FusionAgent: A Multimodal Agent with Dynamic Model Selection for Human Recognition

Supplementary Material

6. Additional Methodology

6.1. Group Relative Policy Optimization (GRPO)

In contrast to RL algorithms such as Proximal Policy Optimization (PPO) [41]— which rely on a critic model to assess policy performance, GRPO eliminates the need for the critic model by directly comparing groups of candidate responses. As shown in Fig. 2, for a given input x, GRPO requires the model to sample N diverse responses $\{o_1, o_2, \ldots, o_N\}$ from the curren t model π_θ and obtains overall rewards $\{r_1, r_2, \ldots, r_N\}$ for o_i based on the reward function $R(x, o_i)$. In our case, it can be formatted as:

$$\begin{aligned} R(x, o_i) = w_f R_f(o_i) + w_{tool} R_{tool}(o_i) \\ + w_{acc} R_{acc}(a_i, y) + w_{mat} R_{mat}(o_i), \end{aligned} \tag{7}$$

where w_f, w_{tool}, w_{acc}, and w_{mat} are the reward weights for format reward R_f, tool success reward R_{tool}, answer accuracy reward R_{acc}, and metric-based reward R_{mat}, respectively. GRPO assesses the relative quality by normalizing r_i using the mean and standard deviation of the group reward:

$$A_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}, \tag{8}$$

where A_i denotes the advantage of the i-th response. With the group normalization, GRPO encourages the model to sample preferred answers with a higher reward. The model is updated via:

$$\begin{aligned} J_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(O|q)} \\ \left[\frac{1}{G} \sum_{i=1}^{G} \min \left(\frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\text{old}}}(o_i \mid q)} A_i, \right. \right. \\ \left. \text{clip} \left(\frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\text{old}}}(o_i \mid q)}, 1-\varepsilon, 1+\varepsilon \right) A_i \right) \\ \left. - \beta D_{\text{KL}}\big(\pi_\theta \parallel \pi_{\text{ref}}\big) \right], \end{aligned} \tag{9}$$

where ε and β are the GRPO clipping hyperparameters and the coefficient weight for controlling the Kullback–Leibler (KL) penalty [41], respectively. π_{ref} is the reference model.

$$\frac{1}{1 + Euc(q, g)}. \tag{10}$$

Prompts. We provide the system prompt used in the agent training and inference in Fig. 9. The tool schema is the function documentation with input and output formats and meaning. Model type dict is the modality type of each biometric model.

6.2. Additional Reward Details

Format Reward. Let C be the number of assistant turns in a conversation, c_i denote the i^{th} turn response. Let $\text{match}(c_i, P)$=1 if c_i matches JSON tag <P>...</P>exactly (0 otherwise). We compute the multi-turn format reward by: $R_{\text{fmt}}(C)$=$\frac{1}{C}\sum_{i=1}^{C} cot(c_i)$ (CoT), where $cot(c_i)$=$\text{match}(c_i, \texttt{think}) \times \big(\text{match}(c_i, \texttt{answer}) \oplus \text{match}(c_i, \texttt{tool\_call})\big)$. $\oplus$ denotes exclusive-or.

For DA, the formulation is: $R_{\text{fmt}}(C)$=$\frac{1}{C}\sum_{i=1}^{C} da(c_i)$ (DA), where $da(c_i)$=$\text{match}(c_i, \texttt{answer}) \oplus \text{match}(c_i, \texttt{tool\_call})$, without the need of $<$ think $>$ tag.

Clarification of augmented model selection on R_{mat}. We set most rows in M_Q to M_{o_i} because TAR and FNIR depend on operating thresholds that are selected from *dataset-level* scores, rather than being determined by a single query. If we set $\gamma = 1.0$ (set all rows of M_Q to M_{o_i}), then each model combination yields exactly one metric-based reward value, which significantly reduces exploration. In contrast, the Rank-1 outcome for a given q can be assessed query-wise, whereas TAR/FNIR are much more sensitive to the global threshold selection. Therefore, we introduce γ to control this augmentation and balance stable threshold estimation with sufficient exploration.

6.3. Tool Design.

Tool design should be in an efficient way for agent learning. If the number of tools or the number of parameters of tools is large, it will become more difficult for the agent to execute the tools. In our scenario, the goal of the agent is to call different biometric recognition models to extract features based on the input. Therefore, we design a universal tool that is suitable for every biometric model. The tool takes sequential images, the biometric model name as the input, and returns the similarity matrix and the predicted label of the images.

Tool Results. FusionAgent only receives the predicted identity label from the tool execution, while the score vectors and selected models are stored within the system. We do not return the similarity scores or confidence weights of the predicted identity, as doing so was found to prematurely

Judge Prompt

Role & Objective
You are an expert-level biometric analysis agent. Your primary mission is to achieve the highest possible identification performance by strategically analyzing input images/videos and selecting the optimal combination of biometric models. Prioritize the model you think is the most suitable. Do not select the same model more than once. Your final answer should be a fused identity prediction based on the evidence from your chosen models.
Loop
Work step-by-step. Each turn you must output exactly TWO blocks—first <think>, then ONE action: <tool_call> or <answer>. Wait for <tool_result> before the next turn.
Strict Output Format (no extra text, no markdown)
1) <think>...</think><tool_call>{JSON}</tool_call>
2) <think>...</think><answer>...</answer>
Tag Rules
- <think> (required, first): Briefly describe what you get, and explain the current decision.
• If calling a tool: you MUST first analyze the input video's characteristics. Consider factors like: Is the face clearly visible? Is the subject close to the camera with high resolution, or far away and low-resolution? etc.
• If answering: summarize tools results, key evidence, and your final prediction.
- <tool_call>: JSON with exactly two keys, "name" and "parameters". You can call ONLY ONE tool per turn.
- <answer>: Identity: The ID of the recognized person.
Tools
You may call one or more functions to assist with the user query.
You are provided with function signatures within <tools></tools> XML tags: <tools> {TOOL_SCHEMA} </tools>
For each function call, return a json format object with function name and arguments within <tool_call></tool_call> XML tags: <tool_call> {{"name": <function-name>, "parameters": <args-json-object>}} </tool_call>. Only call declared tools.
Model Type
You have access to a suite of specialized models. Your key challenge is to understand when to use them for maximum impact: {MODEL_TYPE_DICT}
Stopping Condition
End with <answer> when evidence is sufficient. Never invent tool outputs or identities.

Figure 9. System prompt for FusionAgent.

halt the exploration of model combinations during training. This design choice is justified because a high confidence or similarity score from one model does not preclude further performance gains through fusion with other models, given that each model contributes independently.

Error Handling. When the tool calling fails, for example, calling a wrong model name or an invalid JSON format. It is important to let the agent know the reason for the failure during training. We design error handling for unexpected behaviors and enable the agent to identify the reason for the failures.

7. Additional Implementation Details

Datasets. The dataset statistics are summarized in Tab. 7. They comprise multi-view captures and cross-modal biometric data, enabling rigorous evaluation of generalization across diverse resolutions, viewpoints, and temporal dynamics. This comprehensive benchmarking setup ensures robustness against real-world challenges such as occlusion, motion blur, and sensor heterogeneity, thereby validating the practicality of the proposed approach in unconstrained environments. During training, we only use $2,000$ samples (or medias) maximum for each dataset to perform training set score matrices.

Dataset	Type	#Subjects (Train/Test/Non-mated)	#Query	#Gallery
CCVID	Video	75 / 151 / 31	834	1074
MEVID	Video	104 / 54 / 11	316	1438
LTCC	Image	77 / 75 / 15	493	7050

Table 7. Statistics of the evaluation set of human recognition benchmarks. The number of query and gallery indicate the number of images/sequences for image/video datasets.

Model Pools. We follow [68] to construct the same model pools: AdaFace [22] (ViT-Base, WebFace4M), CAL [11] (ResNet50, CCVID/MEVID/LTCC), Big-

Dataset	#Media (Full)	#Media (10-shot)	Percentage (%)
LTCC	9576	768	8.0

Table 8. Comparison of the full dataset size and the few-shot (10-shot) subset used for training. The percentage indicates the proportion of data used in the few-shot setting relative to the full dataset.

Gait [57] (DINOv2-Small, CCPG), AIM [55] (ResNet50, LTCC), and AGRL [53] (ResNet50, MEVID), where the former denotes the model architecture and the latter indicates the training dataset. We use $\{model\}\_\{dataset\}$ to denote the difference of checkpoints (*i.e.*, CAL_CCVID and CAL_LTCC) during training and inference.

Center Features of Training Set. We follow QME [68] to extract center features as the gallery features for the training set. We compute the center features based on the subject ID, camera ID, and clothing ID. Therefore, each subject may have multiple center features.

Similarity Distances of Each Model. We follow QME [68] to measure the distances between features. AdaFace [22], CAL [11], AIM [55], AGRL [53], and CLIP3DReID [29] use cosine-similarity to measure the distance, while BigGait [57] uses Euclidean distance. We use Eq. 10 to transform Euclidean distance into similarity scores.

Cross-domain Training. We conduct cross-domain evaluation through zero-shot testing and few-shot training. For the few-shot setting, we adopt a 10-shot protocol (*i.e.*, each training subject provides 10 images/videos). The same procedure is applied to extract center features and score matrices from the training set, and only the 10-shot dataset is used for FusionAgent training. Few-shot data size for each dataset is shown in Tab. 8

Additional In-Domain Evaluation.

Accuracy From Agent Answer. As shown in Table 9, the *Agent* predictions are derived by selecting identity labels from the outputs of different tools, rather than directly relying on the score fusion results. Despite this discrete decision process, the agent achieves performance comparable to the score-based fusion method (ACT). We attribute this performance improvement primarily to the use of the answer accuracy reward. However, ACT consistently yields higher accuracy across all datasets, demonstrating that score-level fusion effectively integrates complementary information from multiple models and provides more reliable identity estimation than selection based solely on predicted labels. However, since the agent's answer can only reflect label prediction accuracy—and not other metrics such as ranking quality or calibration—different evaluation criteria may be adopted depending on application requirements. This result also provides a future direction on whether MLLMs or agents can even have a better performance w.r.t. metric results like verification (TAR) and open-set search (FNIR).

Method	CCVID	MEVID	LTCC
ACT	93.4	79.4	98.9
Agent	81.8	76.9	96.8

Table 9. Answer accuracy (Rank 1) performance on CCVID, MEVID, and LTCC. ACT is the Rank 1 result evaluated from the score matrix. Agent is the accuracy evaluated from the agent responses.

Method	*Time/sample (s)*			
	CCVID	**MEVID**	**LTCC**	**Avg**
QME [68]	0.72	0.66	0.64	0.67
FusionAgent (DA)	1.20	0.98	0.91	1.03
FusionAgent (CoT)	2.80	2.43	3.19	2.81

Table 10. Time-consuming Comparison of FusionAgent on Different Datasets. [Keys: DA=Direct answering.]

However, we do not observe the performance gain on CCVID. We hypothesize this is due to the performance gap between the training set and the test set. CAL is trained on CCVID, but AdaFace is not. CAL has a better Rank1 result on the training set, which makes the agent more reliant on the decision on CAL.

Time-consuming. Since the agent only decides the model combination, the task itself is relatively simple. Considering the demand for faster responses in practical applications, we adopt a lightweight 3B model for efficiency. As shown in Tab. 10, the CoT inference of FusionAgent, including tool executions and score-fusion, takes 2.81s per sample on a H100 device, while Direct Answering takes 1.03s. For comparison, QME [68] takes 0.67s per sample on average, including tool executions, quality estimating, and score-fusion.

Tool-call Efficiency. On CCVID, MEVID, and LTCC, FusionAgent reduces tool uses by 32.1%, 32.1%, and 31.3% compared to run-all baselines, respectively.

Advantage of ACT. Tab. 4 and 5 reveal two regimes. With a small domain gap, ACT pairs well with FusionAgent as the anchor prior is reliable. When the prior is miscalibrated, FarSight without anchor selection is more robust. Overall, ACT can reach higher performance, whereas FarSight is more robust under uncertainty.

7.1. Additional Ablation Experiments

Confidence Weights. Tab. 11 ablates the effect of confidence weights in ACT. Compared to Z-score, Min–max normalization underperforms by 0.9% points in mAP and 0.4% in Rank-1, with an even larger drop observed in FNIR

Norm.	Rank1	mAP	TAR@1%FAR	FNIR@1%FPIR
None	75.0	40.8	36.5	62.4 ± 9.2
Min-max	75.1	40.5	36.6	60.8 ± 10.7
Z-score	75.5	41.4	36.5	51.0 ± 9.4

Table 11. **Ablation on confidence weighting strategies in ACT on LTCC.** Both Min-max and Z-score normalization improve over no weighting, with Z-score achieving the best overall performance and substantially reducing FNIR. [Keys: Norm.= the method for confidence weights.]

(9.8%). Omitting confidence weighting leads to the largest FNIR degradation (62.4), confirming that confidence-aware scaling is essential in open-set search. Among the strategies, Z-score consistently yields the best overall results, reducing FNIR by more than 10% over other variants. This advantage likely stems from its robustness to outliers, which allows for more stable calibration across heterogeneous models. These findings indicate that while confidence weighting has a limited effect on closed-set metrics (Rank-1, TAR), it plays a critical role in improving reliability under stricter false-positive constraints.

Effects of Top-k Selection. As shown in Tab. 12, the effect of Top-k selection is consistent across all three datasets. The overall score is computed as the sum of Rank-1, mAP, and TAR, minus FNIR. On CCVID, applying Top-k selection brings clear improvements in Rank-1 accuracy (92.3→93.4), TAR (83.3→85.8), and the overall score (64.6→65.5), while maintaining comparable FNIR. For MEVID, the performance remains relatively stable, with a slight increase in Rank-1 (54.1→54.7) but minor fluctuations in other metrics. Similarly, on LTCC, Top-k selection provides marginal gains in Rank-1 (75.1→75.5) and overall score (25.5→25.6), with negligible changes in TAR and FNIR. Overall, Top-k selection consistently achieves a more favorable trade-off and leads to a better aggregated performance across datasets, confirming its robustness and general effectiveness.

Effects of Top-k Values on Training Set. We visualize the performance comparison on the LTCC training set in Fig. 10. The results show that the overall performance gradually improves as k increases and reaches its peak at $k = 40$, after which further growth of k does not yield additional benefits. Therefore, we adopt $k = 40$ for the testing stage to achieve a good balance between effectiveness and stability. We follow the same strategy when selecting Top-k values for the other datasets as well.

Anchor Sensitivity in ACT. Performance of ACT is more sensitive to anchor selection when single-model performance varies widely (Tab. 13).

Top-k	Rank1	mAP	TAR	FNIR	Overall
			CCVID		
✗	92.3	92.5	83.3	9.7 ± 1.3	64.6
✓	93.4	92.7	85.8	9.9 ± 1.5	65.5
			MEVID		
✗	54.1	29.1	35.6	58.2 ± 8.5	15.2
✓	54.7	28.7	34.9	58.6 ± 7.4	14.9
			LTCC		
✗	75.1	41.8	37.2	51.9 ± 9.4	25.5
✓	75.5	41.4	36.5	51.0 ± 9.4	25.6

Table 12. **Effects of Top-k selection.** Top-k selection gains a better overall performance on three datasets.

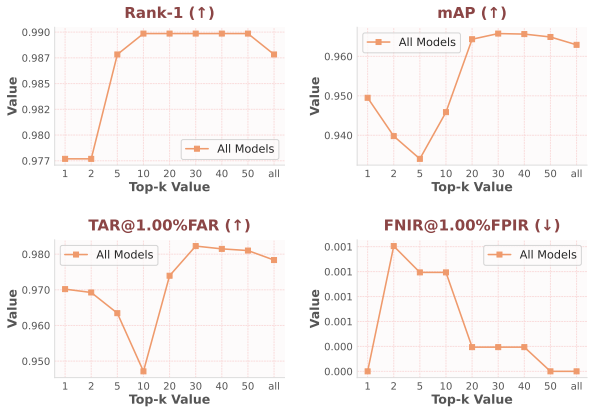

Figure 10. **Effects of Top-k Values on LTCC Training Set.** The overall performance reaches the peak when $k = 40$.

Anchor	Overall (CCVID)
AdaFace	260
BigGait	259
CAL	257
	Overall (LTCC)
AdaFace	77
AIM	95
CAL	102
	Overall (MEVID)
AdaFace	59
AGRL	65
CAL	65

Table 13. Anchor sensitivity.

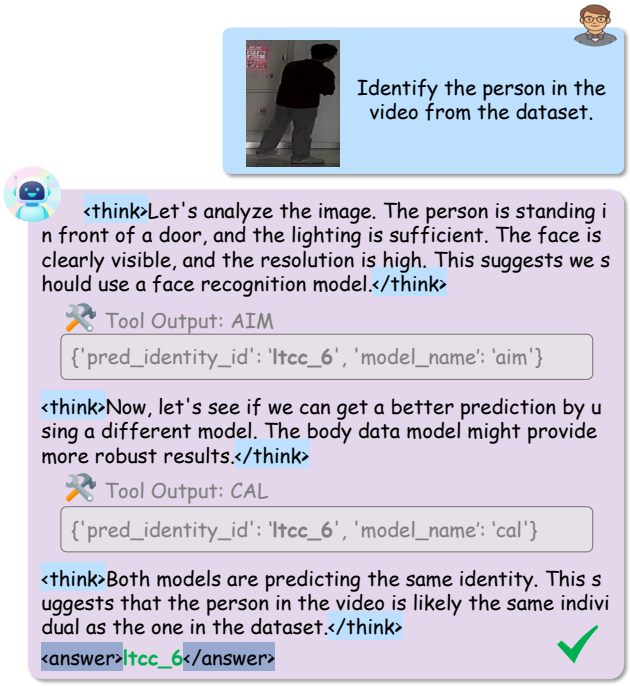

Figure 11. **CoT example of FusionAgent on LTCC.**

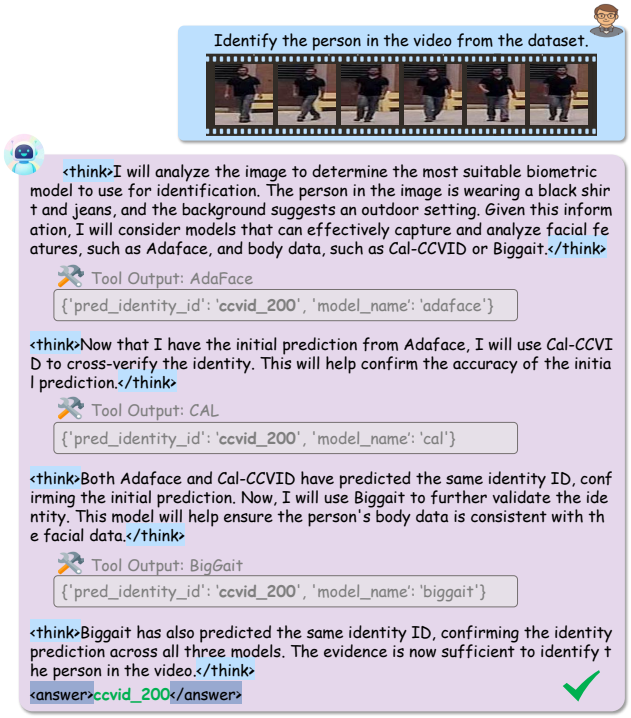

Figure 12. **CoT example of FusionAgent on CCVID.**

7.2. Additional Qualitative Results

Additional Conversations. Fig. 11 and 12 present additional CoT examples from FusionAgent. Depending on the input query pattern, FusionAgent dynamically selects the most suitable models. Fig. 13 illustrates the DA (direct answering) mode of FusionAgent, which bypasses reasoning generation to accelerate inference while maintaining comparable performance.

8. Limitations

Reasoning Collapse. RFT may lead to unstable or degenerate reasoning behaviors during training. For instance, the agent's reasoning content can become repetitive, disregarding the actual differences in model-predicted identities as training progresses. This phenomenon likely arises from *reward hacking*, since no explicit supervision is provided on the quality of reasoning. Stabilizing the reasoning process and ensuring consistent answer quality remains an important direction for future research.

Model Combination Estimation. In our setting, each sample can be associated with multiple possible model combinations. As the number of samples increases, the search space grows exponentially, making grid search for the ground-truth optimal combination computationally infeasible. Consequently, we adopt the proposed metric-based reward to estimate the overall performance without exhaustively enumerating all combinations. In the future, exploring more efficient or learning-based strategies for model combination estimation could further enhance scalability and accuracy.

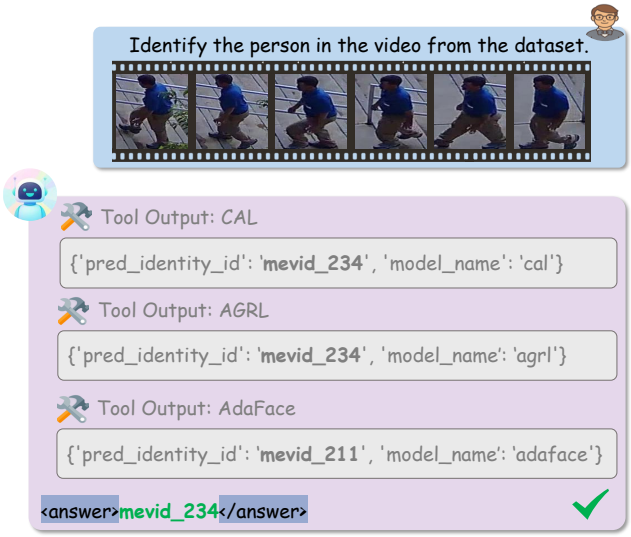

Figure 13. **DA example of FusionAgent on LTCC. FusionAgent directly outputs the tool-use code and answer without thinking.**

9. Potential Societal Impacts

Our paper leverages multiple public biometric datasets for research purposes, with a focus on the similarity score do-

main, which is less directly tied to sensitive biometric data. As biometric recognition tasks grow increasingly complex, integrating multiple models has become a key trend to enhance system performance. It is essential to ensure that the use of biometric datasets and recognition systems adheres to ethical standards and complies with privacy regulations.