2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Multi-View Gating Unit with KL-Based Alignment Toward Real-World Robot Control

Kei Igarashi[1] and Shingo Murata[1]

Abstract— **This paper proposes a framework for integrating latent representations from multi-view images, using adaptive weighting based on situational context to facilitate the generation of robot actions. Specifically, we introduce the multi-view gating unit (MGU), which assigns context-dependent weights to each dimension of the latent representations extracted from different viewpoints. By summing the corresponding dimensions across all viewpoints, we construct a fused latent representation that serves as input to a policy model. To enhance the effectiveness of the MGU and improve the accuracy of action generation, we incorporate a Kullback–Leibler (KL)-based alignment objective that encourages consistency between individual viewpoint representations and the fused representation. We evaluate the proposed framework through imitation-learning experiments in a kitchen-like real-robot environment across five tasks. The experimental results show that the MGU dynamically adapts to different contexts, thereby enabling successful task execution. Additionally, we compare our approach with a modified Action Chunking with Transformers (ACT) baseline and conduct an ablation study to assess the contribution of each component. The results show that our method achieves a task success rate of 84%, outperforming all baseline methods and validating the effectiveness of both the individual components and their integration within the proposed framework.**

I. Introduction

Robots are increasingly expected to operate in diverse real-world environments, but their applications remain largely limited to structured settings such as factories and industrial facilities. To enable more autonomous behavior, it is essential for robots to develop a detailed understanding of their surroundings. A promising approach is to enhance perception and control by leveraging multiple cameras, including top, side, and hand views. Yet, as the volume of acquired information grows, extracting useful features for robot control becomes increasingly difficult [1].

Although cameras are a common choice for robotic perception, occlusions caused by the robot itself or nearby objects can present significant challenges [2]–[7]. Various methods for overcoming these issues have been proposed, including dynamic camera-viewpoint selection based on task context [8], [9], multi-view information integration [10], [11], and multimodal sensor fusion that incorporates tactile, depth, or language inputs [12]–[14]. However, naive concatenation of these sources can cause critical information to be overwhelmed by irrelevant features, making it difficult to capture meaningful relationships across modalities.

*This work was supported by JST PRESTO Grant Number JPMJPR22C9 and JSPS KAKENHI Grant Number JP24K03012

[1] The authors are with the School of Integrated Design Engineering, Keio University, Yokohama, Kanagawa 223-8522, Japan murata@elec.keio.ac.jp

To tackle this challenge, prior research has investigated dynamic weighting using gating networks [15], representation-alignment methods for multi-view features [16]–[19], and attention-based keypoint-extraction methods [1], [20], [21]. However, no unified solution has emerged, underscoring the need for more effective strategies for multi-view integration.

In this work, we focus on gating mechanisms [15] for integrating multi-view image representations. We propose a novel gating mechanism that assigns separate weights to each latent dimension from different viewpoints, enabling dynamic and fine-grained fusion of multi-view information. We refer to this gating mechanism as the multi-view gating unit (MGU). To improve the stability and effectiveness of the MGU, we further incorporate a Kullback–Leibler (KL)-based latent alignment objective to align individual viewpoint representations with a fused multi-view representation. An overview of the proposed framework is shown in Fig. 1. To evaluate our approach, we conducted imitation-learning experiments in a kitchen-like environment across five manipulation tasks. We also compared our method against a modified Action Chunking with Transformers (ACT) baseline [22] and conducted ablation studies to assess the contribution of each model component.

II. Related Work

A. Robot Control Using Multiple Viewpoints

In deep learning-based robot control, camera images are widely used to perceive the environment [2]–[7]. However, occlusions—whether caused by the robot itself or surrounding objects—can hinder visibility, and a single camera viewpoint may not provide sufficient information for precise control. To overcome this, many studies have employed multiple camera viewpoints to achieve a more comprehensive perception of the environment [10], [11].

However, simply increasing the number of viewpoints does not guarantee improved control accuracy. Rather, the way in which information is integrated plays a critical role [23]. Various approaches have been proposed to address this challenge, including attention mechanisms and keypoint-based feature extraction [1], [20], [21], contrastive learning [16]–[19], feature regularization [23], and sensor dropout [24].

Akinola et al. [24] proposed a robot control method involving observations from three camera viewpoints. They compared different integration strategies and showed that applying sensor dropout during training improved generalization and manipulation accuracy. Hsu et al. [23] constructed a dataset, using both overhead and end-effector cameras, and

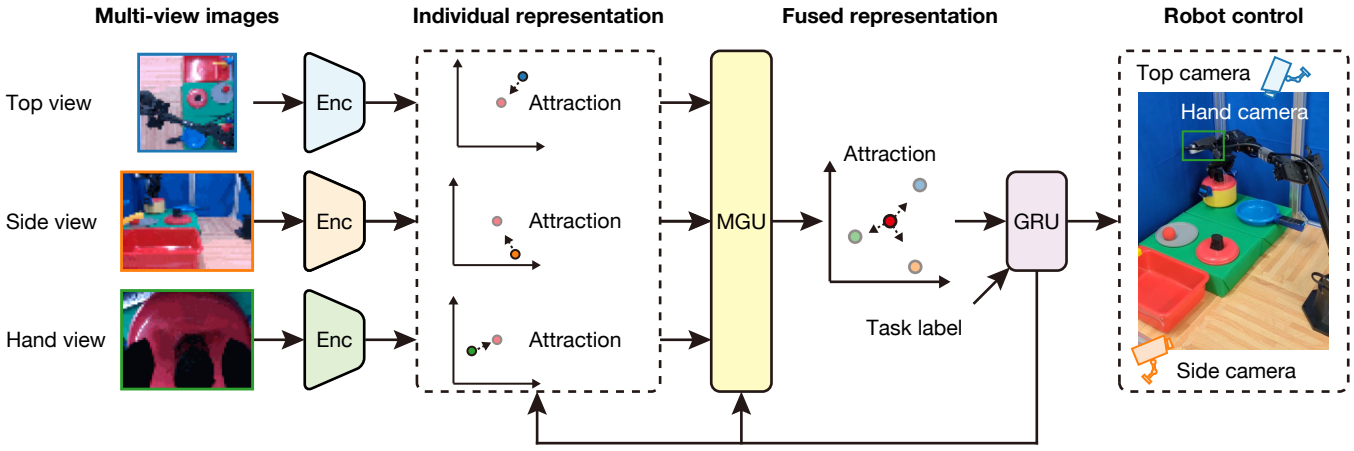

Fig. 1. Overview of the proposed framework. The model receives multi-view images, including top, side, and hand views, and generates action predictions to control the robot. The MGU assigns context-dependent weights to each dimension of the individual latent representations. By summing the weighted dimensions across all viewpoints, a fused latent representation is obtained. Based on this fused representation, the model predicts the robot's actions. The KL-based alignment objective encourages consistency between the individual and fused latent representations, thereby enhancing the effectiveness of the MGU.

investigated the impact of viewpoint selection and feature regularization. Their results indicated that when the end-effector view alone was sufficient for a task, adding an overhead view could actually degrade performance. They also examined regularization, using a variational information bottleneck (VIB) [25], [26], and found that selectively applying regularization to the overhead view led to the highest accuracy.

Zhao et al. [22] introduced ACT, an imitation-learning framework that integrates transformer architectures [27] with action-chunking techniques [28]. ACT uses joint angles and multi-view images to predict future action sequences. To improve control performance, it incorporates two mechanisms: action chunking, which treats sequences of multiple actions as single units, and temporal ensemble, which computes a weighted sum of predictions across different chunks for each timestep. These techniques enabled ACT to outperform baseline methods in terms of task success rate.

B. Gating Mechanisms

Gating mechanisms are widely used to integrate multimodal information, enabling the model to dynamically adjust the contribution of each modality based on context. This strategy has been applied in diverse domains, including image–text fusion for recognition tasks [15], [29] and multimodal integration of visual, tactile, and torque information for robot control [30], [31].

Arevalo et al. [15] proposed the gated multimodal unit, which integrates modalities by a sigmoid-based weighting mechanism. Anzai et al. [30] introduced deep gated multimodal learning for combining visual and tactile data; this method dynamically estimates the reliability of each modality and adjusts the integration accordingly, leading to improved pose estimation of grasped objects.

C. Representation Alignment for Multi-View Images

Representation alignment has become an important strategy for learning feature similarity and consistency across viewpoints. Contrastive learning, as a form of self-supervised learning, is one representative approach in this direction. In the context of robot control, it has increasingly been applied to multi-view image representations [32], [33].

Li et al. [32] combined neural radiance fields [34] with contrastive learning to improve the accuracy of viewpoint-invariant image prediction. Kinose et al. [33] applied contrastive learning to features extracted from two viewpoints and used them to train a world model [35]–[37]. This approach improved feature alignment across viewpoints, resulting in enhanced prediction performance.

D. Positioning of This Study

This study addresses the challenge of integrating multi-view camera inputs to improve robotic manipulation in real-robot settings. Although various approaches have been proposed for multi-view integration, we focus on adaptive weighting strategies based on gating mechanisms [15]. In particular, we leverage the MGU, which dynamically assigns weights to each latent dimension across different viewpoints. To further enhance representation alignment and gating effectiveness, we incorporate a KL-based alignment objective, encouraging consistency between viewpoint-specific and fused representations. By combining fine-grained gating with representation alignment, our framework offers improved adaptability across task phases, leading to superior control performance compared with prior methods such as the transformer-based ACT [22].

III. Proposed Method

The proposed model learns to generate robot actions from multi-view camera images. To effectively extract latent representations from these images, we employ the MGU in combination with a KL-based alignment objective. The

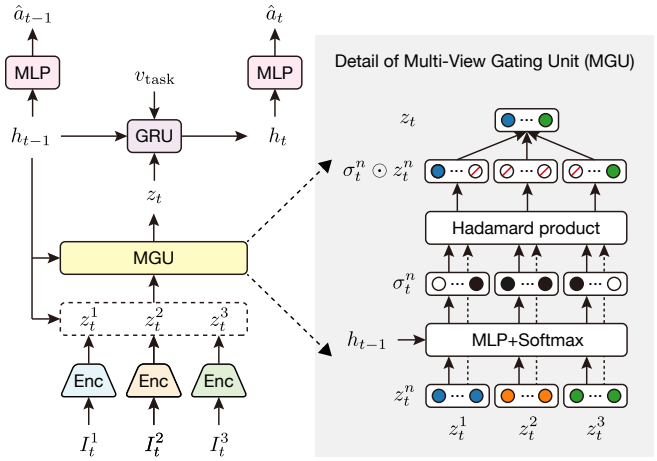

Fig. 2. Overview of the proposed model (left) and the details of the MGU (right). Given the N viewpoint camera images $I_t^{1:N}$ and the GRU hidden state h_{t-1}, the latent representations $z_t^{1:N}$ are obtained ($N = 3$ in the figure). These latent representations are passed through the MGU to compute the fused latent representation z_t. The fused latent representation z_t, the hidden state h_{t-1}, and the task label v_{task} are then input to the GRU to compute the updated hidden state h_t, from which the predicted action $\hat{a}_t$ is generated. The gating weights $\sigma_t^{1:N}$ are computed from the latent representations $z_t^{1:N}$ and the previous hidden state h_{t-1}, and are used to compute the fused latent representation z_t. For simplicity, the figure illustrates the Hadamard product between each gating weight σ_t^n and the corresponding latent representation z_t^n. In the actual implementation, the Hadamard product is first computed between each gating weight and the corresponding mean or variance of the latent representation distribution. Then, the weighted sums of the means and variances are computed and used as the parameters (mean and variance) for the fused latent representation distribution. Finally, the fused latent representation is sampled from this distribution.

MGU allows the model to assign weights to the latent representations obtained from each viewpoint, enabling it to dynamically focus on contextually relevant information. The alignment objective further enhances the MGU by encouraging consistency between the latent representations of individual viewpoints and the fused representation aggregated across all views.

A. Data Flow

An overview of the proposed model is shown in Fig. 2 (left). At each time step t, the model receives images $I_t^1, I_t^2, \ldots, I_t^N$ (collectively denoted as $I_t^{1:N}$ for simplicity) from N camera viewpoints, along with the GRU hidden state h_{t-1} from the previous time step. These inputs are used to compute viewpoint-specific latent representations $z_t^1, z_t^2, \ldots, z_t^N$ (denoted as $z_t^{1:N}$; see the next subsection for details).

These latent representations, together with h_{t-1}, are passed to the MGU, which produces the fused latent representation z_t (see the subsequent subsection for details). Next, the task identity is provided as a one-hot vector v_{task}, and the fused latent representation z_t, h_{t-1}, and v_{task} are input to a gated recurrent unit (GRU) to update the hidden state as follows:

$$h_t = \text{GRU}(z_t, h_{t-1}, v_{\text{task}}). \tag{1}$$

Finally, the updated hidden state h_t is passed through two output heads, each implemented as a separate MLP, to generate the current action prediction $\hat{a}_t$ and the auxiliary F-step-ahead prediction $\hat{a}_{t+F}$:

$$\hat{a}_t = \text{MLP}_\phi(h_t), \tag{2}$$

$$\hat{a}_{t+F} = \text{MLP}_\psi(h_t). \tag{3}$$

B. Extraction of Latent Representations from Viewpoint Images

At each time step t, the images $I_t^{1:N}$ from the N viewpoints are processed by separate image encoders. Each encoder extracts image features, which are combined with the GRU hidden state h_{t-1} to compute viewpoint-specific latent distributions:

$$p(z_t^n \mid I_t^n, h_{t-1}) = \mathcal{N}(\mu_t^n, v_t^n) \quad \text{for } n = 1, \ldots, N. \tag{4}$$

Here, μ_t^n and v_t^n represent the mean and variance, respectively, computed from I_t^n and h_{t-1}. A latent vector z_t^n is sampled from each distribution as follows:

$$z_t^n \sim \mathcal{N}(\mu_t^n, v_t^n). \tag{5}$$

C. Multi-View Gating Unit (MGU)

A schematic of the MGU is shown in Fig. 2 (right). The MGU takes the latent representations $z_t^{1:N}$ and the previous hidden state h_{t-1} as input. These are passed through an MLP with parameters θ, which outputs the gating weights $\sigma_t^{1:N}$. The output dimension of the MLP is $N \times D_{\text{latent}}$, where D_{latent} is the dimension of each latent vector.

The gating weight for the d-th dimension of viewpoint n is computed via softmax across viewpoints:

$$u_t = \text{MLP}_\theta(z_t^{1:N}, h_{t-1}), \tag{6}$$

$$\sigma_t^n[d] = \frac{\exp(u_t^n[d])}{\sum_{n'=1}^{N} \exp(u_t^{n'}[d])}. \tag{7}$$

Using these gating weights, the mean and variance of the fused latent distribution $p(z_t \mid I_t^{1:N}, h_{t-1}) = \mathcal{N}(\mu_t, v_t)$ are computed as follows:

$$\mu_t = \sum_{n=1}^{N} \sigma_t^n \odot \mu_t^n, \tag{8}$$

$$v_t = \sum_{n=1}^{N} \sigma_t^n \odot v_t^n, \tag{9}$$

where $\odot$ denotes Hadamard product.

The fused latent vector z_t is then sampled from this distribution:

$$z_t \sim \mathcal{N}(\mu_t, v_t). \tag{10}$$

D. Loss Function

The overall loss function L consists of three components: an action prediction loss, an auxiliary F-step-ahead prediction loss, and a KL-based alignment loss.

The action prediction loss is defined as the mean squared error (MSE) between the predicted action $\hat{a}_t$ and the expert action a_t. An auxiliary loss is also computed as the MSE

between the F-step-ahead prediction $\hat{a}_{t+F}$ and the corresponding expert action a_{t+F}.

The KL-based alignment term $\mathcal{J}_t^{\mathrm{KL}}$ encourages consistency between the fused latent distribution and each viewpoint-specific latent distribution. To prevent this alignment term from dominating the training and forcing the latent distributions to collapse prematurely, the summed KL divergence is clipped by an upper bound α as follows:

$$\mathcal{J}_t^{\mathrm{KL}} = \min \left\{ \sum_{n=1}^{N} D_{\mathrm{KL}}(p(z_t \mid I_t^{1:N}, h_{t-1}) \mid\mid p(z_t^n \mid I_t^n, h_{t-1})), \alpha \right\}. \tag{11}$$

The total loss is computed as a weighted sum over time steps as follows:

$$L = \sum_t \left[\lambda_1 \mathrm{MSE}(\hat{a}_t, a_t) + \lambda_2 \mathrm{MSE}(\hat{a}_{t+F}, a_{t+F}) + \lambda_3 \mathcal{J}_t^{\mathrm{KL}} \right]. \tag{12}$$

Model parameters are optimized via gradient descent to minimize the loss L.

IV. Experimental Setup

A. Experimental Environment and Hardware

The experimental environment is shown in Fig. 3 (left). It is designed to resemble a kitchen and includes a sink, pot, frying pan, plate, ball, and lid. The robot performs tasks while observing the workspace, using three cameras: a top-view camera mounted above the task area, a side-view camera mounted to the side, and a hand-view camera attached to the robot arm's end-effector.

Two WidowX-250 6-DOF robot arms (Trossen Robotics, Downers Grove, IL) were used in the experiment. Each arm has six joints and one gripper, for a total of seven degrees of freedom. During the expert data collection, one robot arm was fixed above the task space and designated as the follower, while the second robot arm, positioned away from the task area, acted as the leader. The tasks were executed using a leader–follower setup. In this study, the ground-truth action a_t used to supervise the model's predictions was defined as the joint angle command of the follower robot. During testing, only the follower arm was used, and control was performed using actions predicted $\hat{a}_t$ from the trained model. We did not use proprioceptive observations such as joint angles as policy inputs, in order to isolate the effect of multi-view visual fusion in the present study. Additionally, two joints that did not significantly affect task performance were fixed at constant angles during the experiments.

The cameras used in the experiment were all Intel RealSense Depth Camera D435 (Intel RealSense, Santa Clara, CA). For the top-view camera, areas outside the task space were cropped, resulting in images of size $480 \times 480 \times 3$, which were then resized to $48 \times 48 \times 3$ to speed up the training. For the side-view and hand-view cameras, images were captured at $480 \times 640 \times 3$ without cropping and resized to $48 \times 64 \times 3$ for the training.

B. Task and Dataset

To evaluate the proposed and baseline models, five tasks were defined: lid opening, frying pan transport, ball transport, lid closing, and lid transport. Fig. 3 (right) shows the common initial state for all tasks as well as the final state for each task. In all tasks, the robot grasped a designated object and transported it to a specified location. Each task was executed ten times during testing to evaluate the success rates.

For the data collection, each task was performed 36 times, yielding a total of 180 sequences of images and joint angle commands. Of these, 32 trajectories per task were used for training, with the remainder used for validation. Each sequence contained 60 time steps, and included a top-view image I_t^{top} of size (48, 48, 3), a side-view image I_t^{side} and a hand-view image I_t^{hand} of size (48, 64, 3), along with joint angle commands a_t consisting of four arm joint values and one gripper value.

C. Data Augmentation

Multiple data augmentation techniques were employed to improve training efficiency. Three types of augmentations were applied to the input images. First, random cropping was performed on each image by symmetrically trimming the edges along the vertical and horizontal axes. Second, brightness adjustment was applied by multiplying each image by a random scalar within a predefined range to simulate varying lighting conditions. Third, occlusion augmentation was introduced by replacing one randomly selected viewpoint image with an all-zero image of the same size in a subset of the training data. In addition, random noise was added to the ground-truth actions a_t to regularize the model's predictions.

D. Training Details

The number of viewpoints was set to $N = 3$, corresponding to the top-view, side-view, and hand-view cameras. The latent dimension for each viewpoint was set to $D_{\mathrm{latent}} = 16$.

Separate CNN-based encoders were used for each viewpoint. Each encoder consisted of three convolutional layers followed by three fully connected layers, with the final outputs being the mean and variance vectors of a 16-dimensional Gaussian distribution. The MGU was implemented as a simple MLP with one hidden layer, followed by output layers generating 16-dimensional gating weights for each viewpoint (σ_t^{top}, σ_t^{side}, and σ_t^{hand}). The hidden state dimension of the GRU was set to $D_{\mathrm{GRU}} = 64$.

The upper bound (clipping threshold) in Eq. (11) was set to $\alpha = 1.0$. The loss coefficients in Eq. (12) were set as follows: $\lambda_1 = 3 \times 10^3$, $\lambda_2 = 3 \times 10^3$, and $\lambda_3 = 10$. These coefficients were selected empirically so that the action prediction terms and the alignment term had comparable contributions during training. We used $F = 4$ for the auxiliary loss.

The number of training and validation samples was 160 and 20, respectively. The batch size was set to 20. Training was performed using the AdamW optimizer [38] with a learning rate of 0.002.

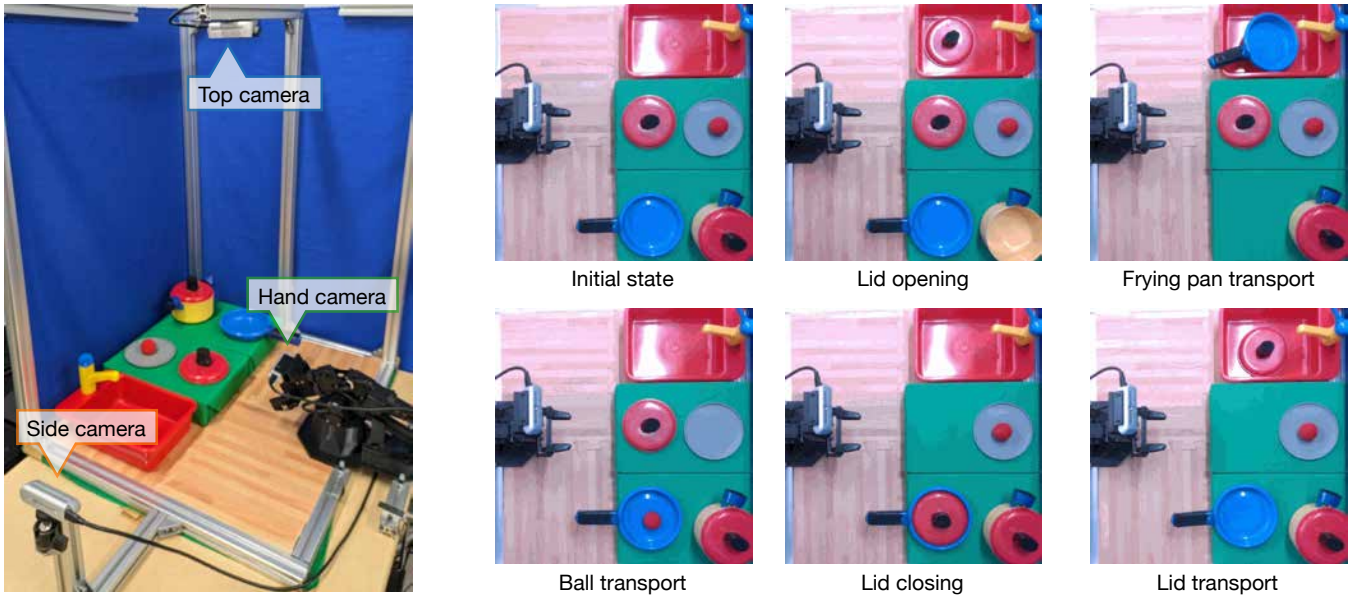

Fig. 3. Experimental environment (left) and task settings (right). The robot performs tasks in a kitchen-like environment, observed by top-view, side-view, and hand-view cameras. The initial object placements are consistent across all tasks. Five evaluation tasks were defined: (1) lid opening, (2) frying pan transport, (3) ball transport, (4) lid closing, and (5) lid transport.

V. Experiments

A. Comparison with a Modified ACT Baseline

To evaluate the effectiveness of the proposed framework, we conducted a comparative experiment with a modified ACT baseline [22], a robot imitation-learning method introduced in a related study. In its original form, ACT is not designed to handle multiple tasks that share the same initial state using a single model. To enable a fair comparison under the same conditions as our proposed method, we modified ACT by incorporating a 5-dimensional one-hot task label. This task label was concatenated with the image features from each viewpoint and provided as input to the transformer encoder of the action sequence decoder. We trained this modified version of ACT using the same dataset as our model and conducted 10 trials for each of the five tasks: lid opening, frying pan transport, ball transport, lid closing, and lid transport. We then compared the success rates of both methods.

B. Ablation Study on the MGU

To investigate the role of the MGU in the proposed framework, we evaluated the following two ablation models: (i) *Concatenation + KL alignment*, where the latent representations from all viewpoints were concatenated and directly input to the GRU without weighting; and (ii) *Uniform Gate + KL alignment*, where a single scalar gate weight was assigned per viewpoint and uniformly applied across all latent dimensions. Both models retained the KL-based alignment component and were trained under the same conditions as the proposed model. We then conducted 10 real-robot trials for each of the five tasks. The resulting success rates were compared against those achieved by the proposed model.

C. Ablation Study on KL-Based Alignment and Reconstruction Learning

To evaluate the effect of KL-based alignment and reconstruction learning in the proposed framework, we compared three ablation models: (i) *MGU + Reconstruction*, which used only image reconstruction learning without KL-based alignment; (ii) *MGU + KL alignment + Reconstruction*, which employed both KL-based alignment and reconstruction learning; and (iii) *MGU only*, which used neither. All three models used the same gating mechanism (MGU) as the proposed model, in which per-dimension gate weights are assigned for each viewpoint. Each variant was trained using the same procedure and dataset as the proposed method, and evaluated through 10 real-robot trials for each of the five tasks. We compared the success rates of these models to evaluate the effectiveness of the learning objectives.

VI. Results and Discussion

A. Task Execution Accuracy of Each Model

Table I shows the success rates for each task with the different models. As shown in Table I, the proposed model achieved the highest overall success rate. This result indicates that the two key components of the proposed framework—the per-dimension gating mechanism and the KL-based alignment objective—effectively contributed to performance.

The proposed model also outperformed the modified ACT baseline, which serves as a strong baseline based on a transformer architecture. This suggests that the combination of the gating mechanism and KL-based alignment in our method enables more effective extraction and weighting of essential information compared with ACT.

In addition, the proposed model performed better than the ablation model using simple concatenation of features from

TABLE I
TASK SUCCESS RATES WITH EACH MODEL.

Model	Lid Opening	Frying Pan Transport	Ball Transport	Lid Closing	Lid Transport	Overall
Proposed (MGU + KL alignment)	**9/10**	**9/10**	**5/10**	**10/10**	9/10	**84%**
ACT [22] (modified)	5/10	**9/10**	4/10	7/10	8/10	66%
Concatenation + KL alignment	6/10	8/10	4/10	6/10	8/10	64%
Uniform Gate + KL alignment	3/10	8/10	3/10	6/10	7/10	54%
MGU + Reconstruction	6/10	7/10	3/10	3/10	4/10	46%
MGU + KL alignment + Reconstruction	6/10	7/10	3/10	**10/10**	8/10	68%
MGU only	5/10	8/10	4/10	7/10	**10/10**	68%

the three viewpoints. Interestingly, the simple concatenation model also outperformed the model with uniform gating, which assigns a single scalar weight per viewpoint and applies it uniformly across all feature dimensions. These results indicate that using identical weights across dimensions, even when they differ by viewpoint, reduces the model's flexibility and degrades performance.

With respect to the auxiliary learning objectives, the proposed model trained only with KL-based alignment achieved the best success rate. Among the ablation models, those trained with both KL-based alignment and reconstruction learning as well as those trained without either exhibited the same overall success rate, although the success rates varied across individual tasks. The model trained only with reconstruction learning had the lowest success rate. These findings suggest that, in the presence of the gating mechanism, KL-based alignment contributes more than reconstruction learning to performance gains and that the use of reconstruction loss alone may hinder task execution in this context.

B. Functionality of the Gating Mechanism

Fig. 4(A) presents example image inputs from each viewpoint during the lid-opening task. Fig. 4(B) shows the dynamics of the gating weights (σ_t^{top}, σ_t^{side}, and σ_t^{hand}) across all 16 dimensions for each viewpoint. From Fig. 4(B), we can see that the model's attention to each viewpoint varies over time, generally following the order of top view, hand view, and side view. Moreover, within each viewpoint, the gating weights across the 16 latent dimensions exhibit temporal variation, suggesting that the model dynamically adjusts the importance assigned to each dimension.

A closer inspection of Fig. 4(B) reveals that while overall trends in viewpoint attention are observable across dimensions, the patterns are not uniform. Instead, different latent dimensions emphasize different viewpoints at different time steps. This implies that the individual dimensions of the latent representation specialize in capturing distinct types of task-relevant information.

For a more detailed analysis, Fig. 4(C) shows the gating weights averaged over all dimensions for each viewpoint across time. Clear trends in viewpoint attention emerge. For instance, from time steps $t = 1$ to $t = 10$, the attention is focused primarily on the top view. This period corresponds to the robot arm moving toward the pot lid, where accurately identifying the pot's position from above is critical. In contrast, during time steps $t = 20$ to $t = 30$, the attention shifts to the hand view. This phase corresponds to the arm descending above the pot to grasp the lid, a phase that requires precise hand-eye coordination—hence the increased emphasis on the hand view.

These observations demonstrate that the proposed model dynamically modulates viewpoint attention according to the phase of the task. Such adaptability likely contributes to the improved success rates observed in task execution.

C. Evaluation of KL-based alignment

To examine the impact of KL-based alignment and reconstruction learning on the latent representations (z_t^{top}, z_t^{side}, and z_t^{hand}) for each viewpoint, principal component analysis was performed on the latent representations obtained during the lid-opening task. The results are shown in Fig. 5. The comparison includes the proposed model, the model with only reconstruction learning, the model with both KL-based alignment and reconstruction learning, and the model without KL-based alignment and reconstruction learning.

As shown in Fig. 5, the proposed model qualitatively exhibits more closely aligned latent representations across viewpoints, and these are located close to the fused representations. In contrast, the model trained only with reconstruction learning exhibits the most dispersed latent spaces among the three viewpoints. The model trained with both KL-based alignment and reconstruction learning shows intermediate behavior; its representations are more separated than those of the proposed model, but less so than in the reconstruction-only model. These results suggest that reconstruction learning tends to promote divergence among viewpoint-specific latent representations, likely because doing so helps reconstruct the input images more precisely. However, such divergence may interfere with the gating mechanism's ability to integrate consistent cross-viewpoint features.

Furthermore, the model without KL-based alignment and reconstruction learning exhibits closer latent representations between viewpoints compared with the model with reconstruction learning, despite the absence of KL-based alignment. This finding indicates that when using a gating mechanism, closer latent representations across viewpoints are beneficial for improving task success rates, which is consistent with the superior performance of the proposed model.

These qualitative observations suggest that better-aligned latent representations across viewpoints may facilitate

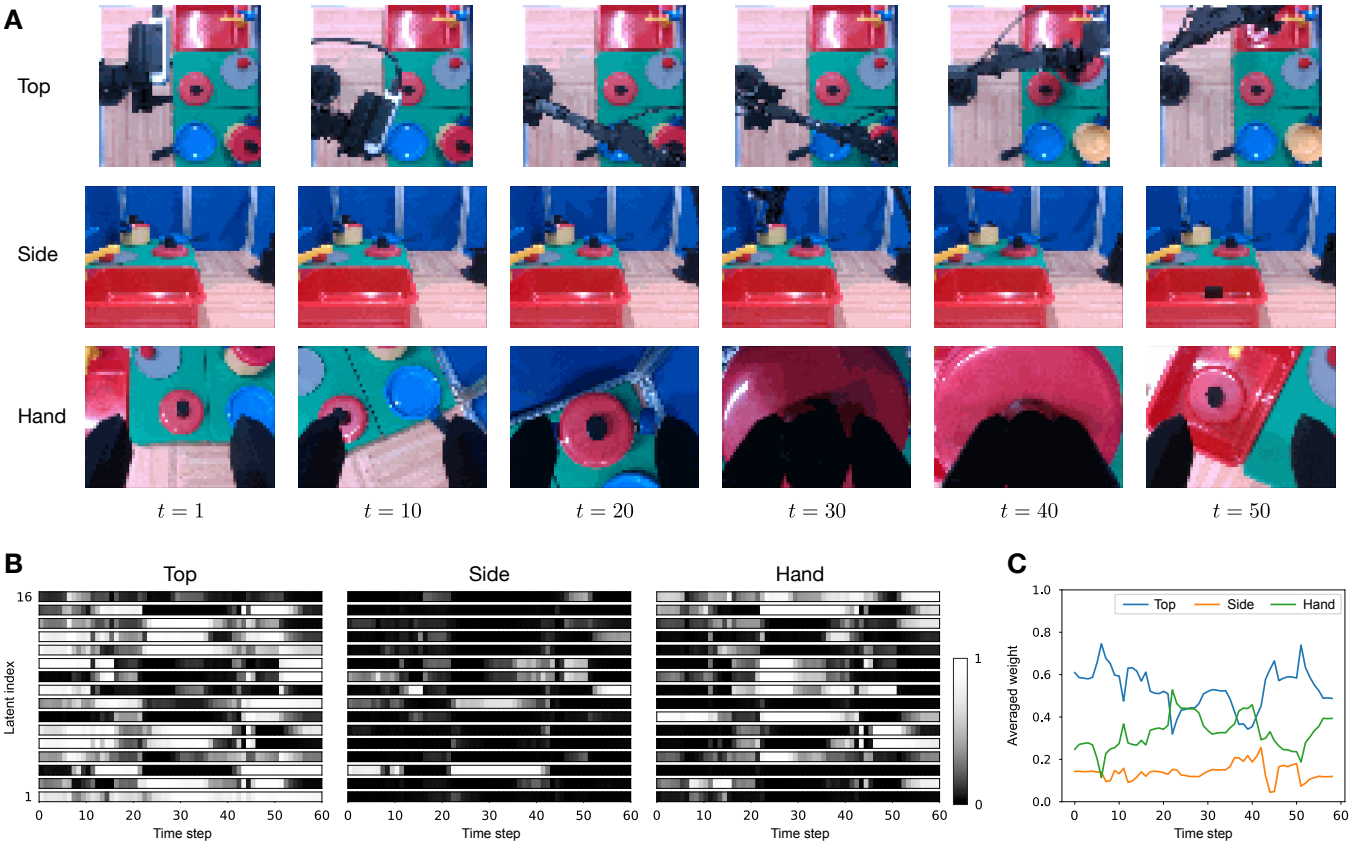

Fig. 4. Dynamics of the gating weights for each viewpoint. (A) Example images captured from each viewpoint camera during the lid-opening task. The first, second, and third rows show top-view (I_t^{top}), side-view (I_t^{side}), and hand-view (I_t^{hand}) images, respectively. (B) Time series of the 16-dimensional gating weights (σ_t^{top}, σ_t^{side}, and σ_t^{hand}) corresponding to the latent representations (z_t^{top}, z_t^{side}, and z_t^{hand}). The horizontal axis denotes the time step, and the vertical axis denotes the latent dimension index. (C) Averaged gating weights across the 16 dimensions for each viewpoint.

smoother adjustment of the gating weights during training. In contrast, when the latent representations of different viewpoints are highly separated, the model becomes more sensitive to the gating weights, increasing the risk of convergence to suboptimal local minima.

VII. Conclusions

This study proposed a method for effectively integrating multi-view images toward real-world robot control applications. Specifically, we introduced the MGU, a per-dimension gating mechanism that dynamically assigns weights to latent representations from each viewpoint. By combining the MGU with a KL-based alignment objective, the model is encouraged to align viewpoint-specific features with a fused representation, thereby enhancing gating performance and improving task execution.

We validated the proposed approach by conducting real-robot experiments in a kitchen-like environment across five manipulation tasks. The results showed that the proposed method achieved consistently high success rates and was able to adaptively focus on relevant viewpoints based on situational context. Comparative evaluations against the strong transformer-based ACT baseline [22] and ablated variants further confirmed the effectiveness of both the overall framework and its individual components.

Although the proposed method successfully enabled context-dependent attention switching via the MGU, the experimental settings were simplified to ensure clarity. Specifically, tasks involved fixed object types, constrained placement areas, and short execution horizons. For practical deployment, the method should be extended to support longer-horizon tasks in more complex environments such as searching for occluded or out-of-view objects. In addition, the task specifications in this study relied on one-hot vector labels, which limit the diversity and scalability of commands. Future work will explore more flexible instruction modalities, including natural language, to enhance generalization and usability in real-world applications.

References

[1] R. Jangir, N. Hansen, S. Ghosal, M. Jain, and X. Wang, "Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3046–3053, 2022.

[2] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 3406–3413.

[3] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3758–3765.

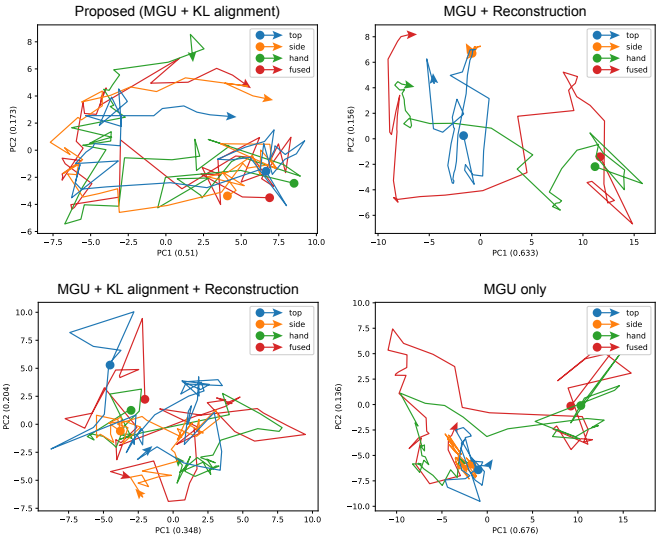

Fig. 5. Visualization of the latent representations for each viewpoint (inputs to the MGU): z_t^{top} (blue), z_t^{side} (orange), and z_t^{hand} (green), as well as the fused representation (output from the MGU): z_t (red), during the lid-opening task. The results are shown across different models: the proposed model, the model with only reconstruction learning, the model with both KL-based alignment and reconstruction learning, and the model without either. Principal component analysis was performed and the first and second principal components were visualized. The variance ratios are indicated on the axes. Note that the representations from the first five and last ten time steps, which have minimal impact on task execution, were excluded from the visualization.

[4] S. Dasari and A. Gupta, "Transformers for one-shot visual imitation," in *Conference on Robot Learning*. PMLR, 2021, pp. 2071–2084.

[5] A. Ganapathi, P. Florence, J. Varley, K. Burns, K. Goldberg, and A. Zeng, "Implicit kinematic policies: Unifying joint and cartesian action spaces in end-to-end robot learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2656–2662.

[6] A. Sharma, A. M. Ahmed, R. Ahmad, and C. Finn, "Self-improving robots: End-to-end autonomous visuomotor reinforcement learning," *arXiv preprint arXiv:2303.01488*, 2023.

[7] A. Zhou, M. J. Kim, L. Wang, P. Florence, and C. Finn, "Nerf in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 907–17 917.

[8] R. Cheng, A. Agarwal, and K. Fragkiadaki, "Reinforcement learning of active vision for manipulating objects under occlusions," in *Conference on Robot Learning*. PMLR, 2018, pp. 422–431.

[9] T. Van de Maele, T. Verbelen, O. Çatal, C. De Boom, and B. Dhoedt, "Active vision for robot manipulators using the free energy principle," *Frontiers in neurorobotics*, vol. 15, p. 642780, 2021.

[10] S. Jang, H. Jeong, and H. Yang, "Murm: utilization of multi-views for goal-conditioned reinforcement learning in robotic manipulation," *Robotics*, vol. 12, no. 4, p. 119, 2023.

[11] J. Yang, D. Sadigh, and C. Finn, "Polybot: Training one policy across robots while embracing variability," *arXiv preprint arXiv:2307.03719*, 2023.

[12] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Multi-view fusion for multi-level robotic scene understanding," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6817–6824.

[13] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[14] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[15] J. Arevalo, T. Solorio, M. Montes-y Gómez, and F. A. González, "Gated multimodal units for information fusion," *arXiv preprint arXiv:1702.01992*, 2017.

[16] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1134–1141.

[17] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *International conference on machine learning*. PMLR, 2020, pp. 5639–5650.

[18] A. Zhan, R. Zhao, L. Pinto, P. Abbeel, and M. Laskin, "Learning visual robotic control efficiently with contrastive pre-training and data augmentation," 12 2020. [Online]. Available: http://arxiv.org/abs/2012.07975

[19] A. Correia and L. A. Alexandre, "Contrastive learning from demonstrations," *arXiv preprint arXiv:2201.12813*, 2022.

[20] K. Ramachandruni, M. Babu, A. Majumder, S. Dutta, and S. Kumar, "Attentive task-net: Self supervised task-attention network for imitation learning using video demonstration," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4760–4766.

[21] B. Chen, P. Abbeel, and D. Pathak, "Unsupervised learning of visual 3d keypoints for control," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1539–1549.

[22] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[23] K. Hsu, M. J. Kim, R. Rafailov, J. Wu, and C. Finn, "Vision-based manipulators need to also see from their hands," *arXiv preprint arXiv:2203.12677*, 2022.

[24] I. Akinola, J. Varley, and D. Kalashnikov, "Learning precise 3d manipulation from multiple uncalibrated cameras." [IEEE], 2020.

[25] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[26] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[27] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[28] L. Lai, A. Z. Huang, and S. J. Gershman, "Action chunking as policy compression," 2022.

[29] G. Zhang, M. Gao, Q. Li, W. Zhai, and G. Jeon, "Multi-modal generative deepfake detection via visual-language pretraining with gate fusion for cognitive computation," *Cognitive Computation*, pp. 1–14, 2024.

[30] T. Anzai and K. Takahashi, "Deep gated multi-modal learning: In-hand object pose changes estimation using tactile and image data," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9361–9368.

[31] H. Ichiwara, H. Ito, K. Yamamoto, H. Mori, and T. Ogata, "Modality attention for prediction-based robot motion generation: Improving interpretability and robustness of using multi-modality," *IEEE Robotics and Automation Letters*, 2023.

[32] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3d neural scene representations for visuomotor control," in *Conference on Robot Learning*. PMLR, 2022, pp. 112–123.

[33] A. Kinose, M. Okada, R. Okumura, and T. Taniguchi, "Multi-view dreaming: Multi-view world model with contrastive learning," *Advanced Robotics*, vol. 37, no. 19, pp. 1212–1220, 2023.

[34] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[35] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*. PMLR, 2019, pp. 2555–2565.

[36] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," *arXiv preprint arXiv:2010.02193*, 2020.

[37] T. Taniguchi, S. Murata, M. Suzuki, D. Ognibene, P. Lanillos, E. Ugur, L. Jamone, T. Nakamura, A. Ciria, B. Lara, and G. Pezzulo, "World models and predictive coding for cognitive and developmental robotics: frontiers and challenges," *Advanced Robotics*, vol. 37, no. 13, pp. 780–806, 2023.

[38] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.