HAZEMATCHING: Dehazing Light Microscopy Images with Guided Conditional Flow Matching

Supplementary Material

A. Overview of the HAZEMATCHING

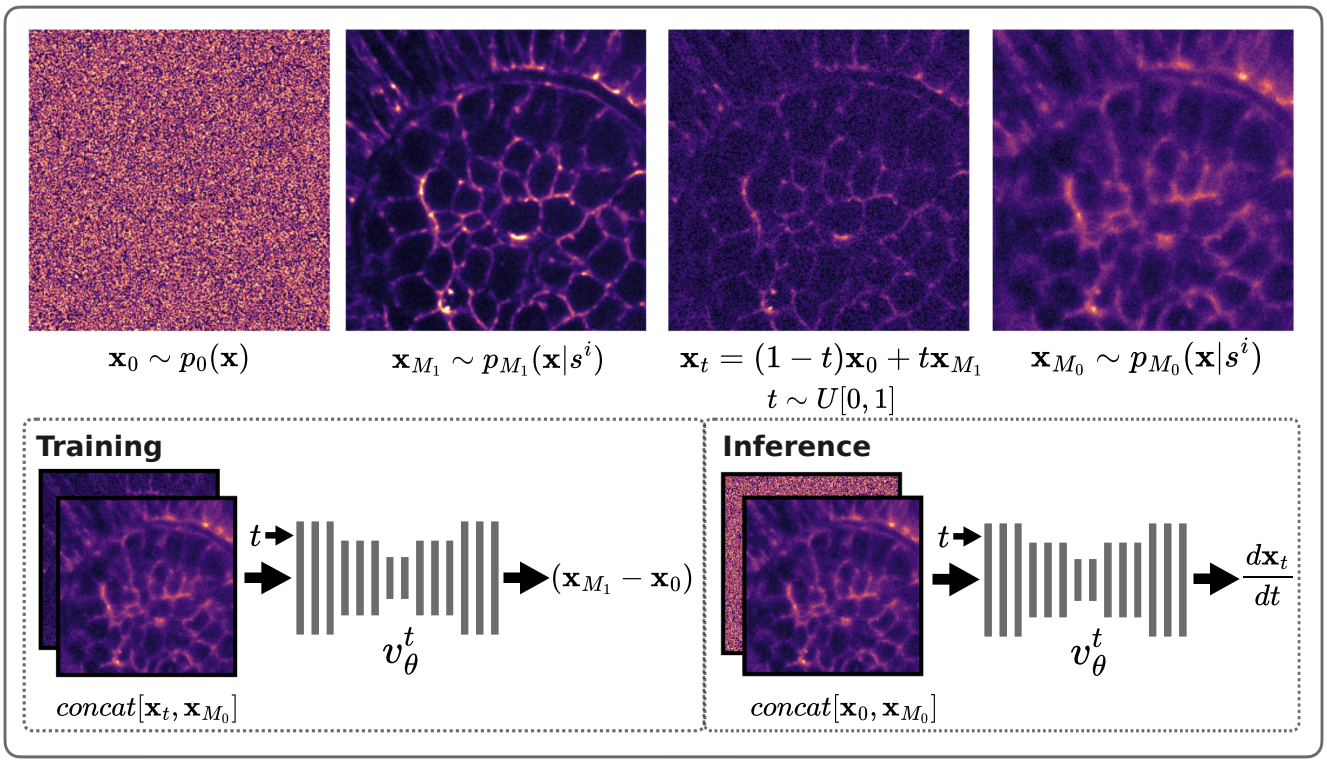


Figure S1. **Overview of our training and inference.** The figure illustrates the different distributions involved in our approach. The top panel shows how samples are drawn from the base noisy distribution $p_0(\mathbf{x})$, the target non-hazy image distribution $p_{M_1}(\mathbf{x}|s^i)$, and the hazy source image distribution $p_{M_0}(\mathbf{x}|s^i)$, and the computing of $\mathbf{x}_t$ with sampling time as $t \sim U[0,1]$. The lower left panel presents the training scheme, where the model is trained to learn a mapping between these distributions using conditional flow matching with inputs $concat[\mathbf{x}_t, \mathbf{x}_{M_0}]$ and t. The lower right panel depicts the inference process, where given a degraded observation $\mathbf{x}_{M_0}$, the trained model takes as inputs $concat[\mathbf{x}_0, \mathbf{x}_{M_0}]$ ($\mathbf{x}_0 \sim p_0(\mathbf{x})$) and t and predicts the time dependent velocity field per pixel ($d\mathbf{x}_t/dt$) which is integrated using an (Euler) ODE solver iteratively using our proposed dehazing function D to produce one prediction.

B. Uncertainty and Calibration

In this section, we describe our uncertainty estimation and calibration procedure [12]. We use pixel-wise variability across multiple predictions of HAZEMATCHING and assess if this value scales with the true prediction error, based on a body of ground-truth data. Under the hood, we adopt a binning-based calibration procedure inspired by [6] where we generate multiple predictions for each test image and analyze the relationship between predicted standard deviation and the observed true prediction error.

For each test image $\mathbf{x} \in \mathbb{R}^{H \times W}$, we generate $k = 50$ samples such that for each pixel p in the test image $\mathbf{x}$, we have k predictions. We then use these k predictions to compute the pixel-wise standard deviation $\sigma(p)$ for the pixel location p. Next, we sort $\sigma(p)$ and bin them over $l = 50$ equally sized bins $B : \{B_1, B_2 ... B_l\}$. We then compute the *Root Mean Variance*

(RMV) and *Root Mean Squared Error* (RMSE) for each B_j as

$$\text{RMV}(j) = \sqrt{\frac{1}{|B_j|} \sum_{p \in B_j} \sigma(p)^2}, \tag{s.1}$$

$$\text{RMSE}(j) = \sqrt{\frac{1}{|B_j|} \sum_{p \in B_j} \left(\hat{y}(p) - y(p)\right)^2}, \tag{s.2}$$

where $\hat{y}$ is the predicted mean (MMSE estimate) and y is the ground truth. Then using our validation set, we then fit a linear relationship between RMSE and RMV as

$$\min_{\alpha,\beta} ||\text{RMSE} - (\alpha \cdot \text{RMV} + \beta)||^2, \tag{s.3}$$

where α is a learnable scaling factor and β is the learned offset. To ensure a meaningful mapping, we constrain $\alpha > 0$. To assess the calibration quality on the test set, we generate multiple predictions for each input and compute the pixel-wise standard deviation σ across these samples. The corresponding estimate of the pixel-wise RMSE is then obtained using the learned linear mapping $\text{RMSE} = \alpha \cdot \sigma + \beta$. Note that this procedure does not alter the original predictions but instead learns a mapping that best predicts the measured error. All the calibrations were performed on the normalized data.

B.1. Sample efficiency for calibration

In Table S1, we report the calibration slope α and the PSNR of the MMSE prediction as a function of the number of posterior samples k across the five datasets. Across all datasets, α increases steadily with k, reflecting improved variance calibration as the posterior sample grows. PSNR also rises consistently but saturates early. By $k \approx 20$ the gains become marginal, and beyond $k = 40$ all datasets exhibit changes below 0.03dB. In this high–k regime, α varies only slightly between $k = 40$ and $k = 50$ (Zebrafish: $+2.6\%$, Organoids1: -0.4%, Organoids2: -2.5%, Microtubule: $+4.5\%$), indicating small fluctuations rather than systematic trends, with only exception being the Neuron data with a change of $+25.9\%$. We therefore use $k = 50$ throughout the manuscript.

k	**Zebrafish**		**Organoids1**		**Organoids2**		**Microtubule**		**Neuron**	
	α	PSNR↑	α	PSNR↑	α	PSNR↑	α	PSNR↑	α	PSNR↑
5	1.0931	27.57	1.2554	36.58	0.6676	34.43	0.3924	27.50	0.5677	28.39
10	1.3012	27.68	1.8286	36.71	0.9729	34.74	0.6153	27.71	0.6295	28.69
20	1.5117	27.74	2.0911	36.78	1.1892	34.92	0.9049	27.80	0.8641	28.86
30	1.5575	27.76	2.0880	36.81	1.2655	34.97	1.0013	27.84	0.7876	28.91
40	1.6602	27.78	2.4993	36.82	1.3992	35.00	1.1114	27.85	0.7804	28.94
50	1.7033	27.78	2.4903	36.83	1.3646	35.02	1.1671	27.87	0.9825	28.96

Table S1. Calibration slope α and PSNR of the MMSE prediction for varying posterior sample count k across the five datasets.

B.2. Calibration factor validity

To verify that the learned calibration parameters (α and β) are meaningful, we apply the `scale` and `offset` computed on the validation set back onto the same validation set (rather than onto the test set used in the main results). This produces curves that align more closely with the identity line, confirming that the calibration behaves as expected as shown in Figure S2.

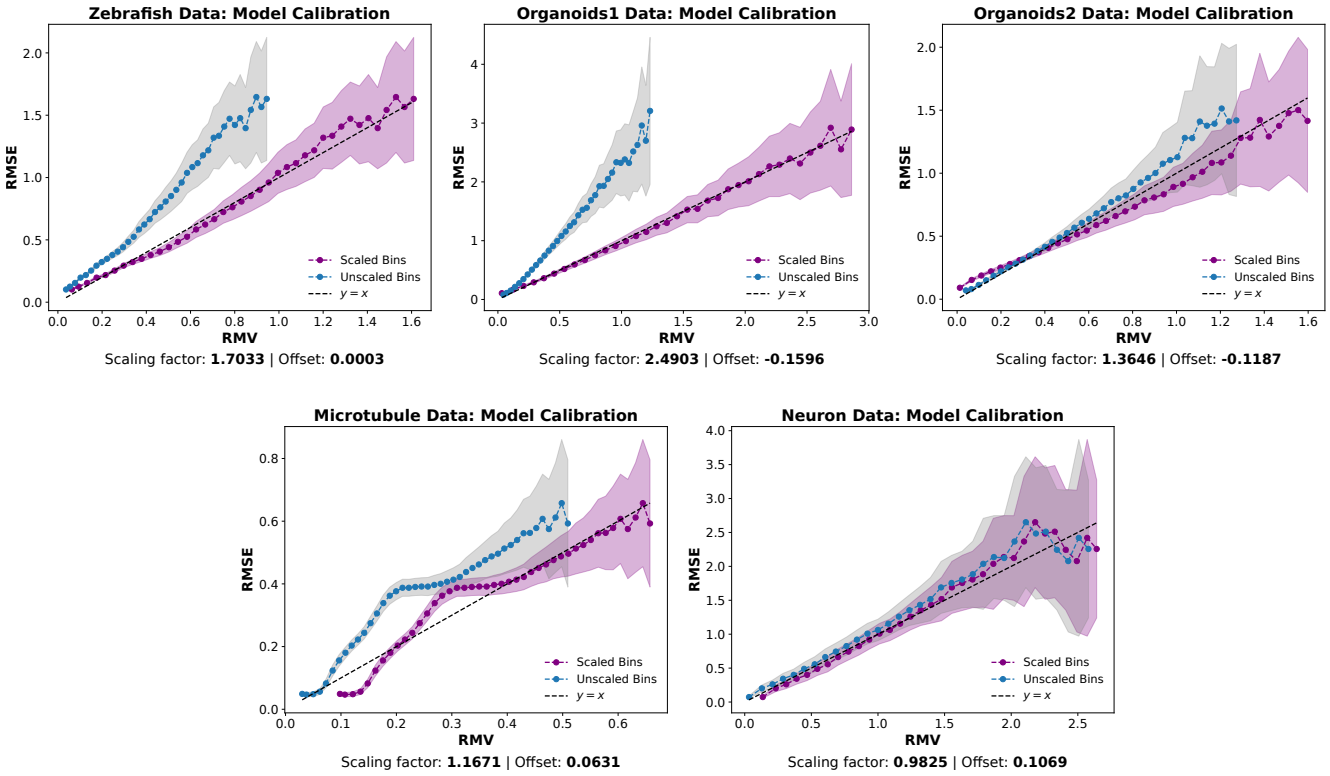


Figure S2. Calibration factors computed on validation data, applied on the validation data.

C. Datasets

C.1. Microsim: Widefield-Confocal simulator

Conceptually, a confocal microscope can be viewed as a widefield microscope augmented with a pinhole aperture placed in a plane conjugate to the focal plane of the specimen. In widefield microscopy, the entire specimen is illuminated, and emitted fluorescence from all depths is collected, leading to images with significant background blur due to out-of-focus light. By introducing a pinhole at the conjugate image plane, the confocal setup selectively allows only in-focus light to reach the detector, effectively rejecting out-of-focus fluorescence and enhancing image clarity. By manipulating the pinhole aperture, one can simulate different imaging conditions:

- **Pinhole Restricted (Confocal Mode):** The pinhole restricts detection to in-focus light, producing images with reduced background haze.
- **Pinhole Open (Widefield Mode):** Without the pinhole, both in-focus and out-of-focus light are detected, resulting in images with increased background haze.

This toggling mechanism facilitates the generation of paired datasets comprising ”clean” (confocal) and ”hazy” (widefield) images. Such datasets are invaluable for training and evaluating image restoration algorithms, particularly in the context of microscopy image dehazing.

We use `microsim` [18] to generate paired data for the *Zebrafish*, *Microtubule*, and *Neuron* datasets. For *Zebrafish* and *Microtubule*, we start from clean confocal images and remove residual pixel-independent noise using `Noise2Void` [16]. We then simulate widefield-like counterparts using realistic physical pixel sizes and a confocal PSF with an open pinhole, incorporating microscopy-specific noise such as shot noise. This results in pairs with independent but statistically identical noise, preventing shortcut learning. For the *Neuron* dataset, we skip denoising since we begin from segmentation labels, which are noise-free. During training, we treat the confocal image as the clean target and the simulated widefield image as the hazy input.

Pixel size: In the `microsim` library, the `scale` parameter defines the physical size of each voxel along the respective spatial axes, typically specified in micrometers. For instance, a scale of `(0.02, 0.01, 0.01)` corresponds to voxel dimensions of 20 nm in the Z-axis and 10 nm in both Y and X axes. This parameter is crucial for ensuring spatial realism

in simulations, as it maps the discrete voxel grid to real-world physical dimensions. Accurate specification of the `scale` allows for realistic modeling of optical phenomena and ensures that simulated images correspond to the physical dimensions observed in actual microscopy data. To ensure spatial realism, we assign a physical pixel size (in nanometers) based on metadata from the original confocal datasets, further refined via expert visual inspection. This calibration enables accurate structure quantification and realistic widefield simulation. Detailed values for pixel sizes and pinhole diameters are provided in each of the dataset description below.

C.2. Zebrafish Data

We use the confocal *Zebrafish* dataset introduced in [40], which contains three distinct structures: nuclei, nuclear membrane, and nuclear envelope. After removing pixel-independent noise, haze is simulated as described in Section C.1, using a pinhole diameter of 30 Airy Units (AU) to emulate widefield degradation. For spatial realism, the `scale` parameter is set to `(0.4, 0.2, 0.2)` for Z, Y and X axes. As the three structures differ in intensity statistics, we normalize each structure independently using the mean and standard deviation computed from the training set. A single HAZEMATCHING model is trained across all structures. The original 3D confocal volume has shape $52 \times 1024 \times 1024$. After simulation, we extract the central plane, which exhibits the most haze. However for evaluating our method against the classical Richardson Lucy (RL) method, we used this entire stack because RL requires the input to be a 3D stack. We use a crop size of 128×128 during training and validation. The dataset contains 15 training images and 3 images each for validation and testing, all of size 1024×1024. We generated a total of 3000 patches for training. Pixel size is 0.2 μm.

C.3. Organoids1

This is a real microscopy dataset acquired using a spinning-disk confocal system, which is particularly relevant to our work due to its ability to toggle between confocal and wide-field modes simply by removing the spinning disk from the optical path. This enables the acquisition of non-hazy (confocal) and hazy (widefield) image pairs under nearly identical conditions, making it ideal for evaluating dehazing algorithms. The dataset was obtained using a spinning disk confocal system, consisting of a CrestOptics V3 Light scan-head (configured with 50μm pinhole) mounted on a `Nikon Ti2-E` inverted microscope equipped with a motorized stage and a Photometrics Prime 95B 25mm camera (pixel size 11μm). The samples were acquired with a PLAN APO Lambda S 40x/1.25NA silicon immersion objective in combination with a 1.5x optovar lens (final magnification 60x) using Celesta Lumencor solid-state lasers as light source. For each acquisition, a specific filter set was used: a penta-band excitation filter (MXR00543-CELESTA-DAPI/FITC/TRITC/Cy5/Cy7-Full Multiband Penta), a penta-band dichroic filter (MXR00543-CELESTA-DAPI/FITC/TRITC/Cy5/Cy7-Full Multiband Penta) and a FITC band-pass emission filter (Semrock FF01-511/20-25). For each Field of View (FOV), two single-plane images were acquired: a confocal and a widefield image. The widefield images were acquired without the Nipkow disk using the 477nm laser line at 1% and 80ms of exposure time, while the confocal images were captured with the Nipkow disk along the light path using the 477nm laser line at 45% and 500ms of exposure time. All images were acquired using a 1024×1024 pixel format and a pixel size of 190nm. We normalize each image using the mean and standard deviation computed from the training set. We use a crop size of 128×128 during training and validation. The dataset contains 15 training images and 2 images for validation and 3 images for testing, all of size 1024×1024. We generated a total of 3000 patches for training. Pixel size is 0.190 μm.

C.4. Organoids2

This human brain organoid dataset was acquired on a Leica Stellaris 8 point-scanning confocal system mounted on a DMI8 inverted microscope with a motorized stage, using a 405nm diode laser line in combination with a white light laser (WLL; 440nm -790nm) as excitation sources. The images were acquired with a HC PL APO 63x/1.40NA oil immersion objective using the 488nm laser line (WLL) at 5% and 500nm – 600nm as emission collecting range. The pixel format of the 16-bit images was 1024×1024 and the pixel size was 90nm, while the pixel dwell time was 887ns. For each FOV, two single-plane images were acquired: one using the fully open pinhole corresponding to 6.28 Airy Units (AU) and 600μm as widefield-like image and the other one using 1 AU pinhole size corresponding to 95.52μm as confocal image. By acquiring matched image pairs with the pinhole closed and fully open, this setup allows for controlled confocal and widefield-like imaging from the same sample, making it well-suited for dehazing evaluation. We normalize each image using the mean and standard deviation computed from the training set. We use a crop size of 128×128 during training and validation. The dataset contains 15 training images and 2 images for validation and 3 images for testing, all of size 1024×1024. We generated a total of 3000 patches for training. Pixel size is 0.090 μm.

C.5. Microtubule data

We use confocal images of *Microtubules* tagged with α-Tubulin from the Allen Cell dataset [3]. Haze is simulated as described in Section C.1, using a pinhole diameter of 45 Airy Units (AU) to emulate widefield degradation. For spatial realism, the `scale` parameter is set to `(0.55, 0.169, 0.169)` for Z, Y and X axes. We normalize each image using the mean and standard deviation computed from the training set. A crop size of 128×128 is used for training and validation. Similar to the Zebrafish data, these are real confocal images, they already contain pixel-independent noise (e.g., Gaussian and Poisson). To prevent the simulator from applying haze on top of existing noise—thus resulting in similar noise statistics for both clean and degraded images—we first denoise them using `Noise2Void` [16], which removes only pixel-independent noise. The denoised image is then passed into the simulator to generate a hazy version with added realistic microscopy noise. Importantly, for training, we use the original confocal image as the clean target and the simulated hazy version as input. As the simulator requires a 3D volume, we use a stack of shape $52 \times 624 \times 624$ and select the middle slice, which contains the most haze, for training and evaluation. However, for evaluating our method against the classical Richardson Lucy (RL) method, we used this entire stack because RL requires the input to be a 3D stack. The dataset includes 52 training images of size 624×624 and 5 images each for validation and testing of size 512×512. We generated a total of 3016 patches for training. Pixel size is 0.1 μm.

C.6. Neuron data

This is a fully simulated dataset constructed from scratch using 3D segmentation labels from the Allen Cell Atlas [2], which contains 667 labeled volumes. To create realistic biological variability, we randomly sample 8–12 of these volumes chosen u.a.r, applying random rotations (sampled uniformly from $[-\pi, \pi]$) and isotropic scaling (sampled uniformly from $[0.8, 1.2]$). The transformed volumes are placed within a fixed spatial volume of size $256 \times 512 \times 512$ (Z, Y, X) and convolved with a point spread function (PSF) using `microsim` to simulate the imaging process. The `scale` parameter is set to `(0.04, 0.02, 0.02)`. For the confocal image, we use a pinhole diameter of 0.5 AU and for the widefield image we use a pinhole diameter of 5.0 AU. Detector noise is then added to produce realistic image degradation. For realistic simulation, we also isotropically downscale the simulated image by a factor of 8. Although the simulation generates full 3D stacks, we use only the middle slice for training and 2D evaluation, as it exhibits the strongest haze. However, for comparison with the classical Richardson Lucy (RL) deconvolution method, we use the entire 3D volume since RL operates on volumetric data. The final dataset consists of 5800 training images and 100 images each for validation and testing, all of size 64×64. Since this dataset is fully simulated, we can generate paired inputs and targets: the noisy, hazy image is used as input, and the clean (non-hazy, noise-free) image as target. This allows evaluation of both dehazing and denoising performance. Owing to the simplicity of the synthetic structures, this dataset exhibits less inherent data uncertainty compared to the other datasets used in this work. Pixel size is 0.16 μm.

C.7. Organoids sample preparation

We introduced two real datasets that were acquired in-house – *Organoids1* and *Organoids2*. Human brain organoids were fixed at 117 div, OCT embedded, frozen and prepared for sectioning. They were cut at 20μm and 50μm thickness using a cryostat. The cryosections were permeabilized with 0.5% Triton X-100 in PBS for 30 min. Blocking was performed in blocking solution (5% Normal Donkey Serum and 0.25% Triton X-100 in PBS) for 30 min. All sections were stained using phalloidin conjugated with AlexaFluor-488 (A12379, ThermoFischer Scientific), to label actin filaments. Phalloidin was incubated in blocking solution for 1 hour. After three washings, the sections were then mounted in Prolong Glass (P36980, ThermoFisher Scientific). All the steps of the immunofluorescence procedure were performed at room temperature.

D. Experimental setup

D.1. Training setup

For training, we use a patch size of 128×128 for *Microtubule*, *Zebrafish*, *Organoids1*, and *Organoids2* datasets, with 3016 patches for *Microtubule* and 3000 patches for each of the other three datasets. For the *Neuron* dataset, we use the original patch size of 64×64 for both training and evaluation with 5800 training patches. For evaluation, we adopt a rolling tile strategy: predictions are made on 128×128 tiles (64×64 for *Neuron*) and only the central 64×64 region (32×32 for *Neuron*) is retained. These are then stitched together to reconstruct the full image. All models are trained with T=20 integration steps. We implement HAZEMATCHING using a U-NET architecture as in [11], and compute the interpolants and solve the forward ODE using the `torchCFM` package [37]. Training is performed in PyTorch [29] on a half NVIDIA V100 GPU, with a batch size of 16, a learning rate of 10^{-4}, and the Adam optimizer.

D.2. Evaluation metrics and procedure

Distortion is evaluated using PSNR (a variant of the PSNR developed in ([40] for light microscopy images) and MicroMS-SSIM (a light microscopy-specific variant of MS-SSIM introduced in ([5]). For perceptual quality, we report LPIPS ([45]) and FID ([13]). LPIPS measures perceptual similarity via deep feature embeddings (from AlexNet), while FID computes the Fréchet distance between feature statistics of real and generated images.

Evaluation uses full-frame images for PSNR and MicroMS-SSIM. Perceptual metrics (LPIPS, FID, FSIM, GMSD) are computed on non-overlapping 64×64 patches (32×32 for *Neuron*) to eliminate stitching–related biases. Due to limited test samples, we estimate the clean image distribution using non-overlapping patches from the training sets and compute FID against predictions on the test set. This ensures robust evaluation without introducing data leakage, as training patches are used only for distribution estimation, not performance evaluation. A more detailed note on this can be found in supplementary Section D.2.1. Additionally, we present extended evaluations on the two additional synthetic datasets—*Neuron* (F.1) and *Microtubule* (F.2). In Section F.3, we highlight trade-offs between fine structure (via MicroMS-SSIM) and perceptual quality (LPIPS and FID). Next, in Section F.6 we provide full quantitative tables for PSNR, LPIPS, FID, MicroMS-SSIM, MS-SSIM [39], and two alternative perceptual quality scores FSIM [44], and GMSD [41]. Finally, in Section F.12, we show additional qualitative results to offer a more comprehensive view of all baseline and HAZEMATCHING. Note that all evaluations were conducted in the denormalized space, except for the *Neuron* dataset. For this dataset, evaluations were performed in the normalized space. In the case of Richardson–Lucy (RL) deconvolution, the *Neuron* data was normalized using min-max normalization to the $[0, 1]$ range instead of standard mean–standard deviation normalization. This unusual deviation from the default denormalization strategy was a practical decision, as the data—being fully simulated, exhibited very high intensity ranges due to the use of non-noisy ground truth segmentation volumes. For LPIPS, we normalize each image independently to the range $[-1, 1]$ using its own minimum and maximum intensity values, whereas for FID, we apply the same per-image normalization to the range $[0, 1]$.

D.2.1. On FID evaluation protocol

Due to the limited number of test patches in our datasets, directly computing the FID between predicted and real test samples can yield unstable results. To address this, we follow a practical approach: we aggregate non-overalpping 64×64 patches (or 32×32) from the training sets for each of the datasets to construct a representative empirical distribution of real clean images. Importantly, this does *not* mean we evaluate model performance on training data; rather, the training patches are used solely to estimate the underlying distribution of real clean images. The predictions on the test set are then compared against this aggregated distribution to compute FID. This strategy allows us to maintain a robust evaluation while ensuring that the test set remains completely unseen during training and inference.

D.2.2. Effect of Patch Size at Inference

While it is true that we train on 128×128 sized patches, the data itself is composed of much larger micrographs. The 128×128 constraint is imposed only by our data loader during training to enable efficient batching and GPU utilization. Since HAZEMATCHING is fully convolutional, it can, in principle, process inputs of arbitrary spatial size, subject only to GPU memory limitations. When full micrographs are too large to fit in memory, we use a tiling strategy during inference (see also Supplementary Section D).

To assess whether larger patch sizes improve performance, we evaluated our trained model on the **Organoids1** real spinning disk dataset using patch sizes of 128, 256, and 512 pixels at inference. The results, summarized in Table S2, show no meaningful improvement in PSNR when increasing the patch size beyond 128 pixels.

Model	**Patch Size**	**PSNR**$\uparrow$ (μ)	**PSNR** (σ)
HAZEMATCHING	128	36.83	5.894
HAZEMATCHING	256	36.85	5.964
HAZEMATCHING	512	36.76	6.217

Table S2. Effect of patch size at inference on **Organoids1** real spinning disk data.

These results indicate that, for our model, larger patch sizes at inference do not yield measurable improvements in restoration quality. This suggests that HAZEMATCHING's performance is not constrained by the spatial context provided during training, and the network can be reliably applied to full-resolution images via tiling when needed.

E. Baselines

We include the Richardson–Lucy (RL) deconvolution algorithm [26], a classical iterative method that operates on 3D image stacks and requires knowledge of the point spread function (PSF). We evaluate RL with varying numbers of iterations and select the best-performing result based on peak signal-to-noise ratio (PSNR). We denote this as RL_m, where m indicates the number of iterations. In the main paper we show the iteration that has the best PSNR. Detailed results across iterations are provided in the supplementary material (Figure S8). We use the implementation provided by `DeconvolutionLab2`[2] Fiji plugin, adapted to our 3D microscopy data.

We evaluate seven point-predictors: a U-NET [31], InDI_1 [10], MIMO-UNet [9], MPRNet [42], RCAN [46], Restormer [43], and ESRGAN [38] all of which produce a single prediction per input. For all of the baselines, we use a batch size of 16 except RCAN on Neuron data, where we use a batch size of 12, and Restormer, where we use a batch size of 8. We use a learning rate of 10^{-4} for all the models except Restormer where we follow the official implementation. In ESRGAN training, the VGG-based discriminator requires a minimum input resolution of 128×128. Since the *Neuron* dataset provides only 64×64 patches, we upsample each predicted patch to 128×128 via nearest-neighbour interpolation before feeding it to the discriminator. Inference remains unchanged. We use the open-source [3] implementation for RCAN and ESRGAN, and the official implementations for MIMO-UNet and MPRNet.

We also evaluate the iterative variant InDI_{20}, which runs the prediction for 20 refinement steps. For InDI, we adopt the training settings specified for the `defocus deblurring` task in their original paper [10]. As the official implementation of InDI is not available, we re-implement it following the setup for the `defocus deblurring` task using code available at [32]. We use a sampling schedule for the `defocus deblurring` task, train with an L_1 loss, a learning rate of 10^{-4}, batch size of 16, and the Adam optimizer.

We also compare with LVAE [34], a ladder VAE model capable of generating diverse predictions from a single input in one forward pass. We use a learned top-level prior initialized to zero. Training is done with a batch size of 16, a learning rate of 10^{-4}, and 5 levels of hierarchy (reduced to 4 for the *Neuron* dataset due to its smaller size).

We present several configurations of SIFM ([1]): $\text{SIFM}\sigma_{a=0.0|b\geq 0.0}$ and $\text{SIFM}\sigma_{a>0.0|b=a}$. These approaches can generate a single output ($\text{SIFM}\sigma_{a=0.0|b=0.0}$) or can generate multiple predictions in an iterative manner (like HAZEMATCHING). Note that for this approach we test different values of a and b for SIFM baselines but report only the configuration with best PSNR. We report the PSNR of all the configurations in Figure S9. As the official code is not available, we use the same HAZEMATCHING code with change in the loss function according to[1].

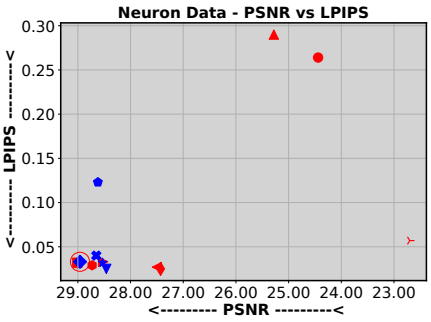
For the real dataset *Organoids1*, we also compare against the proprietary software *Elements* from [27] and for the real dataset *Organoids2*, we compare against the proprietary software *Lightning* from [20]. The *Elements* AR 5.42.02 (Nikon) was used for deconvolution. It was performed using the NIS-integrated deconvolution tool. The deconvolution parameters were set as follows: deconvolution type: 2D; modality: widefield; numerical aperture: 1.25; immersion refractive index: 1.406; calibration: $0.1902\mu m$; iterations: 70-120. The *Lightning* software was from Leica Application Suite X (LAS X, Leica Microsystems) version 4.7.0.28176 and its integrated deconvolution module. The parameters used for the post-processing of the widefield images using Lightning are the following: strategy: adaptive; type: confocal; number of iterations: automatic; optimization: 0; contrast enhancement: automatic; cutoff: automatic; regularization parameter: 0; smoothing: none; excitation wavelength: 488nm; emission wavelength: 515nm; pinhole: 6.28 AU; normalization: range; objective numerical aperture: 1.40; immersion refractive index: 1.518; magnification: 63; z-offset: $0\mu m$; coverslip refractive index 1.523; coverslip thickness: 170 um; mounting medium: Prolong Glass; mounting medium refractive index: 1.520.

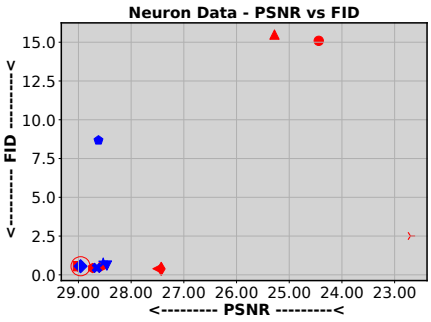
[2] DeconvolutionLab2: An Open-Source Software for Deconvolution Microscopy. Sage et. al, Methods—Image Processing for Biologists, 2017

[3] `BasicSR`: Image and Video Restoration Toolbox

F. Results

F.1. Neuron data results





Methods	PSNR ↑	LPIPS ↓	FID ↓
RL_2	24.44	0.264	15.090
U-Net	29.01	0.032	0.559
$InDI_1$	28.73	0.029	0.438
MIMO-UNet	27.50	0.027	0.395
MPRNet	28.52	0.033	0.495
RCAN	27.43	0.025	0.385
ESRGAN	22.70	0.057	2.505
Restormer	29.01	0.030	0.487
$InDI_{20}$	25.28	0.290	15.493
$SIFM\sigma_{0.0\|0.0}$	28.46	0.025	0.412
LVAE	28.62	0.123	8.677
$SIFM\sigma_{0.0\|0.2}$	28.53	0.032	0.792
$SIFM\sigma_{1.0\|1.0}$	28.65	0.040	0.450
HAZEMATCHING (ours)	28.96	0.033	0.554

Figure S3. **Data fidelity vs. realism for the *Neuron* dataset.** We show PSNR vs. LPIPS (left) and PSNR vs. FID (center), capturing the trade-off between pixel-level fidelity and perceptual quality. Our goal is to find a method that leads to high fidelity (high PSNR) while also leading to realistic looking predictions (low LPIPS/FID). (HAZEMATCHING is highlighted with an additional red circle.) Results tables (right) further summarize our results. Note that HAZEMATCHING consistently achieves balanced performance across all metrics and datasets. Methods displayed in red are point-predicting baselines, while blue methods are generative posterior models (see main text).

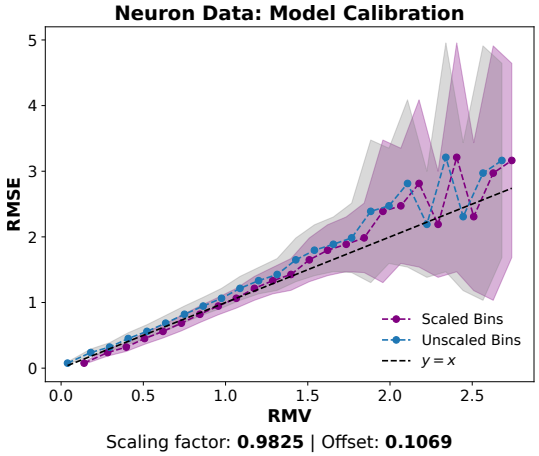
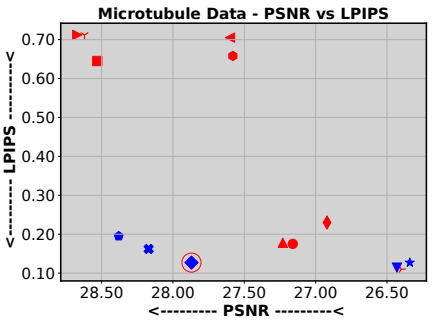
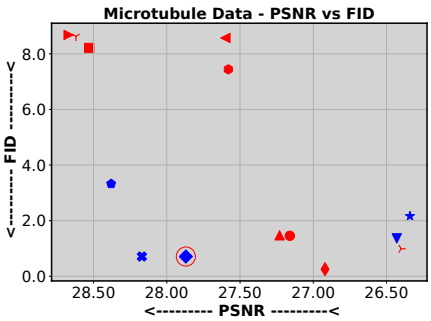


Figure S4. **Calibration of HAZEMATCHING.** RMSE vs. predicted RMV is shown for the *Neuron* data. The dashed line indicates ideal calibration ($y = x$). Blue and purple circles show uncalibrated and calibrated results, respectively, with shaded areas denoting standard error. Calibration parameters (scaling and offset) are shown below.

F.2. Microtubule data results





Methods	PSNR ↑	LPIPS ↓	FID ↓
RL_1	27.16	0.175	1.456
UNet	28.53	0.645	8.209
$InDI_1$	27.58	0.658	7.441
MIMO-UNet	27.60	0.705	8.570
MPRNet	28.67	0.712	8.679
RCAN	26.92	0.230	0.254
ESRGAN	26.40	0.110	0.989
Restormer	28.62	0.712	8.626
$InDI_{20}$	27.23	0.178	1.473
$SIFM\sigma_{0.0\|0.0}$	26.43	0.114	1.243
LVAE	28.38	0.195	3.325
$SIFM\sigma_{0.0\|0.1}$	26.34	0.127	2.164
$SIFM\sigma_{1.0\|1.0}$	28.17	0.162	0.711
HAZEMATCHING (ours)	27.87	0.127	0.710

Figure S5. **Data fidelity vs. realism for the *Microtubule* dataset.** We show PSNR vs. LPIPS (left) and PSNR vs. FID (center), capturing the trade-off between pixel-level fidelity and perceptual quality. Our goal is to find a method that leads to high fidelity (high PSNR) while also leading to realistic looking predictions (low LPIPS/FID). (HAZEMATCHING is highlighted with an additional red circle.) Results tables (right) further summarize our results. Note that HAZEMATCHING consistently achieves balanced performance across all metrics and datasets. Methods displayed in red are point-predicting baselines, while blue methods are generative posterior models (see main text).

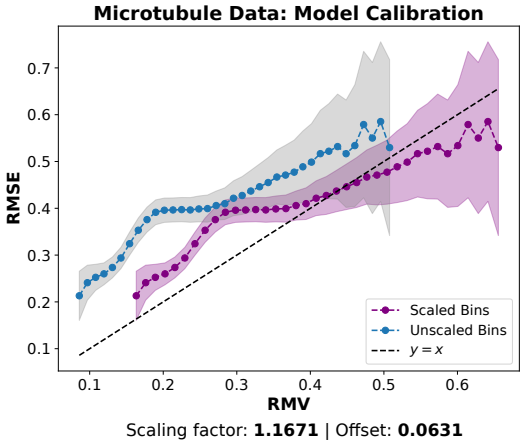
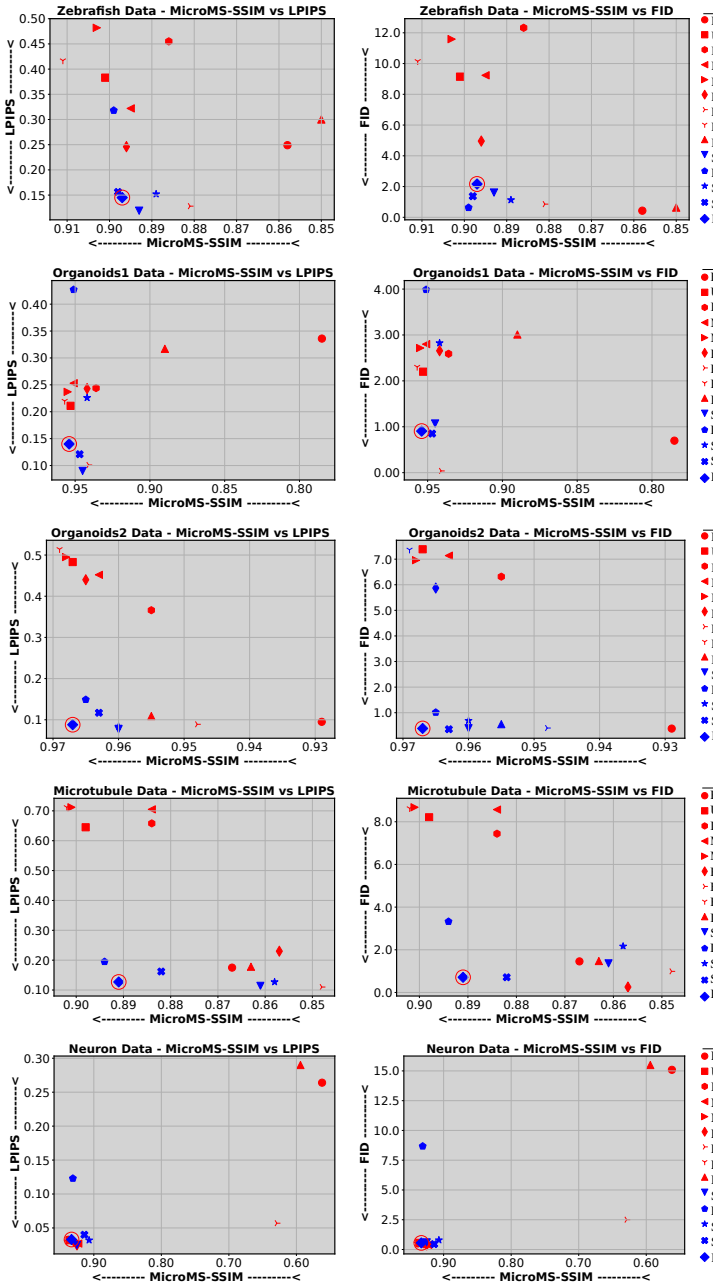


Figure S6. **Calibration of HAZEMATCHING.** RMSE vs. predicted RMV is shown for the *Microtubule* data. The dashed line indicates ideal calibration ($y = x$). Blue and purple circles show uncalibrated and calibrated results, respectively, with shaded areas denoting standard error. Calibration parameters (scaling and offset) are shown below.

F.3. MicroMS-SSIM vs LPIPS/FID



Methods	MMS-SSIM ↑	LPIPS ↓	FID ↓
RL_3	0.858	0.249	0.430
U-Net	0.901	0.383	9.146
$InDI_1$	0.886	0.455	12.326
MIMO-UNet	0.895	0.322	9.235
MPRNet	0.903	0.482	11.587
RCAN	0.896	0.246	4.957
ESRGAN	0.881	0.128	0.857
Restormer	0.911	0.417	10.148
$InDI_{20}$	0.850	0.300	0.633
$SIFM_{\sigma_{0.0\|0.0}}$	0.893	0.119	0.917
LVAE	0.899	0.318	0.631
$SIFM_{\sigma_{0.0\|0.1}}$	0.889	0.152	1.128
$SIFM_{\sigma_{0.2\|0.2}}$	0.898	0.156	1.372
HAZEMATCHING (ours)	0.897	0.145	2.165

Methods	MMS-SSIM ↑	LPIPS ↓	FID ↓
Elements	0.785	0.336	0.697
U-Net	0.953	0.211	2.200
$InDI_1$	0.936	0.244	2.590
MIMO-UNet	0.951	0.253	2.800
MPRNet	0.955	0.237	2.716
RCAN	0.942	0.243	2.656
ESRGAN	0.941	0.101	0.038
Restormer	0.957	0.220	2.303
$InDI_{20}$	0.890	0.317	3.010
$SIFM_{\sigma_{0.0\|0.0}}$	0.945	0.090	0.047
LVAE	0.951	0.427	3.988
$SIFM_{\sigma_{0.0\|0.1}}$	0.942	0.226	2.825
$SIFM_{\sigma_{0.1\|0.1}}$	0.947	0.121	0.851
HAZEMATCHING (ours)	0.954	0.140	0.905

Methods	MMS-SSIM ↑	LPIPS ↓	FID ↓
Lightning	0.929	0.095	0.380
U-Net	0.967	0.483	7.386
$InDI_1$	0.955	0.366	6.316
MIMO-UNet	0.963	0.452	7.138
MPRNet	0.968	0.495	6.959
RCAN	0.965	0.440	5.871
ESRGAN	0.948	0.089	0.401
Restormer	0.969	0.514	7.367
$InDI_{20}$	0.955	0.109	0.552
$SIFM_{\sigma_{0.0\|0.0}}$	0.960	0.077	0.164
LVAE	0.965	0.149	1.013
$SIFM_{\sigma_{0.0\|0.1}}$	0.960	0.084	0.690
$SIFM_{\sigma_{1.0\|1.0}}$	0.963	0.117	0.354
HAZEMATCHING (ours)	0.967	0.088	0.385

Methods	MMS-SSIM ↑	LPIPS ↓	FID ↓
RL_1	0.867	0.175	1.456
UNet	0.898	0.645	8.209
$InDI_1$	0.884	0.658	7.441
MIMO-UNet	0.884	0.705	8.570
MPRNet	0.901	0.712	8.679
RCAN	0.857	0.230	0.254
ESRGAN	0.848	0.110	0.989
Restormer	0.902	0.712	8.626
$InDI_{20}$	0.863	0.178	1.473
$SIFM_{\sigma_{0.0\|0.0}}$	0.861	0.114	1.243
LVAE	0.894	0.195	3.325
$SIFM_{\sigma_{0.0\|0.1}}$	0.858	0.127	2.164
$SIFM_{\sigma_{1.0\|1.0}}$	0.882	0.162	0.711
HAZEMATCHING (ours)	0.891	0.127	0.710

Methods	MMS-SSIM ↑	LPIPS ↓	FID ↓
RL_2	0.562	0.264	15.090
U-Net	0.936	0.032	0.559
$InDI_1$	0.929	0.029	0.438
MIMO-UNet	0.923	0.027	0.395
MPRNet	0.933	0.033	0.495
RCAN	0.925	0.025	0.385
ESRGAN	0.629	0.057	2.505
Restormer	0.940	0.030	0.487
$InDI_{20}$	0.594	0.290	15.493
$SIFM_{\sigma_{0.0\|0.0}}$	0.925	0.025	0.412
LVAE	0.931	0.123	8.677
$SIFM_{\sigma_{0.0\|0.2}}$	0.907	0.032	0.792
$SIFM_{\sigma_{1.0\|1.0}}$	0.914	0.040	0.450
HAZEMATCHING (ours)	0.933	0.033	0.554

Figure S7. **Data fidelity (MicroMS-SSIM [MMS-SSIM]) vs. realism (LPIPS and FID).** Each row corresponds to one dataset, *i.e.* *Zebrafish* (row 1), *Organoids1* (row 2), *Organoids2* (row 3), *Microtubule* (row 4), and *Neuron* (row 5) datasets. For each dataset, we show MicroMS-SSIM vs. LPIPS (left) and MicroMS-SSIM vs. FID (center), capturing the trade-off between pixel-level fidelity and structural/perceptual quality. The MicroMS-SSIM metric highlights the preservation of local structural details and consistency in the restored images, complementing pixel-wise fidelity metrics like PSNR. Our goal is to identify methods that achieve both high fidelity (high MicroMS-SSIM) and strong perceptual realism (low LPIPS and FID). (HAZEMATCHING is highlighted with an additional red circle.) Results tables (right) further summarize our results. Note that HAZEMATCHING consistently achieves a balanced trade-off across all metrics and datasets. Methods displayed in red are point-predicting baselines, while blue methods are generative posterior models (see main text).

F.4. Richardson Lucy iterations

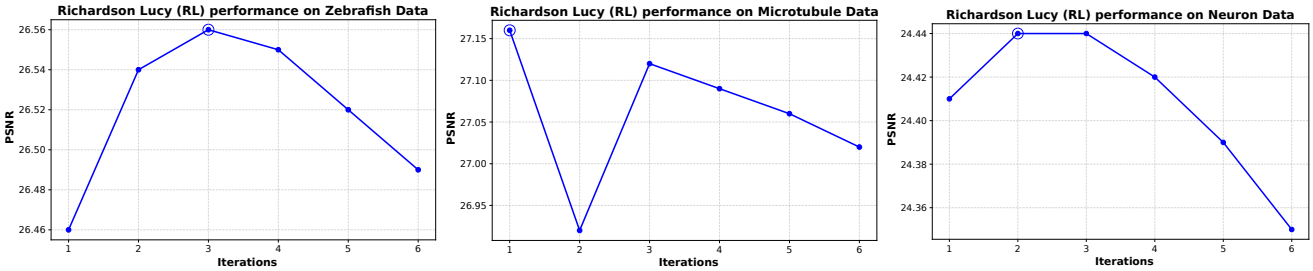


Figure S8. **Selection of optimal Richardson–Lucy iterations across datasets.** For each dataset, we evaluate multiple RL iterations and select the one that yields the highest PSNR (indicated by a an additional circle marker). This ensures a fair comparison against learning-based methods. Note that RL performance varies across datasets, highlighting the importance of iteration selection based on quantitative fidelity.

F.5. SIFM configurations

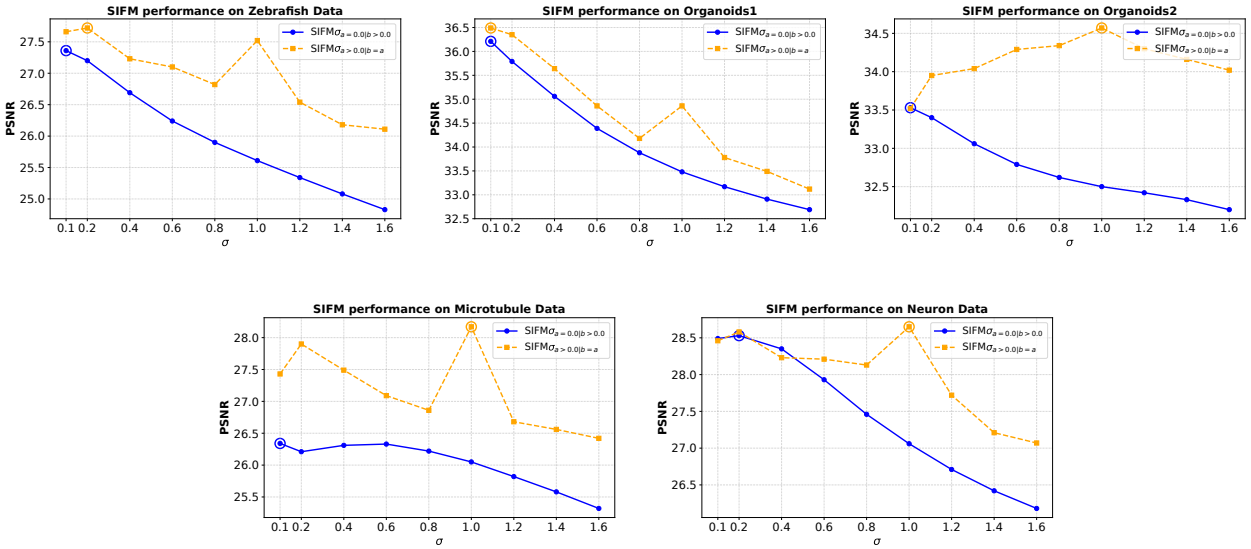


Figure S9. **Selection of optimal $\texttt{SIFM}_{a|b}$ configurations across datasets.** For each dataset, we evaluate multiple configurations of SIFM and chose the one for the main paper the one that achieves the highest PSNR (marked with a circle). This selection enables a fair comparison with other baselines. We experiment with two families of SIFM configurations: (1) **Blue solid lines:** SIFM$\sigma_{a=0.0|b\geq 0.0}$, where the model is trained without any additive noise (i.e., $a = 0.0$), and test-time noise of varying scale b is applied to encourage sample diversity. (2) **Orange dashed lines:** SIFM$\sigma_{a>0.0|b=a}$, where the same level of Gaussian noise $\mathcal{N}(0, I)$ is added both during training and inference. Here, a and b represent the scaling of the noise used during training and inference, respectively. This setup allows us to investigate the impact of noise injection at different stages and assess the trade-off between sample diversity and fidelity. The optimal configuration varies across datasets, reflecting the sensitivity of SIFM to hyperparameter tuning and data-specific characteristics.

F.6. Full quantitative results

Here we report PSNR, LPIPS, FID, MicroMS-SSIM, MS-SSIM, as well as two additional perceptual quality metrics: FSIM and GMSD. These metrics jointly capture pixel-wise fidelity, structural preservation, and perceptual realism. MicroMS-SSIM emphasizes fine structural detail, while LPIPS and FID evaluate perceptual quality based on learned features. FSIM evaluates perceptual similarity based on low-level features such as phase congruency and gradient magnitude, while GMSD quantifies image quality by measuring gradient magnitude similarity and is particularly sensitive to local distortions. Standard deviations are reported wherever applicable (mini-row at the bottom), computed across the test set for each dataset. Note that we consider the average of the samples for our perceptual metrics. In this case, for MMSE/point predictors only have one samples. Nevertheless for completeness we show the perceptual quality of the MMSE estimates for all the methods.

Dataset	Methods	Distortion on MMSE/Point-Prediction			Perception on MMSE/Point-Prediction				Average Perception on Samples			
		PSNR ↑	MS-SSIM ↑	MicroMS-SSIM ↑	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓
Zebrafish	RL_3	26.56 4.105	0.992 0.0018	0.858 0.0368	0.788 0.0351	0.249	0.430	0.166 0.0289				
	UNet	27.85 3.546	0.902 0.0138	0.901 0.0119	0.790 0.0646	0.383	9.146	0.151 0.0306				
	$InDI_1$	27.28 3.833	0.885 0.0201	0.886 0.0178	0.748 0.0907	0.455	12.326	0.165 0.0322				
	MIMO-UNet	27.74 3.538	0.891 0.0237	0.895 0.0128	0.773 0.0800	0.322	9.235	0.155 0.0312				
	MPRNet	28.08 3.590	0.904 0.0152	0.903 0.0122	0.769 0.0844	0.482	11.587	0.151 0.0310				
	RCAN	27.70 3.535	0.892 0.0221	0.896 0.0135	0.792 0.0591	0.246	4.957	0.151 0.0335				
	ESRGAN	27.07 3.841	0.878 0.0197	0.881 0.0234	0.811 0.0353	0.128	0.857	0.151 0.0305				
	Restormer	28.38 3.506	0.912 0.0159	0.911 0.0110	0.772 0.0814	0.417	10.148	0.150 0.0344				
	$InDI_{20}$	26.35 4.158	0.867 0.0208	0.850 0.0398	0.787 0.0326	0.300	0.633	0.157 0.0260				
	$SIFM_{\sigma_{0.0\|0.0}}$	27.42 3.583	0.888 0.0191	0.893 0.0151	0.823 0.0377	0.119	0.917	0.140 0.0318				
	LVAE	27.76 3.729	0.903 0.0126	0.899 0.0143	0.772 0.0759	0.329	6.490	0.149 0.0329	0.7911 0.0323	0.318 0.0210	0.631 0.2790	0.161 0.0361
	$SIFM_{\sigma_{0.0\|0.1}}$	27.36 3.651	0.885 0.0150	0.889 0.0187	0.822 0.0352	0.123	0.668	0.140 0.0302	0.8079 0.0352	0.152 0.0228	1.128 1.0379	0.148 0.0273
	$SIFM_{\sigma_{0.2\|0.2}}$	27.72 3.858	0.903 0.0121	0.898 0.0157	0.815 0.0468	0.179	3.054	0.142 0.0316	0.8093 0.0361	0.156 0.0146	1.372 1.3301	0.151 0.0288
	HAZEMATCHING (ours)	27.78 3.658	0.899 0.0200	0.897 0.0109	0.764 0.0849	0.439	9.996	0.154 0.0315	0.8061 0.0422	0.145 0.0182	2.165 1.7052	0.156 0.0331

Table S3. **Full quantitative comparison for the Zebrafish Dataset**

Dataset	Methods	Distortion on MMSE/Point-Prediction			Perception on MMSE/Point-Prediction				Average Perception on Samples			
		PSNR ↑	MS-SSIM ↑	MicroMS-SSIM ↑	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓
Organoids1	Elements	33.20 5.645	0.876 0.1215	0.785 0.0793	0.866 0.0336	0.336	0.697	0.103 0.0273				
	UNet	37.07 6.023	0.993 0.0045	0.953 0.0283	0.865 0.0510	0.211	2.200	0.119 0.0287				
	$InDI_1$	36.05 6.164	0.990 0.0054	0.936 0.0360	0.849 0.0554	0.244	2.590	0.131 0.0284				
	MIMO-UNet	36.84 6.044	0.993 0.0051	0.951 0.0278	0.857 0.0574	0.253	2.800	0.122 0.0276				
	MPRNet	37.39 6.073	0.993 0.0040	0.955 0.0273	0.865 0.0558	0.237	2.716	0.115 0.0281				
	RCAN	36.55 6.136	0.992 0.0049	0.942 0.0362	0.847 0.0598	0.243	2.656	0.134 0.0307				
	ESRGAN	36.16 6.080	0.991 0.0057	0.941 0.0324	0.852 0.0406	0.101	0.038	0.131 0.0268				
	Restormer	37.52 5.870	0.994 0.0043	0.957 0.0265	0.865 0.0584	0.220	2.303	0.113 0.0290				
	$InDI_{20}$	33.52 5.511	0.977 0.0163	0.890 0.0509	0.822 0.0316	0.317	3.010	0.140 0.0187				
	$SIFM_{\sigma_{0.0\|0.0}}$	36.57 6.526	0.991 0.0047	0.945 0.0301	0.869 0.0437	0.090	0.047	0.115 0.0299				
	LVAE	33.99 1.763	0.991 0.0050	0.951 0.0253	0.851 0.0471	0.197	0.491	0.123 0.0253	0.8031 0.0355	0.427 0.0247	3.988 3.1306	0.156 0.0320
	$SIFM_{\sigma_{0.0\|0.1}}$	36.21 6.392	0.987 0.0044	0.942 0.0287	0.865 0.0421	0.100	0.100	0.117 0.0292	0.8011 0.0668	0.226 0.0877	2.825 3.6920	0.147 0.0382
	$SIFM_{\sigma_{0.1\|0.1}}$	36.49 6.061	0.991 0.0051	0.947 0.0290	0.866 0.0542	0.155	1.070	0.113 0.0291	0.8472 0.0516	0.121 0.0531	0.851 0.1649	0.134 0.0369
	HAZEMATCHING (ours)	36.83 5.894	0.992 0.0042	0.954 0.0256	0.867 0.0533	0.236	2.650	0.114 0.0282	0.8449 0.0487	0.140 0.0755	0.905 0.5233	0.136 0.0288

Table S4. **Full quantitative comparison for the Organoids1 Dataset**

Dataset	Methods	Distortion on MMSE/Point-Prediction			Perception on MMSE/Point-Prediction				Average Perception on Samples			
		PSNR ↑	MS-SSIM ↑	MicroMS-SSIM ↑	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓
Organoids2	Lightning	32.38 1.358	0.935 0.0041	0.929 0.0102	0.824 0.0349	0.095	0.380	0.113 0.0248				
	UNet	35.08 1.159	0.977 0.0007	0.967 0.0057	0.825 0.0566	0.483	7.386	0.094 0.0302				
	InDI_1	34.00 1.057	0.967 0.0005	0.955 0.0076	0.813 0.0545	0.366	6.316	0.113 0.0290				
	MIMO-UNet	34.64 1.107	0.974 0.0010	0.963 0.0062	0.812 0.0625	0.452	7.138	0.100 0.0286				
	MPRNet	35.14 1.151	0.978 0.0011	0.968 0.0056	0.819 0.0613	0.495	6.959	0.092 0.0306				
	RCAN	34.92 1.142	0.977 0.0009	0.965 0.0054	0.825 0.0538	0.440	5.871	0.092 0.0288				
	ESRGAN	32.75 1.144	0.961 0.0018	0.948 0.0107	0.831 0.0347	0.089	0.401	0.109 0.0273				
	Restormer	35.18 1.162	0.978 0.0014	0.969 0.0053	0.818 0.0605	0.514	7.367	0.093 0.0309				
	InDI_{20}	33.28 1.039	0.965 0.0007	0.955 0.0073	0.840 0.0316	0.109	0.552	0.097 0.0232				
	$\text{SIFM}_{\sigma_{0.0\|0.0}}$	33.57 1.223	0.970 0.0015	0.960 0.0080	0.847 0.0358	0.077	0.164	0.093 0.0276				
	LVAE	34.73 1.066	0.975 0.0013	0.965 0.0057	0.799 0.0586	0.448	4.592	0.098 0.0287	0.8205 0.0341	0.149 0.0172	1.013 0.5901	0.120 0.0292
	$\text{SIFM}_{\sigma_{0.0\|0.1}}$	33.53 1.229	0.970 0.0015	0.960 0.0080	0.847 0.0356	0.076	0.196	0.093 0.0278	0.8414 0.0389	0.084 0.0098	0.690 0.3667	0.098 0.0307
	$\text{SIFM}_{\sigma_{1.0\|1.0}}$	34.57 1.113	0.974 0.0012	0.963 0.0060	0.800 0.0576	0.329	3.507	0.099 0.0291	0.8117 0.0420	0.117 0.0117	0.354 0.0948	0.138 0.0294
	HAZEMATCHING (ours)	35.02 1.113	0.977 0.0004	0.967 0.0058	0.825 0.0542	0.435	5.517	0.093 0.0290	0.8300 0.0355	0.088 0.0085	0.385 0.1600	0.113 0.0271

Table S5. **Full quantitative comparison for the Organoids2 Dataset**

Dataset	Methods	Distortion on MMSE/Point-Prediction			Perception on MMSE/Point-Prediction				Average Perception on Samples			
		PSNR ↑	MS-SSIM ↑	MicroMS-SSIM ↑	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓
Microtubule	RL_1	27.16 0.749	0.994 0.0005	0.867 0.0126	0.812 0.0283	0.175	1.456	0.131 0.0207				
	U-Net	28.53 0.729	0.906 0.0089	0.898 0.0087	0.779 0.0367	0.645	8.209	0.131 0.0190				
	InDI_1	27.58 0.657	0.897 0.0091	0.884 0.0064	0.759 0.0425	0.658	7.441	0.133 0.0188				
	MIMO-UNet	27.60 0.659	0.896 0.0082	0.884 0.0077	0.752 0.0519	0.705	8.570	0.141 0.0183				
	MPRNet	28.67 0.782	0.908 0.0087	0.901 0.0082	0.770 0.0402	0.712	8.679	0.131 0.0192				
	RCAN	26.92 0.830	0.869 0.0114	0.857 0.0157	0.806 0.0220	0.230	0.254	0.146 0.0196				
	ESRGAN	26.40 0.772	0.860 0.0103	0.848 0.0168	0.818 0.0194	0.110	0.989	0.142 0.0203				
	Restormer	28.62 0.755	0.909 0.0080	0.902 0.0078	0.764 0.0454	0.712	8.626	0.130 0.0184				
	InDI_{20}	27.23 0.745	0.876 0.0095	0.863 0.0140	0.814 0.0174	0.178	1.473	0.131 0.0199				
	$\text{SIFM}_{\sigma_{0.0\|0.0}}$	26.43 0.734	0.872 0.0104	0.861 0.0158	0.822 0.0194	0.114	1.243	0.133 0.0225				
	LVAE	28.38 0.797	0.900 0.0090	0.894 0.0082	0.759 0.0382	0.458	2.927	0.130 0.0174	0.8030 0.0184	0.195 0.0327	3.325 1.0124	0.150 0.0203
	$\text{SIFM}_{\sigma_{0.0\|0.1}}$	26.34 0.722	0.870 0.0108	0.858 0.0161	0.822 0.0191	0.108	1.357	0.132 0.0220	0.8159 0.0188	0.127 0.0203	2.164 0.6513	0.139 0.0216
	$\text{SIFM}_{\sigma_{1.0\|1.0}}$	28.17 0.744	0.886 0.0103	0.882 0.0082	0.734 0.0436	0.487	4.272	0.137 0.0188	0.7996 0.0222	0.162 0.0201	0.711 0.3178	0.165 0.0231
	HAZEMATCHING (ours)	27.87 0.755	0.904 0.0085	0.891 0.0073	0.758 0.0433	0.607	6.304	0.129 0.0181	0.8151 0.0200	0.127 0.0139	0.710 0.2743	0.142 0.0217

Table S6. **Full quantitative comparison for the Microtubule Dataset**

F.7. Inference runtime comparison

Below in Table S8, we report wall-clock inference times (averaged over 5 runs on an NVIDIA V100) on an image of size 1024×1024, alongside model sizes and sampling capabilities.

Parameter Efficiency At **3.7M** parameters, HAZEMATCHING is more compact than UNet (13.4M) and still supports iterative refinement. Given that iterative generative models are generally computationally intensive, HAZEMATCHING delivers a **practical runtime** ($\sim$ 4.7 s for 1024×1024 images) while enabling **multi-sample posterior inference**, outperforming other iterative methods by over an order of magnitude in speed.

Dataset	Methods	Distortion on MMSE/Point-Prediction			Perception on MMSE/Point-Prediction				Average Perception on Samples			
		PSNR ↑	MS-SSIM ↑	MicroMS-SSIM ↑	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓	FSIM ↑	LPIPS ↓	FID ↓	GMSD ↓
Neuron	RL_2	24.44 2.335	0.493 0.0953	0.562 0.1081	0.458 0.1399	0.264	15.090	0.310 0.0225				
	UNet	29.01 2.956	0.980 0.0206	0.936 0.0532	0.877 0.0869	0.032	0.559	0.108 0.0587				
	$InDI_1$	28.73 2.914	0.973 0.0331	0.929 0.0616	0.890 0.0736	0.029	0.438	0.110 0.0598				
	MIMO-UNet	27.50 2.847	0.974 0.0232	0.923 0.0612	0.875 0.0719	0.027	0.395	0.124 0.0600				
	MPRNet	28.52 2.969	0.976 0.0273	0.933 0.0578	0.888 0.0927	0.033	0.495	0.118 0.0660				
	RCAN	27.43 2.893	0.975 0.0222	0.925 0.0622	0.881 0.0723	0.025	0.385	0.124 0.0606				
	ESRGAN	22.70 2.080	0.889 0.0455	0.629 0.0926	0.680 0.0987	0.057	2.505	0.270 0.0447				
	Restormer	29.01 2.984	0.978 0.0265	0.940 0.0558	0.903 0.0703	0.030	0.487	0.107 0.0634				
	$InDI_{20}$	25.28 2.454	0.948 0.0451	0.594 0.0983	0.468 0.1521	0.290	15.493	0.311 0.0238				
	$SIFM_{\sigma_{0.0\|0.0}}$	28.46 3.010	0.973 0.0330	0.925 0.0669	0.889 0.0774	0.025	0.412	0.115 0.0671				
	LVAE	28.62 2.823	0.977 0.0217	0.931 0.0591	0.870 0.0930	0.032	0.561	0.114 0.0622	0.7224 0.1061	0.123 0.0874	8.677 3.0876	0.310 0.0448
	$SIFM_{\sigma_{0.0\|0.2}}$	28.53 2.992	0.974 0.0303	0.907 0.0682	0.856 0.0944	0.029	0.559	0.118 0.0632	0.8490 0.0971	0.032 0.0386	0.792 1.1414	0.129 0.0658
	$SIFM_{\sigma_{1.0\|1.0}}$	28.65 2.935	0.971 0.0358	0.914 0.0732	0.856 0.0969	0.036	0.425	0.113 0.0593	0.8141 0.0877	0.040 0.0271	0.450 0.5625	0.191 0.0688
	HAZEMATCHING (ours)	28.96 2.961	0.978 0.0183	0.933 0.0541	0.887 0.0711	0.031	0.464	0.107 0.0563	0.8493 0.0856	0.033 0.0299	0.554 0.8043	0.147 0.0653

Table S7. **Full quantitative comparison for the Neuron Dataset**

Model	Mean (5) [sec.]	Std (5)	Per Forward Pass [sec.]	Params (M)
RL	13.6	3.16	N/A	N/A
Elements	3.233	0.136	N/A	N/A
Lightning	1.0 (est.)	N/A	N/A	N/A
UNet	0.2944	0.0393	N/A	13.39
MIMO-UNet	0.6358	0.0032	N/A	6.80
MPRNet	5.4713	0.0046	N/A	20.12
RCAN	8.6271	0.0293	N/A	15.29
ESRGAN	1.1839	0.0566	N/A	16.70
Restormer	4.5067	0.0255	N/A	26.12
$InDI_1$	7.1888	0.0930	N/A	2.44
$InDI_{20}$	143.3886	0.5544	7.16943	2.44
LVAE	1.0118	0.4468	N/A	5.67
SIFM	4.3120	0.0633	0.2156	3.70
HAZEMATCHING	4.7498	0.0049	0.23749	3.70

Table S8. **Inference runtime (mean and std. over 5 runs)**

F.8. Flexibility in Fidelity–Realism tradeoff (Organoids1 Dataset)

Integration Steps (T): Increasing the number of integration steps generally yields smoother flows and improved perceptual realism:

T **Step**	**LPIPS** $\downarrow$ (μ)	**LPIPS** (σ)	**PSNR** $\uparrow$ (μ)	**PSNR** (σ)	**Time (s)**
2	0.1419	0.0572	36.73	6.099	0.26
3	0.1843	0.0935	36.60	6.034	0.48
4	0.1938	0.1096	36.39	6.046	0.68
5	0.1939	0.1155	36.24	6.063	0.89
10	0.1753	0.1092	35.92	6.101	1.96
15	0.1570	0.0929	35.80	6.134	3.04
30	0.1158	0.0436	35.63	6.181	6.05
50	0.1124	0.0383	35.57	6.200	9.92
80	0.1117	0.0367	35.55	6.204	15.75
100	0.1114	0.0362	35.55	6.202	19.67
150	0.1116	0.0359	35.53	6.206	29.42
200	0.1115	0.0359	35.52	6.209	39.20
250	0.1115	0.0359	35.52	6.211	48.99

Table S9. Effect of integration steps T on perceptual (LPIPS) and distortion (PSNR) metrics and runtime per 1024×1024 sample. Note that time shown is for 1 sample.

Sample Averaging: Averaging more stochastic samples reduces variance and increases PSNR (fidelity). Results shown for $T = 20$:

# Samples	**PSNR** $\uparrow$ (μ)	**PSNR** (σ)
01	35.74	6.153
02	36.23	6.042
05	36.58	5.985
10	36.71	5.947
20	36.78	5.916
30	36.81	5.900
40	36.82	5.897
50	36.83	5.894

Table S10. Effect of number of posterior samples on PSNR (mean and std.) for $T = 20$.

F.9. Choice of conditioning mechanism

Following prior work in diffusion models, we adopt **concatenation** of the noisy input and conditioning observation, which has become standard for inverse problems [32]. Additionally, we perform ablation with addition of the condition to the input on *Organoids1* (Table S11).

Conditioning	**PSNR** $\uparrow$	**LPIPS** $\downarrow$
Concatenation (ours)	**36.83**	**0.140**
Element-wise add	33.66	0.217

Table S11. Ablation of conditioning mechanisms with HAZEMATCHING on *Organoids1*.

F.10. Comparison with conditional diffusion

Here, we compare PSNR (of the MMSE) and LPIPS (averaged over samples) between HAZEMATCHING and a conditional diffusion model [32]. For each dataset, we evaluate both methods using two posterior samples generated from a single test image. The conditional diffusion model is run with 1200 backward steps, while HAZEMATCHING uses $T = 20$ steps. The quantitative results are summarized in Table S12, along with the wall-clock time required to generate a single posterior sample for each dataset. The diffusion model takes considerably longer per sample, making it less practical to use.

Dataset	**HAZEMATCHING**			**Cond. Diffusion [32]**		
	PSNR ↑	LPIPS ↓	Time (s)	PSNR ↑	LPIPS ↓	Time (s)
Zebrafish (1024×1024)	31.43	0.170	4.15	30.34	0.144	10067.45
Organoids1 (1024×1024)	34.42	0.094	4.15	32.33	0.280	10244.28
Organoids2 (1024×1024)	32.97	0.096	4.15	30.07	0.415	10203.20
Microtubule (512×512)	25.82	0.145	1.24	23.96	0.128	2766.09
Neuron (64×64)	26.83	0.027	0.24	26.17	0.027	25.24

Table S12. Comparison of HAZEMATCHING and conditional diffusion across five datasets (1 test image, 2 posterior samples) in terms of PSNR, LPIPS, and wall-clock time per posterior sample.

F.11. Effect of different dynamic range regimes

We evaluate the robustness of the model to dynamic range variations by constructing two additional datasets from the original Organoids1 data: an "*OK*" variant obtained by scaling the intensity of the input images by 0.7 and adding a constant offset of 12, and a more extreme "*Bad*" version obtained by scaling the intensity by 0.2 with the same offset of 12. By training with these datasets, we obtain the *OK Model* and the *Bad Model*, respectively. Note that the *Good Model* is the original HAZEMATCHING model trained on the actual Organoids1 data.

All models are then evaluated across all data variants, and their performance is reported in Table S13. Across these settings, PSNR exhibits only a slight decrease under dynamic range mismatch, with a drop of less than $\sim$1 dB even in the most severe case, indicating that reconstruction fidelity is largely preserved. In contrast, LPIPS remains effectively unchanged across all combinations of training and testing conditions, suggesting that perceptual similarity is invariant to such intensity transformations. Overall, these results demonstrate that the model is robust to moderate and even strong dynamic range shifts, with only minimal impact on pixel-wise accuracy.

Due to the data normalization prior to feeding input patches, these results might be, at most, moderately surprising to some readers. We attribute this section to the curiosity of one of our reviewers.

Metric	**Data**	**Good Model**	**OK Model**	**Bad Model**
PSNR ↑	Good (100%)	36.83	35.48	35.22
	OK (70%)	35.72	35.47	35.22
	Bad (20%)	35.73	35.47	35.22
LPIPS ↓	Good (100%)	0.140	0.135	0.138
	OK (70%)	0.140	0.135	0.138
	Bad (20%)	0.140	0.135	0.138

Table S13. Effect of dynamic range variations on model performance. The OK and Bad datasets are generated by scaling the intensity (0.7 and 0.2, respectively) and adding a constant offset of 12.

F.12. More qualitative results

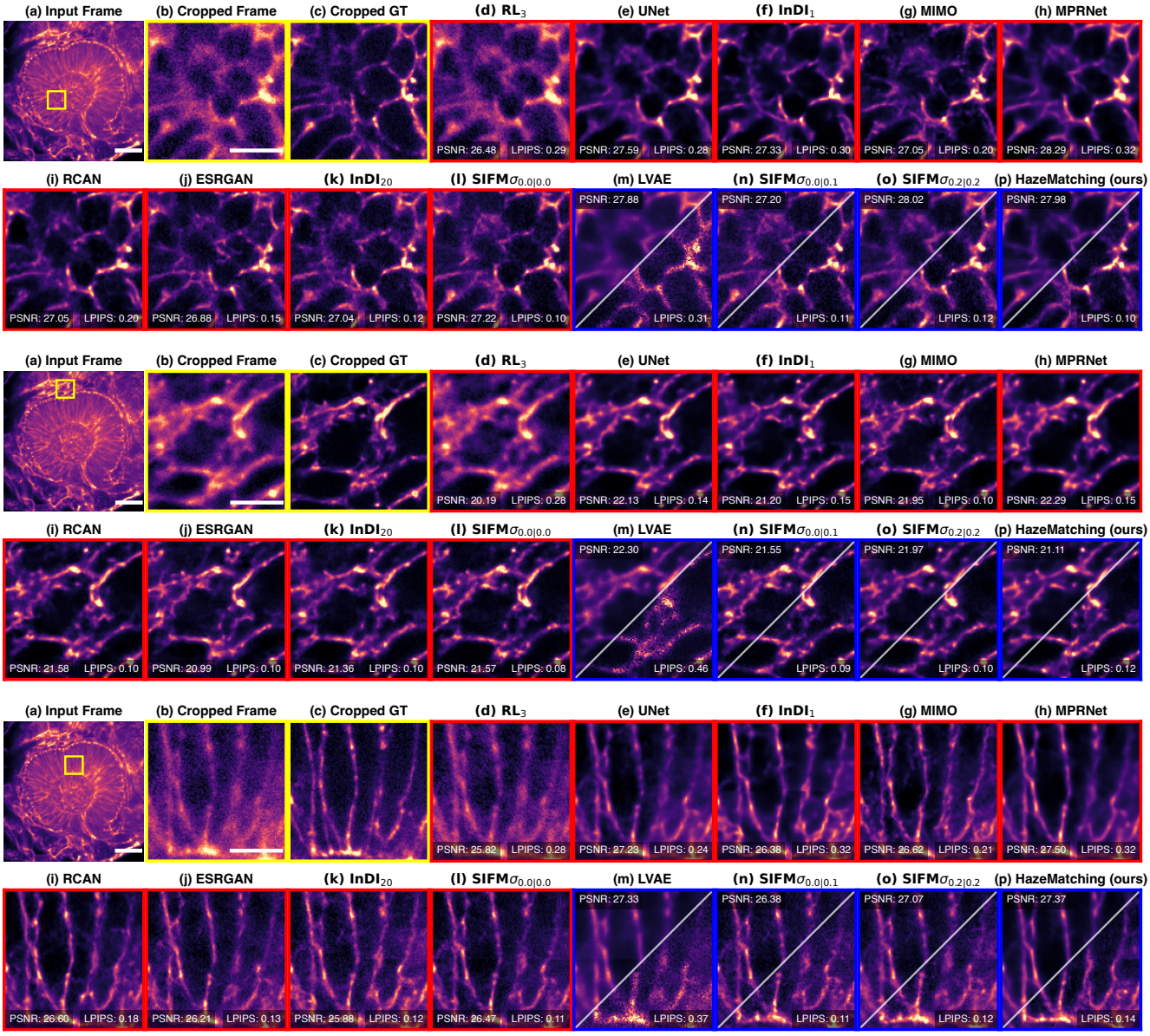


Figure S10. **Qualitative results on Zebrafish Data:** Here we present three examples showing Nuclear Membrane. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 40 μm, **(b)** the selected crop; Scalebar: 10 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

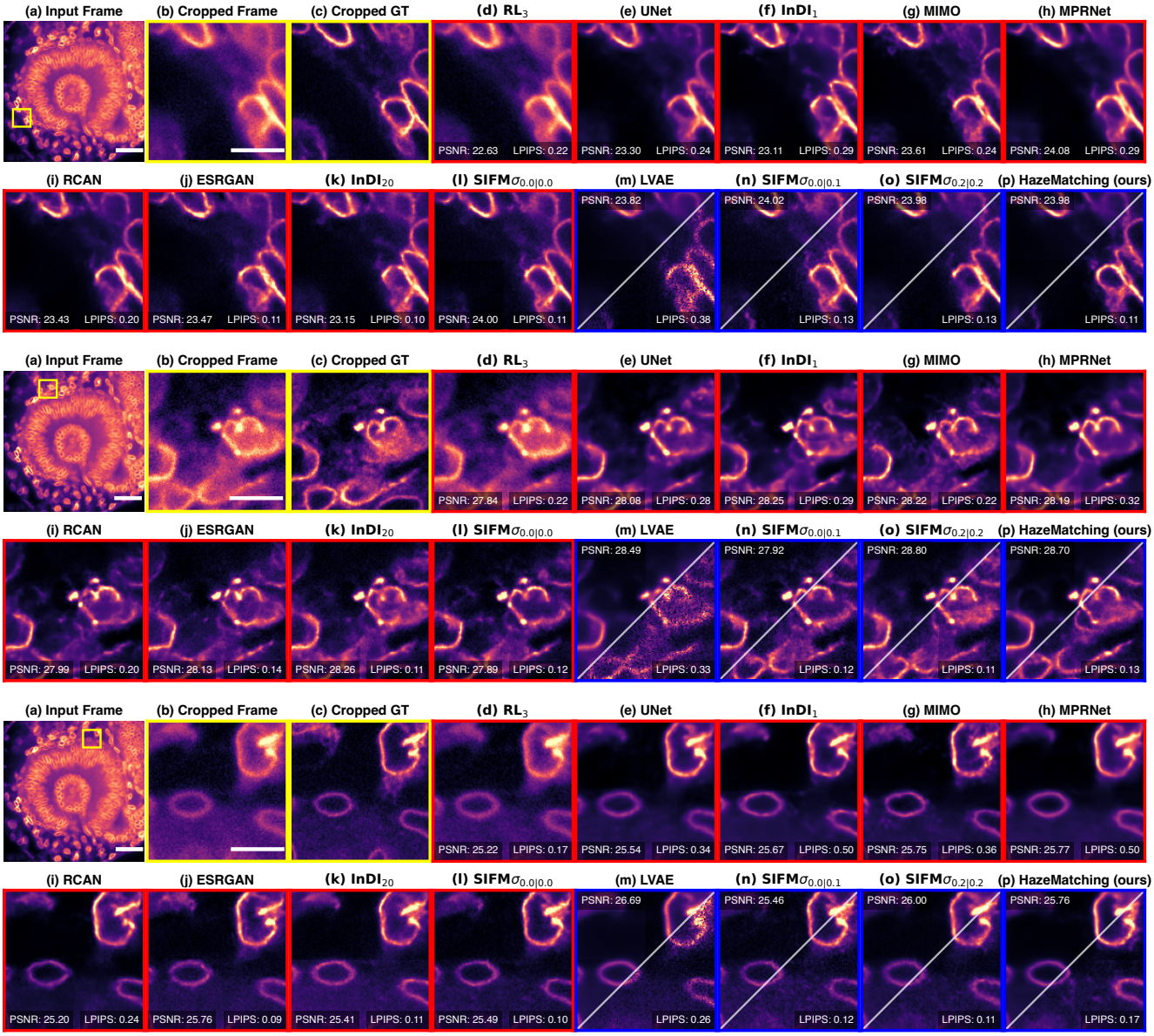


Figure S11. **Qualitative results on Zebrafish Data:** Here we present three examples showing Nuclear Envelope. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 40 μm, **(b)** the selected crop; Scalebar: 10 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

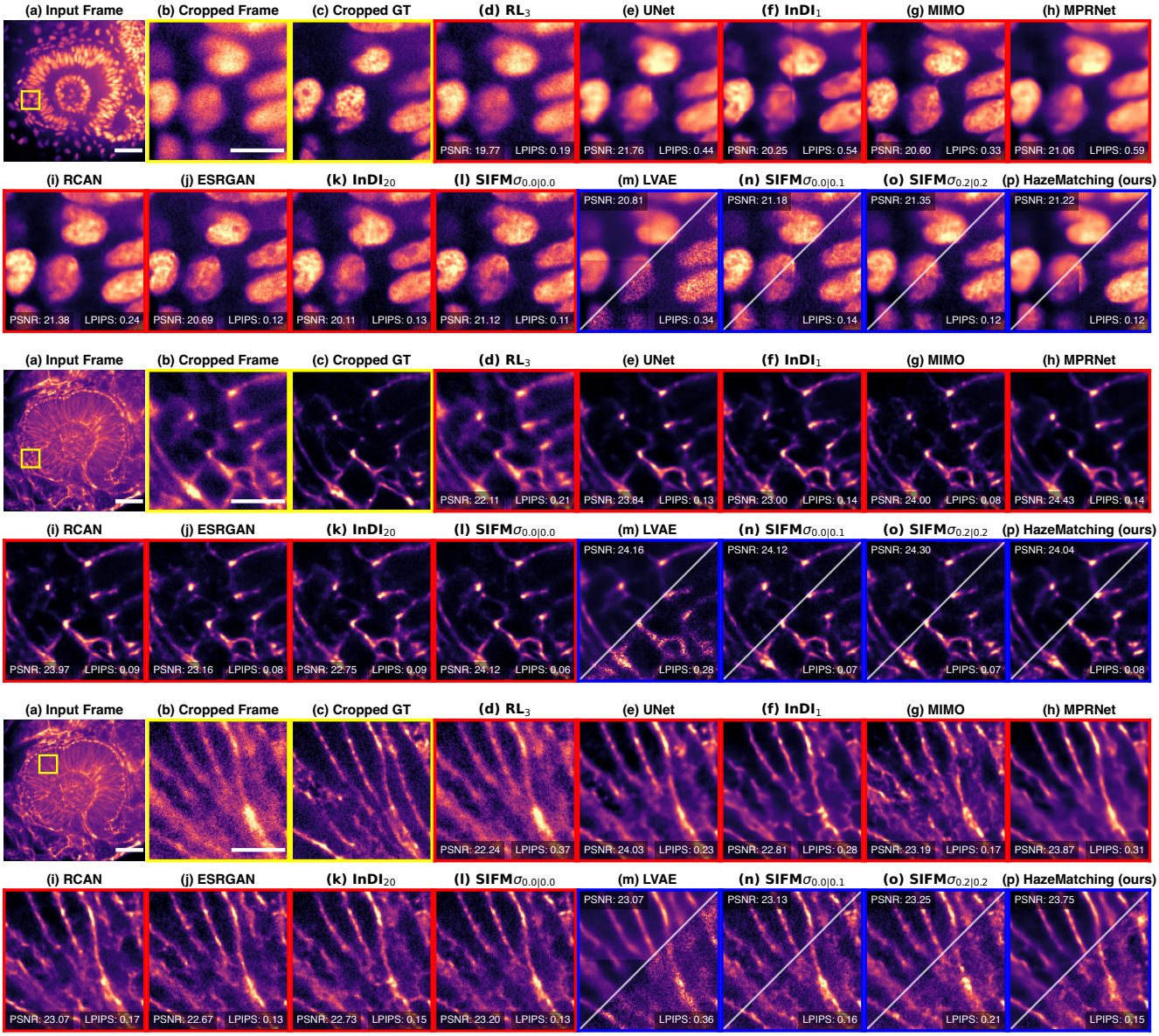


Figure S12. **Qualitative results on Zebrafish Data:** Here we present three examples, each illustrating two distinct structures: the Nuclei and the Nuclear Membrane. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 40 μm, **(b)** the selected crop; Scalebar: 10 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

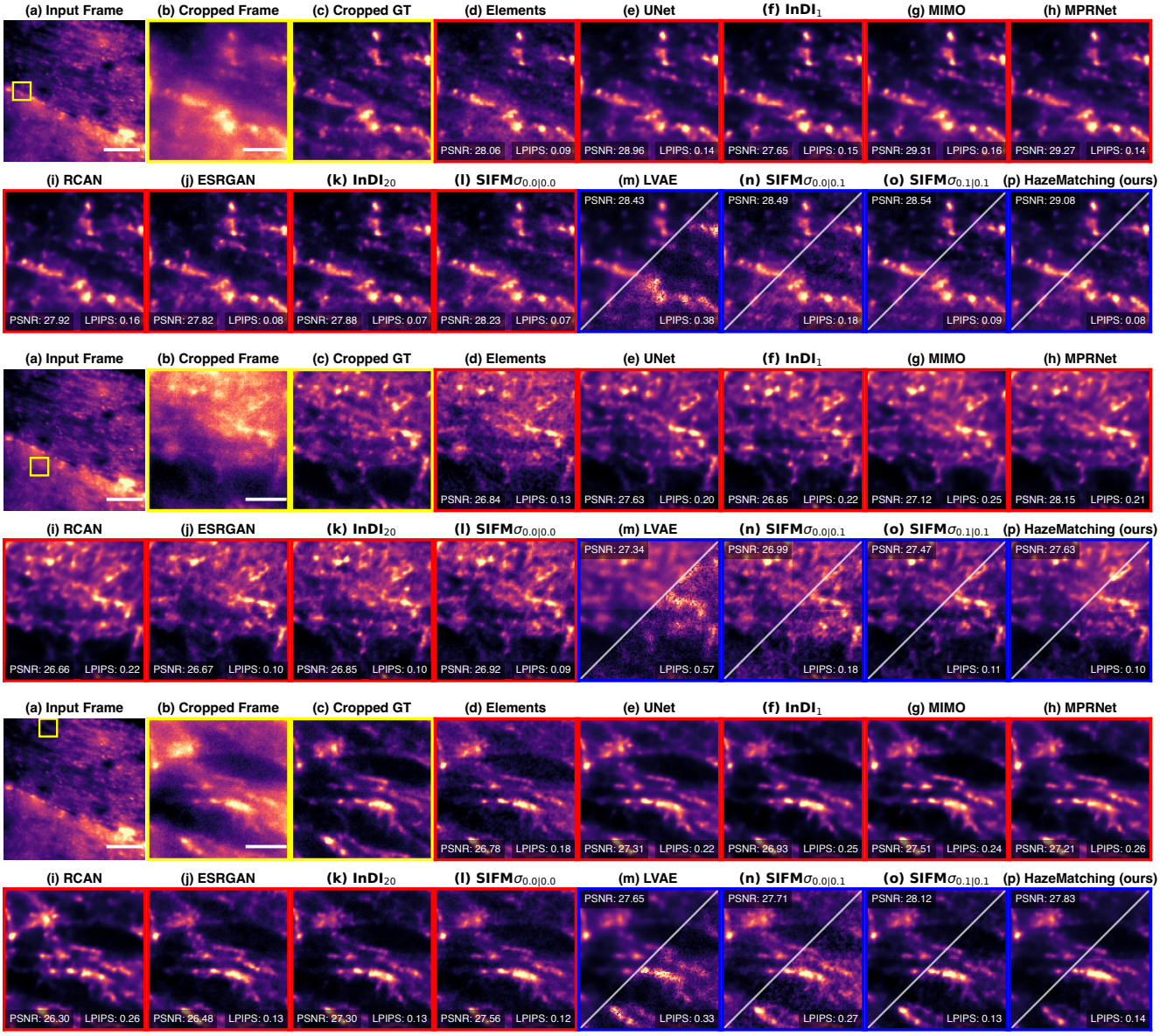


Figure S13. **Qualitative results on Organoids1 Data:** Here we present three examples showing Brain Organoids. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 50 μm, **(b)** the selected crop; Scalebar: 10 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

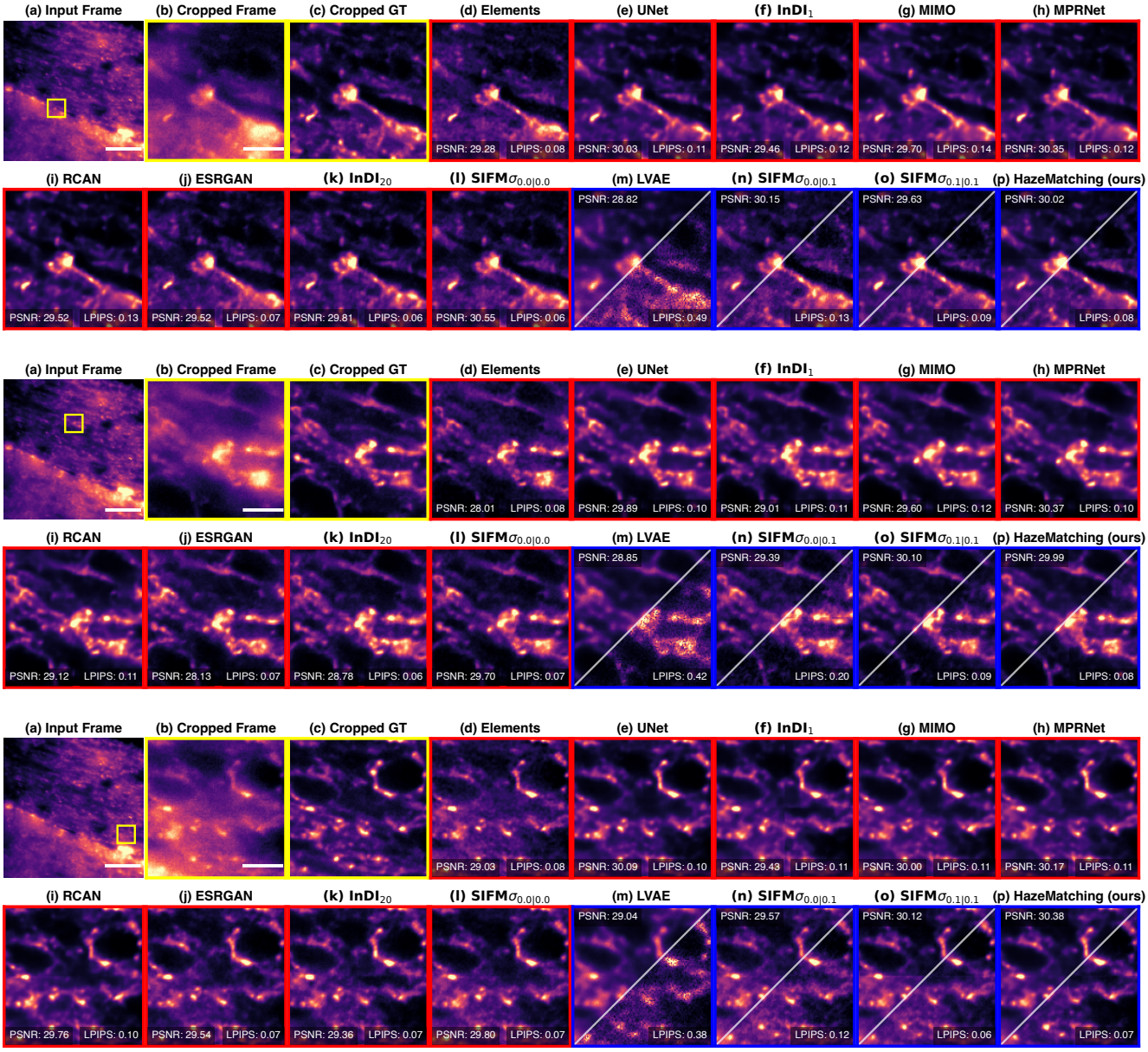


Figure S14. **Qualitative results on Organoids1 Data:** Here we present three examples showing Brain Organoids. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 50 μm, **(b)** the selected crop; Scalebar: 10 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

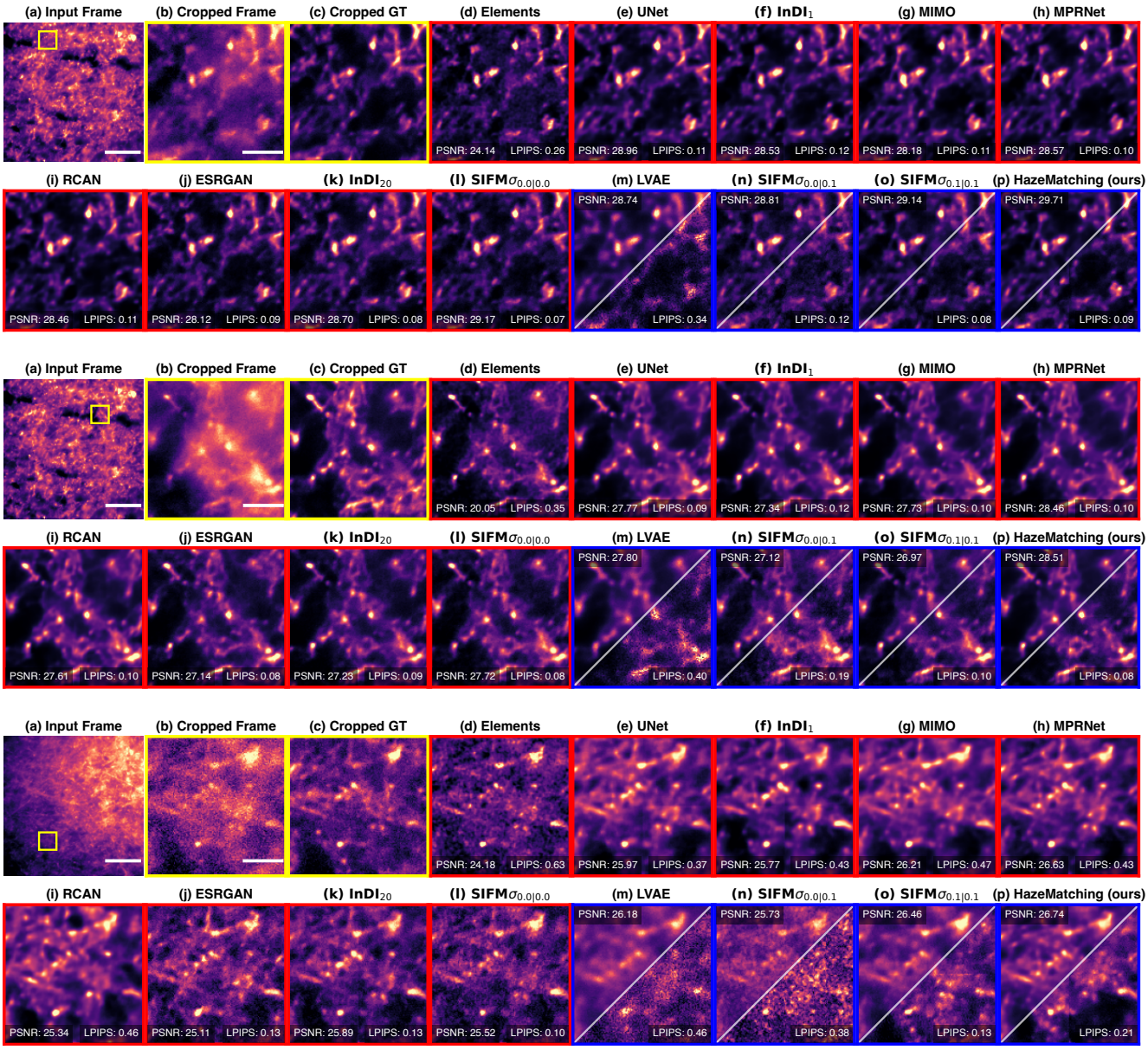


Figure S15. **Qualitative results on Organoids1 Data:** Here we present three examples showing Brain Organoids. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 50 μm, **(b)** the selected crop; Scalebar: 10 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

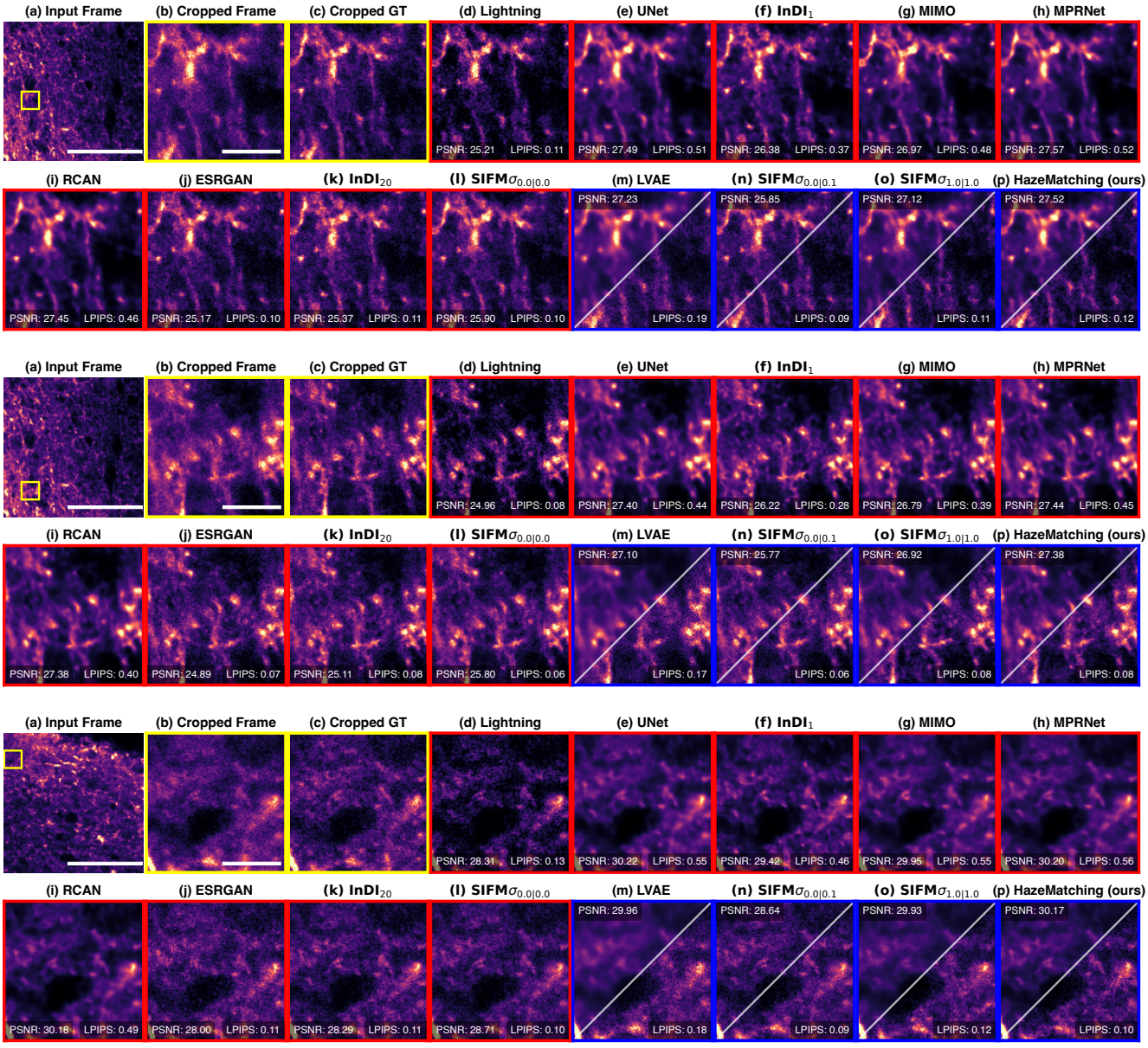


Figure S16. **Qualitative results on Organoids2 Data:** Here we present three examples showing Brain Organoids. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 50 μm, **(b)** the selected crop; Scalebar: 5 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

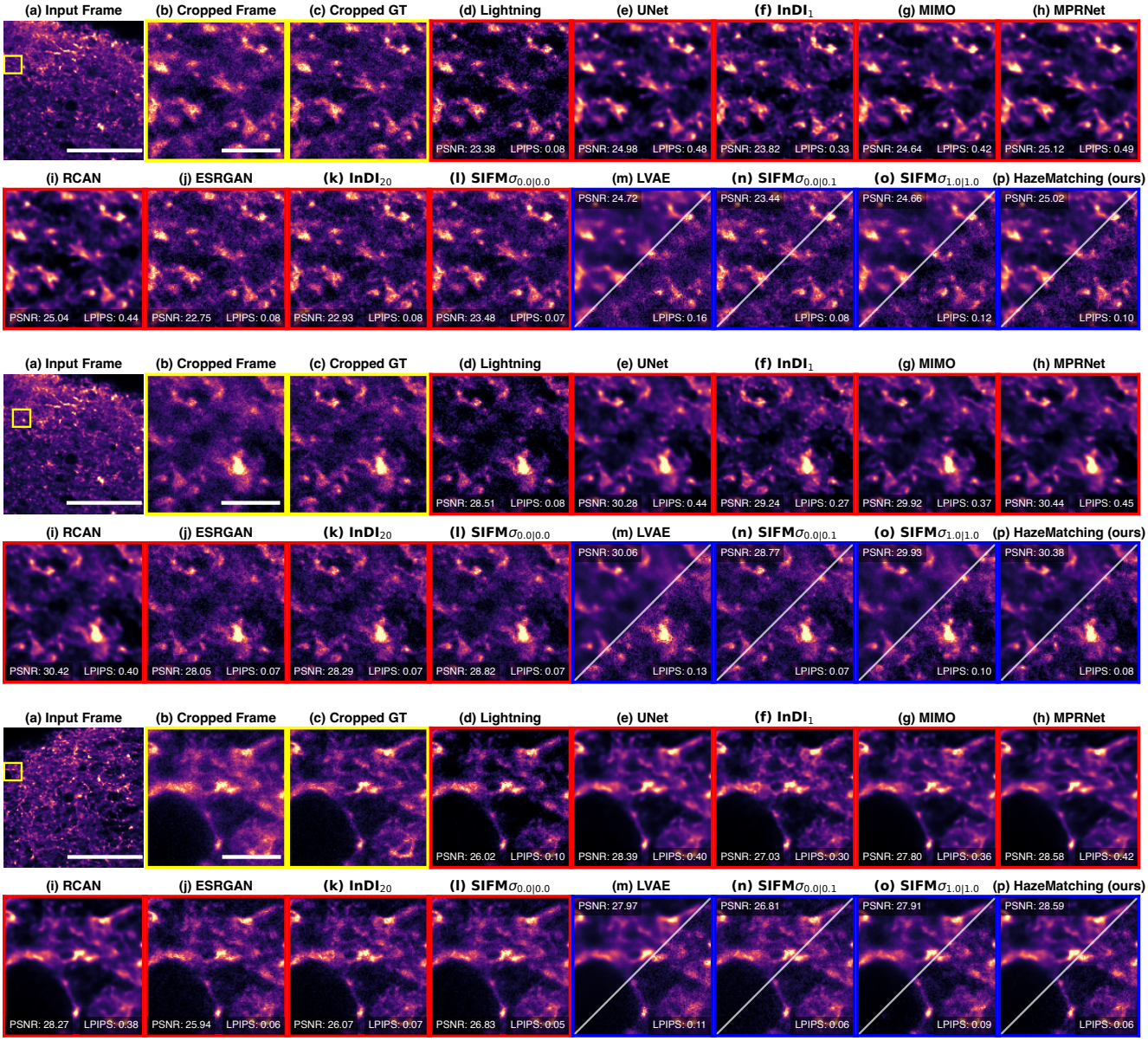


Figure S17. **Qualitative results on Organoids2 Data:** Here we present three examples showing Brain Organoids. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 50 μm, **(b)** the selected crop; Scalebar: 5 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

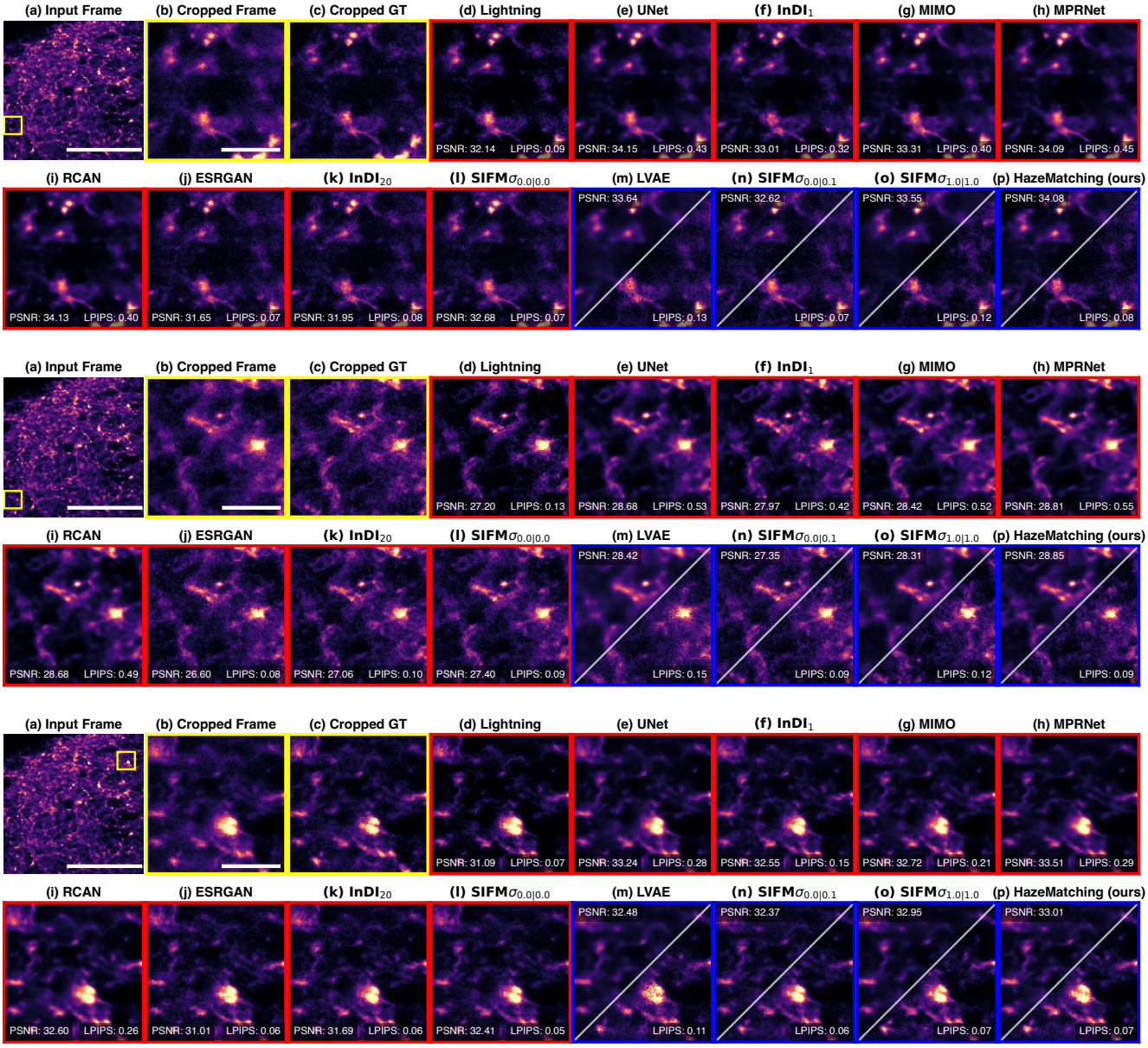


Figure S18. **Qualitative results on Organoids2 Data:** Here we present three examples showing Brain Organoids. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 50 μm, **(b)** the selected crop; Scalebar: 5 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

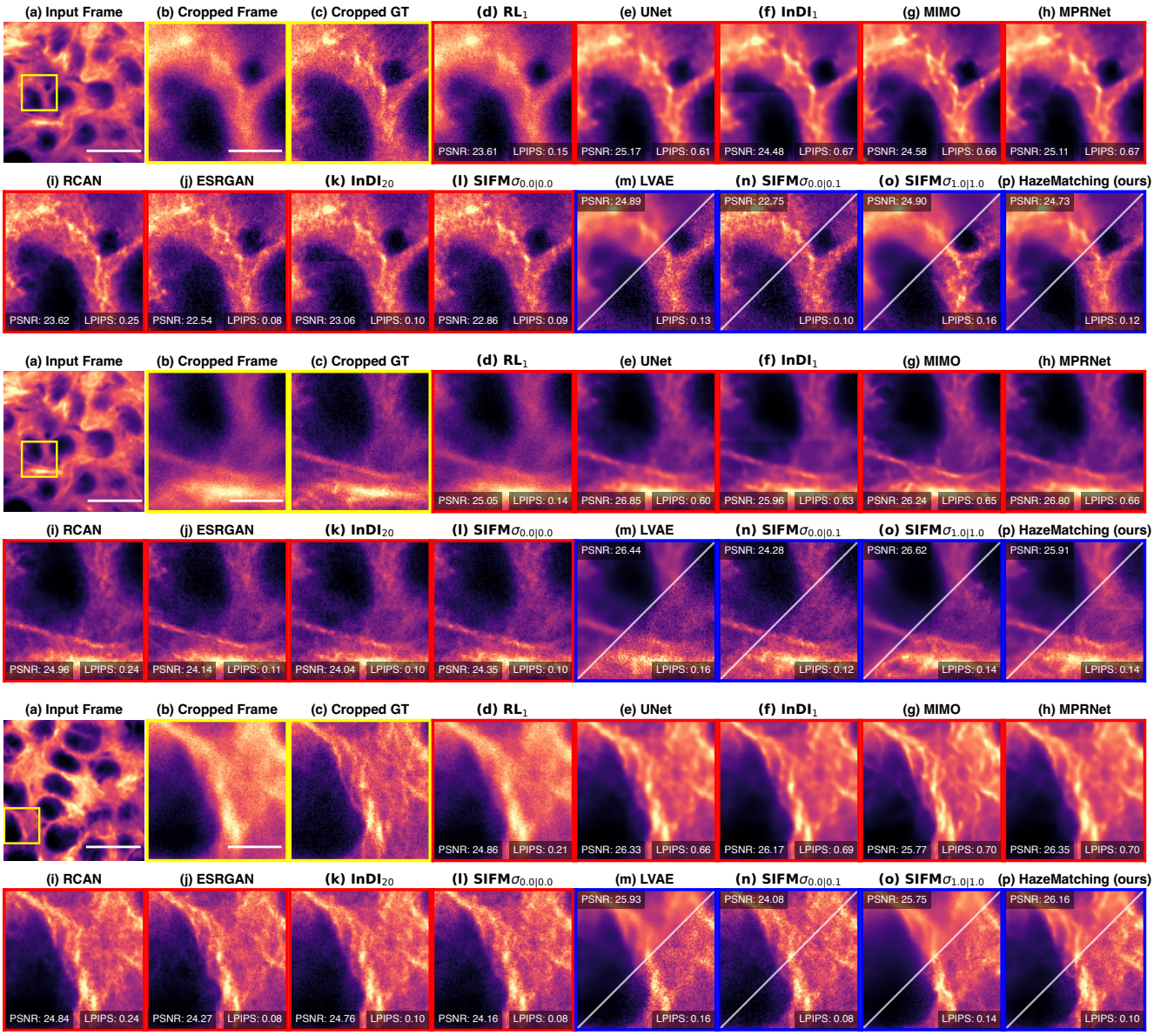


Figure S19. **Qualitative results on Microtubule Data:** Here we present three examples showing Microtubules tagged with α-Tubulin. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 20 μm, **(b)** the selected crop; Scalebar: 5 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

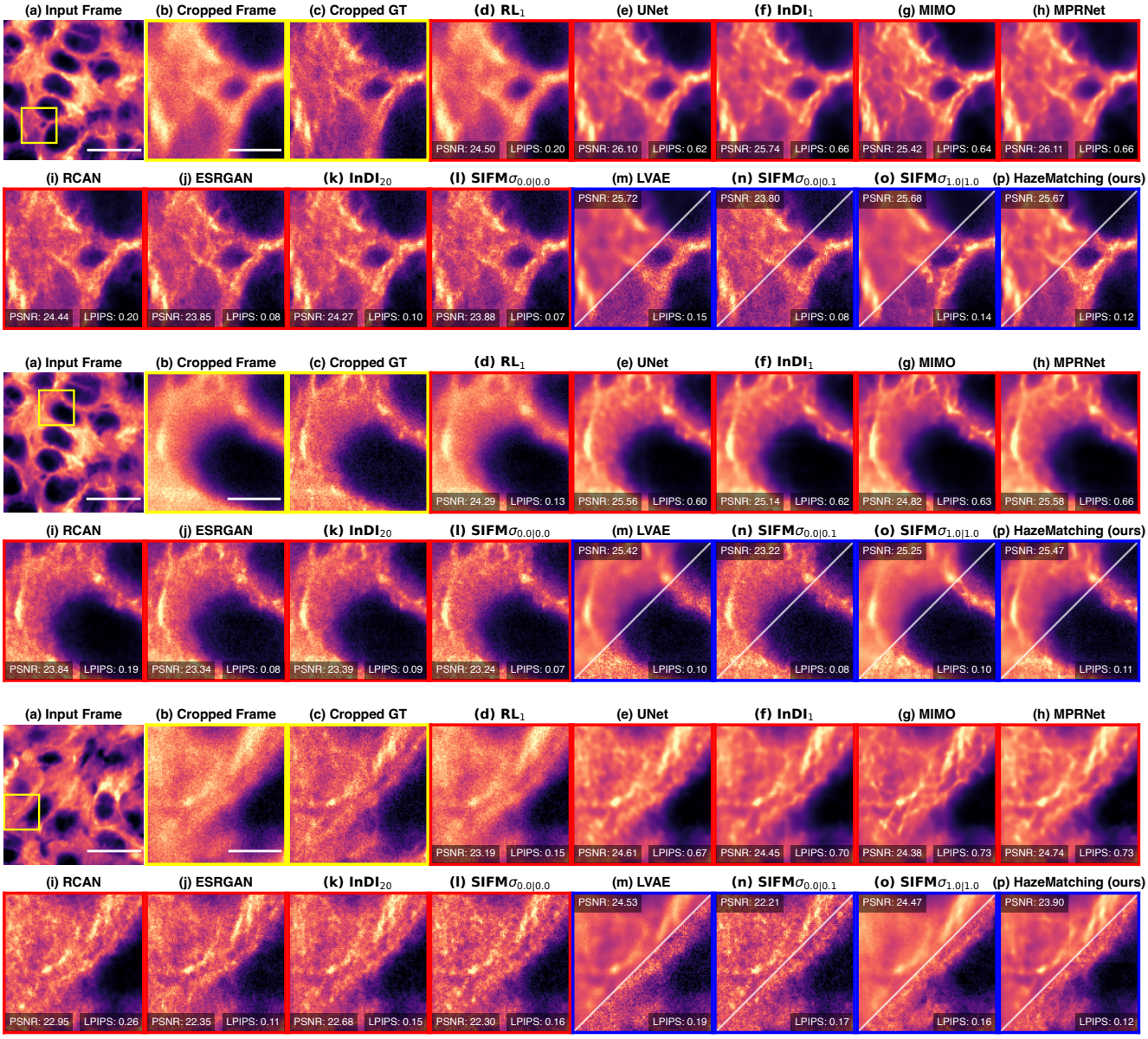


Figure S20. **Qualitative results on Microtubule Data:** Here we present three examples showing Microtubules tagged with α-Tubulin. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 20 μm, **(b)** the selected crop; Scalebar: 5 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

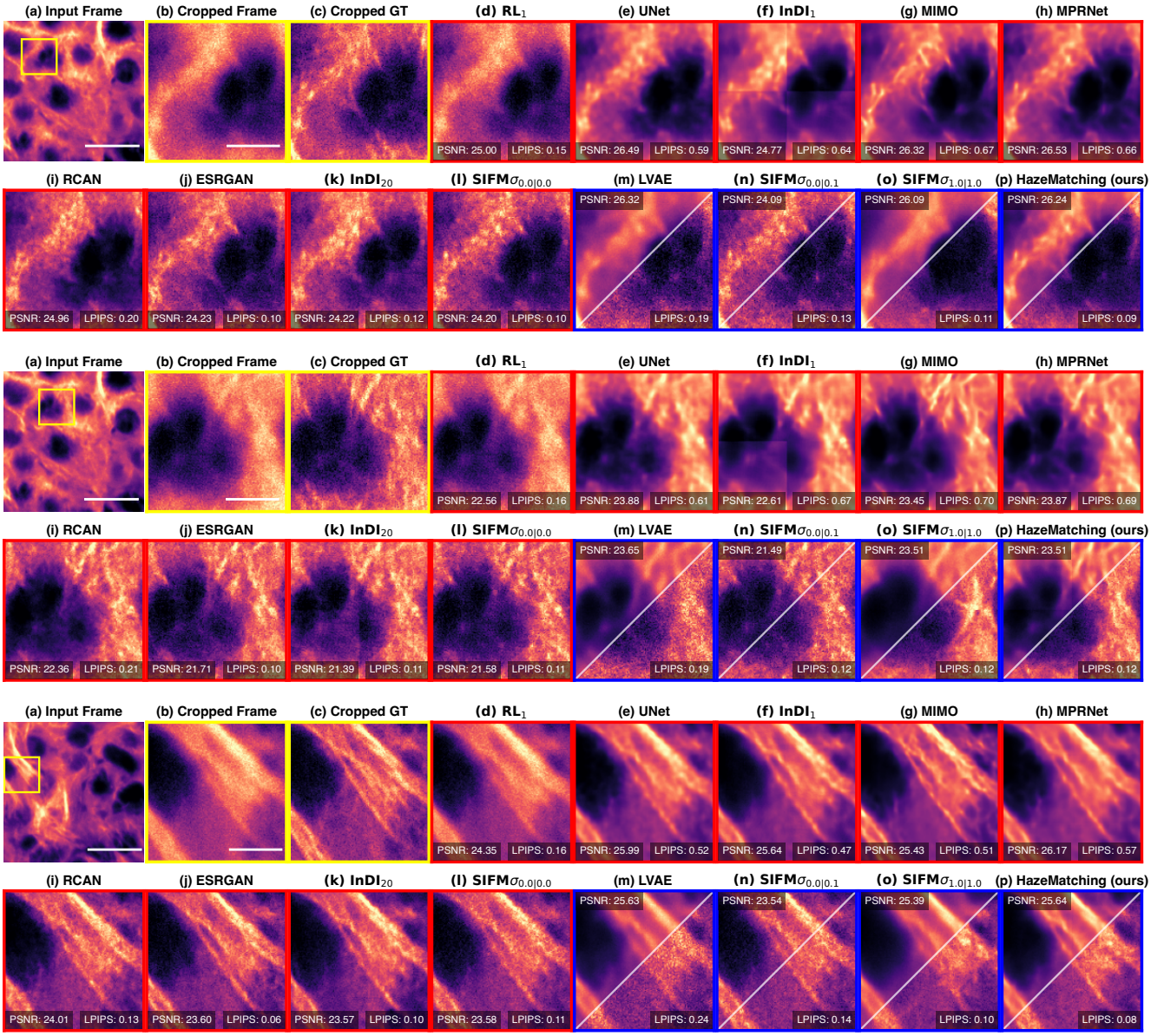


Figure S21. **Qualitative results on Microtubule Data:** Here we present three examples showing Microtubules tagged with α-Tubulin. **(a)** the full input and a selected 128×128 crop (yellow box); Scalebar: 20 μm, **(b)** the selected crop; Scalebar: 5 μm, **(c)** non-hazy ground truth, **(d–o)** predictions by all baseline methods (see Section 4.1), and **(p)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

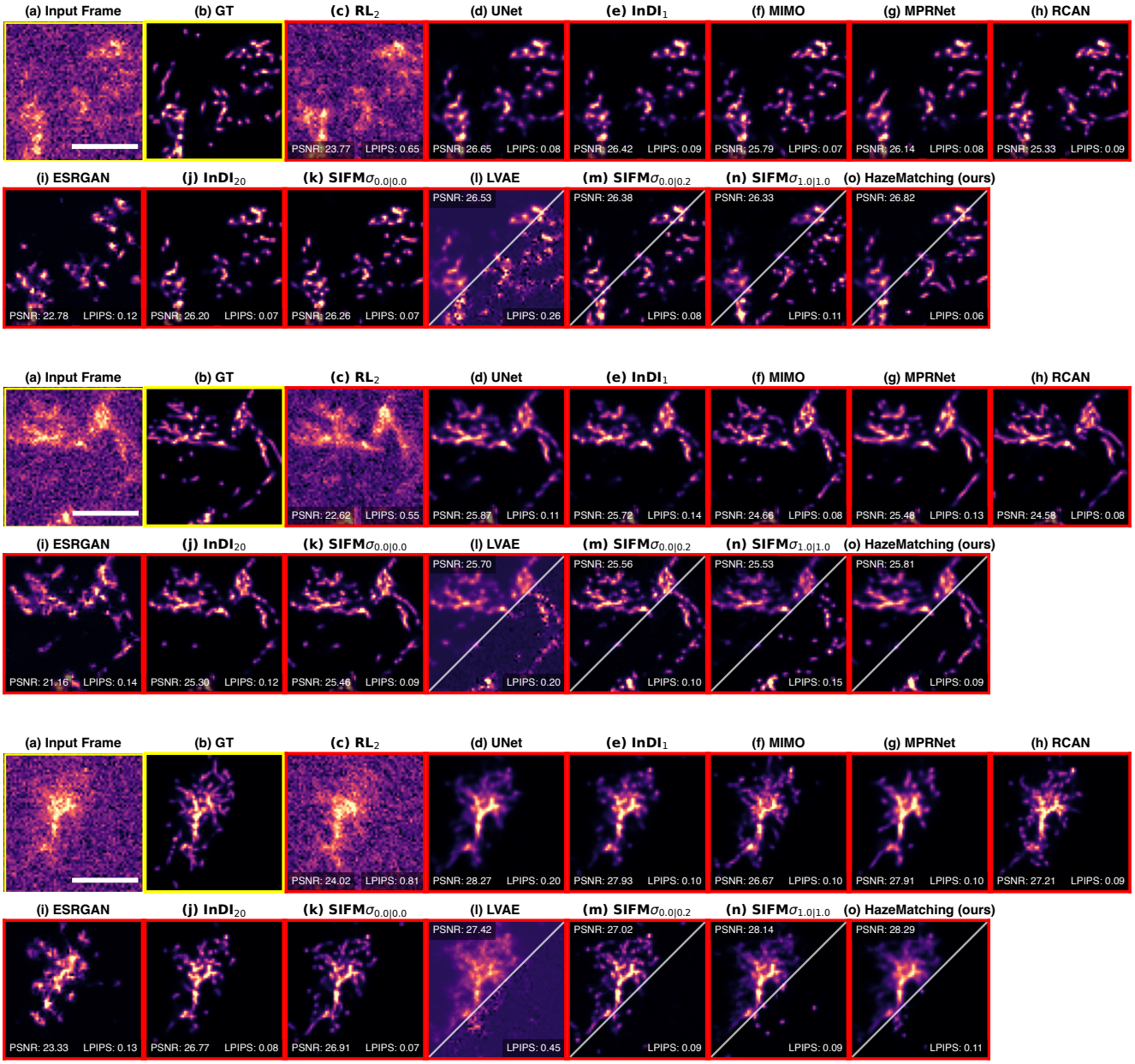


Figure S22. **Qualitative results on Neuron Data:** Here we present three examples showing Neuron data. **(a)** the full 64×64 input frame (yellow box); Scalebar: 5 μm, **(b)** non-hazy ground truth, **(c–n)** predictions by all baseline methods (see Section 4.1), and **(0)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

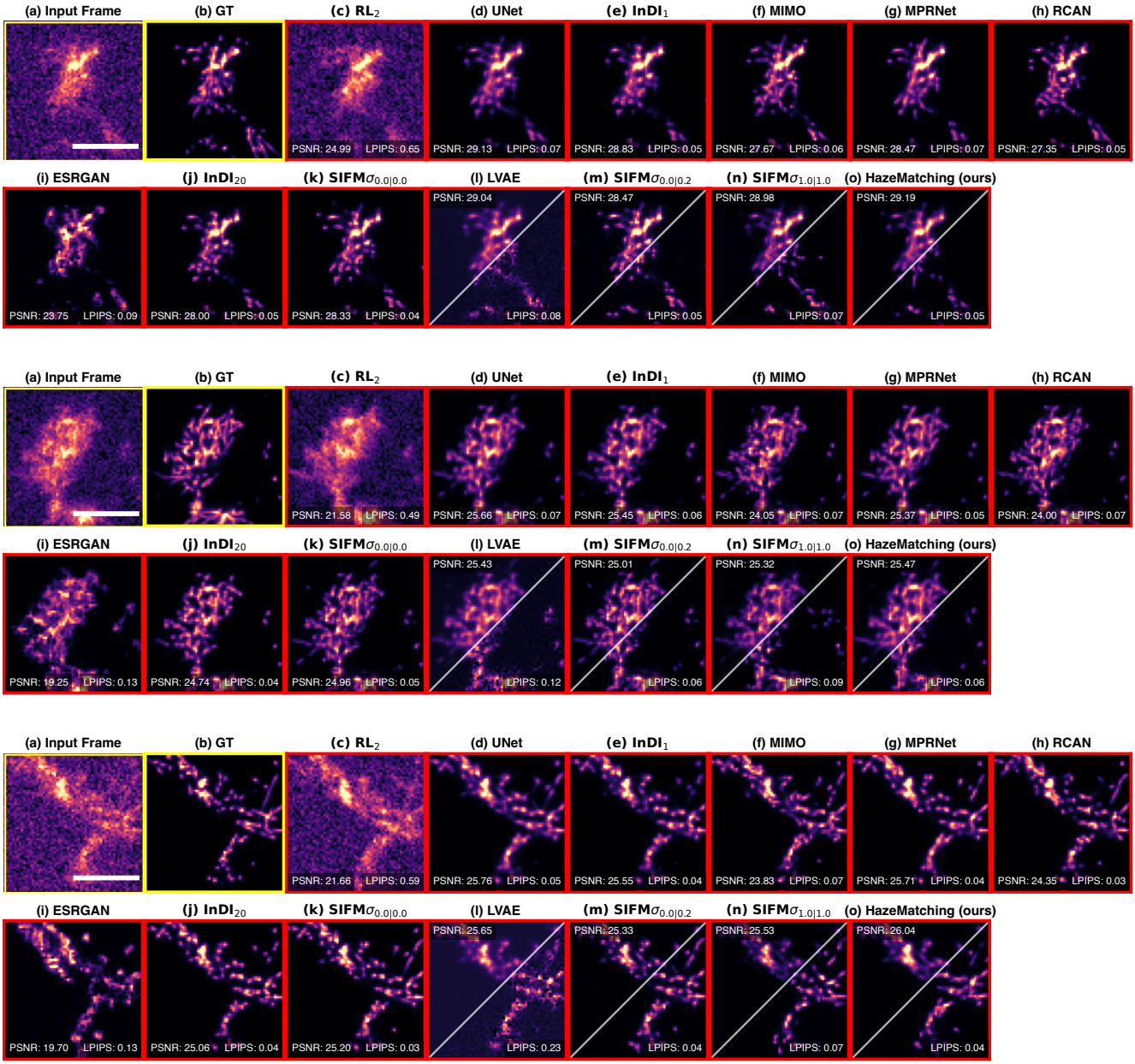


Figure S23. **Qualitative results on Neuron Data:** Here we present three examples showing Neuron data. **(a)** the full 64×64 input frame (yellow box); Scalebar: 5 μm, **(b)** non-hazy ground truth, **(c–n)** predictions by all baseline methods (see Section 4.1), and **(o)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

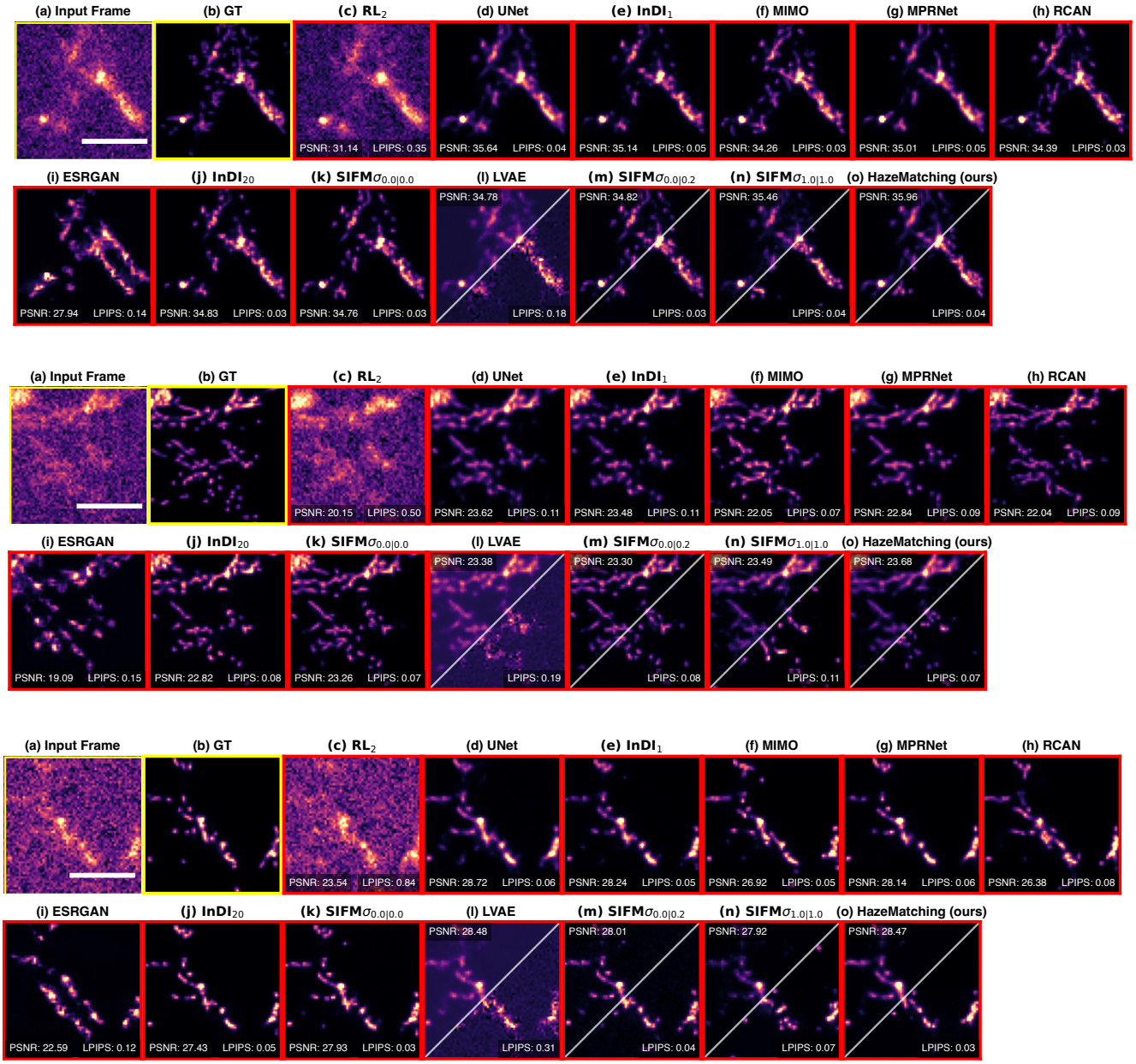


Figure S24. **Qualitative results on Neuron Data:** Here we present three examples showing Neuron data. **(a)** the full 64×64 input frame (yellow box); Scalebar: 5 μm, **(b)** non-hazy ground truth, **(c–n)** predictions by all baseline methods (see Section 4.1), and **(o)** results obtained with HAZEMATCHING. Results with red borders are predictions by point-predictors, while methods with blue borders are results by generative posterior models (see also Figure 2 and main text). Note that HAZEMATCHING consistently produces sharper and more perceptually aligned predictions (lower LPIPS) compared to both deterministic and posterior-based baselines, while maintaining comparative fidelity (PSNR). For the point-prediction methods, PSNR and LPIPS are computed on the cropped region shown in yellow. For posterior models (blue borders), we plot the MMSE estimate in the upper triangle with its PSNR and one posterior sample in the lower-triangle with its LPIPS score.

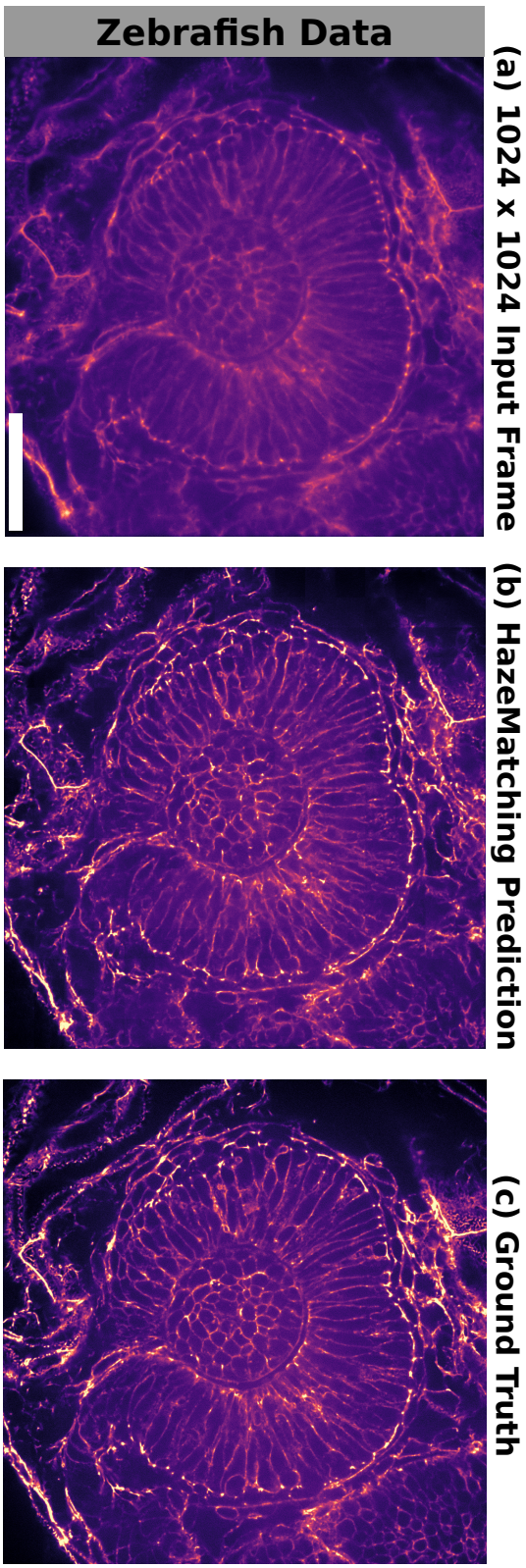


Figure S25. **Qualitative results on Zebrafish Data:** Here we show one full-frame image from the test data. **(a)** the full 1024×1024 input frame; Scalebar: 50 μm, **(b)** Corresponding prediction by HAZEMATCHING, **(c)** the Ground Truth, Best viewed on a screen with 5× zoom.

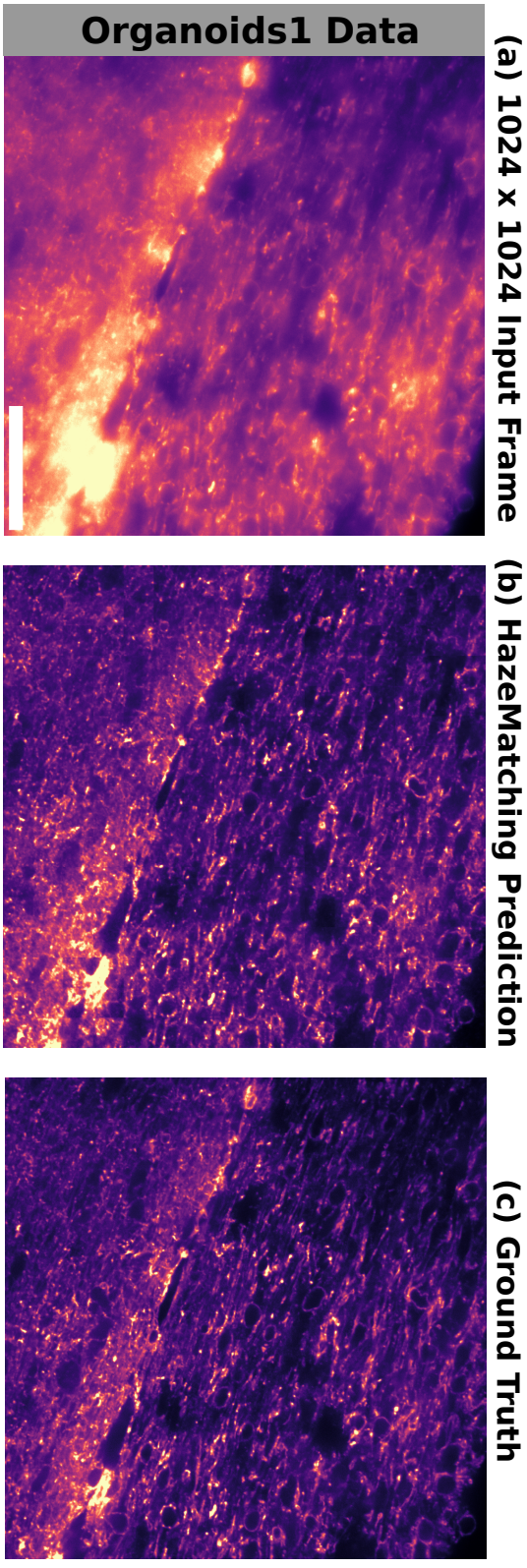


Figure S26. **Qualitative results on Organoids1 Data:** Here we show one full-frame image from the test data. **(a)** the full 1024×1024 input frame; Scalebar: 50 μm, **(b)** Corresponding prediction by HAZEMATCHING, **(c)** the Ground Truth, Best viewed on a screen with 5× zoom.

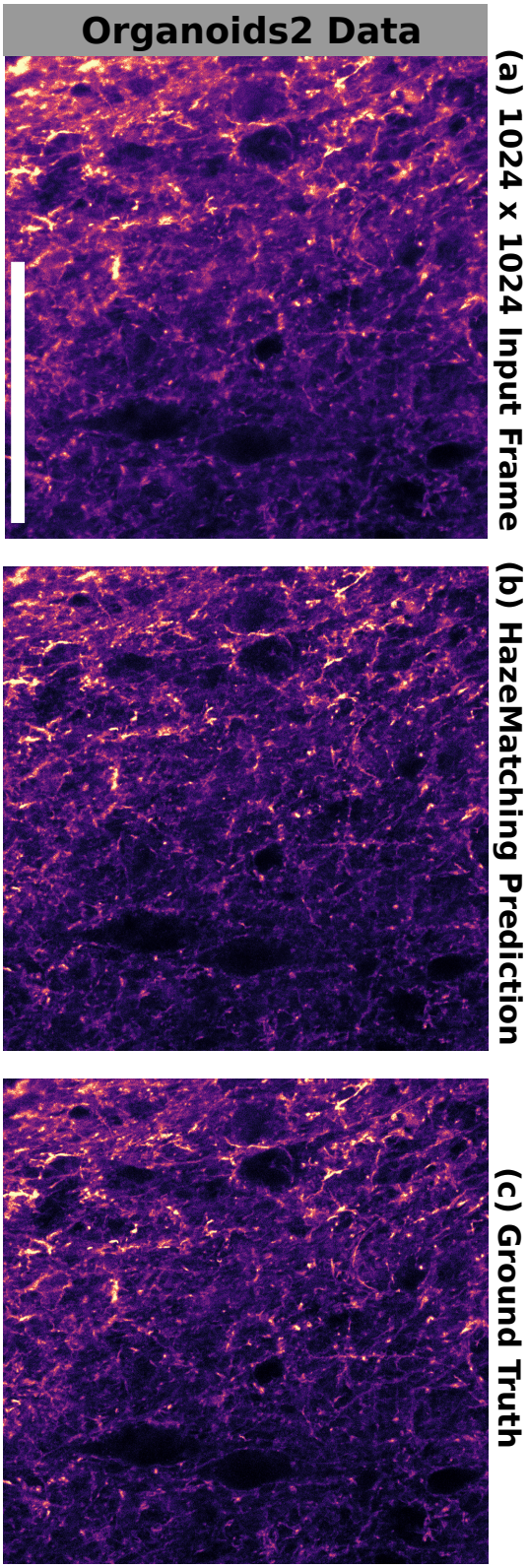


Figure S27. **Qualitative results on Organoids2 Data:** Here we show one full-frame image from the test data. **(a)** the full 1024×1024 input frame; Scalebar: 50 μm, **(b)** Corresponding prediction by HAZEMATCHING, **(c)** the Ground Truth, Best viewed on a screen with 5× zoom.

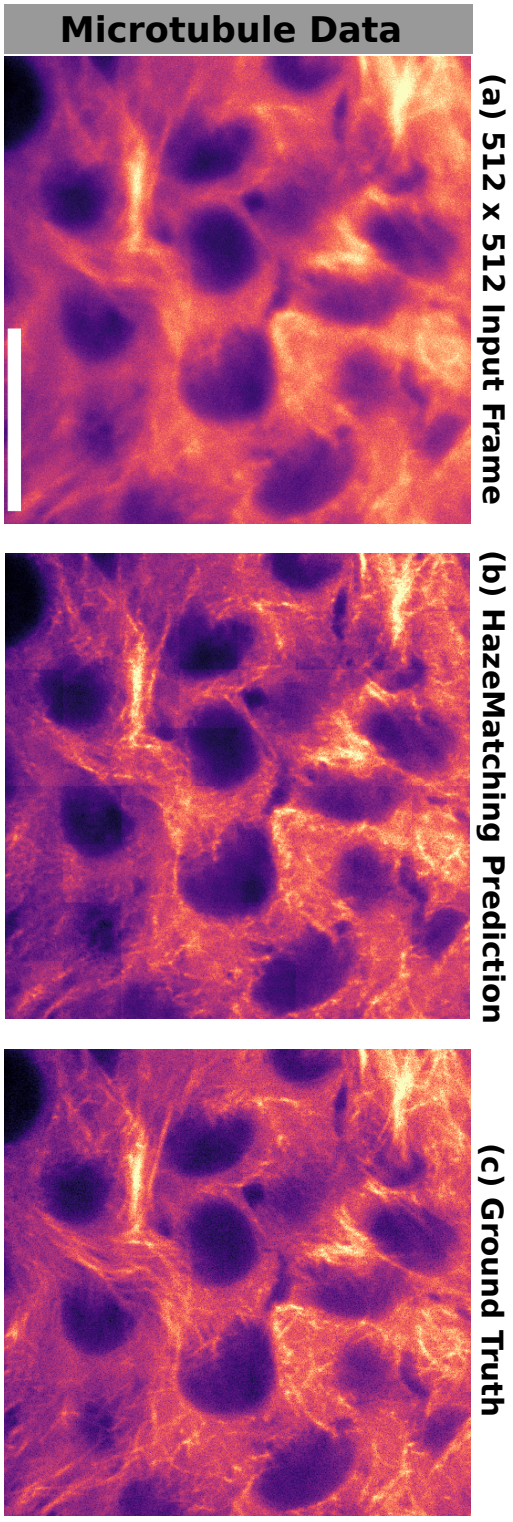


Figure S28. **Qualitative results on Organoids2 Data:** Here we show one full-frame image from the test data. **(a)** the full 1024×1024 input frame; Scalebar: 20 μm, **(b)** Corresponding prediction by HAZEMATCHING, **(c)** the Ground Truth, Best viewed on a screen with 5× zoom.