgaria.
[3]The author is with Hunan University, China.
*Corresponding author: Kunyu Peng (kunyu.peng@kit.edu).

identical tools, prompts, knowledge access, and budgets, we establish a controlled benchmark that instantiates four representative multi-agent topologies. We further introduce two deployment-oriented metrics: Knowledge Base Alignment (KBA) for factual consistency with the curated component knowledge base, and Energy per Successful Answer (E/succ) for energy–utility efficiency in real-time use.

We summarize our contributions as follows:

- A fully offline, perception-grounded industrial assistant that unifies egocentric vision, speech I/O, and role-specialized multi-agent reasoning on edge devices.
- *Adaptive Step Fusion* (ASF), a lightweight fusion and online adaptation mechanism that integrates rule-based workflow constraints with retrieval-based visual similarity and enables real-time correction through natural-language feedback.
- A multi-agent coordination benchmark with standardized protocols and two metrics (KBA and E/succ) tailored to safety-critical industrial assistance.

II. Related Work

A. Real-Time Egocentric Vision in Wearable Systems

Wearable egocentric vision offers direct access to gaze, hand–object interactions, and short-horizon intent, which are central to shop-floor assistance and safety auditing. Recent surveys and outlooks highlight the growing push toward on-device assistance and privacy-preserving perception [12]. Large-scale benchmarks [13], [14], [15], [16], *e.g.*, EPIC-Kitchens [15] and Ego4D [16], have catalyzed progress in segmentation, anticipation, and episodic memory, enabling long-horizon reasoning over first-person video. Beyond general benchmarks, task- and modality-focused resources advance the field toward deployment: Nymeria contributes synchronized, multimodal recordings with full-body motion and Aria-based sensors [17]; EgoSim provides a multi-view simulator plus real data for body-worn cameras [18]; EgoEnv links first-person video to local environment representations for better state awareness [19]. New evaluations target assistance with text and structure, including Ego-TextVQA for scene-text-aware video QA and EgoSG for egocentric 3D scene graphs [20], [21]. Practical assistive prototypes and wearables illustrate end-user benefits and the value of resource-constrained design [22], [23]. Meanwhile, geometry-aware egocentric scene understanding (e.g., EDINA) addresses tilted viewpoints and dynamic foregrounds common on the factory floor [24]. Vinci demonstrates an end-to-end egocentric VLM assistant with streaming memory and grounding on portable devices, pointing to real-time, on-device workflows [25]. While recent egocentric resources and prototypes improve on-device perception, many pipelines remain single-model or cloud-assisted, limiting modularity and guaranteed on-device operation in industrial conditions. Our system targets offline, on-device operation by coupling local perception with role-specialized agents and a safety/KB auditor under compute and connectivity constraints.

B. Multi-Agent Large Language Models

LLMs have moved from single-agent autonomy to multi-agent collaboration, where multiple LLM-based agents communicate, cooperate, or compete to solve tasks beyond a single model's capacity [26], [27], [5], [6]. In manufacturing, multi-agent coordination lets distributed machines and software adapt in real time, balance loads, recover from faults, and optimize throughput across heterogeneous equipment—improving flexibility, scalability, and resilience [28], [29], [30], [31], [32], [33]. Mechanisms that sustain long-horizon interactions include structured roles/memory [34], [35] and reasoning curricula that decompose, search, and vote [9], [10], [36]. Yet most methods remain text-bound without egocentric sensing or actuation, and their coordination reliability degrades under partial/asynchronous observations—conditions at odds with strict cycle-time and safety constraints on the shop floor. Evidence from simulated environments suggests that persistent memory and planning improve long-horizon behavior, while specialization benefits difficult reasoning but may be unnecessary for simple queries [37], [38]. To improve coordination, prior work explores open dialogue [5], [6], structured workflows [7], adversarial debate, and learned cooperation modules (COPPER) for cross-verification and refinement [6], [39], [8], [40]. However, current multi-agent LLM studies are largely evaluated in simulated domains, focusing on communication algorithms rather than real-world perception [5], [7]. MICA grounds collaboration in sensed state with explicit time/energy budgets and safety auditing, supporting reliable workflows beyond simulated domains.

III. Methodology

Our intelligent industrial assistance system, **MICA** (Multi-Agent Industrial Coordination Assistant), addresses the core challenge of providing accurate, real-time assembly guidance in dynamic factory environments, where visual occlusion, step ambiguity, and safety constraints make robust recognition essential. As illustrated in Fig. 1, the system integrates three tightly coupled modules: (1) *Depth-guided Object Context Extraction*, which focuses on the most relevant components from the worker's viewpoint; (2) *Adaptive Assembly Step Recognition*, which resolves step ambiguities and adapts online with user feedback; and (3) *Multi-Agent Collaborative Reasoning via MICA-core*, which delivers task-specific guidance under safety auditing. Together, these modules form a pipeline in which perception refines context, step recognition constrains reasoning, and reasoning returns adaptive feedback to the worker.

A. Depth-guided Object Context Extraction

To ensure reliable perception under dynamic assembly conditions, we adopt YOLOv11 [41] as the base detector, trained following [11] on the Gear8 dataset. Each frame produces raw component detections, which are stabilized by aggregating results over a sliding window of L frames. We denote by $\mathbf{b}_i$ the bounding box and by c_i its confidence.

Detections with $\mathrm{IoU}(\mathbf{b}_i, \mathbf{b}_j) \geq \tau_{\mathrm{IoU}}{=}0.5$ are clustered as $\mathcal{C} = \{(\mathbf{b}_i, c_i)\}_{i=1}^{m}$, and fused by confidence-weighted averaging:

$$\hat{\mathbf{b}} = \frac{\sum_{i=1}^{m} c_i\, \mathbf{b}_i}{\sum_{i=1}^{m} c_i}, \qquad \hat{c} = \frac{1}{m} \sum_{i=1}^{m} c_i. \tag{1}$$

On this fused result, Depth-Anything [42] estimates pixel-wise depth. The nearest component relative to the camera center in the depth map is taken as the worker's primary focus, while nearby components within spatial and depth thresholds (τ_p, τ_d) are also included to capture peripheral interactions:

$$\mathcal{O}_{\mathrm{rel}} = \{\, o_i \mid \|x_i - x^*\| \leq \tau_p,\ |d_i - d^*| \leq \tau_d \,\} \tag{2}$$

where (x_i, d_i) denote the spatial and depth coordinates of object o_i, and (x^*, d^*) correspond to the nearest component. Only this fused, depth-refined context is passed to subsequent modules.

B. Adaptive Assembly Step Recognition

We estimate the assembly step from *streaming* first-person video by integrating two complementary detectors and a lightweight adaptive fusion. The *state–graph detector* leverages workflow constraints automatically derived from the component knowledge base (KB) and assembly procedure templates to score each candidate step according to required components and their multiplicities, enforcing structural consistency. The *retrieval detector* compares the current frame against a gallery of reference states in an embedding space to provide a similarity–based estimate. The two detectors are complementary: the former supplies structure and interpretability, the latter is robust to occlusion and detection noise; our *Adaptive Step Fusion (ASF)* combines them at the class level and adapts online from speech–driven feedback.

a) State–graph detector. Let $\mathcal{S} = \{S_1, \ldots, S_K\}$ be the set of steps. For each step S_j, the KB specifies a rule triple $(\mathrm{all\_of}_j, \mathrm{any\_of}_j, \mathrm{forbid}_j)$, where sets list required, alternative, and forbidden components (by KB part IDs). For brevity, denote $\mathcal{A}_j := \mathrm{all\_of}_j$, $\mathcal{O}_j := \mathrm{any\_of}_j$, and $\mathcal{F}_j := \mathrm{forbid}_j$. Counts $n(k)$ are computed from the depth–refined context O_{rel} (Sec. III-A) aggregated over the sliding window; the required multiplicity $r_j(k)$ comes from the KB (default $r_j(k){=}1$ if unspecified). We score each step by

$$C_s(j) = \alpha\, \mathrm{all}_j + (1 - \alpha)\, \mathrm{any}_j - \mathrm{pen}_j, \tag{3}$$

where

$$\phi_j(k) := \min\Big(1, \frac{n(k)}{\max(1, r_j(k))}\Big), \tag{4}$$

$$\mathrm{all}_j = \frac{1}{\max(1, |\mathcal{A}_j|)} \sum_{k \in \mathcal{A}_j} \phi_j(k), \tag{5}$$

$$\mathrm{any}_j = \mathbb{I}\Big[\, |\mathcal{O}_j| = 0 \ \vee\ \exists\, k \in \mathcal{O}_j :\ n(k) \geq r_j(k) \Big], \tag{6}$$

$$\mathrm{pen}_j = \tfrac{1}{2}\, \mathbb{I}\Big[\, \exists\, k \in \mathcal{F}_j :\ n(k) > 0 \Big]. \tag{7}$$

Here $\mathbb{I}[\cdot]$ is the indicator, $\alpha \in [0, 1]$ (we use $\alpha{=}0.6$). The detector outputs

$$S_s = \arg\max_j C_s(j), \qquad C_s = \max_j C_s(j). \tag{8}$$

b) Retrieval detector. Let $f(\cdot)$ be an image encoder [43], $\{g_j\}$ be per-step references and q denote the current frame. We score each step by the cosine similarity

$$C_r(j) = \cos\big(f(q),\, f(g_j)\big) \quad \text{(or top-}k\text{ average)}, \tag{9}$$
$$S_r = \arg\max_j C_r(j), \qquad C_r = \max_j C_r(j).$$

c) ASF scoring. To fuse the detectors, our *Adaptive Step Fusion (ASF)* maintains per–class expert weights $W_{j,e} \geq 0$, per–class biases b_j, and global gates $g_e \geq 0$ with $g_s + g_r = 1$. To preserve weak signals from non–winning experts we define

$$c_{e,j} = \begin{cases} C_e, & S_e = S_j, \\ \lambda_e C_e, & \text{otherwise}, \end{cases} \qquad e \in \{s, r\}, \tag{10}$$

with leak parameters $\lambda_e \in [0, 1)$. We define the KB coverage as $\mathrm{cov}_j := \mathrm{all}_j$, i.e., the averaged satisfaction over required components (Sec. III-Ba). Non–jumping dynamics are encoded by an allowed set $\mathcal{A}(S_{\mathrm{prev}})$ from the previous fused step. The overall score is

$$\begin{aligned} \mathrm{score}_j = b_j + g_s W_{j,s} c_{s,j} + g_r W_{j,r} c_{r,j} \\ + \lambda_{\mathrm{cov}}\, \mathrm{cov}_j - \lambda_{\mathrm{tr}}\, \mathbb{I}\big[S_j \notin \mathcal{A}(S_{\mathrm{prev}})\big], \end{aligned} \tag{11}$$

We use nonnegative weights $\lambda_{\mathrm{cov}}, \lambda_{\mathrm{tr}} \geq 0$ to balance coverage and transition penalties. The fused step is $S_f = \arg\max_j \mathrm{score}_j$, with a calibrated confidence obtained by softmax over $\{\mathrm{score}_j\}$.

d) ASF online adaptation. User feedback $y \in \mathcal{S}$ is used to update (W, b, g) without backpropagation. We define a focal–style impact $\kappa_e = (1 - C_e)^{\gamma}$ with $\gamma > 0$; confident hits ($S_e = y, C_e \geq C_{\mathrm{freeze}}$) are frozen by setting $\kappa_e = 0$. To reduce collapse into a single class, the effective step size is scaled as

$$\eta_{\mathrm{eff}} = \eta\, n_y^{-\rho}\, d, \tag{12}$$

where n_y is the number of feedback events on class y, $\rho \in (0, 1]$, and d depends on the recent fraction of y in a sliding history window. Let $\hat{\imath} := \arg\max_{j \neq y} \mathrm{score}_j$ be the highest-scoring non-target class at feedback time. Weights are updated multiplicatively per column within a trust-region bound τ_{trust}:

$$W_{y,e} \leftarrow W_{y,e}\, (1 + \delta_{y,e}), \tag{13}$$
$$W_{\hat{\imath},e} \leftarrow W_{\hat{\imath},e}\, (1 - \delta_{\hat{\imath},e}), \tag{14}$$

with $\delta \leftarrow \min\big(\eta_{\mathrm{eff}} \kappa_e,\ \tau_{\mathrm{trust}}\big)$. If both experts err, we correct only the column with lower C_e to avoid oscillation. Biases are adjusted conservatively with conservation across classes and clipped to $|b_j| \leq b_{\max}$. Gates are nudged only when exactly one expert hits and then renormalized to $g_s + g_r = 1$. After each update, columns $\{W_{j,e}\}_j$ are clamped and renormalized, and a floor $W_{j,e} \geq \varepsilon_{\mathrm{floor}}$ avoids starving classes. All parameters are persisted and warm–started across sessions. Together, ASF introduces three key innovations: (i) explicit incorporation of workflow compatibility and non-jumping transitions into the fusion score, (ii) class–wise fusion with

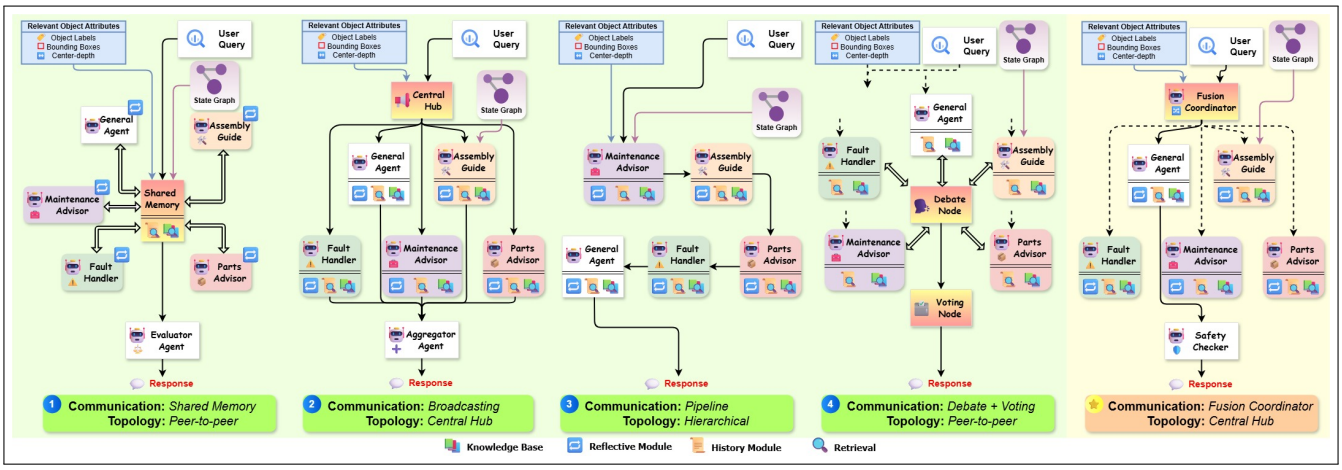

Fig. 2. An overview of the multi-agent LLM baseline architectures for comparison: (1) SharedMemory: decentralized peer-to-peer with a shared memory and evaluator; (2) CentralizedBroadcast: hub-and-spoke publish–subscribe with an aggregator; (3) HierarchicalPipeline: fixed sequential relay across specialists; (4) DebateVoting: peer debate followed by consensus voting.

confidence–aware online updates, and (iii) anti–collapse regularization through history–based scaling and weight floors. These choices provide a lightweight yet effective mechanism for online adaptation in streaming assembly recognition.

C. Multi-Agent Collaborative Reasoning via MICA-core

To transform raw perceptual signals into actionable guidance, we introduce *MICA-core*, a modular multi-agent reasoning framework built on an instruction-tuned LLM [44]. MICA-core receives structured inputs from the preceding modules, namely (i) object contexts from depth-guided detection, and (ii) assembly step hypotheses from ASF. Together with natural-language queries transcribed by Speech-to-Text (STT) [45], these signals form a unified reasoning context.

Within MICA-core, a lightweight LLM router dynamically assigns each query to one of five specialized agents: *Assembly Guide*, *Parts Advisor*, *Maintenance Advisor*, *Fault Handler*, and a fallback *General Agent*. Each agent operates under a Retrieval-Augmented Generation (RAG) paradigm, retrieving agent-specific evidence from the structured KB and refining responses through iterative reasoning.

To guarantee reliability in safety-critical assembly contexts, all agent outputs are audited by a dedicated safety checker that enforces rule-based assembly constraints and verifies responses against the KB. This layer enforces domain constraints such as correct tool usage, assembly order, and hazard warnings, thereby preventing unsafe recommendations from reaching the user. The combination of dynamic routing, specialized RAG agents, and explicit safety auditing allows MICA-core to deliver contextually precise, semantically rich, and industrially safe responses.

D. Speech-based Interactive Feedback Loop

The reasoning outputs of MICA-core are embedded into an interactive feedback loop with the worker. Queries are captured via Speech-to-Text (STT) [45], while system responses and status updates are synthesized through Text-to-Speech (TTS) [46]. Crucially, workers can verbally confirm or correct ASF's step predictions in real time, directly influencing the online adaptation of the fusion module. This human-in-the-loop mechanism improves recognition accuracy while making the adaptation process explicit to the worker.

IV. Experiments

A. Implementation Details

Experiments are implemented in PyTorch 2.6.0 with CUDA 12.4. The YOLOv11-L detector [41] is fine-tuned on Gear8 following [11]. Multi-frame fusion uses $\tau_{\text{IoU}} = 0.5$, confidence threshold 0.4, at least $m = 3$ detections, and persistence over $T = 5$ consecutive frames. Depth estimation is performed with Depth-Anything-V2-Large [42], using spatial and depth thresholds (τ_p, τ_d) for context refinement. In ASF, we set $\alpha = 0.6$ (rule balance), focal factor $\gamma = 2$, base step size $\eta = 0.1$ and history scaling $\rho = 0.5$. Regularization includes trust-region bound $\tau_{\text{trust}} = 0.2$, bias bound $b_{\max} = 1.0$, and weight floor $\varepsilon_{\text{floor}} = 10^{-3}$. Semantic retrieval uses SentenceTransformer (all-MiniLM-L6-v2) [47] with FAISS [48], and multi-agent reasoning uses Qwen2.5-7B-Instruct [44]. Speech recognition uses Whisper-small [45] (16 kHz, 8 s windows), and TTS uses pyttsx3 [46] (180 wpm, 22.05 kHz).

B. Multi-Agent Coordination Benchmark

To systematically study coordination under identical tools, prompts, KB access, and backbone LLM, we establish a controlled benchmark comprising four representative interaction topologies (Fig. 2). Each topology is instantiated as an engineering counterpart of a well-studied paradigm, providing a standardized protocol for fair comparison.

SharedMemory (Fig. 2 (1)). Peer agents read and write a shared blackboard context, submit independent proposals, and a separate evaluator selects the final answer [34], [35].

CentralizedBroadcast (Fig. 2 (2)). A central hub broadcasts the task state to all agents, collects parallel responses, and aggregates them into a single output [5], [6].

TABLE I

PER-STEP PERFORMANCE OF ASF BEFORE AND AFTER ONLINE ADAPTATION (10 UPDATES PER STEP). BEST RESULTS IN EACH COLUMN ARE BOLD; IN CASE OF TIES, ALL BEST ENTRIES ARE BOLD.

Step	Acc (%) ↑		Prec (%) ↑		Rec (%) ↑		F1 (%) ↑		ECE ↓	
	Baseline	w/ ASF	Baseline	w/ ASF	Baseline	w/ ASF	Baseline	w/ ASF	Baseline	w/ ASF
S1	**97.63**	92.71	70.17	**85.37**	**97.63**	96.84	81.65	**90.74**	0.49	**0.38**
S2	81.82	**81.88**	**91.45**	90.68	**77.54**	**77.54**	**83.92**	83.59	**0.50**	**0.50**
S3	88.98	**97.08**	**97.41**	95.73	**88.98**	88.19	**93.00**	91.80	0.55	**0.54**
S4	0.00	**95.34**	0.00	**91.46**	0.00	**89.29**	0.00	**90.36**	0.52	**0.43**

HierarchicalPipeline (Fig. 2 (3)). Agents are arranged in a fixed relay, where each stage refines the previous output before passing it to the next [10], [9].

DebateVoting (Fig. 2 (4)). Agents independently draft responses, critique one another, and then vote to select a consensus output [8], [36].

We evaluate all topologies on five task categories (*General*, *Assembly*, *Part Attributes*, *Maintenance*, and *Fault Handling*) under identical compute budgets and the same knowledge grounding, which enables a controlled assessment of coordination efficacy.

C. Evaluation Metrics and Setup

We evaluate (i) the effect of online Adaptive Step Fusion (ASF, Sec. III-B) and (ii) the comparative performance of the multi-agent topologies (Sec. IV-B).

a) ASF evaluation. We report pre/post-adaptation performance on step prediction S_f using accuracy (Acc), precision (Prec), recall (Rec), F1-score (F1), and Expected Calibration Error (ECE) [49], which measures the alignment between predicted confidence and empirical correctness.

b) Benchmark protocol and metrics. We evaluate the four benchmark topologies using three families of metrics (Tab. II):

(i) Automatic evaluation metrics. We use three automatic metrics: (a) task success (TS, %), a binary indicator of whether an answer satisfies the task-specific success criterion defined by deterministic KB-derived rules; (b) BLEU (BL) [50] and ROUGE-L (RG) [51], which measure lexical and subsequence overlap with reference responses; and (c) *Knowledge Base Alignment* (KBA, %), a benchmark-specific metric for factual consistency with the curated component KB. Given an answer a, we extract canonical KB phrases appearing in a and compute the coverage of KB attribute categories referenced by these phrases. Let $P(a)$ denote phrase precision and $R(a)$ the fraction of covered attribute categories. The final KBA score is defined as the harmonic mean

$$\mathrm{KBA}(a) = \frac{2P(a)R(a)}{P(a) + R(a)}.$$

(ii) GPT-based evaluation. Following recent LLM evaluation practice [52], [53], we use GPT-4o [54] as a judge to score factual accuracy (Acc), relevance (Rel), consistency (Con), helpfulness (Help), and safety (Safe).

(iii) Resource-oriented metrics. We report end-to-end Average Latency (AL, s), measured from the availability of the ASF output to completion of the assistant's response, and Energy per Successful Answer (E/succ, kJ), computed from GPU power measurements collected via NVIDIA NVML after subtracting the idle baseline.

c) Experimental setup.

(i) ASF adaptation. We consider four assembly steps with annotated ground truth. Pre-adaptation uses initial ASF parameters; post-adaptation is measured after ten updates per step. The ten-update budget balances operator effort and adaptation efficacy in industrial workflows.

(ii) Benchmark evaluation. To isolate coordination effects, we use fixed video segments and ground-truth labels as inputs, thereby removing perception noise from the comparison. The Gear8 dataset [11] contains eight components; for each component and category, we formulate four queries, yielding 32 queries per category (160 in total across five categories: general, assembly, attributes, maintenance, and fault handling). All topologies are evaluated under identical budgets and knowledge grounding. Unless otherwise noted, all LLM calls use deterministic decoding with fixed prompts and a frozen KB snapshot, without self-consistency sampling or retries.

D. Quantitative Results

We report the impact of ASF adaptation on step recognition and present a controlled comparison of coordination topologies across five categories, evaluated by automatic, GPT-based, and efficiency metrics.

a) ASF adaptation. Tab. I demonstrates that online ASF substantially improves robustness with minimal feedback. Ten lightweight corrections per step reduce calibration error (ECE) across all steps and correct systematic late-stage failures. In particular, S_4 improves from 0% to 95.34% accuracy and achieves $+90.36$ F1, showing that feedback-driven reweighting is most effective when the state graph and retrieval detector diverge. Mid-sequence steps (S_3) also benefit with $+8.1$ accuracy and reduced ECE ($0.55 \rightarrow 0.54$). By contrast, S_1 saturates quickly: accuracy drops marginally ($97.63\% \rightarrow 92.71\%$) while precision rises by $+15.2$, reflecting a precision–recall rebalancing. These dynamics confirm ASF's practicality: a small, fixed supervision budget converts an initially brittle fusion into a calibrated, generalizable predictor without requiring prolonged operator involvement.

b) Benchmarking coordination topologies. Tab. II benchmarks MICA-core against four representative coordination structures under controlled conditions. On average, MICA achieves the highest task success (TS 63.13%) and the strongest knowledge base alignment (KBA 19.12%), while maintaining the lowest latency (0.71 s) and the lowest energy per successful answer (2.05 kJ). This profile indicates that *MICA* balances factual faithfulness, responsiveness, and efficiency, whereas baselines tend to sacrifice at least one of these dimensions.

(i) Category-specific behaviors. Performance varies by query type. *SharedMemory* is strongest on maintenance (TS 37.50%), where co-occurrence heuristics align with routine safety checks, but it exhibits high latency due to evaluator overhead and generalizes poorly beyond this category.

TABLE II

BENCHMARK RESULTS FOR *MICA-core* AND FOUR COORDINATION TOPOLOGIES ACROSS FIVE CATEGORIES (GENERAL, ASSEMBLY-RELATED, PART ATTRIBUTE, MAINTENANCE-RELATED, FAULT HANDLING). BEST RESULTS IN EACH COLUMN ARE BOLD; IN CASE OF TIES, ALL BEST ENTRIES ARE BOLD. KBA IS NOT COMPUTED FOR THE *General Question* BECAUSE IT LACKS A STRUCTURED KB ALIGNMENT TARGET.

Topology	Automatic Evaluation Metrics				GPT-based Evaluation Metrics					AL (s) ↓	E/succ (kJ) ↓
	TS (%) ↑	BL ↑	RG ↑	KBA (%) ↑	Acc ↑	Rel ↑	Con ↑	Help ↑	Safe ↑		
General Question											
SharedMemory	31.25	0.52	0.50	N/A	3.25	3.19	**4.88**	2.91	**5.00**	2.79	2.03
CentralizedBroadcast	12.50	0.46	0.44	N/A	2.22	3.25	4.47	2.66	**5.00**	2.87	5.13
HierarchicalPipeline	23.13	0.48	0.50	N/A	3.12	3.25	3.78	3.22	**5.00**	2.92	2.33
DebateVoting	59.38	0.64	0.65	N/A	3.31	4.22	4.56	3.75	**5.00**	5.08	2.66
MICA-core (ours)	**90.63**	**0.76**	**0.77**	N/A	**4.19**	**4.41**	4.59	**4.25**	5.00	**0.58**	**0.71**
Assembly-related Question											
SharedMemory	15.63	0.07	0.09	9.10	1.94	2.16	2.91	1.84	**5.00**	3.77	5.07
CentralizedBroadcast	**46.88**	**0.24**	**0.36**	19.10	**3.00**	**2.94**	**4.94**	2.75	**5.00**	4.19	2.31
HierarchicalPipeline	22.88	0.13	0.24	10.24	2.03	2.22	3.31	2.19	**5.00**	3.88	4.68
DebateVoting	37.50	0.07	0.18	5.43	2.12	2.31	3.88	2.28	**5.00**	7.51	4.30
MICA-core (ours)	43.75	0.19	0.28	**21.18**	2.91	2.88	**4.94**	**2.78**	**5.00**	**0.74**	**1.61**
Part Attribute Question											
SharedMemory	78.13	0.12	0.06	9.32	2.16	2.53	3.84	2.47	**5.00**	4.06	1.20
CentralizedBroadcast	71.88	0.50	0.58	**36.98**	3.91	4.09	**4.91**	4.03	**5.00**	3.95	1.84
HierarchicalPipeline	93.75	0.20	0.27	33.57	3.97	4.00	4.47	3.91	**5.00**	3.95	1.27
DebateVoting	**96.88**	**0.57**	**0.62**	30.68	4.12	4.03	4.88	**4.09**	**5.00**	7.92	1.40
MICA-core (ours)	**96.88**	0.38	0.45	36.68	**4.22**	**4.12**	4.59	**4.09**	**5.00**	**0.77**	**0.73**
Maintenance-related Question											
SharedMemory	**37.50**	0.11	0.06	**10.53**	**2.31**	**2.22**	3.06	**1.97**	**5.00**	3.28	5.36
CentralizedBroadcast	25.00	**0.24**	**0.36**	5.10	2.03	2.12	2.94	1.72	**5.00**	3.31	**3.64**
HierarchicalPipeline	6.25	0.06	0.09	3.04	1.03	1.19	**3.62**	1.12	**5.00**	3.11	10.29
DebateVoting	12.50	0.04	0.06	9.09	1.16	1.22	3.56	1.25	**5.00**	6.85	46.11
MICA-core (ours)	21.88	0.05	0.08	5.13	1.22	1.38	2.66	1.47	**5.00**	**0.78**	5.67
Fault Handling Question											
SharedMemory	56.25	0.23	0.35	5.45	2.91	3.00	3.84	3.19	**5.00**	3.76	2.51
CentralizedBroadcast	46.88	**0.24**	**0.36**	5.98	3.03	3.09	3.56	3.03	**5.00**	3.59	1.78
HierarchicalPipeline	**62.50**	0.22	**0.36**	5.84	3.19	3.16	3.84	3.06	**5.00**	3.84	2.04
DebateVoting	50.00	0.12	0.24	5.13	3.03	3.12	2.91	3.06	**5.00**	7.48	2.89
MICA-core (ours)	**62.50**	0.13	0.25	**13.49**	**3.31**	**3.84**	**4.34**	**3.69**	**5.00**	**0.68**	**1.53**
Overall Average											
SharedMemory	43.75	0.21	0.21	8.60	2.51	2.62	3.71	2.48	**5.00**	3.53	3.23
CentralizedBroadcast	40.63	**0.34**	**0.42**	16.79	2.84	3.10	4.16	2.84	**5.00**	3.58	2.94
HierarchicalPipeline	41.70	0.22	0.29	13.17	2.67	2.76	3.80	2.70	**5.00**	3.54	4.12
DebateVoting	51.25	0.29	0.35	12.58	2.75	2.98	3.96	2.89	**5.00**	6.97	11.47
MICA-core (ours)	**63.13**	0.30	0.37	**19.12**	**3.17**	**3.33**	**4.22**	**3.26**	**5.00**	**0.71**	**2.05**

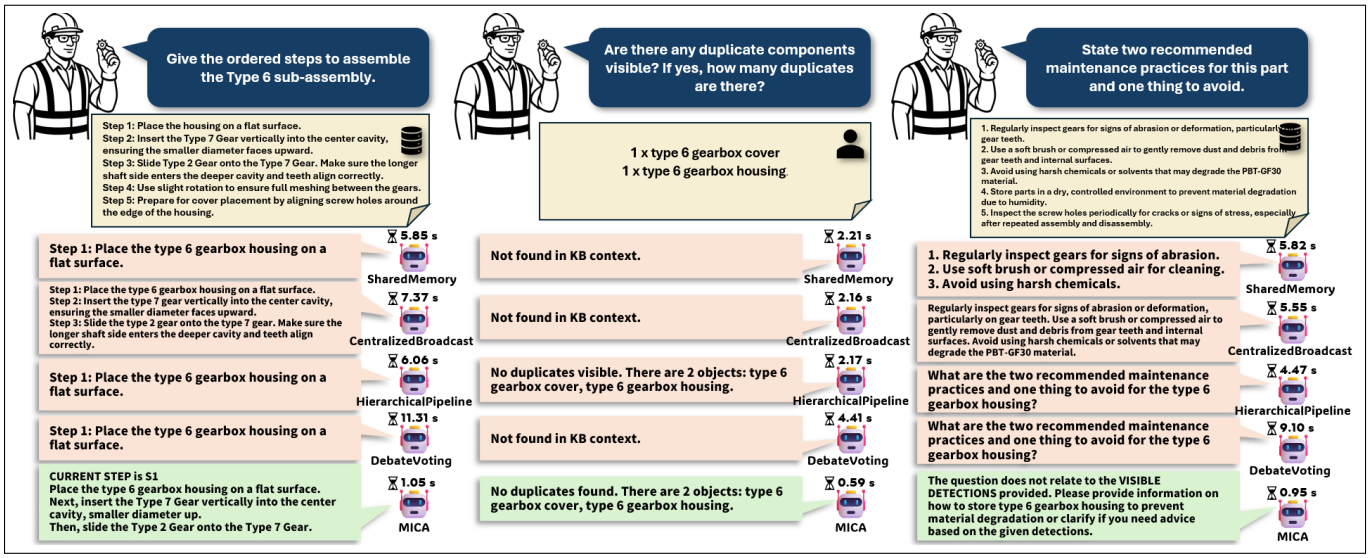

Fig. 3. Qualitative comparison of four representative multi-agent topologies (*SharedMemory*, *CentralizedBroadcast*, *HierarchicalPipeline*, *DebateVoting*) against *MICA* on three representative queries.

CentralizedBroadcast peaks on assembly (TS 46.88%), benefiting from synchronized access to step context, at the cost of higher energy consumption caused by parallel yet redundant agent activations. *DebateVoting* excels on part attributes (TS 96.88%, BL 0.57, RG 0.62), where surface lexical correctness dominates and peer critique can sharpen phrasing; however, it degrades on assembly and maintenance, as repeated critique on partially incorrect premises amplifies noise and increases latency and energy. *HierarchicalPipeline* delivers coherent but brittle outputs: once an upstream error occurs, downstream agents have no mechanism to correct it, which explains its stable yet moderate scores. MICA leads on general (TS 90.63%) and fault handling (TS 62.50%) through KB grounding and adaptive routing; its conservative router occasionally under-recalls in maintenance, which accounts for the weaker relative score in that category.

(ii) Error modes and router sensitivity. Failure cases reveal diagnostic patterns. Ambiguous phrasing and domain synonyms weaken intent signals and lead to conservative routing to a KB-grounded agent. The answer remains factual but may be incomplete relative to the success criterion, reducing TS. *SharedMemory* benefits from accumulated cross-agent co-occurrence, while *CentralizedBroadcast* mitigates misrouting by exposing the same context to all agents, although both incur higher latency or energy.

(iii) BLEU/ROUGE versus grounded quality. CentralizedBroadcast attains higher BLEU/ROUGE (0.34/0.42) than MICA (0.30/0.37) yet underperforms in KBA (16.79% versus 19.12%). This reflects a tendency to produce longer, templated responses that overlap lexically with references but deviate from KB facts. MICA enforces canonical terminology and safety auditing, yielding concise, action-oriented outputs with lower surface overlap yet stronger factual alignment. GPT-based judgments confirm that lexical overlap is an unreliable proxy for procedural quality in safety-critical tasks, motivating the use of KBA in this benchmark.

(iv) Efficiency and utility. Resource measurements reveal clear trade-offs. *DebateVoting* and *CentralizedBroadcast* incur high latency (6.97 s and 3.58 s) and energy (11.47 kJ and 2.94 kJ), consistent with redundant agent activations. *SharedMemory* also suffers high latency (3.53 s) due to evaluator overhead. MICA's sparse activation yields approximately 0.71 s responsiveness and the lowest energy cost (2.05 kJ). These results expose a three-way frontier among grounded quality, coordination accuracy, and efficiency; *MICA* occupies the region most suitable for deployment.

Overall, the benchmark shows that while individual baselines exhibit narrow advantages, MICA uniquely balances factual alignment, efficiency, and adaptivity for real-world assistance.

E. Qualitative Results

To complement the quantitative benchmark, we present targeted case studies that reveal how perception grounding, router-based specialization, and ASF shape system behavior across distinct query types. Fig. 3 compares *SharedMemory*, *CentralizedBroadcast*, *HierarchicalPipeline*, *DebateVoting*, and *MICA* on three queries.

a) Assembly-related. The KB contains a canonical sequence, yet *MICA* does not copy it. The router sends the query to the Assembly Guide, which conditions on detected components and rewrites the steps into a concise, user-oriented list rather than a raw KB block. *SharedMemory* and *CentralizedBroadcast* often yield truncated or fragmented sequences due to evaluator selection and hub aggregation; *HierarchicalPipeline* propagates early omissions; *DebateVoting* increases delay without gains. These outcomes follow the coordination mechanics: single-agent routing in *MICA*, shared evaluator, hub aggregation, fixed relays, and peer debate with voting.

b) General. The duplicate-check has no KB entry and must rely on perception. *MICA* correctly reports two distinct objects with no duplicates by routing to a generalist agent that answers from detections, avoiding reliance on retrieval. Baselines fail with "not found in KB" or misread detections because their decision paths prioritize retrieval and cross-agent aggregation over perception-grounded routing.

c) Maintenance-related. A diagnostic failure occurs when *MICA* misroutes to a detection-focused agent, producing a factual but intent-mismatched answer. The safety checker still audits outputs and prevents unsafe advice. *SharedMemory* and *CentralizedBroadcast* succeed by exposing the same KB content to multiple specialists and selecting or merging a maintenance response, at the cost of higher latency. This shows the trade-off: sparse routing in *MICA* yields efficiency and interpretability, yet intent ambiguity can reduce task success if dispatch is incorrect.

Overall, these cases highlight the synergy between ASF-driven procedural grounding and router-based specialization in *MICA*: with KB support, steps are reformulated for clearer execution; without KB, perception-grounded routing yields accurate answers; and when routing errs, failures remain attributable and auditable.

V. Conclusion

We presented MICA, a multi-agent industrial coordination assistant that unifies perception-grounded reasoning, adaptive step understanding, and speech-based interaction for real-time factory support. Our contributions include Adaptive Step Fusion (ASF), which enables continual step-level adaptation through expert blending and speech feedback, and a benchmark with tailored evaluation metrics for systematic comparison of multi-agent coordination strategies. Experiments show that MICA consistently improves task success, reliability, and responsiveness over representative baselines while remaining practical for offline deployment on resource-constrained hardware. Beyond these gains, MICA suggests a pathway toward deployable, privacy-preserving industrial assistants capable of adapting to dynamic workflows. Future work will extend user studies, improve robustness under perception noise and industrial acoustic conditions, and explore deployment on embedded edge platforms.

REFERENCES

[1] M. Capponi, *et al.*, "Assembly complexity and physiological response in human-robot collaboration: Insights from a preliminary experimental analysis," *RCIM*, 2024.

[2] L. M. Daling and S. J. Schlittmeier, "Effects of augmented reality-, virtual reality-, and mixed reality–based training on objective performance measures and subjective evaluations in manual assembly tasks: a scoping review," *Hum. Factors*, 2024.

[3] J. Gu *et al.*, "A survey on LLM-as-a-judge," *The Innovation*, 2024.

[4] Y. Yao *et al.*, "A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly," *HCC*, 2024.

[5] Q. Wu, *et al.*, "AutoGen: enabling next-gen LLM applications via multi-agent conversations," in *COLM*, 2024.

[6] C. Qian *et al.*, "ChatDev: Communicative agents for software development," in *ACL*, 2024.

[7] S. Hong *et al.*, "MetaGPT: Meta programming for A multi-agent collaborative framework," in *ICLR*, 2024.

[8] Y. Du *et al.*, "Improving factuality and reasoning in language models through multiagent debate," in *ICML*, 2023.

[9] S. Yao *et al.*, "Tree of thoughts: Deliberate problem solving with large language models," in *NeurIPS*, 2023.

[10] D. Zhou *et al.*, "Least-to-most prompting enables complex reasoning in large language models," in *ICLR*, 2023.

[11] D. Wen *et al.*, "Snap, segment, deploy: A visual data and detection pipeline for wearable industrial assistants," in *SMC*, 2025.

[12] C. Plizzari, *et al.*, "An outlook into the future of egocentric vision," *IJCV*, 2024.

[13] Y. Li *et al.*, "EgoCross: Benchmarking multimodal large language models for cross-domain egocentric video question answering," in *AAAI*, 2026.

[14] D. Zhang *et al.*, "EgoNight: Towards egocentric vision understanding at night with a challenging benchmark," in *ICLR*, 2026.

[15] D. Damen, *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *ECCV*, 2018.

[16] K. Grauman *et al.*, "Ego4D: Around the world in 3,000 hours of egocentric video," in *CVPR*, 2022.

[17] L. Ma, *et al.*, "Nymeria: A massive collection of multimodal egocentric daily motion in the wild," in *ECCV*, 2024.

[18] D. Hollidt *et al.*, "EgoSim: An egocentric multi-view simulator and real dataset for body-worn cameras during motion and activity," in *NeurIPS*, 2024.

[19] T. Nagarajan *et al.*, "EgoEnv: Human-centric environment representations from egocentric video," in *NeurIPS*, 2023.

[20] S. Zhou *et al.*, "EgoTextVQA: Towards egocentric scene-text aware video question answering," in *CVPR*, 2025.

[21] C. Zhang *et al.*, "EgoSG: Learning 3D scene graphs from egocentric RGB-D sequences," in *CVPR*, 2024.

[22] R. Liu *et al.*, "ObjectFinder: An open-vocabulary assistive system for interactive object search by blind people," *arXiv preprint arXiv:2412.03118*, 2024.

[23] J. Zheng *et al.*, "MateRobot: Material recognition in wearable robotics for people with visual impairments," in *ICRA*, 2024.

[24] T. Do *et al.*, "Egocentric scene understanding via multimodal spatial rectifier," in *CVPR*, 2022.

[25] Y. Huang, *et al.*, "Vinci: A real-time embodied smart assistant based on egocentric vision-language model," *arXiv preprint arXiv:2412.21080*, 2024.

[26] Y. Zhang, *et al.*, "Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration," in *AAAI*, 2025.

[27] Y. Yu *et al.*, "FinCon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making," in *NeurIPS*, 2024.

[28] Y. Wu *et al.*, "A novel joint optimization method of multi-agent task offloading and resource scheduling for mobile inspection service in smart factory," *TVT*, 2024.

[29] Z. Nie and K.-C. Chen, "Predictive path coordination of collaborative transportation multirobot system in a smart factory," *TSMC*, 2024.

[30] L. Wang, *et al.*, "Multi-agent cooperative swarm learning for dynamic layout optimisation of reconfigurable robotic assembly cells based on digital twin," *J. Intell. Manuf.*, 2024.

[31] J. Lim, B. Vogel-Heuser, and I. Kovalenko, "Large language model-enabled multi-agent manufacturing systems," in *CASE*, 2024.

[32] O. Antons and J. C. Arlinghaus, "Designing distributed decision-making authorities for smart factories–understanding the role of manufacturing network architecture," *IJPR*, 2024.

[33] V. Siatras, *et al.*, "Production scheduling based on a multi-agent system and digital twin: a bicycle industry case," *Information*, 2024.

[34] G. Li *et al.*, "CAMEL: Communicative agents for "mind"' exploration of large language model society," in *NeurIPS*, 2023.

[35] C. Packer *et al.*, "MemGPT: Towards LLMs as operating systems," *arXiv preprint arXiv:2310.08560*, 2023.

[36] X. Wang *et al.*, "Self-consistency improves chain of thought reasoning in language models," in *ICLR*, 2023.

[37] J. S. Park, *et al.*, "Generative agents: Interactive simulacra of human behavior," in *UIST*, 2023.

[38] J. Becker, "Multi-agent large language models for conversational task-solving," *arXiv preprint arXiv:2410.22932*, 2024.

[39] Y. Yang *et al.*, "Minimizing hallucinations and communication costs: Adversarial debate and voting mechanisms in LLM-based multi-agents," *Applied Sciences*, 2025.

[40] X. Bo, *et al.*, "Reflective multi-agent collaboration based on large language models," in *NeurIPS*, 2024.

[41] R. Khanam and M. Hussain, "YOLOv11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[42] L. Yang *et al.*, "Depth anything V2," in *NeurIPS*, 2024.

[43] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[44] Q. Team, "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[45] A. Radford *et al.*, "Robust speech recognition via large-scale weak supervision," in *ICML*, 2023.

[46] nateshmbhat, "pyttsx3," https://github.com/nateshmbhat/pyttsx3, 2024, python text-to-speech library.

[47] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[48] M. Douze, *et al.*, "The faiss library," *arXiv preprint arXiv:2401.08281*, 2024.

[49] C. Guo, *et al.*, "On calibration of modern neural networks," in *ICML*, 2017.

[50] K. Papineni *et al.*, "BLEU: A method for automatic evaluation of machine translation," in *ACL*, 2002.

[51] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, 2004.

[52] C. Zhou *et al.*, "LIMA: Less is more for alignment," in *NeurIPS*, 2023.

[53] J. Wang *et al.*, "Is ChatGPT a good NLG evaluator? A preliminary study," *arXiv preprint arXiv:2303.04048*, 2023.

[54] OpenAI, "GPT-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.