Test-time adaptation with slot-centric models

Mihir Prabhudesai[†,2], Sujoy Paul[1], Sjoerd van Steenkiste[1], Mehdi S. M. Sajjadi[1], Anirudh Goyal[3], Deepak Pathak[2], Katerina Fragkiadaki[2], Gaurav Aggarwal[1,*], and Thomas Kipf[1,*]

[1]Google Research [2]Carnegie Mellon University [3]Mila, DeepMind

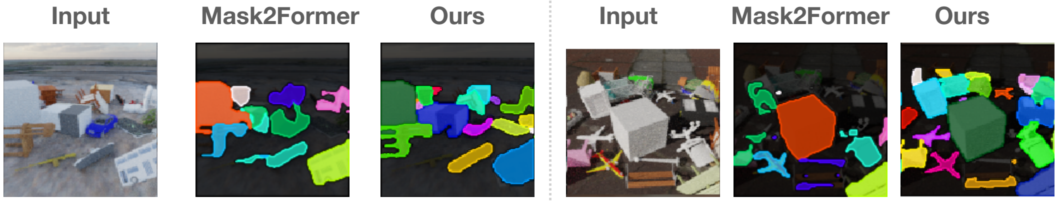

Figure 1: **Image segmentation**. Slot-TTA parses completely novel scenes into familiar entities via slow inference, i.e., gradient descent on the reconstruction error of the scene example under consideration. Slot-TTA outperforms Mask2Former [6], a SOTA 2D image segmentor, on segmenting novel images by gradient descent on image synthesis of neighboring image views.

Abstract

We consider the problem of segmenting scenes into constituent objects. Current supervised visual detectors, though impressive within their training distribution, often fail to segment out-of-distribution scenes. Recent test-time adaptation methods use auxiliary self-supervised losses to adapt the network parameters to each test example independently and have shown promising results towards generalization outside the training distribution for the task of image classification. In our work, we find evidence that these losses can be insufficient for instance segmentation tasks, without also considering architectural inductive biases. For image segmentation, recent slot-centric generative models break such dependence on supervision by attempting to segment scenes into entities in a self-supervised manner by reconstructing pixels. Drawing upon these two lines of work, we propose Slot-TTA, a semi-supervised instance segmentation model equipped with a slot-centric image rendering component, that is adapted per scene at test time through gradient descent on reconstruction or novel view synthesis objectives. We show that test-time adaptation greatly improves segmentation in out-of-distribution scenes. We evaluate Slot-TTA in scene segmentation benchmarks and show substantial out-of-distribution performance improvements against state-of- the-art supervised feed-forward detectors and self-supervised domain adaptation models. Please find the full version of our paper at: https://arxiv.org/abs/2203.11194

1 Introduction

While significant progress has been made in machine scene perception and segmentation within the last decade, object detectors continue to generalize poorly outside their training distribution [12, 22]. Consider the extremely cluttered scenes shown in Figure 1. We can intuitively reason about meaningful parts that this scene could be broken into. Yet, a state-of-the-art 2D detector [6]

[†]Work done while interning at Google, [*]equal contribution. Contact: `mprabhud@andrew.cmu.edu`

5th Workshop on Meta-Learning at NeurIPS 2022, held hybrid.

trained to segment similar objects in less cluttered scenes (5-7 object instances) struggles with this decomposition. This lack of generalization requires us to build systems that can robustly adapt to such changes in distribution.

Test-time adaptation (TTA) [13, 37, 40] describes a setting where a model adapts to changes in distribution at test-time, at the cost of additional computation. In recent years, a variety of methods based on TTA have been proposed, focusing on few-shot adaptation [32] where the network is given access to a few labelled examples, or unsupervised domain adaptation (UDA) [43] where the network is given access to many *unlabelled* examples from the new distribution. Of particular relevance is a specific UDA setting where model parameters are adapted *independently* to each unlabelled example in the test-set. This setting has been previously referred to as single-example UDA, and here we also refer to it as *slow inference* since it is similar to a human taking more time to parse a difficult example. Existing approaches for this setting typically devise a self-supervised loss that aligns well with the task of image classification and then optimize this loss during test-time adaptation [37, 11, 1, 20]. However, despite their success for image classification, these approaches do not provide adequate support for other scene understanding tasks, and in particular scene segmentation, as we showcase in Section 3.1.One potentially important aspect to supporting TTA for other scene understanding tasks is the inductive bias of the underlying architecture. In the context of segmentation, there has been a lot of recent development in building models that segment scenes into entities in an unsupervised way by optimizing a reconstruction objective [10, 18, 38, 15, 8, 28, 44]. These methods differ in details but share the notion of incorporating a fixed set of entities, also known as *slots* or *object files*. Each slot extracts information about a single entity during encoding, and is "synthesized" back to the input domain during decoding. Their ability to distinguish visual objects at a representation level makes them a particularly promising candidate for TTA for segmentation tasks.

In light of the above, we propose Slot-centric Test-time adaptation (Slot-TTA), a semi-supervised approach that combines Slot Attention [28] (in the 2D image) or Object Scene Representation Transformer [34] (in multi-view image setting) with a supervised segmentation loss to enable it to leverage instance-level image. Slot-TTA is trained jointly to synthesize and segment scenes. At test time, the model adapts without supervision to a single test sample by optimizing the self-supervised objective alone. Different from fully-unsupervised object-centric generative models, Slot-TTA uses annotations at training time to help it develop the notion of what an object is, which lets it scale to more complex visual settings. Different from existing TTA methods, Slot-TTA uses a slot-centric architecture and self-supervised synthesis loss that better aligns with the task of segmentation. Different from state-of-the-art detectors, Slot-TTA is equipped with reconstruction feedback that allows it to adapt at test time without supervision, i.e. without using additional annotated data. Indeed, we show that test-time adaptation via image synthesis in Slot-TTA enables successfully parsing completely unfamiliar scenes composed of familiar entities.

We test Slot-TTA's instance segmentation ability on the two datasets: MultiShapeNet-Hard [35] and Multi-Shape. We evaluate Slot-TTA's ability to parse out-of-distribution scenes and compare it against state-of-the-art entity-centric generative models [34] and supervised visual detectors [6] trained with labeled data to segment objects. We show improvements over all baselines in Slot-TTA ability to segment novel scenes. Additionally, we ablate different design choices of Slot-TTA.

2 Method

We consider Slot-TTA in two settings: (i) auto-encoding of images, and (ii) novel view synthesis. For (i), we use the architecture proposed by Slot Attention [28], where a 2D broadcast decoder [41] is used to render the input view. For (ii), we use OSRT's architecture which combines the Slot Attention bottleneck with the geometry-free backbone of SRT [35] to perform object-centric novel view synthesis.

Training for joint segmentation and reconstruction We train all the parameters of our model to jointly optimize image reconstruction or novel view image synthesis objectives and the task segmentation objective over all the n examples in the training set, where x represents the input scene and y the segmentation labels:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \lambda_s l_{seg}(x_i, y_i; \theta) + \lambda_r l_{recon}(x_i; \theta) \tag{1}$$

For reconstruction, we minimize the mean squared error between the predicted and ground truth RGB images. For segmentation, we supervise the alpha masks a_i of each slot as provided by the decoders. We use Hungarian matching [27] to associate the ground truth masks with the predicted masks, and upon association we apply a categorical cross-entropy loss l_{seg} . We weight the segmentation and reconstruction loss by λ_s and λ_r.

Test-time adaptation In this work, we refer to a single forward pass through our trained model without any test-time adaptation as *fast inference* (same as regular inference). We call the process of test-time adapting the model on each example independently *slow inference*, using only the reconstruction objective of Eq. 1. We use this terminology to emphasize that the only difference between both settings is the added computation time which results in an effective speed difference between the two inference schemes. We adapt only the encoder parameters θ_{enc} in our model, which we found to improve results compared to adapting the entire model as shown in our supplementary Section D.1. We train for 150 steps per example using the Adam optimizer [25]. For further details on our model architectures and model figure please refer to supplementary Section B

3 Experiments

We test Slot-TTA capability for segmenting posed multi-view and single-view RGB images. We use Adjusted Random Index (ARI) as our evaluation metric for segmentation accuracy [31]. Our experiments aim to answer the following questions: (i) How does Slot-TTA compare against state-of-the-art 2D segmentation models [6]? (ii) How does slow inference through reconstruction feedback affect segmentation accuracy in Slot-TTA and its variants? (iii) How much does supervision during training contribute to segmentation performance?

3.1 Segmenting RGB images in multi-view scenes

Dataset We evaluate Slot-TTA on the MultiShapeNet (MSN) dataset from SRT [35]. The dataset is constructed by rendering 51K ShapeNet objects using Kubric [16] against 382 HDR backgrounds so that there is no overlap of objects between the train and test sets. Further train and tests sets differ in the number of objects present: scenes with 5-7 object instances are in the training set and scenes with 16-30 objects are in the test set. For more details please refer to supplementary Section A.

Method	**in-dist (5-7 instances)**		**out-of-dist (16-30 instances)**	
	Fast Infer.	Slow Infer.	Fast Infer.	Slow Infer.
Slot-TTA-w/o supervision	0.32	0.30	0.33	0.29
Mask2Former	**0.93**	N/A	0.74	N/A
Mask2Former+BYOL	0.93	0.**95**	**0.75**	0.74
Mask2Former+Recon	0.93	0.92	0.74	0.67
Slot-TTA (Ours)	0.92	**0.95**	0.70	**0.83**

Table 1: **Instance Segmentation accuracy** (higher is better) in the multi-view RGB setup for in-distribution test set of 5-7 object instances and out-of-distribution 16-30 object instances.

Baselines We compare to three baselines: (i) Mask2Former [6], a state-of-the-art 2D image segmentor which adapts detection transformers [4] to image segmentation by using multiscale segmentation decoders with masked attention. (ii) Mask2Former+BYOL which combines the segmentation model [6] with test time adaptation using BYOL self-supervised loss [1]. (iii) Mask2Former+Recon which combines the segmentation model [6] with rendering submodules and image reconstruction loss for test-time adaptation.

Results We show quantitative segmentation results of our model and baselines on target camera viewpoints in Table 1 and qualitative TTA results in Figure 2. In Slot-TTA-w/o supervision, instead of training jointly for reconstruction and segmentation, we train using only cross-view image synthesis, similar to OSRT [34].

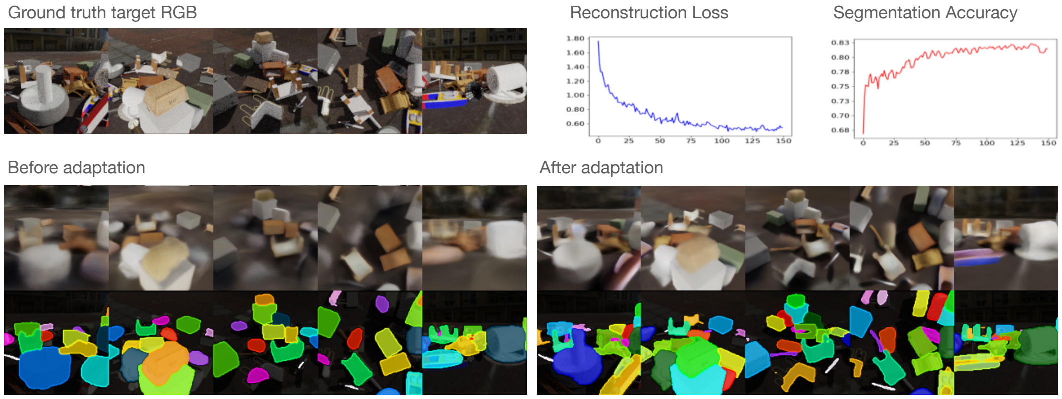

Figure 2: **Test-time adaptation via slow inference in Slot-TTA for multi-view scenes**. In the right top we visualize the RGB loss (blue curve) and the segmentation ARI accuracy (red curve). As can be seen, during slow inference the segmentation accuracy improves as reconstruction loss reduces.

It can be observed that: (i) Slot-TTA-Slow outperforms the feedforward Mask2Former-Fast, especially for out-of-distribution scenes; (ii) adding self-supervised losses of SOTA image classification methods [1] to Mask2Former (eg. Mask2Former+BYOL) does not suffice to adapt them effectively at test time and (iii) Slot-TTA without supervision, which is identical to OSRT [34] is not competitive with supervised models for object segmentation.[1]

3.2 Segmenting single-view RGB images

As a proof of concept, in this section, we test our model and the baseline Mask2Former in segmenting single RGB images comprised of multiple samples from five shapes of distinct colors, organized in heavily occluded configurations, a dataset we create and we call Multi-Shape. Our training set consists of images with 3-5 object instances, while the test set consists of images with 10-16 object instances. For this setting, we report the ARI scores for the foreground objects only, since in this dataset the background occupies a large image area and a method that assigns most pixels to background already achieves a very high ARI. We find the performance accuracy ordering of the methods to be the same. As can be seen in Table 2, before TTA Mask2former and Mask2former+Recon outperform our method. After TTA, our method significantly outperforms the baselines.

Method	**in-dist (3-5 instances)**		**out-of-dist(10-16 instances)**	
	Fast Infer.	Slow Infer.	Fast Infer.	Slow Infer.
Mask2Former	**0.96**	N/A	**0.44**	N/A
Mask2Former+Recon	0.95	0.94	0.43	0.47
Slot-TTA (Ours)	**0.96**	**0.95**	0.39	**0.69**

Table 2: **Foreground instance segmentation accuracy** (higher is better) for single-view RGB images. In-distribution images have 3-5 objects and out-of-distribution images have 10-16 objects.

Please refer to Section D.1 and Section D.2 in the supplementary for qualitative results and ablations in our mult-view RGB and single-view RGB settings respectively.

4 Conclusion

In this work we show that the architectural choices found in unsupervised object discovery methods such as Slot Attention, could be very helpful in test-time adaptation. Additionally we show sufficient evidence in our work that a future version of Slot-TTA could potentially compete with state-of-the-art segmentation methods when allowed to do test-time adaptation.

[1] Although OSRT performs poorly in the ARI metric, it achieves substantially better results in terms of foreground-ARI (yet still not competitive). This is because it is unable to segment out the background.

References

[1] Bartler, A., Bühler, A., Wiewel, F., Döbler, M., Yang, B.: Mt3: Meta test-time training for self-supervised test-time adaption. In: International Conference on Artificial Intelligence and Statistics. pp. 3080–3090. PMLR (2022)

[2] Bateson, M., Lombaert, H., Ben Ayed, I.: Test-time adaptation with shape moments for image segmentation. In: MICCAI. pp. 736–745. Springer (2022)

[3] Burgess, C.P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., Lerchner, A.: Monet: Unsupervised scene decomposition and representation. arXiv preprint arXiv:1901.11390 (2019)

[4] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)

[5] Chang, W.G., You, T., Seo, S., Kwak, S., Han, B.: Domain-specific batch normalization for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 7354–7362 (2019)

[6] Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. CoRR **abs/2112.01527** (2021), `https://arxiv.org/abs/2112.01527`

[7] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)

[8] Du, Y., Smith, K., Ulman, T., Tenenbaum, J., Wu, J.: Unsupervised discovery of 3d physical objects from video. arXiv preprint arXiv:2007.12348 (2020)

[9] Engelcke, M., Kosiorek, A.R., Jones, O.P., Posner, I.: Genesis: Generative scene inference and sampling with object-centric latent representations. arXiv preprint arXiv:1907.13052 (2019)

[10] Eslami, S., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., Hinton, G.E.: Attend, infer, repeat: Fast scene understanding with generative models. arXiv preprint arXiv:1603.08575 (2016)

[11] Gandelsman, Y., Sun, Y., Chen, X., Efros, A.A.: Test-time training with masked autoencoders. arXiv preprint arXiv:2209.07522 (2022)

[12] Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020)

[13] Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D., Li, W.: Deep reconstruction-classification networks for unsupervised domain adaptation. In: European conference on computer vision. pp. 597–613. Springer (2016)

[14] Goyal, A., Lamb, A., Gampa, P., Beaudoin, P., Levine, S., Blundell, C., Bengio, Y., Mozer, M.: Object files and schemata: Factorizing declarative and procedural knowledge in dynamical systems. arXiv preprint arXiv:2006.16225 (2020)

[15] Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., Schölkopf, B.: Recurrent independent mechanisms. In: ICLR (2021)

[16] Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D.J., Gnanapragasam, D., Golemo, F., Herrmann, C., et al.: Kubric: A scalable dataset generator. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3749–3761 (2022)

[17] Greff, K., Kaufman, R.L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., Lerchner, A.: Multi-object representation learning with iterative variational inference. In: International Conference on Machine Learning. pp. 2424–2433. PMLR (2019)

[18] Greff, K., Rasmus, A., Berglund, M., Hao, T.H., Schmidhuber, J., Valpola, H.: Tagger: Deep unsupervised perceptual grouping. arXiv preprint arXiv:1606.06724 (2016)

[19] Greff, K., Van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. arXiv preprint arXiv:2012.05208 (2020)

[20] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)

[21] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

[22] Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8349 (2021)

[23] Ke, N.R., Didolkar, A., Mittal, S., Goyal, A., Lajoie, G., Bauer, S., Rezende, D., Bengio, Y., Mozer, M., Pal, C.: Systematic evaluation of causal discovery in visual model based reinforcement learning. arXiv preprint arXiv:2107.00848 (2021)

[24] Khurana, A., Paul, S., Rai, P., Biswas, S., Aggarwal, G.: Sita: Single image test-time adaptation. arXiv preprint arXiv:2112.02355 (2021)

[25] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[26] Kosiorek, A., Kim, H., Teh, Y.W., Posner, I.: Sequential attend, infer, repeat: Generative modelling of moving objects. Advances in Neural Information Processing Systems **31**, 8606–8616 (2018)

[27] Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)

[28] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. Advances in Neural Information Processing Systems **33**, 11525–11538 (2020)

[29] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)

[30] Rahaman, N., Goyal, A., Gondal, M.W., Wuthrich, M., Bauer, S., Sharma, Y., Bengio, Y., Schölkopf, B.: S2rms: Spatially structured recurrent modules. arXiv preprint arXiv:2007.06533 (2020)

[31] Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association **66**(336), 846–850 (1971)

[32] Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. arXiv preprint arXiv:1803.00676 (2018)

[33] Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. arXiv preprint arXiv:1710.09829 (2017)

[34] Sajjadi, M.S., Duckworth, D., Mahendran, A., van Steenkiste, S., Pavetić, F., Lučić, M., Guibas, L.J., Greff, K., Kipf, T.: Object scene representation transformer. arXiv preprint arXiv:2206.06922 (2022)

[35] Sajjadi, M.S., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lučić, M., Duckworth, D., Dosovitskiy, A., et al.: Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6229–6238 (2022)

[36] Shin, I., Tsai, Y.H., Zhuang, B., Schulter, S., Liu, B., Garg, S., Kweon, I.S., Yoon, K.J.: Mm-tta: Multi-modal test-time adaptation for 3d semantic segmentation. In: CVPR (2022)

[37] Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: International conference on machine learning. pp. 9229–9248. PMLR (2020)

[38] Van Steenkiste, S., Chang, M., Greff, K., Schmidhuber, J.: Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. arXiv preprint arXiv:1802.10353 (2018)

[39] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[40] Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726 (2020)

[41] Watters, N., Matthey, L., Burgess, C.P., Lerchner, A.: Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. arXiv preprint arXiv:1901.07017 (2019)

[42] Zablotskaia, P., Dominici, E.A., Sigal, L., Lehrmann, A.M.: Unsupervised video decomposition using spatio-temporal iterative inference. arXiv preprint arXiv:2006.14727 (2020)

[43] Zhang, Y.: A survey of unsupervised domain adaptation for visual recognition. arXiv preprint arXiv:2112.06745 (2021)

[44] Zoran, D., Kabra, R., Lerchner, A., Rezende, D.J.: Parts: Unsupervised segmentation with slots, attention and independence maximization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10439–10447 (2021)

A Datasets

A.1 Multi-view RGB

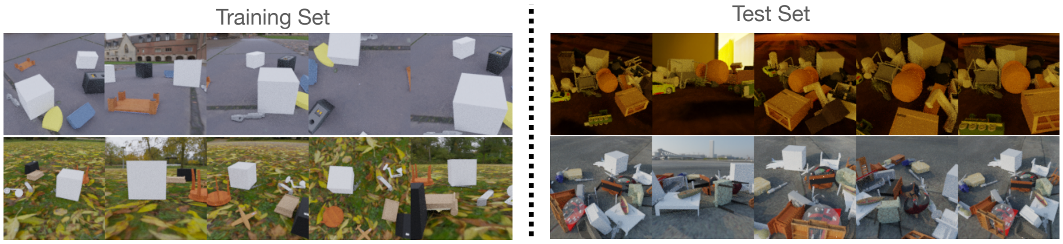

Figure 3: We visualize samples from the train-test split used by us in experiment Section 3.1. Different rows correspond to different scenes and different columns correspond to different viewpoints.

We use the MultiShapeNet-Hard dataset of Scene Representation Transformer, a complex photo-realistic dataset for Novel View Synthesis [35]. Our train split consists of 5-7 ShapeNet objects placed at random locations and orientations in the scene. The backgrounds are sampled from 382 realistic HDR environment maps. Our test set consists of 16-30 objects placed at novel arrangements. We sample objects from a pool of 51K ShapeNet objects across all categories, we divide the pool into train and test such that the test set consists of objects not seen during training. The train split has 200K scenes, and the test set consists of 4000 scenes, each with 10 views. We had to regenerate the dataset for this specific train-test split.

A.2 Single-view RGB

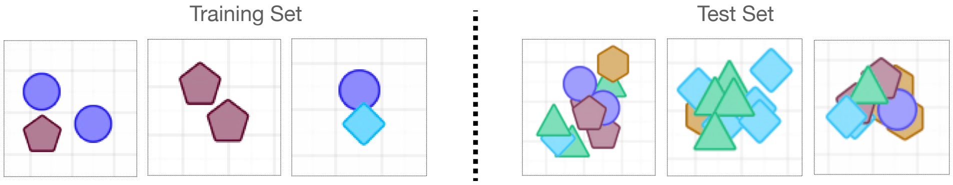

Figure 4: We visualize the samples of our MultiShape dataset.

Multi-Shape is a dataset built by us for proof-of-concept. It consists of 5 shapes of distinct colors uniformly placed at a random location in a 2D canvas. Our training set consist of 3-5 object instances, while the test set consists of a highly occluded setting with 10-16 object instances.

B Method

Slot Attention Current state-of-the-art detectors and segmentors instantiate slots (i.e. the query vectors) from 2D visual feature maps [4]. Most works use iterative cross-attention (features to slots) and self-attention (slot-to-slots) operations [4] to map a set of N input feature vectors to a set of K slot vectors. Attention-based competition amongst slots and iterative routing popularized in [15, 28] encourages a single location in the input to be assigned to a unique slot vector.

Given a visual scene encoded as a set of feature vectors $\mathrm{M} \in \mathbb{R}^{N \times \mathrm{C}}$ and K randomly initialized slots sampled from a multivariate Gaussian distribution with a diagonal covariance $S \sim \mathcal{N}(\mu, \mathrm{Diag}(\sigma^2)) \in \mathbb{R}^{K \times \mathrm{D}}$, where $\mu, \sigma \in \mathbb{R}^C$ are learnable parameters of the Gaussian, Slot Attention [28] computes an attention map a between the feature map M and the slots S:

$$a = \mathrm{Softmax}(\mathrm{k}(\mathrm{M}) \cdot \mathrm{q}(S)^T, \text{axis=``slots''}) \in \mathbb{R}^{N \times K}. \tag{2}$$

k, q, and v are learnable linear transformations that map inputs and slots to a common dimension D. The softmax normalization over slots ensures competition amongst them to attend to a specific feature vector in M. Updates to the slots are computed based on the input features they attend to:

$$updates = a^T v(M) \in \mathbb{R}^{K \times C}, \text{where } a_{i,k} = \frac{a_{i,k}}{\sum_{i=0}^{N-1} a_{i,k}} \tag{3}$$

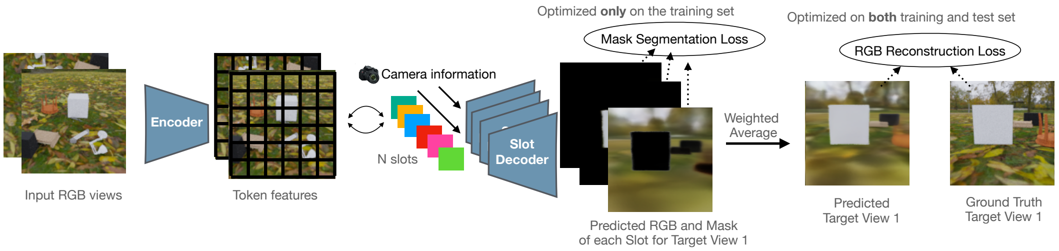
Optimized **only** on the training set
Mask Segmentation Loss
Optimized on **both** training and test set
RGB Reconstruction Loss
Camera information
Encoder
N slots
Slot Decoder
Weighted Average
Input RGB views
Token features
Predicted RGB and Mask of each Slot for Target View 1
Predicted Target View 1
Ground Truth Target View 1

Figure 5: **Model architecture for multi-view images**. Given multi-view RGB images as input. Slot-TTA (here using OSRT [34] as a backbone) maps them to a set of token features, which are then mapped to a set of slot vectors. Conditioned on the camera-viewpoint Slot-TTA then decodes each slot into its respective segmentation mask and RGB image. It then uses weighted averaging to render the RGB image for the whole scene as seen from the camera viewpoint. On the training dataset, we jointly optimize using reconstruction and segmentation loss. On the test set, we optimize only using the reconstruction loss. We use a similar training pipeline for other input modalities.

which are then fed into a GRU [7]: $S = \mathrm{GRU}(\text{state} = S, \text{input} = updates)$. We iterate 3 times over equations 2 and 3. For detailed description, please refer to [28].

We first describe the encoders and decoders that form the foundation of Slot-TTA for each modality. Further we detail how we train Slot-TTA and perform test time adaptation through slow inference.

B.0.1 Encoding and Decoding Backbones

Posed multi-view 2D RGB images As shown in Figure 5, Slot-TTA builds upon the architecture of OSRT [34], which is an object-centric, geometry-free novel view synthesis method. Given a set of multi-view RGB images as input, a CNN encodes each input image I_i into a feature grid, which is then flattened into a set of tokens with camera pose and ray direction information added in each of the tokens, similar to SRT [35]. These are then encoded into a set of latent features using a transformer [39] Enc with multiple self-attention blocks $z = \mathrm{Enc}(\mathrm{CNN}(\mathrm{I}_i))$. The latent features z are then mapped into a set of slots S using Slot Attention (Section B). For decoding, we adopt the spatial broadcast decoder [41] formulation, where a render MLP takes as input the slot vector S_k and the pixel location p parameterized by the camera position and the ray direction pointing to the pixel to be decoded. It outputs an RGB color c_k and an unnormalized alpha score a_k for each pixel location $c_k, a_k = \mathrm{Dec}(p, S_k)$. The a_k's are normalized using a Softmax and used as weights to aggregate the predicted RGB values c_k for each slot.

Single-view 2D RGB images For this setting, Slot-TTA uses a ResNet-18 [21] to encode the input RGB image into a feature grid. We then add positional vectors to the feature grid and map to a set of slot vectors using Slot Attention. Similar to the multi-view setting, each slot vector is decoded to the RGB image and an alpha mask using an MLP renderer. We parameterize pixel location p as (x, y) points on the grid instead of camera position as the above setting.

C Related Work

Entity-centric generative models for scene decomposition *Entity-centric* (or *object-centric*) models use architectural inductive biases to represent perceptual inputs, such as an observation of a visual scene, in terms of separate object variables, often referred to as *slots* or *object files* [19, 33, 26, 9, 14, 23, 3, 17, 42, 30]. Prominent examples of such models include MONet [3], GENESIS [9], IODINE [17], and Slot Attention (SA) [28], which are trained in a fully-unsupervised setting via a simple auto-encoding objective. Object representations and scene decomposition emerge via the inductive bias of the model architecture (and in some cases, additional regularizers). However, without any form of supervision, scene decompositions can be ambiguous, which is particularly challenging for complex real-world scenes or in the presence of complicated textures. In Slot-TTA, we aid the competition mechanism in SA to address this issue by jointly training with a supervised segmentation loss. OSRT [34] is a cross-view geometry-free encoder-decoder method, that segments an image into objects through reconstructing novel viewpoints. OSRT combines SA with SRT [35], a view synthesis model that uses transformer encoder and decoders to fuse information across views, as well as the camera pose, without any explicit 3D information. Our multi-view RGB Slot-TTA builds upon their architecture.

Test-time adaptation In test-time adaptation, model parameters are updated at test-time to better generalize to the distribution shift. In recent years, there has been significant development in this direction. Methods such as pseudo labelling and entropy minimization [36, 40, 2] have demonstrated that supervising the model using its own confident predictions could help improve its accuracy. Adaptive BatchNorm methods [24, 5] have shown that updating the BatchNorm parameters using the new examples can help adaptation. Despite these successes, these methods by definition are data inefficient as they require confident predictions or a batch of examples to adapt. Self-supervised learning (SSL) [37, 1, 11] based methods on the other hand, have empirically shown to be data efficient. During training, they jointly train using the task and SSL loss, and during test-time, they train only using the SSL loss. All of the methods in the SSL setting thus far focus on the task of classification and mainly differ in terms of the SSL loss used. For example TTT [37] uses rotation angle prediction as their SSL loss, MT3 [1] uses a BYOL [20] loss and TTT-MAE [11] uses Masked autoencoding loss [29]. In our work, we show that these losses do not generalize to segmentation, and how we might need specific architectural biases to close the gap.

D Additional Experiments

D.1 Segmenting RGB images in multi-view scenes

Method	**in-dist (5-7 instances)**		**out-of-dist (16-30 instances)**	
	Fast Infer.	Slow Infer.	Fast Infer.	Slow Infer.
Slot-TTA-SlotMixer_Decoder	**0.94**	0.89	0.65	0.72
Slot-TTA-SRT_Decoder	0.92	0.88	0.60	0.63
Slot-TTA-tta_All_param	N/A	0.92	N/A	0.82
Slot-TTA-tta_Norm_param	N/A	0.94	N/A	0.79
Slot-TTA-tta_Slot_param	N/A	0.94	N/A	0.76
Slot-TTA w/o Weighted_Sample	N/A	0.93	N/A	0.81
Slot-TTA (Ours)	0.92	**0.95**	**0.70**	**0.83**

Table 3: **ARI Segmentation accuracy (higher is better)** in the in-distribution test set of 5-7 object instances and out-of-distribution 16-30 object instances.

We conduct various ablations of Slot-TTA in Table 3. In Figure 6, we show additional qualitative results comparing Slot-TTA-Fast and Slot-TTA-Slow.

(i) We ablate different decoder choices in the topmost section where instead of using the broadcast decoder we use the Scene representation transformer (SRT) decoder [35] which we refer to as **Slot-TTA-SRT_Decoder** or the SlotMixer decoder [34], referred to as **Slot-TTA-SlotMixer_Decoder**.

(ii) We ablate what parameters to adapt at test time. As it's unclear since TENT [40] optimizes BatchNorm or LayerNorm parameters, but TTT [37] optimizes the shared parameters between the SSL and the task-specific branch, which in our case will be all the parameters in the network. In Table 3, **Slot-TTA-tta_All_param** is when we adapt all the network parameters, **Slot-TTA-tta_Norm_param** adapts only the Layer or BatchNorm parameters and **Slot-TTA-tta_Slot_param** adapts only the learnable slot embeddings. We find that optimizing only the encoder parameters works the best for our setting.

(iii) Further, we ablate error-conditioned pixel sampling where **Slot-TTA w/o Weighted_Sample** refers to our model that uses uniform sampling instead of the error weighted sampling.

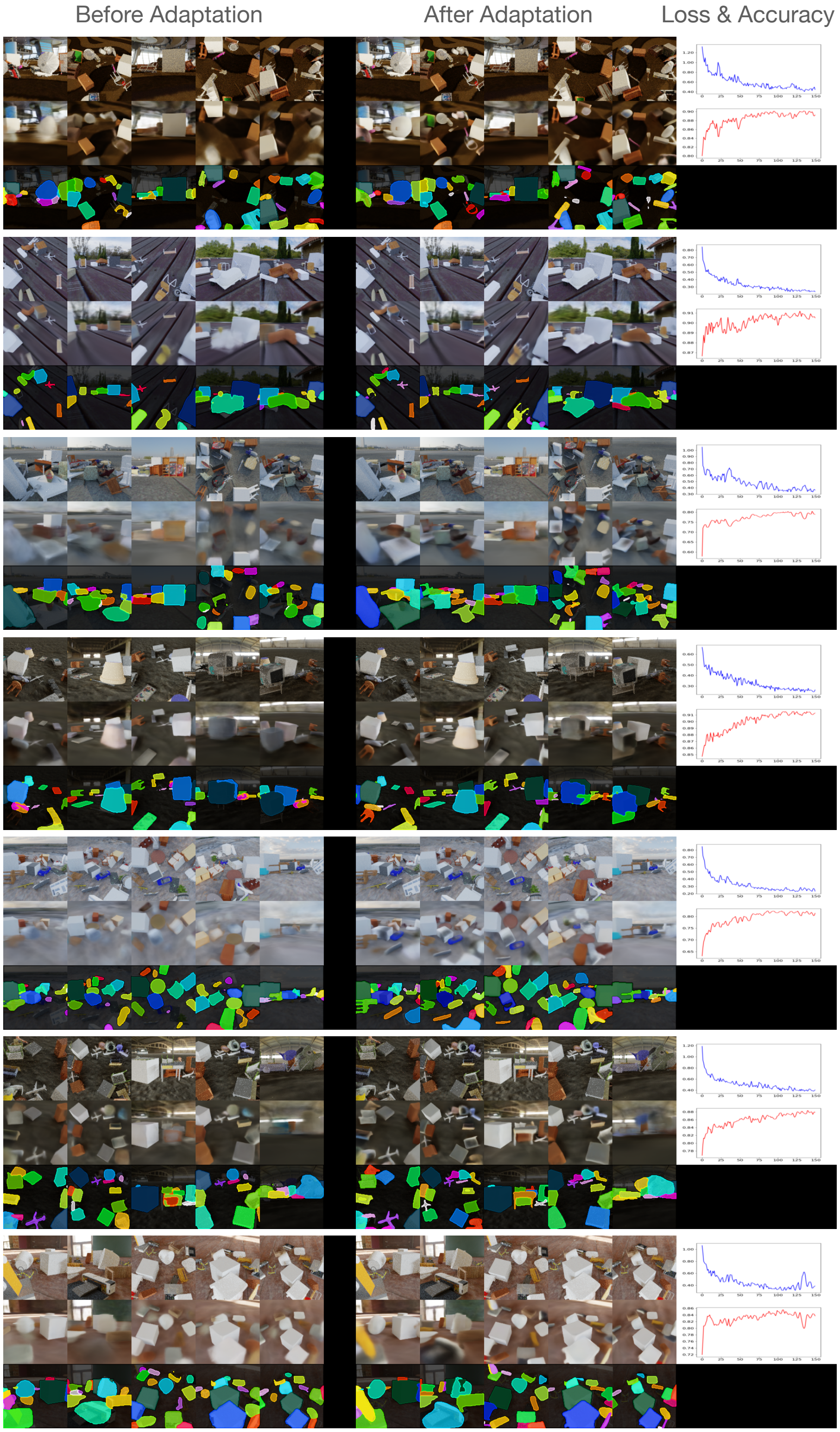

Figure 6: On the left, we visualize Slot-TTA-Fast. In the middle, we visualize Slot-TTA-Slow. In the first row we visualize the ground truth target RGB views. In the second and third row we visualize Slot-TTA predicted target RGB views and their segmentation masks. On the right-most column we visualize the RGB loss and segmentation accuracy when doing slow inference.

D.2 Segmenting single-view RGB images

In Figure 7 we qualitatively compare Slot-TTA-Slow with Slot-TTA-Fast. We show that slow inference can help discover objects missed by Slot-TTA. We also show some failure cases where slow inference could override the object-centric bottleneck to achieve higher reconstruction accuracy.

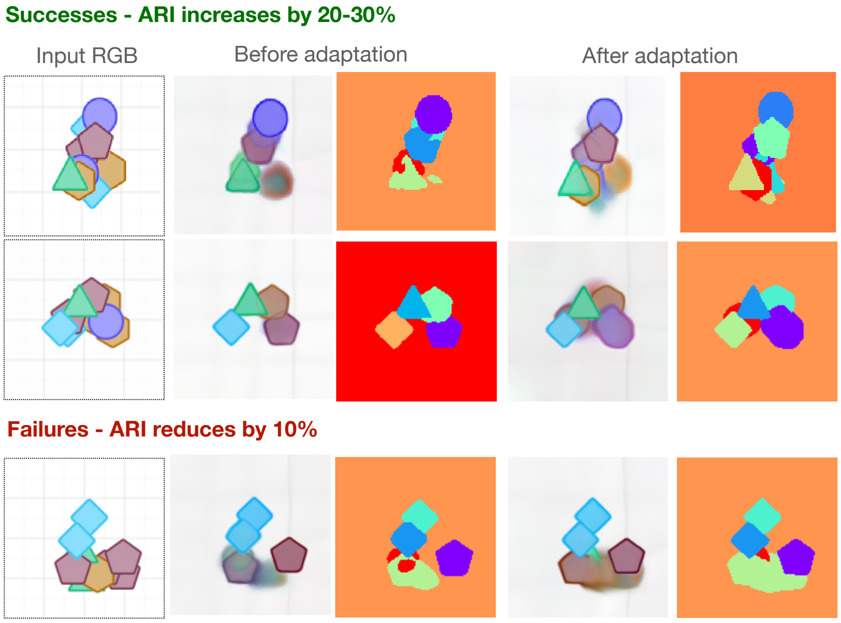
Successes - ARI increases by 20-30%
Input RGB
Before adaptation
After adaptation
Failures - ARI reduces by 10%

Figure 7: Success and Failure cases of slow-inference on Multi-Shape dataset. Same setting as Section 3.2