ViSaRL: Visual Reinforcement Learning Guided by Human Saliency

Anthony Liang[†], Jesse Thomason[†], Erdem Bıyık[‡]

***Abstract*— Training autonomous agents to perform complex control tasks from high-dimensional pixel input using reinforcement learning (RL) is challenging and sample-inefficient. When performing a task, people visually attend to task-relevant objects and areas. By contrast, pixel observations in visual RL are comprised primarily of task-irrelevant information. To bridge that gap, we introduce Visual Saliency-Guided Reinforcement Learning (ViSaRL). Using ViSaRL to learn visual scene encodings improves the success rate of an RL agent on four challenging visual robot control tasks in the Meta-World benchmark. This finding holds across two different visual encoder backbone architectures, with average success rate absolute gains of 13% and 18% with CNN and Transformer-based visual encoders, respectively. The Transformer-based visual encoder can achieve a 10% absolute gain in success rate even when saliency is only available during pretraining.**

I. INTRODUCTION

Human visual attention helps to efficiently process and understand complex scenes by focusing on the most important regions in an image [1]. We hypothesize saliency maps capturing that human visual attention are a useful signal for visual scene encodings for AI agents. In this paper, we ask whether *human* visual attention helps *agents* perform tasks.

A key ingredient in solving visual control tasks is to learn visual representations that capture useful features of the sensory input to simplify the decision making process. Many works in the deep reinforcement learning (RL) community have proposed to learn such representations through various self-supervised objectives including contrastive learning [2] and data augmentation [3]. By contrast, we focus on self-supervision using *saliency* as additional human domain knowledge to inform the representation of task-relevant features in the visual input while filtering out perceptual noise.

We present **Vi**sual **Sa**liency **R**einforcement **L**earning (ViSaRL), a general approach for incorporating human-annotated saliency maps into learned visual representations.[1] The key idea of ViSaRL is to train a multimodal autoencoder that learns to reconstruct both RGB and saliency inputs, and an RL policy on top of the frozen autoencoder as shown in Figure 1. By using a masked reconstruction objective for the autoencoder, our approach encourages the learned representations to encode useful visual invariances and attend to the most salient regions for downstream task learning. To circumvent the manual labor of annotating saliency maps, we train a state-of-the-art saliency predictor model using only a few human-annotated examples to augment RGB observations with saliency.

[†]Department of Computer Science, University of Southern California, anthony.liang@usc.edu, jessetho@usc.edu

[‡]Center for Human-Compatible Artificial Intelligence, University of California, Berkeley, ebiyik@berkeley.edu

[1]The code implementation for reproducing the results can be found on the project website: https://sites.google.com/view/visarl.

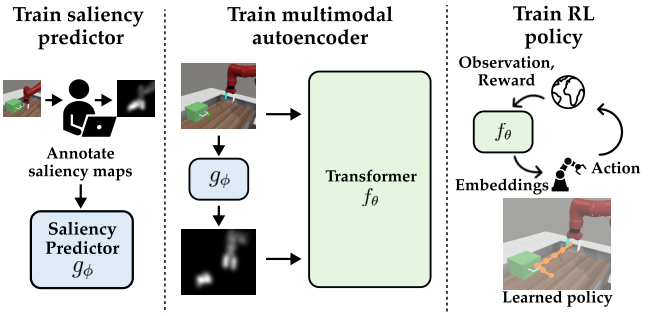

Fig. 1. ViSaRL learns a saliency-augmented visual scene encoder for downstream RL policy training. A saliency prediction network is trained from a handful of human-annotated maps. Then, RGB and predicted saliency are used to pretrain the scene encoder.

Overall, our contributions can be summarized as: noitemsep,nosep

1) We propose the ViSaRL framework for utilizing humans' attention in visual control tasks;
2) We develop an easy procedure, consisting of a user interface and a state-of-the-art saliency prediction model, for collecting and predicting human saliency maps as proxies for attention; and
3) We conduct extensive experiments that demonstrate ViSaRL conveniently and consistently improves success rate in various visual control tasks.

II. VISUAL SALIENCY-GUIDED REINFORCEMENT LEARNING

ViSaRL is a simple approach for incorporating human-annotated saliency to learn more robust representations for pixel-based control. We collect saliency annotations and utilize them for training a saliency predictor model. We augment an offline image dataset with saliency to pretrain CNN- and Transformer-based encoders (see Fig. 2) for extracting image representations that can be used during downstream reinforcement learning.

A. Generating Saliency Maps

We need only a handful of human saliency maps which we use to bootstrap the saliency predictor. We chose to use Pixel-wise Contextual Attention network (PiCANet) [23] which we discuss more in depth in Appendix E. We develop a custom graphical user interface (GUI) shown in Figure 5 to collect the saliency annotations. We then use the trained PiCANet

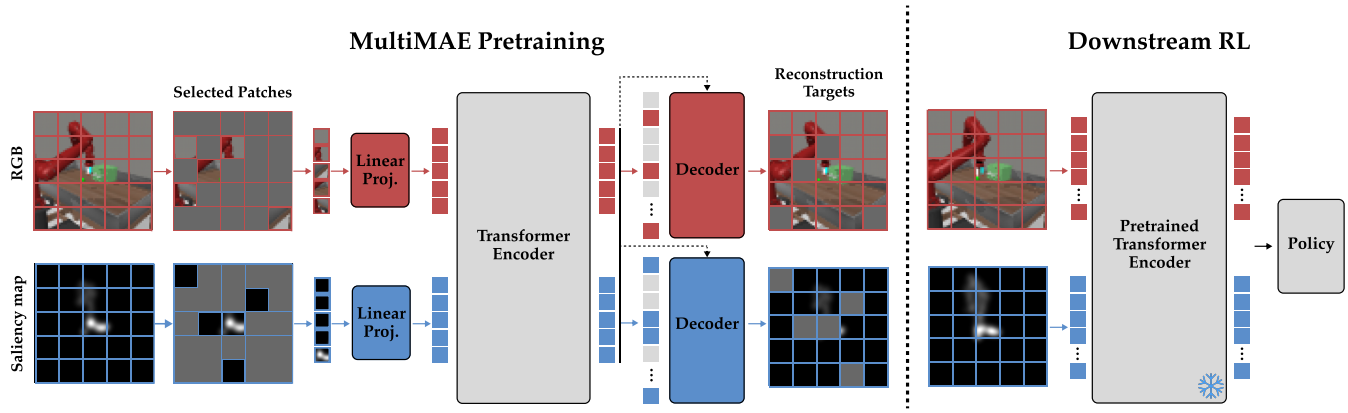

Fig. 2. **ViSaRL Approach**. We pretrain a MultiMAE model on a dataset of RGB images and their corresponding saliency obtained from a saliency predictor trained via supervised learning on a few labelled examples. The pretraining objective for MultiMAE is to reconstruct the masked patches for both input modalities given the encodings of only the visible patches. The pretrained model is frozen and used for extracting image representations used as input to the policy during task learning. There is no masking during downstream RL.

model to pseudo-label an offline dataset of RGB observations collected from the environment. That saliency-augmented image dataset can subsequently be used for pretraining any visual encoder.

B. Pretraining Visual Representation

In this work, we experiment with various techniques for augmenting both CNN-based and Transformer encoders with saliency information. To add saliency to a CNN architecture, we can multiply RGB with saliency (Figure 4) or simply add saliency as a fourth channel per pixel. We find that a Transformer backbone pretrained with a patch masking objective achieves higher task performance, even getting a 10% gain in success rate by using RGB and saliency at pretraining time when only RGB is available during downstream RL.

Masked autoencoders (MAE) [4] are an effective and scalable approach for visual representation learning. MAE masks out random patches of an image and reconstructs the masked patches using a Vision Transformer (ViT) [5]. More concretely, an observation $o_t \in \mathbb{R}^{H \times W \times C}$ is processed into a sequence of 2D patches $h_t \in \mathbb{R}^{K \times (P^2 C)}$ where P is the patch size and $K = HW/P^2$ is the number of patches. A subset of these patches are randomly masked out with a masking ratio of m. Masking reduces the input sequence length and encourages learning global, contextualized representations from limited visible patches. Only the *visible, unmasked patches* are then used as input to a ViT encoder which first embeds the patches via a linear projection, adds positional embeddings and then processes the set of tokens via a series of Transformer blocks. Finally, a ViT decoder reconstructs the original input by processing all of the tokens including the encoded visible patches and mask tokens. Following [4], we employ a high masking ratio $m = 0.75$ and a heavy-encoder light-decoder architecture design to enable efficiently learning good representations.

The MAE pretraining objective is limited to processing a single modality, RGB images. We propose to incorporate saliency as an additional modality during pretraining following the MultiMAE [6] architecture as shown in Figure 2. MultiMAE extends MAE to support pretraining with multiple input modalities (e.g., depth and segmentation maps). MultiMAE uses a different linear projection for each input modality. Similar to the RGB-only MAE, MultiMAE passes a small randomly sampled subset of tokens to the Transformer encoder to obtain the encoded tokens. Each modality has a separate lightweight decoder for reconstructing the masked tokens from the visible tokens. A cross attention layer is used in each decoder to incorporate information from the encoded tokens of other modalities using the tokens as queries and all the encoded tokens as keys and values. For efficiency, we fix the number of encoded tokens between both modalities to be 32 which roughly corresponds to 1/4 of all the tokens. Following [6], we employ a uniform sampling strategy to select tokens from each modality to encode. MultiMAE's pretraining objective requires the model to perform well in both the original MAE objective of RGB in-painting and cross-modal reconstruction, resulting in a stronger cross-modal visual representation.

C. Pretraining

We use visual observations of $64 \times 64 \times 3$. We use a 4-layer ViT encoder and a 3-layer ViT decoder with a patch size of 8×8 pixels. We pretrain the model for 400 epochs on an offline dataset of 200k images collected from the replay buffer of state-based SAC training. We use the same training hyperparameters as the MultiMAE paper. Additional pretraining details can be found in the Appendix (Table II).

D. Downstream Reinforcement Learning

After pretraining the MultiMAE we freeze the ViT encoder and use it to extract visual representations for downstream RL training. During RL, only the policy and Q-functions are trained and image inputs are not masked. We take an average over all the 128 token embeddings to generate a global representation of the image. We also tried using the

global learned token embedding, similar to a CLS token in ViT, which yielded similar results.

III. EXPERIMENTS

To demonstrate the effectiveness of using human-annotated saliency information to enhance visual representations for task learning, we show quantitative results of our approach with two different encoder backbones, CNN and MultiMAE, across four challenging Meta-World benchmark tasks [7]. Figure 3 and Table I summarize our main findings. Incorporating saliency input substantially improves downstream task success rate irrespective of the encoder backbone. Additionally, our proposed approach of using a MultiMAE objective for fusing the saliency annotations yields the best overall task performance between all the baseline methods.

A. Task Details

We evaluate our method on four different control tasks in the Meta-World robot manipulation benchmark [7]: {`Reach`, `Faucet Open`, `Door Open`, `Drawer Open`} shown in Figure 4. In all four tasks, the action space $\mathcal{A} \subset \mathbb{R}^4$, is the $\Delta(xyz)$ of the end-effector, and a continuous scalar value for gripper torque. Object and goal positions are randomized at the start of every episode, requiring the learned representation to be robust to visual shifts. As a consequence, the agent cannot exploit spurious correlations or memorize trajectories to solve the task.

B. Saliency Map Annotation

For each task, $N = 30$ observations from the environment are selected at random and presented to a human annotator in sequence. The annotator clicks on the pixels in the image that they think is relevant for performing the given downstream task. Each mouse click creates a standard Gaussian centered around the clicked pixel (see Fig.5). For manipulation tasks, this could include the end-effector position, typically with more saliency concentrated near the gripper tip, as well as the task specific objects and the goal location. Each observation takes roughly 30 seconds to a minute to annotate and 25 ± 5.8 mouse clicks (mean$\pm$std).

C. Accuracy of Saliency Prediction Model

We use 80% of the annotations for training and 20% for testing. We follow the training procedure and hyperparameters outlined in [8] and additionally apply random mirror-flipping for data augmentation. To evaluate that the trained predictor network is accurate, we report two main evaluation metrics from the original paper: F-measure score and Mean Absolute Error (MAE). F-measure score balances between both precision and recall:

$$F_\zeta = \frac{(1+\zeta^2) Precision \times Recall}{\zeta^2 Precision + Recall}, \tag{1}$$

where ζ^2 is set to 0.3. Saliency maps are first binarized before computing F_ζ. MAE computes the average absolute per-pixel difference between the predicted saliency maps and ground truth saliency maps. We find that on our testing data, we obtain an F_ζ score of 0.78 ± 0.02 and MAE of 0.004 ± 0.003 averaged across the four tasks. These values are consistent with the competitive results reported in [8]. We provide qualitative results of the saliency prediction for different observations in Figure 7 in the Appendix.

D. CNN Encoder Results

We first present results using a CNN-based encoder trained through RL critic updates. We follow the CNN architecture and hyperparameters used in [9], [10], the full details of which can be found in Appendix F. We evaluate several different methods of incorporating saliency using a CNN encoder:

1) **RGB**: The CNN encoder and the policy are jointly trained with RGB images as inputs.
2) **Saliency**: The CNN encoder and the policy are jointly trained with the predicted saliency maps as inputs.
3) **RGB $\times$ Saliency**: The CNN encoder and the policy are jointly trained with RGB $\times$ saliency map as inputs.
4) **RGBS**: The CNN encoder and the policy are jointly trained with inputs that consist of the RGB images and saliency predictions as an additional channel.

In Table I, we find that naive ways of utilizing saliency do not yield good performance and is unable to learn to solve the task. We hypothesize that using only saliency as input (**Saliency**) fails to solve any of the tasks because the saliency map alone is not sufficient for the model to infer the exact orientation of the end-effector position which is critical especially for fine manipulation. Supporting this hypothesis, we find that using saliency to mask the RGB observation (**RGB $\times$ Saliency**) yields better performance than **Saliency**, but is still worse than providing the full RGB input (**RGB**). Although masking should help the encoder identify the important image features, it may still be nontrivial for the encoder to differentiate between similarly masked observations. Lastly, we find that incorporating saliency as an additional channel to the RGB input (**RGBS**) can improve task success rate by $> 10\%$ across all tasks. We hypothesize that the CNN encoder is able to utilize the saliency information to more effectively associate the observed rewards to the relevant features in the image.

E. MultiMAE Results

Unlike the CNN-based experiments where the encoder weights are learned from scratch along with the policy, we first pretrain the MultiMAE encoder with an offline dataset of (image, saliency map) pairs as an autoencoder. We then keep the encoder weights frozen during downstream RL, decoupling the representation learning and policy learning. MultiMAE experiments use the same SAC training hyperparameters as the CNN experiments (see Appendix F). We highlight the benefits of saliency and using a MultiMAE pretraining procedure by employing the following methods:

1) **RGB**: The MultiMAE encoder is pretrained using RGB images only.
2) **RGB+Saliency Pretrain Only (PO)**: The MultiMAE encoder is pretrained with both the RGB image and

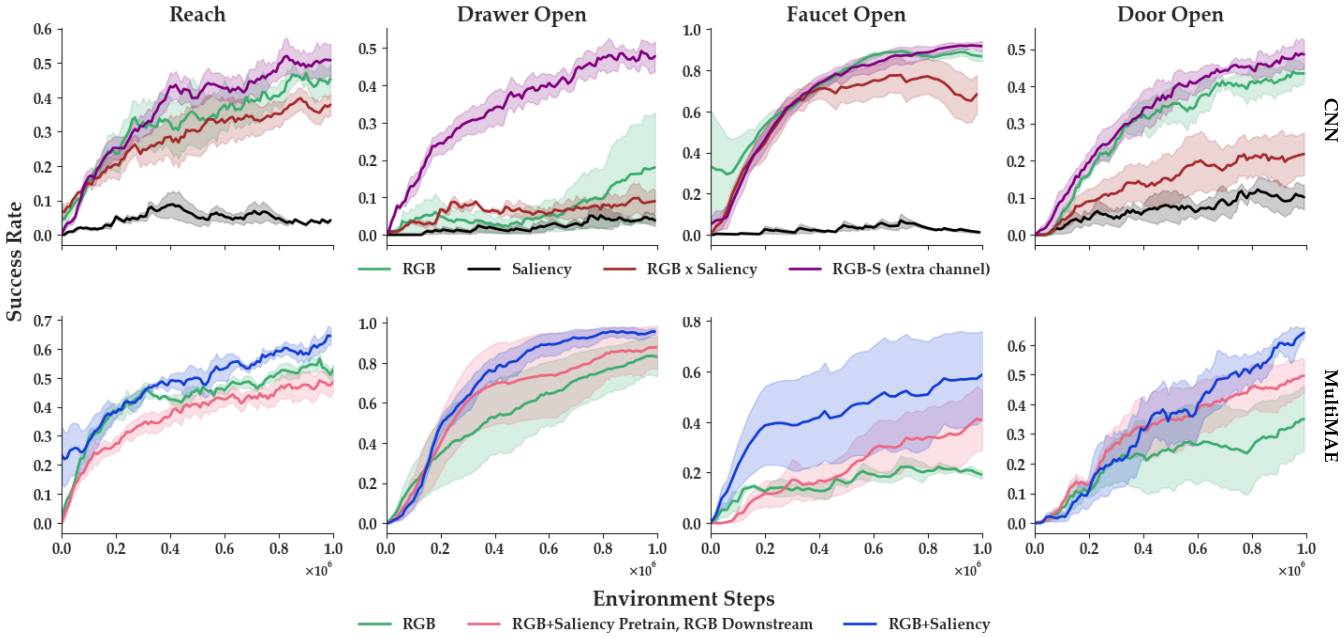

Fig. 3. **Learning curves** for four visual robotic manipulation tasks in Meta-World evaluated by task success rate. **(Top)** CNN encoder methods. **(Bottom)** Transformer encoder methods. We select tasks that require manipulating small objects with different motions such as a pushing, pulling, and reaching. The solid lines represent the mean and shaded region the standard error across three seeds.

predicted saliency, but uses only the RGB input during downstream RL.

3) **RGB+Saliency**: The MultiMAE encoder is pretrained with both the RGB image and predicted saliency, and uses both as inputs during downstream RL.

We compare the full ViSaRL method (**RGB+Saliency**) to pretraining using only the RGB images (**RGB**) in Table I, which shows that multimodal pretraining with saliency information significantly outperforms single modality pretraining by at least a 10% margin across all tasks. Notably, **RGB** achieves only 19% success on `Faucet Open`, while our approach can solve the task with 61% success rate. We also show that even without saliency input during downstream RL, using saliency as an additional input modality during pretraining still improves downstream performance on 3 of the 4 tasks. Except for the `Reach` task, where performances are similar, **RGB+Saliency(PO)** achieves better success rate than **RGB**, with an average absolute gain of 10% across tasks. Using saliency as an input for both pretraining and downstream RL (**RGB+Saliency**) is better than just during pretraining likely because there are new observations during online training that were not in the pretraining dataset and the encoder could benefit from the saliency input to generate a better state representation. These results show that our approach of incorporating human-annotated saliency information can help learn better visual representations to facilitate better task learning.

IV. Conclusion

Summary. In this paper, we propose to use human saliency as an additional input modality for solving challenging visual robot control tasks and present a simple approach to incorporate saliency for improving task performance. We train a state-of-the-art saliency predictor model using only a handful of human annotations to accurately predict saliency maps of unseen frames. We then pretrain a Multimodal MAE model on a saliency-augmented offline dataset of RGB images to generate visual representations. We find that both saliency input and the Transformer pretraining are crucial for achieving strong performance on a variety of visual control tasks in Meta-World.

Limitations and Future Work. One potential limitation of our user interface is that it could be tedious to collect saliency annotations when scaling to more complex real world applications or video saliency [11]. Future work could investigate alternative interfaces that will enable collecting more saliency data, e.g., area-based methods or by tracking the eye gaze of the user [12]. Additionally, a comprehensive study of these various user interfaces with many human subjects could reveal their strengths and weaknesses, potentially pointing out venues for improvement for more reliable and cheaper saliency maps.

In this paper, we only considered static frame saliency maps for atomic manipulation tasks that interact with only a single object. To extend our approach to handle multi-object manipulation and longer-horizon tasks, one could consider video saliency models [13] which can learn to encode more flexible temporal saliency representations across a sequence of frames. This extension could be done by asking the human users to watch some video clips of the trajectories and annotate saliency over these clips.

REFERENCES

[1] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[2] 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.

[3] I. Kostrikov, D. Yarats, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," *arXiv preprint arXiv:2004.13649*, 2020.

[4] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[6] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir, "Multimae: Multi-modal multi-task masked autoencoders," *arXiv preprint arXiv:2204.01678*, 2022.

[7] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.

[8] N. Liu, J. Han, and M.-H. Yang, "Picanet: Learning pixel-wise contextual attention for saliency detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3089–3098.

[9] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," *Advances in neural information processing systems*, vol. 33, pp. 19 884–19 895, 2020.

[10] D. Bertoin, A. Zouitine, M. Zouitine, and E. Rachelson, "Look where you look! saliency-guided q-networks for visual rl tasks," *arXiv preprint arXiv:2209.09203*, 2022.

[11] W. Wang, J. Shen, F. Guo, M.-M. Cheng, and A. Borji, "Revisiting video saliency: A large-scale benchmark and a new model," in *Proceedings of the IEEE Conference on computer vision and pattern recognition*, 2018, pp. 4894–4903.

[12] D. P. Papadopoulos, A. D. Clarke, F. Keller, and V. Ferrari, "Training object class detectors from eye tracking data," in *European conference on computer vision*. Springer, 2014, pp. 361–376.

[13] D. Rudoy, D. B. Goldman, E. Shechtman, and L. Zelnik-Manor, "Learning video saliency from human gaze using candidate selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1147–1154.

[14] S. Cabi, S. G. Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, *et al.*, "Scaling data-driven robotics with reward sketching and batch reinforcement learning," *arXiv preprint arXiv:1909.12200*, 2019.

[15] A. Bobu, M. Wiggert, C. Tomlin, and A. D. Dragan, "Feature expansive reward learning: Rethinking human input," in *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021, pp. 216–224.

[16] N. Wilde, E. Biyik, D. Sadigh, and S. L. Smith, "Learning reward functions from scale feedback," in *Conference on Robot Learning*. PMLR, 2022, pp. 353–362.

[17] S. Tao, X. Li, T. Mu, Z. Huang, Y. Qin, and H. Su, "to-executable trajectory translation for one-shot task generalization," in *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

[18] Y. Tong, H. Konik, F. Cheikh, and A. Tremeau, "Full reference image quality assessment based on saliency map analysis," *Journal of Imaging Science and Technology*, vol. 54, no. 3, pp. 30 503–1, 2010.

[19] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[20] T. N. Mundhenk, B. Y. Chen, and G. Friedland, "Efficient saliency maps for explainable ai," *arXiv preprint arXiv:1911.11293*, 2019.

[21] R. Zhao, W. Oyang, and X. Wang, "Person re-identification by saliency learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 2, pp. 356–370, 2016.

[22] A. Atrey, K. Clary, and D. Jensen, "Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning," *arXiv preprint arXiv:1912.05743*, 2019.

[23] M. Rosynski, F. Kirchner, and M. Valdenegro-Toro, "Are gradient-based saliency maps useful in deep reinforcement learning?" *arXiv preprint arXiv:2012.01281*, 2020.

[24] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[26] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations: A survey," *ACM Transactions on Applied Perception (TAP)*, vol. 7, no. 1, pp. 1–39, 2010.

[27] J. Zhang and S. Sclaroff, "Saliency detection: A boolean map approach," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 153–160.

[28] A. Boyd, K. W. Bowyer, and A. Czajka, "Human-aided saliency maps improve generalization of deep learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2735–2744.

[29] A. Boyd, P. Tinsley, K. W. Bowyer, and A. Czajka, "Cyborg: Blending human saliency into the loss improves deep learning-based synthetic face detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 6108–6117.

[30] C. Craye, T. Lesort, D. Filliat, and J.-F. Goudou, "Exploring to learn visual saliency: The rl-iac approach," *Robotics and Autonomous Systems*, vol. 112, pp. 244–259, 2019.

[31] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa, "dm_control: Software and tasks for continuous control," *Software Impacts*, vol. 6, p. 100022, 2020.

[32] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[33] A. Sax, B. Emi, A. R. Zamir, L. Guibas, S. Savarese, and J. Malik, "Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies," *arXiv preprint arXiv:1812.11971*, 2018.

[34] X. Wang, L. Gao, J. Song, and H. Shen, "Beyond frame-level cnn: saliency-aware 3-d cnn with lstm for video action recognition," *IEEE signal processing letters*, vol. 24, no. 4, pp. 510–514, 2016.

[35] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" *Computer Vision and Image Understanding*, vol. 163, pp. 90–100, 2017.

[36] Q. Li, Y. Zhou, and J. Yang, "Saliency based image segmentation," in *2011 International Conference on Multimedia Technology*. IEEE, 2011, pp. 5068–5071.

[37] M. Kümmerer, L. Theis, and M. Bethge, "Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet," *arXiv preprint arXiv:1411.1045*, 2014.

[38] S. S. Kruthiventi, K. Ayush, and R. V. Babu, "Deepfix: A fully convolutional neural network for predicting human eye fixations," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4446–4456, 2017.

[39] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[41] M. Kümmerer, T. S. Wallis, and M. Bethge, "Deepgaze ii: Reading fixations from deep features trained on object recognition," *arXiv preprint arXiv:1610.01563*, 2016.

[42] A. Linardos, M. Kümmerer, O. Press, and M. Bethge, "Deepgaze iie: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 919–12 928.

APPENDIX

A. Related Works

Different forms of human data can be leveraged when solving control tasks. Researchers created various interfaces to collect different data modalities from humans, for example: reward sketches [14], feature traces [15], scaled comparisons [16], and abstract trajectories [17]. Robots and machine learning models may benefit from tapping into different human data sources. Attention saliency maps are special in that they do not require humans to work with abstract concepts like rewards and task features, or those that require watching lengthy trajectories like comparisons.

Attention saliency maps have been used in both computer vision and machine learning for various applications. Methods of annotating saliency vary, as do those for incorporating saliency and other mid-level vision information into learned scene representations for reinforcement learning.

B. Saliency Maps

Saliency maps approximate which parts of an image tend to attract human visual attention, which corresponds to where the human eye would likely fixate when viewing an image [18]. They have been widely studied in the computer vision and explainable artificial intelligence communities to understand how a model is making its predictions and to identify the most informative regions of an image for a particular task [19], [20], [21]. Most existing works explore using saliency maps only as tools for interpretation [22], [23]. [22] use saliency maps to rationalize and explain the actions of RL agents in Atari games. [23] use various backpropagation-based techniques to visualize the saliency for trained RL policies.

Saliency maps can be categorized as either bottom-up or top-down. Bottom-up saliency maps, also known as feature-based saliency maps, highlight regions of the input that have the most distinctive or important features. They are typically computed by taking the gradient of the model output with respect to the input features such as Guided Backprop [24] and GradCam [25]. Top-down saliency maps use task-specific information to identify the important elements of the input that are most relevant to the task by taking into account prior knowledge or context about the task [26], [27]. [28] and [29] show top-down saliency maps for encoding prior human knowledge help tackle the problem of biometric attack detection and enable better generalization of deep learning models, respectively. [30] propose a method for incrementally learning saliency maps in autonomous robot navigation and utilizing them to improve exploration. ViSaRL uses top-down saliency maps, requiring a small number of human annotated maps that provide important information about task-relevant regions of images.

Saliency maps can be categorized as fixation-based [1] or area-based [27]. Fixation-based saliency maps measure the probability of a human fixating on a given pixel location, while area-based saliency maps consider objects as an entity similar to object segmentation. ViSaRL utilizes fixation-based saliency, but could be extended to incorporate area-based saliency.

Closely related to our work is [10] which incorporates saliency maps as a self-supervised regularization objective for robot control tasks in DMControl benchmark [31]. By contrast, we do not use saliency as a regularization objective during the policy training, but rather use human saliency annotations to highlight the crucial input pixels and distill this knowledge into the visual representation.

C. User Interfaces for Human Saliency

ViSaRL needs a small number of human-annotated saliency maps to bootstrap the saliency prediction network (Figure 1). Prior work used superpixel segmentation [32] to first divide each image into segments, and then asked humans to click on the segments that are salient [21]. However, that method requires manually checking and combining the segments that belong to the same object before showing the images to annotators, burdening system designers.

As an alternative, [28], [29] used interfaces where the annotators created binary masks by simply clicking on images. These binary maps were then smoothed by averaging over multiple annotators. We employ a similar but simpler interface: one annotator clicks on the salient parts of the image, and a simple Gaussian kernel is applied around activated pixels to achieve smooth saliency maps.

D. Representation Learning for RL

Saliency maps are essentially representations of the environment that carry useful domain knowledge about which regions of the visual input are important for the downstream task. Such representations are crucial in reinforcement learning because they enable agents to tractably deal with large observation spaces like images. Several approaches have been proposed to improve representation learning for RL.

Prior works have shown that self-supervised learning with data augmentation helps achieve good performance in image-based RL. Contrastive Unsupervised RL (CURL) [2] employed a contrastive learning objective as an auxiliary loss to learn representations for off-policy RL. RL with Augmented Data (RAD) [9] and Data Regularized Q-Learning (DrQ) [3] showed that simple image augmentations such as random cropping and color jittering may provide strong regularization and introduce inductive biases that enhance the performance of RL algorithms. ViSaRL does not use data augmentation directly in the value function or policy update. Instead, saliency augmentation is introduced during the visual encoder pretraining phase.

[33] demonstrated that mid-level visual representations such as surface normals or depth predictions from RGB images can boost performance of RL tasks by removing unimportant information and providing linearly separable outputs to simplify downstream decision making. Similar to [33], ViSaRL utilizes saliency maps as mid-level feature. However, we empirically show that our approach for incorporating the saliency information into the visual representation

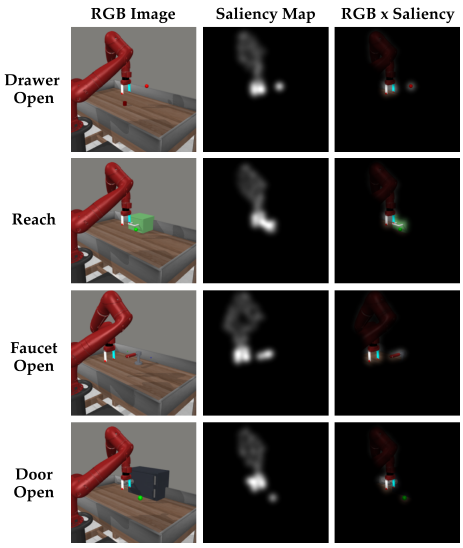

Fig. 4. Examples of visual observations, annotated saliency maps, and RGB $\times$ saliency from the tasks in our experiments.

improves task performance beyond just using saliency as direct input to the policy.

E. Saliency Predictor Network

Saliency prediction is widely used in a variety of computer vision applications including activity recognition [34], question answering [35], and object segmentation [36]. Given an input RGB image observation, $o_t \in \mathbb{R}^{H \times W \times 3}$, a saliency prediction model g maps the input image I to a continuous saliency map $M_t = g(o_t) \in [0,1]^{H \times W}$ which highlight important parts of the image for the downstream task.

There have been many deep convolutional neural network (CNN) based saliency prediction models, e.g., [37], [38], that have been proposed in the literature. We chose to use Pixel-wise Contextual Attention network (PiCANet) [8]. PiCANet hierarchically embeds global and local pixel-wise attention modules to selectively attend to informative context. Global attention can attend to backgrounds for foreground objects while local attention can attend to regions that have similar appearance which makes the saliency prediction more homogeneous and consistent.

PiCANet samples feature maps from different CNN layers to facilitate saliency inference at each pixel. Given convolutional feature maps at different scale $\mathcal{F} \in \mathbb{R}^{H \times W \times C}$, global attention generates attention over the whole feature map at each spatial location (w, h) in $\mathcal{F}$ while local attention works on a local region centered at (w, h). PiCANet is based on a U-net [39] architecture. The encoder is a VGG [40] backbone that operates on images of size 224×224. The original PiCANet uses 6 decoder modules. Each decoder module upsamples the intermediate feature maps and either applies a global or local PiCANet to obtain the attended contextual feature map. We opted to remove the global attention decoder layers, reducing our inference time per frame from 0.1 seconds to 0.01 seconds. Qualitatively, we observed little degradation in saliency prediction without these layers.

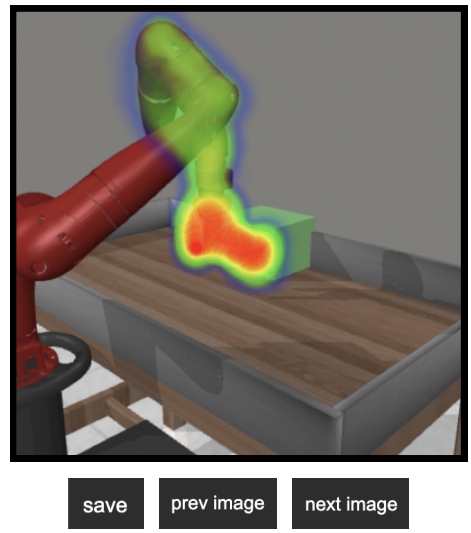

Fig. 5. Web interface for collecting saliency annotations. Frames are presented to the user one at a time. Users click on salient pixels in the frame, and each click generates a Gaussian kernel centered at that location with $\sigma = 10$ pixels. Warmer colors represent higher saliency, with unmarked regions having none.

F. Network architecture

The CNN encoder implementation is based on [9] and [10]. The encoder consist of a stack of 11 convolutional layers, each with 32 filters of 3x3 kernels, no padding, stride of 2 for the first and 1 for all the others. This results in a feature map of dimension $32 \times 12 \times 12$ given an input image of shape $64 \times 64 \times 3$.

The policy head π_θ and action-value functions Q_{ψ_i} are parameterized by multi-layer perceptions (MLP).

The policy head is composed of a linear projection of dimension 100 with normalization followed by 3 linear layers with 1024 hidden units each and a final linear output layer for the action prediction. The embedding from CNN encoder is first flattened before inputted to the policy network. Q_{ψ_i} share the same structure as the policy network.

G. Implementation Details

We explain the implementation details for the PiCANet architecture used for generating saliency maps.

		Reach	Drawer Open	Faucet Open	Door Open	**Average**
CNN	RGB	0.39 ± 0.13	0.18 ± 0.25	0.82 ± 0.02	0.42 ± 0.04	0.45 ± 0.11
	Saliency	0.04 ± 0.01	0.04 ± 0.02	0.01 ± 0.01	0.10 ± 0.06	0.05 ± 0.03
	RGB × Saliency	0.38 ± 0.05	0.09 ± 0.04	0.71 ± 0.16	0.22 ± 0.10	0.35 ± 0.09
	RGBS	0.51 ± 0.07	0.48 ± 0.07	**0.85** ± 0.01	0.48 ± 0.07	0.58 ± 0.37
MMAE	RGB	0.49 ± 0.03	0.83 ± 0.20	0.19 ± 0.03	0.35 ± 0.19	0.47 ± 0.11
	RGB+Saliency(PO)	0.48 ± 0.06	0.88 ± 0.21	0.40 ± 0.21	0.52 ± 0.08	0.57 ± 0.14
	RGB+Saliency (ours)	**0.62** ± 0.05	**0.94** ± 0.03	0.61 ± 0.16	**0.64** ± 0.02	**0.65** ± 0.07

TABLE I

SUCCESS RATE ACHIEVED BY VISARL ON FOUR MANIPULATION TASKS FROM META-WORLD AVERAGE ACROSS 50 ROLLOUTS AND 3 SEEDS FOR THE CNN AND MULTIMAE (MMAE) VISUAL ENCODER BACKBONES. VISARL ACHIEVES THE BEST PERFORMANCE ON 3 OF THE 4 TASKS AND BEST AVERAGE PERFORMANCE AMONGST ALL OF THE BASELINES. TEXT IN **MAROON** REPRESENT THE BEST PERFORMING METHOD.

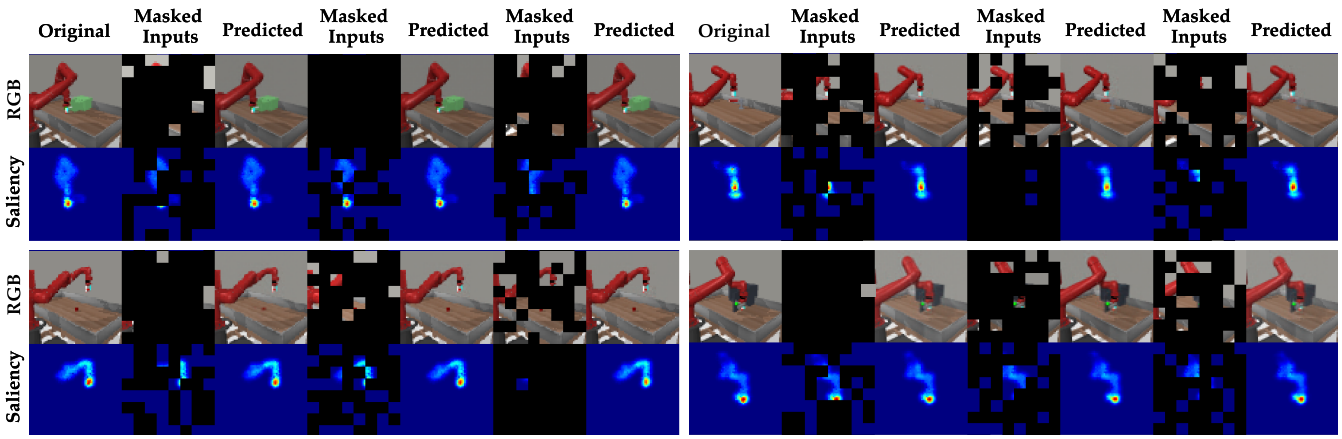

Fig. 6. **MultiMAE predictions for different random masks.** We visualize the masked predictions for RGB observation from each of the four tasks. For each input image, we randomly sample three different masks from a uniform distribution between RGB and saliency. Only 1/4 of the total patches are unmasked. Even when there are a few unmasked patches from one modality, the reconstructions are still very accurate due to cross-modal interaction. Saliency maps are shown with color for the purposes of visualization.

Hyperparameter	Value
Augmentations	RandomResizedCrop
Optimizer	AdamW
Base learning rate	1e-4
Weight decay	0.05
Warmup learning rate	1e-6
Warmup epochs	40
Num epochs	200
Batch size	512
Dataset size	200k
Loss function	WeightedMSE
Non-masked tokens	32
Sampling α	1.0
Input resolution	64 x 64 x 3
Number of parameters	12M

TABLE II

HYPERPARAMETERS USED FOR PRETRAINING MULTIMAE MODEL FOLLOWING [6].

Hyperparameter	Value
Augmentations	ColorJitter
Optimizer	AdamW
Learning rate	3e-4
Num epochs	1000
Batch size	16
LR decay	0.1
Weight decay	0.005
Momentum	0.9
Dataset size	30
Loss function	WeightedMSE

TABLE III

HYPERPARAMETERS USED FOR TRAINING PICANET MODEL TO GENERATE PSEUDO-SALIENCY MAPS ANNOTATIONS FOLLOWING [8].

Method	Mean Absolute Error	F_ζ score
DeepGaze II [41]	0.0273 ± 0.006	0.7142 ± 0.021
DeepGaze IIE [42]	0.0153 ± 0.006	0.7283 ± 0.024
PiCANet [8]	0.0032 ± 0.002	0.7970 ± 0.015

TABLE IV

PERFORMANCE OF DIFFERENT STATE-OF-THE-ART PREDICTORS

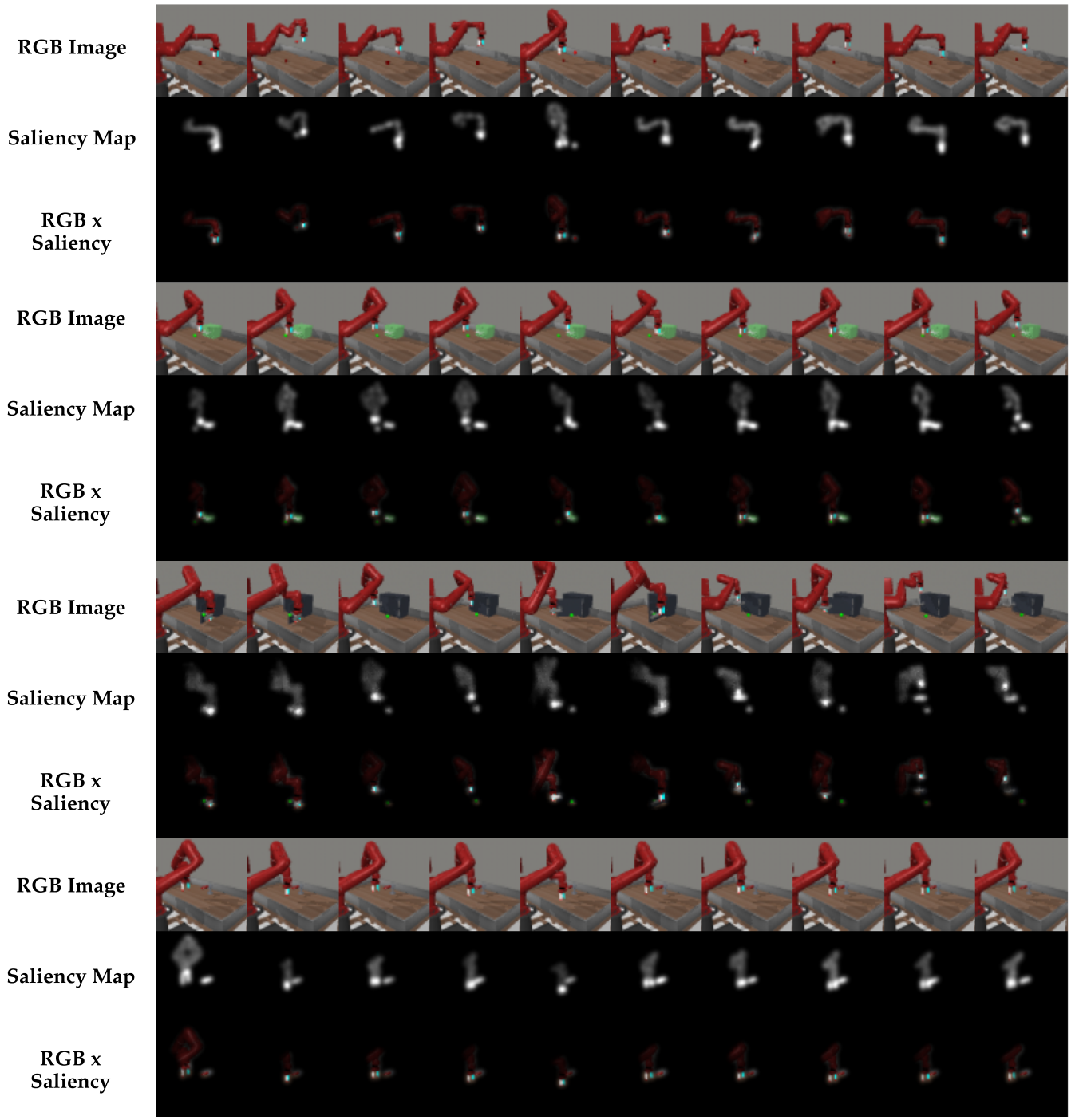

Fig. 7. Additional examples of visual observations, predicted saliency, and RGB $\times$ saliency from the tasks in our experiments.

Number of annotations	Mean Absolute Error	F_ζ score
5	0.0086	0.7983
10	0.0063	0.7948
20	0.0053	0.8006
30	0.0020	0.8006

TABLE V

EFFECT OF TRAINING DATASET SIZE ON SALIENCY PREDICTION

Number of annotations	Mean Absolute Error	F_ζ score
PiCANet [8] scratch	0.0032 ± 0.002	0.7970 ± 0.015
PiCANet [8] fine-tuned	0.0025 ± 0.0015	0.8010 ± 0.018

TABLE VI

PICANET FROM SCRATCH V.S. FINE-TUNED FROM PRETRAINED MODEL