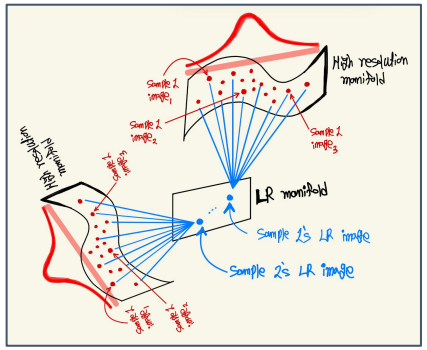
Deep Learning based Optical Image Super-Resolution via Generative Diffusion Models for Layerwise in-situ LPBF Monitoring (2024)

❑ Literature Review

❖ Overview (Ch.1)

➢ LPBF enables complex geometries & rapid prototyping, but critical limitation is potential defects across multiple scales

- Microscale: 1) Insufficient melting → large, irregularly shaped pores
  2) Excessive heat → vaporization cavities
- Macroscale: 1) Irregular events during a build process → Part deformation, super-elevation
  2) Irregular events during a powder re-coating process → non-uniform powder bed & geometric defects



➢ Defects must be detected before build completion: in-situ

- X-ray is intrusive & expensive, so many layer-wise "optical monitoring techniques" are being used:
  - Retrofit-friendly, non-intrusive, build-level field of view (FOV)
  - Detects in-plane & out-of-plane defects, and powder contamination
- Key trade-offs are about the quality:
  - High-resolution imaging by X-ray → accurate for small-scale features like roughness and morphology, but cause storage constraints & real-time monitoring constraints
  - Low-cost cameras are scalable but lack sufficient resolution

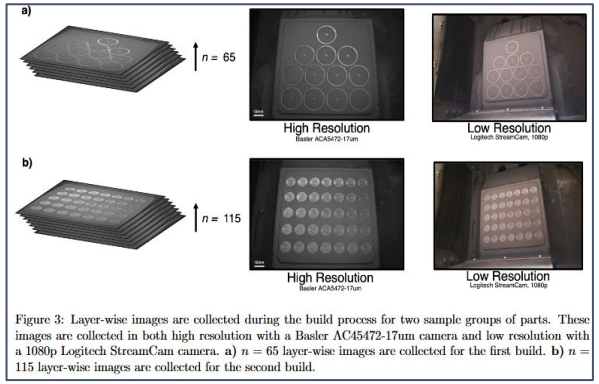


Figure 3: Layer-wise images are collected during the build process for two sample groups of parts. These images are collected in both high resolution with a Basler AC45472-17um camera and low resolution with a 1080p Logitech StreamCam camera. **a)** $n = 65$ layer-wise images are collected for the first build. **b)** $n = 115$ layer-wise images are collected for the second build.

➢ How to get HR images with low cost and time?

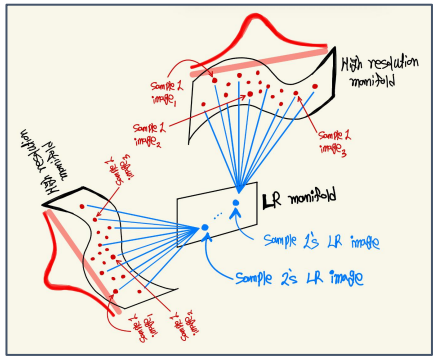
- Assumption: 1) A manifold of LPBF images include LR and HR sub-manifolds
  2) LR is a result of dimension reduction of HR
- Goal: Link low-cost & LR monitoring to HR information via Conditional diffusion model: "**learn distribution of p(HR | |LR)**"
  - Preserve high-frequency details & variability
  - Release the limitations by deterministic Super-resolution (SR) methods: PCA, FPCA, Very deep CNN and DenseNet
  - Challenge: high inference cost → **Latent Diffusion**: Perform diffusion in latent space & Enables efficient, real-time, in-situ monitoring

❑ Literature Review

❖ Methods: How to learn? (Ch.2)

➢ Learn a probabilistic super-resolution mapping from LR in-situ monitoring images to HR optical images in LPBF: **p(HR | |LR)**

- Acquire layer-wise HR optical images and LR webcam images during the build
  - Processing HR images as well as implementing reverse process of diffusion models take a while → How to tackle this?
- Generate HR samples conditioned on LR inputs using conditional latent diffusion, not just a classical DDPM
- Evaluate performance via distributional comparisons, 3D morphology, and surface roughness

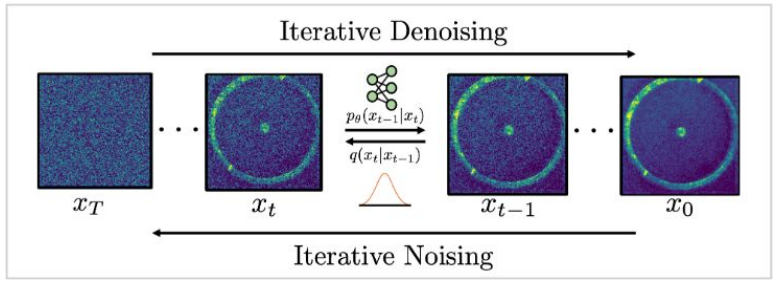


➢ Denoising Diffusion Probabilistic Model (DDPM)

- Forward process
  - Given a clean target sample $x_0 \sim p_{data}(x)$:

    $q(x_t \mid x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\, x_0,\ (1-\bar{\alpha}_t)I\right)$, or equivalently

    $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I)$



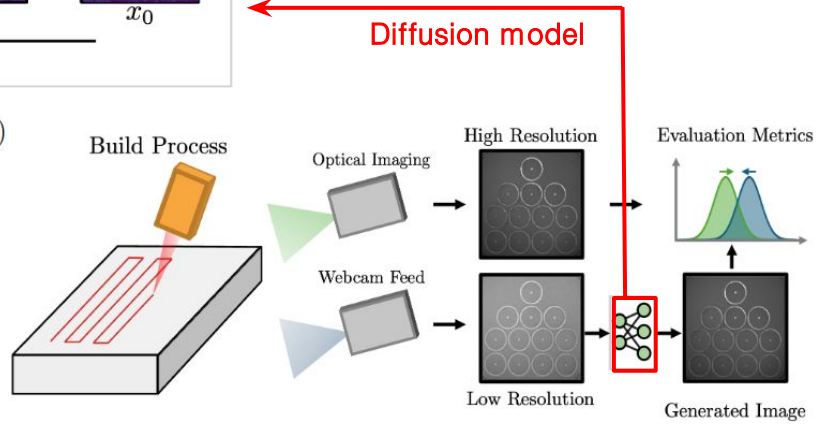
Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

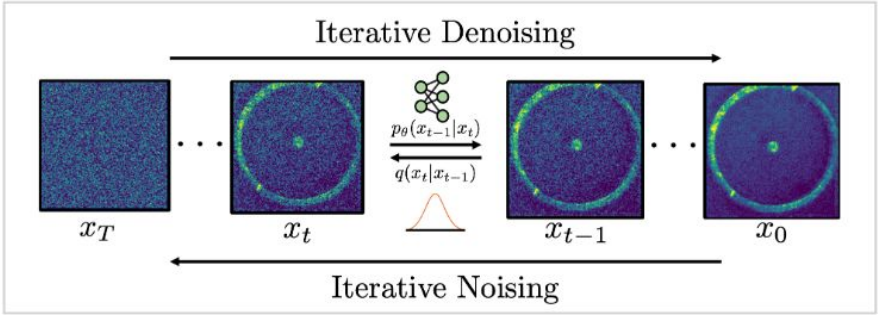
- Reverse process
  - A **U-Net architecture** parameterizes the denoising network: $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$
  - The denoising network is trained to predict the injected noise: $\mathcal{L}_{\text{diff}} = \mathbb{E}_{x_0,t,\varepsilon}\left[\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\right]$
- Sampling process
  - Sampling starts from isotropic Gaussian noise: $x_T \sim \mathcal{N}(0, I)$
  - The model iteratively removes noise: $x_{t-1} \leftarrow \text{Denoise}(x_t,\ \varepsilon_\theta(x_t, t)), \quad t = T, \dots, 1$
- Conditional Diffusion for Super-Resolution
  - Super-resolution is formulated as conditional generation:
    - Transform LR images into embeddings & Concat them with corrupted HR x_t in training reverse process
    - The denoising network learns: $\varepsilon_\theta(x_t, t \mid y), \quad y = x_{LR}$
    - This way, the diffusion models learn the relationship between LR and HR: $P(x_{HR} \mid x_{LR})$
- Still exist potential problem: **memory** and **time** constraints involved in dealing with HR images



❑ Literature Review

❖ Methods: How to learn? (Ch.2)

➢ Latent Diffusion for efficient inference



- Vanila Conditional Diffusion learns distribution of **P(HR | LR)**
  - Images of LR and HR themselves are high-dimensional data
    → Take a lot of time to process them while training the model
  - **Pixel-space diffusion** is computationally expensive for real-time monitoring
- Idea: Transform pixel-space into 'latent space' & Implement diffusion in a learned latent space!
  - **Step 1.** Latent encoding
    - Two autoencoders are trained: $z_{HR} = \mathcal{E}_{HR}(x_{HR}),\ z_{LR} = \mathcal{E}_{LR}(x_{LR})$
    - Now, HR images are downscaled to match Low resolution
      → Latent representations share identical spatial dimensions
  - **Step 2.** Conditional Latent Diffusion (Rombach et al.): SR model
    - Noise is added to HR latent (Z_{HR}) only during forward process: x_t
    - The LR latent (Z_{LR}) is fixed and time-invariant
    - At each diffusion timestep: $\text{Input}_t = \text{concat}(x_t,\ z_{LR})$
    - The network predicts the noise: $\hat{\varepsilon} = \varepsilon_\theta(x_t, t \mid z_{LR})$
  - **Step 3.** Z_{HR} is generated after iterative denoising
    & Decode it back to the original image: $x_0 \approx z_{HR},\ \hat{x}_{HR} = \mathcal{D}_{HR}(z_{HR})$
- Advantages: Reduced memory and inference cost
- In our case,
  - Encode X_{obs} and X_{tar} into latent space by a neural network
  - Train conditional diffusion s.t: $P(Z_{tar} \mid Z_{obs}, s)$
  - Generate multiple Z_{tar} by sampling from the learned distribution, $P(Z_{tar} \mid Z_{obs}, s)$
  - Decode Z_{tar} into X_{tar}

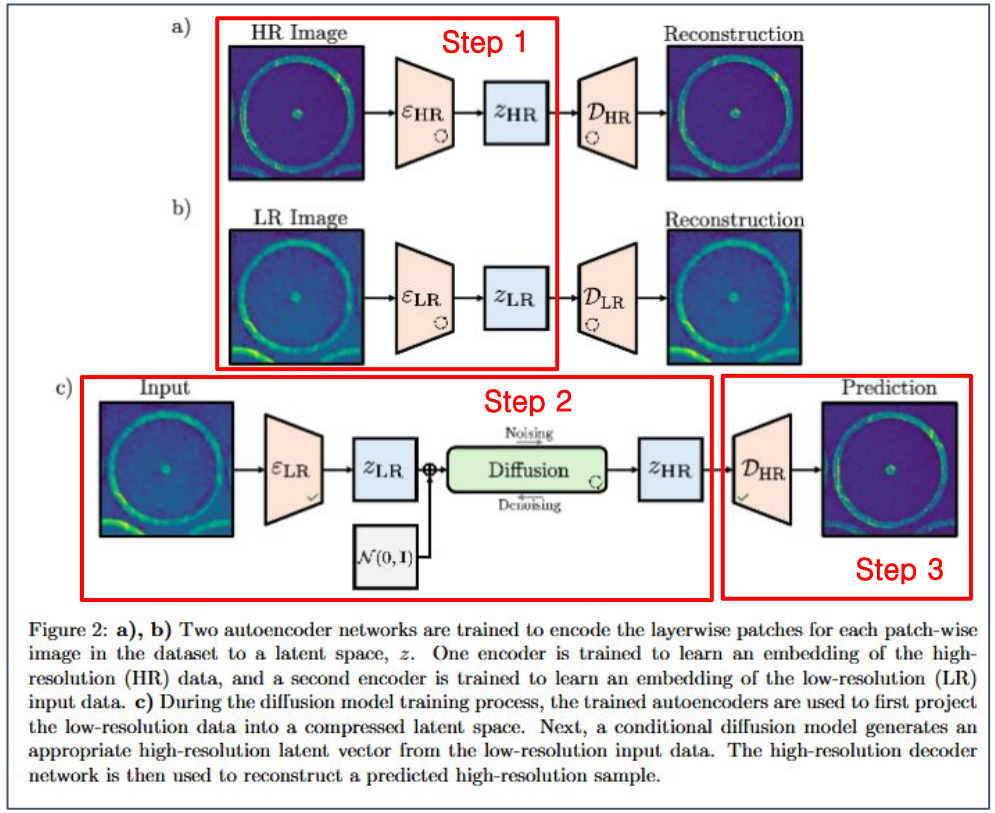


Figure 2: **a), b)** Two autoencoder networks are trained to encode the layerwise patches for each patch-wise image in the dataset to a latent space, z. One encoder is trained to learn an embedding of the high-resolution (HR) data, and a second encoder is trained to learn an embedding of the low-resolution (LR) input data. **c)** During the diffusion model training process, the trained autoencoders are used to first project the low-resolution data into a compressed latent space. Next, a conditional diffusion model generates an appropriate high-resolution latent vector from the low-resolution input data. The high-resolution decoder network is then used to reconstruct a predicted high-resolution sample.

❑ Literature Review

❖ Results (Ch.3)

➢ Model Performance Evaluation Metrics

- Compare the performance of both **latent diffusion** and **pixel-space diffusion** in constructing HR features
  - All hyperparameters are held constant to isolate computational effects
- Performance Metrics
  - **Pixel-Level Reconstruction accuracy Metrics**

    a) *Mean Absolute Error (MAE)*

    - Measures average absolute difference in pixel intensities
    - Lower values indicate better reconstruction accuracy

    $$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i|$$

    b) *Peak Signal-to-Noise Ratio (PSNR)*

    - Quantifies the ratio between the preserved signal and reconstruction noise
    - MAX_I: maximum possible pixel intensity
    - Higher PSNR indicates higher reconstruction fidelity

    $$\mathrm{PSNR} = 10\log_{10}\left(\frac{MAX_I^2}{\mathrm{MSE}}\right)$$

    c) *Structural Similarity Index Measure (SSIM)*

    - Measures perceived structural similarity between images by accounting for local pixel dependencies
    - Higher SSIM indicates higher reconstruction fidelity

    $$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

  - **Texture and Distributional Metrics: captures non-linear dependencies in image texture across frequency bands!**

    a) *Normalized Covariance Distance (CVD)*

    - Measures the discrepancy between texture statistics of generated and HR images
    - Lower values indicate closer agreement in texture statistics
    - Wavelet Covariance Operator:

    $$\mathrm{CVD} = \frac{\|\tilde{K}_{R_x} - \tilde{K}_{\tilde{R}_x}\|}{\|\tilde{K}_{\tilde{R}_x}\|}$$

    $$\tilde{K}_{R_x} = \frac{1}{|G|}\sum_{g\in G}\left(R_v(g(x)) - \tilde{M}_R(v)\right)\left(R_{v'}(g(x)) - \tilde{M}_R(v')\right)$$

    - $R_v, R_{v'}$: wavelet coefficients at different scales, angles, or spatial locations
    - G: set of translation operators to enforce translational invariance
    - $g(x)$: translated version of image x
  - Pixel-level metrics (MAE, PSNR, SSIM) quantify reconstruction accuracy and structural similarity, while wavelet-based covariance distance captures high-frequency texture fidelity critical for evaluating generative super-resolution models!

❑ Literature Review

❖ Results (Ch.3)

➢ Patch-Based Training Setup

- Motivation
  - Full build-plate images are memory-intensive
  - Adopt a **patch-based strategy** to enable efficient, scalable super-resolution first
  - Afterward, upscale the entire build plate piecewise, while preserving fine-scale texture
- Training Configuration
  - Patch size: 64×64 (Patches are extracted from each layer-wise build plate image)
  - 115 patches per layer are randomly sampled
  - Data split: 8:2 for training and test data
- Model Training Details
  - **Autoencoders:** Trained for 100 epochs & Learning rate: 4.6×10^{-5}
  - **Latent diffusion model:** Trained for 300 epochs & Learning rate: 1.0×10^{-5}

➢ Patch-Based Results

- Qualitative results
  - LR patches: [1] Capture large-scale part geometry & [2] Fail to resolve powder bed texture
  - SR patches by Diffusion: [1] Reconstruct fine-scale powder bed variations & [2] Preserve sharp boundaries

    - Accurate powder bed texture reconstruction is critical for detecting: reacoter impact, and spatter
- Quantitative results
  - Significant reduction in MAE and CVD → Good for reconstructing pixel intensity & texture statistics
  - Large increase in PSNR → Significant improvements in overall image quality from the low-resolution sample
  - SSIM is less pronounced due to the existing large scale structural agreement between HR and LR samples
- Latent diffusion vs. Pixel-space diffusion: Enables real-time & high-throughput sample generation

  [1] Image quality metrics remain comparable across models, [2] Latent diffusion achieves ~13× faster inference

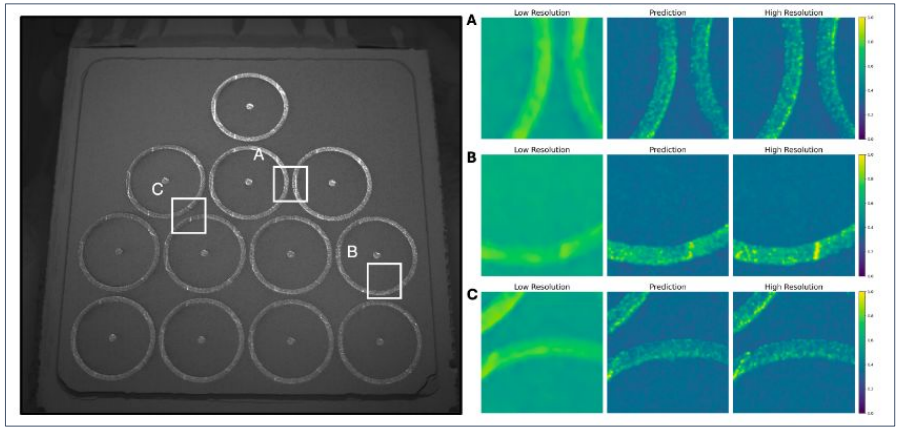
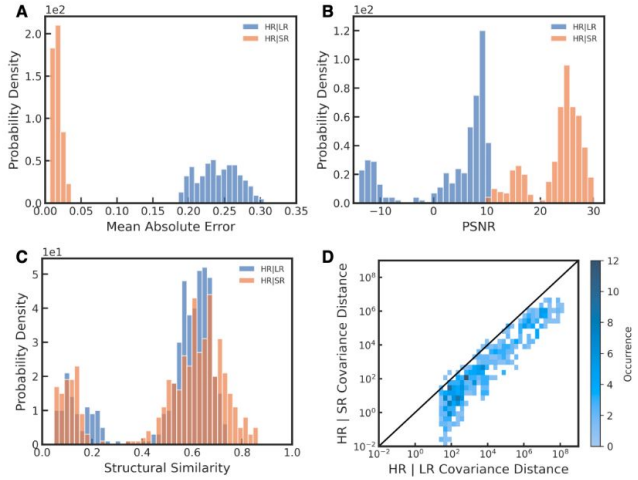


Table 2: Performance Metrics for the Latent Diffusion and Pixel Diffusion models.

Model	PSNR ↑	MAE ↓	nCVD ↓	SSIM ↑	Inference Time
Latent Diffusion	**23 ± 4.7**	**0.017 ± 0.01**	1.9 ± 1.1	**0.52 ± 0.23**	**0.01**
Pixel Space Diffusion	21 ± 4.6	0.025 ± 0.01	**0.28 ± 0.18**	0.46 ± 0.21	0.13



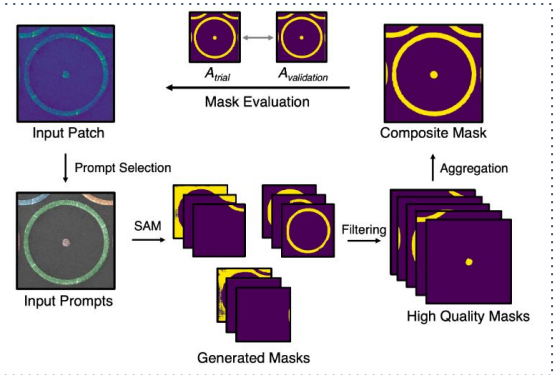
Configuration		Image Metrics			
Dataset	Comparison	MAE ↓	PSNR ↑	SSIM ↑	CVD ↓
Dataset A	HR | LR	0.241	3.20	0.512	3.03×10^{6}
Dataset A	HR | SR	**0.017**	**23.3**	**0.524**	$\mathbf{1.45\times10^{5}}$
Dataset B	HR | LR	0.134	15.0	**0.511**	1.31×10^{4}
Dataset B	HR | SR	**0.043**	**21.4**	0.464	$\mathbf{1.11\times10^{3}}$

❑ Literature Review

❖ Results (Ch.3)

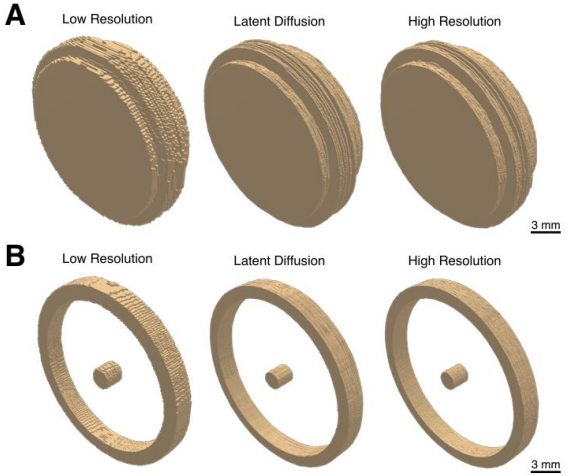
➢ Image Segmentation for 3D Morphology Evaluation

- Idea: i) Layer-wise optical images are segmented into: **Foreground** (part), **Background** (powder bed)
  ii) Segmentation enables reconstruction of 3D part morphology by stacking masks across layers
  iii) Compare two reconstructions of 3D part morphology btw generated layers and ground-truth
- **A pre-trained foundation model** is used for robust segmentation
  - **Segment Anything Model (SAM) by A. Kirillov et al.**
  - Masks are filtered based on confidence and stability



➢ Part-Based Results

- Setup: 1) Bounding boxes enclosing individual parts are extracted from the build plate
  2) Layer-wise **foreground masks** are extracted from optical images
  3) The 3D part geometry is reconstructed by stacking the segmented masks across layers
- New metrics added: IoU, H, V
  - **Intersection-over-Union (IoU)** between reconstructed volumes
  - **Hausdorff Distance (H)** between largest contours
  - **Voxel mismatch (V)** between 3D reconstructions
- Qualitative results
  - LR: [1] Contain minor artifacts, [2] Fail to accurately capture fine geometric details
  - Latent diffusion: [1] Closely match the HR geometry, [2] Preserve smooth surfaces and sharp boundaries
- Quantitative results: Latent diffusion substantially improves both image reconstruction quality and 3D geometric accuracy!
  - **91.3% reduction** in MAE, **188% reduction** in PSNR, and **18.8% increase** in SSIM
  - Improved IoU with reduced Hausdorff distance and voxel mismatch



Configuration		Optical Image			Part Reconstruction		
Dataset	Comparison	MAE ↓	PSNR ↑	SSIM ↑	IoU ↑	H ↓	V ↓
Dataset A	HR | LR	0.23	9.40	0.64	0.854	2.05	0.166
Dataset A	HR | SR	**0.02**	**27.12**	**0.76**	**0.875**	**0.36**	**0.140**