Modeling Saliency Dataset Bias - Appendix

A. Full results on the MIT/Tuebingen Saliency Benchmark

In Tab. 3, 4 and 5 we list the full evaluation on the MIT/Tuebingen Saliency Benchmark for MIT300, CAT2000 and COCO-Freeview, including the metrics that we had to skip in the main text due to space reasons.

B. Comparison with more saliency models

In Table 6 we extend the results table from the main paper with four additional saliency models, which are not DNN based but take inspiration from neuroscience and psychology: Itti & Koch [24] (using the implementation of [20]), RARE2012 [51], GBVS [20] and CovSal [18]. From those models, on most datasets, CovSal performs best with the exception GBVS performs substantially better. Interestingly, on CAT2000, CovSal and GBVS perform substantially better than the deep learning model EML-Net, and RARE2012 performs more similarly, but still better than EML-Net.

C. Bias parameter sensitivity analysis:

In Figure 10, we perform a bias parameter sensitivity analysis: for the our foll model trained jointly on all five datasets, we evaluate each dataset in six different settings: once with each set of dataset bias parameters, and once with the averaged dataset bias parameters. We find that using dataset parameters from a different dataset usually results in a substantial performance drop. This includes the average dataset parameters, however they are usually among the best "wrong" dataset parameters and result in best average performance across datasets. DAEMONS seems to be most different from all other datasets: its dataset parameters result in worst performance on all other datasets. Overall, this analysis shows that the dataset bias parameters control important mechanisms and setting them right can make a large difference in performance.

D. Generalization and adaptation on new datasets

We test our model on three additional datasets. We use the full joint model trained on all five datasets with dataset bias parameters per dataset. The new datasets are tested both in the generalization setting (averaging the dataset bias parameters including the centerbiases) and adaptation (finetuning the dataset bias parameters on the new dataset).

Kienzle dataset: The Kienzle dataset [29] consists of only 200 images which are random crops of grayscale images of natural scenes, making it an challenging testcase. On this dataset, genrealization already results in substantially improved performance compared to earlier models (8% in AUC). Adapting the dataset biase parameters to the Kienzle dataset improves performance further (Tab. 7). In Appendix D we also test the Toronto dataset [5].

Toronto dataset: In Table 8, we apply our model to the Toronto dataset [5]. The Toronto dataset consists of 120 images and hence is too small for training full deep learning models which makes it an interesting test case. On the Toronto dataset, generalization results in improved performance compared to earlier models. Adapting the few dataset parameters to the Toronto dataset improves performance further. Overall, the performance boost, however, is not as large as on the Kienzle dataset. This shows that the Toronto dataset is closer to common saliency datasets and emphasizes the need for new challenging saliency datasets.

SALICON dataset: We also tested our model on the SALICON dataset. To that end, since our default models all are pretrained on SALICON, we trained a new version of the full model without previous pretraining on SALICON and then again tested generalization and adaptation, comparing to full training on SALICON (Tab. 9). We see that the achievable information gain is 0.31 bit/fix. The model trained on our combined dataset (without pretraining on SALICON) and applied with average dataset parameters performs very bad, even slightly worse than the center bias alone. However, adapting the dataset parameters to SALICON results in a performance of 0.26bit/fix, closing 85% of the generalization gap. This is in line with the results from our leave-one-dataset-out generalization test. In the case of SALICON the dataset parameters seem to account for even more of the generalization gap. This might be due to the extremely different experimental setup of SALICON (e.g., mouse traces instead of eye movements, mechanical turk instead of controlled lab environment).

E. Considerations for good centerbias models

On the full datasets, we usually use a centerbias model which is a KDE over all fixations with an additional uniform regularization component. However, in low-data settings, this approach most likely does not average over enough images to result in a good prediction for new images and hence we evaluated different options:

- The modeling approach of the full dataset: a KDE with uniform regularization. Bandwidth and regularization weight are selected for maximum likelihood in a leave-one-image-out crossvalidation setting on the training data.
- A simple parametric model consisting of a centered Gaussian with a uniform regularization component (this is quite close to what many other models use, *e.g.* [15]. Horizontal and vertical variance of the Gaussian as well as the weight of the uniform component are selected to result in maximum likelihood on the training data

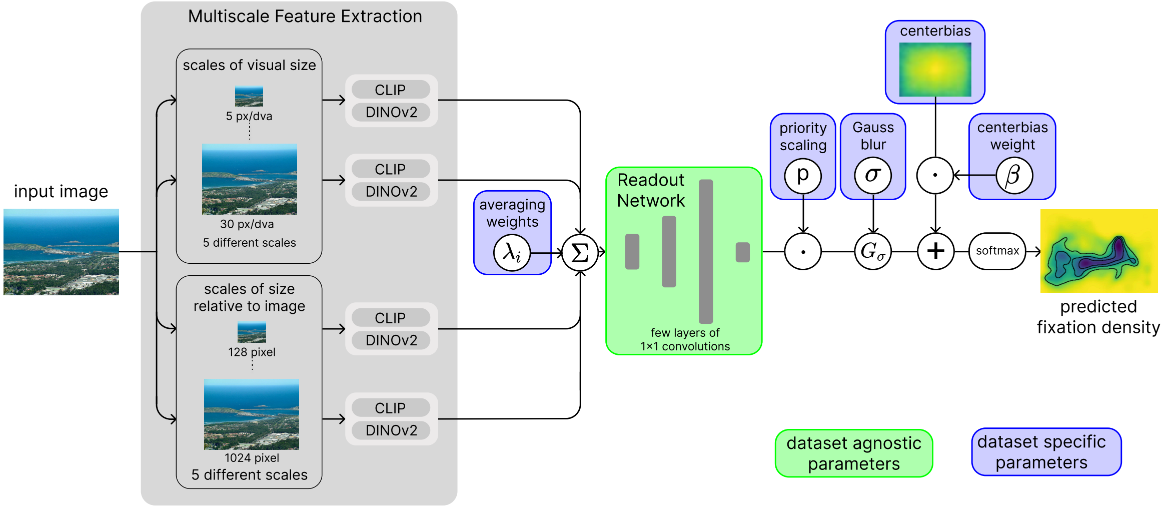

Figure 9. Model Architecture: An input image is rescaled into different resolutions, some defined in total image size in pixels, others in pixels per degree of visual angle. For each image, deep activations from CLIP and DINOv2 encoders are extracted and averaged across scales, from which a priority map is decoded which is then postprocessed with Blur, priority scaling and centerbias

Table 3. MIT300 Benchmark

Model	IG	AUC	sAUC	NSS	CC	KLDiv	SIM
SalTR	-	-	0.7900	2.4500	0.8000	0.3600	0.5900
TempSAL	-	0.8626	0.7483	2.0092	0.7181	0.5509	0.6202
DeepGaze II	0.9505	0.8759	0.7840	2.3689	0.7851	0.4149	0.6746
EML-NET	-	0.8762	0.7469	2.4876	0.7893	0.8439	0.6756
SalFBNet	0.8194	0.8769	0.7858	2.4702	0.8141	0.4151	0.6933
UNISAL	0.9505	0.8772	0.7840	2.3689	0.7851	0.4149	0.6746
GSGNet	-	0.8780	0.7880	2.4230	0.8110	0.4100	0.6900
Clueify	-	0.8811	0.7651	1.4946	0.5750	0.8885	0.4773
DeepGaze IIE	1.0715	0.8829	0.7942	2.5265	0.8242	0.3474	0.6993
Ours (generalized)	1.1975	0.8926	0.8139	2.6697	0.8665	0.2791	0.7311
Ours (adapted)	1.2355	0.8936	0.8149	2.7229	0.8795	0.2588	0.7478
Ours (full joint training)	**1.2463**	**0.8942**	**0.8159**	**2.7439**	**0.8832**	**0.2540**	**0.7518**
Interobserver consistency	1.3239	0.8982	-	2.8481	-	-	-

Table 4. CAT2000 Benchmark

Model	IG	AUC	sAUC	NSS	CC	KLDiv	SIM
TempSAL	-	0.8444	0.6378	1.7037	0.6607	0.6282	0.6173
SalFBNet	-	0.8549	0.6330	1.8791	0.7028	1.2004	0.6426
ICF	-0.0229	0.8561	0.6187	1.9588	0.7791	0.4448	0.6697
UNISAL	0.0321	0.8604	0.6684	1.9359	0.7399	0.4703	0.6633
DeepGaze II	0.0839	0.8640	0.6498	1.9619	0.7950	0.3815	0.6865
DeepGaze IIE	0.1893	0.8692	0.6677	2.1122	0.8189	0.3448	0.7060
Ours (generalized)	0.2031	0.8712	0.6889	2.1460	0.8176	0.3397	0.7200
Ours (adapted)	0.4333	0.8806	0.6900	2.4591	0.8997	0.2430	0.7726
Ours (full joint training)	**0.4932**	**0.8847**	**0.7002**	**2.5127**	**0.9155**	**0.2098**	**0.7891**
Interobserver consistency	0.4730	0.8840	0.6930	2.4878	-	-	-

Table 5. COCO Freeview Benchmark

Model	IG	AUC	sAUC	NSS	CC	KLDiv	SIM
TempSAL	-	0.8567	0.7076	1.7508	0.6473	0.7026	0.5626
DeepGaze II	0.6636	0.8699	0.7399	2.0028	0.6909	0.5858	0.6043
SalFBNet	-	0.8722	0.7099	2.0275	0.7088	0.8623	0.6178
UNISAL	0.7494	0.8774	0.7585	2.0954	0.7155	0.5515	0.6203
DeepGaze IIE	0.8596	0.8825	0.7669	2.2558	0.7563	0.4863	0.6447
Ours (generalized)	0.9475	0.8896	0.7855	2.3782	0.7907	0.4331	0.6695
Ours (adapted)	1.0114	0.8932	0.7884	2.4413	0.8048	0.4078	0.6805
Ours (full joint training)	**1.0727**	**0.8968**	**0.7951**	**2.5251**	**0.8258**	**0.3743**	**0.6882**
Interobserver consistency	0.8673	0.8829	-	2.2837	-	-	-

Table 6. Performance of our model and previous state-of-the-art models. Best performance in is indicated in bold, second best is underlined. "generalization" refers to training on the respective four other datasets and evaluation with average dataset biases, "adaptation" refers to training on the respective four other datasets and evaluation after adapting the dataset bias parameters to the target dataset. Models are sorted by average AUC.

Model	MIT1003		CAT2000		COCO Freeview		DAEMONS		FIGRIM		average	
	IG	AUC	IG	AUC	IG	AUC	IG	AUC	IG	AUC	IG	AUC
Itti & Koch	-	0.757	-	0.759	-	0.702	-	0.699	-	0.766	-	0.736
RARE2012	-	0.772	-	0.777	-	0.771	-	0.706	-	0.787	-	0.762
GBVS	-	0.803	-	0.802	-	0.796	-	0.710	-	0.821	-	0.786
CovSal	-	0.809	-	0.847	-	0.803	-	0.679	-	0.835	-	0.795
EML-NET	-	0.842	-	0.766	-	0.817	-	0.766	-	0.832	-	0.805
SalFBNet	-	0.883	-	0.858	-	0.868	-	0.774	-	0.886	-	0.854
UNISAL	1.006	0.887	0.099	0.865	0.712	0.873	0.712	0.809	0.771	0.892	0.660	0.865
our model, generalization	1.172	0.902	0.249	0.878	0.889	0.886	0.538	0.800	0.883	0.905	0.746	0.874
DeepGaze IIE	1.113	0.894	0.315	0.878	0.846	0.881	1.006	0.822	0.877	0.899	0.831	0.875
our model w/o biases, generalization	1.123	0.898	0.259	0.879	0.897	0.887	0.625	0.808	0.954	0.907	0.772	0.876
our model, adaptation	1.217	0.904	0.469	0.887	0.965	0.890	1.149	0.840	1.059	0.911	0.972	0.886
our model, trained on all	**1.240**	**0.905**	0.522	0.891	1.031	0.895	1.258	0.848	**1.117**	**0.915**	1.034	0.891
our model, trained per dataset	1.217	0.903	**0.535**	**0.891**	**1.040**	**0.895**	**1.272**	**0.850**	1.105	0.914	**1.034**	**0.891**
Gold Standard (subject-LOO)	1.213	0.901	0.494	0.885	0.869	0.880	1.347	0.850	1.054	0.907	0.995	0.885
Gold Standard (upper bound)	1.829	0.945	0.873	0.920	1.511	0.935	1.722	0.899	1.642	0.947	1.515	0.929

- A combination of the two previous options: A mixture of a KDE, a centered Gaussian and a uniform component. Horizontal and vertical bandwidth of the Gaussian are computed on the training fixations. The bandwidth of the KDE and the mixture weights are selected for maximum likelihood in a leave-one-image-out crossvalidation setting on the training data.

For each of our five datasets and random subsets thereof, we fit the different models and evaluate on the corresponding validation splits. The results are visible in Fig. 12. We see that the first option ("KDE") sometimes results in bad scores if little data is available. The second option ("Gaussian + uniform") performs much better in these cases but fails to reach the performance of the nonparametric centerbias with more data. The third option ("KDE+Gaussian+uniform") combines the advantages of both: reasonable performance already with a few images and good convergence with more data. Interestingly, whether the KDE or Gaussian+uniform performs better for low data is different from dataset to dataset. This is why we select the third option for our low-data adaptation experiments.

These results also serve to demonstrate that modeling the centerbias as a simple Gaussian is not sufficient for many datasets and can result in substantial performance penalties (see also Fig. 13).

F. Multiscale ablation

We evaluated the benefits of our multiscale feature extraction state in an ablation study where we trained the jointly trained model in different settings: we varied whether the model used absolute, relative or both scales. We also varied the numbers of scale per type and whether we added scales starting with the low or high resolutions. We find that the large scales are crucial for performance (Fig. 11a, b). We also evaluated computational demand via epoch times and find acceptable performance tradeoffs with fewer but large, preferably relative, scales, resulting in a few percent perfor-

Table 7. Kienzle Dataset

Model	IG	AUC	NSS	CC	KLDiv
EML-NET	-	0.677	0.648	0.314	1.058
UNISAL	0.510	0.817	1.770	0.648	0.628
DeepGaze IIE	0.662	0.819	2.048	0.692	0.549
our model, average parameters	1.499	0.895	2.596	**0.879**	0.284
our model, fine-tuned dataset parameters	**1.509**	**0.896**	**2.603**	0.879	**0.281**

Table 8. Toronto Dataset

Model	IG	AUC	NSS	CC	KLDiv
EML-NET	-	0.847	2.098	0.719	2.734
UNISAL	0.846	0.885	2.360	0.812	0.396
DeepGaze IIE	1.004	0.892	2.572	0.859	0.330
our model, average parameters	1.080	0.896	2.629	**0.879**	0.284
our model, fine-tuned dataset parameters	**1.092**	**0.897**	**2.640**	0.879	**0.281**

Table 9. SALICON Dataset

Model	IG
Our model, average parameters	-0.03
Our model, adapted parameters	0.26
Our model, trained on SALICON	0.31

mance drop (Fig. 11c).

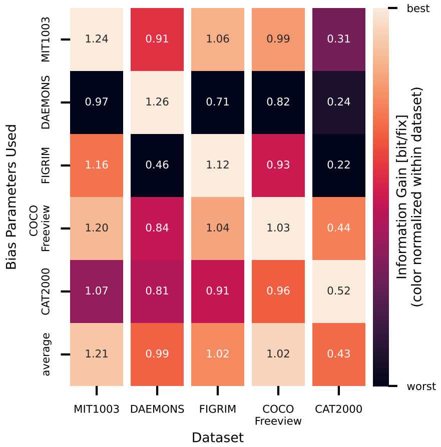

Figure 10. Parameter sensitivity analysis: We evaluate the full model trained jointly across all five dataset on each dataset with the dataset parameters from all datasets and average dataset parameters.

G. The joint model excels at hard images

In Figure 15, we compare the models on a per-image-level. We quantify model performance in terms of the information gain difference to the gold standard, i.e., we measure an prediction error: how much explainable information gain is missed by the the models. This reveals that the joint model profits most from those images where the individual models make the largest prediction error, which means that it performs better on hard images.

We now analyze model predictions on specific images. From each dataset, we select those images where the jointly trained model outperforms the individually trained models most and visualize the model predictions. In addition, we also visualize the *pixelwise information gain difference* [35]: For each pixel, we visualize $p_{\text{gold}} \left(\log p_{\text{joint}} - \log p_{\text{individual}} \right)$. This visualization technique results in highlighting those image areas where predictions differ in a relevant way and makes comparing model predictions more intuitive. For more details on pixelwise information gain, see [35].

The resulting images are shown in Figure 16. We see that the joint model often is better at detecting the exact outline of salient objects (MIT1003, first image), at predicting which one of two salient objects is more important (CAT2000, first image, where more salience is moved to the bird compared to the structure in the foreground; also MIT1003, third image, where the map fragements in the center are downweighted and the peripheral text is upweighted). Also, it appears that the joint model is better at capturing the interplay between local image salience and centerbias, sometimes increasing saliency in the periphery (MIT1003, third image) and sometimes increasing saliency in the center (CAT2000, first image and FIGRIM, first im-

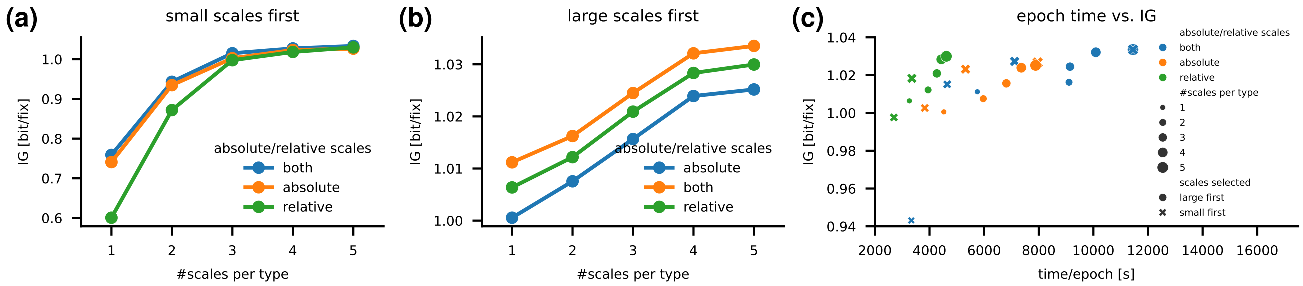

Figure 11. Multiscale ablation: prediction performance depending on the number of scales in the multiscale backbone and whether the scale weights are dataset agnostic or dataset specific

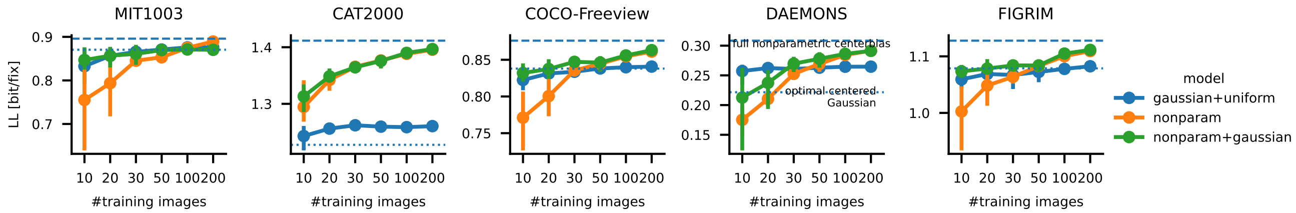

Figure 12. Different centerbias modeling strategies and their performance in low-data settings: we compare three different model classes for centerbias models and how well they perform depending on the number of images used to fit them. Error bars indicate 95% intervals over multiple runs with different random subsets.

age). Since the model architecture can be seen as computing a posterior from the local image salience as likelihood and the centerbias as prior, this suggests that the joint model manages to extract more evidence from the image features, resulting in overwriting the centerbias more often.

H. Details about the model architecture

Mathematical model formulation Given an image I, we denote with $R_{k,j}(I)$ the rescaled images ($k = 1, 2$ differentiates between the two scale type “absolute” and “relative”, and $j = 1 \dots 5$ indexes the specific resolutions). We extract deep features $F_{k,j}(I) = F(R_{k,j}(I))$ with our backbone F. Given weights $\lambda_{k,j} \geq 0$, $\sum_{k,j} \lambda_{k,j} = 1$ we then compute the averaged deep features $\bar{F} = \sum_{k,j} \lambda_{k,}(R'(F_{k,j}))$, where R' indicates a rescaling operation that rescales all deep features to the same resolution. From $\bar{F}$, the readout network RN computes a spatial priority map $S = RN(\bar{F})$, which is postprocessed with the priority scaling p, the Gauss blur size σ, the center bias distribution $p_{\text{cb}}(x \mid I)$ and the center bias weight β to yield the prediction $\hat{p}(x \mid I) = \operatorname{softmax}(\mathcal{G}_\sigma(p \cdot S) + \beta \log p_{\text{cb}}(x \mid I))$.

Multiscale resolutions We use a total of 10 scales in our multiscale feature extraction. Five scales are resizing the input image to match a certain resolution in terms of pixel per degree of visual angle and use resolutions of 5, 10, 17.5, 24 and 30 px/dva. The other five scales are resizing the input image to match a certain image width or height (whatever is larger) in terms of pixel and uses sizes of 128, 256, 512, 768 and 1024 pixels. Before averaging extracted features across scales, we rescale all of them to 1/8th of the original image resolution to achieve matching sizes. The rescaling operation uses bilinear interpolation.

The scales were chosen to include 17.5 px/dva which is the scale of DeepGaze IIE (MIT1003 has 35px/dva, and DeepGaze IIE downsamples by a factor of 2). From there on we added larger scales until we ran into computational constraints, and smaller scales to the point that we still considered sensible. For the relative scales, the approach was similar: 512 pixel corresponds to the resolution that DeepGaze IIE uses internally on its training dataset, from there on we added smaller and larger scales. In an Ablation (see Appendix F), we found that including the larger scales is crucial for improving prediction performance.

CLIP and DINO We use the implementations and checkpoints from `https://github.com/openai/CLIP` and `https://github.com/facebookresearch/dinov2`. In the case of CLIP, we use the ResNet50x64 architecture and extract the layer `layer4.2.conv2`. In the case of DINOv2, we use the ViTB14 architecture and extract the layers `blocks.6` and `blocks.10`. In total, this gives us 2560. To regain spatial feature maps from the ViT tokens, we rearange the tokens from the deep layers back into their original layout in the input image. Depending on the image size, we might have differently sized feature maps (for the convolutional CLIP implementation)

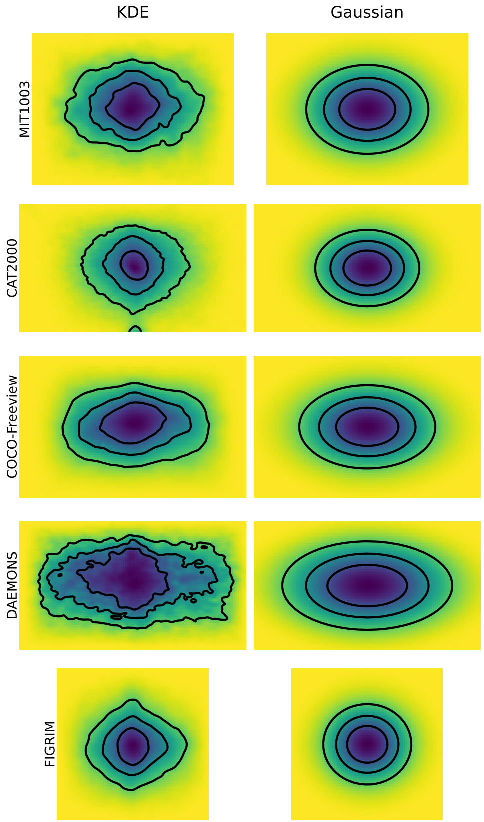

Figure 13. Different centerbias models: We compare a nonparametric centerbias with a centered Gaussian and see that the centered Gaussian often misses a lot of structure present in the average spatial fixation distribution across images.

or different numbers of tokens (for the transformer based DINOv2 implementation). This is not a problem since we don't require the original readout layers and can simply remove them. The extracted deep layers have been choosen with a random search on MIT1003. Interestingly, we found that for CLIP, the convolutional backbones worked better, while in the case of DINOv2, the ViT based backbones resulted in higher performance. Hence we use the ResNet implementation of CLIP and the ViT implementation of DINOv2.

Readout Network The readout network consists of five layers of 1x1 convolutions, processing the 2560 feature maps from the multiscale encoder. The five layers of the readout network produce 8, 16, 1, 128, and 1 feature maps respectively. Each layer is prepended by a layer norm and uses softplus as activation function.

Gaussian blur The output of the readout network is upscaled to 1/2 of the original image resolution before the Gaussian blurring is applied, which is specified in degree of visual angle.

I. Details about the datasets

All datasets except for DAEMONS are accessed via their wrapper in the pysaliency python library https://github.com/matthias-k/pysaliency. Since DAEMONS is a very new dataset, it's not yet included in pysaliency and we had to write our own pysaliency wrapper. MIT1003, CAT2000 and FIGRIM don't come with official validation splits, here we create our own using `pysaliency.filter_datasets.train,validation_fold(s fixations, crossval_folds=10, test_folds=0, val_folds=1)`. For CAT2000, we furthermore specify `stratified_attributes=['category']` to guarantee a uniform distribution of the image categories over splits.

J. Details about the training

We use the Adam optimizer for optimizing models together with a learning rate schedule consisting of decays of the learning rate by a factor of 10. For each dataset, initial learning rate and points for first and second decay have been selected with a random search. Third and fourth decay always happen after one additional epoch, after the fourth decay training is stopped. The specific learning rate schedules are given in Table 10.

Pretraining on the SALICON dataset [26], uses the mouse data from the 2017 SALICON edition. To save compute, for the pretraining we use only one scale with 1024 pixels.

K. Baseline models

We include two baseline models to put model performances into perspective: the *centerbias model* is a KDE which, for each image, uses the fixation locations from all other images in the dataset. The centerbias quantifies how well fixations can be predicted without taking image content into account. Bandwidth and a uniform regularization component have been selected for maximum likelihood using leave-one-image-out crossvalidation. For each image in the combined dataset, we use the centerbias prediction from the respective dataset centerbias.

The *gold standard model* estimates inter-observer consistency. As suggested by [32], we use a mixture of a uniform component, the centerbias model and a KDE. The latter uses, for each observer, the fixations from all other observers on the same image. Mixture weights and KDE bandwidth have been chosen for maximum likelihood, where the parameters are fitted for each image individually to make sure that the prediction is as good as possible per image. Unless otherwise indicated, we specify the gold standard performance as the leave-on-subject-out crossvalidation performance. For some figures, we specify the gold

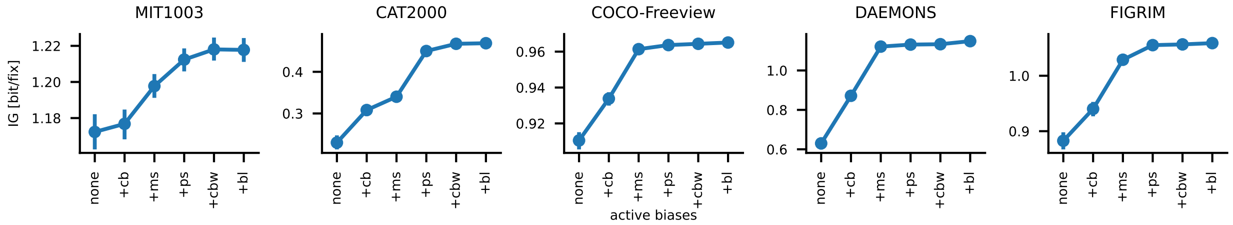

Figure 14. Contribution of different biases to closing the generalization gap, split up target dataset. cb=centerbias, ms=multiscale weights, ps=priority scaling, cbw=centerbias weight, bl=blur size

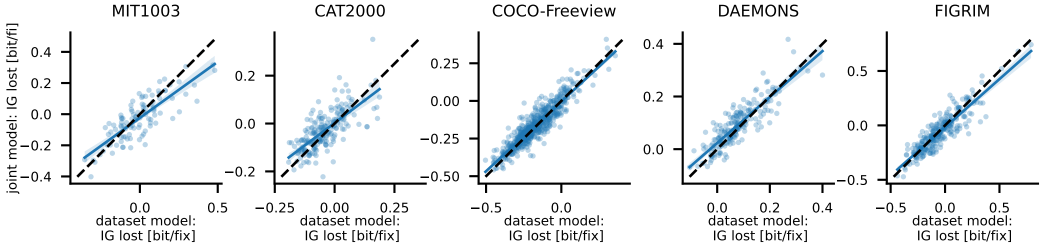

Figure 15. The model trained across dataset tends to perform better at hard images: For each dataset (subplots) we plot prediction error for the individually trained model (x-axis) and the jointly trained model (y-axis) for each image in the validation part of the respective dataset (points). Prediction error is quantified as information gain difference to the gold standard performance. It can be seen that for images where the models trained on only one dataset (right end of the x axis), the joint model tends to perform better than the individually trained model.

standard as a range ranging from the leave-on-subject-out crossvalidation performance up to the performance when including all image fixations in the KDE but keeping the parameters fitted in the crossvalidation. The first is essentially a lower bound on inter-observer consistency, the latter is an upper bound.

L. CAT2000 artifacts

We noticed that the CAT2000 dataset contains an artifact: for some scanpaths, all fixations are all clustered in a small image area far from the image center. All these scanpaths are from the same subject, indicating eye tracking problems with this subject. For this reason, we excluded all these scanpaths from the dataset by removing all scanpaths from subject number 20 with a mean y position of larger than 950 pixels. Extensive visual tests confirmed that this indeed removes those and only those scanpaths (see Figure 17 for example cases)

M. Assets

Our models where implemented in python using pytorch [46]. Model evaluations and saliency metrics were using the public pysaliency toolbox (`github.com/matthias-k/pysaliency`, MIT license). All datasets except for DAEMONS were used via their pysaliency wrapper. The models were used via their implementations from `https://github.com/rdroste/unisal` (Apache 2 license), `https://github.com/SenJia/EML-NET-Saliency`, `https://github.com/gqding/SalFBNet` and `https://github.com/matthias-k/deepgaze`. Also used were scipy [66] and numpy [21] for computations, pandas [50] for statistics and data handling as well as matplotlib [23] and seaborn [68] for plotting.

N. Compute Ressources

All main experiments where conducted on A100s. Model trainings on individual datasets took around 6–24 hours, trainings on the combined dataset around 3-4 days. The learning rate random search was conducted using an earlier model version on 2080Ti GPUs. Around 500 random search iterations were performed taking on average 5 hours each.

References

[1] Hossein Adeli, Françoise Vitu, and Gregory J. Zelinsky. A Model of the Superior Colliculus Predicts Fixation Locations during Scene Viewing and Visual Search. *The Journal of Neuroscience*, 37(6):1453–1467, 2017. 1

[2] Bahar Aydemir, Ludo Hoffstetter, Tong Zhang, Mathieu Salzmann, and Sabine Süsstrunk. TempSAL –

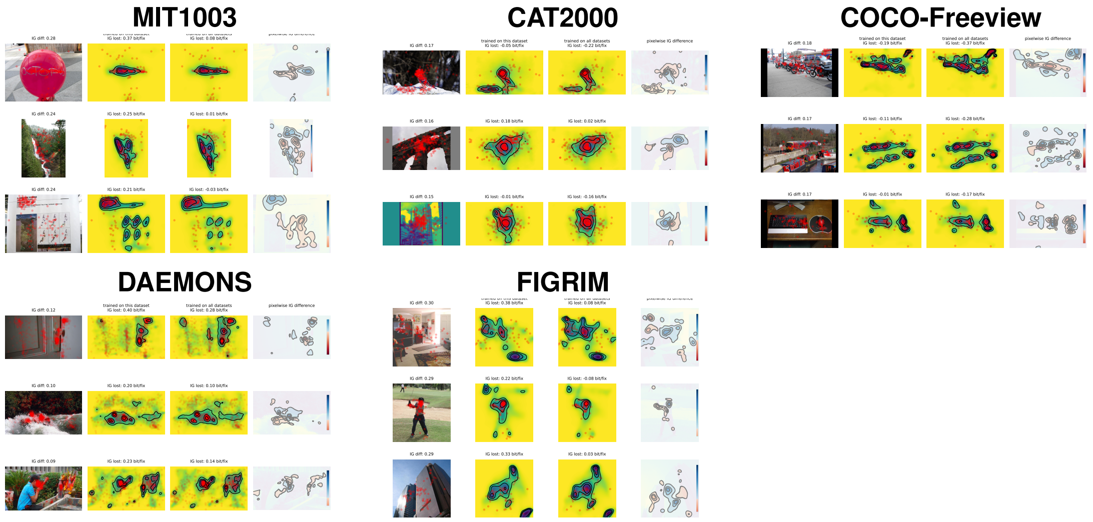

Figure 16. Example predictions of individually and joint trained models. For each dataset, we show the three images where the difference in performance between the joint model and the individual models is largest.

Table 10. Learning rate schedule for each dataset

Dataset	initial learning rate	decay epochs
MIT1003	0.005623	3, 9, 10, 11
CAT2000	0.01	6, 9, 10, 11
COCO Freeview	0.01	12, 15, 16, 17
DAEMONS	0.005012	12, 15, 16, 17
FIGRIM	0.01	9, 15, 16, 17
Combined	0.001585	15, 21, 22, 23
SALICON (pretraining)	0.01	3.75, 7.5, 11.25

Uncovering Temporal Information for Deep Saliency Prediction, 2024. 2

[3] Bahar Aydemir, Deblina Bhattacharjee, Tong Zhang, Mathieu Salzmann, and Sabine Süsstrunk. Data Augmentation via Latent Diffusion for Saliency Prediction. In *Computer Vision – ECCV 2024*, pages 360–377. Springer Nature Switzerland, Cham, 2025. 2

[4] Ali Borji and Laurent Itti. CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research. *CVPR 2015 workshop on "Future of Datasets"*, 2015. 1, 3

[5] Neil D. B. Bruce and John K. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, 9(3):5–5, 2009. 2, 13

[6] Zoya Bylinskii, Phillip Isola, Constance Bainbridge, Antonio Torralba, and Aude Oliva. Intrinsic and extrinsic effects on image memorability. *Vision Research*, 116:165–178, 2015. 3

[7] Kai-Yueh Chang, Tyng-Luh Liu, Hwann-Tzong Chen, and Shang-Hong Lai. Fusing generic objectness and visual saliency for salient object detection. In *2011 International Conference on Computer Vision*, pages 914–921, 2011. 2

[8] Shi Chen, Ming Jiang, and Qi Zhao. What Do Deep Saliency Models Learn about Visual Attention? *Advances in Neural Information Processing Systems*, 36: 9543–9555, 2023. 2

[9] Yupei Chen, Zhibo Yang, Seoyoung Ahn, Dimitris Samaras, Minh Hoai, and Gregory Zelinsky. COCO-Search18 fixation dataset for predicting goal-directed attention control. *Scientific Reports*, 11(1):8776, 2021. 1

[10] Yupei Chen, Zhibo Yang, Souradeep Chakraborty, Sounak Mondal, Seoyoung Ahn, Dimitris Samaras, Minh Hoai, and Gregory Zelinsky. Characterizing Target-Absent Human Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5031–5040, 2022. 1, 3

[11] Denis Cousineau. Confidence intervals in within-subject designs: A simpler solution to Loftus and Masson's method. *Tutorials in Quantitative Methods for Psychology*, 1(1):42–45, 2005. 4, 6

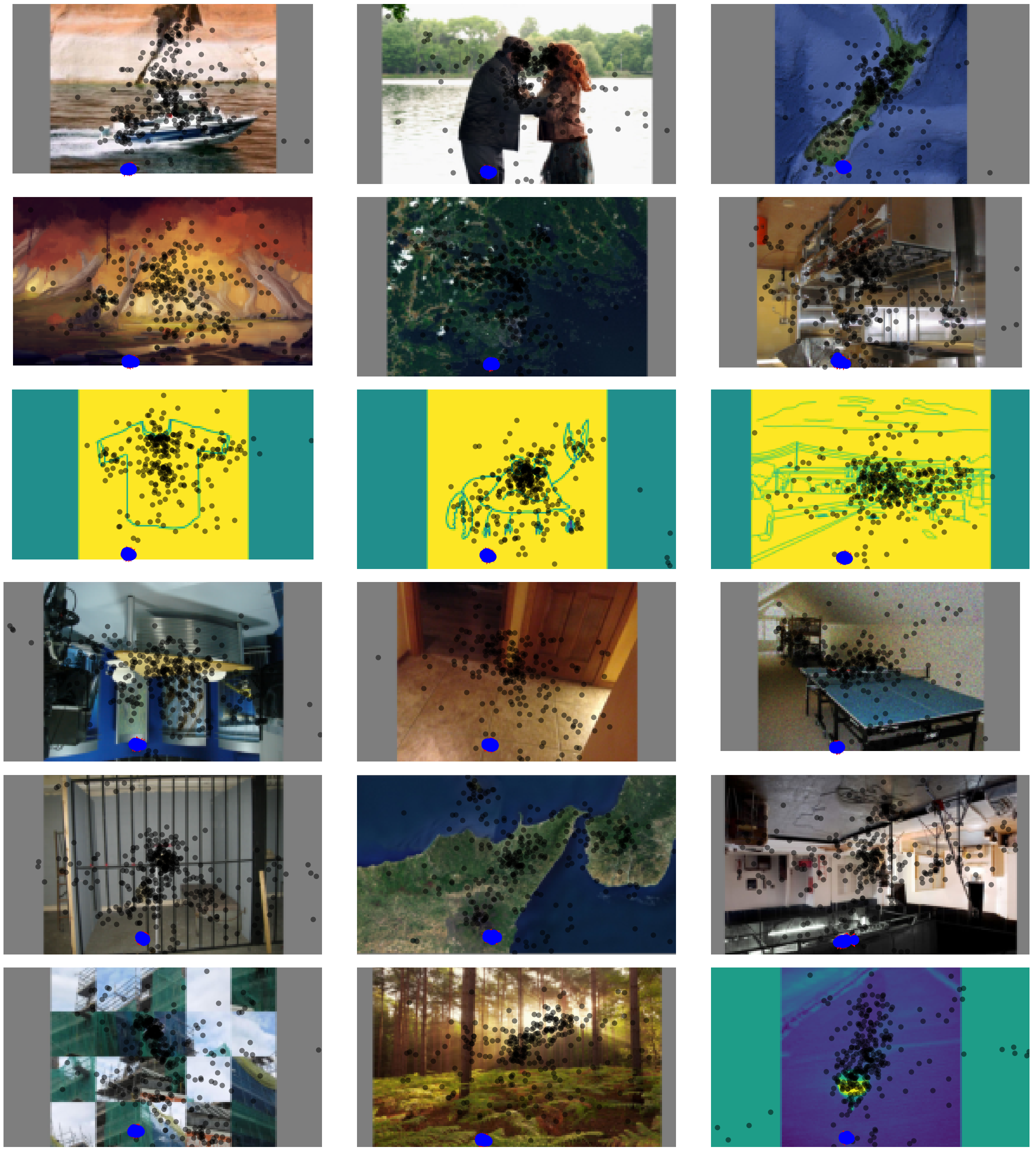

Figure 17. Artifacts in the CAT2000 dataset: some scanpaths have all fixations clustered in always the same image area. This likely indicates eye tracking problems and hence we excluded the scanpaths. The fixtions of the respective scanpath are shown in blue, for comparison all fixations from other subjects are additionaly shown in black.

[12] Benjamin de Haas, Alexios L. Iakovidis, D. Samuel Schwarzkopf, and Karl R. Gegenfurtner. Individual differences in visual salience vary along semantic dimensions. *Proceedings of the National Academy of Sciences*, page 201820553, 2019. 8

[13] Guanqun Ding, Nevrez İmamoğlu, Ali Caglayan, Masahiro Murakawa, and Ryosuke Nakamura. SalFBNet: Learning pseudo-saliency distribution via feedback convolutional networks. *Image and Vision Computing*, 120:104395, 2022. 2, 4

[14] Yasser Abdelaziz Dahou Djilali, Kevin McGuiness, and Noel O'Connor. Learning Saliency From Fixations, 2023. 2

[15] Richard Droste, Jianbo Jiao, and J. Alison Noble. Unified Image and Video Saliency Modeling. In *Computer Vision – ECCV 2020*, pages 419–435, Cham, 2020. Springer International Publishing. 1, 2, 4, 13

[16] Wolfgang Einhäuser, Merrielle Spain, and Pietro Perona. Objects predict fixations better than early saliency. *Journal of Vision*, 8(14):18–18, 2008. 3

[17] Ralf Engbert, Hans A. Trukenbrod, Simon Barthelmé, and Felix A. Wichmann. Spatial statistics and attentional dynamics in scene viewing. *Journal of Vision*, 15(1):14–14, 2015. 1

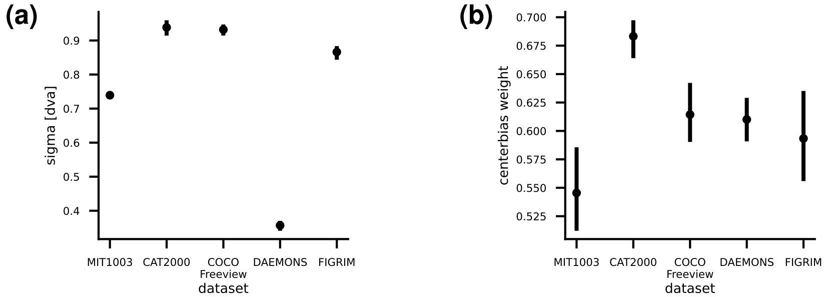

Figure 18. Dataset specific parameters. (a): Bandwidth of Gaussian blur per dataset in degree of visual angle (dva). (b) centerbias weight per dataset. The reported errors are bootstraped 95% confidence intervals of the mean across four random seeds.

[18] Erkut Erdem and Aykut Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision*, 13(4):11–11, 2013. 13

[19] Adhiraj Ghosh, Sebastian Dziadzio, Ameya Prabhu, Vishaal Udandarao, Samuel Albanie, and Matthias Bethge. ONEBench to Test Them All: Sample-Level Benchmarking Over Open-Ended Capabilities, 2024. 8

[20] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-Based Visual Saliency. In *Advances in Neural Information Processing Systems 19*, pages 545–552. MIT Press, 2007. 2, 13

[21] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825): 357–362, 2020. 19

[22] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. SALICON: Reducing the Semantic Gap in Saliency Prediction by Adapting Deep Neural Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 262–270, Santiago, Chile, 2015. IEEE. 2

[23] John D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. 19

[24] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998. 2, 13

[25] Sen Jia and Neil D. B. Bruce. EML-NET: An Expandable Multi-Layer NETwork for saliency prediction. *Image and Vision Computing*, 95:103887, 2020. 2, 4

[26] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. SALICON: Saliency in Context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1072–1080, 2015. 4, 18

[27] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th International Conference On*, pages 2106–2113. IEEE, 2009. 1, 2, 3

[28] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? Investigating their struggle with spatial reasoning, 2023. 3

[29] Wolf Kienzle, Matthias O. Franz, Bernhard Schölkopf, and Felix A. Wichmann. Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of Vision*, 9(5):7–7, 2009. 2, 13

[30] Alexander Kroner, Mario Senden, Kurt Driessens, and Rainer Goebel. Contextual encoder–decoder network for visual saliency prediction. *Neural Networks*, 129: 261–270, 2020. 2

[31] S. S. S. Kruthiventi, K. Ayush, and R. V. Babu. DeepFix: A Fully Convolutional Neural Network for Predicting Human Eye Fixations. *IEEE Transactions on Image Processing*, 26(9):4446–4456, 2017. 2

[32] Matthias Kümmerer and Matthias Bethge. Predicting Visual Fixations. *Annual Review of Vision Science*, 9 (1):269–291, 2023. 1, 4, 8, 18

[33] Matthias Kümmerer, Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durant, Aude Oliva, Antonio Torralba, and Matthias Bethge. MIT/Tuebingen Saliency Benchmark. saliency.tuebingen.ai. 1, 5

[34] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. Deep Gaze I: Boosting Saliency Prediction with Feature Maps Trained on ImageNet. *ICLR Workshop Track*, 2015. 2, 7

[35] Matthias Kümmerer, Thomas S. A. Wallis, and Matthias Bethge. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112(52):16054–16059, 2015. 4, 7, 16

[36] Matthias Kümmerer, Thomas S. A. Wallis, Leon A. Gatys, and Matthias Bethge. Understanding Low- and High-Level Contributions to Fixation Prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 4789–4798, 2017. 2, 3, 7

[37] Matthias Kümmerer, Thomas S. A. Wallis, and Matthias Bethge. Saliency Benchmarking Made Easy: Separating Models, Maps and Metrics. In *Computer Vision – ECCV 2018*, pages 798–814. Springer International Publishing, 2018. 4

[38] Peizhao Li, Junfeng He, Gang Li, Rachit Bhargava, Shaolei Shen, Nachiappan Valliappan, Youwei Liang, Hongxiang Gu, Venky Ramachandran, Golnaz Farhadi, Yang Li, Kai J. Kohlhoff, and Vidhya Navalpakkam. UniAR: A Unified model for predicting human Attention and Responses on visual content. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[39] Akis Linardos, Matthias Kümmerer, Ori Press, and Matthias Bethge. DeepGaze IIE: Calibrated Prediction in and Out-of-Domain for State-of-the-Art Saliency Modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12919–12928, 2021. 2, 3, 4, 7

[40] Panagiotis Linardos, Eva Mohedano, Juan Jose Nieto, Noel E. O'Connor, Xavier Giro-i-Nieto, and Kevin McGuinness. Simple vs complex temporal recurrences for video saliency prediction, 2019. 1

[41] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual Spatial Reasoning, 2023. 3

[42] Zhuang Liu and Kaiming He. A Decade's Battle on Dataset Bias: Are We There Yet?, 2024. 3

[43] Jianxun Lou, Hanhe Lin, David Marshall, Dietmar Saupe, and Hantao Liu. TranSalNet: Towards perceptually relevant visual saliency prediction. *Neurocomputing*, 494:455–467, 2022. 2

[44] Richard D. Morey. Confidence Intervals from Normalized Data: A correction to Cousineau (2005). *Tutorials in Quantitative Methods for Psychology*, 4(2): 61–64, 2008. 4, 6

[45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2023. 3

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019. 19

[47] Ameya Prabhu, Vishaal Udandarao, Philip Torr, Matthias Bethge, Adel Bibi, and Samuel Albanie. Lifelong Benchmarks: Efficient Model Evaluation in an Era of Rapid Progress, 2024. 8

[48] Michael J. Proulx and Monique Green. Does apparent size capture attention in visual search? Evidence from the Müller–Lyer illusion. *Journal of Vision*, 11(13): 21, 2011. 3

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *arXiv:2103.00020 [cs]*, 2021. 3

[50] Jeff Reback, jbrockmendel, Wes McKinney, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, Simon Hawkins, Matthew Roeschke, gfyoung, Sinhrks, Adam Klein, Patrick Hoefler, Terji Petersen, Jeff Tratner, Chang She, William Ayd, Shahar Naveh, J. H. M. Darbyshire, Marc Garcia, Richard Shadrach, Jeremy Schendel, Andy Hayden, Daniel Saxton, Marco Edward Gorelli, Fangchen Li, Matthew Zeitlin, Vytautas Jancauskas, Ali McMaster, Pietro Battiston, and Skipper Seabold. Pandas-dev/pandas: Pandas 1.4.1. Zenodo, 2022. 19

[51] Nicolas Riche, Matei Mancas, Matthieu Duvinage, Makiese Mibulumukini, Bernard Gosselin, and Thierry Dutoit. RARE2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. *Signal Processing: Image Communication*, 28(6):642–658, 2013. 2, 13

[52] Constantin A. Rothkopf, Dana H. Ballard, and Mary M. Hayhoe. Task and context determine where you look. *Journal of Vision*, 7(14):16–16, 2007. 1

[53] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems*, pages 11539–11551. Curran Associates, Inc., 2020. 3

[54] Heiko H. Schütt, Lars O. M. Rothkegel, Hans A. Trukenbrod, Ralf Engbert, and Felix A. Wichmann. Disentangling bottom-up versus top-down and low-level versus high-level influences on eye movements over time. *Journal of Vision*, 19(3):1–1, 2019. 7

[55] Lisa Schwetlick, Lars Oliver Martin Rothkegel, Hans Arne Trukenbrod, and Ralf Engbert. Modeling the effects of perisaccadic attention on gaze statistics during scene viewing. *Communications Biology*, 3(1): 1–11, 2020. 1

[56] Lisa Schwetlick, Daniel Backhaus, and Ralf Engbert. A dynamical scan-path model for task-dependence during scene viewing. *Psychological Review*, 130(3): 807–840, 2023. 1

[57] Lisa Schwetlick, Matthias Kümmerer, Matthias Bethge, and Ralf Engbert. Potsdam data set of eye movement on natural scenes (DAEMONS). *Frontiers in Psychology*, 15, 2024. 3

[58] Benjamin W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14):4–4, 2007. 3

[59] Benjamin W. Tatler, Roland J. Baddeley, and Iain D. Gilchrist. Visual correlates of fixation selection: Effects of scale and time. *Vision Research*, 45(5):643–659, 2005. 4

[60] Chakkrit Termritthikun, Ayaz Umer, Suwichaya Suwanwimolkul, Feng Xia, and Ivan Lee. SalNAS: Efficient Saliency-prediction Neural Architecture Search with self-knowledge distillation. *Engineering Applications of Artificial Intelligence*, 136: 109030, 2024. 2

[61] Chakkrit Termritthikun, Ayaz Umer, Suwichaya Suwanwimolkul, and Ivan Lee. Semi-PKD: Semi-supervised Pseudoknowledge Distillation for saliency prediction. *ICT Express*, 11(2):364–370, 2025. 2

[62] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528, 2011. 3

[63] Antonio Torralba, Aude Oliva, Monica S. Castelhano, and John M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological Review*, 113(4):766–786, 2006. 2

[64] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980. 2

[65] Eleonora Vig, Michael Dorr, and David Cox. Large-Scale Optimization of Hierarchical Features for Saliency Prediction in Natural Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2798–2805, 2014. 2

[66] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, pages 1–12, 2020. 19

[67] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *International Conference on Learning Representations*, 2020. 3

[68] Michael Waskom. Seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. 19

[69] Niklas Wilming, Torsten Betz, Tim C. Kietzmann, and Peter König. Measures and Limits of Models of Fixation Selection. *PLOS ONE*, 6(9):e24038, 2011. 8

[70] Calden Wloka, Iuliia Kotseruba, and John K. Tsotsos. Active Fixation Control to Predict Saccade Sequences. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3184–3193, Salt Lake City, UT, USA, 2018. IEEE. 1

[71] Jiawei Xie, Zhi Liu, Gongyang Li, Xiaofeng Lu, and Tao Chen. Global semantic-guided network for saliency prediction. *Knowledge-Based Systems*, 284: 111279, 2024. 2

[72] Zhibo Yang, Sounak Mondal, Seoyoung Ahn, Gregory Zelinsky, Minh Hoai, and Dimitris Samaras. Predicting Human Attention using Computational Attention, 2023. 2, 3

[73] Alfred L. Yarbus. *Eye Movements and Vision*. Plenum Press, 1967. 2

[74] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and Why Vision-Language Models Behave like Bags-Of-Words, and What to Do About It? In *The Eleventh International Conference on Learning Representations*, 2022. 3

[75] Lingyun Zhang, Matthew H. Tong, Tim K. Marks, Honghao Shan, and Garrison W. Cottrell. SUN: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7):32–32, 2008. 2