Appendix

Anomalies-by-Synthesis: Anomaly Detection using Generative Diffusion Models for Off-Road Navigation

Sunshine Jiang*[1], Siddharth Ancha*[1], Travis Manderson[1], Laura Brandt[1],
Yilun Du[1], Philip R. Osteen[2] and Nicholas Roy[1]

Website: https://bit.ly/anomalies-by-synthesis†

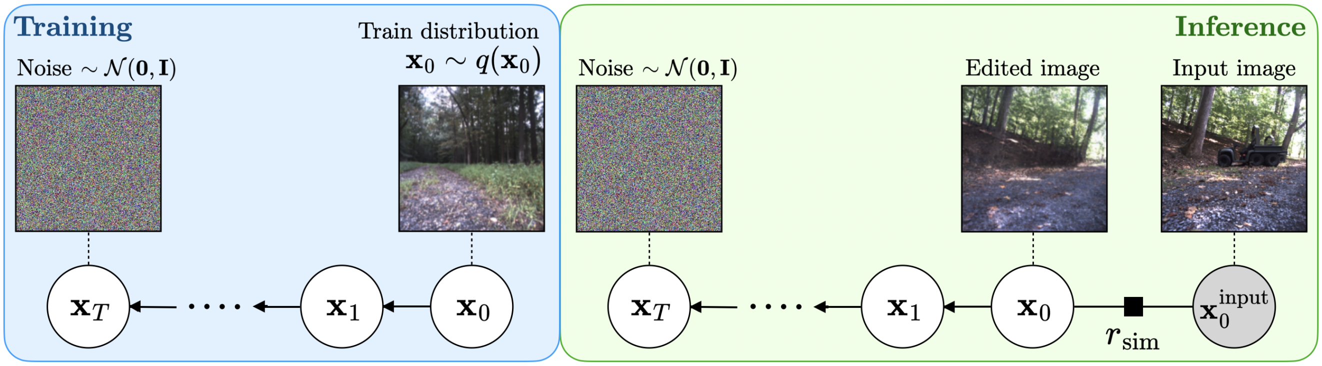

Fig. 4: Probabilistic graphical model of the *conditional* forward diffusion process. We wish to sample the random variable $\mathbf{x}_0$ corresponding to the training data distribution $q(\mathbf{x}_0)$. The *directed* edges between $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ (for $t = 1, \ldots, T$) correspond to the vanilla forward diffusion process. Each directed edge denotes the sampling distribution $q(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})$ which successively adds a small amounts of Gaussian noise: $q(\mathbf{x}_t \,|\, \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_t \,;\, \sqrt{1-\beta_t}\, \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$ [32, 38]. However, we are interested in sampling from $q(\mathbf{x}_0 \,|\, \mathbf{x}_0^{\text{input}}) = q(\mathbf{x}_0), r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})$. This objective corresponds to adding an additional *undirected factor* $r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})$ between $\mathbf{x}_0$ and $\mathbf{x}_0^{\text{input}}$; $\mathbf{x}_0^{\text{input}}$ is treated as constant. Our task is to perform inference over this graphical model and sample from $q(\mathbf{x}_0 \,|\, \mathbf{x}_0^{\text{input}})$ using a diffusion model that was trained to perform reverse diffusion in the absence of the r_{sim} factor.

Appendix I
Proof: Generalized Conditional Guidance Gradient

Classifier guidance is not only restricted to classifiers, it also requires training a classifier $p(y \,|\, \mathbf{x}_t)$ for each intermediate latent state [34, 39]. First, we extend classifier guidance to the more general setting where the conditioner is any non-negative function:

Theorem 1: When a diffusion model $\boldsymbol{\epsilon}_t^{\boldsymbol{\theta}}$ is trained to sample from $q(\mathbf{x}_0)$, the conditional distribution $q(\mathbf{x}_0 \,|\, \mathbf{x}_0^{\text{input}}) \propto q(\mathbf{x}_0)\, r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})$ can be sampled by using the following guidance gradient during reverse diffusion:

$$\begin{aligned}\mathbf{g}_t(\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \log \int_{\mathbf{x}_0} q(\mathbf{x}_0 \,|\, \mathbf{x}_t)\, r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})\, \mathrm{d}\mathbf{x}_0 \\ &= \nabla_{\mathbf{x}_t} \log \mathbb{E}_{q(\mathbf{x}_0 \,|\, \mathbf{x}_t)}\left[r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}}) \right] \end{aligned} \tag{4}$$

A. Proof using the variational inference perspective

$$q(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{x}_0^{\text{input}}) \propto q(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{x}_0^{\text{input}})$$

$$= \int_{\mathbf{x}_0} q(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_0 \,|\, \mathbf{x}_0^{\text{input}})\, \mathrm{d}\mathbf{x}_0$$

$$= \int_{\mathbf{x}_0} q(\mathbf{x}_0 \,|\, \mathbf{x}_0^{\text{input}})\, q(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{x}_0, \mathbf{x}_0^{\text{input}})\, \mathrm{d}\mathbf{x}_0$$

$$= \int_{\mathbf{x}_0} q(\mathbf{x}_0 \,|\, \mathbf{x}_0^{\text{input}})\, \underbrace{q(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{x}_0)}\, \mathrm{d}\mathbf{x}_0$$

(since $\mathbf{x}_{1:T}$ is independent of $\mathbf{x}_0^{\text{input}}$ conditioned on $\mathbf{x}_0$)

$$\propto \int_{\mathbf{x}_0} \underbrace{q(\mathbf{x}_0)\, r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})}\, q(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0$$

(by the definition of $q(\mathbf{x}_0 \,|\, \mathbf{x}_0^{\text{input}}) \propto q(\mathbf{x}_0)\, r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})$)

$$= \int_{\mathbf{x}_0} r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})\, q(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_0)\, \mathrm{d}\mathbf{x}_0$$

$$= \int_{\mathbf{x}_0} r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})\, q(\mathbf{x}_{t+1})\, q(\mathbf{x}_t \,|\, \mathbf{x}_{t+1})\, q(\mathbf{x}_0 \,|\, \mathbf{x}_t, \mathbf{x}_{t+1})\, \mathrm{d}\mathbf{x}_0$$

$$= \int_{\mathbf{x}_0} r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})\, q(\mathbf{x}_{t+1})\, q(\mathbf{x}_t \,|\, \mathbf{x}_{t+1})\, \underbrace{q(\mathbf{x}_0 \,|\, \mathbf{x}_t)}\, \mathrm{d}\mathbf{x}_0$$

(since $\mathbf{x}_0$ and $\mathbf{x}_{t+1}$ are independent conditioned on $\mathbf{x}_t$)

$$\propto q(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}) \int_{\mathbf{x}_0} q(\mathbf{x}_0 \,|\, \mathbf{x}_t)\, r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})\, \mathrm{d}\mathbf{x}_0$$

$$= q(\mathbf{x}_t \,|\, \mathbf{x}_{t+1})\, \mathbb{E}_{q(\mathbf{x}_0 \,|\, \mathbf{x}_t)}\left[r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}}) \right]$$

Therefore, following the same analysis as the first-order Gaussian approximation, the guidance gradient is $\nabla_{\mathbf{x}_t} \log \mathbb{E}_{q(\mathbf{x}_0 \mid \mathbf{x}_t)} \left[r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}}) \right]$ evaluated at $\mathbf{x}_t = \boldsymbol{\mu}_{t+1}(\mathbf{x}_{t+1})$.

B. Proof using the score functions perspective

The score function of intermediate states $\mathbf{x}_t$ of the vanilla forward diffusion process is defined as $\mathbf{s}(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$. However, we're interested in the score function of the *conditional* forward diffusion process $\mathbf{s}(\mathbf{x}_t \mid \mathbf{x}_0^{\text{input}}) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_0^{\text{input}})$. The additional term that needs to be added to $\mathbf{s}(\mathbf{x}_t)$ to obtain $\mathbf{s}(\mathbf{x}_t \mid \mathbf{x}_0^{\text{input}})$ is the guidance gradient. Therefore, we now derive $\mathbf{s}(\mathbf{x}_t \mid \mathbf{x}_0^{\text{input}})$ in terms of $\mathbf{s}(\mathbf{x}_t)$:

$$
\begin{aligned}
\underbrace{\mathbf{s}(\mathbf{x}_t \mid \mathbf{x}_0^{\text{input}})}_{\text{conditional score function}} &= \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_0^{\text{input}}) \\
&= \nabla_{\mathbf{x}_t} \log \int_{\mathbf{x}_0} q(\mathbf{x}_t, \mathbf{x}_0 \mid \mathbf{x}_0^{\text{input}}) \, d\mathbf{x}_0 \\
&= \nabla_{\mathbf{x}_t} \log \int_{\mathbf{x}_0} q(\mathbf{x}_0 \mid \mathbf{x}_0^{\text{input}}) \, q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_0^{\text{input}}) \, d\mathbf{x}_0 \\
&= \nabla_{\mathbf{x}_t} \log \int_{\mathbf{x}_0} q(\mathbf{x}_0 \mid \mathbf{x}_0^{\text{input}}) \, \underbrace{q(\mathbf{x}_t \mid \mathbf{x}_0)}_{} \, d\mathbf{x}_0 \\
&\quad \textit{(since } \mathbf{x}_{1:T} \textit{ is independent of } \mathbf{x}_0^{\text{input}} \textit{ conditioned on } \mathbf{x}_0\textit{)} \\
&= \nabla_{\mathbf{x}_t} \log \int_{\mathbf{x}_0} \underbrace{q(\mathbf{x}_0) \, r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})}_{} \, q(\mathbf{x}_t \mid \mathbf{x}_0) \, d\mathbf{x}_0 \\
&\quad \textit{(by the definition of } q(\mathbf{x}_0 \mid \mathbf{x}_0^{\text{input}}) \propto q(\mathbf{x}_0) \, r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}})\textit{)} \\
&= \nabla_{\mathbf{x}_t} \log \int_{\mathbf{x}_0} r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}}) \, q(\mathbf{x}_t, \mathbf{x}_0) \, d\mathbf{x}_0 \\
&= \nabla_{\mathbf{x}_t} \log \int_{\mathbf{x}_0} r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}}) \, q(\mathbf{x}_t) \, q(\mathbf{x}_0 \mid \mathbf{x}_t) \, d\mathbf{x}_0 \\
&= \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log \int_{\mathbf{x}_0} q(\mathbf{x}_0 \mid \mathbf{x}_t) \, r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}}) \, d\mathbf{x}_0 \\
&= \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log \mathbb{E}_{q(\mathbf{x}_0 \mid \mathbf{x}_t)} \left[r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}}) \right] \\
&= \mathbf{s}(\mathbf{x}_t) + \underbrace{\nabla_{\mathbf{x}_t} \log \mathbb{E}_{q(\mathbf{x}_0 \mid \mathbf{x}_t)} \left[r_{\text{sim}}(\mathbf{x}_0, \mathbf{x}_0^{\text{input}}) \right]}_{\text{guidance gradient}}
\end{aligned}
$$

Appendix II
Proof: Approximating expected denoised image $\mathbf{x}_0$ given $\mathbf{x}_t$ using the diffusion model

Appendix III
Training on the RUGD dataset

A. Information about the RUGD dataset

The RUGD dataset (Fig. 5, [15]) is an off-road dataset of video sequences captured from a small, unmanned mobile robot traversing in unstructured environments. It contains over 7,453 labeled images from 17 scenes, annotated with pixel-level segmentation over 24 semantic classes. The annotated frames are spaced five frames apart.

We split the 24 semantic categories as 16 in-distribution labels: $\mathbb{C}_{\text{ID}}$ = {`dirt`, `sand`, `grass`, `tree`, `pole`, `sky`, `asphalt`, `gravel`, `mulch`, `rock-bed`, `log`, `fence`, `bush`, `sign`, `rock`, `concrete`}, and 8 OOD labels corresponding to "obstacle" classes: $\mathbb{C}_{\text{OOD}}$ = {`vehicle`, `container/generic-object`, `building`, `bicycle`, `person`, `bridge`, `picnic-table`, `water`}

B. Training a diffusion model on the RUGD Dataset

Our diffusion model is trained on samples from the RUGD train split that does not contain humans and artificial constructs (Fig. 6 *(left)*). The out-of-distribution and anomalous images are 'held out' for evaluation (Fig. 6 *(right)*). As can be seen from Fig. 7, our trained diffusion model successfully generates realistic images containing only in-distribution classes such as trees, grass, and the occasional footpath.

C. Performance on RUGD Dataset

We evaluate our method on the RUGD [15] dataset and share qualitative results in Fig. 8. The diffusion model is trained on RUGD data without artificial constructs. At test-time, we present our method with OOD images containing anomalies from held-out classes. Fig. 9 shows the impact of each component of our *analysis* pipeline.

D. Tuning the Guidance Strength

The strength of the guidance term in our diffusion model can be tuned to enforce a variable level of consistency between the image being generated $\mathbf{x}'$ and the target image $\mathbf{x}$. Fig. 10 *(left)* shows the impact of the guidance term on the image generated, for a sample anomalous image $\mathbf{x}$ from the RUGD dataset.

E. Limitations

Our method's performance appears to degrade on samples with a large number of anomalies (Fig. 10 *(right)*), compared to input images with a fewer number of anomalies. This can be largely attributed to under-segmentation by SAM, which uses an (input-independent) hyperparameter that dictates the number of segments to output. Future work could aim to develop a criterion by which SAM can vary its segmentation resolution, without requiring human intervention.

Appendix IV
Toy experiments on the CLEVR dataset

A. Validating our improved diffusion guidance

We show that the SoftRect energy function improves our ability to remove anomalies via guided diffusion from the CLEVR dataset [79] in Section IV-B . In Section III-D, we also show what happens as we tune the guidance strength hyperparameter.

B. Validating the SoftRect Guidance Function

To validate our choice of SoftRect over L2 guidance, we train two diffusion models — one for each guidance method — on examples from the CLEVR data [79] containing no reds, yellows, or browns (Fig. 11 *(left)*). We then present the trained model with anomalous images containing those held out colours, and compare their ability to remove anomalies without modifying non-anomalous parts of the image (Fig. 11 *(right)*). We find that SoftRect indeed outperforms L2 guidance when it comes to diffusion model-based anomaly removal.

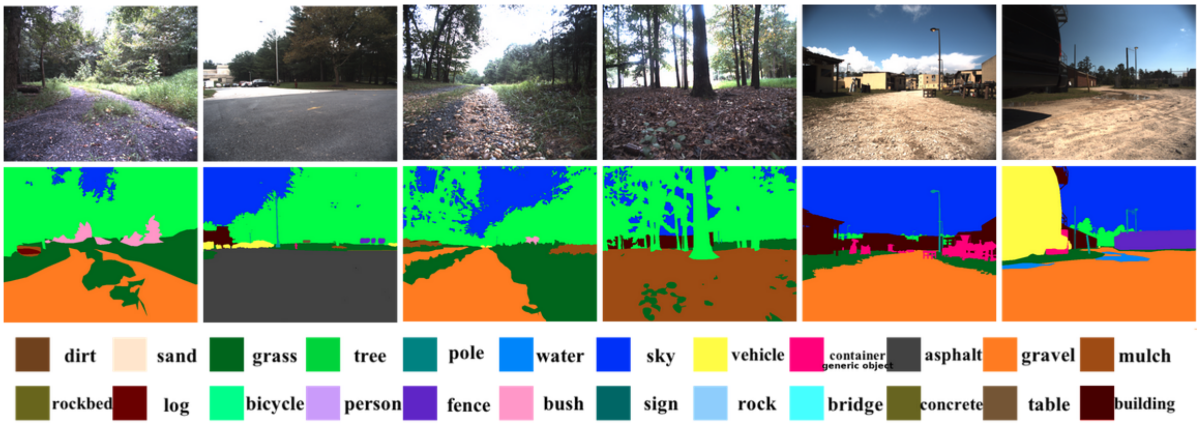

Fig. 5: Examples of video frames, annotations and semantic classes from the full RUGD dataset [15].

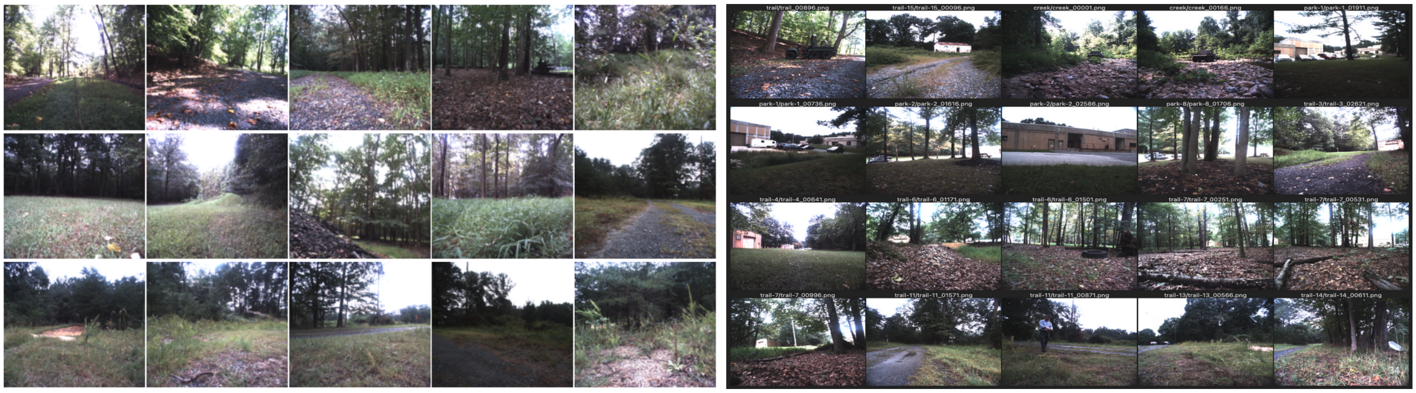
trail/trail_00896.png
trail-15/trail-15_00096.png
creek/creek_00001.png
creek/creek_00166.png
park-1/park-1_01911.png
park-1/park-1_00736.png
park-2/park-2_01616.png
park-2/park-2_02586.png
park-8/park-8_01706.png
trail-3/trail-3_02621.png
trail-4/trail-4_00641.png
trail-6/trail-6_01171.png
trail-6/trail-6_01501.png
trail-7/trail-7_00251.png
trail-7/trail-7_00531.png
trail-7/trail-7_00996.png
trail-11/trail-11_01571.png
trail-11/trail-11_00871.png
trail-13/trail-13_00566.png
trail-14/trail-14_00611.png

Fig. 6: In-distribution and out-of-distribution images from the RUGD dataset. ***Left:*** Examples of the in-distribution images on which our RUGD diffusion model was trained. In general, these images contain natural, off-road vegetation — a mixture of forest, meadow, mulch, and paths, without any humans or artificial constructions like buildings or vehicles. ***Right:*** Examples of held-out, out-of-distribution RUGD images the robot might encounter. These contains anomaly objects like buildings and vehicles. The diffusion model trained on the images on the left must remove anomalies from the images on the right.

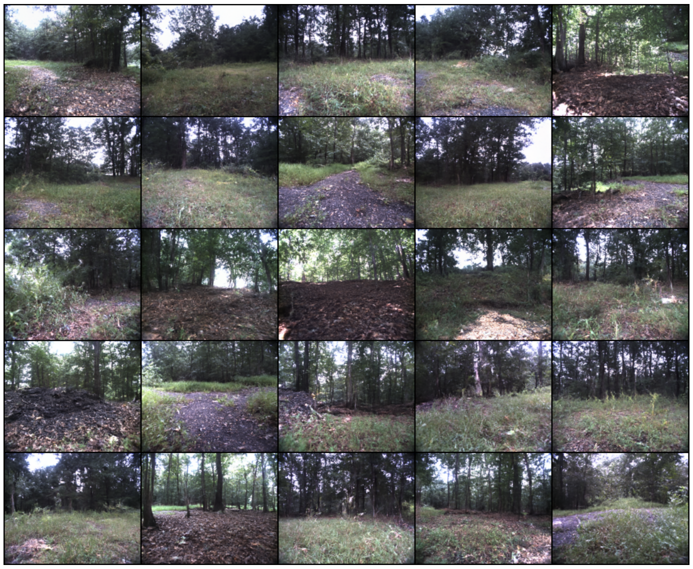

Fig. 7: Samples generated from the trained diffusion model (without conditioning). The training data is shown in **??** *(left)*. The generated samples are photorealistic, and appear very similar to the training images.

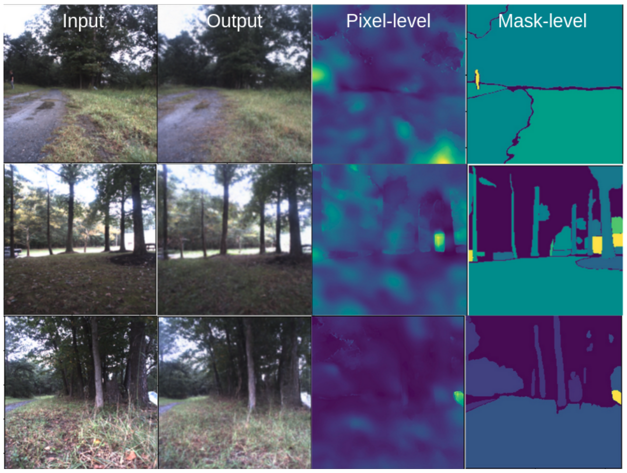

Fig. 8: Qualitative results on small, anomalous examples from the RUGD dataset. Our full `DiffUnc` pipeline does particularly well at detecting small and camouflaged/human-imperceptible anomalies.

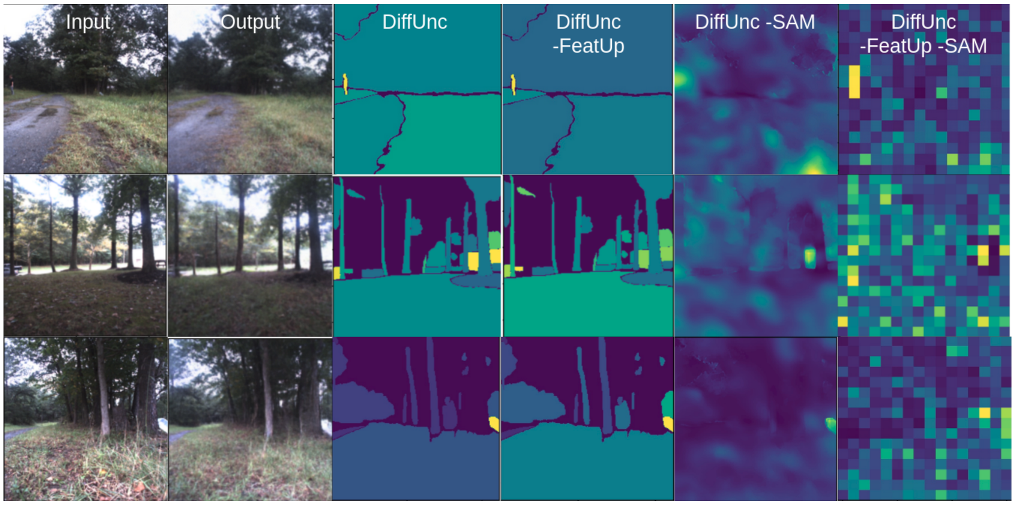

Fig. 9: Impact of SAM and FeatUp on `DiffUnc` performance. Removing FeatUp lowers the contrast between in- and out-of- distribution segments. Removing SAM particularly degrades performance on small anomalies.

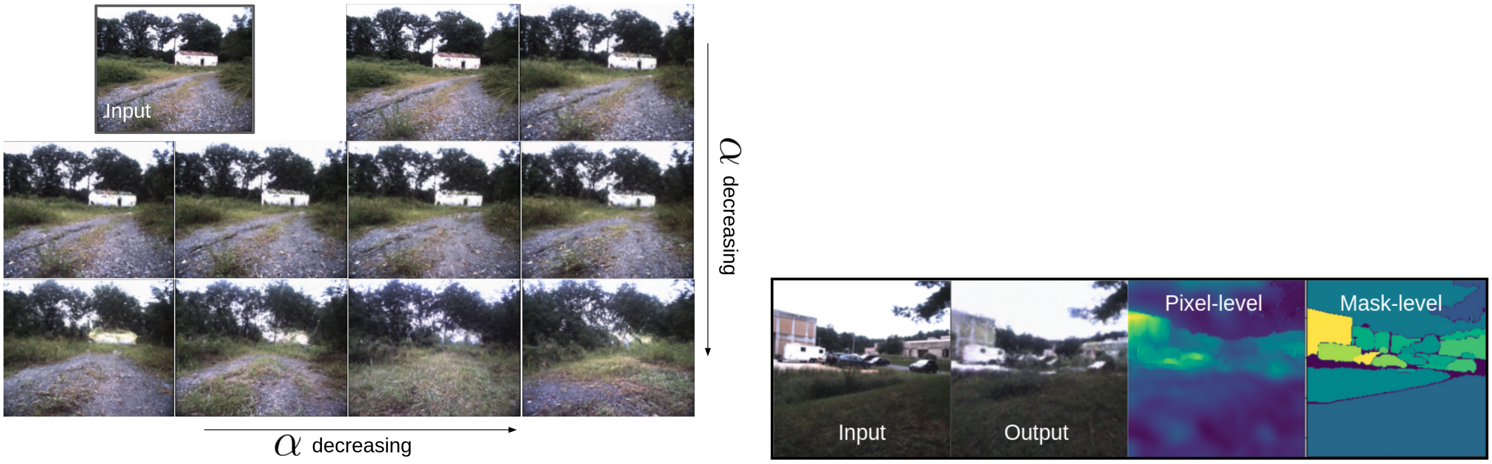

Fig. 10: ***Left:*** When there are many anomalies in a single image (in this case all vehicles and buildings are anomalies), the performance of our model appears to degrade. This is likely because SAM is under-segmenting an image with a high number of objects. The number of masks SAM generates is a hyperparameter in our pipeline and so could be adjusted, but the question of how to handle images with a varying number of objects is a non-trivial one. ***Right:*** The strength of our diffusion model's guidance term can be tuned by a hyperparameter α. As α decreases, the guidance enforcing consistency between the target image and the diffusion process' output weakens, and an increasing number of changes are allowed.

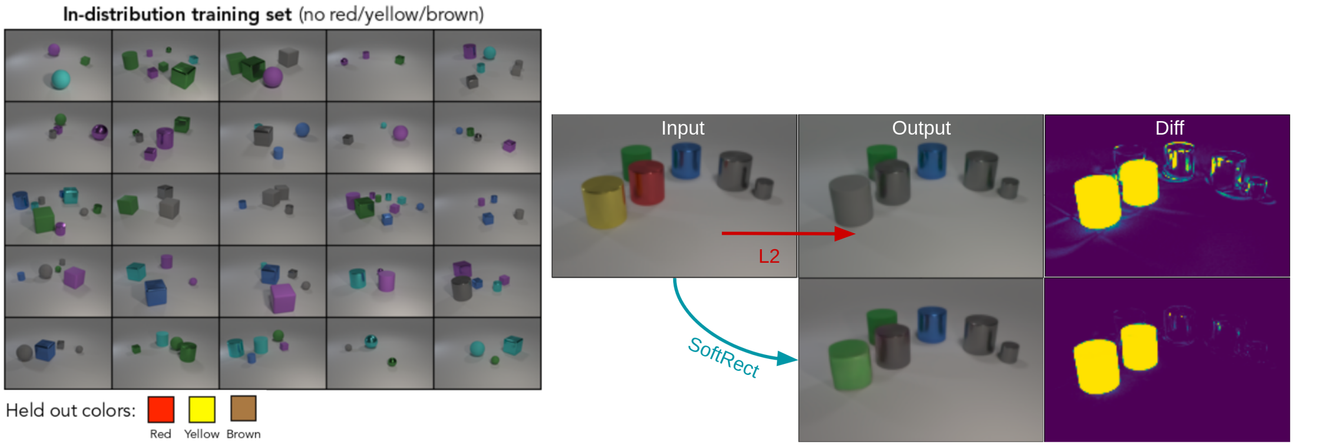

Fig. 11: ***Left:*** Examples of CLEVR images without reds, yellows, or browns, upon which our tabletop diffusion models were trained. ***Right:*** Comparing the performance of SoftRect- and L2- guided diffusion models shows that SoftRect successfully guides the diffusion process toward fewer unecessary changes on in-distribution objects. As a result, SoftRect-guided diffusion yields an uncertainty map with few false-positive pixels.