Wave2Plane: Learning to Recover Planar Scenes Under Turbulent Water

Nippun Sabharwal, Advisor: Narendra Ahuja

Electrical and Computer Engineering, Grainger College of Engineering,University of Illinois Urbana-Champaign

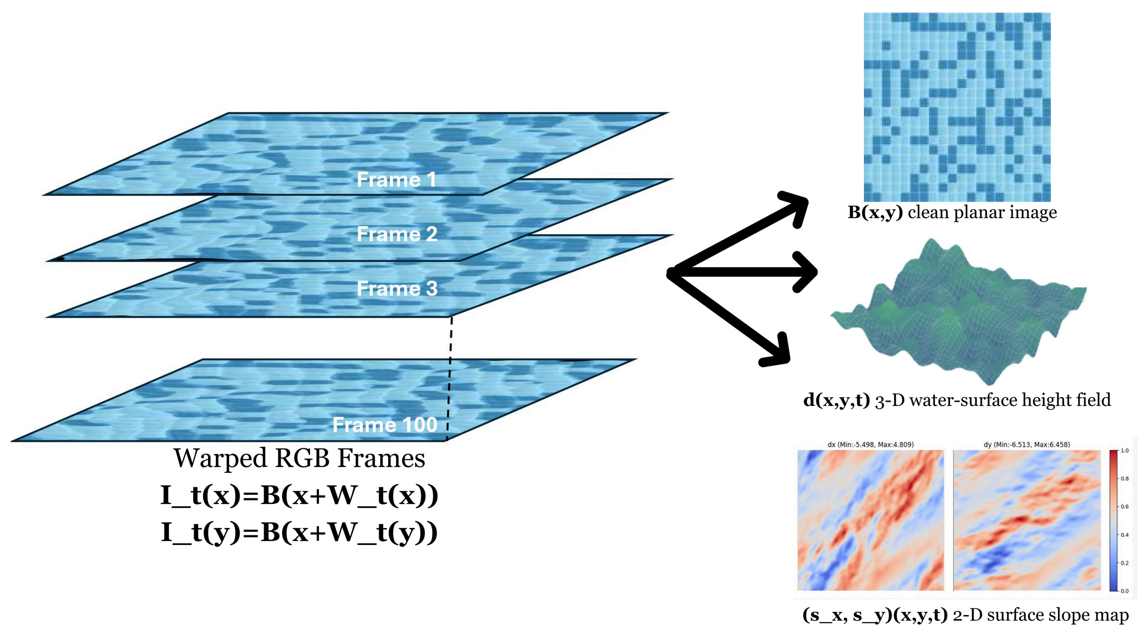
INTRODUCTION

We present an unsupervised, physics-informed framework for recovering planar scenes from videos captured through turbulent water refraction, without clean ground-truth supervision. Our method couples a differentiable optical forward model, built from wave dynamics and Snell-law ray tracing, with learned spatiotemporal inference. We further introduce interpretable slope-estimation modules, including trajectory-statistics models and compact 3D CNNs, to identify low-distortion regions and guide cross-frame pixel stitching.

Problem Statement

Given a video of a static underwater texture viewed through a moving water surface, recover a cleaner image of the bottom. The distortion is caused by time-varying refraction at the air-water interface: each frame warps the clean image differently, and the least-distorted frame changes from pixel to pixel.

For each pixel, which video frame is closest to the undistorted view, and how can a learned model infer that choice from local refractive evidence?



Input: video of a static planar scene viewed through turbulent water

Recover: three jointly-constrained physical quantities from that single video

Wave Simulator

① Ocean (Tessendorf)

Phillips-spectrum gravity waves

$$P(\mathbf{k}) = A\frac{e^{-1/(kL)^2}}{k^4}(\hat{\mathbf{k}}\cdot\hat{\mathbf{w}})^2 e^{-k^2\ell^2}$$

Deep-water dispersion: $\omega(\mathbf{k}) = \sqrt{gk}$

IFFT gives time-varying height. Captures long-range directional swells.

② Ripple (radial interference)

3 point sources with radial wavefronts:

$$h_{\text{ripple}} = \sum_{j=1}^{3} a_j \sin(k_j r_j - \omega_j t + \phi_j)$$

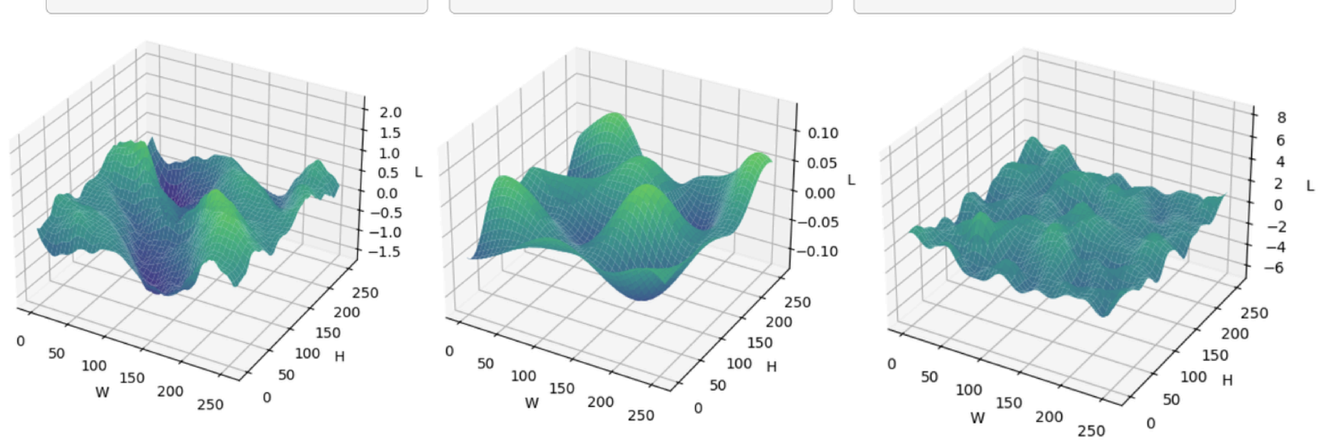
Creates localized, directional interference patterns that broaden slope statistics.

③ Tian (linear wave PDE)

Discretized wave equation:

$$h^{t+1} = 2h^t - h^{t-1} + c^2\nabla^2 h^t$$

Propagates disturbances numerically on the pixel grid. Coherent propagation, easy to control, periodic boundaries.



Simulator: From wave height to warped image

1. **Depth field** $d = h + d_0$ (bias keeps $d > 0$).

2. **Slope (spectral)**

$$s_x = \mathcal{F}^{-1}(ik_x\mathcal{F}d),\ s_y = \mathcal{F}^{-1}(ik_y\mathcal{F}d)$$

FFT differentiation no stencil bias, consistent with the spectral wave model.

3. **Surface normal**

$$\mathbf{n} = \frac{(-s_x, -s_y, 1)}{\|(-s_x,-s_y,1)\|}$$

4. **Snell's law** (exact vector form on $\mathbf{n}$, $\eta = n_{\text{air}}/n_{\text{water}}$)

Small-slope limit used everywhere downstream:

$$W_t(p) \approx -d_{\text{eff}}(1-\eta)\, s_t(p)$$

5. **Bilinear sampling**

$$I_t(x,y) = \text{bilinear}\big(B,\ x+u_t,\ y+v_t\big)$$

Sub-pixel bilinear is the only approximation no sensor noise or blur added.

METHOD

Learning Slopes to Unwarp Turbulence

Why Slopes? Local surface slope is the information-preserving intermediate for undoing refractive distortion.

Zero-slope means zero warp: W≈−d_eff(1−η) s ⇒ s=0 ⇒ **W = 0**

A pixel observed while the surface is flat above it suffers no refractive displacement it is the cleanest possible observation of **B(x, y)**.

Vector nature matters: Scalar tilt θ=arctan(||s||). Two pixels with the same θ can warp in opposite directions. A slope map without direction cannot tell apart a crest (warp up), a trough (warp down), or a saddle so scalar regression alone cannot support restoration.

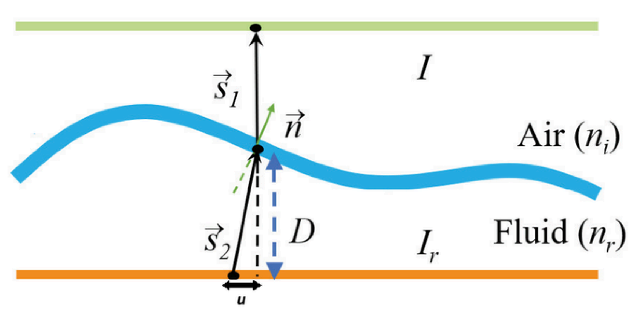
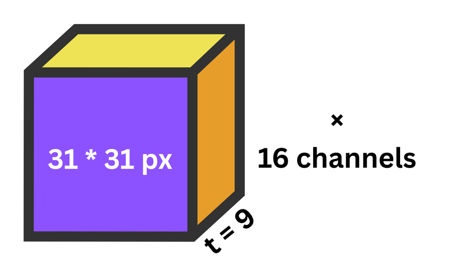


Image credit: Dynamic Fluid Surface Reconstruction Using Deep Neural Network Simron Thapa, Nianyi Li, Jinwei Ye.

Slope Regressor with Local Context

The simulator shows that refractive distortion is a local problem: a pixel's appearance is determined mainly by the slope and warp in a small neighborhood around that pixel, not by the entire image at once. It also shows that the useful temporal information is short-range: nearby frames remain correlated, but that correlation decays quickly as the wave phase moves on.

For that reason, the model is built around an **ordered local cuboid** rather than a single frame or a full-video input. We use a 31 *31 spatial window to cover almost all local refractive displacements, and a 9-frame temporal window to capture the short-time wave dynamics around the center frame. This gives the network the local context it needs without asking it to solve an unnecessary global problem inside the regressor itself.



$1\text{–}3: R_Q - R_P,\ G_Q - G_P,\ B_Q - B_P$

$4\text{–}6: R_Q, G_Q, B_Q$

$7\text{–}8: \text{mag}_Q, \text{ang}_Q$

$9\text{–}10: \text{mag}_P, \text{ang}_P$

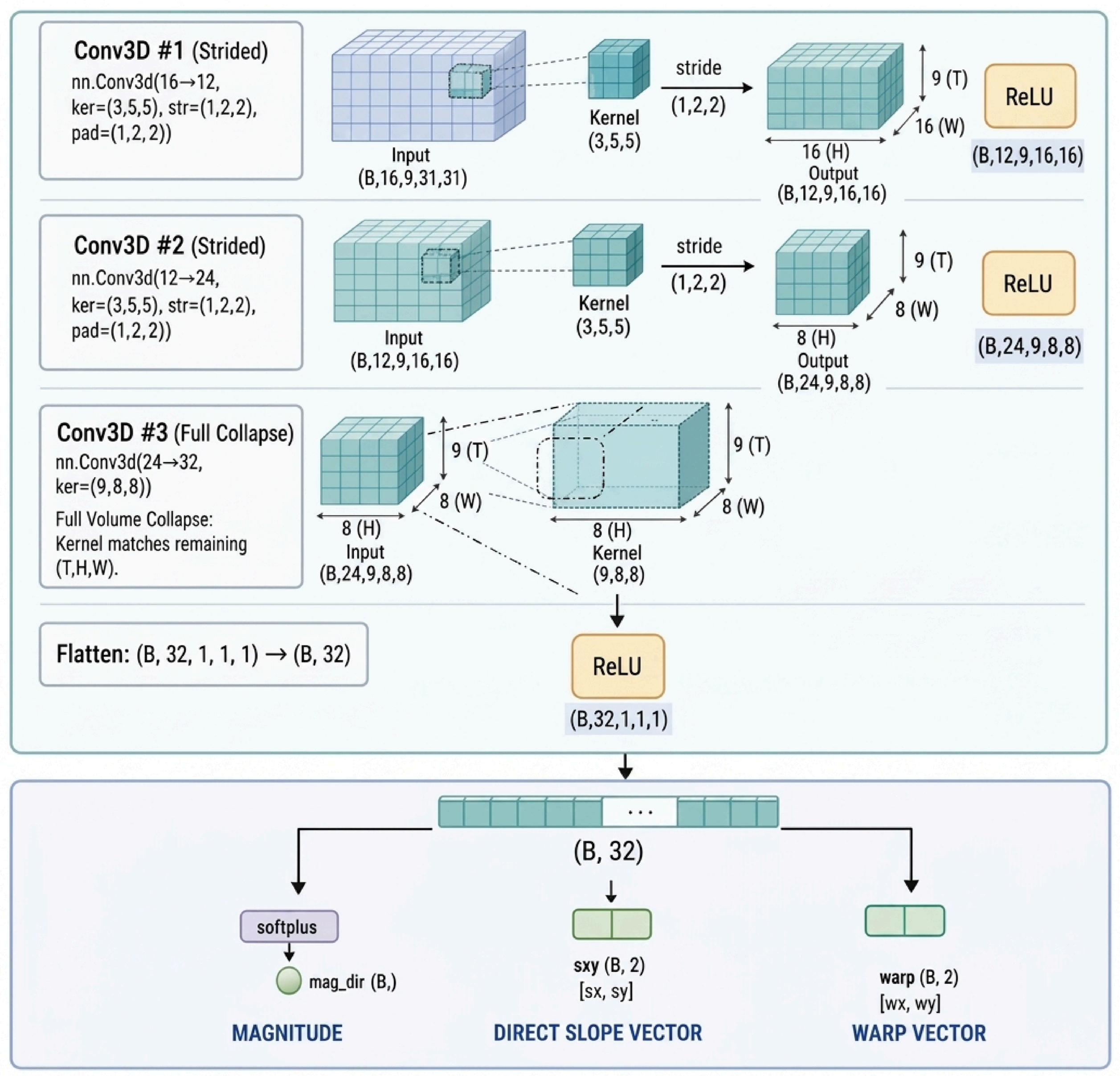
$11\text{–}12: \text{harris}_P, \text{harris}_Q$

$13: \text{temporal\_diff}_t(x,y) \approx \frac{I_{t+1}(x,y) - I_{t-1}(x,y)}{2}$

$14: \text{temporal\_accel}_t(x,y) \approx I_{t+1}(x,y) - 2I_t(x,y) + I_{t-1}(x,y)$

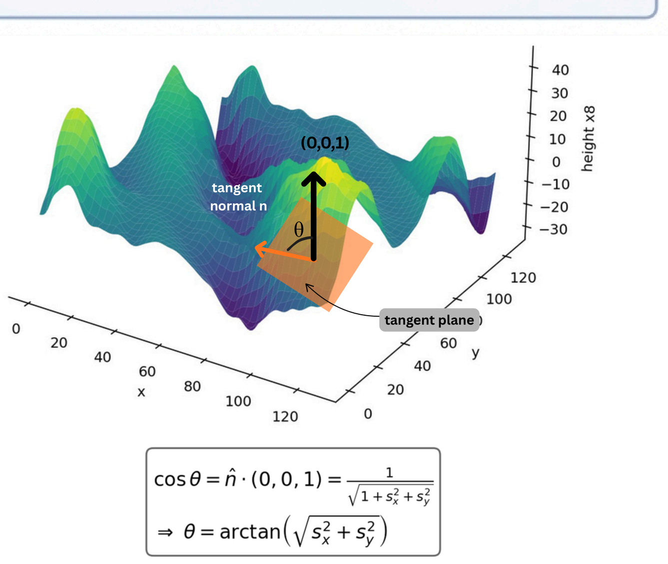
$15: \text{temporal\_var}_t(x,y) = \text{Var}_{\tau\in[t-\Delta t,\ t+\Delta t]}\, I_\tau(x,y)$

$16: \text{temporal\_grad\_mag}_t(x,y)$



Three heads, each with a reason:

- $\hat{m}$ (**scalar**) selection-valid target. Oracle equivalence $\arg\min\theta \equiv \arg\min\|W\|$ lets a single score pick the flattest frame.
- $\hat{s} = (\hat{s}_x, \hat{s}_y)$ (**vector**) scalar θ is direction-blind (crest/trough/saddle collapse); vector head restores direction for any warp-aware stage.
- $\hat{W} = (\hat{u}, \hat{v})$ (**warp**) Ch.3 result: under the fixed Snell map, warp is the *information-preserving intermediate* for slope. Predicting it keeps the embedding aligned with the observable.



Training Objective

$$\mathcal{L}_{7b} = \mathcal{L}_{\text{rank}} + \lambda_m\mathcal{L}_{\text{mag}} + \lambda_s\mathcal{L}_{\text{slope}} + \lambda_w\mathcal{L}_{\text{warp}} + \lambda_p\mathcal{L}_{\text{slope,phys}} + \lambda_c\mathcal{L}_{\text{cons}}$$

Term	Role
$\mathcal{L}_{\text{rank}}$	listwise KL over K=8 times trains the selector directly
$\mathcal{L}_{\text{mag}}$	SmoothL1 $\hat{m}$ vs $\lvert s\rvert$ calibrates scalar output
$\mathcal{L}_{\text{slope}}$	SmoothL1 $\hat{s}$ vs s direction supervision
$\mathcal{L}_{\text{warp}}$	SmoothL1 $\hat{W}$ vs GT direct geometry supervision
$\mathcal{L}_{\text{slope,phys}}$	$\lvert\hat{s}_{\text{phys}} - s\rvert$, $\hat{s}_{\text{phys}} = -\hat{W}/(\hat{d}_r(1-\eta))$ ties $\hat{W}$ to slope *through Snell*
$\mathcal{L}_{\text{cons}}$	$\lvert\hat{s} - \hat{s}_{\text{phys}}\rvert$ Snell coherence / integrability prior

Dense Inference and Reconstruction

At inference time, the trained model is run densely over all valid pixels and all candidate frames. For each pixel-frame pair it predicts a scalar flatness score together with auxiliary geometry. The learned reconstruction is then formed by:

- scoring each candidate frame locally,
- selecting or softly weighting the flattest frames per pixel,
- stitching those local choices into one reconstructed image.

This keeps the deployed learned system simple: the output image is still formed by frame selection, but the selection scores now come from a representation that is regularized by explicit refraction physics.

Global Optimizer

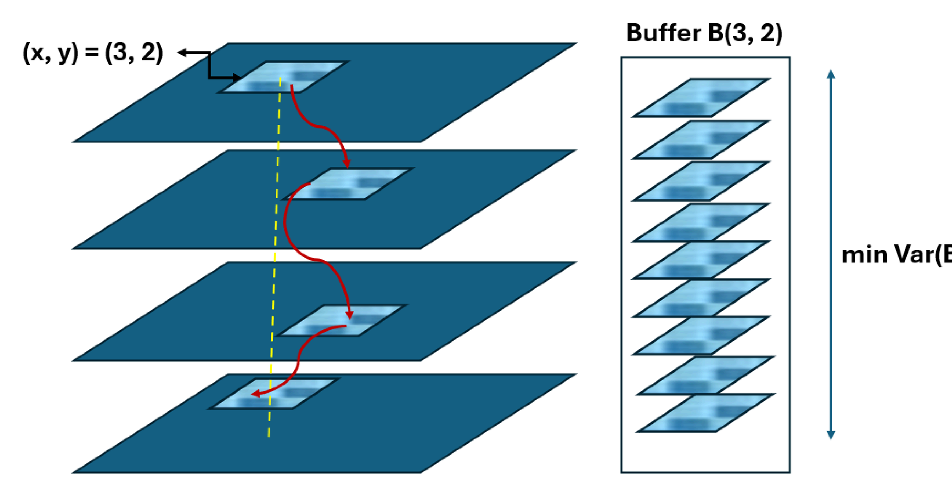
Motivation: The CNN predicts slope per pixel; B and the surface are global. At test time we optimize the slope field so pixels that should agree actually do.

Forward splat. Splat every pixel p at time t to its predicted clean coordinate q=p+ W_t(p), accumulating buffer[q]={RGBi} across all frames.

If $\hat{W}$ is **geometrically correct**, every RGB in $\text{buffer}[q]$ observes the *same* clean texel:

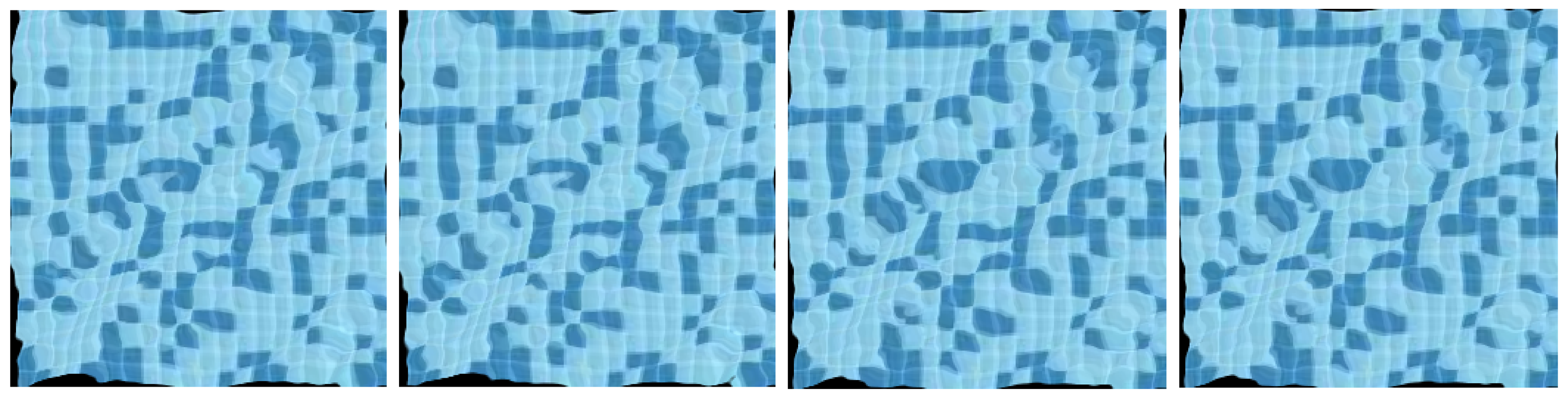
$$\mathcal{L}_{\text{buffer}} = \text{mean}_q\ \text{Var}_{\text{RGB}}\big[\text{buffer}(q)\big]$$

Minimizing buffer variance makes slope/warp consistent across **all** frames simultaneously **no clean image needed**.

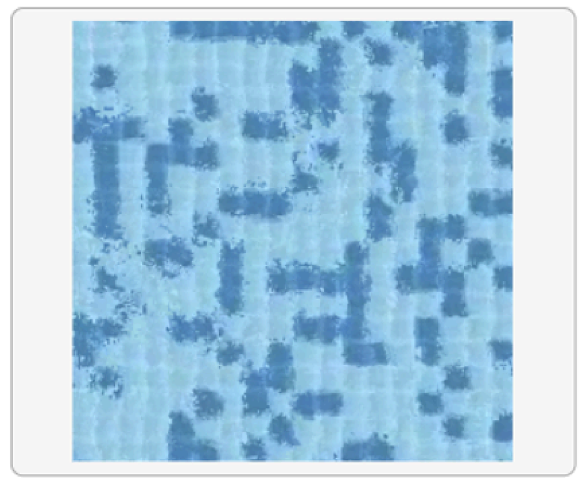


RESULTS

Input Video:

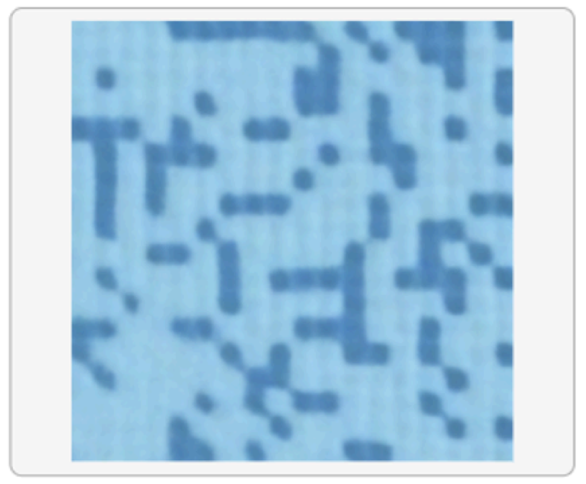


Model Output:



physics regressor (learned)

24.67 dB / SSIM 0.612



physics regressor + physics optimizer

26.67 dB / SSIM 0.702