# Supplementary Material for
# Neuralizer: Neuroimage Analysis without Re-Training

## A. Samples

We provide examples of model inputs – target image and context set – and Neuralizer-seen predicted outputs. The inputs are sampled at random from the test dataset. The context set length is sampled from the discrete random uniform distribution $\mathcal{U}_{\{1,32\}}$. To reduce visual clutter, we display up to eight context image pairs and omit the rest in the visualization. We also only show one channel, excluding additional inputs like multiple modalities, or the binary mask for in-painting tasks. We provide a collection of images from the first 50 samples from the test dataset. We only excluded examples to avoid duplication of tasks.

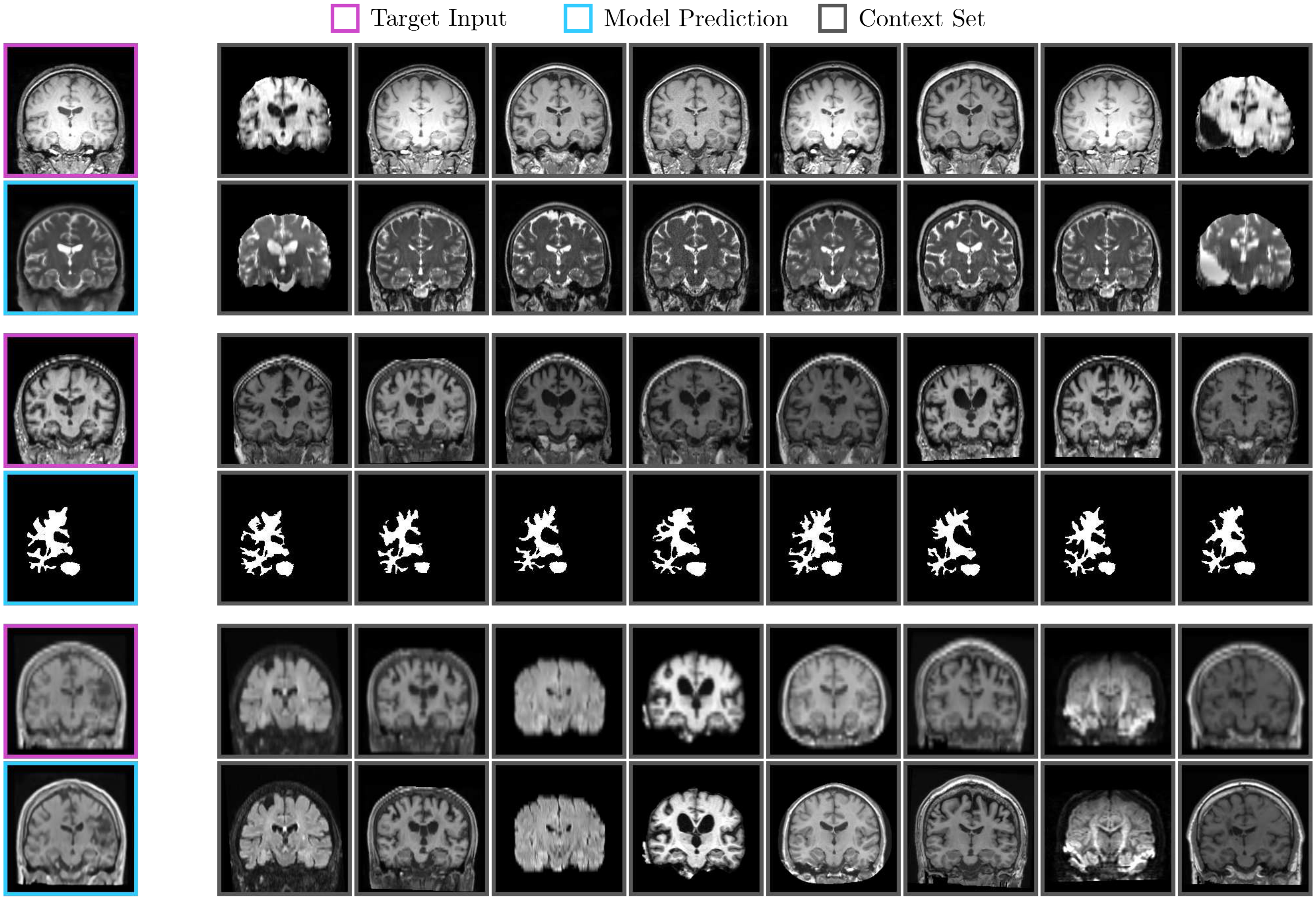

Figure 6. Sample Neuralizer-seen predictions. Left: Target input (magenta frame) and model prediction (blue frame). Right: context set supplied to inform the task (grey frame). We provide more samples on the next pages.

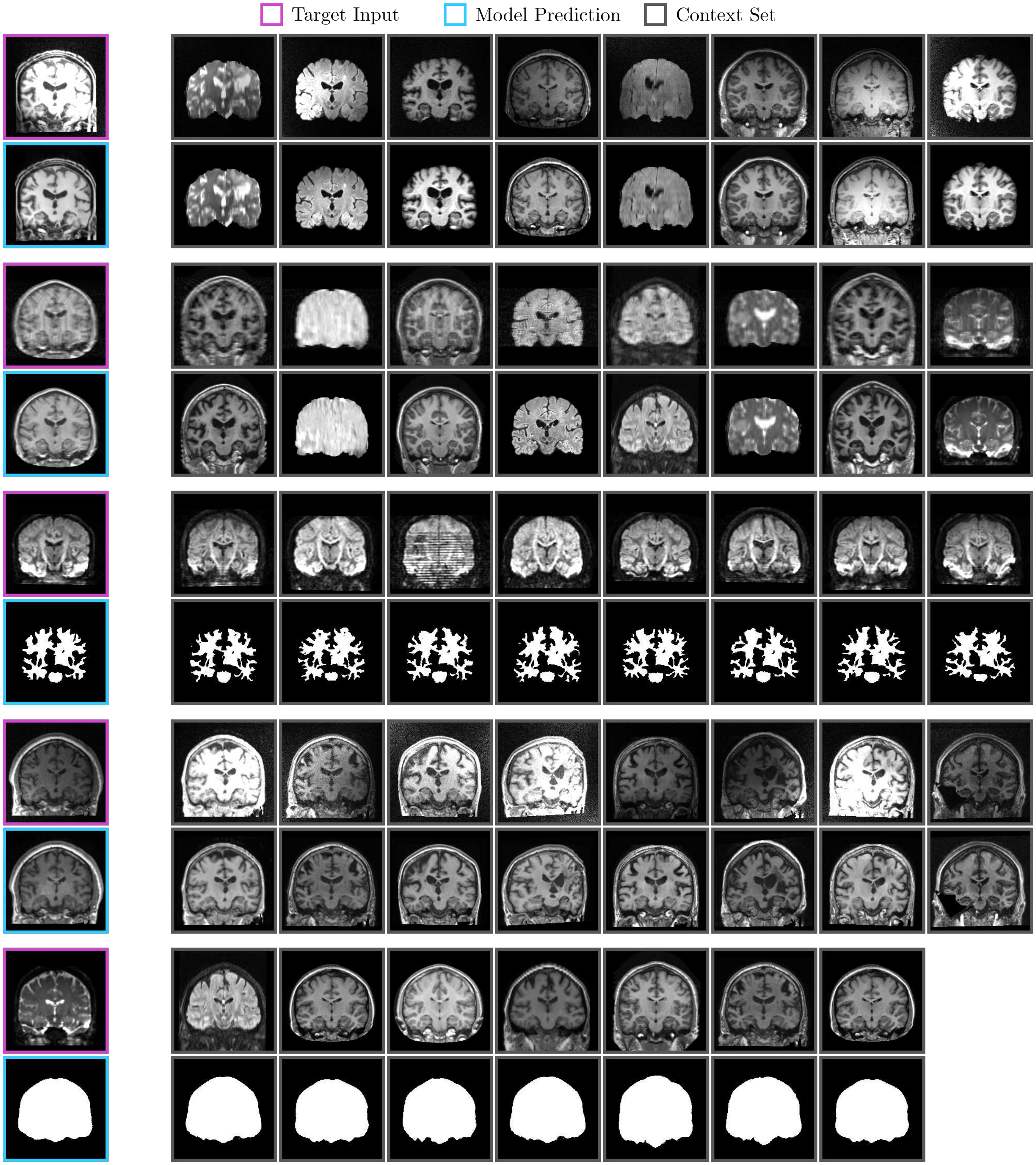

Figure 7. Sample Neuralizer-seen predictions (continued). Left: Target input (magenta frame) and model prediction (blue frame). Right: context set supplied to inform the task (grey frame).

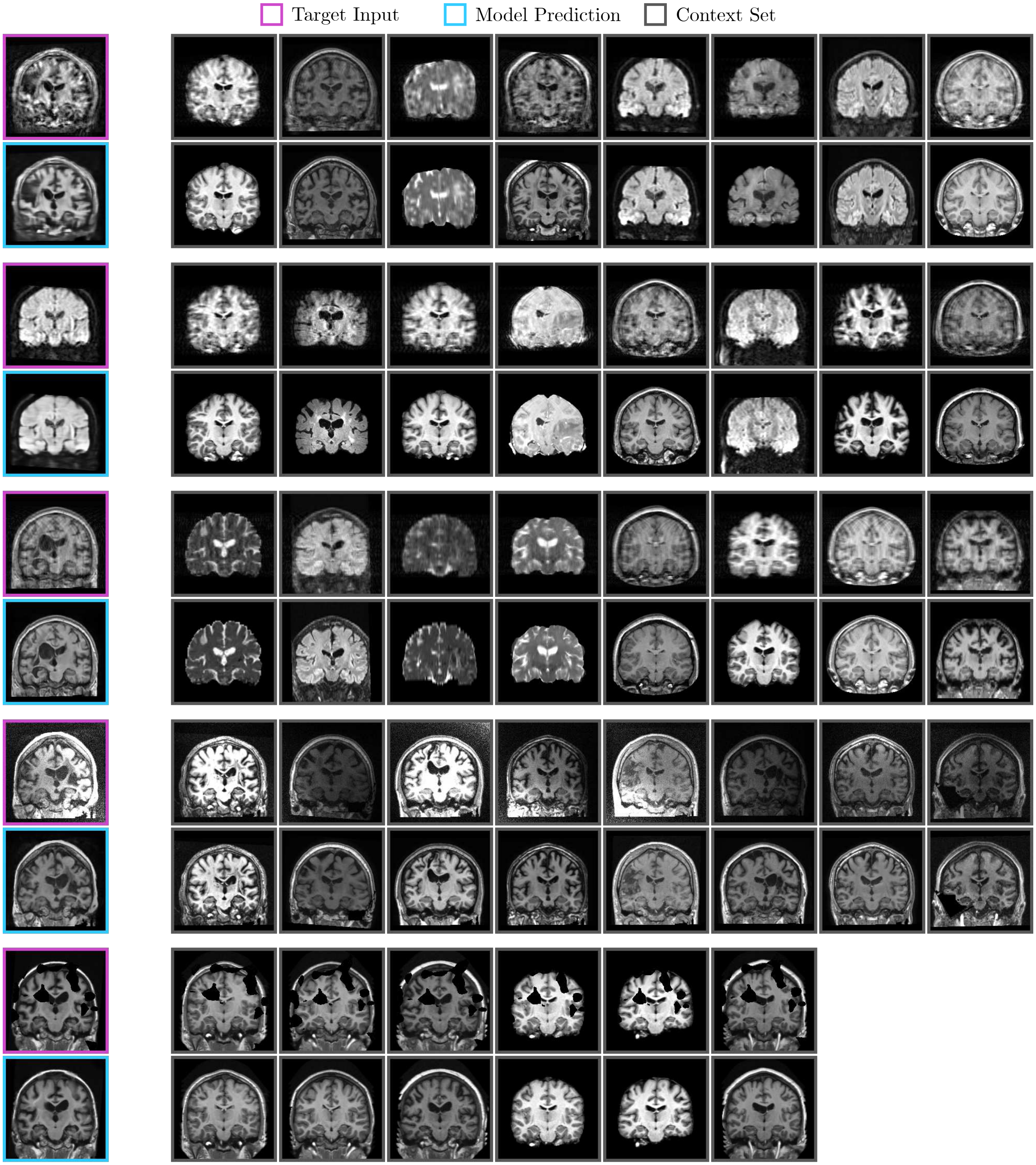

Figure 8. Sample Neuralizer-seen predictions (continued). Left: Target input (magenta frame) and model prediction (blue frame). Right: context set supplied to inform the task (grey frame).

## B. Train samples

We provide samples from the train set, including data and task augmentations, and show all three input channels. Further examples of the visual diversity possible with task augmentations are shown in Fig. 11.

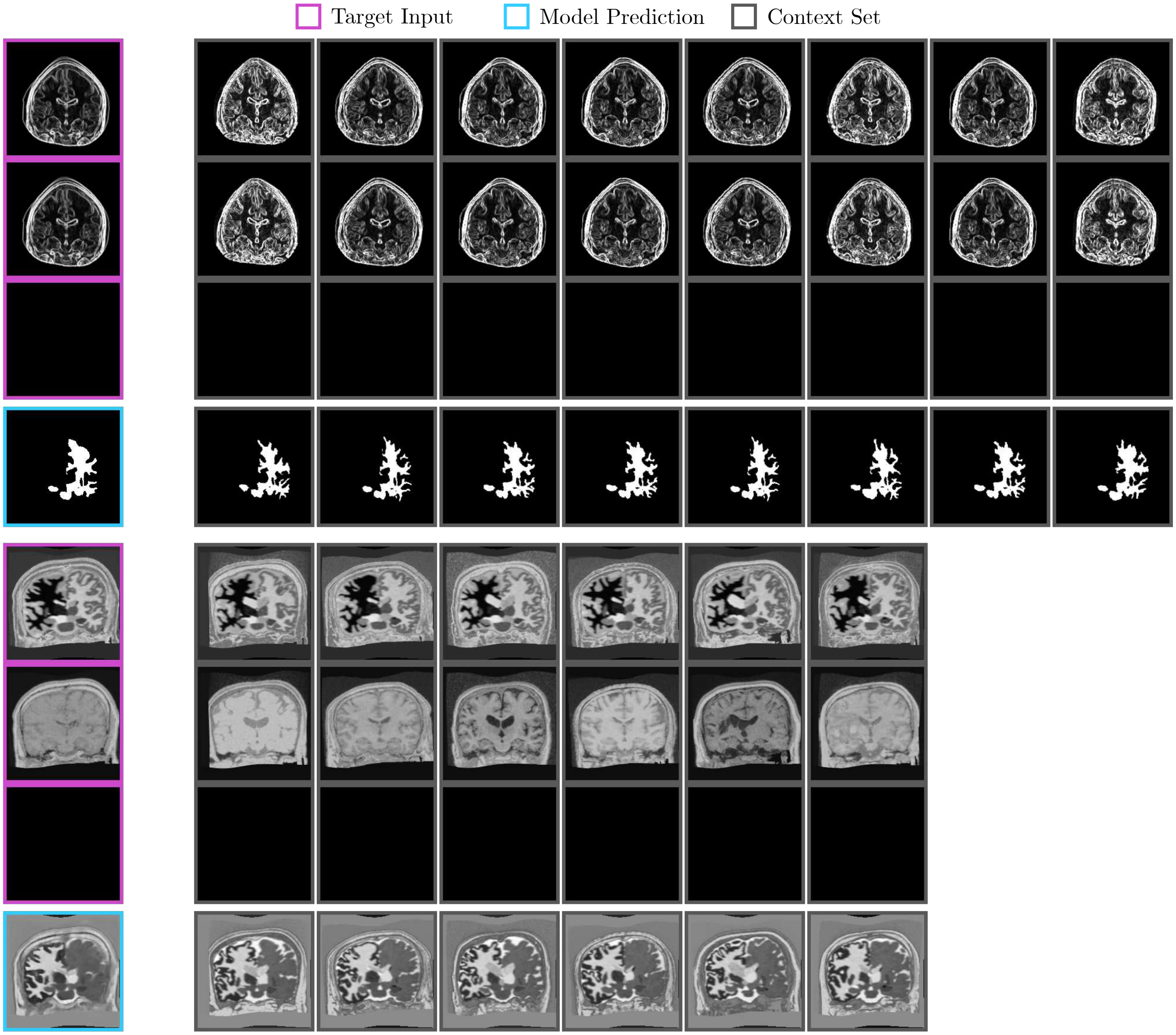

Figure 9. Sample Neuralizer-seen predictions from the train set, with data and task augmentations. All three channels of the input are shown. Left: Target input (magenta frame) and model prediction (blue frame). Right: context set supplied to inform the task (grey frame).

## C. Task augmentations

Task augmentations are randomized data augmentations applied to both input and target images or segmentation maps. These change not only the appearance of the input image, but also the target and members of the context set, essentially altering the task itself. We apply task augmentations not to create plausible neuroimaging tasks, but instead to expand the set of tasks the model is exposed to during training. This prevents memorization of the training tasks, and aids generalization to unseen tasks during inference. We first describe the task augmentations in C.1, then discuss their composition in C.2, and finally provide examples in Fig. 11. Hyper-parameters for all augmentations are selected by visual inspection.

### C.1. Task augmentations

We provide a description of each task augmentation. In addition to the task augmentations, we use data augmentations via random affine movements, random elastic deformations, and random flips along the sagittal plane.

**SobelFilter.** A Sobel filter is applied to an intensity image.

**IntensityMapping.** The intensity of an image is remapped [45] To perform this operation, the image intensity values are split into histogram bins, and each bin is assigned a new intensity difference value. To obtain new intensity values, we compute a distance from the original intensity value to the two neighboring bin centers, using linear interpolation.

**SyntheticModality.** An intensity image is replaced with a synthetic one generated from an anatomical segmentation map of the subject, using previous work [45]. Each anatomical segmentation class is randomly assigned an intensity mean and standard deviation and the new synthetic modality image of the brain is generated according to these distributions. As our anatomical segmentations do not cover the skull, we take an extra step to ensure skulls are present in the synthetic data: If the original intensity image had a skull, the generated brain is overlaid onto the original image, thus keeping the skull.

**MaskContour.** We extract a contour of the binary mask in a segmentation task, which then represents the new target segmentation mask. Contoured Masks are always dilated to a width of 3 voxels.

**MaskDilation.** The binary segmentation mask is dilated by 1 voxel.

**MaskInvert.** The binary segmentation mask is inverted.

**PermuteChannels.** The input images are represented by three channels. On each input during training, we permute the input channels. This encourages the network to ignore the specific channel order.

**DuplicateChannels.** We overwrite empty input channels with the duplication of a non-zero channel. The augmentation is applied to each empty channel with a probability $p$.

### C.2. Composition and likelihood of task augmentations

We compose task and data augmentations during training. Some task augmentations can be combined (e.g. MaskDilation and MaskInvert), while others are exclusive to each other (e.g. SobelFilter and SyntheticModality). To model these dependencies, we define the default composition tree used for most tasks in Fig. 10. The augmentation groups “Mask Augmentations”, “Intensity Augmentations”, “Channel Augmentations”, and “Spatial Augmentations” are applied in this order. Augmentations in child nodes of “Compose” are applied left to right, while “OneOf” selects a single child augmentation to apply. A node is applied with probability $p$ stated on the node.

Some tasks use modified versions of this composition tree. As a safety feature, we do not use RandomFlip for segmentation-related tasks, as this can lead to information leakage when evaluating on non-symmetric class-holdouts (in our experiments presented here we always hold out the same anatomical class on both sides of the brain, but this has not always been the case during development). To simplify other tasks, we omit MaskContour and MaskDilate from the inpainting task, and SobelFilter and SyntheticModality form the modality transfer task.

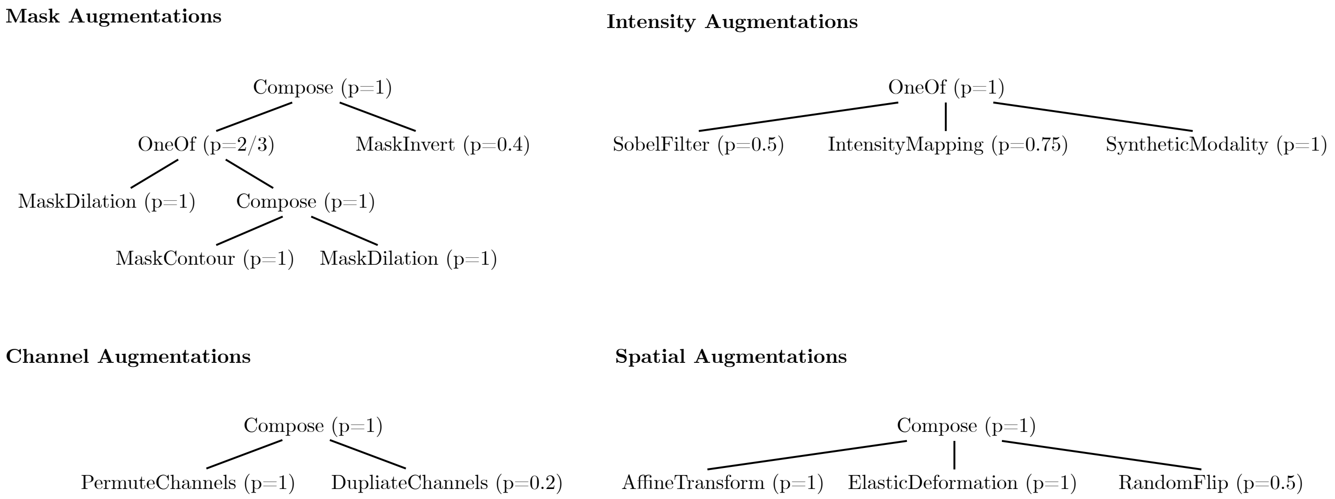

Figure 10. Default composition of augmentations used for most tasks during training. We use "Compose" and "OneOf" nodes to model these restrictions. Augmentations in child nodes of "Compose" are applied left to right, while "OneOf" selects a single child augmentation to apply. A node is applied with probability $p$.

## C.3. Examples of task augmentations

Fig. 11 provides visual examples of task augmentations applied to a segmentation and bias correction task.

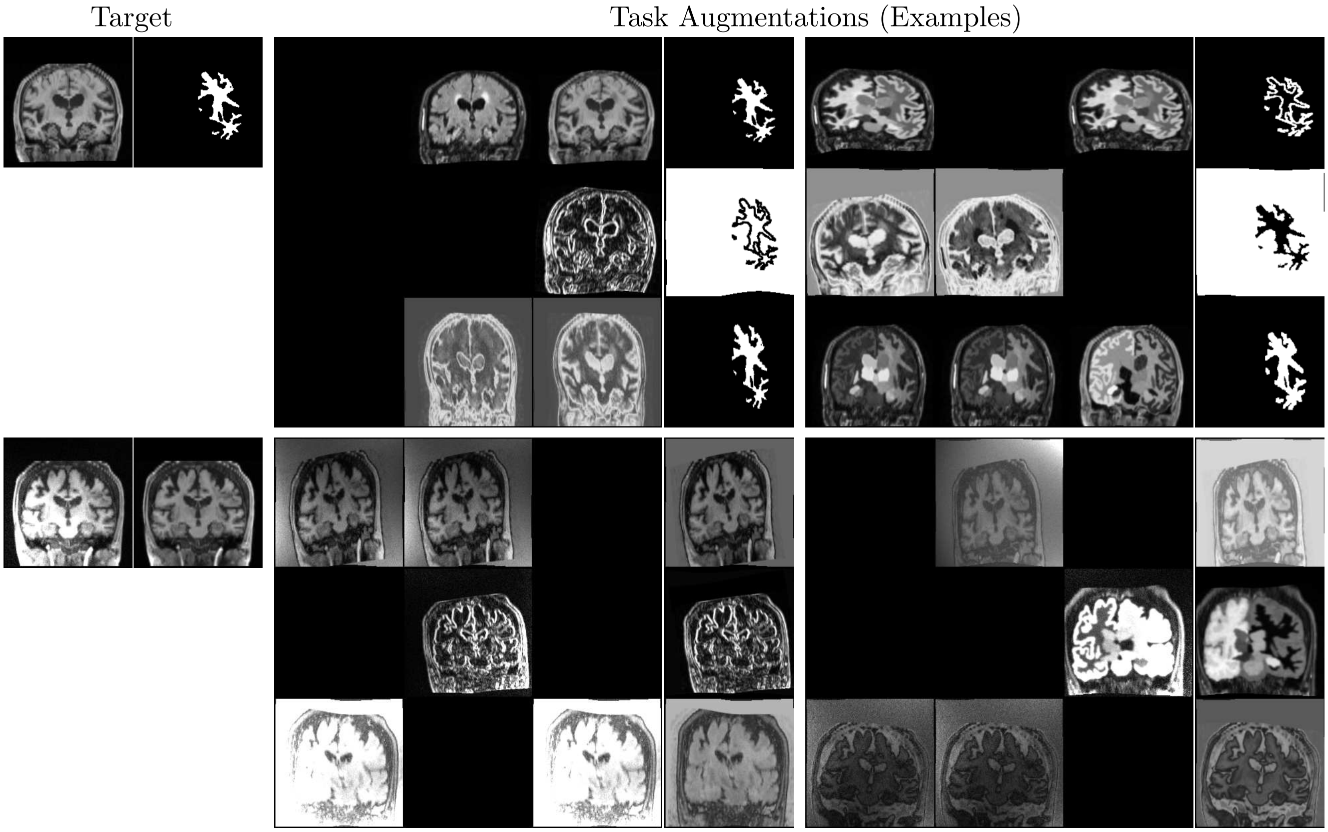

Figure 11. Examples of task augmentations, designed to increase the diversity of neuroimaging tasks seen by the model during training. We show non-augmented target input and output image of T1 modality on the left. We show examples of random data- and task-augmentations applied to the target during training on the right. The augmented target input is represented by up to three channels of real and synthetic modalities of the subject. The target output is augmented with synthetic image modalities and alterations to the segmentation mask. The same augmentations are applied to the context set.

## D. Evaluation on T1 modality

We aggregated scores across all modalities in Fig. 4. To aid comparison to existing literature, which most often focuses on T1 images, we provide the same evaluation, performed on just the T1 modality here. Some tasks are easier on T1 data, thus improving scores. For small dataset sizes of 1 or 2 subjects, the baselines sometimes underperform on the T1 modality. This is often because images of the T1 modality may not always present in small training sets. For sizes of 4 subjects and larger, the T1 modality is always included in the training set.

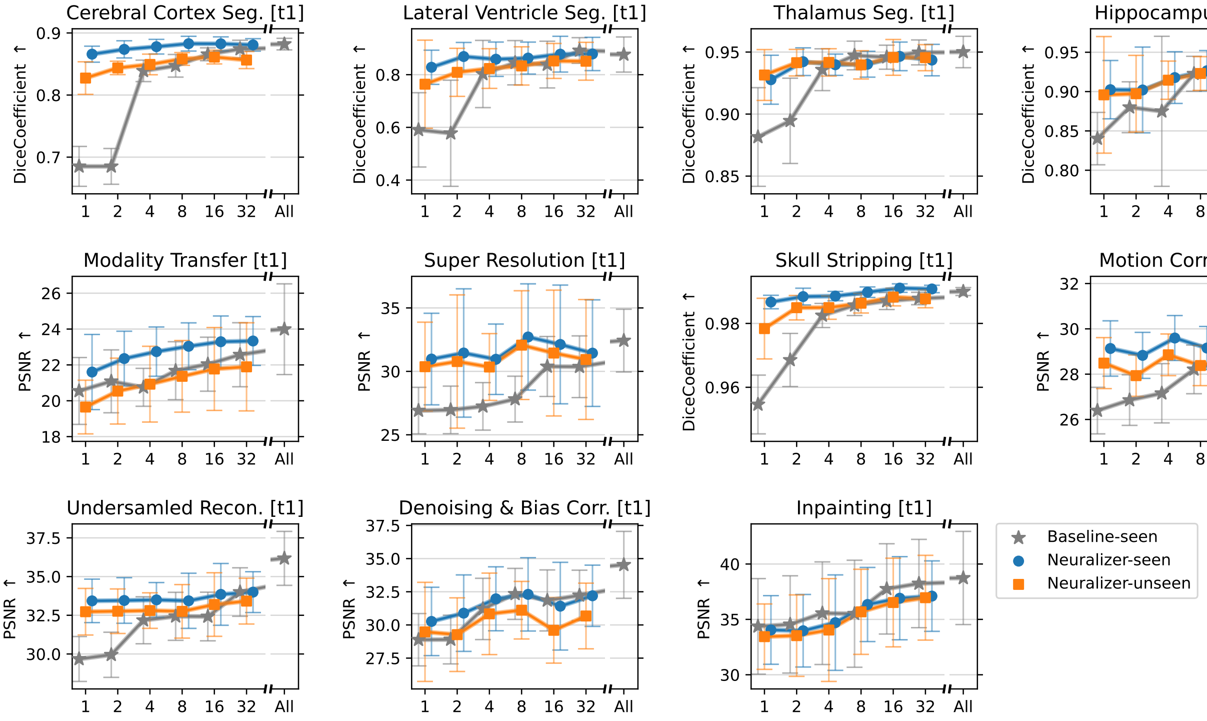

Figure 12. Performance of multi-task Neuralizer and the task-specific baselines on each task, T1 modality only. The tasks being evaluated were included in the training of Neuralizer-seen (orange), held out in Neuralizer-unseen (blue), and specifically trained on by each task-specific baseline (gray). The x-axis is the size of the train/context set, and the y-axis is the Dice/PSNR score. Some points on the x-axis are omitted for better visibility. ‘All’ refers to all available train data for the task, ranging from 249 to 2,282 subjects depending on the task. The bars denote the standard deviation across subjects.

## E. Experiments 1 and 2 tabular results

| Model | Trained | Subjects | Segmentation | Mod. Transfer | Super Res. | Skull Strip. | Motion Recon. | Undersamp. Recon. | Noise Recon. | Inpainting |
|---|---|---|---|---|---|---|---|---|---|---|
| Baseline-seen | ✓ | all | .83 ± .08 | 25.9 ± 3.0 | 33.6 ± 2.8 | .98 ± .01 | 31.8 ± 2.9 | 36.1 ± 2.7 | 33.5 ± 3.8 | 38.8 ± 2.4 |
| | | 32 | .80 ± .09 | 24.4 ± 2.5 | 31.6 ± 2.3 | .98 ± .01 | 29.3 ± 2.1 | 33.7 ± 2.2 | 30.4 ± 3.3 | 38.3 ± 2.4 |
| | | 16 | .78 ± .09 | 24.0 ± 2.3 | 31.3 ± 2.1 | .97 ± .01 | 28.8 ± 2.1 | 32.3 ± 2.1 | 30.2 ± 3.5 | 37.7 ± 2.1 |
| | | 8 | .77 ± .11 | 23.7 ± 2.2 | 29.0 ± 1.9 | .97 ± .02 | 28.1 ± 2.1 | 31.8 ± 2.0 | 30.0 ± 3.2 | 36.4 ± 2.2 |
| | | 4 | .75 ± .14 | 23.0 ± 2.3 | 29.2 ± 2.5 | .96 ± .03 | 27.9 ± 2.2 | 31.7 ± 2.0 | 28.5 ± 3.4 | 36.4 ± 2.3 |
| | | 2 | .65 ± .12 | 22.8 ± 2.1 | 28.7 ± 1.8 | .97 ± .01 | 27.2 ± 1.4 | 30.4 ± 1.2 | 27.8 ± 0.8 | 35.8 ± 2.0 |
| | | 1 | .59 ± .16 | 22.2 ± 2.1 | 29.0 ± 2.4 | .95 ± .02 | 27.0 ± 1.8 | 30.3 ± 1.9 | 27.6 ± 1.0 | 35.8 ± 1.8 |
| Neuralizer-seen | ✓ | 32 | .84 ± .07 | 25.3 ± 2.2 | 32.3 ± 2.8 | .99 ± .00 | 30.2 ± 2.5 | 34.3 ± 2.7 | 32.1 ± 3.1 | 36.1 ± 3.2 |
| | | 16 | .83 ± .07 | 25.1 ± 2.1 | 32.9 ± 3.1 | .99 ± .00 | 30.2 ± 2.6 | 34.2 ± 2.7 | 31.7 ± 3.0 | 35.8 ± 2.9 |
| | | 8 | .82 ± .09 | 24.8 ± 2.2 | 32.7 ± 3.3 | .98 ± .00 | 30.1 ± 2.6 | 34.3 ± 2.6 | 32.1 ± 3.2 | 35.7 ± 3.1 |
| | | 4 | .80 ± .09 | 24.2 ± 2.0 | 32.3 ± 3.2 | .98 ± .00 | 30.1 ± 2.6 | 34.3 ± 2.7 | 31.9 ± 3.2 | 35.1 ± 2.6 |
| | | 2 | .78 ± .10 | 23.9 ± 2.0 | 32.3 ± 2.5 | .98 ± .01 | 29.9 ± 2.5 | 34.2 ± 2.6 | 30.9 ± 2.9 | 35.0 ± 2.7 |
| | | 1 | .74 ± .13 | 23.0 ± 2.0 | 32.1 ± 2.9 | .98 ± .01 | 29.9 ± 2.6 | 34.1 ± 2.5 | 30.9 ± 3.2 | 34.5 ± 2.7 |
| Neuralizer-unseen | ✗ | 32 | .84 ± .07 | 24.4 ± 2.1 | 32.1 ± 2.7 | .98 ± .00 | 30.0 ± 2.6 | 34.2 ± 2.6 | 30.8 ± 3.9 | 36.4 ± 3.3 |
| | | 16 | .83 ± .07 | 24.2 ± 2.1 | 32.7 ± 3.1 | .98 ± .00 | 29.9 ± 2.6 | 34.1 ± 2.7 | 30.3 ± 3.6 | 36.0 ± 2.7 |
| | | 8 | .82 ± .08 | 23.8 ± 2.0 | 32.6 ± 3.2 | .98 ± .00 | 29.9 ± 2.6 | 34.2 ± 2.6 | 30.7 ± 3.7 | 35.8 ± 2.8 |
| | | 4 | .81 ± .08 | 23.3 ± 1.9 | 32.2 ± 3.2 | .98 ± .01 | 29.9 ± 2.6 | 34.1 ± 2.7 | 30.7 ± 3.9 | 35.2 ± 2.5 |
| | | 2 | .78 ± .09 | 22.9 ± 2.0 | 32.1 ± 2.4 | .98 ± .01 | 29.6 ± 2.5 | 34.0 ± 2.6 | 29.7 ± 3.4 | 35.2 ± 2.6 |
| | | 1 | .74 ± .11 | 22.1 ± 2.0 | 31.9 ± 2.9 | .97 ± .01 | 29.7 ± 2.5 | 33.9 ± 2.5 | 30.0 ± 3.9 | 34.5 ± 2.7 |

Table 4. Model scores (Dice for segmentation and skull-stripping, PSNR for other tasks) for each model and task as a function of the available subjects for training (U-Net) or context set (Neuralizer). Higher values are better. We average scores across all test subjects, eight modalities, and four segmentation classes (Cerebal cortex, Lateral ventricle, Thalamus, Hippocampus). Standard deviation across modalities and segmentation classes.

## F. Class names for Hammers Atlas dataset (experiment 3)

We provide label names and indices for the tissue classes in Tab. 3, re-compiled from [27, 37, 40].

| Abbreviation | Class Index | Class Name |
|---|---:|---|
| Hip | 2 | Hippocampus |
| PAG | 10 | Parahippocampal and ambient gyri |
| STG | 12 | Superior temporal gyrus |
| MIG | 14 | Middle and inferior temporal gyri |
| FuG | 16 | Lateral occipitotemporal gyrus (fusiform gyrus) |
| Stm | 19 | Brainstem |
| Ins | 20 | Insula |
| PCG | 26 | Gyrus cinguli, posterior part |
| Tha | 40 | Thalamus |
| CC | 44 | Corpus callosum |
| 3V | 49 | Third ventricle |
| PrG | 50 | Precentral gyrus |
| PoG | 60 | Postcentral gyrus |
| ALG | 94 | Anterior long gyrus |

Table 5. Hammers Atlas label abbreviations.

## G. Training dataset creation

We dynamically generate input image $x_t$, ground truth output $y_t$, and context set $\{(x_{t,j}, y_{t,j})\}_{j=1}^{N}$ from a collection of underlying datasets (Tab. 1) during training.

In every training iteration, we first sample a task $t$ from $T_{\text{seen}}$. Next, one of the underlying datasets is selected to generate the sample $(x, y)$. Due to the makeup of the datasets, not every task can be performed on every dataset. For example, a dataset involving a single modality can not naturally be used to generate a modality transfer task. From the list of valid datasets, we sample the datasets for the input and context images independently, with a $1/3$rd chance of all context images coming from the same dataset as the input, $1/3$rd chance that context datasets are sampled at random from the valid datasets, and $1/3$rd chance that the context does not contain any subjects of the input dataset.

After the selection of task and dataset, we create the input and output images. This creation varies by task. We draw the subjects from each dataset at random, but exclude the input subject to re-occur as a context set member. For most tasks, we sample a subset of between one to three image modalities from the subject. For the segmentation task, we join a random subset of available segmentation classes into a binary target mask. For reconstruction and denoising tasks, noise and artifacts in the input images are simulated according to [92]. For the modality transfer task, we select a separate target modality. For the inpainting task, we create a random binary mask from Perlin noise mask these areas from the input image. For skull stripping, the target is a binary brain mask. For tasks other than segmentation and modality transfer, the modality of context images can vary from the input image.

## H. Inference cost and model size

We provide model parameter counts and inference costs. We use a Baseline U-net with 64 channels for experiments with limited data set sizes, and a U-Net with 256 channels for experiments on all data. For Neuralizer, we use the same model in all experiments, but the inference cost increases linearly with the size of the context set.

Table 6. Model size and inference cost.

| Model | inference FLOP (g) | Parameters (m) |
|---|---:|---:|
| Baseline, 64 channels | 20.7 | 0.62 |
| Baseline, 256 channels | 329.7 | 9.84 |
| Neuralizer, 1 ctx image | 39.1 | 1.27 |
| Neuralizer, 32 ctx images | 610.5 | 1.27 |

## I. Task weights

To speed up training, we use weighted sampling of tasks during training. Task weights are shown in Tab. 7. These values have been tuned experimentally. Tasks that converge fast and achieve high-quality results are given a lower weight. Tasks that take longer to converge or are given a higher weight.

Table 7. Task weights during training.

| Task | Weight |
|---|---|
| Binary Segmentation | 2.0 |
| Modality Transfer | 2.0 |
| Superresolution | 1.0 |
| Skull Stripping | .5 |
| Motioncorrection Reconstruction | .5 |
| Denoising & Bias correction | .5 |
| k-space Undersampling Recon. | 1.0 |
| Inpainting | 1.0 |
| Simulated Modality Transfer | 1.0 |
| Masking | .5 |