# Data Augmentation via Latent Diffusion for Saliency Prediction - Supplementary material

Bahar Aydemir[1], Deblina Bhattacharjee[1], Tong Zhang[1], Mathieu Salzmann[1], and Sabine Süsstrunk[1]

School of Computer and Communication Sciences, EPFL, Switzerland
{bahar.aydemir, deblina.bhattacharjee, tong.zhang, mathieu.salzmann, sabine.susstrunk}@epfl.ch

In this supplementary material, we provide additional quantitative and qualitative results and ablation studies of the proposed model. The document is structured as follows:



## 1 Details on saliency-guided cross-attention

Please visit our demo to see the animated version of this mechanism. We extract cross-attention features from the input image and prompt, then multiply them with the initial saliency to create spatial attention maps. Each word generates a spatial attention map, and we select the one with the highest sum as the target editing region. These selected spatial maps are shown in Figure-4 in the main paper. We utilize prompts from the MS-COCO dataset [17] and modify them as described in Section 4 during editing in the main paper. We use the target region to edit and the edited image to train the models with augmentation. We provide the details of the loss function in our demo and in Figure 1.

In brightness and contrast edits, we use an empty token. In the original images, each cross-attention map corresponds to a word in the prompt. To edit a region, we add the selected spatial attention map to the cross-attention maps, increasing their count by one. Thus, we need a word for this map. For colors, we use words representing the target color. Yet, for brightness and contrast, there are no direct linguistic equivalents. Terms like "bright" and "contrasted" create glowing cyan regions or black-white artifacts. Duplicating the word associated

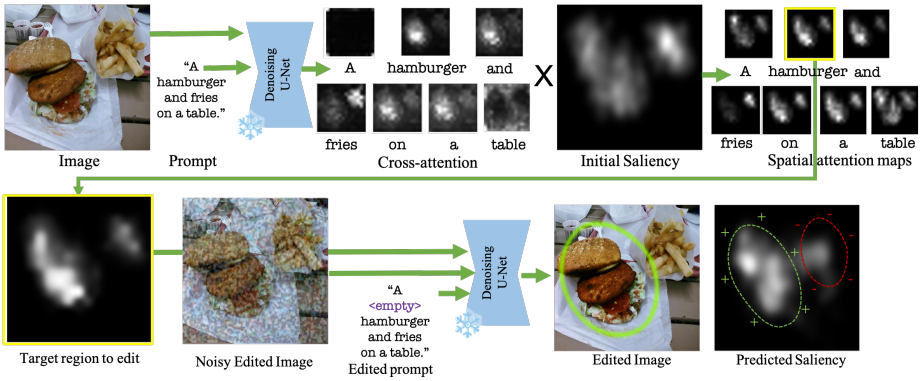


**Fig. 1:** We extract cross-attention features from the input image and prompt, then multiply them with the initial saliency to create spatial attention maps. Each word generates a spatial attention map, and we select the one with the highest sum as the target editing region. These selected spatial maps are shown in Figure-4 in the main paper. We use the target region to edit and the edited image pairs to train the models with augmentation.

with the selected region undesirably amplifies its characteristics. Therefore, we use an empty prompt("") with no semantic meaning but has a corresponding cross-attention map, allowing us to perform edits.

## 2 Quantitative Results on the Edited Images

| Model | CC ↑ | KL ↓ | NSS ↑ | SIM ↑ |
|---|---|---|---|---|
| DeepGazeIIE [18] | 0.644 | 0.512 | 1.180 | 0.600 |
| TempSAL [1] | 0.814 | 0.471 | 1.519 | 0.733 |
| **Ours** | **0.893** | **0.392** | **1.623** | **0.799** |

**Table 1:** Quantitative comparison on the *edited* images. Our model, driven by its controllability of saliency, consistently outperforms the baselines [1,18] which lack controllability. Here, we show results for the best-performing baselines; i.e. DeepGazeIIE [18] on the MIT1003 and CAT2000 benchmarks, and TempSAL [1] on the SALICON benchmark.

We see from Table 1 that our method consistently outperforms the baselines on all the saliency metrics on edited images. Being a generative model, our method is able to control the saliency of the edited images, thus achieving better results. In essence, our model has a notion of both the generated image edits and the corresponding saliency of those edits. The baseline methods [1,18] which are

discriminative cannot infer the changes in saliency based on the generated image edits, resulting in poorer performance. Our method is thus controllable, whereas the baseline methods are not. Please note that we only report results on the best-performing discriminative baselines, which are TempSAL [1] on SALICON [10], and DeepGazeIIE [18] on MIT1003 [13] as well as CAT2000 [2] (c.f. Tables 1-main paper, 3, 4). We use [28] and [16] to create the text prompts for the images from MIT1003 [13] and CAT2000 [2], respectively. For unedited images, we show the comparison of the performances of our model and the baselines in the following section.

## 3 Comparison with Attention Retargeting Models as Augmentation Methods

In this section, we compare our augmentation method with existing attention-retargeting techniques. Specifically, we use two baselines: RSGIE [21] and GSN [20], as shown in Table 2. We have used the segmentation masks from [17] to select a region to augment with these methods.

Our model, having a diffusion backbone that is trained on extensive datasets, possesses rich image priors enabling the generation of high-quality, realistic images and augmentations. Our method modifies the latent code during the diffusion process, allowing the model to produce images based on desired edits. This approach ensures inherently realistic augmented images with better control, unlike other methods that verify realism post-generation.

Moreover, using RSGIE [21] or GSN [20] with automated mask generation may not yield optimal masks compared to cross-attention maps. This results in suboptimal performance, as evidenced by our experiments. GSN [20] typically creates color edits that direct visual attention but are insufficient for effective augmentations. While RSGIE [21] improves NSS and SIM, it fails to enhance KL and CC, indicating over-estimated or under-estimated saliency predictions. This suggests that the edits produced by RSGIE [21] are too intense. Our method demonstrates superior performance across all evaluated metrics, highlighting the efficacy of using diffusion models for data augmentation in saliency prediction tasks. The results indicate that our approach not only enhances prediction accuracy but also maintains high realism and control over the augmentation process.

| Augmentation method | CC ↑ | KL ↓ | NSS ↑ | SIM ↑ |
|---|---|---|---|---|
| None [24] | <u>0.907</u> | <u>0.193</u> | 1.926 | 0.797 |
| *w/* GSN [20] | 0.904 | 0.196 | 1.919 | 0.793 |
| *w/* RSGIE [21] | 0.905 | 0.194 | <u>1.930</u> | <u>0.800</u> |
| *w/* Ours | **0.911** | **0.185** | **1.937** | **0.805** |

**Table 2:** Performance comparison of different augmentation methods on saliency prediction. Our method shows superior results across all metrics compared to the baseline methods.

## 4 Additional Quantitative Results

For a fairer comparison with the existing baselines [6–9, 14, 15, 18, 19, 22, 27] we compare our base model with existing discriminative methods on *unedited* images from MIT1003 [13] and CAT2000 [2] benchmarks, as shown in Tables 3 and 4, respectively. In this setting, our base model is trained only with the denoising task without any kind of editing denoted as ours w/o augmentation. We also report the results of our saliency prediction method with augmentation in the last row. In Table 3, we report the performance on the MIT1003 [13] benchmark where we show that our method outperforms the baselines on the AUCJ, KL and CC metric while being comparable on NSS, sAUC, and SIM metrics. Additionally, in Table 4 for the CAT2000 [2] benchmark, we see a similar trend in performance, where our method outperforms the baselines on the AUCJ, KL, CC and SIM metrics while being comparable on NSS and sAUC metrics. Although the baselines perform relatively well to our method on *unedited* images, they do not perform well on *edited* ones as seen in Table 1. It should be noted that the baseline models lack the capability to extract multi-level features from the diffusion model, thus being unable to leverage generative features to boost their performance. In contrast, our method makes use of the diffusion method to achieve more accurate saliency estimations compared to the baselines.

sAUC accounts for center bias, which occurs because photographers often place salient objects near the center. Our method focuses on regions marked by cross-attention maps, which usually contain salient objects. Inadequately accounting for this center bias results in a lower sAUC score. NSS emphasizes accuracy at exact fixation points; thus, a model may score well on CC, SIM, and KLD by capturing the general distribution but score low on NSS if it misses exact fixation locations.

## 5 Details of the Metrics

Our saliency prediction evaluations use the following standard metrics:
**Area Under the Curve (AUC) [4]**: Interprets saliency prediction as fixation vs non-fixation classification, with the AUC score reflecting the balance between true and false positives. A higher AUC indicates fewer false positives. The **sAUC** [3] variant adjusts for center bias and observer variability by sampling false positives from other observers' fixations.
**Normalized Scanpath Saliency (NSS)** [23]: Measures how predicted saliency at actual fixation points compares to average prediction. A unit NSS suggests predictions at fixation points are one standard deviation above the mean.
**Kullback–Leibler Divergence (KL)** [26]: Measures the discrepancy between predicted and actual saliency maps. Scores near zero signify a closer match to ground truth.
**Pearson's Correlation Coefficient (CC)** [11]: Assesses the linear correlation between predicted and actual saliency maps, with scores near one indicating a strong correlation.

| Model | AUCJ ↑ | KL ↓ | NSS ↑ | CC ↑ | sAUC ↑ | SIM ↑ |
|---|---|---|---|---|---|---|
| DINet [27] | 0.907 | 0.704 | 2.855 | 0.766 | 0.636 | 0.561 |
| DSCLRCN [19] | 0.880 | 0.725 | 2.813 | 0.750 | 0.624 | 0.530 |
| SalNet [22] | 0.877 | 0.759 | 2.699 | 0.728 | 0.630 | 0.547 |
| SAM [6] | 0.911 | 0.682 | **2.888** | 0.768 | 0.613 | 0.552 |
| SALICON [8] | 0.871 | 0.818 | 2.757 | 0.728 | 0.609 | 0.533 |
| CEDN [14] | 0.895 | 0.660 | 2.525 | 0.790 | 0.630 | 0.592 |
| DeepGaze II [15] | 0.881 | 0.744 | 2.480 | 0.794 | 0.627 | 0.567 |
| EML Net [9] | 0.886 | 0.779 | 2.477 | 0.790 | 0.630 | 0.563 |
| UNISAL [7] | 0.904 | 0.777 | 2.678 | 0.750 | 0.692 | 0.610 |
| DeepGaze IIE [18] | 0.889 | 0.516 | 2.599 | 0.774 | **0.788** | **0.772** |
| **Ours w/o augmentation** | 0.912 | 0.591 | 2.726 | 0.784 | 0.699 | 0.689 |
| **Ours w/ augmentation** | **0.913** | **0.502** | 2.873 | **0.810** | 0.734 | 0.722 |

**Table 3:** Quantitative results on *unedited* images from MIT1003 [13]. Our method outperforms on the AUCJ, KL and CC metrics and yields the second-best performance on NSS, SAUC, and SIM metrics.

**Similarity (SIM) Score [12]**: is defined as the sum of minimum values between predicted and actual saliency maps across all pixels. A score of 1 denotes perfect prediction, as both maps are probability distributions.

## 6 Additional Qualitative Results

In Figures 3, 4, 5, 6, 7, and 8, we compare the saliency maps obtained with our method with the different ground truths. In the first column, we use the ground truth saliency maps from SALICON [10] for the original unedited image, and for the latter columns, we use the ground truth collected via our user study on edited images. We show the predictions of the baselines, namely DeepGazeIIE [18] and TempSAL [1], alongside our models' predictions. As shown in these figures, our model learns the photometric cues that increase saliency and is able to predict human visual attentional patterns under these image edits. We now explain each qualitative result shown in Figures 3, 4, 5, 6, 7, and 8 in what follows.

### 6.1 Contrast

In Figures 3 and 4, we increase the contrast of a selected region. We compare the saliency maps obtained with our method with the different ground truths. In the first column, we use the ground truth saliency maps from SALICON [10] for the original unedited image, and for the latter columns, we use the ground truth collected via our user study on edited images. For example, we increase the contrast of the object that the man is holding in Figure 3 and the dog

| Model | AUCJ ↑ | KL ↓ | NSS ↑ | CC ↑ | sAUC ↑ | SIM ↑ |
|---|---|---|---|---|---|---|
| DINet [27] | 0.871 | 0.590 | 2.377 | 0.877 | 0.609 | 0.770 |
| DSCLRN [19] | 0.862 | 0.846 | 2.360 | 0.833 | 0.550 | 0.685 |
| SAM [6] | 0.880 | 0.560 | **2.388** | 0.889 | 0.582 | 0.770 |
| SALICON [10] | 0.861 | 0.866 | 2.340 | 0.803 | 0.529 | 0.648 |
| CEDN [14] | **0.881** | 0.360 | 2.300 | 0.870 | 0.590 | 0.751 |
| DeepGaze II [15] | 0.875 | 0.810 | 1.974 | 0.880 | 0.605 | 0.772 |
| EML Net [9] | 0.874 | 0.971 | 2.380 | 0.880 | 0.591 | 0.752 |
| DeepGaze IIE [15] | 0.869 | 0.345 | 2.112 | 0.819 | **0.668** | 0.706 |
| **Ours w/o augmentation** | **0.881** | 0.354 | 2.233 | 0.871 | 0.612 | 0.753 |
| **Ours w/ augmentation** | **0.881** | **0.322** | 2.385 | **0.897** | 0.639 | **0.780** |

**Table 4:** Quantitative results on *unedited* images from CAT2000 [2]. Our method outperforms on AUCJ, KL,CC and SIM metric and yields the second-best performance on NSS and SAUC metrics.

in Figure 4. Our model is able to control the saliency of the contrast-edited sections whereas the baselines report nearly identical predictions for the original and edited images.

### 6.2 Brightness

In Figures 5 and 6, we increase the brightness of a selected region. For example, we amplify the brightness of the cat in Figure 5 and that of the pizza in Figure 6. Our model is able to control the saliency of the brightness-edited regions within the cat and the pizza, whereas the baselines report constant predictions. To depict the results, we compare the saliency maps obtained with our method with the different ground truths. Similar to the setting in the previous section, we use the ground truth saliency maps from SALICON [10] for the original unedited image and the ground truth collected via our user study for the edited images, as shown on the first and latter columns, respectively.

### 6.3 Color

In Figures 7 and 8, we edit the color of a selected region. For example, we change the color of the toy in Figure 7 to pink with increasing intensity. Our model is able to follow the shift in human visual attention towards the toy, under this color edit. Similarly, in Figure 8, we change the color of the traffic lights to green progressively. Our model accurately improves the saliency prediction over the edited regions whereas the baselines produce nearly identical predictions for the original and the edited images. We show the ground truth saliency maps from SALICON [10] in the first column and we use the ground truth collected by our user study in the latter columns, similar to the settings in the previous sections.

### 6.4 Reducing Contrast and Brightness

We focus on the edits that increase saliency to keep the edited regions salient in our study. However, when an edit reduces saliency, it is unclear whether the reduction makes the region non-salient or not. We show such edits that diminish brightness and contrast and their effects on saliency in Figure 9 and 10. For example, in Figure 9, we reduce the brightness of the horse. Our model shifts the saliency prediction away from the horse as it becomes less visible. Similarly, in Figure 9, we decrease the contrast of the pizza. As the intensity of the edit increases, our model shifts its focus to the bottle on the left-hand side. In both examples, our model is still able to control (increase or decrease) its saliency prediction on the edited regions.

## 7 Editing

We present our inference process in Algorithm 7. During the editing process, we denote one pass of the frozen denoising U-Net [25] using the modified cross-attention maps as denoise(.). We denote all types of editing as edit(.) in this algorithm for simplicity. We refer to Section 4 of the main paper for the details of the different editing operations.

```
# Input: image I, saliency map S, prompt P

def editing(I, S, P):
    # Get epsilon from inverted image
    epsilon = invert(I)
    # Encode image I
    Z_I = encode(I)
    # Noise Z_I with epsilon based on timestep t
    Z_I_t = noise_forward(Z_I, t, epsilon)
    # Find the target region to edit M
    M = get_crossatn(I,S,P)
    # Edit Z(t_S) based on saliency map S
    Z_E_t = edit(Z_I_t, S, M, P, strength)
    # Loop for editing and denoising process
    for t in reversed(range(1, t + 1)):
        # Denoise and edit Z_E_t
        Z_E_prime, epsilon_prime = denoise(Z_E_t, t)
        # Check for excessive edits
        is_ok =  readout(Z_E_prime)
        if not is_ok :
            Z_E_t = edit(Z_I_t, S, M, P, strength*0.9)
            continue
        else:
            Z_E_t_minus_1 = denoise(Z_E_prime,t)
    # Return edited and denoised image
    I_E_prime = decode(Z_E_prime)
    return I_E_prime
```

**Listing 1.1:** Editing Algorithm

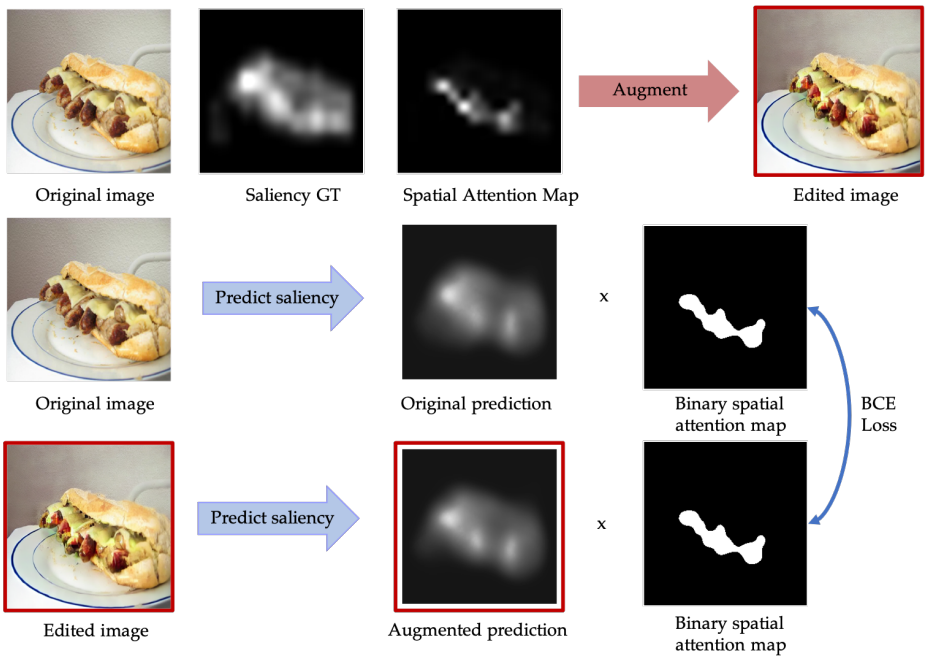


**Fig. 2:** Overview of the data augmentation process. Given an input image, saliency ground truth, and a selected region to edit (denoted by the spatial attention map), we modify the image to create an augmented version. Our goal is to increase the saliency in the edited regions. A saliency prediction model then estimates saliency for both the original and the edited images. These predictions are multiplied by binary spatial attention maps to mask the areas of interest. Binary Cross-Entropy (BCE) loss is used to determine whether the model has increased its saliency prediction in the edited region. We train the saliency prediction models by comparing the difference in the predicted saliency maps to this expected outcome.

In the editing process, we leverage the saliency-guided cross-attention map to perform the localized edits. To this end, we append a word to the prompt depending on the type of edit and inject the selected saliency-guided cross-attention map into the denoising U-Net. This process modifies $\mathbf{Z}'$, which is shared between the generated image and the predicted saliency. By denoising and decoding this edited $\mathbf{Z}'_E$, we obtain the generated image edits $\mathbf{I}'_E$. This enables us to use the saliency-guided cross-attention maps to perform the localized edits. We follow this approach through the results presented in our work.

## 8 Ablation on Losses and Hyperparameters

For training our model, we use readout, saliency, and editing losses. In Table 5, we present an ablation study for the impact of these losses on our approach.

We add each loss one by one and show that all the incorporated losses improve the overall saliency prediction. We tested multiple hyperparameters in the range $[10^{-3}, 10^{2}]$. We found that $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$ (global contrast), $\lambda_3 = 0.5$ (classification), $\lambda_4 = -0.2$, $\lambda_5 = 0.2$, $\lambda_6 = 0.1$ yield the best results. Hence, we use these hyperparameters through the experiments presented in our work. Additionally, we illustrate our loss in the augmentation process in Figure 2.

| Model | CC ↑ | KL ↓ | NSS ↑ | SIM ↑ |
|---|---|---|---|---|
| Original | 0.860 | 0.338 | 1.837 | 0.747 |
| + Readout losses | 0.876 | 0.273 | 1.673 | 0.760 |
| + Saliency losses | 0.908 | 0.179 | 1.927 | 0.788 |
| + Augmentation | **0.915** | **0.191** | **1.946** | **0.807** |

**Table 5:** Results of ablation studies of the losses on the SALICON [10] validation dataset. The first row denotes the model with only the denoising losses. In the second row, the model has both the denoising and readout. The third row denotes the performance with the saliency losses. As evidenced by the improved accuracy metrics, these losses effectively learn the multi-level features to read out the saliency maps to refine the saliency prediction.

## 9 Ablation on the impact of mixing ratio p

The ablation study on the augmentation ratio between augmented and original images are presented in Table 6. A value of p=1.0 denotes training the model without any augmentation. Notably, optimal performance is observed when the quantity of augmented data matches that of the original data.

| $p$ | KL ↓ | NSS ↑ | CC ↑ | SIM ↑ |
|---|---|---|---|---|
| $p = 1.0$ | 0.191 | 1.927 | 0.908 | 0.788 |
| $p = 0.7$ | 0.199 | 1.933 | 0.911 | 0.792 |
| $p = 0.5$ | **0.179** | **1.946** | **0.915** | **0.807** |
| $p = 0.3$ | 0.289 | 1.784 | 0.884 | 0.780 |
| $p = 0.1$ | 0.437 | 1.790 | 0.843 | 0.733 |

**Table 6:** Results of the ablation for the ratio between augmented images and original images. p=1.0 denotes the model is trained without augmentation. The model performs best when the amount of the augmented data is equal to the original data.

## 10 Comparison of the classical data augmentation methods

### 10.1 Discussion

**Cropping:** Cropping can remove crucial context or salient features from an image, altering visual attention. Saliency is linked to both the features of interest

and their surrounding context. By cropping an image, the model may lose access to contextual cues that help determine the saliency of regions within the original image.

**Vertical flip:** Vertical flipping changes the natural orientation of objects and scenes in ways that are often not encountered in the real world. Many objects and scenes have an 'up' and 'down' orientation. Moreover, the objects stay on the ground, hence creating an expectation toward the ground/horizon line [5]. It could disrupt the model's ability to learn gravity-centered or orientation-specific cues that are important for determining saliency.

**Horizontal flip:** Horizontal flipping stands out as the most effective classical data augmentation method for saliency, as it maintains the image's overall layout and structure, introducing realistic variations seen in real-world scenarios. Due to the inherent bilateral symmetry in most natural scenes and objects, horizontal flipping is a realistic variation for saliency prediction. This augmentation varies training data without altering the essential characteristics that define saliency.

**Rotation:** introduces variability that mirrors natural viewing experiences by presenting objects in varying orientations, but excessive rotations can negatively impact model performance due to humans' preference for seeing objects in their typical orientation.

| Augmentation | KL ↓ | NSS ↑ | CC ↑ | SIM ↑ |
|---|---|---|---|---|
| Rotate45 | 0.193 | 0.906 | 1.914 | 0.789 |
| Rotate135 | 0.196 | 0.904 | 1.926 | 0.787 |
| JpegCompression1 | 0.200 | 0.905 | 1.921 | 0.753 |
| JpegCompression2 | 0.203 | 0.901 | 1.916 | 0.749 |
| Noise1 | 0.240 | 0.906 | 1.892 | 0.716 |
| Noise2 | 0.222 | 0.903 | 1.902 | 0.798 |
| Cropping1 | 0.204 | 0.904 | 1.926 | 0.752 |
| Cropping2 | 0.199 | 0.904 | 1.871 | 0.763 |
| Inversion | 0.200 | 0.903 | 1.922 | 0.721 |
| MotionBlur1 | 0.197 | 0.905 | 1.915 | 0.766 |
| MotionBlur2 | 0.202 | 0.903 | 1.828 | 0.747 |
| Vertical Flip | 0.196 | 0.903 | 1.926 | 0.741 |
| Horizontal Flip | 0.187 | 0.908 | 1.930 | 0.801 |
| Shearing1 | 0.198 | 0.903 | 1.921 | 0.762 |
| Shearing2 | 0.201 | 0.899 | 1.893 | 0.731 |
| Shearing3 | 0.198 | 0.902 | 1.905 | 0.724 |
| Original | 0.193 | 0.906 | 1.926 | 0.797 |
| Ours | **0.185** | **0.911** | **1.937** | **0.805** |

**Table 7:** Results of the ablation for using classical augmentation methods. For each method, the model [24] is trained from scratch. The model performs best with our augmentation method. All of the classical methods except horizontal flip, decrease performance.

**Shearing:** Shearing distorts the geometry of the image, altering the shape and

sometimes the apparent size of objects within it. Such geometric distortion can affect the perceived importance of objects or regions, moving away from the natural way humans perceive saliency.
**Color inversion:** which flips the image's colors to their opposites on the color wheel, significantly changes an image's appearance and can hinder the model's ability to recognize salient features, as it contrasts with the natural color distributions the human visual system expects.
**Motion blur:** simulates the blurring effect in human vision during rapid movement, aiding models in recognizing salient features under challenging conditions. However, too much motion blur can obscure critical details.
**JPEG Compression:** JPEG compression degrades image quality through lossy compression, which can affect the visibility of fine details in an image. However, high levels of compression can distort the image enough to affect the model's ability to accurately predict salient regions. The compression artifacts might either distract the model or obscure subtle details that contribute to an object's saliency.

### 10.2 Quantitative Results

We evaluate our augmentation method against the classical augmentation methods listed in Table 7. We apply the same transformation to the image and its ground truth saliency map. We train all models on the SALICON dataset and the augmented images with these methods. We observe that most of the augmentations decrease performance except horizontal flip as seen in Table 7. Our augmentation method outperforms the classical augmentation methods in all metrics.

| **Method** | **Parameters** |
|---|---|
| Rotation-1 | 45 degrees |
| Rotation-2 | 135 degrees |
| Jpeg Compression-1 | quality = 5 |
| Jpeg Compression-2 | quality = 0 |
| Noise-1 | Gaussian, standard deviation = 0.1 |
| Noise-2 | Gaussian, standard deviation = 0.3 |
| Cropping-1 | Cut 64x64 random patch |
| Cropping-2 | Cut 128x128 random patch |
| Inversion | bitwise_not |
| MotionBlur-1 | size=15 pixels, angle=90 degrees |
| MotionBlur-2 | size=35 pixels, angle=90 degrees |
| Vertical flip | N/A |
| Horizontal flip | N/A |
| Shearing-1 | Affine transform the top right corner to center |
| Shearing-2 | Affine transform the top left corner to center |
| Shearing-3 | Affine transform the top-right corner to mid-right |

**Table 8:** List of classical data augmentation methods with parameters.

## 11 Details of the classical data augmentation methods

We present the parameters of the classical data augmentations in Table 7. We implemented all augmentations using cv2 and numpy libraries. We will make our code publicly available upon acceptance.

## 12 Limitations

Our current study is limited to photometric edits, such as changes in brightness, contrast, and color. We have not studied geometric edits, which alter the shape or position of objects within the image. In this case, the saliency of the image alters significantly since human visual attention is highly scene-dependent. This has to be addressed in future work. The cross-attention maps extracted from the latent diffusion model have a lower resolution than the image. This may result in some artifacts in the edited image.

## References

1. Aydemir, B., Hoffstetter, L., Zhang, T., Salzmann, M., Süsstrunk, S.: TempSAL - uncovering temporal information for deep saliency prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 2, 3, 5, 13, 14, 15, 16, 17, 18
2. Borji, A., Itti, L.: CAT2000: A large scale fixation dataset for boosting saliency research. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2015) 3, 4, 6
3. Borji, A., Sihite, D.N., Itti, L.: Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. IEEE Transactions on Image Processing (TIP) **22**(1), 55–69 (2013). https://doi.org/10.1109/tip.2012.2210727 4
4. Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., Durand, F.: What do different evaluation metrics tell us about saliency models? IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **41**(3), 740 (2019). https://doi.org/10.1109/tpami.2018.2815601 4
5. Bylinskii, Z., Recasens, A., Borji, A., Oliva, A., Torralba, A., Durand, F.: Where should saliency models look next? In: European Conference on Computer Vision (ECCV). pp. 809–824. Springer (2016). https://doi.org/10.1007/978-3-319-46454-1_49, https://www.ebook.de/de/product/27952074/computer_vision_eccv_2016.html 10
6. Cornia, M., Baraldi, L., Serra, G., Cucchiara, R.: Predicting human eye fixations via an LSTM-based saliency attentive model. IEEE Transactions on Image Processing (TIP) **27**(10), 5142–5154 (2018). https://doi.org/10.1109/tip.2018.2851672 4, 5, 6
7. Droste, R., Jiao, J., Noble, J.A.: Unified Image and Video Saliency Modeling. In: European Conference on Computer Vision (ECCV) (2020) 4, 5
8. Huang, X., Shen, C., Boix, X., Zhao, Q.: SALICON: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In: IEEE International Conference on Computer Vision (ICCV). pp. 262–270 (2015). https://doi.org/10.1109/iccv.2015.38 4, 5

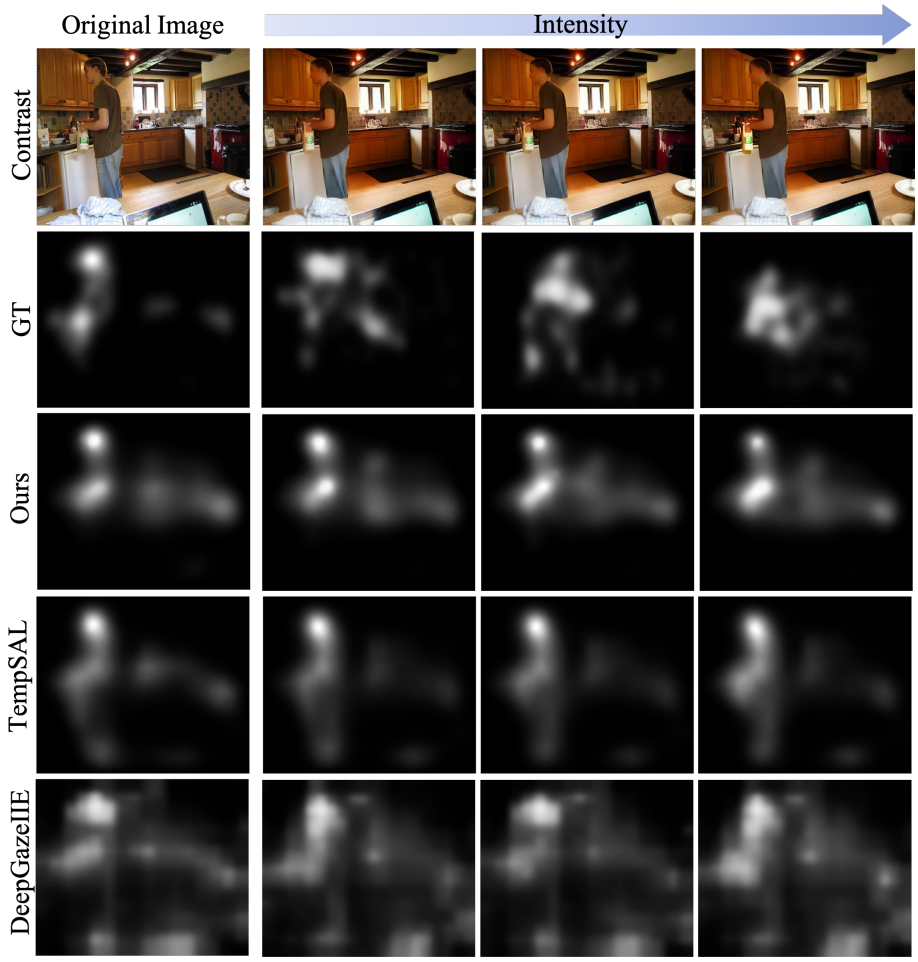


**Fig. 3:** We show the original image and the edited images with increasing contrast in the first row. The second row shows the ground truth maps for saliency and the third row shows our predictions. The fourth and the fifth rows show predictions of baselines TempSAL [1] and DeepGazeIIE [18] respectively. We amplify the contrast of the object that the man is holding in this image. Observe the shift in saliency towards the region with high contrast. Our model is able to enhance and thus, control the saliency of the edited region.

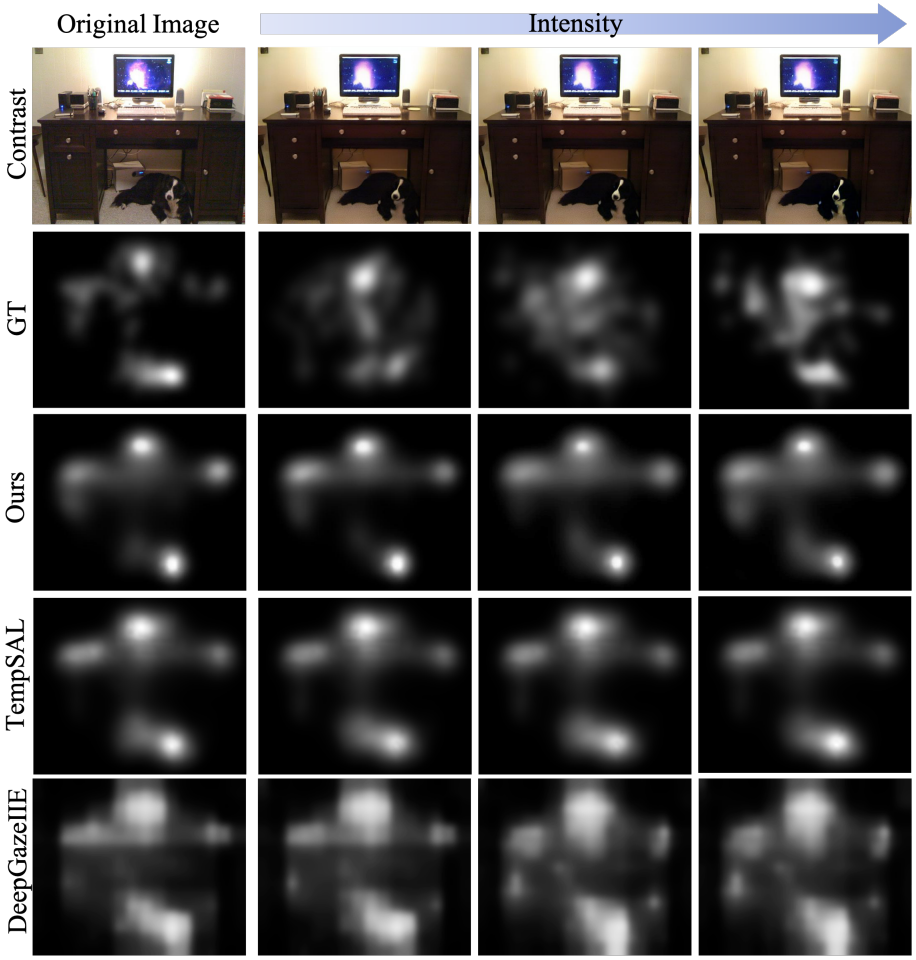


**Fig. 4:** We show the original image and the edited images with increasing contrast in the first row. The second row shows the ground truth maps for saliency and the third row shows our predictions. The fourth and the fifth rows show predictions of baselines TempSAL [1] and DeepGazeIIE [18] respectively. In this example, we amplify the contrast of the dog. While the items on the desk remain salient, more visual attention is focused on the dog with an increase in its contrast. The baselines remain constant in their predictions whereas our model is able to increase its saliency prediction over the edited region, reflecting the shift in human visual attention.

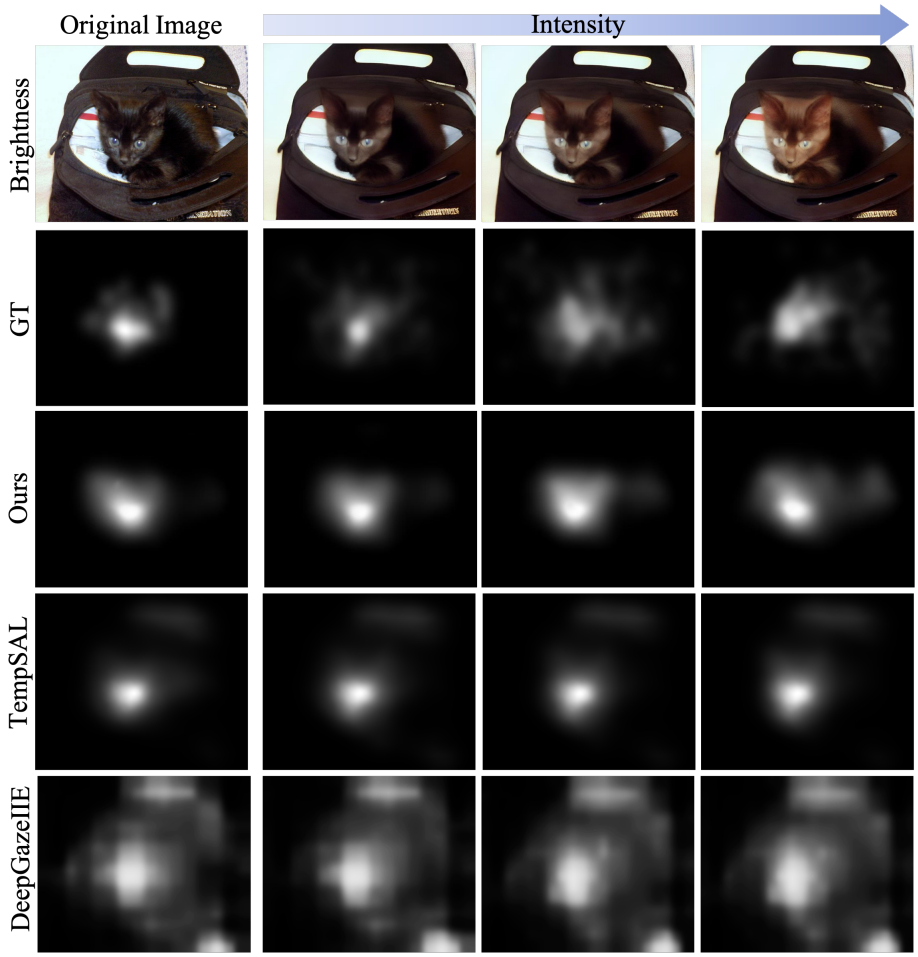


**Fig. 5:** We show the original image and the edited images with increasing brightness in the first row. The second row shows the ground truth maps of saliency and the third row shows our predictions. The fourth and the fifth rows show predictions of baselines TempSAL [1] and DeepGazeIIE [18] respectively. Our model is able to predict the saliency by following the brightness change towards the rear end of the cat, whereas the baseline predictions remain constant.

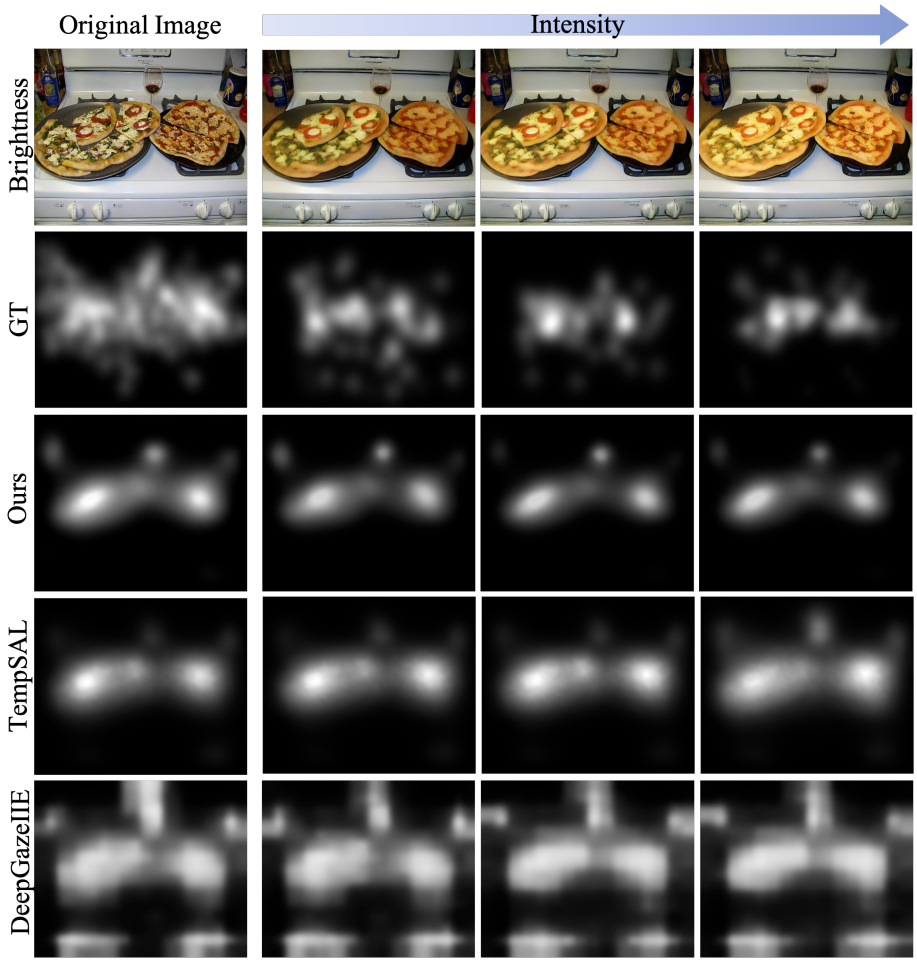


**Fig. 6:** We show the original image and the edited images with increasing brightness in the first row. The second row shows the ground truth maps of saliency and the third row shows our predictions. The fourth and the fifth rows show predictions of baselines TempSAL [1] and DeepGazeIIE [18] respectively. In this example, we amplify the brightness of the pizzas. Although the pizzas remain salient, we show the saliency of the bottle on the left and that of the glass in the middle decrease. Our model correctly predicts this decrease in human visual attention in predictions while the baselines remain constant or increase its prediction incorrectly as seen in the last column of TempSAL's [1] predictions.

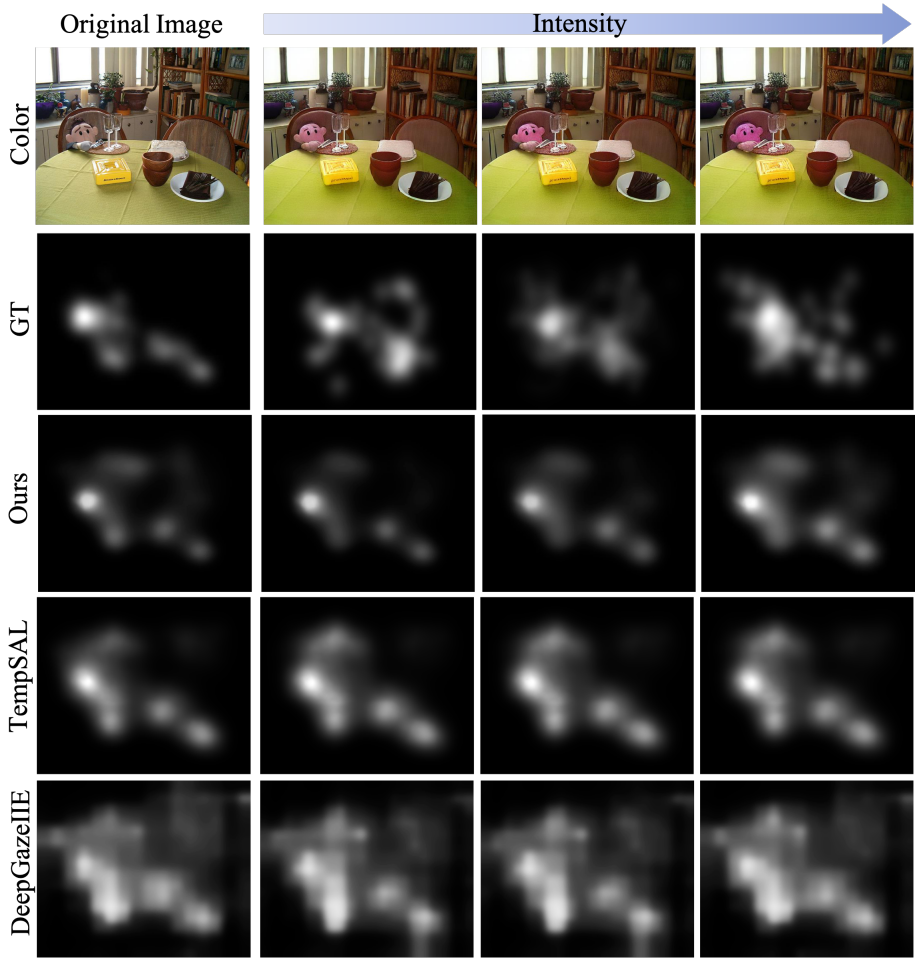


**Fig. 7:** We show the original image and the edited images with the color changing to "pink" in the first row. The second row shows the ground truth maps of saliency and the third row shows our predictions. The fourth and the fifth rows show predictions obtained with the baselines TempSAL [1] and DeepGazeIIE [18], respectively. By changing the color of the toy sitting on the chair, the attention on the face of the toy increases according to the intensity of the color edit. This increase in attention is evidenced in the ground truth saliency maps as well as in our predictions. TempSAL [1] initially enhances its prediction over the edited region, shown in the first column, but fails to further increase the saliency as the edit intensity increases.

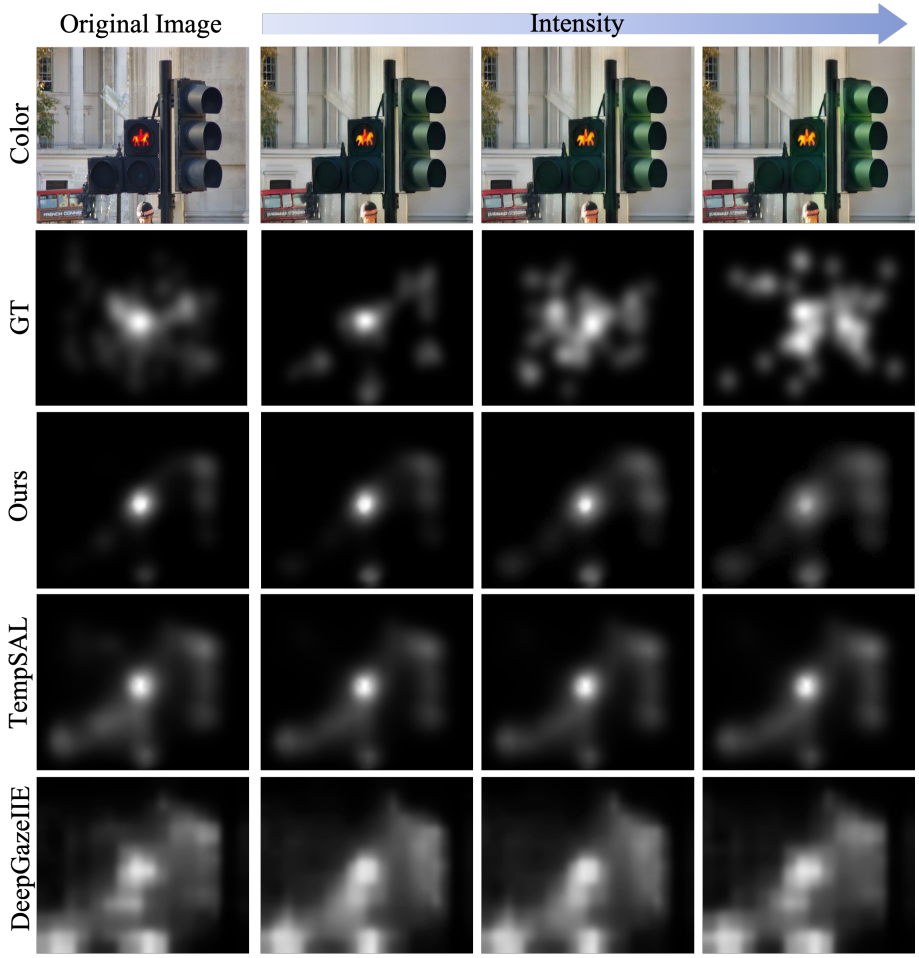


**Fig. 8:** We show the original image and the edited images with the color changing to "green" in the first row. The second row shows the ground truth saliency and the third row shows our predictions. The fourth and the fifth rows show predictions of baselines TempSAL [1] and DeepGazeIIE [18] respectively. Here, we increase the intensity of the green color on the traffic light. As the color intensifies, our model is able to enhance the saliency prediction within these color-edited regions, while the baselines fail to do so.

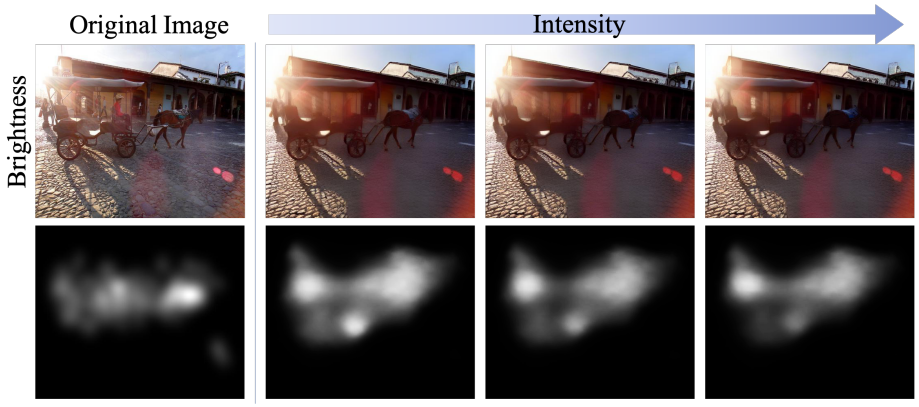


**Fig. 9:** We show the original image and the edited images with *decreasing* brightness in the first row. The second row shows the ground truth saliency map for the original image in the first column and our predictions in the latter columns. We diminish the brightness of the horse in this example. Our model shifts its saliency prediction away from the edited region as the horse becomes less visible, blending into the background.

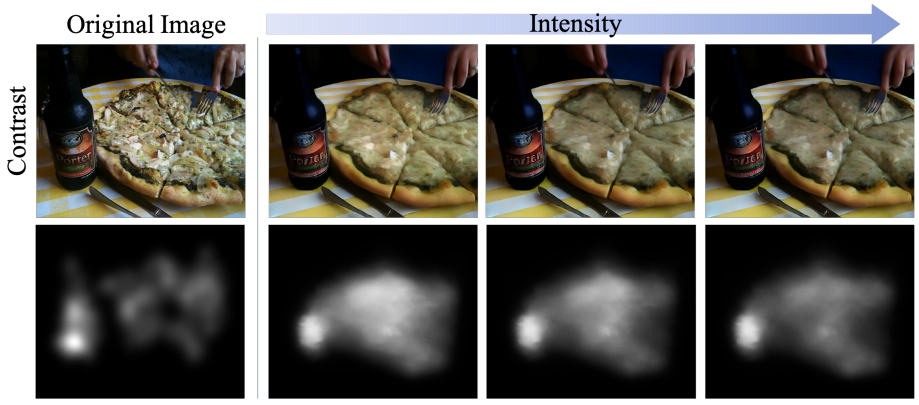


**Fig. 10:** We show the original image and the edited images with *decreasing* contrast in the first row. The second row shows the ground truth saliency map for the original image in the first column and our predictions in the latter columns. In this example, we reduce the contrast of the pizza. As the intensity of the edit increases, our model shifts its focus to the bottle on the left-hand side. This also demonstrates our model's ability to control the saliency of the edited region.

9. Jia, S., Bruce, N.D.B.: EML-NET: An expandable Multi-Layer NETwork for saliency prediction. Image and Vision Computing **95**, 103887 (2020). https://doi.org/10.1016/j.imavis.2020.103887, http://arxiv.org/abs/1805.01047 4, 5, 6
10. Jiang, M., Huang, S., Duan, J., Zhao, Q.: SALICON: Saliency in context. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015). https://doi.org/10.1109/cvpr.2015.7298710 3, 5, 6, 9
11. Jost, T., Ouerhani, N., von Wartburg, R., Müri, R., Hügli, H.: Assessing the contribution of color in visual attention. Computer Vision and Image Understanding **100**(1-2), 107–123 (2005). https://doi.org/10.1016/j.cviu.2004.10.009, http://www.sciencedirect.com/science/article/pii/S107731420500041X 4
12. Judd, T., Durand, F., Torralba, A.: A benchmark of computational models of saliency to predict human fixations. MIT Technical Report (2012) 5
13. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: IEEE International Conference on Computer Vision (ICCV). IEEE (2009). https://doi.org/10.1109/iccv.2009.5459462 3, 4, 5
14. Kroner, A., Senden, M., Driessens, K., Goebel, R.: Contextual encoder–decoder network for visual saliency prediction. Neural Networks **129**, 261 – 270 (2020). https://doi.org/10.1016/j.neunet.2020.05.004, http://www.sciencedirect.com/science/article/pii/S0893608020301660 4, 5, 6
15. Kümmerer, M., Wallis, T., Bethge, M.: DeepGaze II: Predicting fixations from deep features over time and tasks. Journal of Vision (JOV) **17**(10), 1147 (2017). https://doi.org/10.1167/17.10.1147, http://arxiv.org/abs/1610.01563 4, 5, 6
16. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022) 3
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision (ECCV). pp. 740–755. Springer (2014) 1, 3
18. Linardos, A., Kümmerer, M., Press, O., Bethge, M.: Deepgaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12919–12928 (2021) 2, 3, 4, 5, 13, 14, 15, 16, 17, 18
19. Liu, N., Han, J.: A deep spatial contextual long-term recurrent convolutional network for saliency detection. IEEE Transactions on Image Processing (TIP) **27**(7), 3264–3274 (2018). https://doi.org/10.1109/tip.2018.2817047 4, 5, 6
20. Mejjati, Y.A., Gomez, C.F., Kim, K.I., Shechtman, E., Bylinskii, Z.: Look here! a parametric learning based approach to redirect visual attention. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 343–361. Springer International Publishing, Cham (2020) 3
21. Miangoleh, S.M.H., Bylinskii, Z., Kee, E., Shechtman, E., Aksoy, Y.: Realistic saliency guided image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 186–194 (2023) 3
22. Pan, J., Sayrol, E., I-Nieto, X., McGuinness, K., OConnor, N.E.: Shallow and deep convolutional networks for saliency prediction. In: IEEE Conference on Computer Vision and Pattern Recognition (2016). https://doi.org/10.1109/cvpr.2016.714 4, 5
23. Peters, R.J., Iyer, A., Itti, L., Koch, C.: Components of bottom-up gaze allocation in natural images. Vision Research **45**(18), 2397–2416 (2005). https://doi.org/10.1016/j.visres.2005.03.019, http://www.sciencedirect.com/science/article/pii/S0042698905001975 4

24. Reddy, N., Jain, S., Yarlagadda, P., Gandhi, V.: Tidying deep saliency prediction architectures. In: International Conference on Intelligent Robots and Systems (IROS) (2020), https://arxiv.org/abs/2003.04942 3, 10
25. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2021) 7
26. Vidyasagar, M.: Kullback-leibler divergence rate between probability distributions on sets of different cardinalities. In: IEEE Conference on Decision and Control (CDC). pp. 948–953 (2010). https://doi.org/10.1109/cdc.2010.5716982 4
27. Yang, S., Lin, G., Jiang, Q., Lin, W.: A dilated inception network for visual saliency prediction. IEEE Transactions on Multimedia **22**(8), 2163–2176 (2020). https://doi.org/10.1109/tmm.2019.2947352 4, 5, 6
28. Zanca, D., Zugarini, A., Dietz, S., Altstidl, T.R., Ndjeuha, M.A.T., Schwinn, L., Eskofier, B.: Contrastive language-image pretrained models are zero-shot human scanpath predictors. arXiv preprint arXiv:2305.12380 (2023) 3