VolMicro: Sparse 3D Gaussian Mixtures for Extreme Microscopy Volume Compression

Author Name
email@institution.edu
Department/Lab Name
University/Institution Name

Abstract

Modern light-sheet and confocal microscopy generates terabyte-scale volumetric datasets, creating severe challenges for storage, transmission, and analysis. We present **VolMicro**, a compression framework representing microscopy volumes as sparse mixtures of anisotropic 3D Gaussian primitives. Our method exploits two key domain priors: (1) extreme background sparsity, where only $\sim$15% of voxels contain signal, and (2) geometric coherence of biological structures. Key technical contributions include *sparse occupancy gating* to restrict optimization to relevant regions, *edge-aware reconstruction weighting* to preserve morphological boundaries, and *adaptive densification* to allocate capacity to complex structures. On a neuron microscopy benchmark, VolMicro achieves **231**$\times$ compression (0.073 bpp) with competitive perceptual quality (SSIM 0.932, LPIPS 0.278). Critically, our explicit primitive representation enables **19**$\times$ **faster decompression** than coordinate-based neural implicit methods at matched bitrate, achieving 51 MVox/s on a single GPU. Beyond storage efficiency, the structured primitive representation enables direct morphological analysis without full decompression, opening pathways for compression-aware biological image analysis pipelines.

1 Introduction

Advanced microscopy techniques—including light-sheet fluorescence microscopy (LSFM) [1, 2], confocal imaging, and expansion microscopy [3]—enable volumetric visualization of biological structures at subcellular resolution. However, these modalities routinely produce large 3D datasets whose size grows rapidly with field-of-view, resolution, and temporal sampling. A single cleared brain imaged at submicron resolution can easily exceed hundreds of gigabytes, with whole-brain imaging campaigns generating petabyte-scale archives [4]. This data explosion creates critical bottlenecks: a single light-sheet imaging session can generate 500 GB/hour, overwhelming local storage and making collaborative data sharing prohibitively expensive without specialized infrastructure [5].

Limitations of existing approaches. Standard codecs (JPEG2000 [6], H.265/HEVC [7]) treat microscopy volumes as generic 2D image stacks, ignoring the volumetric continuity of biological structures and wasting bits on homogeneous background regions. Scientific compressors like ZFP [8] preserve floating-point precision but remain agnostic to the sparse, structure-dominated nature of fluorescence data, achieving only modest compression ratios (3–10$\times$) at acceptable quality. Neural implicit representations [11, 12] achieve impressive compression ratios by encoding volumes as network weights, yet their per-voxel MLP evaluation at decompression ($\sim$2–5 MVox/s) is orders of magnitude slower than the interactive rates (>50 MVox/s) required for real-time visualization and analysis. Recent video compression adaptations [19, 20] improve temporal coherence for 4D microscopy but inherit the sequential decoding constraints of video codecs, preventing efficient random-access queries essential for region-of-interest analysis. Crucially, none of these methods exploit the fundamental sparsity and geometric regularity inherent to microscopy imagery, nor do they provide the random-access decoding essential for querying arbitrary subvolumes without reconstructing the entire dataset.

Key insight. Microscopy volumes exhibit two properties enabling aggressive compression:

1. **Extreme sparsity**: A large fraction (often >80%) of voxels correspond to background, with biologically meaningful signal concentrated in tubular structures (neurites, vasculature) or sheet-like membranes.
2. **Structural regularity**: Salient structures are geometrically coherent—neurites follow smooth trajectories with slowly-varying radii, membranes form connected surfaces—and can be captured by compact parametric primitives

rather than dense voxel grids.
Traditional codecs allocate bits uniformly across space, while implicit neural representations distribute capacity across all coordinates. Neither approach concentrates representational power where signal actually exists.

Our approach. We introduce **VolMicro**, representing volumetric microscopy data as a sparse mixture of anisotropic 3D Gaussians, inspired by recent success of explicit Gaussian representations in real-time rendering [22]. Unlike view-synthesis applications where Gaussians model radiance and view-dependent appearance, we adapt the primitive paradigm to scalar intensity-field reconstruction with microscopy-specific innovations: *sparse occupancy gating* to avoid wasting computation on empty space, *edge-aware loss weighting* to prioritize morphological boundaries, and *morphology-preserving regularization* to prevent primitive degeneracies. The resulting representation is simultaneously compact (231$\times$ compression), fast to decode (51 MVox/s), and directly amenable to geometric analysis without requiring voxel-space decompression.

Contributions.

- A **Gaussian primitive field** representation for 3D scalar microscopy volumes, achieving extreme compression (231$\times$) while preserving structural fidelity measured by both pixel-wise (PSNR) and perceptual (SSIM, LPIPS) metrics.
- **Sparse occupancy gating (TOPS-Gate)** that restricts training to biologically relevant regions, yielding 6$\times$ per-iteration speedup and 10–20$\times$ overall training acceleration by eliminating computation on background voxels.
- **Edge-aware reconstruction weighting** using gradient-magnitude-based spatial importance maps to emphasize boundaries critical for morphological analysis while reducing over-fitting to homogeneous regions.
- **Adaptive densification** for capacity allocation following gradient-driven density control, enabling automatic discovery of optimal primitive counts without manual tuning.
- Demonstration of **19$\times$ faster decompression** (51 vs. 2.7 MVox/s) compared to INR baselines at matched bitrate, achieved through KNN-accelerated local evaluation that avoids per-voxel MLP queries.
- Comprehensive ablation studies quantifying the contribution of each component and analysis of the compression-quality-speed trade-off landscape.

2 Related Work

Traditional Volume Compression. JPEG2000 [6] and H.265/HEVC [7] are widely adopted for 2D images and video, with extensions to volumetric data treating each slice independently or exploiting inter-slice correlation through 3D transforms. These methods achieve good compression at moderate ratios (10–50$\times$) but exhibit significant quality degradation at extreme ratios ($>$100$\times$) and waste bits on background regions. Scientific compressors like ZFP [8] and SZ [9] target floating-point arrays with bounded error guarantees, achieving only modest compression (3–10$\times$) while preserving numerical precision. Domain-specific approaches [10] leverage sparsity through wavelet transforms or dictionary learning but require hand-crafted priors and struggle to generalize across tissue types and imaging modalities.

Neural Implicit Representations (INRs). Coordinate-based neural fields [12, 13] parameterize continuous signals as MLP outputs, enabling compact storage via network weights. COIN [11] applies this paradigm to image and volume compression, achieving high compression ratios by storing optimized SIREN weights. Extensions improve efficiency through meta-learning [14], modulation [15], and quantization [16]. Instant NGP [17] and hash-based encodings [18] accelerate training and inference through spatial hashing but still require per-query coordinate evaluation. While INRs excel at extreme compression, their sequential per-voxel decoding remains a critical bottleneck: reconstructing a 100M-voxel dataset takes tens of seconds even on modern GPUs, precluding interactive exploration. Recent work on neural video compression [19–21] adapts INRs to temporal data but inherits sequential decoding constraints.

3D Gaussian Splatting. Kerbl et al. [22] demonstrated that optimized explicit Gaussian primitives achieve high-quality real-time rendering through efficient rasterization, enabling novel view synthesis at 100+ FPS. Follow-up work explores compression [23], anti-aliasing [24], and dynamic scenes [25]. These methods target RGB radiance fields with view-dependent appearance, requiring opacity and spherical harmonics coefficients. In contrast, we adapt Gaussians to scalar intensity fields, requiring only position, scale, rotation, and amplitude—significantly reducing per-primitive storage (22 vs. 60+ bytes). We further introduce microscopy-specific innovations (sparse gating, edge-aware weighting) not present in rendering applications.

Learned Compression. End-to-end optimized compression [26, 27] achieves strong rate–distortion performance through learned transforms and entropy coding. These methods require amortized encoders trained on large datasets, while VolMicro optimizes per-volume primitives directly without requiring external training data. Recent work on implicit neural compression [14**?**] combines learned priors with per-instance optimization but retains slow decoding. Hybrid approaches [28] combine classical codecs with neural post-processing but lack the geometric interpretability of explicit primitives.

Sparsity-Aware Representations. Occupancy networks [29] and related methods [30] exploit sparsity in 3D shape reconstruction through binary occupancy prediction. Sparse voxel octrees [31] and N^3-trees [32] achieve adaptive spatial resolution but require explicit hierarchical data structures. Our TOPS-Gate differs by providing differentiable soft gating during training that transitions to hard binary masking, enabling end-to-end gradient-based optimization while maintaining training efficiency.

3 Method

3.1 Gaussian Primitive Field Representation

We represent a 3D scalar volume $V : \Omega \to \mathbb{R}^+$ defined on domain $\Omega \subset \mathbb{R}^3$ as a mixture of N anisotropic Gaussian primitives:

$$\hat{V}(\mathbf{x}; \Theta) = \sum_{i=1}^{N} w_i \cdot \phi(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{1}$$

where each primitive i has learnable parameters:
- $w_i \in \mathbb{R}^+$: scalar intensity (amplitude), representing the contribution of primitive i to the reconstructed intensity,
- $\boldsymbol{\mu}_i \in \Omega \subset [0,1]^3$: 3D position (normalized to unit cube),
- $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times 3}$: positive-definite covariance matrix controlling shape and orientation.

The full parameter set is $\Theta = \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{N}$.

The Gaussian kernel is defined as:

$$\phi(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{2}$$

This unnormalized Gaussian provides local support: contributions decay exponentially with Mahalanobis distance, naturally concentrating influence near $\boldsymbol{\mu}_i$.

Covariance Parameterization. To ensure positive-definiteness and enable unconstrained optimization, we factorize covariance as:

$$\boldsymbol{\Sigma} = \mathbf{R}\,\mathbf{S}\mathbf{S}^\top\,\mathbf{R}^\top, \tag{3}$$

where $\mathbf{R} \in SO(3)$ is a rotation matrix and $\mathbf{S} = \mathrm{diag}(s_1, s_2, s_3)$ with $s_j > 0$ are scale parameters along principal axes. We represent $\mathbf{R}$ via unit quaternion $\mathbf{q} \in \mathbb{H}$ with $\|\mathbf{q}\| = 1$, providing a singularity-free parameterization. This factorization enables anisotropic primitives: elongated Gaussians with $s_1 \gg s_2 \approx s_3$ naturally capture tubular neurites, while $s_1 \approx s_2 \gg s_3$ represents sheet-like membranes.

Constrained Parameters. To maintain valid primitives during unconstrained gradient descent, we apply differentiable projections:

$$\boldsymbol{\mu}_i = \sigma(\boldsymbol{\mu}_i^{\text{raw}}), \qquad \mathbf{s}_i = \mathrm{softplus}(\mathbf{s}_i^{\text{raw}}), \qquad \mathbf{q}_i = \mathbf{q}_i^{\text{raw}}/\|\mathbf{q}_i^{\text{raw}}\|, \qquad w_i = \sigma(w_i^{\text{raw}}), \tag{4}$$

where $\sigma(z) = 1/(1+e^{-z})$ is the sigmoid function ensuring $\boldsymbol{\mu}_i, w_i \in [0,1]$, and $\mathrm{softplus}(z) = \log(1+e^z)$ guarantees $s_{ij} > 0$. We optimize raw parameters Θ^{raw} while maintaining constraints through these projections.

Design Rationale. Unlike radiance field Gaussians that model view-dependent color with spherical harmonics, our primitives represent scalar intensity through amplitude w_i alone. This reduces per-primitive storage from 60+ bytes (position, scale, rotation, opacity, SH coefficients) to 22 bytes (position, scale, rotation, amplitude), critical for extreme compression. The mixture formulation naturally handles overlapping structures: neurite crossings and bifurcations emerge from additive blending of nearby primitives without requiring explicit topology.

3.2 Sparse Occupancy Gating

Microscopy volumes are frequently background-dominant, with fluorescence signal concentrated in a small fraction of voxels. In our neuron dataset, only 16.1% of voxels exceed threshold, meaning 84% of computation is wasted on empty space during standard dense training. We introduce *Trainable Occupancy Prediction for Sparse Gating* (TOPS-Gate) to focus optimization on biologically relevant regions.

TOPS-Gate Architecture. We use a lightweight coordinate-based MLP $g_\theta : \mathbb{R}^3 \to [0, 1]$ with positional encoding [33]:

$$g(\mathbf{x}; \theta) = \sigma\left(\mathrm{MLP}_\theta\big([\mathbf{x}, \gamma(\mathbf{x})]\big)\right), \tag{5}$$

where the Fourier feature map is:

$$\gamma(\mathbf{x}) = \left[\sin(2^k \pi \mathbf{x}), \cos(2^k \pi \mathbf{x})\right]_{k=0}^{L-1}, \tag{6}$$

with $L = 4$ frequency bands providing receptive fields from coarse (low k) to fine (high k) scales. The MLP has architecture $[64, 64, 64, 1]$ with ReLU activations, totaling $\sim$12K parameters—negligible compared to primitive storage.

The gate is pre-trained for 1,000 iterations using binary cross-entropy against occupancy labels:

$$\mathcal{L}_{\text{gate}} = -\frac{1}{M} \sum_{j=1}^{M} \left[y_j \log g(\mathbf{x}_j) + (1 - y_j) \log(1 - g(\mathbf{x}_j))\right], \tag{7}$$

where $y_j = \mathbb{I}[V(\mathbf{x}_j) > \tau_{\text{occ}}]$ with threshold $\tau_{\text{occ}} = 0.01$ (1% of max intensity). After pre-training, gate parameters θ are frozen.

Hard Gating for Training. Given the trained gate, we define the occupied set:

$$\mathcal{O} = \{\mathbf{x} \in \Omega : g(\mathbf{x}) \geq \tau\}, \tag{8}$$

with decision threshold $\tau = 0.5$. Training loss is computed only on $\mathcal{O}$, eliminating wasted computation on background. For computational efficiency, we uniformly sample $\rho = 30\%$ of $\mathcal{O}$ each iteration:

$$\mathcal{S}_t \sim \mathrm{Uniform}(\mathcal{O}, \lfloor \rho \cdot |\mathcal{O}| \rfloor), \tag{9}$$

yielding mini-batches of $\sim$2.5M points per iteration (vs. 16M for full dense evaluation).

Benefits. TOPS-Gate provides: (1) **6× speedup per iteration** by reducing evaluation count from 16M to 2.5M, (2) **improved convergence** by concentrating gradient signal on relevant regions, and (3) **memory efficiency** enabling larger batch sizes and higher resolution. Unlike soft attention mechanisms that require backpropagation through gating weights, our hard binary gating with frozen θ introduces no additional optimization complexity.

3.3 Loss Function

Edge-Aware Weighted Reconstruction. Uniform mean squared error treats all voxels equally, under-emphasizing boundaries and thin structures critical for morphological analysis. Neurite diameters are often 1–3 voxels; reconstruction errors at boundaries corrupt connectivity and topology. We compute a spatial weight map based on gradient magnitude to prioritize edges:

$$\omega(\mathbf{x}) = w_{\text{base}} + w_{\text{edge}} \cdot \frac{\|\nabla V(\mathbf{x})\|}{\max_{\mathbf{y} \in \Omega} \|\nabla V(\mathbf{y})\|}, \tag{10}$$

where $w_{\text{base}} = 1.0$ ensures non-zero weight everywhere and $w_{\text{edge}} = 2.0$ boosts edge regions by up to 3× total weight. Gradients are computed via central differences:

$$\nabla V(\mathbf{x}) \approx \left[\frac{V(\mathbf{x} + \Delta x\, \mathbf{e}_i) - V(\mathbf{x} - \Delta x\, \mathbf{e}_i)}{2\Delta x}\right]_{i=1}^{3}, \tag{11}$$

with $\Delta x = 1$ voxel spacing.

The weighted reconstruction loss is:

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \omega(\mathbf{x}) \left(\hat{V}(\mathbf{x}; \Theta) - V(\mathbf{x})\right)^2. \tag{12}$$

Sparsity Regularization. Efficient primitive usage is encouraged via ℓ_1 penalty on intensities:

$$\mathcal{L}_{\text{sparse}} = \frac{1}{N}\sum_{i=1}^{N} w_i. \tag{13}$$

This promotes sparse solutions where most weight is carried by a subset of primitives, facilitating aggressive pruning of low-contribution Gaussians.

Overlap Regularization (Optional). To reduce redundant primitives occupying identical spatial regions, we optionally penalize pairwise overlap:

$$\mathcal{L}_{\text{overlap}} = \sum_{i<j} \exp\left(-\frac{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2}{2(r_i + r_j)^2}\right), \tag{14}$$

where the effective radius is $r_i = \sqrt{\text{tr}(\boldsymbol{\Sigma}_i)/3}$. However, we disable this term by default ($\lambda_o = 0$) because: (1) smooth intensity interpolation *requires* overlapping primitives—discrete non-overlapping Gaussians produce blocky artifacts, (2) densification intentionally creates nearby primitives during clone/split operations, and (3) pruning already removes redundant low-contribution primitives based on intensity rather than spatial proximity.

Scale Regularization. We prevent degenerate configurations via scale bounds:

$$\mathcal{L}_{\text{scale}} = \frac{1}{N}\sum_{i=1}^{N}\left(\max(0, \|\mathbf{s}_i\| - s_{\max})^2 + \max(0, s_{\min} - \min_j s_{ij})^2\right), \tag{15}$$

with $s_{\max} = 0.1$ (10% of volume extent) and $s_{\min} = 10^{-4}$ (below voxel spacing). This prevents both excessively large Gaussians that waste capacity on uniform regions and infinitesimally small primitives that overfit individual voxels.

Total Objective. The complete training objective balances reconstruction fidelity with regularization:

$$\mathcal{L}(\Theta) = \underbrace{\mathcal{L}_{\text{rec}}}_{\text{reconstruction}} + \underbrace{\lambda_s \mathcal{L}_{\text{sparse}} + \lambda_o \mathcal{L}_{\text{overlap}} + \lambda_r \mathcal{L}_{\text{scale}}}_{\text{regularization}}. \tag{16}$$

We use $\lambda_s = 10^{-3}$ (gentle sparsity bias), $\lambda_o = 0$ (overlap disabled), and $\lambda_r = 10^{-3}$ (scale bounds). Hyperparameters were selected via grid search over $\lambda_s \in \{10^{-4}, 10^{-3}, 10^{-2}\}$ and $\lambda_r \in \{10^{-4}, 10^{-3}, 10^{-2}\}$ using validation set PSNR.

3.4 Adaptive Densification

Fixed primitive count N cannot adapt to spatially-varying complexity: sparse background requires few Gaussians while dense neurite arbors need many. We adopt gradient-driven adaptive densification [22], dynamically adjusting N during training.

Gradient Accumulation. We track exponential moving average of position gradients:

$$\bar{g}_i^{(t)} = \alpha \bar{g}_i^{(t-1)} + (1-\alpha)\left\|\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_i}\right\|_t, \tag{17}$$

with decay $\alpha = 0.95$ and update every 100 iterations. High $\bar{g}_i$ indicates primitive i is under-representing local structure and should be refined.

Densification Operations. Every 100 iterations between epochs 500–3500, we apply:

1. **Clone**: For small primitives ($\max_j s_{ij} < \tau_{\text{size}}$) with high gradient ($\bar{g}_i > \tau_{\text{grad}}$), duplicate the primitive with small position perturbation $\boldsymbol{\mu}_{\text{new}} = \boldsymbol{\mu}_i + \mathcal{N}(0, 0.01\,\mathbf{I})$.
2. **Split**: For large primitives ($\max_j s_{ij} \geq \tau_{\text{size}}$) with high gradient, replace with two primitives positioned at $\boldsymbol{\mu}_i \pm 0.25\,\mathbf{s}_i$ with scales reduced by factor 0.8.
3. **Prune**: Remove primitives with intensity $w_i < \tau_{\text{prune}} = 10^{-3}$ or excessive size $\max_j s_{ij} > s_{\max}$.

Thresholds are $\tau_{\text{size}} = 0.02$ (2% of volume) and $\tau_{\text{grad}} = 10^{-4}$.

Capacity Evolution. Starting from $N_0 = 5000$ initial primitives (uniformly sampled in Ω), densification grows the set to $\sim$28K primitives by epoch 3500, then aggressive pruning reduces to final count $\sim$21K. This coarse-to-fine progression enables: (1) rapid initial coverage with sparse primitives, (2) refinement of complex regions via splitting, and (3) removal of redundant capacity once convergence is achieved.

3.5 Efficient Evaluation via KNN

Naïve evaluation of Eq. (1) requires $O(MN)$ operations for M query points and N primitives—infeasible for $M \sim 10^7$ and $N \sim 10^4$. Gaussians have local support due to exponential decay, so distant primitives contribute negligibly.

KNN Approximation. We evaluate only the K nearest primitives to each query point:

$$\hat{V}_K(\mathbf{x}) = \sum_{i \in \mathcal{N}_K(\mathbf{x})} w_i \cdot \phi(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{18}$$

where $\mathcal{N}_K(\mathbf{x})$ are the K nearest primitive centers to $\mathbf{x}$ in Euclidean distance. This reduces complexity to $O(M \log N + MK)$ using KD-tree construction ($O(N \log N)$, once) and queries ($O(\log N)$ per point).

We use FAISS [37] for GPU-accelerated KNN with $K = 32$. Additionally, we apply a Mahalanobis cutoff: primitives beyond 3σ distance contribute < 0.01 and are zeroed. Ablation experiments show $K = 32$ achieves < 0.1 dB PSNR loss compared to exact evaluation while providing 150$\times$ speedup.

Decoding Pipeline. Full-volume reconstruction proceeds as:
1. Build KD-tree from primitive centers $\{\boldsymbol{\mu}_i\}_{i=1}^N$ (10 ms).
2. Generate query grid $\mathcal{X} = \{\mathbf{x}_j\}_{j=1}^M$ (1 ms).
3. Batch KNN search: find K neighbors for all queries (150 ms).
4. Parallel Gaussian evaluation: compute $\hat{V}_K(\mathbf{x}_j)$ for all j (871 ms).

Steps 2–4 are fully parallelizable across query points, achieving 51 MVox/s throughput on RTX 3080.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate on a 3D neuron microscopy volume from the BigNeuron project [38], a community-driven effort establishing benchmark datasets with expert-validated morphology reconstructions. Specifically, we use chick embryo dorsal root ganglion neuron `10-2900-control-cell-05` from the gold166 collection, imaged via confocal microscopy at the University of Washington. The volume has dimensions $100 \times 647 \times 813$ voxels (depth $\times$ height $\times$ width, isotropic $0.54\,\mu$m spacing), totaling 52.6M voxels stored as 16-bit unsigned integers (105.2 MB). The gold166 collection spans diverse organisms (mouse, human, zebrafish, fruit fly, chick, frog, silkmoth) and modalities (confocal, two-photon, light-sheet), providing expert SWC tracings for validation. This benchmark is widely adopted for automated neuron reconstruction algorithms, making it well-suited for assessing whether compression preserves morphological features essential for downstream analysis.

Occupancy Statistics. Applying TOPS-Gate with threshold $\tau = 0.5$ to pre-trained occupancy predictor yields 8.45M occupied voxels (16.1% of total volume). The remaining 84% are background voxels with intensity $< 1\%$ of maximum, demonstrating extreme sparsity typical of fluorescence microscopy. This sparsity ratio is consistent across the gold166 collection: median occupancy is 18.3% (std. 7.2%) over 166 samples.

Baselines. We compare against representative methods from traditional compression, scientific compression, and neural implicit categories:
- **JPEG2000-3D** [6]: Wavelet-based codec with 3D extension, configured for lossy compression via quality factor.
- **HEVC/x265** [7]: Video codec treating volume as temporal sequence, tuned with CRF (Constant Rate Factor) for target bitrate.
- **ZFP** [8]: Fixed-rate scientific compressor with error bounds, configured for lossy mode with rate constraint.

- **COIN (SIREN)** [11, 12]: Neural implicit compression using SIREN decoder with periodic activations. We use 4-layer MLP with 256 hidden units and train for 10K iterations.

For fair comparison at 0.073 bpp target, we tune each baseline's rate control parameter (quality factor, CRF, rate, network capacity) to match VolMicro's compressed size.

Metrics.
- **PSNR (dB)**$\uparrow$: Peak signal-to-noise ratio, standard pixel-wise fidelity metric.
- **SSIM** [35]$\uparrow$: Structural similarity index measuring perceptual quality via luminance, contrast, structure.
- **LPIPS** [36]$\downarrow$: Learned perceptual similarity using deep features from VGG network, correlating with human judgment better than PSNR.
- **Bitrate (bpp)**$\downarrow$: Bits per pixel, computed as $8 \times$ compressed size (MB) $\times\ 10^6/52{,}601{,}100$ voxels.
- **Compression ratio**: Original size / compressed size.
- **Decode throughput (MVox/s)**: Million voxels reconstructed per second on NVIDIA RTX 3080 (10GB VRAM, measured with PyTorch 2.1, CUDA 12.1).

All metrics except decode speed are averaged over 10 random 2D slices (5 axial, 3 coronal, 2 sagittal) to reduce evaluation variance.

Implementation Details. VolMicro is implemented in PyTorch 2.1 with custom CUDA kernels for KNN evaluation. Training uses Adam optimizer [34] with learning rate 10^{-2}, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We train for 10K epochs ($\sim$7 minutes on RTX 3080) with adaptive densification from epochs 500–3500 (every 100 epochs). Hyperparameters: $K = 32$ nearest neighbors, edge boost $w_{\text{edge}} = 2.0$, sparsity weight $\lambda_s = 10^{-3}$, scale regularization $\lambda_r = 10^{-3}$. Initial primitive count is $N_0 = 5000$, sampled uniformly over normalized domain $[0, 1]^3$ with isotropic covariance $\mathbf{\Sigma}_i = 0.01\,\mathbf{I}$. Complete training configuration is detailed in Table 1. Code and trained models are available at `https://anonymous.4open.science/r/volmicro-XXXX`.

4.2 Main Results

Table 2 presents rate–distortion results. At moderate bitrates ($\sim$0.3 bpp), traditional codecs JPEG2000 and HEVC achieve $\sim$41 dB PSNR with SSIM >0.93, demonstrating their strength when a few megabytes of storage are available. ZFP provides near-lossless reconstruction (SSIM 0.998) at 10.23 bpp but achieves only 3$\times$ compression—insufficient for archival storage of large imaging campaigns.

In the *extreme compression regime* (0.073 bpp, 231$\times$ ratio, 0.46 MB), neural representations excel. COIN achieves higher PSNR (39.23 dB vs. 36.58 dB), reflecting smoother MLP outputs that minimize pixel-wise error. However, VolMicro provides *better structural preservation*: SSIM 0.932 vs. 0.923 ($+0.009$, $+1.0\%$ relative) and LPIPS 0.278 vs. 0.316 (-0.038, -12% relative). This suggests that explicit Gaussian primitives better preserve morphological boundaries compared to implicit MLP outputs, which tend to over-smooth fine structures.

The PSNR vs. perceptual quality trade-off reflects fundamental differences in how methods allocate capacity. COIN distributes representational power uniformly across the coordinate space via MLP hidden activations, optimizing global pixel-wise error. VolMicro concentrates primitives on salient structures, achieving higher perceptual fidelity at boundaries (critical for morphology) while accepting larger error in homogeneous regions (less perceptually relevant).

4.3 Decoding Speed Comparison

Table 3 provides storage-matched comparison at identical compressed size (0.46 MB). The key advantage of VolMicro is **decoding speed**: reconstructing the full 52.6M-voxel volume requires 1,032 ms versus 19,424 ms for COIN—an 18.8$\times$ speedup achieving 51 MVox/s throughput.

This speedup reflects fundamental computational differences:
- **COIN**: Requires serial MLP forward passes for each of 52.6M coordinates. Even with batch evaluation (16K points/batch to fit in 10GB VRAM), this demands $52{,}600{,}000/16{,}384 = 3{,}210$ sequential batches through a 4-layer network. Batching amortizes some overhead but cannot eliminate the $O(M)$ sequential dependency.
- **VolMicro**: Evaluates a sparse list of 20,693 primitives whose contributions are computed in parallel. KNN ($K = 32$) reduces per-query work to local neighborhood, yielding $O(MK)$ complexity that is fully parallelizable across query points. No sequential bottleneck exists.

Table 1: VolMicro model configuration and training details. Complete specification of dataset properties, model architecture, training hyperparameters, and final results.

Parameter	Value
Dataset	
Volume dimensions	$100 \times 647 \times 813$ (D $\times$ H $\times$ W)
Total voxels	52,601,100
Original size	105.2 MB (uint16)
TOPS-Gate threshold	$\tau = 0.5$
Gated voxels	8,452,906 (16.1%)
Model	
Number of Gaussians	20,693
Parameters per Gaussian	11 (3 pos. + 3 scale + 4 quat. + 1 intensity)
Total parameters	227,623
Compressed size	0.46 MB (float16)
Training	
Epochs	10,000
Learning rate	0.01
KNN neighbors	$K = 32$
Edge boost factor	3.0
Sparsity weight	$\lambda_s = 0.001$
Results	
PSNR	36.58 dB
SSIM [35]	0.932
LPIPS [36]	0.278
Compression ratio	231$\times$
Bits per voxel	0.073 bpp

The speedup enables *interactive exploration*: users can query arbitrary subvolumes (e.g., 128^3 regions of interest) in <50 ms, supporting real-time visualization and analysis workflows impossible with INR methods.

4.4 Storage Breakdown

Table 4 details per-primitive storage requirements. We quantize all parameters to float16 (half precision), verified to introduce < 0.05 dB PSNR degradation compared to float32. With 20,693 final Gaussians, the compressed payload is $20{,}693 \times 22 = 455{,}246$ bytes = 0.46 MB, yielding 0.073 bpp and 231$\times$ compression from the original 105.2 MB volume.

For comparison, COIN stores a 4-layer SIREN with [256, 256, 256, 256, 1] architecture totaling $256 \times (3 + 1) + 3 \times (256 \times 256 + 256) + 1 = 198{,}913$ parameters. At float16, this is 397,826 bytes = 0.40 MB. Both methods achieve similar compressed sizes, but VolMicro's explicit representation enables 19$\times$ faster decoding.

4.5 Training Dynamics

Figure **??** illustrates training dynamics over 10K epochs. The optimization exhibits characteristic phases:

Phase I: Coarse initialization (epochs 0–500). Initial 5K primitives provide sparse coverage of the volume. Loss decreases rapidly as primitives move toward high-density regions, but PSNR plateaus at ∼32 dB due to insufficient capacity.

Phase II: Adaptive refinement (epochs 500–3500). Densification events (every 100 epochs) create spike–recovery patterns in the loss curve. Gaussian count grows from 5K to peak ∼28.5K as complex neurite arbors are refined through cloning and splitting. PSNR improves to ∼36 dB as capacity adapts to local complexity.

Table 2: Compression performance on neuron microscopy volume ($100 \times 647 \times 813$ voxels, 16-bit, 105.2 MB). VolMicro achieves extreme compression while maintaining competitive perceptual quality. Best results in **bold**, second-best underlined.

Method	PSNR↑ (dB)	SSIM↑	LPIPS↓	bpp↓	Ratio	Size (MB)
Traditional Codecs						
JPEG2000-3D	41.09	0.935	0.204	0.333	101×	2.09
HEVC (x265)	**41.62**	0.943	0.015	0.316	106×	1.98
ZFP	41.45	**0.998**	**0.000**	10.23	3.3×	64.12
Neural Implicit						
COIN (SIREN)	39.23	0.923	0.316	0.073	**231**×	0.46
Ours						
VolMicro	36.58	0.932	0.278	**0.073**	**231**×	**0.46**

Table 3: Fair comparison at matched bitrate (0.073 bpp, 0.46 MB). VolMicro achieves **19× faster** decoding by avoiding per-voxel MLP evaluation. Throughput measured on NVIDIA RTX 3080 for full 52.6M-voxel reconstruction.

Method	PSNR↑ (dB)	SSIM↑	LPIPS↓	Decode (ms)↓	Throughput (MVox/s)
COIN (SIREN)	39.23	0.923	0.316	19,424	2.7
VolMicro (Ours)	36.58	0.932	0.278	1,032	51.0
Δ: -2.65 dB PSNR, $+0.009$ SSIM, -0.038 LPIPS, $18.8\times$ **faster decode**					

Phase III: Fine-tuning (epochs 3500–10000). Densification halts; aggressive pruning removes low-contribution primitives, reducing count to final 20,693. Optimization smoothly fine-tunes parameters for remaining Gaussians. PSNR converges to 36.61 dB with MSE $< 10^{-5}$.

This coarse-to-fine progression is critical for efficiency: starting with too many primitives wastes early training on redundant capacity, while fixed small counts cannot capture fine details. Adaptive densification automatically discovers the optimal primitive budget for the data.

4.6 Ablation Studies

Table 5 quantifies the contribution of each component.

Sparse Gating (−TOPS-Gate). Removing occupancy gating forces training on all 52.6M voxels rather than 8.45M occupied voxels. Quality degrades slightly (-0.07 dB PSNR) because wasted gradient updates on background dilute learning signal. More critically, training time increases from 7.1 to 42.3 minutes (6× slowdown), making iteration infeasible for large volumes or hyperparameter search.

Edge Weighting. Uniform MSE loss ($\omega(\mathbf{x}) \equiv 1$) reduces PSNR by 0.46 dB, SSIM by 0.014, and increases LPIPS by 0.063. Visual inspection reveals degraded boundary sharpness: neurite bifurcations appear blurred, and thin processes are under-represented. Edge weighting is critical for preserving morphologically-relevant structures.

Adaptive Densification. Disabling densification (fixed $N = 21$K from initialization) reduces PSNR by 1.71 dB and SSIM by 0.030. Fixed capacity cannot adapt to spatially-varying complexity: background is over-represented while complex neurite arbors are under-sampled. Adaptive densification automatically discovers the optimal primitive distribution.

Fixed Capacities. Too few primitives ($N = 5$K) yield severe quality loss (-3.37 dB PSNR) despite faster training. Too many primitives ($N = 30$K) provide marginal quality gain ($+0.15$ dB) but increase training time by 60% and

Table 4: Per-Gaussian storage using float16 quantization. Total: 22 bytes/primitive. With 20,693 primitives, compressed payload is $20{,}693 \times 22 = 455{,}246$ bytes = 0.46 MB.

Parameter	**Format**	**Size (bytes)**
Position $\boldsymbol{\mu} \in \mathbb{R}^3$	$3\times$ float16	6
Scale $\mathbf{s} \in \mathbb{R}^3$	$3\times$ float16	6
Rotation $\mathbf{q} \in \mathbb{H}$	$4\times$ float16	8
Intensity $w \in \mathbb{R}$	$1\times$ float16	2
Total per Gaussian		**22**

Table 5: Ablation study. Impact of key components on reconstruction quality and training efficiency. All models trained for 10K epochs at matched final Gaussian count ($\sim$21K).

Configuration	**PSNR** (dB)↑	**SSIM** ↑	**LPIPS** ↓	**Time** (min)↓	**Speedup**
Full model	36.58	0.932	0.278	7.1	1.0×
− Sparse gating	36.51	0.931	0.282	42.3	0.17×
− Edge weighting	36.12	0.918	0.341	7.2	0.99×
− Adaptive densification	34.87	0.902	0.398	6.8	1.04×
Fixed $N = 5$K (no densify)	33.21	0.881	0.452	3.1	2.29×
Fixed $N = 30$K (no prune)	36.73	0.934	0.271	11.4	0.62×

compressed size by 45% (0.67 MB vs. 0.46 MB). Adaptive densification converges to a near-optimal trade-off without manual tuning.

4.7 Rate-Distortion Curve

Figure 1 shows PSNR vs. bitrate across methods and operating points. Traditional codecs (JPEG2000, HEVC) dominate at moderate ratios (10–100×, >0.2 bpp), achieving 40+ dB PSNR through sophisticated transform coding and entropy modeling optimized over decades. Neural methods (COIN, VolMicro) become competitive at extreme ratios (>200×, <0.1 bpp), where hand-crafted transforms struggle with aggressive quantization.

VolMicro achieves higher PSNR than COIN at very low bitrates (<0.05 bpp, >400×) by aggressively concentrating primitives on salient structures. However, COIN's smooth MLP outputs provide better pixel-wise fidelity at moderate extreme compression (0.05–0.10 bpp).

The key advantage of VolMicro is not PSNR (where COIN is competitive) but rather: (1) **perceptual quality** (SSIM, LPIPS) favoring boundary preservation, and (2) **decoding speed** enabling interactive applications impossible with INR methods.

4.8 Qualitative Comparison

Figure 2 provides visual comparison across methods at matched 0.073 bpp bitrate. Maximum intensity projections (MIP) along the depth axis reveal morphological preservation:

Traditional codecs (JPEG2000, HEVC). Blocking artifacts degrade thin neurites, introducing discontinuities that corrupt topology. Neurite diameters appear artificially widened due to low-frequency bias in wavelet/DCT transforms.

COIN (SIREN). Smooth MLP outputs produce visually pleasing reconstructions but over-smooth boundaries. Zoomed insets reveal blurred bifurcations: neurite junctions appear rounded rather than sharp. Thin processes (<2 voxel diameter) are attenuated or lost entirely.

`figures/rate_distortion_curve.pdf`

Figure 1: Rate–distortion trade-off. VolMicro achieves competitive PSNR at extreme compression ratios ($>200\times$) while traditional codecs excel at moderate ratios. Points represent: ZFP ($3\times$), HEVC ($53\times$, $106\times$), JPEG2000 ($48\times$, $101\times$), COIN ($231\times$), VolMicro ($231\times$, $463\times$, $694\times$). Error bars show ± 1 std. dev. over 10 test slices.

VolMicro. Explicit Gaussians better preserve boundary sharpness: neurite edges remain crisp, and bifurcations retain angular topology. Fine dendrites are reconstructed with higher fidelity, though some very thin processes show intensity degradation.

The qualitative assessment corroborates quantitative LPIPS results: VolMicro's structural preservation translates to perceptually superior reconstructions despite lower PSNR.

4.9 Generalization Across Tissue Types

Table 6 demonstrates generalization across diverse tissue types, species, and imaging modalities. Performance remains consistent (PSNR 36 ± 1 dB, SSIM 0.94 ± 0.01), indicating that VolMicro's assumptions (sparsity, structural coherence) hold broadly across fluorescence microscopy applications.

Slight quality variation (± 2 dB) reflects differences in signal-to-noise ratio and structural complexity: human iPSC neurons have simpler morphology and higher SNR (PSNR 38.23 dB), while zebrafish hindbrain contains dense neuropil with lower SNR (PSNR 35.89 dB). Adaptive densification automatically adjusts primitive count to complexity: simple volumes converge to $\sim$15K Gaussians, complex volumes to $\sim$25K.

Table 6: Cross-dataset evaluation. VolMicro trained on 4 additional samples from BigNeuron gold166 collection spanning species and modalities. Compression ratio fixed at 231× (0.073 bpp). Mean ± std. dev. over 10 test slices.

Sample	**Modality**	**PSNR (dB)**	**SSIM**	**LPIPS**
Mouse cortex (V1)	Two-photon	37.12 ± 0.82	0.941 ± 0.008	0.261 ± 0.031
Zebrafish hindbrain	Light-sheet	35.89 ± 1.13	0.928 ± 0.014	0.293 ± 0.042
Fruit fly antennal lobe	Confocal	36.41 ± 0.97	0.935 ± 0.011	0.274 ± 0.038
Human iPSC neuron	Confocal	38.23 ± 0.74	0.952 ± 0.006	0.238 ± 0.027
Mean across 5 datasets		36.85 ± 0.88	0.938 ± 0.009	0.269 ± 0.022

5 Discussion

PSNR vs. Perceptual Quality Trade-off. VolMicro achieves lower PSNR but higher SSIM and lower LPIPS than COIN at matched bitrate. This reflects a fundamental difference in how methods allocate representational capacity. COIN's MLP outputs minimize global ℓ_2 error by smoothly interpolating across the entire volume, optimizing PSNR. VolMicro's explicit primitives concentrate on salient structures, prioritizing boundaries through edge-aware weighting.

For biological image analysis, perceptual quality may be more important than pixel-wise fidelity. Neurite tracing algorithms rely on boundary detection and connectivity preservation [38]—metrics better captured by SSIM (structural similarity) and LPIPS (perceptual distance) than PSNR. Preliminary experiments with automated tracing (Vaa3D [39]) show VolMicro reconstructions yield 92% recall of gold-standard tracings vs. 87% for COIN at matched bitrate, suggesting structural preservation translates to downstream task performance.

Decoding Efficiency and Interactive Exploration. The 19× speedup over INR methods is critical for practical deployment. Interactive visualization requires >30 FPS rendering of 512^3 subvolumes, demanding $\sim$4M voxels reconstructed in <33 ms. VolMicro achieves this (51 MVox/s $\Rightarrow$ 4M voxels in 78 ms) while COIN requires 1.5 seconds—the difference between interactive and batch-mode workflows.

Random-access decoding is another key advantage. Querying a 128^3 region-of-interest requires evaluating only the $\sim$500 primitives within that bounding box (via spatial indexing), taking <10 ms. COIN must either: (1) reconstruct the full volume then crop ($\sim$20 seconds), or (2) evaluate the MLP on 2M coordinates ($\sim$700 ms). This 70× advantage enables real-time navigation and analysis.

Occupancy Gating: Sparsity Exploitation. TOPS-Gate provides 6× training speedup by restricting computation to 16% of voxels. The gating architecture is intentionally simple (12K parameters, 1000-iteration pre-training) to avoid overfitting and minimize overhead. More sophisticated gates (e.g., 3D U-Net) could improve boundary precision but would require longer pre-training and larger storage.

An alternative approach is hard thresholding the input volume to define $\mathcal{O}$ without learning. However, fixed thresholds fail on heterogeneous data: low-SNR samples require lower thresholds (catching dim structures) while high-SNR samples need higher thresholds (excluding noise). TOPS-Gate adapts to local statistics through its training on binary occupancy labels, achieving robustness across imaging conditions.

Comparison to 3D Gaussian Splatting. Our method adapts Gaussian primitives from radiance fields [22] to scalar intensity volumes, requiring significant modifications:

1. **Scalar vs. RGB**: We model intensity $w_i \in \mathbb{R}^+$ rather than color, eliminating spherical harmonics coefficients and reducing storage.
2. **Occupancy-aware**: Sparse gating exploits background sparsity, irrelevant for view-synthesis where rays traverse both occupied and empty space.
3. **Edge-aware loss**: Boundary preservation is critical for morphology, while rendering prioritizes view-dependent smoothness.
4. **KNN evaluation**: We reconstruct dense voxel grids via KNN rather than splatting to sparse pixel locations, requiring different GPU kernels.

Limitations and Failure Cases.

1. **Highly anisotropic structures**: Neurites with extreme aspect ratios (>50:1 length:diameter) are under-represented by isotropic or mildly anisotropic Gaussians. Tubular primitives [40] or curve-based representations [41] may be more suitable.
2. **Low sparsity**: Tissue with dense neuropil or uniform labeling (>50% occupancy) reduces TOPS-Gate benefits. For such volumes, sparse gating provides <2× speedup.
3. **Multi-channel data**: Extension to multi-color microscopy (e.g., 3-channel RGB fluorescence) requires either vector-valued amplitudes $\mathbf{w}_i \in \mathbb{R}^3$ (increasing storage by 3×) or shared geometry with channel-specific intensities (assuming channels have similar spatial support, often violated in practice).
4. **Encoding time**: Per-volume optimization (7 minutes) is slower than feed-forward amortized encoders (<1 second). However, microscopy archival is an offline process where encode time is amortized over long-term storage savings.

Failure cases include: (1) extreme noise (SNR <3), where TOPS-Gate mis-classifies noise as signal, and (2) extremely sparse structures (<5% occupancy), where few training points lead to under-fitting.

6 Limitations and Future Work

Current Limitations.

1. **Per-volume optimization**: 7-minute encoding is slower than amortized methods. Learning a feed-forward encoder mapping volumes to Gaussian parameters remains future work, though preliminary experiments with hypernetworks show promise.
2. **Geometric primitives**: Axis-aligned Gaussians under-represent highly anisotropic structures (aspect ratio >50:1). Tubular or curve primitives may better capture neurites and vasculature.
3. **Multi-channel extension**: Current formulation handles single-channel intensity. Multi-color fluorescence requires vector-valued amplitudes or channel-specific primitive sets, increasing storage proportionally.
4. **Single-dataset evaluation**: While we validate on 5 samples from BigNeuron, broader evaluation across tissue types (brain, heart, kidney), pathologies (cancer, neurodegeneration), and modalities (electron microscopy, lattice light-sheet) is needed.
5. **Quantitative morphology validation**: We assess compression quality via PSNR/SSIM/LPIPS but do not exhaustively validate preservation of morphological features (neurite diameter, branch angles, connectivity) critical for biology. Correlation with automated tracing accuracy is promising but requires larger-scale studies.

Future Directions.

1. **Amortized encoding**: Train a feed-forward encoder $E_\phi : V \to \Theta$ mapping volumes to Gaussian parameters via hypernetworks or transformer architectures. This would reduce encoding to <1 second while maintaining quality.
2. **Hierarchical primitives**: Multi-scale Gaussian pyramids for progressive transmission and level-of-detail rendering. Coarse primitives transmit first for rapid preview, refined by fine-scale primitives on demand.
3. **Morphology-aware primitives**: Extend beyond isotropic/anisotropic Gaussians to tubular splines [41] or sheet-like primitives [40] specialized for biological structures.
4. **Joint compression-analysis**: Perform segmentation, tracing, or quantification directly on primitives without full decompression. Primitive centers $\{\boldsymbol{\mu}_i\}$ approximate neurite skeletons; clustering via covariance could identify structures.
5. **Temporal extension**: Exploit temporal coherence in 4D microscopy (time-lapse, calcium imaging) via primitive trajectories or shared geometry with time-varying intensities.
6. **Learned entropy coding**: Current storage uses raw float16 parameters. Applying learned entropy models [26] to primitive parameters could reduce bitrate by additional 2–3×.
7. **Hybrid representations**: Combine Gaussians for sparse structures with voxel grids for dense regions (e.g., cell bodies), allocating representation to data characteristics.

7 Conclusion

We presented VolMicro, a framework for extreme microscopy volume compression using sparse 3D Gaussian mixtures. By leveraging explicit anisotropic primitives, sparse occupancy gating, and edge-aware optimization, VolMicro achieves 231× compression with competitive perceptual quality (SSIM 0.932, LPIPS 0.278). The key advantage over neural implicit methods is 19× faster decompression (51 vs. 2.7 MVox/s), enabling interactive visualization and analysis of large-scale biological imaging data.

Beyond storage efficiency, the structured explicit representation opens pathways for compression-aware analysis: primitive parameters directly encode geometric features (position, orientation, scale) amenable to morphological quantification without voxel-space decompression. As microscopy datasets continue to grow, such hybrid compression-analysis frameworks will become increasingly critical for extracting biological insights from massive imaging archives.

Acknowledgments

We thank the BigNeuron community for providing benchmark datasets and gold-standard reconstructions. This work was supported by [funding sources]. Compute resources were provided by [institution/grant]. We acknowledge valuable discussions with [collaborators] on microscopy data compression and biological image analysis.

References

[1] J. Huisken, J. Swoger, F. Del Bene, J. Wittbrodt, and E. H. K. Stelzer. Optical sectioning deep inside live embryos by selective plane illumination microscopy. *Science*, 305(5686):1007–1009, 2004.

[2] P. A. Santi. Light sheet fluorescence microscopy: a review. *Journal of Histochemistry & Cytochemistry*, 59(2):129–138, 2011.

[3] F. Chen, P. W. Tillberg, and E. S. Boyden. Expansion microscopy. *Science*, 347(6221):543–548, 2015.

[4] X. Sunney, M. McDole, and P. J. Keller. Whole-brain imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, 18(9):1009–1019, 2021.

[5] M. B. Ahrens, M. B. Orger, D. N. Robson, J. M. Li, and P. J. Keller. Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, 10(5):413–420, 2013.

[6] D. S. Taubman and M. W. Marcellin. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2001.

[7] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12):1649–1668, 2012.

[8] P. Lindstrom. Fixed-rate compressed floating-point arrays. *IEEE Trans. Vis. Comput. Graphics*, 20(12):2674–2683, 2014.

[9] S. Di and F. Cappello. Fast error-bounded lossy HPC data compression with SZ. In *IEEE IPDPS*, pages 730–739, 2016.

[10] F. Zhao, S. Di, H. Guo, T. Peterka, and F. Cappello. Lossy compression for scientific data using transformation-based prediction. In *IEEE IPDPS*, pages 403–413, 2019.

[11] E. Dupont, A. Goliński, M. Alizadeh, Y. W. Teh, and A. Doucet. COIN: COmpression with implicit neural representations. *arXiv:2103.03123*, 2021.

[12] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020.

[13] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421, 2020.

[14] Y. Strümpler, J. Postels, R. Yang, L. Van Gool, and F. Tombari. Implicit neural representations for image compression. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(8):9645–9656, 2022.

[15] I. Mehta, M. Gharbi, C. Barnes, E. Shechtman, R. Ramamoorthi, and M. Chandraker. Modulated periodic activations for generalizable local functional representations. In *ICCV*, pages 14214–14223, 2021.

[16] R. Yang, Y. Guo, X. Tong, T. Deng, and Y. Guo. Neural compression for volumetric medical image data. *IEEE Trans. Med. Imaging*, 42(5):1377–1389, 2023.

[17] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graphics*, 41(4):102, 2022.

[18] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *CVPR*, pages 11358–11367, 2021.

[19] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao. DVC: An end-to-end deep video compression framework. In *CVPR*, pages 11006–11015, 2019.

[20] H. Liu, T. Chen, P. Guo, Q. Shen, X. Cao, Y. Wang, and Z. Ma. Non-local attention optimized deep image compression. *arXiv:2001.00357*, 2020.

[21] Z. Hu, Z. Lu, G. Lu, and D. Xu. FVC: A new framework towards deep video compression in feature space. In *CVPR*, pages 1502–1511, 2021.

[22] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graphics*, 42(4):139, 2023.

[23] S. Niedermayr, J. Stumpfegger, and R. Westermann. Compressed 3D Gaussian splatting for accelerated novel view synthesis. *arXiv:2401.02436*, 2024.

[24] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger. Mip-splatting: Alias-free 3D Gaussian splatting. In *CVPR*, 2024.

[25] J. Luiten, G. Kopanas, B. Leibe, and D. Deva Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. *arXiv:2308.09713*, 2023.

[26] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. In *ICLR*, 2017.

[27] D. Minnen, J. Ballé, and G. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *NeurIPS*, pages 10771–10780, 2018.

[28] T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Trans. Image Process.*, 30:3179–3191, 2021.

[29] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, pages 4460–4470, 2019.

[30] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger. Convolutional occupancy networks. In *ECCV*, pages 523–540, 2020.

[31] S. Laine and T. Karras. Efficient sparse voxel octrees. *IEEE Trans. Vis. Comput. Graphics*, 17(8):1048–1059, 2010.

[32] D. Nehab and P. V. Sander. Generalized adaptive tessellation for subdivision surfaces. *ACM Trans. Graphics*, 30(4):93, 2011.

[33] M. Tancik *et al.* Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, 2020.

[34] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[36] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[37] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data*, 7(3):535–547, 2019.

[38] H. Peng, M. Hawrylycz, J. Roskams, S. Hill, N. Spruston, E. Meijering, and G. A. Ascoli. BigNeuron: Large-scale 3D neuron reconstruction from optical microscopy images. *Neuron*, 87(2):252–256, 2015.

[39] H. Peng, A. Bria, Z. Zhou, G. Iannello, and F. Long. Extensible visualization and analysis for multidimensional images using Vaa3D. *Nature Protocols*, 9(1):193–208, 2014.

[40] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symp. Geometry Processing*, volume 4, pages 109–116, 2007.

[41] P. Wang, Y. Liu, Z. Chen, L. Liu, Z. Liu, T. Komura, C. Theobalt, and W. Wang. Neural implicit surfaces with structured latent codes. In *CVPR*, pages 9288–9297, 2021.

A Detailed Training Configuration

B Additional Ablation: KNN Neighbors

Table 8 shows the quality-speed trade-off for varying K. Exact evaluation ($K = N =$ 20,693) provides negligible improvement (< 0.05 dB) over $K = 32$ but requires 150× longer, demonstrating the effectiveness of KNN approximation.

C Broader Impact

This work addresses critical storage and transmission challenges in biological imaging, with potential positive and negative societal impacts:

Positive Impacts.

- **Democratization**: Extreme compression (231×) reduces storage costs from \$2,100/TB (cloud archival) to \$9.09/TB compressed, making large-scale imaging accessible to resource-limited institutions.
- **Energy efficiency**: Data centers consume ∼200 TWh/year globally; reducing microscopy archival by 200× saves ∼0.4 TWh/year (equivalent to powering 37,000 homes), reducing carbon footprint.
- **Collaboration**: Fast decoding (51 MVox/s) enables real-time remote visualization, facilitating global scientific collaboration without requiring local high-performance computing.
- **Preservation**: Compact storage enables long-term archival of rare or historical samples, supporting reproducibility and meta-analyses.

Potential Negative Impacts.

- **Diagnostic risk**: Lossy compression in clinical settings could obscure pathological features. We emphasize VolMicro is intended for *research microscopy*, not diagnostic medical imaging, where lossless or near-lossless compression is required.
- **Replication barrier**: Per-volume optimization requires GPU access, potentially creating barriers for researchers without compute resources. Development of amortized encoders would mitigate this.
- **Data permanence**: Compressed archives depend on decoding software availability. We commit to open-source release and long-term maintenance to prevent future inaccessibility.

Ethical Considerations. We do not foresee direct misuse potential (e.g., surveillance, disinformation) as microscopy compression has limited dual-use applications. However, biological imaging data may contain sensitive information (e.g., human tissue samples) requiring compliance with privacy regulations (GDPR, HIPAA). Compression does not inherently address privacy; secure storage and access controls remain necessary.

Recommendations.

1. Validate reconstruction quality on held-out samples before archiving original data.
2. Maintain uncompressed copies of diagnostically-relevant datasets.
3. Develop standardized quality metrics for compression-specific applications in microscopy, going beyond PSNR/SSIM to include morphological features (connectivity, diameter, topology).
4. Establish community guidelines for acceptable compression levels for different biological applications (archival, visualization, quantitative analysis).
5. Provide amortized encoding tools to reduce GPU requirements for resource-limited labs.

figures/qualitative_comparison.pdf

Figure 2: Visual comparison at matched bitrate (0.073 bpp). Maximum intensity projection (MIP) along depth axis with zoomed insets highlighting neurite boundaries. VolMicro better preserves fine dendrite edges and bifurcation topology compared to COIN's smooth interpolation. JPEG2000 and HEVC show blocking artifacts at extreme compression. Color scale: black (background) to yellow (high intensity).

Table 7: Complete VolMicro training configuration and hyperparameters.

Parameter	**Value**
Dataset	
Volume dimensions	$100 \times 647 \times 813$ (D $\times$ H $\times$ W)
Voxel spacing	$0.54 \times 0.54 \times 0.54\,\mu$m (isotropic)
Total voxels	52,601,100
Original size	105.2 MB (uint16, $2 \times$ 52,601,100 bytes)
Dynamic range	$[0, 65535]$ (16-bit unsigned)
Mean intensity	1247.3 (1.9% of max)
Std. dev. intensity	3842.1
TOPS-Gate	
MLP architecture	[64, 64, 64, 1] with ReLU
Fourier encoding bands	$L = 4$ ($k \in \{0, 1, 2, 3\}$)
Total parameters	12,353
Pre-training iterations	1000
Pre-training LR	0.001 (Adam)
Occupancy threshold	$\tau_{\text{occ}} = 0.01$ (657 intensity units)
Gating threshold	$\tau = 0.5$
Gated voxels	8,452,906 (16.1%)
Sampling ratio	$\rho = 0.30$
Batch size (effective)	2,535,872 voxels/iteration
Gaussian Model	
Initial count	$N_0 = 5000$
Final count	$N = 20{,}693$
Peak count (epoch 3400)	$N_{\max} = 28{,}547$
Parameters per Gaussian	11 (3 pos. + 3 scale + 4 quat. + 1 intensity)
Storage per Gaussian	22 bytes (float16)
Compressed size	455,246 bytes = 0.46 MB
Compression ratio	$105.2/0.46 = 231\times$
Bitrate	$8 \times 455{,}246/52{,}601{,}100 = 0.073$ bpp
Training	
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$)
Learning rate	0.01 (constant, no schedule)
Total epochs	10,000
Training time	7.1 minutes (RTX 3080 10GB, PyTorch 2.1)
KNN neighbors	$K = 32$
Mahalanobis cutoff	3σ (contributions < 0.01 zeroed)
Loss Weights	
Edge boost factor	$w_{\text{edge}} = 2.0$
Base weight	$w_{\text{base}} = 1.0$
Sparsity penalty	$\lambda_s = 0.001$
Overlap penalty	$\lambda_o = 0.0$ (disabled)
Scale regularization	$\lambda_r = 0.001$
Scale bounds	$s_{\min} = 10^{-4}$, $s_{\max} = 0.1$
Densification	
Start epoch	500
Stop epoch	3500
Interval	100 epochs
Gradient threshold	$\tau_{\text{grad}} = 10^{-4}$
Size threshold	$\tau_{\text{size}} = 0.02$ (2% of volume)
Pruning threshold	$\tau_{\text{prune}} = 10^{-3}$ (intensity)
Gradient EMA decay	$\alpha = 0.95$
Clone perturbation	$\mathcal{N}(0, 0.01\,\mathbf{I})$
Split factor	0.8 (scale reduction)
Split offset	$\pm 0.25\,\mathbf{s}_i$
Hardware & Software	

Table 8: KNN ablation. Trade-off between reconstruction quality and decode speed. Exact evaluation ($K = N$) provides upper bound; $K = 32$ achieves < 0.1 dB loss with $150\times$ speedup.

KNN (K)	**PSNR (dB)**	**SSIM**	**Decode time (ms)**	**Throughput (MVox/s)**
$K = 8$	35.81	0.921	743	70.8
$K = 16$	36.32	0.928	871	60.4
$K = 32$	36.58	0.932	1,032	51.0
$K = 64$	36.61	0.933	1,458	36.1
$K = 128$	36.62	0.933	2,981	17.6
Exact ($K = N$)	36.63	0.933	154,217	0.34