Bridging Language, Vision and Action: Multimodal VAEs in Robotic Manipulation Tasks

Gabriela Sejnova[1], Michal Vavrecka[1], Karla Stepanova[1]

Abstract— In this work, we focus on unsupervised vision-language-action mapping in the area of robotic manipulation. Recently, multiple approaches employing pre-trained large language and vision models have been proposed for this task. However, they are computationally demanding and require careful fine-tuning of the produced output. A more lightweight alternative would be the implementation of multimodal Variational Autoencoders (VAEs) which can extract the latent features of the data and integrate them into a joint representation, as has been demonstrated mostly on image-image or image-text data for the state-of-the-art models. Here, we explore whether and how multimodal VAEs can be employed in unsupervised robotic manipulation tasks in a simulated environment. Based on the results obtained, we propose a model-invariant training alternative that improves the models' performance in a simulator by up to 55 %. Moreover, we systematically evaluate the challenges raised by individual tasks, such as object or robot position variability, number of distractors, or task length. Our work thus also sheds light on the potential benefits and limitations of using the current multimodal VAEs for unsupervised learning of robotic motion trajectories based on vision and language.

Index Terms—Sensor Fusion, Visual Learning, Semantic Scene Understanding

I. INTRODUCTION

Teaching robots new manipulation tasks by demonstration/kinesthetic guidance is one of the key research areas in robotics. Here we focus on the scenario where the robot also receives a visual observation of the scene and natural language (NL) description of the task and has to fuse this information with kinesthetic teaching to successfully fulfill the goal. While traditional rule-based approaches usually parse the NL input into a formal language for task disambiguation [1], [2], the most recent methods utilize large language (LLM) [3], [4], [5] and large vision models (LVM) [6], [7] pre-trained on internet-scale data, which are additionally fine-tuned, e.g., via prompting methods. However, these models often struggle with repeatability and can be more difficult to control in terms of predictable outputs [8].

One potentially promising approach towards the vision-language-action fusion challenge would be the use of probabilistic generative models such as Multimodal Variational Autoencoders (VAEs). These models have emerged as a powerful extension of the unimodal Variational Autoencoder and are suitable for mapping multiple modalities into a unified joint space, allowing for the generation of one modality from another [9]. However, most studies on multimodal VAEs

[1] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Prague, Czech Republic, [gabriela.sejnova, karla.stepanova]@cvut.cz
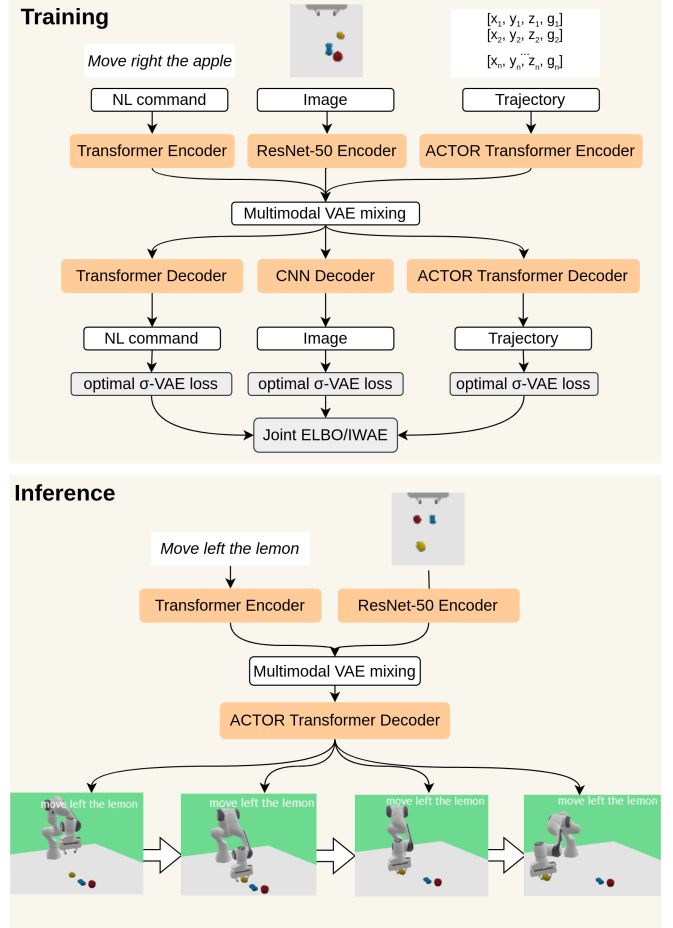

Fig. 1. The proposed architecture scheme for the training procedure (top) and inference (bottom). During training, the language command, image and motion trajectory are fed into individual model encoders. The multimodal mixing method (MVAE, MMVAE or MoPoE) learns the joint posterior distribution. The reconstruction loss for each modality is calculated using the σ-VAE loss and used in the ELBO or DReG (for MMVAE) objective. During inference, the model predicts the whole motion trajectory based on the provided image and command.

have focused mainly on tasks involving image-to-image or image-to-text mappings, leaving a significant gap in their application to robotic actions [10], [11], [12], [13].

In this paper, we explore the potential of the state-of-the-art (SOTA) multimodal VAEs in the context of robotic scenarios, where actions are learned from a combination of action demonstrations, images, and natural language instructions (see Fig. 1). Our primary objective is to understand the challenges and requirements for leveraging VAEs in such

a multimodal setting where each modality has a different level of abstraction and complexity (e.g., high-level language instruction versus low-level end-effector trajectory) and there is no additional supervision such as ground truth of the object positions. Our contributions are the following:

- We employ three selected SOTA multimodal VAEs and adapt the encoder-decoder architecture for mapping natural language inputs, images and whole motion trajectories in a robotic manipulation task (see Fig. 1).
- We propose a model-independent training objective adjustment that is more suitable for such robotic tasks and improves the performance of the implemented models by up to 55 % compared to the standard training objectives.
- We train the models on 36 synthetic robotic datasets with variable complexity in terms of the number of tasks, distractors, position variability and task length. We then evaluate which aspects of the task are the most challenging for the multimodal VAEs.

Our project repository is available on GitHub [1].

II. RELATED WORK

A. Vision-Language-Action mapping in robotics

In recent years, the most popular approach to multimodal learning of robotic actions based on natural language commands incorporates the usage of pre-trained large language models (LLMs) [5], [3], [4] or large pre-trained vision-language models (VLMs) [6], [7]. Such models can parse complex natural language inputs into symbolic task representation and enable semantic reasoning. Moreover, they are often able to generalize to new, unseen data from a small number of training examples. However, they require manual (and often iterative) fine-tuning to condition the model on the specific task, so that the produced results are within the desired range. They also typically cannot be used on a local machine due to the high computational resource requirements.

In our work, we consider instructions for robotic manipulation which can be learned from scratch and bring our attention to the multimodal mapping among modalities with different levels of abstraction (i.e., symbolic instruction, raw image of the scene and low-level motion trajectory). We employ the generative capability of the SOTA multimodal VAEs to learn such mapping without any supervision. Such an approach is computationally lightweight, does not require manual fine-tuning of the input data and produces more predictable results. Furthermore, multimodal VAEs can also generate natural language captions from the input image and trajectories and can thus serve as an action-recognition module at the same time.

B. Multimodal VAEs

Multimodal VAEs enable joint integration and reconstruction of two or more modalities. Various approaches to learning joint representation in multimodal VAEs have

been proposed [14], [10], [11], [15], [13], [16]. The JMVAE model [14] learns the joint multimodal probability through a joint inference network and requires an additional inference network for each subset of modalities. A more scalable solution is the MVAE model [10], where the joint posterior distribution is approximated using the product of experts (PoE). The MMVAE approach [11] uses a mixture of experts (MoE) to estimate the joint variational posterior based on individual unimodal posteriors. The MoPoE architecture [13] combines the benefits of the PoE and MoE approaches by computing the joint posterior for all subsets of modalities.

Although the above-mentioned models have been evaluated on various image-image or image-text benchmark datasets, they have not yet been tested in a robotic experiment where they would generate whole robotic trajectories. Zambelli et al. [17] proposed a multimodal VAE in which the joint latent space is learned using shared encoder and decoder network layers. Their model was evaluated on a sensomotoric learning scenario with an iCub robot playing a piano. However, the model can predict only one upcoming timestep at a time and thus does not allow learning and distinguishing multiple action classes based on the whole trajectories. Similarly, Meo et al. [18] adopted the MVAE model [10] for an adaptive torque controller that estimates the proprioceptive state of a Panda robot from multiple sensory inputs. Such adaptation, however, allows only one-on-one mapping among the modalities (e.g., joint positions to image of the robot) and does not focus on whole complex trajectories and their mapping to language.

In our research, we explore and propose modifications to state-of-the-art multimodal VAEs, enabling them to infer entire motion trajectories for object manipulation from raw image inputs and natural language instructions in an unsupervised manner. This task presents challenges since each modality provides distinct information at various levels of abstraction. For instance, images convey implicit details about scene objects and their positions, instructions specify target objects and actions symbolically, and the action modality informs about low-level task execution.

III. PRELIMINARIES

A. Task Description

We focus on a robotic manipulation task, where a robotic arm should manipulate randomly positioned objects according to a single top-view RGB image of the scene and a natural language command that specifies the type of action and/or the name of the target object. The robot is controlled using the x,y and z end-effector positions and an additional binary value g for opening and closing the gripper. The task requires mapping the actions, images, and language commands into a joint representation so that during inference the model is capable of: 1) generating the action based on the provided NL command and image, and 2) generating the NL description based on the provided trajectory and image. The mapping is achieved by training the model on a synthetic dataset.
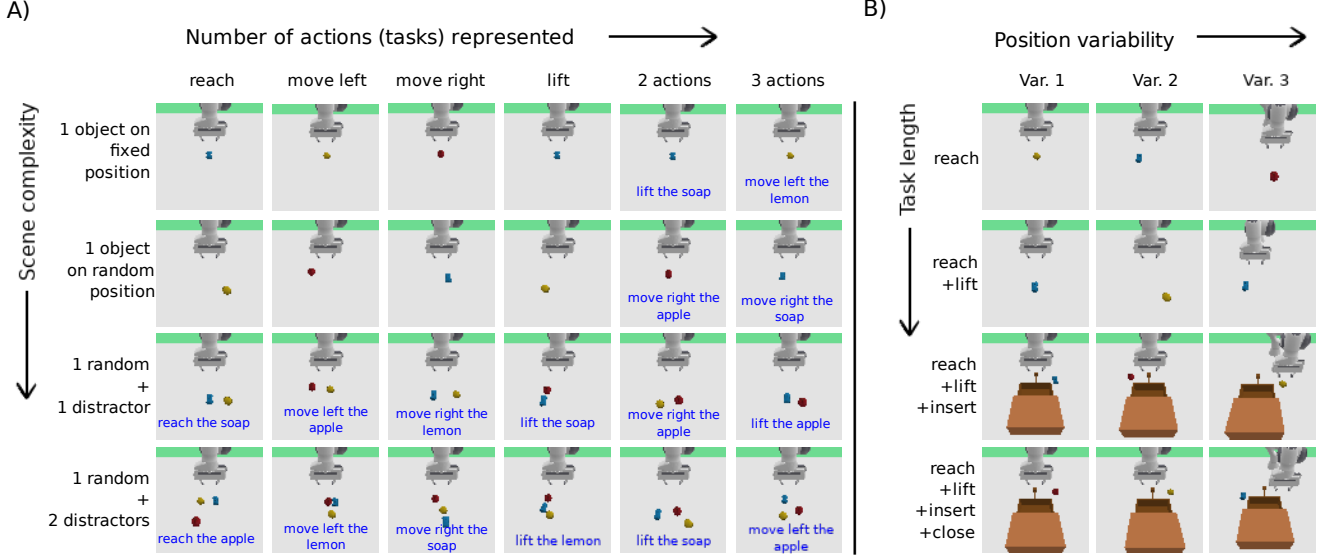
Fig. 2. Examples from the 36 datasets generated for our experiments. In each dataset, we use the top view of the scene with the robot as the visual input and the instruction as the text input (where necessary for task disambiguation). We also provide the task motion trajectories as another modality. **A:** the scene complexity (rows) and number of actions (tasks) concurrently represented by the model (columns), e.g., one action, *move right*, is represented by the model, or the model has to represent concurrently more actions (i.e., for 2 actions, the model learns together *move right* and *lift*). **B:** the task length (rows) and position variability (columns), i.e., Var. 1 varies object positions along the x axis, Var. 2. varies positions along axes x, y and Var. 3. additionally varies the robot position. For more details, see Section V.

B. VAEs

In this section, we briefly cover the theoretical background of the state-of-the-art multimodal VAEs that we adapted for the robotic experiment. For more detailed equations, please see the original papers.

Multimodal VAEs are built on top of the standard (unimodal) VAE as proposed by Kingma et al. [19], which is a probabilistic encoder-decoder network that learns to approximate the variational posterior $q_\phi(z|x)$ and optimize the evidence lower bound (ELBO):

$$\mathcal{L}_{ELBO}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}_i)||p(\boldsymbol{z})) \quad (1)$$

where the first term is the reconstruction loss, i.e., the expected negative log-likelihood of the i-th datapoint $-\sum_{i=1}^{M} \boldsymbol{y}_i log \hat{\boldsymbol{y}}_i$, and the second term is the Kullback-Leibler divergence between the encoder's distribution $q_\theta(z|x)$ and the prior $p(z)$, which is typically a Gaussian.

In addition to the individual posteriors for each input modality, multimodal VAEs also learn their joint representation, so that it is possible to provide any subset of the modalities on the input at test time. Such a model should then be able to perform cross-generation, i.e., to reconstruct one modality from another.

In MVAE [10], the joint variational posterior of the N modalities is approximated using the Product of Experts as

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \prod_n q_{\phi_n}(\boldsymbol{z}|\boldsymbol{x}_n), \quad (2)$$

assuming that the individual experts are Gaussian and the N modalities, $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$, are conditionally independent given the common latent variable z.

In contrast, MMVAE [11] uses a Mixture of Experts to factorize the joint variational posterior as a combination of unimodal posteriors

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}_1, ..., \boldsymbol{x}_N) = \sum_n \alpha_n \cdot q_{\phi_n}(\boldsymbol{z}|\boldsymbol{x}_n), \text{ where } \alpha_n = 1/N, \quad (3)$$

assuming that all inputs are of comparable complexity and all of them are present during the whole training. Although the MMVAE model can be trained with the traditional ELBO objective, the authors propose using the importance-weighted autoencoder (IWAE) objective together with the reparametrised gradient estimator DReG [11]. We thus use IWAE with DReG for the MMVAE model in our experiments.

The Mixture of Products of Experts (MoPoE) model [13] is then a generalization of the MVAE and MMVAE models:

$$q_{MoPoE}(\mathbf{z} \mid X) = \frac{1}{2^N} \sum_{X_k \in \mathcal{P}(X)} q_{PoE}(\mathbf{z} \mid X_k), \quad (4)$$

where $q_{PoE}(\mathbf{z} \mid X_k) \propto \prod_{\mathbf{x}_n \in X_k} q_{\phi_n}(\mathbf{z} \mid \mathbf{x}_n)$ and $\mathcal{P}(X)$ is the powerset of the N modalities. For more details, see the original paper [9].

IV. MULTIMODAL VAE FOR ROBOTIC SCENARIOS

First, we adjust the multimodal VAE-based architecture so that it can integrate the three input modalities in the vision-language-action scenario and produce motion trajectories from images and commands during inference. This adaptation is independent of the multimodal mixing methods (described in Sec. III, see also Fig. 1). Second, we employ the σ-VAE reconstruction loss term [20] in a multimodal

scenario and compare with the classical mean squared error loss.

Fig. 3.   The objects used in our experimental setup: apple, soap and lemon.

A. Architecture

In this section, we describe the encoder and decoder networks that were used for individual input modalities. You can see the overall architecture in Fig. 1.

1) Images: We fed a single RGB image with a top view of the scene to the VAE encoder. To extract the underlying image features, we used a ResNet-50 network pre-trained on ImageNet [21] as the encoder and 3 fully connected layers followed by 3 convolutional layers as the decoder.

2) Robotic Actions: We adapted the encoder and decoder networks for the motion trajectory of the ACTOR model designed for the generation of human motion, conditioned on the action label and the sequence length [22]. ACTOR is a VAE with transformer encoder and decoder networks with positional encodings, allowing variable-length sequence generation. To allow conditioning, the authors used a learnable bias b_a^{token} to shift the latent representation to an action-dependent space. We do not use this feature as the latent representation is already shifted after mixing with language instructions and the image in the joint posterior. We use an 8-layer (each with self-attention and feed-forward network) transformer network with 2 heads and 1024 hidden dimensions as the encoder and an 8-layer (each with self-attention, multi-head attention and feed-forward network) transformer network with 2 heads and 1024 hidden dimensions as the decoder. The model can handle the generation of variable-length trajectories due to positional encodings.

3) Language instructions: For the language encoder and decoder, we used transformer networks similar to the ones we used for the actions (see Sec. IV-A.2). The main difference is that we added a trainable embedding layer to process the text one-hot encodings on the input and compressed them into 2-dimensional features. These features were then fed into the positional encoder followed by the transformer. The decoder architecture remained the same as for the actions. Compared to the action encoder and decoder networks, we used different parameters - 1 layer, 2 heads, and 128 hidden dimensions for both the encoder and decoder.

B. Proposed training adjustment

The reconstruction loss term used in the ELBO objective can have an impact on the final reconstruction quality, as well as the stability of the model performance [23]. Compared to the standardly used mean squared error (MSE) or negative log-likelihood (NLL), we find that the quality

of the produced reconstructions of all multimodal models is improved when using the *optimal σ-VAE* proposed by Rybkin et al. [20] which uses an analytical estimate of the decoder variance:

$$\sigma^{*2} = \arg\max_{\sigma^2}\mathcal{N}\left(x \mid \mu, \sigma^2 I\right) = \text{MSE}(x, \mu), \qquad (5)$$

where $\text{MSE}(x,\mu) = \frac{1}{D}\sum_i\left(x_i - \mu_i\right)^2$. Here, the maximum likelihood estimate of the variance σ given a known mean μ is the average squared distance from the mean. In our experiments in Sec. V, we compare the MSE loss with the σ-VAE objective on 36 robotic datasets. We do not compare with the NLL loss as the results produced by training with this reconstruction loss are significantly worse than MSE or σ-VAE.

V. Experimental Setup

A. Simulated robotic environment

We used an adapted version of the LANRO (Language Robotics) simulator for our experiments [24]. The environment includes the Franka Emika robot equipped with a 2-finger gripper pointing downward, controlled using end-effector positions and binary values for gripper closing or opening. The robot is positioned on top of a table where objects are spawned during the experiment. The top-view camera is located above the centre of the table. For dataset collection, we used 3 textured objects - a soap, an apple and a lemon (see Fig. 3). In each trial, one or more objects are spawned 60-80 cm from the robot base (see Subsections V-B and V-C) and the environment provides a natural language instruction that disambiguates the task, i.e., provides information on the desired action in multi-task scenarios (*reach, lift, move right, move left*) and specifies the object in the case of multi-object datasets (e.g., *move left the soap*). In an additional set of experiments, we also include tasks where the robot has to place an object into a drawer and close it. Examples of setup and specific tasks are shown in Fig. 2.

B. Scene complexity and task types

To explore the multimodal VAE capabilities, we created data with task demonstrations performed by the robot based on the provided image of the scene and language instructions (used for task disambiguation when necessary). Using the virtual setup described in Section V-A, we first generated six datasets, each requiring the representation of a different number and type of actions (tasks), with four distinct levels of scene complexity for each case. This resulted in a total of 24 (6x4) datasets (see Fig. 2A). They differ based on the **scene complexity** as follows:

- **1 object in fixed position** - in each trial, one randomly selected object out of 3 is placed in the scene, always in the same location 60 cm from the robot base. The robot then performs one of the actions on the object (e.g., *move left*).
- **1 object in a random position** - one randomly selected object out of 3 is placed in the scene in a random

position varying between 60-80 cm from the robot in the *x*-axis and between -10 and +10 cm in the y axis.

- **1 object in a random place + 1 distractor** - one randomly selected object is placed in the scene in a random location. Additionally, another (different) object is placed somewhere in the scene. The instruction then specifies which object should be moved.
- **1 object in a random place + 2 distractors** - similar to the previous scenario, but with two distractors in the scene.

For each of the dataset scenes above, we generated 6 different training sets comprising a various number and types of tasks (robot actions) to perform on the selected object. The variability is the following:

- **reach** - the robot has to approach the selected object within the distance of 6 cm from the centre of the object, which is the optimal position for grasping
- **move left** - the robot has to lift the object and move it to the left by at least 10 cm.
- **move right** - the robot has to lift the object and move it to the right by at least 10 cm.
- **lift** - the robot has to lift the object up by at least 10 cm.
- **2 actions (move right or lift)** - the robot has to either perform action *lift* or *move right* based on the provided instruction.
- **2 actions (move right or move left or lift)** - the robot has to either perform action *lift* or *move right* or *move left* based on the provided instruction.

All of the aforementioned trainsets comprised 10,000 samples in single-task scenarios and 20,000 and 30,000 samples in the 2-action and 3-action datasets, respectively. Each trial involved collecting data from 3 modalities: the initial scene image (RGB, size 64x64x3), natural language instructions (one-hot encodings for each word), and the robotic action trajectory (4 values per timestep: x, y, and z coordinates of the end effector, and a binary value for the gripper).

C. Position variability and task length

Next, to test the robustness of the multimodal VAEs towards the position variability and task length, we generated datasets with four different task lengths (action sequences), each featuring three levels of positional variability. This resulted in 12 datasets (see Fig. 2B). The tasks used here were the following:

- **reach** - the robot has to approach the selected object under the distance of 6 cm (corresponds to *reach* in Subsection V-B)
- **reach+lift** - the robot has to lift the object and move it to the left by at least 10 cm (corresponds to *lift* in Subsection V-B).
- **reach+lift+insert** - the robot has to lift the object and place it inside an open drawer.
- **reach+lift+insert+close** - the robot has to lift the object, place it into the drawer and close it (see example in Fig. 4).

For each of the above tasks, we generated 3 trainsets differing in the amount of positional variability:

- **Variability 1** - the object (and drawer) position varied only along the *x* axis (e.g., closer or further from the robot) within the range of 20 cm.
- **Variability 2** - the positions of objects (and drawers) varied along the *x* and *y* axes by 20 cm (for *reach* and *reach+lift*, the variability of the position is the same as in the datasets used in Section V-B).
- **Variability 3** - the same as Variability 2, but the robot base position also varied along the *y* axis (e.g., along the table edge) by 40 cm.

We chose a positional variability of 20 cm in these experiments to ensure that the object remains visible in the camera image and to avoid interference with the drawer. As the task was always the same within each dataset (and there were no distractors), we did not include the natural language instructions in these datasets and provided only the image and the robotic action trajectories to the models. The number of time steps varied between tasks and trials, ranging from 13 time steps (*reach*) to 68 timesteps (*reach+lift+insert+close*).
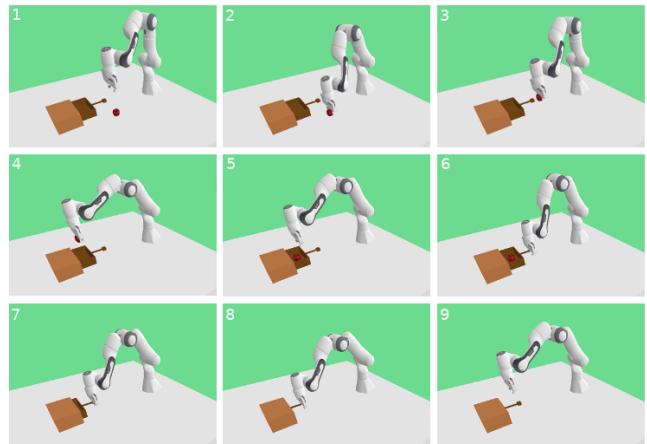

Fig. 4. Example of the *reach+lift+insert+close* sequence (ordered by numbers), which is the longest task in our datasets. The length of the sequence is up to 68 timesteps. For more examples, see the attached video.

D. Training setup

We compared the MVAE, MMVAE, and MoPoE models using the identical architecture proposed in Section IV and visualized in Fig. 1. The compared models thus have the same encoder and decoder networks, allowing for a direct comparison of the multimodal VAE approaches, with differences primarily arising from the mixing approach. We trained each model for 400 epochs and with the optimal latent space size based on a hyperparameter grid search (the final training parameters can be found on our GitHub [1]). After training, we used the test data to create previously unseen scenes in the LANRO simulator and used the model to predict the motion trajectories for the robot.

VI. RESULTS AND DISCUSSION

After training, we tested the multimodal VAEs in our test sets consisting of 500 trials. We provided only the image of the scene and natural language command (when needed for

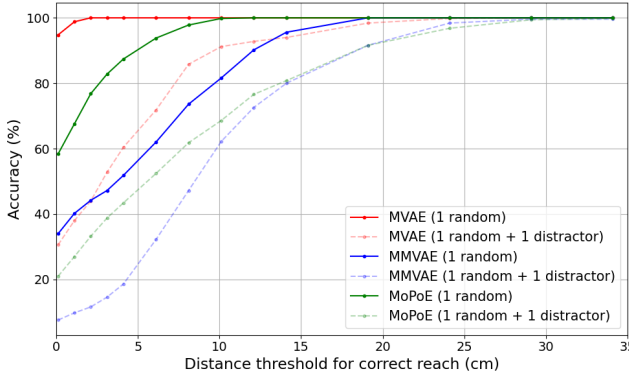| | | | Represented actions (tasks) | | | | | | | | | | | |
| | Accuracy [%] | Model | **Reach** | | **Move left** | | **Move right** | | **Lift** | | **2 Actions** | | **3 Actions** | |
			MSE	σ-VAE (proposed)	MSE	σ-VAE (proposed)	MSE	σ-VAE (proposed)	MSE	σ-VAE (proposed)	MSE	σ-VAE (proposed)	MSE	σ-VAE (proposed)
Scene complexity	1 fixed	MVAE	100	100	100	100	100	100	100	100	59	**67**	28	**30**
		MMVAE	100	100	100	100	100	100	100	100	100	100	67	**100**
		MoPoE	100	100	100	100	100	100	100	100	66	**67**	12	**22**
	1 random	MVAE	67	**97**	50	**67**	51	**63**	39	**54**	22	**28**	18	**20**
		MMVAE	7	**35**	5	**13**	1	**5**	4	**9**	4	4	1	**6**
		MoPoE	5	**60**	11	11	8	**17**	2	**10**	4	**15**	10	**21**
	1 random +1 distractor	MVAE	21	**32**	19	**21**	21	**23**	10	**14**	16	**19**	12	12
		MMVAE	5	**7**	2	**6**	6	6	3	**4**	3	**4**	1	**3**
		MoPoE	10	**21**	10	**12**	8	**15**	2	**4**	4	**6**	6	6
	1 random +2 distractors	MVAE	15	**20**	15	**16**	11	**14**	4	**6**	10	10	6	**7**
		MMVAE	0	**6**	0	**5**	1	**2**	0	**1**	1	**4**	4	**5**
		MoPoE	8	**9**	8	**10**	6	**10**	1	**3**	3	**7**	3	3

Fig. 5. Accuracy for the reach task based on the threshold of the final distance between the gripper and target reach position, which is 6 cm from the object centroid (optimal distance for grasping). We show the results for MVAE, MMVAE and MoPoE. *1 random* corresponds to the dataset with 1 randomly placed object, the data for *1 random + 1 distractor* also included a random distractor in the scene.

disambiguation) and the model had to generate the correct action. We considered the generated sample correct only if it reached or moved the object within the thresholds used to generate the training data (i.e., the gripper approached the object within less than 6 cm distance for the *reach* task, measured from the centroid of the object).

A. Comparing the reconstruction loss terms

In Table I, we compare the mean squared error loss (MSE) used as the reconstruction loss term with our proposed adjustment, the σ-VAE loss [20] (see Section IV-B). For all models and all datasets (shown in Fig. 2A), σ-VAE performed the same or better than MSE. The total average improvement in accuracy over all models and scenarios is 6.4 % (min. 0 %, max. 55 %). These results indicate that the batch-wise analytic estimate of the variance is beneficial also for multimodal VAE models, regardless of the multimodal mixing method.

B. Impact of the scene complexity and number of tasks

We find that all multimodal VAEs are able to learn a single action in a scenario with a non-variable setup with one object on a fixed position (see Table I, first row, columns 1-8). Although training two or three actions at the same time causes a continuous decrease in accuracy for the MVAE and MoPoE models, MMVAE was not affected by the number of tasks (see Table I, row 1, columns 9-12). The *reach* task accuracy provides insights into the model's capability to localize the correct object based solely on vision. Given the accuracy differences between *reach* and individual actions such as *move left* are generally not substantial, we can assume that the most challenging aspect lies indeed in the mapping between the pixel space into cartesian positions.

As for the scene complexity, all models were significantly impacted by the variability in object positions and by the presence of distractors. In Fig. 5, we show the accuracy of the *reach* task based on the threshold of the distance from the target object. When using one randomly positioned object, all of the inferred trajectories for MVAE and MoPoE were within the maximum distance of 10 cm from the target reach position (for MMVAE, the maximum distortion was 19 cm). However, when introducing one distractor, the proportion of severely distorted trajectories (more than 15 cm) increased. Such results indicate that mapping between pixel and cartesian space remains a challenging task for the SOTA multimodal VAEs. The incorporation of additional vision pre-processing modules might be beneficial.

C. Impact of the position variability and task length

In Table II, we show how task accuracy is influenced by both task length and position variability. Although there was a slight decrease in accuracy due to variations in object and robot positions, this impact was not as significant as the effect of the task length. Notably, both MVAE and MMVAE faced challenges in executing the longest task, which involved placing an object into the drawer and closing it. A potential solution to improve performance in such long-horizon tasks could involve training individual VAE experts for each subtask, taking into account variable object and robot positions, and then chaining these experts together to execute the entire task seamlessly.

D. Comparing the multimodal VAE models

Tables I and II show that the MVAE model consistently outperforms both MMVAE and MoPoE across most tasks. MVAE achieved 50 % accuracy in all single-task datasets with one randomly placed object. On the other hand, MM-VAE demonstrates higher accuracy on multi-task datasets (learning 2 or 3 actions simultaneously), yet only in the scenario with one fixed object, indicating limited robustness towards scene complexity. MoPoE, as a combination of both MVAE and MMVAE approaches, produced results with accuracies between the two models.

TABLE II

INFERENCE ACCURACIES (%) FOR MVAE AND MMVAE EVALUATED ON 4 TASKS: REACH, REACH + LIFT, REACH + LIFT + INSERT INTO DRAWER, REACH + LIFT + INSERT + CLOSE DRAWER. IN DATASET VAR. 1, THE OBJECTS' POSITIONS VARIED ONLY ALONG THE x AXIS, IN VAR. 2, THE OBJECTS MOVED ALONG BOTH THE x AND y AXES AND IN DATASET VAR. 3, WE ALSO VARIED THE POSITION OF THE ROBOT. WE USED THE σ-VAE OBJECTIVE TO TRAIN BOTH MODELS.

Accuracy [%]	Model	Var. 1	Var. 2	Var. 3
Reach	MVAE	**100**	**97**	**92**
	MMVAE	69	67	64
Reach+Lift	MVAE	**72**	**54**	**50**
	MMVAE	52	39	25
Reach+Lift+Insert	MVAE	**58**	**45**	**37**
	MMVAE	39	12	11
Reach+Lift+Insert+Close	MVAE	**16**	**14**	**13**
	MMVAE	1	0	0

VII. CONCLUSIONS

We employed three SOTA multimodal VAE models for vision-language-action mapping in robotic kinesthetic teaching and performed their systematic evaluation on 36 simulated datasets. We proposed an adapted architecture (see Fig. 1) with fine-tuned encoder and decoder networks for each modality and employed the σ-VAE objective which improved the models' performance by up to 55 %. We showed that the most suitable is the MVAE model, which had the most robust performance in terms of task length and scene complexity or position variability. In our future work, we would like to further explore the models' capabilities to recognize the correct object in a multi-object scene, as this task showed to be out of scope for the SOTA multimodal VAEs when combined with action learning.

ACKNOWLEDGMENT

REFERENCES

[1] J. K. Behrens, K. Stepanova, R. Lange, and R. Skoviera, "Specifying dual-arm robot planning problems through natural language and demonstration," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2622–2629, 2019.

[2] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, "Robots that use language," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 25–55, 2020.

[3] D. Shah, B. Osiński, S. Levine, *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on Robot Learning*. PMLR, 2023, pp. 492–504.

[4] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[5] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *arXiv preprint arXiv:2305.05658*, 2023.

[6] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *7th Annual Conference on Robot Learning*, 2023.

[7] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[8] Y. Chang, X. Wang, J. Wang, Y. Wu, K. Zhu, H. Chen, L. Yang, X. Yi, C. Wang, Y. Wang, *et al.*, "A survey on evaluation of large language models," *arXiv preprint arXiv:2307.03109*, 2023.

[9] M. Suzuki and Y. Matsuo, "A survey of multimodal deep generative models," *Advanced Robotics*, vol. 36, no. 5-6, pp. 261–278, 2022.

[10] M. Wu and N. Goodman, "Multimodal generative models for scalable weakly-supervised learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[11] Y. Shi, B. Paige, P. Torr, *et al.*, "Variational mixture-of-experts autoencoders for multi-modal deep generative models," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[12] M. Lee and V. Pavlovic, "Private-shared disentangled multimodal vae for learning of latent representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1692–1700.

[13] T. M. Sutter, I. Daunhawer, and J. E. Vogt, "Generalized multimodal ELBO," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=5Y21V0RDBV

[14] M. Suzuki, K. Nakayama, and Y. Matsuo, "Joint multimodal learning with deep generative models," *arXiv preprint arXiv:1611.01891*, 2016.

[15] M. Vasco, F. S. Melo, and A. Paiva, "Mhvae: a human-inspired deep hierarchical generative model for multimodal representation learning," *arXiv preprint arXiv:2006.02991*, 2020.

[16] T. Joy, Y. Shi, P. Torr, T. Rainforth, S. M. Schmon, and S. N, "Learning multimodal VAEs through mutual supervision," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=1xXvPrAshao

[17] M. Zambelli, A. Cully, and Y. Demiris, "Multimodal representation models for prediction and control from partial information," *Robotics and Autonomous Systems*, vol. 123, p. 103312, 2020.

[18] C. Meo and P. Lanillos, "Multimodal vae active inference controller," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2693–2699.

[19] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[20] O. Rybkin, K. Daniilidis, and S. Levine, "Simple and effective vae training with calibrated decoders," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9179–9189.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] M. Petrovich, M. J. Black, and G. Varol, "Action-conditioned 3d human motion synthesis with transformer vae," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 985–10 995.

[23] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "Videogpt: Video generation using vq-vae and transformers," *arXiv preprint arXiv:2104.10157*, 2021.

[24] F. Röder and M. Eppe, "Language-conditioned reinforcement learning to solve misunderstandings with action corrections," *arXiv preprint arXiv:2211.10168*, 2022.