

469 **7 Supplementary**

470 **7.1 Assumptions and Future Research**

471 **7.1.1 Assumptions**

472 During the data collection phase, ActAIM2 operates under the assumption that: 1) the manipulations
473 are straightforward enough to be captured using a limited set of action primitives such as grasping,
474 pushing, or pulling; 2) an interaction mode is identified upon observing significant visual changes;
475 3) the interaction modes can be categorized into a few distinct types. A more detailed discussion of
476 these assumptions is provided below.

477 **Simple Action Space** – We employ a scripted, self-supervised method to collect actions that en-
478 compass diverse interaction modes. The action space is sufficiently simple, focusing primarily on
479 heuristic grasping and random actions. For more complex tasks, such as hammering, washing dishes,
480 or cooking, our current method fails to collect adequate data. Addressing these more intricate tasks
481 would require a more comprehensive and extensive dataset.

482 **Significant Visual Change** – Our data collection is entirely self-supervised, devoid of any expert data
483 or privileged information. We define an interaction as successful if it results in a significant visual
484 alteration to the targeted objects. This approach is effective for articulated objects in our studies, such
485 as doors, windows, or tables, which typically remain stationary except for their movable components.
486 However, challenges arise with objects like tools (e.g., hammers, cups, knives), where it is difficult to
487 discern visual changes either in the tools themselves or the targeted objects (e.g., nails, cup holders,
488 or deformable objects). Especially in tasks requiring repetitive actions, like continuously striking a
489 nail or repeatedly wiping dishes, a more nuanced and generalized method is necessary to determine if
490 meaningful interactions are occurring.

491 **Discrete Interaction Modes** – Articulated objects, by design, often have limited manipulation options.
492 However, when dealing with other objects such as tools, the number of potential interaction modes
493 significantly increases. The functionality of these objects can be diverse; for example, a hammer
494 might be used not only for hammering but also for hooking or reaching. Even the act of grasping
495 these objects presents countless variations, complicating the task of clustering them into discrete
496 modes.

497 **7.1.2 Future Research**

498 Based on the assumptions discussed earlier, we have identified two primary avenues for extending
499 our current research: long-horizon planning tasks and enhancing tool manipulation strategies.

500 **Long-horizon Planning Tasks** – Leveraging the discrete representation of interaction modes provided
501 by ActAIM2, we propose its application to long-horizon planning tasks. Examples of such tasks
502 include sequentially opening a table drawer, locating and opening a box within the drawer, and
503 finally pressing a button inside the box. These tasks illustrate the potential of ActAIM2 to serve as a
504 foundational prior, streamlining the process to discrete searches within complex sequences. To ensure
505 the robustness of our approach, it is crucial that the model accurately predicts all feasible interaction
506 modes based on the given scenario.

507 **Extension to Tool Manipulation Tasks** – Another direction for expansion involves applying our work
508 to tool manipulation. Here, defining the interaction modes for various tools will be pivotal. A robust
509 dataset specifically tailored for tool manipulation is essential to support this endeavour. Additionally,
510 a more sophisticated scene descriptor is required to effectively determine which objects to manipulate
511 and which to designate as targets. This development would facilitate more nuanced and effective tool
512 interactions in automated systems.

513 **7.2 Dataset Generation**

514 **7.2.1 Iterative Data Collection Method**

515 When collecting data, we employ a strategy of random sampling, subsequently filtering successful
 516 actions as determined by our vision model without resorting to any privileged information. Drawing
 517 inspiration from [43, 51], we delineate the task of manipulating articulated objects into four fundamen-
 518 tal poses: initiation, reaching, grasping, and manipulating. Throughout these stages, we capture the
 519 robot's key action poses $a_i = (\mathbf{p}, \mathbf{R}, \mathbf{q})_i$ and RGBD observations O_i from a configuration of five cam-
 520 eras encircling the articulated object. Upon collecting the trajectory $T_j = \{(a_i, O_i) | i = 0, 1, 2, 3\}_j$,
 521 we also archive the initial and final observations, O_j^{init} and O_j^{final} , respectively, captured from the
 522 multi-view cameras with the robot occluded, to facilitate manipulation success evaluation.

523 We introduced our method of identifying successful interacted trajectories, which can be purely from
 524 vision data, specifically the initial and final observation. For each trajectory, characterized by the
 525 initial observation O_j^{init} and final observation O_j^{final} , we utilize a pre-trained image encoder \mathcal{E}_O to
 526 transform the image observations into a latent vector v . The task embedding z_j for each trajectory T_j
 527 is defined as follows:

$$z_j = v_j^{init} - v_j^{final} = \mathcal{E}_O(O_j^{init}) - \mathcal{E}_O(O_j^{final}) \quad (9)$$

528 In our implementation, we employ a pre-trained VGG-19 network [52], without the final fully
 529 connected layers, to serve as our image encoder \mathcal{E}_O . To determine the success or failure of a
 530 manipulation, we introduce a threshold \bar{z} , defining a trajectory T_j as successful if $z_j > \bar{z}$. It is
 531 important to note that this process does not rely on any privileged information. To illustrate the
 532 validity of our method, we define the trajectory's success as a 30% change in the ground-truth DoF
 533 value. The efficacy of this criterion is validated against the ground-truth DoF values, demonstrating a
 534 97.4% accuracy rate across our training and testing dataset. The collected trajectories must exhibit
 535 the diversity of interaction modes of the articulated objects. Thus, we employ three distinct methods
 536 of action sampling, as outlined below. The final dataset is a composite of these three methods.

537 **1. Random Sampling** – We generate play data for manipulation without prior interaction. First,
 538 we select an interaction point $p_1 \in \mathbb{R}^3$ on the articulated object, ensuring it lies within the robot's
 539 workspace. Subsequently, we sample a uniformly random manipulation rotation $\mathbf{R}_0 \in SO(3)$ and
 540 a manipulation position p_2 within the valid area, applying filters to exclude any configurations that
 541 would result in a collision. The robot's initial position p_0 is also determined through random sampling,
 542 which is a specified distance from the interaction point p_1 , ensuring a feasible starting position for the
 543 manipulation task. Based on the previous sampling, we define the randomly sampled action sequence
 544 as $\{(p_0, \mathbf{R}_0, 0), (p_1, \mathbf{R}_0, 0), (p_1, \mathbf{R}_0, 1), (p_2, \mathbf{R}_0, 1)\}$.

545 **2. Heuristic Grasping Sampling** – Heuristic grasping sampling is employed to select interaction
 546 points on the articulated object to enhance the precision of grasping actions. Utilizing the RGB-D
 547 observations, we crop the articulated object and transform it into an RGB point cloud, which under-
 548 goes preprocessing with DBSCAN clustering [53], aimed at identifying segments with significant
 549 geometric features, such as handles or buttons. After clustering, each segment is analyzed by a
 550 pre-trained GraspNet model [54] to generate a set of potential grasps. From this set, grasps with
 551 the highest scores are selected, with the grasp point designated as the interaction point and the grasp
 552 orientation as the gripper rotation for the trajectory. The initial and manipulation poses are determined
 553 using the previously described random sampling approach. This heuristic approach to grasping not
 554 only bolsters the stability of grasp actions but also enriches the dataset with a higher proportion of
 555 complex interaction modes, such as "grasp to open", enhancing the dataset's diversity and utility for
 556 training models to manipulate articulated objects in 'hard' interaction scenarios.

557 **3. GMM-based Adaptive Sampling** – To foster a wide array of interaction modes within our
 558 dataset, we implement GMM-based adaptive sampling inspired by the methodology outlined in [12].
 559 Following the acquisition of M trajectory datasets $\{T_j | j = 1, 2, \dots, M\}$ through random and heuristic
 560 grasping sampling from previous interactions, we compute the task embeddings $\{z_j | j = 1, 2, \dots, M\}$
 561 based on Equation 9. A Gaussian Mixture Model (GMM) prior is constructed from these task

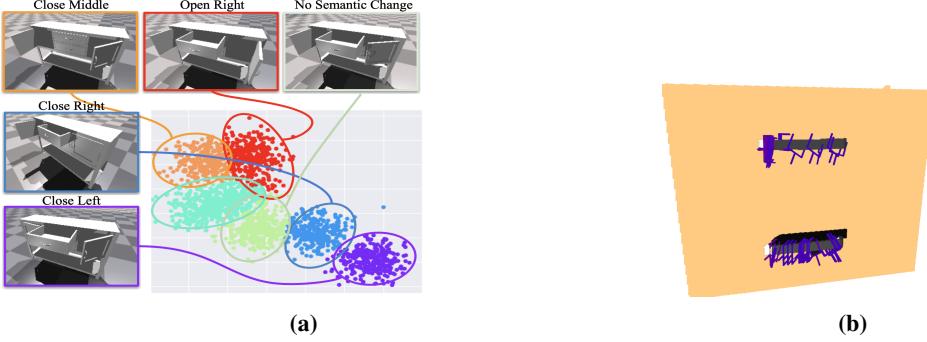


Figure 5: (a) GMM clustering adaptive sampling: his figure illustrates the visualization of using GMM to represent different interaction modes.

(b) Visualization of Heuristic Grasping: We illustrate the proposed grasping using our predefined heuristic with ContactGraspNet [55].

embeddings, denoted as $\mathbb{P}(z|\theta) = \sum_{k=1}^K \kappa_k p(a|\theta_k)$, where θ_k represents the parameters of each Gaussian component within the mixture. The choice of K , the number of clusters, is a hyper-parameter that reflects the presumed number of interaction modes inherent to the object.

Subsequently, we cluster the task embeddings z_j , assigning a unique cluster label to each corresponding trajectory. We found that the task embeddings from different trajectories grouped within the same cluster indicate a similar interaction mode, as they share proximate visual characteristics from initial and final observation. Upon clustering, a new GMM is formulated for each cluster, based on the action sequences, represented as $\mathbb{P}_k(a|\phi) = \sum_{l=1}^L \beta_l p_k(a|\phi_l)$. We then aim to sample an equal number of actions from each cluster, ensuring that the representation of actions—and, by extension, interaction modes—within the dataset are as diverse as possible, thus facilitating a comprehensive exploration of the articulated object’s potential interactions.

Utilizing these sampling methodologies, we concurrently collect data across all articulated objects within our dataset, culminating in a dataset denoted as:

$$D = \{T_j\}_{\text{random}} \cup \{T_j\}_{\text{grasp}} \cup \{T_j\}_{\text{GMM}} \quad (10)$$

$$= \{(a_i, O_i)_j\}_{\text{random}} \cup \{(a_i, O_i)_j\}_{\text{grasp}} \cup \{(a_i, O_i)_j\}_{\text{GMM}} \quad (11)$$

$$= \{(O_i, a_i)\}_{\text{random} \cup \text{grasp} \cup \text{GMM}} \quad (12)$$

After data collection, we enrich each trajectory within our dataset by associating the respective task embedding with the data tuple (O, a) , thereby forming atomic training data instances represented as $(O, a, \epsilon)_j$.

7.2.2 Data Collection Algorithm

The dataset we developed for training purposes is available on our official website. Our dataset was constructed through a combination of random sampling, heuristic grasp sampling, and Gaussian Mixture Model (GMM)-based adaptive sampling, featuring the Franka Emika robot engaging with various articulated objects across multiple interaction modes. It encompasses categories such as faucets, tables, storage furniture, doors, refrigerators, and switches, with 8 unique instances per category. For each instance, we collected 150 trajectories, ensuring comprehensive coverage of the objects’ interaction modes. Objects were scaled to realistic size and initialized in a ‘half-open’ state, denoting a median value for each degree of freedom (DoF). The data collection methodology is detailed in Algorithm 1.

7.3 Model Architecture and Implementation Details

This section outlines the detailed implementation of the model architecture, encompassing both the mode selector and the action predictor components.

Algorithm 1 Data Collection Algorithm

Require: Initial observation O^i , Number of GMM component K , hyper-paramter M for GMM in each cluster

Ensure: All sampled trajectories are filtered successful by evaluating $\epsilon > \bar{\epsilon}$

```

 $D \leftarrow \emptyset$                                 ▷ Set the initial dataset to be empty
while  $D$  not have enough data do
     $D_r = \{(a, o)_i\} \sim \text{RandomSampling}$           ▷ Random Sampling
     $G = \{g_i\} \sim \text{GraspNet}(O^i)$                   ▷ Sample Grasp using GraspNet
     $D_g = \{(a, o)_i\} \sim \text{GenerateTraj}(G)$         ▷ Generate trajectory based on grasp
     $D \leftarrow D \cup D_r \cup D_g$ 
     $\epsilon_i \sim D$                                          ▷ Compute task embedding in current  $D$ 
    Cluster  $\epsilon_i$  with GMM, assign cluster label on each trajectory
     $\{D_j | j = 1, \dots, K\} \leftarrow D$ 
     $D_{GMM} \leftarrow \emptyset$ 
    for  $j$  in range  $K$  do
        Extract  $D_j$  in  $D$  based on cluster label
         $p(D_j | \pi, \mu, \Sigma) = \prod_{n=1}^N \left( \sum_{m=1}^M \pi_m \mathcal{N}(\mathbf{x}_n | \mu_m, \Sigma_m) \right)$       ▷ fit GMM
         $\hat{D}_j \leftarrow \{(a, o)_i\} \sim p(D_j | \pi, \mu, \Sigma)$                                      ▷ Sample action from GMM
         $D_{GMM} \leftarrow D_{GMM} \cup \hat{D}_j$ 
     $D \leftarrow D \cup D_{GMM}$ 

```

591 **7.3.1 Mode Selector Architecture and Implementation Detail**

592 This section revisits the stochastic variables' definitions and distributions, as previously emphasized.
593 The distributions of the model parameters are formalized as follows:

$$p(c) = \text{Multi}(\pi) \quad (13)$$

$$p(y) = \mathcal{N}(0, \mathbf{I}) \quad (14)$$

$$p_{\xi, \beta}(\epsilon, x, y, c | O^i) = p(y)p(c)p_{\xi}(x|y, c, O^i)p_{\beta}(\epsilon|x, O^i) \quad (15)$$

$$p_{\xi}(x|y, c, O^i) = \prod_{k=1}^K \mathcal{N}(\mu_{c_k}(y, O^i), \Sigma_{c_k}(y, O^i)) \quad (16)$$

$$p_{\beta}(\epsilon|x, O^i) = \mathcal{N}(\mu_{\beta}(x, O^i), \Sigma_{\beta}(x, O^i)) \quad (17)$$

594 Here, $\mu_{c_k}, \Sigma_{c_k}, \mu_{\beta}, \Sigma_{\beta}$ are the model parameters to be optimized. Furthermore, we delineate the
595 generative model and compute the inference at test time by defining the posterior as follows:

$$q(x, y, c | \epsilon, O^i) = \prod_i q_{\psi_x}(x | \epsilon, O^i)q_{\psi_y}(y | \epsilon, O^i)q_{\psi_c}(c | x, y, O^i) \quad (18)$$

596 This necessitates the computation of three additional network parameters: $q_{\psi_x}, q_{\psi_y}, q_{\psi_c}$. We then
597 elaborate on deriving the posterior $q_{\psi_c}(c | x, y, O^i)$ for categorical variables c , employing the Gumbel
598 Softmax for the representation of categorical distributions.

599 Notice that c is a categorical parameter that $c \sim \text{Multi}(\pi)$. We defined that $c \in \mathcal{C} = \{c_1, c_2, \dots, c_k\}$
600 and the each class probability is described as $\{\pi_1, \pi_2, \dots, \pi_k\}$. We use the Gumbel Softax trick which
601 provides a simple and efficient way to draw samples c from a categorical distribution with class
602 probabilities $\{\pi_1, \pi_2, \dots, \pi_k\}$. The following form represents the categorical c as,

$$c = \text{one-hot}(\text{argmax}_i[g_i + \log \pi_i]) \quad (19)$$

603 where $\{g_1, g_2, \dots, g_k\}$ are i.i.d samples drawn from $\text{Gumbel}(0,1)$. Assuming that categorical samples
604 c are encoded as k -dimensional one-hot vectors ω lying on the corners of the $(k - 1)$ -dimensional
605 simplex Δ^{k-1} We use the softmax function as a continuous, differentiable approximation to arg max ,
606 and generate k -dimensional sample vectors $\omega \in \Delta^{k-1}$. We defined ω as

$$\omega_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{i=1}^k \exp((\log(\pi_i) + g_i)/\tau)} \quad (20)$$

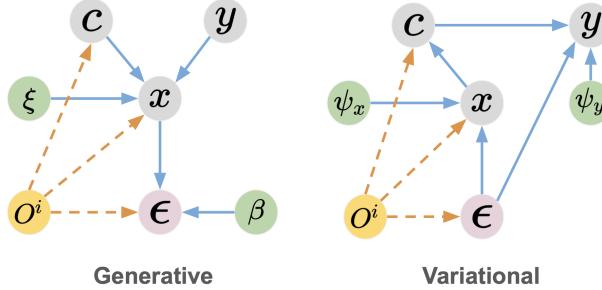


Figure 6: The graphical representations elucidate the Conditional Gaussian Mixture Variational Autoencoder (CGMVAE) framework, showcasing two distinct models: the generative model on the left and the variational family on the right. These graphical models serve to visually communicate the structural and functional relationships between variables within the CGMVAE, illustrating the data generation process and the approximation strategy employed by the variational family to infer latent variable distributions.

607 Where τ is the temperature as the hyperparameter. Therefore, we define the density of the Gumbel-
 608 Softmax distribution as,

$$p(c) = p_{\pi, \tau}(\omega_1, \dots, \omega_k) = \Gamma(k)\tau^{k-1} \left(\sum_{i=1}^k \frac{\pi_i}{\omega_i^\tau} \right)^{-k} \prod_{i=1}^k \frac{\pi_i}{\omega_i^\tau} \quad (21)$$

609 Now, given the representation of the categorical distribution of c from [Equation 21](#), we derive how
 610 we compute the posterior q_{ψ_c} for c . We consider the posterior $q_{\psi_c}(c = c_j | x, y, O^i)$ given $c = c_j$,

$$q_{\psi_c}(c = c_j | x, y, O^i) = \frac{p(c = c_j)p(x|c = c_j, y, O^i)}{\sum_{l=1}^k p(c = c_l)p(x|c = c_l, y, O^i)} \quad (22)$$

$$= \frac{\pi_j p(x|c = c_j, y, O^i)}{\sum_{l=1}^k \pi_l p(x|c = c_l, y, O^i)} \quad (23)$$

611 Therefore, we derive the posterior q_{ψ_c} directly and leave 2 posterior network q_{ψ_x}, q_{ψ_y} to be trained.
 612 Based on the following discussion, we draw the generative model and variational model view as
 613 graphical models in the [Figure 6](#).

614 In the implementation detail, we write parameters $p_\beta = (\mu_\beta, \Sigma_\beta)$ and $p_\xi = (\mu_{c_k}, \Sigma_{c_k})$ to generate
 615 a Gaussian distribution with each representing the mean and variance. We implement the network
 616 $\mu_{c_k}, \Sigma_{c_k}, \psi_x, \psi_y$ with a multi-layer ResNet and implement the network μ_β, Σ_β as a multi-view
 617 transformer since both O^i and ϵ represent multi-view information with the same number on the
 618 channel as the correspondent view number. We show our model μ_β, Σ_β architecture in [Figure 7](#).

619 7.3.2 Mode Selector Training and Inference

620 We illustrate the functionality and application of our mode selector through two distinct plots,
 621 highlighting both the training process and the inference mechanism for task embedding generation.
 622 Figure [Figure 9a](#) depicts the model's operation during training, where it processes the conditional
 623 variable O^i along with the ground truth data ϵ , to accurately reconstruct the task embedding.
 624 Conversely, Figure [Figure 9b](#) demonstrates the inference stage, where the model, requiring only the
 625 initial observation O^i and a discretely sampled cluster (employing an 8-cluster configuration for
 626 implementation), successfully generates the corresponding task embedding ϵ .

627 7.3.3 Action Predictor

628 We provide the architecture of the action predictor which is a joint transformer that takes in task
 629 embedding ϵ and novel view as input. The detailed implementation is shown at [Figure 8](#).

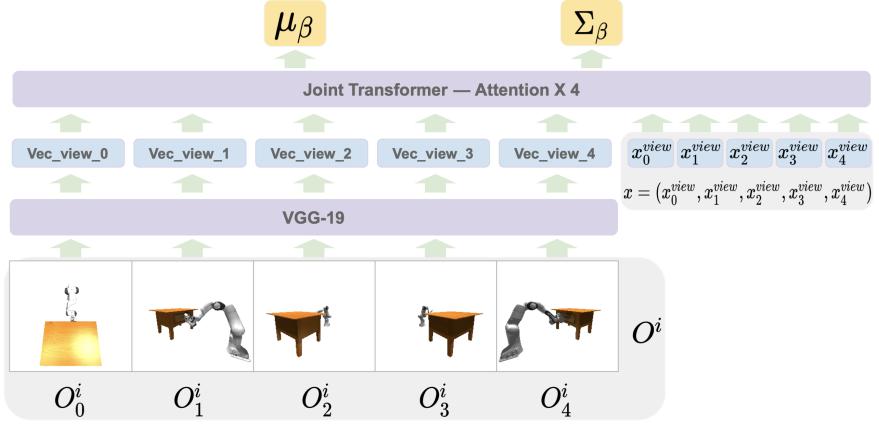


Figure 7: Mode Selector Decoder Architecture: The depicted architecture highlights the functionality of the mode selector decoder, which is designed to process two primary inputs: multi-view RGBD images $O^i = (O_0^i, O_1^i, O_2^i, O_3^i, O_4^i)$, and the Mixture of Gaussian (GMM) variable x . It is important to note that x can be represented as a multi-view feature vector, with our encoding approach preserving the separation of multi-view channels. Initially, the multi-view RGBD images are passed through a pre-trained VGG-19 image encoder to extract feature vectors for each view. Subsequently, these feature vectors, along with the GMM variable x , are inputted into a joint transformer. This transformer, featuring four attention layers, is tasked with producing the means and variances associated with the reconstructed task embedding ϵ .

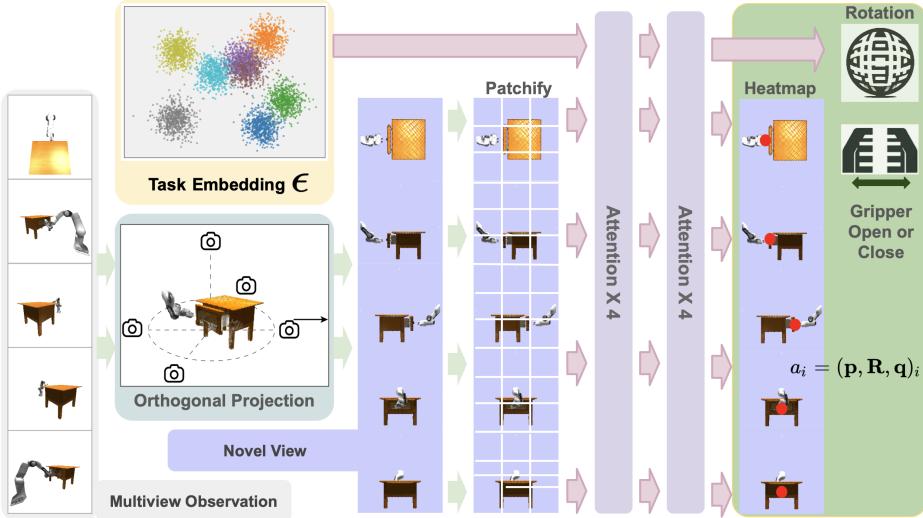
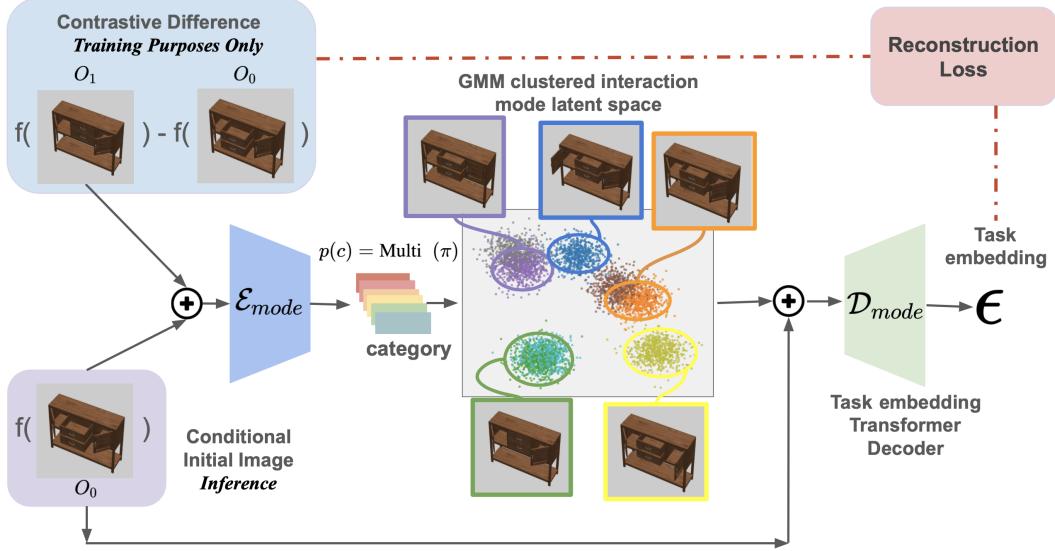


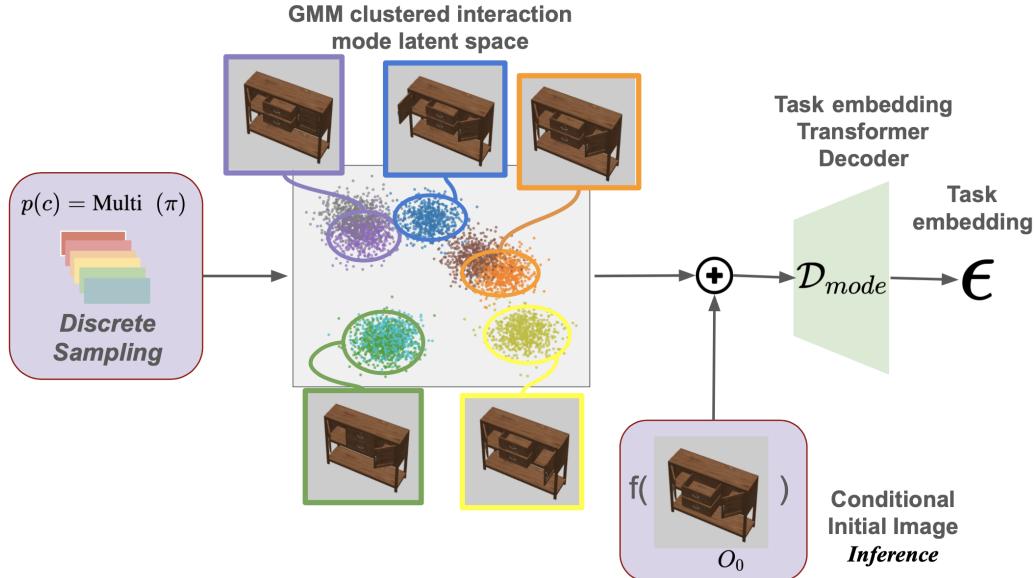
Figure 8: Action Predictor Architecture: This model integrates multi-view observations directly as input, sourced from predefined cameras within the scene. The process begins with the extraction of five RGBD images, which are subsequently transformed into RGB point clouds. These are then subject to orthogonal projection to generate five novel view images. Subsequently, these novel views are partitioned into smaller patches and fed into a joint transformer. This transformer, characterized by four attention layers, integrates the sampled task embedding derived from a Mixture of Gaussian distribution. The architecture of the joint transformer encompasses eight attention layers, culminating in the production of a heatmap. This heatmap delineates the action’s translation, the discretized rotation, and a binary variable indicating the gripper’s state—open or closed.

630 7.4 More Qualitative Results

631 We supplement our presentation with additional qualitative results, further elucidating the model’s
 632 proficiency in learning the disentanglement of interaction modes. Initially, we demonstrate the
 633 efficacy of the mode selector through a t-SNE plot. This choice of visualization is motivated by our
 634 methodology of training the mode selector and action predictor independently, allowing for a focused
 635 examination of the mode selector’s performance.



(a) Training Process of the Mode Selector: This figure illustrates the training procedure of the mode selector, mirroring the approach of a conditional generative model. It highlights the contrastive analysis between the initial and final observations—the latter serving as the ground truth for task embedding—to delineate generated data against the backdrop of encoded initial images as the conditional variable. The process involves inputting both the generated task embedding data and the conditional variable into a 4-layer Residual network-based mode encoder, which then predicts the categorical variable c . Following the Gaussian Mixture Variational Autoencoder (GMVAE) methodology, the Gaussian Mixture Model (GMM) variable x is computed and introduced alongside the conditional variable to the task embedding transformer decoder. This model is tasked with predicting the reconstructed task embedding, sampled from the Gaussian distribution as outlined in the architecture of the mode selector decoder, and calculating the reconstruction loss against the input ground truth data.



(b) Inference Process: In the inference phase, the agent discretely samples a cluster from the trained Gaussian Mixture Variational Autoencoder (GMVAE) model to calculate the Mixture of Gaussian variable x . This variable x , in conjunction with the conditional variable (initial image observation), is then inputted into the mode selector transformer decoder. The objective is to reconstruct the task embedding for inference, effectively translating the conditional information and sampled cluster into actionable embeddings.

636 Subsequently, we extend our qualitative analysis with figures akin to those presented in the main
637 paper, offering a comprehensive view of the model's capabilities. These additional figures serve to
638 reinforce the insights gained from the initial results, showcasing the model's nuanced understanding
639 of interaction modes through the distinct visual representations of the data.

640 **7.4.1 Mode selector TSNE plot Figure 14**

Utilizing our pre-trained Conditional Gaussian Mixture Variational Autoencoder (CGMVAE) mode selector, we conduct disentanglement learning visualization on our comprehensive dataset. Specifically, we focus on the "single drawer" object (object ID: 20411), employing the mode selector to delineate the generated clusters and compare them with the ground truth task embeddings. The data for this visualization is derived from our dataset, and we calculate the task embedding ϵ_j for each data point as the difference between the initial and final object states, represented by

$$\epsilon_j = v_j^{init} - v_j^{final} = \mathcal{E}_O(O_j^{init}) - \mathcal{E}_O(O_j^{final})$$

641 .

642 Subsequently, we employ a t-SNE plot to simultaneously visualize the ground truth and generated
643 task embeddings. In this visualization, distinct colors within the ground truth plot indicate data points
644 originating from different interaction modes. Similarly, varied colors in the generated plot correspond
645 to data points arising from disparate clusters within the Mixture of Gaussians model. Through this
646 approach, we demonstrate that:

- 647 1. The ground truth task embeddings ϵ are distinctly clustered based on the interaction modes.
- 648 2. The CGMVAE model effectively generates clusters that categorize data points by their respective
649 categories c .
- 650 3. The reconstructed data closely aligns with the ground truth data points, with the majority of the
651 clustered data encompassed within the respective ground truth clusters.

652 This visualization underscores the efficacy of our generative model mode selector in extracting task
653 embeddings for further application in the action predictor, highlighting the model's capability to
654 discern and categorize interaction modes accurately.

655 **7.4.2 Action Predictor Qualitative Results**

656 We present extensive qualitative results in [Figure 15a](#), [Figure 15b](#), [Figure 16a](#), and [Figure 16b](#),
657 demonstrating the model's ability to predict distinct interaction modes through discrete sampling. For
658 each object, we explore three different clusters, each representing a unique interaction mode. The
659 initial state of the robot and the articulated object is depicted from three perspectives: top-down, front,
660 and side views. The heatmaps, derived from the top view during manipulation steps, highlight the
661 variance in action space corresponding to different sampled interaction modes. Subsequent imagery
662 illustrates the robot's movement within the simulator and the outcome following interaction with the
663 articulated objects. It is important to note that comprehensive **video demonstrations** accompany this
664 document and are accessible on our website, <https://actaim2.github.io/>.

665 **7.4.3 Comparison of ActAIM2 and VQVAE-RVT**

666 Inspired by the Genie [48] approach, we have compared our ActAIM2 with VQVAE-RVT to assess
667 the efficacy of these models in discerning discrete interaction modes in robotic manipulation tasks.
668 Our primary objective was to evaluate the distinction between interaction modes using a simplified
669 scenario, a single-drawer table, which naturally exhibits two distinct interaction modes: opening and
670 closing.

671 In our experiments, we visualized the latent spaces generated by both ActAIM2 and VQVAE-RVT.
672 Particularly for VQVAE-RVT, the latent space visualization involved examining the distribution of
673 eight code vectors. As depicted in Figure 10, these vectors clustered into two categories, which ideally
674 should correspond to the two expected interaction modes of the drawer. This clustering pattern was
675 anticipated and desired as it suggests a clear demarcation between the distinct modes of interaction.

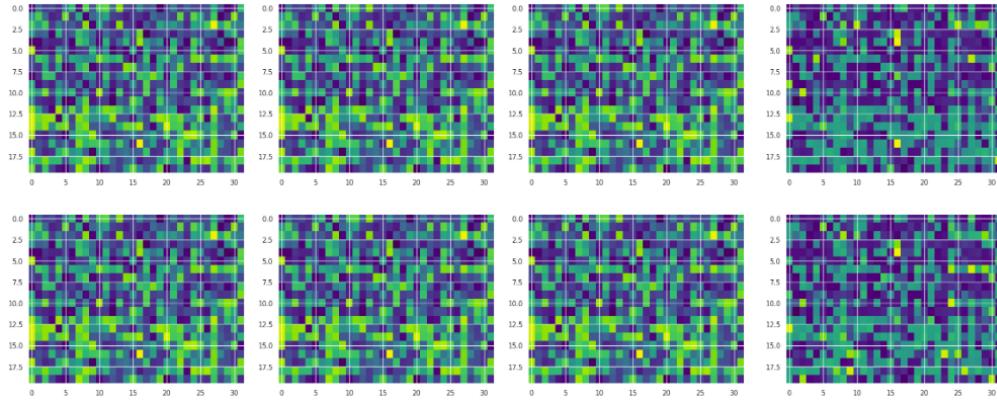


Figure 10: Visualization of Latent Space Clustering in VQVAE-RVT: This figure illustrates the distribution of eight code vectors within the latent space, categorized into two distinct clusters. These clusters are intended to represent the discrete interaction modes of opening and closing a drawer. The spatial arrangement highlights the expected separation of code vectors, symbolizing the potential for mode-specific action mapping in robotic manipulation tasks. Despite this apparent clustering, subsequent heatmaps (see Figure 11) reveal a lack of diversity in the action predictions, undermining the practical utility of this model configuration.

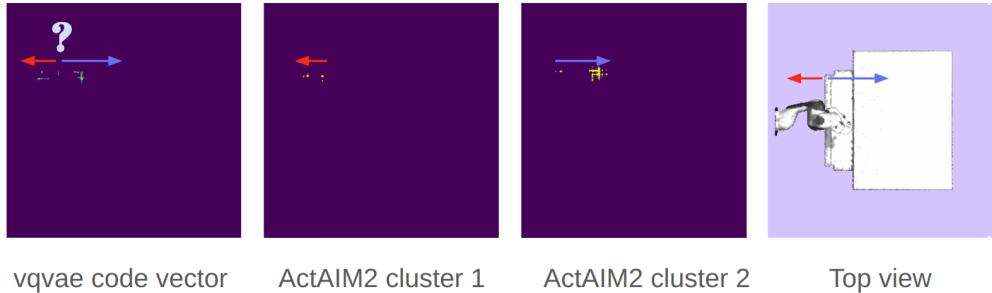


Figure 11: Comparative Visualization of Action Heatmaps and Observational Data From left to right: (1) VQVAE-RVT action heatmap synthesized using all eight code vectors, showing identical outcomes across the board, indicating a failure to differentiate interaction modes. (2) Action heatmap generated by \algoName when sampling from one cluster, demonstrating a specific interaction mode. (3) Action heatmap from \algoName when sampling from a different cluster, showcasing another distinct mode of interaction. (4) Top-view observation of the drawer, correlating with the spatial contexts of the heatmaps, providing a visual reference for the interaction zones mapped by the heatmaps. This series highlights \algoName's capability to discern and represent distinct action strategies through targeted cluster sampling.

676 However, subsequent visualizations raised concerns about the practical efficacy of the VQVAE-RVT
 677 model in our application context. When we explored the heatmaps generated by the VQVAE-RVT
 678 model, we observed a critical limitation: all 8 code vectors produced essentially the same heatmap,
 679 despite their differing positions in the latent space. This heatmap, illustrated in Figure [Y], consistently
 680 depicted all plausible interaction modes for the drawer, regardless of the specific code vector used.
 681 This outcome was in stark contrast to the results from ActAIM2, where distinct heatmaps clearly
 682 indicated specific interaction actions like pushing or pulling, depending on the sampled cluster within
 683 the latent space.

684 These findings led us to conclude that merely replacing the GMVAE component with a VQVAE in
 685 the setup did not achieve the desired disentanglement of interaction modes. The VQVAE-RVT model
 686 failed to map the code vectors to unique, mode-specific interaction strategies, instead converging on a
 687 generalized representation that was not useful for distinguishing between the actionable options of
 688 opening and closing the drawer. Consequently, ActAIM2's ability to discriminate between distinct
 689 interaction modes via cluster-specific sampling proves superior in contexts demanding discrete and
 690 distinguishable action representations.

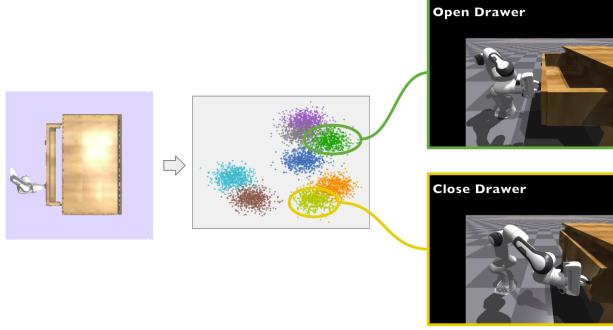


Figure 12: Opening and Closing a Drawer: This figure demonstrates the effective action sequence generated by ActAIM2 for a drawer. The left part of the image shows the drawer being opened, showcasing the robot's approach and grip adjustment. The right part of the image captures the drawer in a fully closed position, illustrating the final state after the action sequence execution.

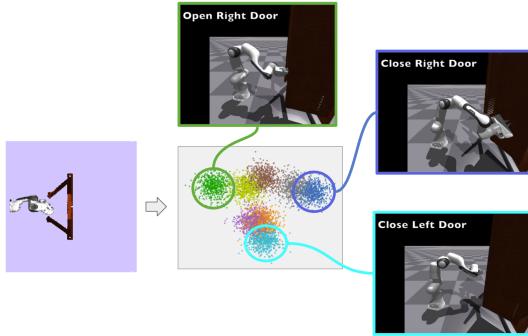


Figure 13: Opening and Closing a Door: This figure illustrates the ActAIM2's manipulation capability with a door. The left image displays the door being opened, highlighting the robot's positioning and the initial interaction phase. The right image shows the door completely closed, detailing the end of the manipulation sequence and the effectiveness of the action predictor.

691 8 Generation of Demonstration Videos

692 To illustrate the practical applications and effectiveness of ActAIM2, we generated demonstration
693 videos by employing its inference mechanism. The process involves several key steps:

- 694 1. **Generative Mode Selection:** Initially, observations are inputted into the generative mode selector
695 of ActAIM2. This component is responsible for reconstructing the task's latent space, which is
696 modeled as a Mixture of Gaussians. This structure enables discrete sampling of clusters, which
697 represent distinct interaction modes that the robotic system can execute.
- 698 2. **Sampling and Action Prediction:** From the reconstructed latent space, we sample the task em-
699 beddings by selecting a cluster within the Gaussian Mixture Model (GMM) and its corresponding
700 Gaussian distribution. This sampled task embedding is then forwarded to the action predictor. The
701 action predictor generates the specific actions needed to interact with the environment effectively.
- 702 3. **Simulation and Recording:** As depicted in Figure 12 and Figure 13, ActAIM2 reconstructs
703 an object-based GMM and samples different task embeddings. Depending on the sampled task
704 embedding, different interactions are reconstructed and executed within a simulator. We recorded
705 the manipulation processes, which are detailed in the video provided in the supplementary files.
706 Each video showcases how ActAIM2 navigates through different interaction scenarios, reflecting
707 the diverse capabilities of the model in real-time applications.

708 This comprehensive demonstration not only validates the functionality of ActAIM2 but also provides
709 a visual understanding of its potential in diverse robotic manipulation tasks. The videos highlight the
710 nuanced interactions achievable through targeted sampling within the model's structured latent space.

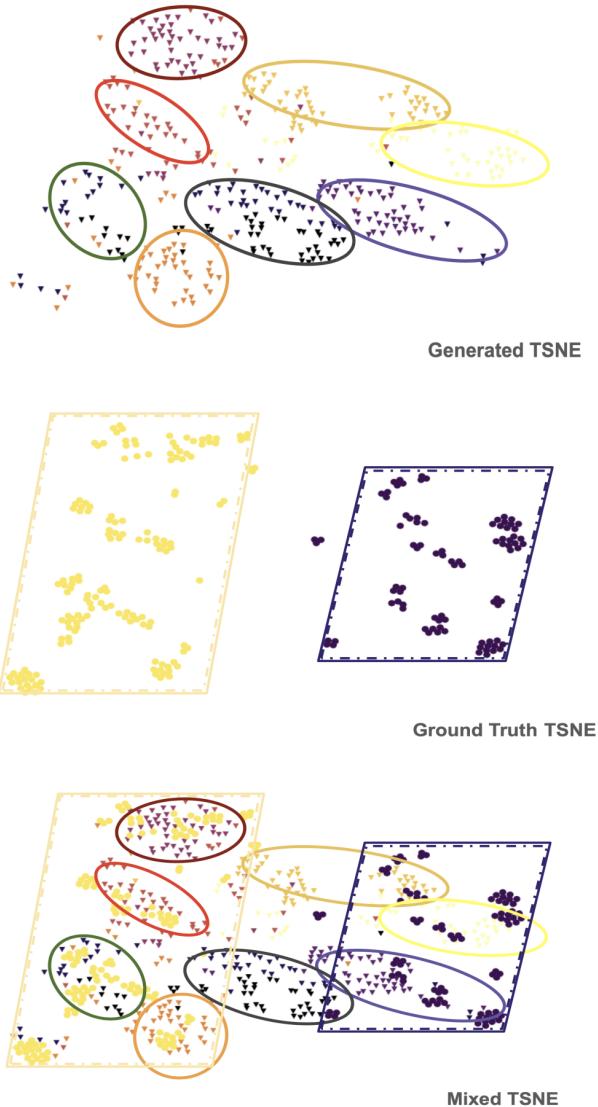
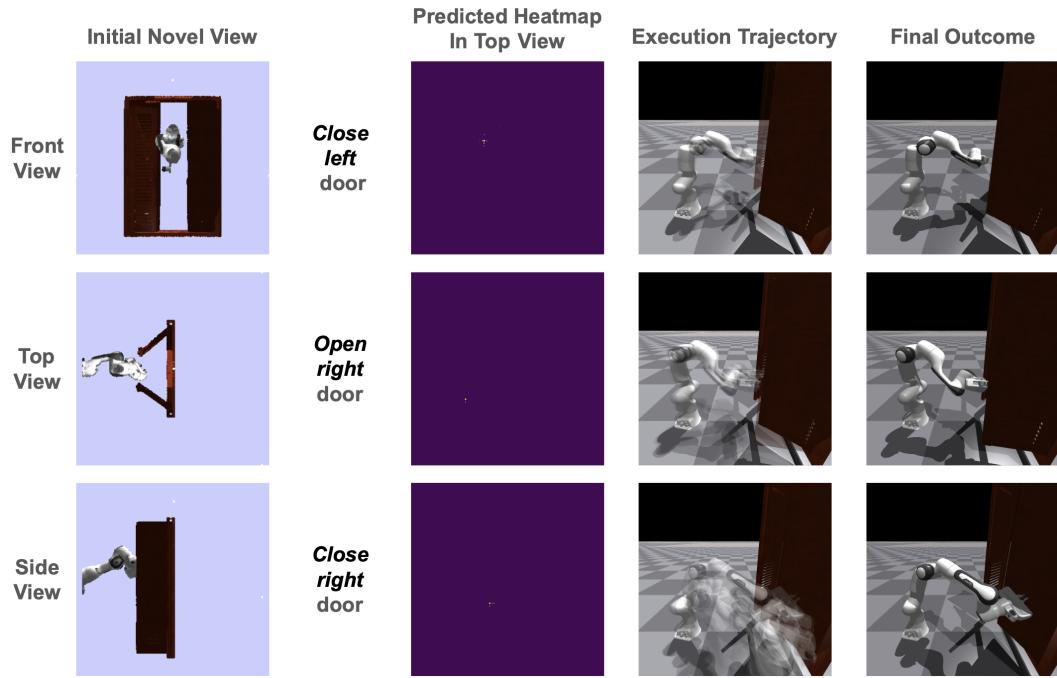
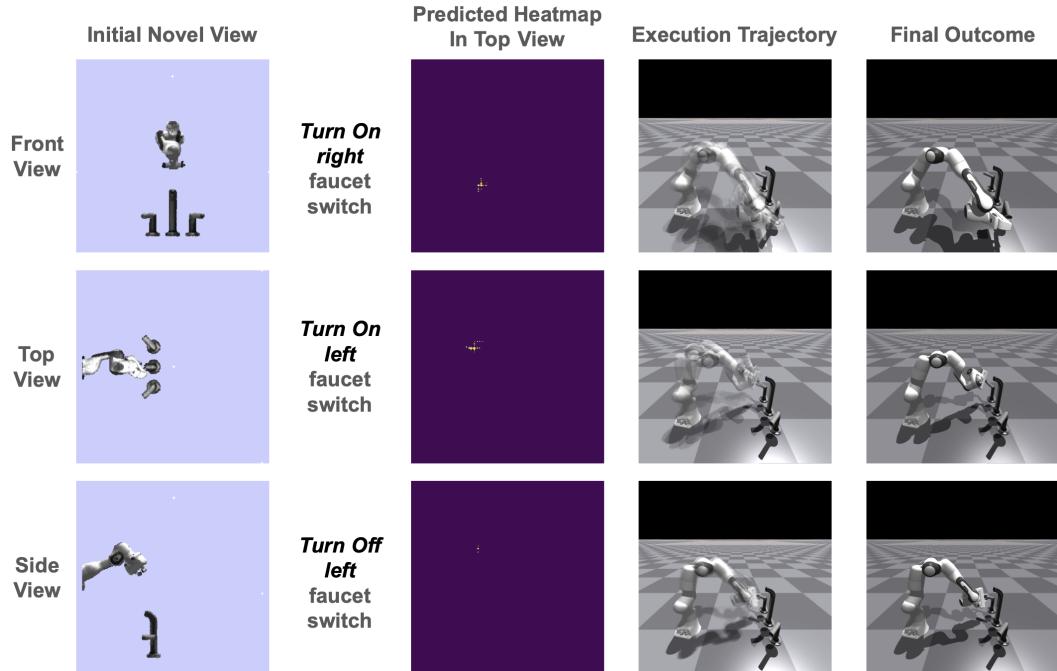


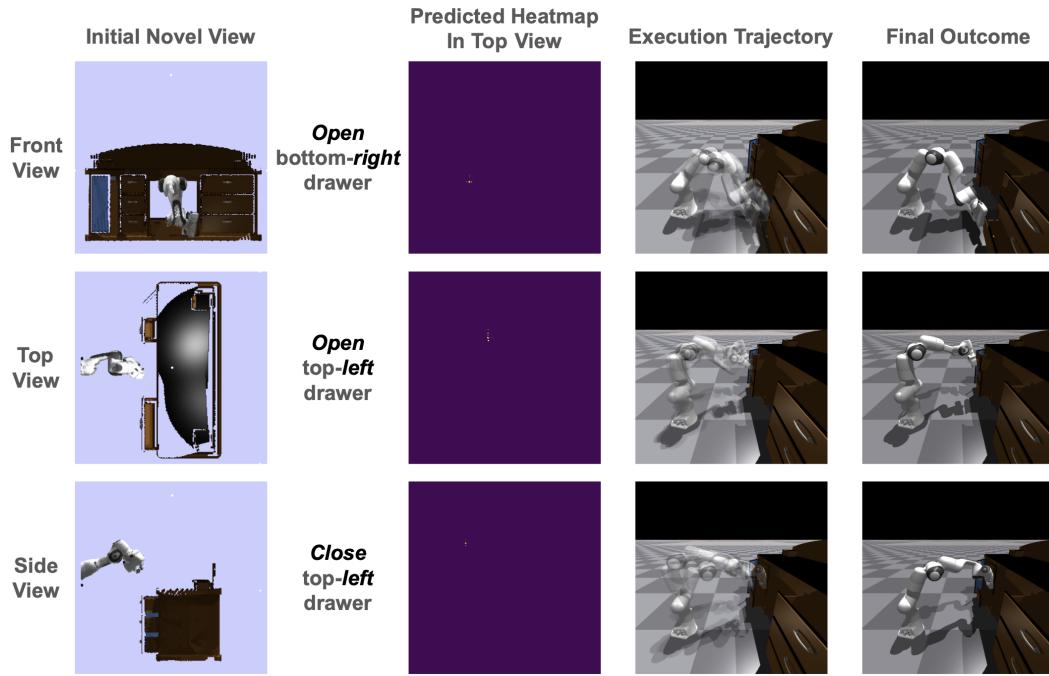
Figure 14: Disentanglement Visualization with CGMVAE: This figure illustrates the efficacy of the Conditional Gaussian Mixture Variational Autoencoder (CGMVAE) in disentangling interaction modes for the "single drawer" object (ID: 20411), using a t-SNE plot for visualization. Task embeddings ϵ_j , defined by the variance between initial and final object states, are visualized in distinct colors to denote various interaction modes and clusters. The sequence of figures demonstrates the CGMVAE's precision in clustering and aligning data points with their respective interaction modes: (1) Generated clusters from the CGMVAE mode selector reveal distinct groupings. (2) Ground truth task embeddings confirm the model's capacity for accurate interaction mode classification. (3) A combined visualization underscores the alignment between generated clusters and ground truth, showcasing the model's ability to consistently categorize tasks within identical interaction modes.



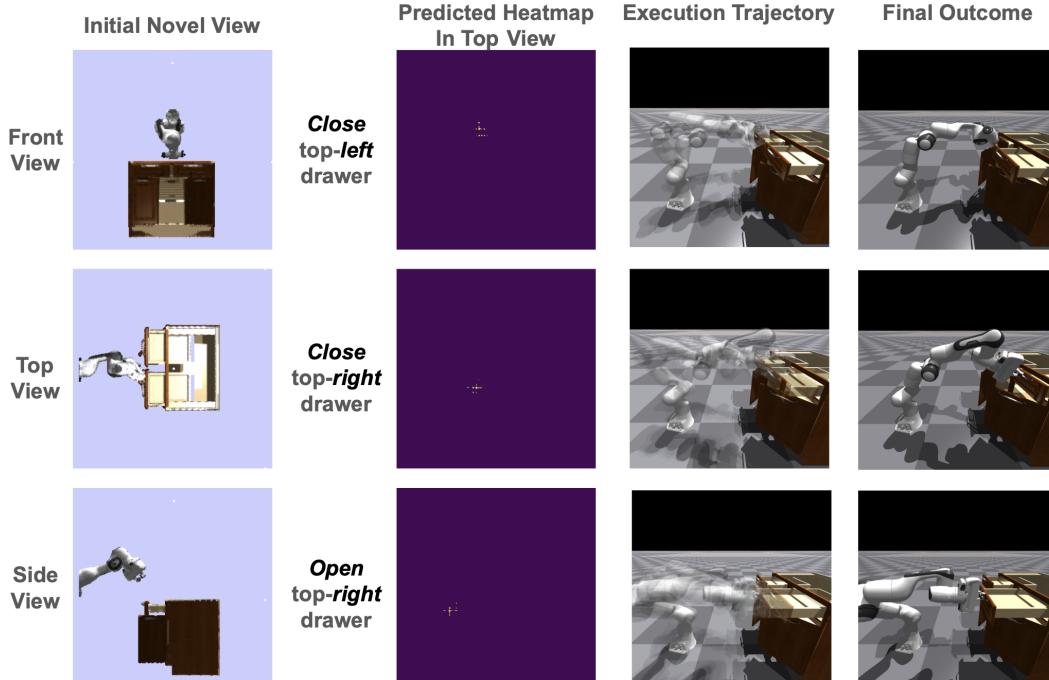
(a) Door, Object ID: 8961



(b) Faucet, Object ID: 154



(a) Table, Object ID: 19898



(b) Table, Object ID: 41083