Towards Real-Time Full-Waveform LiDAR Transformers via Intensity-Guided Token Reduction and Physics-Aware Augmentation

Kotaro Oishi[*]
koutarou00140805@keio.jp
Keio University

Kazuma Ikeda[*]
kazu2080@keio.jp
Keio University

Ryosei Hara
ryosei_hara@keio.jp
Keio University

Ryo Yoshida
ryo-yoshida@keio.jp
Keio University

Mariko Isogawa[†]
mariko.isogawa@keio.jp
Keio University

Kentaro Yoshioka[†]
kyoshioka47@keio.jp
Keio University

Abstract

LiDAR is a critical sensor for autonomous driving and robotics, yet it remains vulnerable to adverse conditions such as fog, rain, transparent objects, and low-reflectance surfaces. Full-waveform LiDAR (FWL) addresses these limitations by capturing the complete return waveform as a histogram, preserving rich reflection characteristics including multi-path echoes and weak signals. While recent Transformer-based approaches have demonstrated promise on FWL data, two fundamental bottlenecks remain: high computational cost from redundant tokens in low-intensity background regions, and limited generalization due to scarce annotated training data. We address both bottlenecks by exploiting the physical structure inherent to FWL data. First, we propose FWL-ToPM, a token pruning and merging mechanism that leverages intensity distributions to selectively reduce tokens while retaining critical waveform content. Second, we introduce FWLAug, a data augmentation framework designed as a regularization method that preserves waveform structure, retaining the Time-of-Flight waveform of each ray and the frustum geometry, to improve generalization to unseen scenes. Evaluated on the Ghost-FWL benchmark, FWL-ToPM achieves over 8$\times$ speedup at real-time throughput alongside a nearly 9-point F1 gain over the prior state of the art, establishing the first real-time FWL Transformer that simultaneously improves accuracy. FWLAug further improves performance on both the standard benchmark and an out-of-distribution test set of unseen scenes. Together, these methods push FWL-based perception toward practical deployment. The code and the OOD dataset are publicly available and can be accessed via the project page: https://keio-csg.github.io/FWL-ToPM

1 Introduction

LiDAR is a key sensor for autonomous driving, robotics, and mapping, measuring 3D geometry via Time-of-Flight (ToF) of reflected laser pulses. Conventional LiDAR, however, records only the peak intensity of each pulse, and therefore cannot capture complex reflection characteristics such as multi-path returns from glass or backscatter from fog and rain. These limitations cause spurious point

[*]These authors contributed equally and share first authorship. Either author may list their name first in CVs and other academic materials.

[†]These authors are co-corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

generation (ghosts and noise), degrading object detection and SLAM performance and introducing safety risks.

Full-waveform LiDAR (FWL) addresses these limitations by recording the complete temporal intensity waveform of each pulse as a histogram, thereby preserving rich reflection characteristics including multi-path echoes and weak-signal components. Transformer-based FWL models (FWLTs) exploit this richer representation to enable tasks fundamentally beyond conventional LiDAR, including ghost detection [10] and fog-robust point cloud reconstruction [31]. Despite this promise, two fundamental challenges remain unresolved for practical FWLT deployment.

The first challenge is real-time inference. Because FWL records a complete histogram for every pixel, its data volume is far larger than that of a point cloud, yet the vast majority consists of low-intensity background regions carrying no object information. This redundancy severely inflates computational cost, limiting existing FWLTs to inference rates well below real-time requirements even on high-end GPUs [10]. Existing token reduction methods [2, 30], designed for camera images and videos, assume informative content is broadly distributed across 2D space and thus indiscriminately discard the sparse high-intensity tokens that encode object-relevant reflections in FWL data. Exploiting FWL's physical structure for token reduction requires a dedicated mechanism.

The second challenge is generalization. FWL observations vary substantially across acquisition environments, object configurations, and reflectance conditions, yet publicly available data remains scarce. Minority classes such as glass and water lack sufficient samples to learn their characteristic reflection signatures, pushing models to overfit dominant classes. Standard image and point cloud augmentations ignore FWL's ToF waveform structure and frustum geometry and thus distort the waveform of each ray. A dedicated augmentation framework that respects these constraints is essential.

We address both challenges with two methods designed around the physical structure of FWL data. FWL-ToPM reduces computational cost by pruning redundant background tokens based on intensity distributions and merging the remainder, while preserving critical waveform content. FWLAug expands training diversity while respecting FWL's ToF waveform structure and frustum geometry. Together, these methods advance FWLTs toward practical deployment by simultaneously achieving real-time operation and improved generalization to unseen environments. Our contributions are as follows:

- We propose **FWL-ToPM**, a token pruning and merging mechanism tailored to FWL data. By leveraging intensity distributions to prune low-intensity background tokens and merge the remainder, FWL-ToPM reduces computational cost while retaining critical waveform content. It achieves an 8$\times$ speedup (53.2 FPS) and an approximately 9-point F1 improvement (0.503$\rightarrow$0.592), establishing the first real-time FWLT.
- We find that, unlike image-based Transformers where aggressive pruning destabilizes training [16, 30], even pruning up to 70% of tokens can consistently improve FWLT accuracy. Removing low-intensity background tokens allows the model to focus on object-relevant signals that are otherwise diluted by uninformative waveform regions.
- We propose **FWLAug**, a data augmentation framework tailored to FWL data. FWLAug expands training diversity while preserving the ToF waveform structure and frustum geometry that standard image and point cloud augmentations violate.
- We construct an out-of-distribution (OOD) test set captured entirely outside the Ghost-FWL [10] recording locations, on which FWLAug achieves the best F1-mean among all compared configurations.

2 Related Work

Deep Learning Methods for Full-waveform LiDAR. Conventional LiDAR internally selects several high-intensity peak positions for each laser pulse and stores data in a point cloud format ($x \times y \times z$), recording only the 3D coordinates of detected points. In contrast, Full-waveform LiDAR (FWL) captures complete temporal light intensity information at each pixel of the receiver, which is stored in the FWL data format ($u \times v \times t$), where u and v denote the azimuth and elevation of each ray and t denotes the ToF bin. From the complete histogram at each pixel, arbitrary peaks can be selected and converted into 3D coordinates, enabling the use of all waveform information and providing richer

representations. In recent years, various FWL-based studies have attracted attention, taking advantage of its characteristics, such as Classification [5, 17, 32–34, 45], 3D Object Detection [6], Jamming Recovery [41], Fog Removal [31, 38], Ghost Removal [10].

While early FWL-based methods mainly used CNN-based networks, recent advancements in Transformers in the vision field [7, 8, 28] have led to the adoption of Transformer-based networks [10, 31]. Scheuble et al. [31] proposed a Neural Digital Signal Processor (Neural DSP) using U-Net spatio-temporal transformers with FWL data as input, enabling robust point cloud reconstruction in low SNR environments such as fog. However, the Neural DSP partitions tokens only along the time axis while treating spatial dimensions as 1×1, causing computational cost to scale poorly with spatial resolution and limiting throughput to 2 FPS. Ikeda et al. [10] proposed Ghost-FWL, a large-scale annotated FWL dataset for ghost detection and FWL-MAE, a self-supervised pre-training method for FWL data. FWL-MAE treats the time axis as a single tube and applies a 16×16 patch division in the spatial direction to reduce computational complexity. Despite this, FWL-MAE reaches only 6 FPS, falling well short of the 10 Hz required for real-time operation.

Computational cost thus remains a fundamental bottleneck for FWLTs. We address this by introducing intensity-guided token pruning and merging tailored to FWL's sparse structure, targeting the ghost detection task in Ghost-FWL [10] as a representative real-time perception challenge.

Token Pruning and Merging. LiDAR-based systems require real-time performance, whereas deep learning–based methods tend to have high computational costs and low throughput. There is a trade-off between the accuracy and inference time of deep learning models. In particular, for Transformers, efficient vision transformers [4, 20, 28] and token reduction methods such as pruning [9, 13, 14, 19, 30] and merging [1, 2, 15, 22, 23, 27, 29, 37] have been widely studied to reduce computational cost while maintaining accuracy. Recently, methods that combine pruning and merging have also been proposed [3, 12, 24]. However, most of these studies focus on token reduction for images or videos, and their application to FWL data remains unexplored. These methods assume that significant information is widely distributed in 2D space, whereas FWL data has a sparse structure in 3D space where peaks appear at only a few locations and the rest are almost zero, making them unsuitable for FWL data.

In FWL data, regions around peaks contain more important information, while the majority of the data consists of intensities close to zero, making regions outside the peaks redundant and increasing computational cost. Therefore, we propose FWL-ToPM, a token pruning and merging method tailored to FWL data that enables faster inference while maintaining high classification performance.

Augmentation for Full-waveform LiDAR. Augmentation is widely used to improve generalization by introducing diversity into limited training data, with extensive work spanning image recognition and 3D point clouds. For images, geometric and intensity transforms such as CutMix [42] have been proposed, while 3D point cloud methods account for the geometric structure of the data. Mix3D [26] generates new training samples by blending multiple 3D scenes, and PolarMix [40] exploits the polar coordinate structure of LiDAR scans for rotation and region swapping.

No augmentation method has been designed specifically for FWL data. FWL data retains a continuous temporal waveform along each ray direction and forms a frustum-shaped observation structure with the LiDAR origin as its apex. Applying existing image or point cloud augmentations directly disrupts the ToF waveform structure and frustum geometry, causing unnatural distortions in object shape and distance. Augmentations proposed for ToF-based transient imaging [11, 35] are limited to basic operations such as noise addition and rotation, which are insufficient to improve generalization for FWL data.

3 Method

We propose a fast and robust framework for estimating the class logits $\boldsymbol{y} \in \mathbb{R}^{H\times W\times T\times C}$ of each voxel from FWL data $\boldsymbol{V} \in \mathbb{R}^{H\times W\times T}$, even for scenes that are not included in the training data. Here, H, W, and T denote the spatial and temporal resolutions, respectively; H and W correspond to the elevation (v) and azimuth (u) axes of the ($u \times v \times t$) format, and T to the ToF (t) axis, and C denotes the number of target classes, such as *Noise*, *Object*, *Glass*, and *Ghost*.

The proposed framework consists of FWL-ToPM, which enables fast and accurate FWL recognition, and FWLAug, which improves generalization performance for unseen scenes. FWL-ToPM focuses

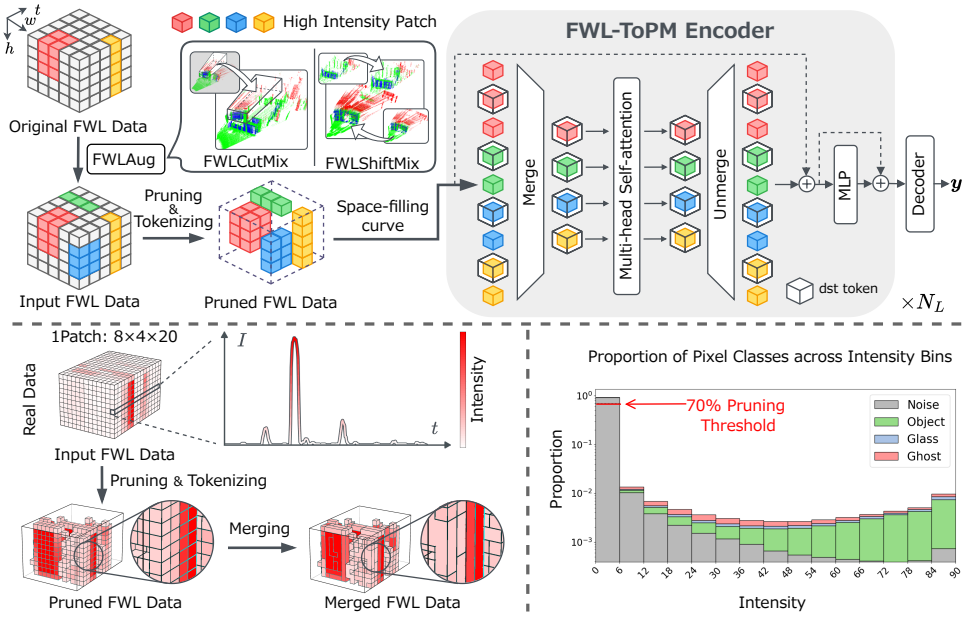

Figure 1: Overview of the proposed framework. **(Top)** During training, FWLAug augments FWL samples while preserving ToF waveform structure and frustum geometry. High-intensity patches are selected by pruning, spatially reordered along a space-filling curve, and processed by the **FWL-ToPM** Encoder, which merges redundant tokens before self-attention and unmerges afterward across L Transformer blocks. **(Bottom Left)** Pruning removes low-intensity background tokens, and merging aggregates the remainder, concentrating computation on object-relevant waveform content. **(Bottom Right)** Proportion of pixel classes across intensity bins in FWL data. Most regions in FWL data correspond to noisy background, and our pruning strategy removes these dominant low-intensity regions.

on the spatial sparsity of signal intensity specific to FWL data and achieves efficient feature extraction by reducing redundant tokens. FWLAug generates diverse training samples of waveform structures, object arrangements, and minority classes while preserving the ToF-direction waveform structure specific to FWL data and the frustum geometry based on the LiDAR viewpoint. This increases the diversity of the training data and enables the model to learn more robust representations for unseen scenes. The details of FWL-ToPM and FWLAug are described in §3.1 and §3.2, respectively.

3.1 FWL-ToPM

Recently, the effectiveness of Transformer architectures has been reported for the analysis of FWL data [10, 31]. However, existing methods are largely based on Vision Transformers (ViTs), which were originally designed for images, and therefore do not sufficiently account for the characteristics of FWL data. Unlike images, FWL data are characterized by spatially sparse peaks with high signal intensity, while most regions are occupied by low-intensity signals that contribute little to the classification task. As a result, applying uniform ViT-based tokenization generates many uninformative tokens, which can obscure important high-intensity regions. Consequently, a large portion of computational resources is spent on ineffective tokens. Moreover, since low-intensity tokens dominate the representation, an imbalance in feature representations occurs, leading to degraded recognition performance. These issues cannot be resolved simply by applying token merging methods for 2D images [1, 2], which are designed under the assumption that information is relatively homogeneous across pixels.

To address these issues, we propose FWL-ToPM, which explicitly exploits the signal characteristics of FWL data to optimize token representations specifically for FWL Transformers. Specifically, FWL-ToPM combines (1) signal-intensity-based token pruning and (2) spatial-structure-aware token merging to achieve both computational efficiency and recognition performance.

Model Architecture. FWL-ToPM uses a Vision Transformer (ViT) [7] as the backbone for voxel-wise classification. However, the main contribution of our method lies in its token selection and aggregation mechanisms, which are applicable to any token-based models. First, the input frame $\boldsymbol{V} \in \mathbb{R}^{H \times W \times T}$ is divided into 3D patches of patch size $\boldsymbol{p} = (p_H, p_W, p_T)$. Then, each patch is projected into an E-dimensional patch token by a 3D convolution with both kernel size and stride set to $\boldsymbol{p}$. A 3D positional embedding corresponding to the position of each patch is added to these tokens, yielding the token sequence used as the input to the encoder. Next, these tokens are input into a Transformer encoder composed of N_L blocks. Each Transformer block consists of a multi-head self-attention layer with N_h attention heads and an MLP. When pruning and merging are not introduced, the encoder processes all patch tokens as they are, and E-dimensional features are extracted for each patch at the final layer. Finally, a single-layer linear decoder is applied to each token to predict class logits for the voxels within the corresponding patch. These logits are rearranged according to the original patch layout to obtain the final output $\boldsymbol{y} \in \mathbb{R}^{H \times W \times T \times C}$. The following describes the details of the proposed token pruning and token merging.

Intensity-Based Pruning. In FWL data, peaks in the LiDAR histogram with high signal intensity are sparsely distributed, while most low signal intensity regions correspond to noise that does not contribute to the classification task. Therefore, our method evaluates the importance of each patch based on its maximum signal intensity and performs token pruning to remove low signal intensity patches in advance. Specifically, the procedure is as follows: First, we store the maximum signal intensity within the i-th patch as m_i. Next, all patches are sorted in ascending order according to m_i, and the bottom ρ_{low} proportion of patches are assigned to $\boldsymbol{P}_{\mathrm{low}}$ as removal targets, while the remaining patches are assigned to $\boldsymbol{P}_{\mathrm{high}}$. Only the patches belonging to $\boldsymbol{P}_{\mathrm{high}}$ are tokenized and fed into the encoder. Since the removed patches do not pass through the encoder, the decoder cannot predict class logits for the corresponding patch regions. Therefore, during training, the loss is computed by excluding the losses for the pruned patch regions, while during inference, all voxels in those patch regions are predicted as the noise label.

Our proposed pruning method consists only of patch selection and modification of the token pathway. It does not modify the weights of the patch embedding, encoder, and decoder. Therefore, this pruning method can be applied to a pretrained model at inference time without additional training.

Sparse-Aware Token Merging. Although pruning can remove most highly redundant tokens, redundant tokens may still remain among the patches in $\boldsymbol{P}_{\mathrm{high}}$ after pruning. Therefore, we apply token merging to the tokens in $\boldsymbol{P}_{\mathrm{high}}$ to further reduce the number of tokens used in attention computation. Token merging is a method that selects representative destination tokens (dst tokens) from a token sequence and aggregates the remaining source tokens (src tokens) into the dst tokens based on feature similarity. To make token merging work effectively, it is necessary to appropriately sample dst tokens in a way that reflects the characteristics of the data.

In the case of FWL data, spatially neighboring patches are likely to contain similar waveform information originating from the same object or reflection region. Therefore, the dst tokens used for merging should be distributed across the original 3D space rather than being biased toward specific local regions. A key challenge in applying token merging after pruning is that $\boldsymbol{P}_{\mathrm{high}}$ forms a *sparse and irregular* subset of the original 3D patch grid. Existing dst token selection methods divide the patch grid into fixed-size regions and sample one dst token per region [1, 36, 43, 44]; these methods assume a dense, uniform token distribution and fail when applied to a post-pruning sparse set, as regions may contain no tokens at all.

We resolve this by serializing $\boldsymbol{P}_{\mathrm{high}}$ along a space-filling curve prior to dst selection. A space-filling curve maps the 3D patch grid to a 1D sequence such that patches that are close in 3D space are placed near each other in the sequence. Using this property, we compute the position along the curve of each patch token in $\boldsymbol{P}_{\mathrm{high}}$ from its coordinates in the 3D patch grid before pruning, and reorder the tokens accordingly. This constructs a token sequence that preserves spatial relationships even when gaps are introduced into the token sequence by pruning. We then select dst tokens at equal intervals from the serialized token sequence according to the dst token ratio ρ_d, and treat the remaining tokens as src tokens. Next, we compute the cosine similarity between the features of each src token and each dst token, and assign each src token to the dst token with the highest similarity. Then, based on the merge ratio ρ_m, src tokens with high similarity are merged by averaging them with their corresponding dst tokens.

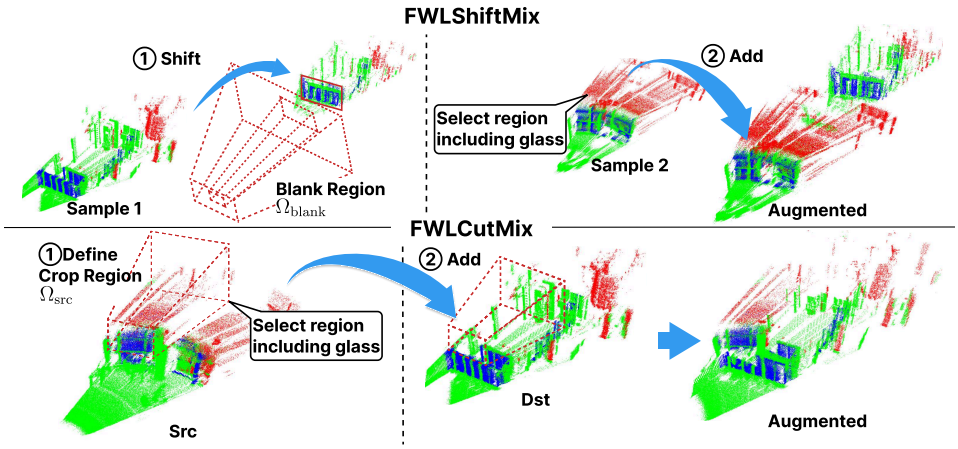

Figure 2: Illustration of FWLAug. **FWLShiftMix** (top): an object in Sample 1 is shifted away from the LiDAR, creating a blank region Ω_{blank}. A glass-containing region Ω_{src} is then selected from Sample 2 and pasted into the blank, producing an augmented scene in which the waveform structure of each ray is preserved. **FWLCutMix** (bottom): a crop region Ω_{src} containing minority class (glass) is selected from a source frame and pasted into a destination frame at a random location. Both methods prioritize glass-containing regions to mitigate class imbalance.

This serialization does not depend on a particular curve, as long as the curve preserves spatial proximity in the 1D sequence. In our experiments, we adopt the Morton order [25], which is obtained by interleaving the coordinate bits of the three axes for each patch. Serializing 3D data with space-filling curves has also been adopted in Point Transformer V3 (PTv3) [39], which orders point clouds along Z-order and Hilbert curves to partition them into patches for local attention. Our approach shares the same principle of preserving spatial proximity in a 1D sequence, but differs in its input and purpose: we serialize the sparse and irregular set of 3D waveform patches that remain after intensity-based pruning, in order to select spatially distributed dst tokens for merging.

Since token merging is performed to reduce the computational cost of attention, after the attention computation, the number of tokens must be restored by reassigning the merged destination features to each token, thereby maintaining consistency with the residual connections. Therefore, after the attention computation, for each merged src token, we copy the feature of the dst token into which it was merged and redistribute it back to the original token position. Our merging method also does not change the parameter weights, and can be applied to pre-trained models without additional training.

3.2 FWLAug

We propose FWLAug, an augmentation framework designed around the physical structure of FWL data (Figure 2). In FWL data $\boldsymbol{V} \in \mathbb{R}^{H\times W\times T}$, each per-pixel histogram $\boldsymbol{Q}_{h,w}$ encodes the temporal intensity response of a laser pulse, preserving time-of-flight delays, reflectance, and multi-path waveform characteristics. Since each $\boldsymbol{Q}_{h,w}$ corresponds to a ray from the LiDAR origin, the full FWL frame forms a frustum-shaped observation structure. These properties impose two requirements on any FWL augmentation: (1) the T-axis waveform structure of each $\boldsymbol{Q}_{h,w}$ must be preserved, and (2) consistency with frustum geometry must be maintained. While FWLAug is not intended to serve as a strict optical simulator that reproduces the complete light transport of the augmented scene, it is strategically designed as an effective regularization method. By intentionally focusing on preserving the ray-wise waveform structure, it efficiently increases training diversity and the representation of minority classes while satisfying the two requirements above.

FWLAug comprises two augmentations that address independent challenges. FWLCutMix alleviates minority-class under-representation by prioritizing glass-containing regions during patch pasting, while FWLShiftMix expands diversity in observation distance and position by translating objects in 3D space. Used together, they improve generalization to unseen scenes while preserving in-domain accuracy.

FWLShiftMix. Shifting objects away from the LiDAR increases distance diversity but creates blank regions due to the compression effect. Zero-filling these regions introduces unrealistic data dis-

tributions absent from real FWL scenes, causing the model to learn spurious artifacts. FWLShiftMix instead fills the blank regions with FWL data from another scene, producing synthetic samples that preserve the waveform structure of each ray.

Concretely, FWLShiftMix proceeds in three steps (Figure 2). First, each pixel's peak position $t^*_{h,w} = \arg\max_t \boldsymbol{Q}_{h,w}(t)$ is converted to a 3D point $\boldsymbol{r}_{h,w}$ via ToF geometry. Second, we translate each point along its LiDAR ray direction. Let $\boldsymbol{u}_{h,w} = \boldsymbol{r}_{h,w}/\|\boldsymbol{r}_{h,w}\|_2$ denote the unit ray direction and let $\Delta d \sim \mathcal{U}(-3, 15)$ denote the sampled radial displacement in meters. The translated point is computed as $\boldsymbol{r}'_{h,w} = \boldsymbol{r}_{h,w} + \Delta d \boldsymbol{u}_{h,w}$ and is then reprojected back into the range view. The waveform at each reprojected pixel is shifted along the T axis according to the corresponding range change, so that the histogram remains consistent with the new object position. Unlike a direct tensor shift, this 3D-space manipulation preserves object geometry. Third, the blank region Ω_{blank} left by the shifted object is filled with waveform data from another scene $\boldsymbol{V}^{\text{src}}$. We select a glass-containing source region Ω_{src} from $\boldsymbol{V}^{\text{src}}$ and define a mapping $\phi : \Omega_{\text{blank}} \to \Omega_{\text{src}}$ that assigns each blank pixel to a source pixel. For each pixel in Ω_{blank}, we copy the entire temporal waveform from its mapped source pixel in Ω_{src}. This fills the unrealistic blank region while preserving the ToF waveform structure of the pasted FWL data. Prioritizing glass-containing regions further increases minority-class frequency during training.

FWLCutMix. FWLCutMix extends CutMix [42] to FWL data by combining a destination sample $\boldsymbol{V}_{\text{dst}} \in \mathbb{R}^{H\times W\times T}$ and a source sample $\boldsymbol{V}_{\text{src}} \in \mathbb{R}^{H\times W\times T}$ to generate a synthetic scene. The crop size is determined by a mixing ratio λ sampled randomly: $ell_H = \lfloor H\sqrt{1-\lambda} \rfloor$, $ell_W = \lfloor W\sqrt{1-\lambda} \rfloor$. A minority-class region Ω_{src} is preferentially selected from $\boldsymbol{V}_{\text{src}}$ and pasted at a random location in $\boldsymbol{V}_{\text{dst}}$ to mitigate class imbalance. The waveform $\boldsymbol{Q}_{h,w}$ is pasted in its entirety along the T axis, preserving the temporal waveform structure.

4 Experimental Settings

Implementation details. All methods are trained using the Ghost-FWL dataset [10], the only publicly available FWL dataset with peak-level annotations to date. The raw FWL data in Ghost-FWL has a size of $(H \times W \times T) = (512 \times 400 \times 700)$. First, the top and bottom 88 bins corresponding to reflections from the ceiling and floor and the front 25 bins containing noise from internal sensor reflections were removed. We also removed the rear 375 bins, since the data contains only a few object peaks in the rear region, resulting in a full-frame size of $(H \times W \times T) = (336 \times 400 \times 300)$. Furthermore, due to GPU memory limitations in the comparison methods, we use a crop corresponding to one quarter of the full frame, with size $(H \times W \times T) = (168 \times 200 \times 300)$, as the input for all methods in order to keep the input size consistent. For the Vision Transformer, the patch size is set to $(p_H \times p_W \times p_T) = (8 \times 4 \times 20)$, the number of Transformer blocks N_L is set to 6, and the number of attention heads N_h is set to 6. The feature dimension E is set to 240, and the ratio of dst tokens ρ_d is set to 0.1. Following Ghost-FWL [10], we use the focal loss [18] as the training objective. We use AdamW [21] as the optimizer, set the learning rate to 1.0×10^{-4}, and train the model for 50 epochs. All training and evaluation were performed on a computer equipped with an Intel Core i9-14900K CPU and a single RTX 6000 Ada Generation GPU. Further details are provided in the supplemental material.

Metrics. Following several prior studies [33, 34], we evaluate performance using the mean F1 score over all classes excluding Noise (F1-mean). To verify the inference speedup achieved by token pruning and merging, we report the number of full-size frames that can be inferred per second (FPS).

Baseline Methods. To investigate the effectiveness of the proposed method, we compare it with three FWL-based methods: (1) the proposed method (Ours), (2) a Transformer-based model using pretraining, which is the state-of-the-art model for FWL-based ghost detection (Ikeda et al.) [10], and (3) a model obtained by attaching a classification head for ghost detection to the backbone of a spatio-temporal Transformer model with a U-Net architecture for digital signal processing using FWL (Scheuble et al.) [31]. We also compare our method with pruning and merging methods used for image-based models: (1) DynamicViT [30], which learns whether each token should be pruned using an MLP and then performs pruning, and (2) ToMe [2], which divides tokens into two groups and merges them based on cosine similarity between tokens.

Table 1: Comparison with FWL-based methods and ablation of token pruning and merging components. For FWL-ToPM, we set $\rho_{\text{low}} = 0.7$ and $\rho_m = 0.9$. FWL-ToPM achieves the highest F1 and improves throughput by $8\times$ over the prior state of the art. Our pruning and merging components each outperform their image-based counterparts (DynamicViT and ToMe) and their combination achieves the best F1–FPS trade-off. All methods in this table are trained without FWLAug.

Method	Pruning	Merging	F1-mean	FPS
Scheuble et al. [31]	–	–	0.497	1.97
Ikeda et al. [10]	–	–	0.503	6.54
Ours w/o FWL-ToPM	–	–	0.555	2.01
DynamicViT [30]	✓	–	0.480	20.0
Ours w/Pruning	✓	–	0.591	20.3
ToMe [2]	–	✓	0.543	5.24
Ours w/Merging	–	✓	0.565	18.0
Ours w/FWL-ToPM	✓	✓	**0.592**	**53.2**

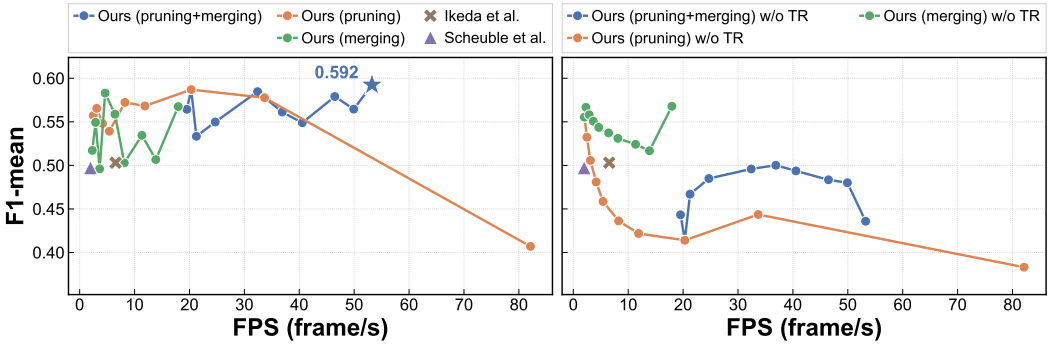

Figure 3: **(Left)** F1-mean vs. FPS trade-off for all compared methods. **(Right)** Comparison of F1-mean and FPS without additional training. For pruning and merging, ρ_{low} and ρ_m are varied from 0 to 0.9, respectively. For pruning+merging, ρ_{low} is fixed to 0.7 while ρ_m is varied. All models are trained without FWLAug.

Augmentation. At each training iteration, either FWLCutMix or FWLShiftMix is selected with equal probability (0.5). For FWLCutMix, 1–3 glass regions are sampled from a randomly selected training frame and pasted into the current frame. For FWLShiftMix, the shift magnitude is sampled uniformly from $[-3, 15]$ meters.

OOD Test Set. To evaluate generalization, we construct an OOD test set at a location entirely separate from Ghost-FWL [10] (see supplementary). The data were collected with the same FWL sensor platform and acquisition configuration as Ghost-FWL and annotated following the same protocol, so that the evaluation isolates changes in scene distribution rather than changes in sensor hardware or annotation procedure. The 500-frame set contains over 300 glass instances and scenes with multiple spatially separated glass regions not present in Ghost-FWL, providing a challenging benchmark for FWLT out-of-distribution evaluation.

5 Experiments and Results

In this section, we conducted the following experiments. To evaluate the effectiveness of the FWL-ToPM architecture, we conducted (1) a comparison with existing FWL-based recognition methods and (2) a comparison with image-based token reduction baselines. We then ablate (3) the pruning and merging components and the (4) FWL augmentations. We also (5) analyze the training-free applicability of token reduction, and (6) discuss limitations.

Comparison with existing FWL-based recognition methods. To evaluate the effectiveness of the FWL-ToPM architecture, we compared its accuracy and computational cost with those of existing FWL-based recognition methods. The results are presented in Table 1 and the left part of Figure 3.

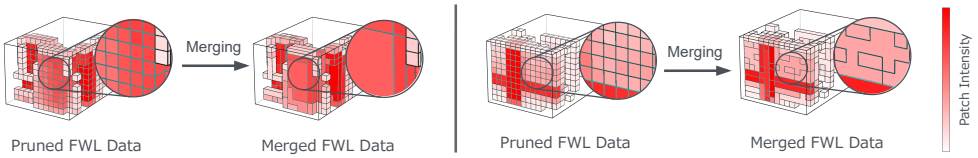

Figure 4: Visualization of token merging on two real Ghost-FWL scenes. Color indicates patch intensity (red: high, white: low). After pruning low-intensity background tokens, the remaining high-intensity tokens are merged by aggregating similar-feature source tokens into destination tokens, reducing the total token count while preserving object-relevant content. The zoomed-in regions show that merged tokens retain the spatial structure of reflected objects across diverse scenes.

Our FWL-ToPM simultaneously achieved a high F1-mean score of 0.592 and an inference speed of 53.2 FPS. Compared with the baselines of Scheuble et al. [31] and Ikeda et al. [10], this corresponds to an approximately 9-point improvement in the F1 score and a speedup of more than $8\times$ in inference speed, demonstrating that FWL-ToPM achieves both high accuracy and high efficiency. A qualitative evaluation of the classification results for the comparison with existing FWL-based methods is provided in the supplementary material.

Comparison with existing pruning and merging methods. To verify the effectiveness of the proposed pruning and merging methods, we compared them with existing methods designed for image-based Transformers. The results are presented in Table 1. As a result, both our pruning and merging methods individually outperformed the existing methods, confirming that each component is independently effective.

Ablation study on pruning and merging. To clarify the contribution of each component in the proposed method, we conducted an ablation study on pruning and merging. The results are presented in Table 1 and the left part of Figure 3. As shown in Table 1, the F1-mean score was 0.555 when neither component was applied, whereas it increased to 0.591 when only token pruning was applied and to 0.565 when only token merging was applied. These results confirm that both components contribute to improving accuracy.

Moreover, as shown in the left part of Figure 3, token pruning alone maintains high accuracy up to around 20 FPS, with an F1-mean of approximately 0.59. However, when the token pruning ratio is increased to achieve a speed of around 80 FPS, the accuracy drops substantially. This is likely because token pruning also removes high-intensity information that is essential for classification. On the other hand, token merging alone suppresses the degradation in accuracy, but its ability to reduce computational cost is limited, resulting in a speed of only around 18 FPS. By combining pruning and merging, the proposed method removes unnecessary background regions through pruning while aggregating the remaining effective waveform information through merging. This complementary design mitigates the limitations of each individual component and enables the method to simultaneously achieve high accuracy and real-time performance exceeding 50 FPS.

Ablation Study of FWLAug. Table 2 shows the contribution of each FWLAug component to both in-distribution and OOD performance. The full FWLAug achieves the largest OOD F1 gain (4.6-point over baseline), particularly on the minority Glass class (8.4-point), confirming improved generalization to unseen scenes. FWLShiftMix contributes more to in-distribution accuracy, while the combination of both drives the OOD improvement, suggesting complementary effects.

Applicability to pretrained models. To evaluate the behavior of token reduction without additional training, we analyzed the performance when pruning and merging were directly applied to pretrained models. The results are presented in the right part of Figure 3. When token pruning was applied without additional training, the accuracy dropped sharply as the reduction ratio increased. This is likely because the model, which was trained using all tokens, also relies on information from low-intensity patches. In contrast, when token merging was applied without additional training, the F1-mean remained above 0.5 even when the inference speed was increased to around 20 FPS, indicating that severe accuracy degradation did not occur. This result indicates that the proposed merging method can effectively reduce computational cost without substantially degrading feature representations, even without additional training.

Table 2: Ablation study of FWLAug. To evaluate out-of-distribution generalization, we construct an OOD test set captured at a location entirely separate from Ghost-FWL [10]. The full FWLAug achieves the best F1 on the OOD test set, particularly for the minority Glass class, confirming improved generalization to unseen scenes.

Method	FWLCutMix	FWLShiftMix	Ghost-FWL [10]		OOD Test	
			F1-mean	Glass	F1-mean	Glass
Ours w/o Aug	✗	✗	0.592	0.352	0.406	0.491
Ours w/ FWLAug	✗	✓	**0.622**	<u>0.376</u>	0.425	0.494
	✓	✓	<u>0.617</u>	**0.385**	**0.452**	<u>0.575</u>
	✓	✗	0.586	0.315	<u>0.434</u>	**0.606**

Limitations. While our proposed framework demonstrates significant improvements, we acknowledge several limitations that provide directions for future work:

- **Risk of removing weak signals:** FWL-ToPM prunes low-intensity regions by regarding them as background or noise, and therefore may incorrectly remove useful weak signals, such as distant objects, low-reflectance objects, or weak reflections under adverse weather conditions. Since pruned patches are treated as noise during inference, such erroneous removal can directly lead to false negatives. Thus, adaptively determining ρ_{low}, ρ_m, and ρ_d according to the scene and signal distribution remains an important direction for future work.
- **Single sensor platform:** Our evaluation relies on a single FWL sensor platform: generalization is evaluated only on unseen scenes acquired with the same sensor platform, acquisition configuration, and annotation protocol as Ghost-FWL, and cross-sensor evaluation has not yet been conducted. Broader validation across different sensors, resolutions, and weather conditions remains necessary.
- **Scope of physical consistency in FWLAug:** FWLAug is a regularization method that preserves waveform structure rather than an optical simulator. Specifically, FWLShiftMix does not model distance-dependent intensity attenuation or recompute the optical interactions between the pasted glass regions and the shifted objects, and FWLCutMix does not guarantee consistent reflection paths across the entire scene because of spatial discontinuities at the paste boundaries.
- **System-level efficiency evaluation:** Our current efficiency evaluation is limited to model-level throughput. Sensor-to-decision latency, including sensor input, preprocessing, and system integration, as well as deployment on actual embedded robotic platforms, are important directions for future work.

6 Conclusion

We presented FWL-ToPM and FWLAug, two complementary methods that address the core bottlenecks of real-time inference and limited generalization for Full-Waveform LiDAR Transformers. FWL-ToPM exploits the intensity sparsity inherent to FWL data to prune redundant background tokens and merge the remainder, achieving 53.2 FPS with an 8× speedup and a 9-point F1 gain over the prior state-of-the-art — establishing the first real-time FWLT. A notable finding is that pruning up to 70% of tokens consistently improves accuracy, suggesting that low-intensity tokens act as noise that dilutes object-relevant signals. FWLAug improves generalization through FWLCutMix and FWLShiftMix, both preserving FWL's ToF waveform structure and frustum geometry, with gains confirmed on an OOD test set of unseen scenes. Together, these advances bring real-time FWL perception a step closer to deployment in autonomous driving and robotics.

Acknowledgements and disclosures. We gratefully acknowledge Takahiro Kado, Ibuki Fujioka, and Taro Beppu at Sony Semiconductor Solutions for their generous support and for providing the Sony LiDAR sensor used in this work. The Sony LiDAR system played an essential role in enabling this research. This research was supported in part by JST Next-generation Edge AI Semiconductor JPMJES2515, JST CREST JPMJCR21D2, JPMJCR23M4, JST PRESTO JPMJPR22PA, JST ASPIRE JPMJAP2515, JST FOREST JPMJFR242I, JSPS KAKENHI 24K02940, and 25H01159.

References

[1] Daniel Bolya and Judy Hoffman. Token Merging for Fast Stable Diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4599–4603, 2023.

[2] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT But Faster. In *International Conference on Learning Representations (ICLR)*, 2023.

[3] Maxim Bonnaerens and Joni Dambre. Learned Thresholds Token Merging and Pruning for Vision Transformers. *Transactions on Machine Learning Research*, 2023.

[4] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. EfficientViT: Lightweight Multi-Scale Attention for High-Resolution Dense Prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17302–17313, 2023.

[5] Nesrine Chehata, Li Guo, and Clément Mallet. Airborne LIDAR feature selection for urban classification using random forests. In *Laser Scanning*, 2009.

[6] Jean-Luc Déziel, Pierre Merriaux, Francis Tremblay, Dave Lessard, Dominique Plourde, Julien Stanguennec, Pierre Goulet, and Pierre Olivier. Pixset: An Opportunity for 3D Computer Vision to Go Beyond Point Clouds with a Full-Waveform LiDAR Dataset. In *IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2987–2993, 2021.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[8] Han, Kai and Wang, Yunhe and Chen, Hanting and Chen, Xinghao and Guo, Jianyuan and Liu, Zhenhua and Tang, Yehui and Xiao, An and Xu, Chunjing and Xu, Yixing and Yang, Zhaohui and Zhang, Yiman and Tao, Dacheng. A Survey on Vision Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(1):87–110, 2023.

[9] Joakim Bruslund Haurum, Sergio Escalera, Graham W. Taylor, and Thomas B. Moeslund. Which Tokens to Use? Investigating Token Reduction in Vision Transformers. In *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 773–783, 2023.

[10] Kazuma Ikeda, Ryosei Hara, Rokuto Nagata, Ozora Sako, Zihao Ding, Takahiro Kado, Ibuki Fujioka, Taro Beppu, Mariko Isogawa, and Kentaro Yoshioka. Ghost-FWL: A Large-Scale Full-Waveform LiDAR Dataset for Ghost Detection and Removal. *arXiv preprint arXiv:2603.28224*, 2026.

[11] Mariko Isogawa, Ye Yuan, Matthew O'Toole, and Kris M Kitani. Optical non-line-of-sight physics-based 3d human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7013–7022, 2020.

[12] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1383–1392, 2024.

[13] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, Minghai Qin, and Yanzhi Wang. SPViT: Enabling Faster Vision Transformers via Latency-Aware Soft Token Pruning. In *European Conference on Computer Vision (ECCV)*, pages 620–640, 2022.

[14] Sheng Li, Qitao Tan, Yue Dai, Zhenglun Kong, Tianyu Wang, Jun Liu, Ao Li, Ninghao Liu, Yufei Ding, Xulong Tang, and Geng Yuan. Mutual Effort for Efficiency: A Similarity-based Token Pruning for Vision Transformers in Self-Supervised Learning. In *International Conference on Learning Representations (ICLR)*, 2025.

[15] Xirui Li, Chao Ma, Xiaokang Yang, and Ming-Hsuan Yang. VidToMe: Video Token Merging for Zero-Shot Video Editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7486–7495, 2024.

[16] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

[17] Wenzhi Liao, Frieke Van Coillie, Lianru Gao, Liwei Li, Bing Zhang, and Jocelyn Chanussot. Deep Learning for Fusion of APEX Hyperspectral and Full-Waveform LiDAR Remote Sensing Data for Tree Species Mapping. *IEEE Access*, 6:68716–68729, 2018.

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[19] Yifei Liu, Mathias Gehrig, Nico Messikommer, Marco Cannici, and Davide Scaramuzza. Revisiting Token Pruning for Object Detection and Instance Segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2658–2668, 2024.

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[21] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2017.

[22] Chenyang Lu, Daan de Geus, and Gijs Dubbelman. Content-aware Token Sharing for Efficient Semantic Segmentation with Vision Transformers . In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23631–23640, 2023.

[23] Wenbo Lu, Shaoyi Zheng, Yuxuan Xia, and Shengjie Wang. ToMA: Token Merge with Attention for Diffusion Models. In *International Conference on Machine Learning (ICML)*, 2025.

[24] Junzhu Mao, Yang Shen, Jinyang Guo, Yazhou Yao, Xiansheng Hua, and Hengtao Shen. Prune and Merge: Efficient Token Compression for Vision Transformer With Spatial Information Preserved. *IEEE Transactions on Multimedia*, 27:4670–4683, 2025.

[25] G.M. Morton. *A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing*. International Business Machines Company, 1966.

[26] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. In *International Conference on 3D Vision*, pages 116–125, 2021.

[27] Narges Norouzi, Svetlana Orlova, Daan De Geus, and Gijs Dubbelman. ALGM: Adaptive Local-then-Global Token Merging for Efficient Semantic Segmentation with Plain Vision Transformers . In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15773–15782, 2024.

[28] Lorenzo Papa, Paolo Russo, Irene Amerini, and Luping Zhou. A Survey on Efficient Vision Transformers: Algorithms, Techniques, and Performance Benchmarking . *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(12):7682–7700, 2024.

[29] Sam Pollard and Michael Wray. Video, How do Your Tokens Merge? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3338–3347, 2025.

[30] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[31] Dominik Scheuble, Hanno Holzhüter, Steven Peters, Mario Bijelic, and Felix Heide. Lidar Waveforms are Worth 40x128x33 Words. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 28913–28924, 2025.

[32] Takayuki Shinohara, Haoyi Xiu, and Masashi Matsuoka. FWNetAE: Spatial Representation Learning for Full Waveform Data Using Deep Learning. In *IEEE International Symposium on Multimedia (ISM)*, pages 259–2597, 2019.

[33] Takayuki Shinohara, Haoyi Xiu, and Masashi Matsuoka. FWNet: Semantic Segmentation for Full-Waveform LiDAR Data Using Deep Learning. *Sensors*, 20(12), 2020.

[34] Takayuki Shinohara, Haoyi Xiu, and Masashi Matsuoka. Semantic Segmentation for Full-Waveform LiDAR Data Using Local and Hierarchical Global Feature Extraction. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, page 640–650, 2020.

[35] Shuochen Su, Felix Heide, Gordon Wetzstein, and Wolfgang Heidrich. Deep end-to-end time-of-flight imaging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6383–6392, 2018.

[36] Wenhao Sun, Rong-Cheng Tu, Jingyi Liao, Zhao Jin, and Dacheng Tao. AsymRnR: Video Diffusion Transformers Acceleration with Asymmetric Reduction and Restoration. In *International Conference on Machine Learning (ICML)*, 2025.

[37] Tuan Anh Tran, Duy Minh Ho Nguyen, Hoai-Chau Tran, Michael Barz, Khoa D. Doan, Roger Wattenhofer, Vien Anh Ngo, Mathias Niepert, Daniel Sonntag, and Paul Swoboda. How Many Tokens Do 3D Point Cloud Transformer Architectures Really Need? In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[38] Andrew M. Wallace, Abderrahim Halimi, and Gerald S. Buller. Full Waveform LiDAR for Adverse Weather Conditions. *IEEE Transactions on Vehicular Technology*, 69(7):7064–7077, 2020.

[39] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler Faster Stronger. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[40] Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, and Ling Shao. Polarmix: A general data augmentation technique for lidar point clouds. *Conference on Neural Information Processing Systems (NeurIPS)*, 35:11035–11048, 2022.

[41] Ryo Yoshida, Takami Sato, Wenlun Zhang, Yuki Hayakawa, Shota Nagai, Takahiro Kado, Taro Beppu, Ibuki Fujioka, Yunshan Zhong, and Kentaro Yoshioka. Neural Reconstruction of LiDAR Point Clouds under Jamming Attacks via Full-Waveform Representation and Simultaneous Laser Sensing. *arXiv preprint arXiv:2604.00371*, 2026.

[42] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019.

[43] Evelyn Zhang, Bang Xiao, Jiayi Tang, Qianli Ma, Chang Zou, Xuefei Ning, Xuming Hu, and Linfeng Zhang. Token Pruning for Caching Better: 9$\times$ Acceleration on Stable Diffusion for Free. *arXiv preprint arXiv:2501.00375*, 2024.

[44] Evelyn Zhang, Jiayi Tang, Xuefei Ning, and Linfeng Zhang. Training-Free and Hardware-Friendly Acceleration for Diffusion Models via Similarity-based Token Pruning. *AAAI Conference on Artificial Intelligence (AAAI)*, 39(9):9878–9886, 2025.

[45] Stefano Zorzi, Eleonora Maset, Andrea Fusiello, and Fabio Crosilla. Full-Waveform Airborne LiDAR Data Classification Using Convolutional Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(10):8255–8261, 2019.

Supplementary Material

Contents

A Implementation Details **14**

A.1 Training 14

A.2 Inference 14

A.3 Baseline Methods 15

B FWL Generalization Benchmark **15**

C Additional Experiments and Results **15**

C.1 Statistical Significance of FWL-Based Comparison 15

C.2 Comparison with Top-K Peak Extraction 15

C.3 Ablation on Space-Filling Curves 16

C.4 Qualitative Evaluation 17

A Implementation Details

A.1 Training

We evaluated our method on the Ghost-FWL dataset [2]. Following the original data split, we assigned Scenes 001, 003, 004, 005, 006, 008, and 010 to the training and validation set, and Scenes 002, 007, and 009 to the test set. The original dataset provided by Ikeda et al. [2] contains 13,853 training frames and 2,994 validation frames. To reduce computational cost, we randomly sampled one-third of the data from both sets while preserving the scene distribution. Consequently, our final subsets consist of 4,617 training frames and 998 validation frames. For the test set, we did not perform any sampling and used all 1,427 original frames for evaluation.

To mitigate overfitting in the Transformer encoder, we applied a dropout rate of 0.2 to the positional embeddings, the attention weights, and the output of each block. We also set the drop path rate for the residual connections to 0.2.

The model was trained using the AdamW optimizer with a batch size of 4. The optimizer hyperparameters were set to β_1=0.9, β_2=0.999, $\epsilon = 1 \times 10^{-8}$, and weight decay $\lambda = 1 \times 10^{-2}$. We employed CosineLRScheduler, a cosine learning rate scheduler with a linear warmup, setting the total number of epochs to 50, the minimum learning rate to 0, the warmup duration to 3 epochs, and the initial warmup learning rate to 1.0×10^{-5}.

For the loss function, we utilized the focal loss [3] with the class weights set to $\alpha_{\mathrm{glass}} = 1.56$, $\alpha_{\mathrm{ghost}} = 1.13$, $\alpha_{\mathrm{object}} = 0.536$, and $\alpha_{\mathrm{noise}} = 0.0625$, and the focusing parameter γ set to 2.0.

All training and processing were performed on a computer equipped with an Intel Core i9-14900K CPU and a single RTX 6000 Ada Generation GPU, running the Ubuntu 20.04 operating system. In this environment, the training time was approximately 5.5 minutes per epoch.

A.2 Inference

Because the model input is limited to one-quarter of the full frame, processing an entire frame in a single forward pass is not feasible. To evaluate the full-frame performance, we followed Ikeda et al. [2] by sequentially applying a non-overlapping sliding window to infer local regions. The resulting outputs were then spatially concatenated to reconstruct the full-frame predictions. Furthermore, following Ikeda et al. [2] for label assignment, a voxel was classified into the category with a predicted probability exceeding 0.5; if no class probability exceeded this threshold, it was labeled as *Undefined*.

A.3 Baseline Methods

The baseline methods evaluated in our experiments are primarily based on their official implementations, with appropriate modifications to accommodate our specific task and the 3D data structure. The specific adaptations and settings for each method are detailed below.

Scheuble et al. [5]: As the original model targets a different task, we appended a new prediction head for voxel-wise classification. After upsampling the decoder output features to the original spatio-temporal resolution, we applied $1 \times 1 \times 1$ 3D convolution to generate the class logits for each voxel.

Ikeda et al. [2]: The architecture and training procedure strictly follow the official implementation. However, because this model divides the spatial dimensions into 16×16 patches, the input spatial resolution must be divisible by 16. Therefore, we adjusted the input size from $(H \times W \times T) = (168 \times 200 \times 300)$ used in our method to $(H \times W \times T) = (176 \times 208 \times 300)$ for evaluation.

DynamicViT [4]: We extended this 2D image-based method to accommodate 3D FWL data. To align the conditions with our proposed pruning strategy, we configured the model to perform token selection at the input stage and set the keeping ratio to 0.3. This ensures that 70% of the input tokens are pruned, matching our pruning ratio.

ToMe [1]: Similar to DynamicViT, we extended this method for 3D data. The original algorithm alternately assigns tokens as dst tokens and src tokens on a 2D grid; we adapted this mechanism to alternate across a 3D token grid. This structural constraint limits the maximum merge ratio to 0.5. Because our proposed method utilizes a merge ratio of 0.9, we set the merge ratio in ToMe to its maximum allowable value of 0.5 to approximate our experimental conditions. To align with our setup, we applied the merging process to the attention layers. Additionally, mirroring our merging strategy, we incorporated a step to copy the features of the dst tokens back to the merged src tokens after the attention computation to maintain consistency with the residual connections.

B FWL Generalization Benchmark

To evaluate generalization to unseen scenes acquired with the same sensor platform, we construct a new FWL test set distinct from Ghost-FWL [2]. Data were collected using the same FWL sensor as Ghost-FWL, ensuring that any performance difference reflects scene distribution rather than sensor variation. The set consists of 500 frames collected from 10 viewpoints (50 frames each), annotated following the procedure of Ikeda et al. [2]. All scenes were captured outdoors during daytime at a seven-story building clad in glass on all four sides, with additional glass installations around elevators and connecting corridors, totaling over 300 glass windows. Such a glass-dense environment does not appear in Ghost-FWL [2], making this test set a challenging out-of-distribution benchmark (Figure 5).

C Additional Experiments and Results

C.1 Statistical Significance of FWL-Based Comparison

Table 3 presents the comparison with existing FWL-based methods under different random seeds. To avoid drawing conclusions from a single trial, we evaluated each method over 3 independent runs and report the mean and standard deviation of F1-mean. This evaluation captures seed-dependent variability and demonstrates that our method achieves superior performance robustly, rather than only under a particular random seed.

The results on the OOD test set further suggest the applicability of FWLAug to unseen data. While Ours fails to detect glass regions located behind other glass surfaces, Ours + FWLAug partially detects such regions, albeit with some inaccuracies. This indicates that FWLAug improves robustness to configurations not observed in the training data.

C.2 Comparison with Top-K Peak Extraction

Conventional multi-return ToF LiDAR keeps only the Top-K highest-intensity peaks of each ray. In contrast, FWL-ToPM ranks 3D patches by their maximum intensity over the entire frame, removes a fixed proportion ($\rho_{\text{low}} = 0.7$) of low-ranked patches, and retains the complete temporal waveform

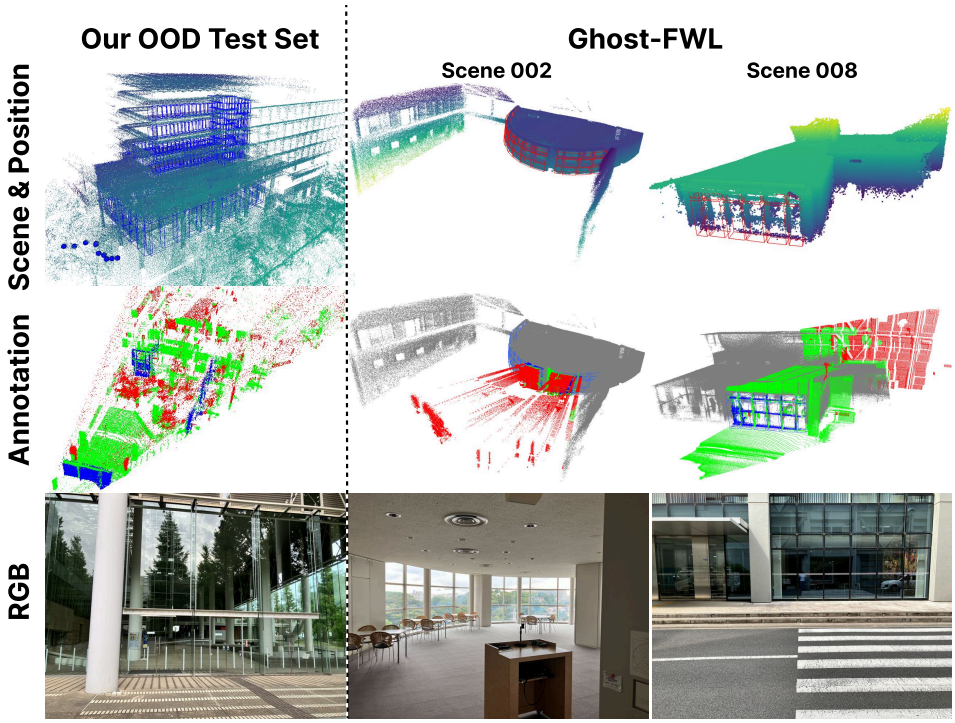

Figure 5: Our OOD Test Set. Based on the dense 3D maps as shown in Scene, we annotated FWL data with 4 labels: Ghost (red), Object (green), Glass (blue), Noise. Blue crosses indicate the LiDAR positions during data acquisition. The RGB images show the scenery of the capture locations.

Table 3: Statistical significance of the main comparison with FWL-based methods. F1-mean is reported as mean ± standard deviation over 3 independent runs with different random seeds. All methods are trained without FWLAug.

Method	F1-mean	Object-F1	Glass-F1	Ghost-F1	FPS
Scheuble et al. [5]	0.497 ± 0.043	0.648	0.304	0.541	1.97
Ikeda et al. [2]	0.503 ± 0.005	0.680	0.268	0.561	6.54
Ours	**0.592 ± 0.005**	**0.741**	**0.358**	**0.678**	**53.2**

within each remaining patch. Figure 6 shows waveforms along single rays from two Ghost-FWL scenes. In Scene 001, the Top-3 peaks consist of a Glass return and two Ghost returns, so the Object return around depth bin 60 is discarded. In Scene 007, two Object returns and a Ghost return occupy the Top-3 peaks, so the Glass return around depth bin 90 is discarded. Since the Top-2 peaks are a subset of the Top-3 peaks, these returns are also discarded by Top-2 extraction. In both cases, the discarded returns lie well above the pruning threshold and are therefore retained by our pruning. These examples illustrate that a fixed per-ray peak budget can drop important Object and Glass returns when strong Ghost returns are present along the same ray, whereas our patch-level pruning preserves them together with their full waveforms.

C.3 Ablation on Space-Filling Curves

In FWL-ToPM, the tokens remaining after pruning are serialized along a space-filling curve before dst token selection, and our experiments adopt the Morton order as this curve. To examine whether the choice of curve affects performance, we replace the Morton order with the Hilbert curve [**?**] while keeping all other settings unchanged, and evaluate both the merging-only and the pruning + merging settings. As shown in Table 4, the two curves yield nearly identical F1-mean scores in both settings, with the Morton order slightly higher: 0.565 versus 0.564 for merging only, and 0.592 versus

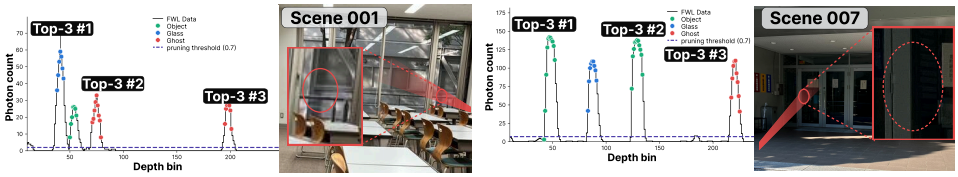

Figure 6: Waveforms along single rays in Ghost-FWL Scene 001 (left) and Scene 007 (right). Markers indicate annotated Object (green), Glass (blue), and Ghost (red) bins, labels indicate the Top-3 highest-intensity peaks of each ray, and the dashed line indicates the intensity threshold below which patches are pruned at $\rho_{\text{low}} = 0.7$. The photographs show the corresponding scenes, with the red marker indicating the observed ray. Top-3 peak extraction discards the Object and the Glass return, whereas our intensity-based pruning retains both because they lie above the pruning threshold.

Table 4: Comparison of space-filling curves used to serialize tokens for merging. F1 scores on the Ghost-FWL test set are reported. We set $\rho_{\text{low}} = 0.7$ (pruning + merging only), $\rho_m = 0.9$, and $\rho_d = 0.1$. All models are trained without FWLAug.

Setting	Curve	Object-F1	Glass-F1	Ghost-F1	F1-mean
Merging	Hilbert	0.690	0.349	**0.652**	0.564
	Morton (Ours)	**0.702**	**0.369**	0.624	**0.565**
Pruning + Merging	Hilbert	0.740	0.355	0.672	0.589
	Morton (Ours)	**0.741**	**0.358**	**0.678**	**0.592**

0.589 for pruning + merging. This indicates that the benefit comes from serializing the sparse tokens in an order that preserves spatial proximity, rather than from a particular choice of curve.

C.4 Qualitative Evaluation

Figure 7 shows a qualitative comparison of the classification results with existing FWL-based recognition methods. We evaluate the methods on scenes from the Ghost-FWL test split and on additional scenes not included in Ghost-FWL [2]. Across these scenes, Ours classifies Glass and Ghost regions more accurately than the existing methods. In particular, improvements are observed in regions where Object and Ghost overlap, as well as on Glass surfaces that produce only weak waveform peaks. On the additional test scene, Ours also improves the classification of Objects located behind Glass regions, although some Glass regions tend to be partially confused with Object.

Nevertheless, failure cases remain. In particular, Glass regions located behind other Glass surfaces are still difficult to classify accurately. We consider this to be partly due to the limited number of such configurations in Ghost-FWL [2]. By increasing the occurrence of Glass-containing regions during training, FWLAug provides additional waveform and spatial variations associated with Glass. As a result, Ours+FWLAug shows partial improvements in detecting Glass regions in such challenging configurations, although the predictions remain imperfect. This suggests that FWLAug can improve robustness to glass configurations that are under-represented in the original training data.

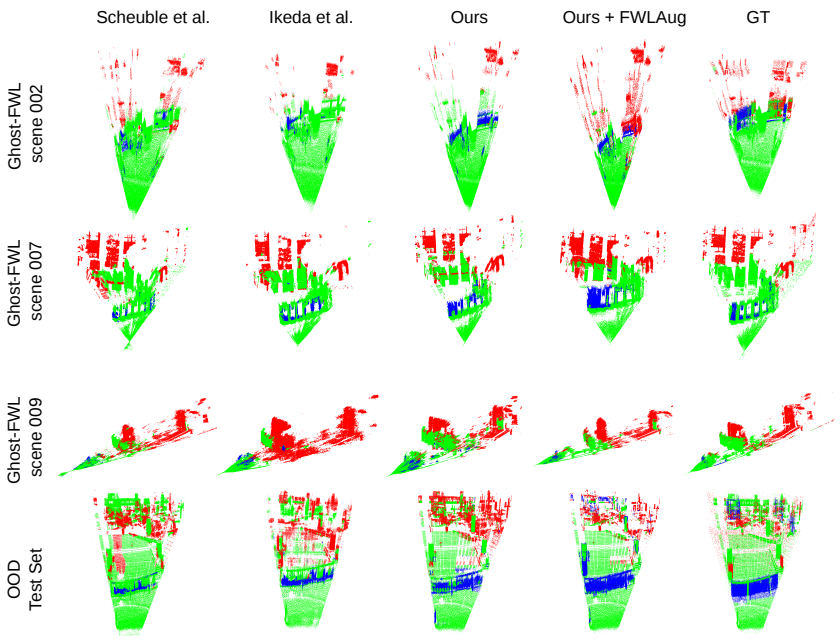

Figure 7: Qualitative comparison of classification results with existing FWL-based recognition methods.

References

[1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT But Faster. In *International Conference on Learning Representations (ICLR)*, 2023.

[2] Kazuma Ikeda, Ryosei Hara, Rokuto Nagata, Ozora Sako, Zihao Ding, Takahiro Kado, Ibuki Fujioka, Taro Beppu, Mariko Isogawa, and Kentaro Yoshioka. Ghost-FWL: A Large-Scale Full-Waveform LiDAR Dataset for Ghost Detection and Removal. *arXiv preprint arXiv:2603.28224*, 2026.

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[4] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[5] Dominik Scheuble, Hanno Holzhüter, Steven Peters, Mario Bijelic, and Felix Heide. Lidar Waveforms are Worth 40x128x33 Words. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 28913–28924, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are limited to the empirical findings reported in the paper. The experimental evidence supporting these claims is provided in §5, and the limitations are discussed in §5

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations at the end of §5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results such as theorems, lemmas, or formal guarantees, so this question is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper discloses the information needed to reproduce the main experimental results, including model details in §3, experimental settings in § 4, and additional implementation details in §A of the supplemental material. These details are sufficient to reproduce the results supporting the main claims and conclusions of this paper.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and the OOD test set are publicly available via the project page: `https://keio-csg.github.io/FWL-ToPM` together with instructions for reproducing the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper specifies the experimental settings in §4. Additional implementation details are provided in §A of the supplemental material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main comparison with existing FWL-based methods is shown in Table 3 in Supplementary Material §C.1. For all methods, F1-mean is reported as mean $\pm$ standard deviation over 3 independent runs with different random seeds, capturing seed-dependent run-to-run variability under the same experimental setting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This paper provides the compute resources in §A of the supplemental material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational research, and we do not identify direct positive or negative societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release any data or models with a high risk of misuse, so additional release safeguards are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing datasets and code are properly cited, and their licenses and terms of use are respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the code of FWL-ToPM and FWLAug and the OOD test set introduced in this paper via the project page: `https://keio-csg.github.io/FWL-ToPM`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing experiments or research with human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing, editing, or formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.