LiDAR-in-the-loop Hyperparameter Optimization

Félix Goudreault[1] Dominik Scheuble[2] Mario Bijelic[2,3] Nicolas Robidoux[1] Felix Heide[1,3]
[1]Algolux [2]Mercedes-Benz [3]Princeton University

Abstract

LiDAR has become a cornerstone sensing modality for 3D vision. LiDAR systems emit pulses of light into the scene, take measurements of the returned signal, and rely on hardware digital signal processing (DSP) pipelines to construct 3D point clouds from these measurements. The resulting point clouds output by these DSPs are input to downstream 3D vision models – both, in the form of training datasets or as input at inference time. Existing LiDAR DSPs are composed of cascades of parameterized operations; modifying configuration parameters results in significant changes in the point clouds and consequently the output of downstream methods. Existing methods treat LiDAR systems as fixed black boxes and construct downstream task networks more robust with respect to measurement fluctuations. Departing from this approach, the proposed method directly optimizes LiDAR sensing and DSP parameters for downstream tasks. To investigate the optimization of LiDAR system parameters, we devise a realistic LiDAR simulation method that generates raw waveforms as input to a LiDAR DSP pipeline. We optimize LiDAR parameters for both 3D object detection IoU losses and depth error metrics by solving a nonlinear multi-objective optimization problem with a 0th-order stochastic algorithm. For automotive 3D object detection models, the proposed method outperforms manual expert tuning by 39.5% mean Average Precision (mAP).

1. Introduction

Environment perception for autonomous drones [12, 53] and vehicles [72] requires precise depth sensing for safety-critical control decisions. Scanning LiDAR sensors have been broadly adopted in autonomous driving [3, 7, 56] as they provide high temporal and spatial resolution, and recent advances in MEMS scanning [67] and photodiode technology [64] have reduced their cost and form factor.

The 3D LiDAR point cloud (PC) data that existing 3D detection methods take as input is produced by a LiDAR and digital signal processor (DSP) pipeline with many measurement and processing steps. Typical LiDAR sensors operate by sending out a laser pulse and measuring the temporal response through an Avalanche Photo Diode (APD). This temporal wavefront signal is fed to a DSP that extracts peaks corresponding to echos from candidate targets within the scene [70]. As such, DSP processing results in a 1000-fold data reduction for a single emitted beam, producing single or multiple 3D points per beam. Compressing the waveform into points in 3D space with minimal information loss is challenging because of object discontinuities, sub-surface scattering, multipath reflections, and scattering media, see Fig. 1. In particular, significant scattering occurs in adverse weather conditions like fog [3,4,6,20,23,25,66], rain [6,18,66] and snow [19,25,27,34]. LiDAR manufacturers currently handle such complications by manually adjusting internal sensing and DSP parameters in controlled environments and restricted real-world scenarios using a combination of visual inspection and depth quality metrics.

Generally, LiDAR production units are black boxes with configuration parameters hidden from the user. To account for noisy point cloud measurements with spurious artifacts, existing work has explored methods that add simulated adverse effects and point cloud degradations that model rain [18, 26, 58], fog [3, 20, 50] and snow [19] to real black-box LiDAR datasets. Augmented point clouds in hand, downstream vision models are retrained for predictions more robust to point cloud data corruption. Another approach consists of generating synthetic measurements from 3D scenes using rendering engines [24, 57, 69]. Such existing methods typically avoid simulating transient light propagation and signal processing by converting 3D scene depth directly into a point cloud, thus lack physically realistic modeling of fluctuations arising from multipath effects or measurement noise. Notably, existing simulation methods that alter measurements or generate synthetic point clouds do not optimize sensing or DSP parameters for downstream vision performance.

In this work, we directly optimize LiDAR pulse configuration and DSP hyperparameters for end-to-end downstream 3D object detector losses and PC depth quality metrics, a challenging task because hyperparameter space involves tens to hundreds of categorical, discrete and effectively continuous parameters affecting downstream tasks in complex nonlinear ways via the intermediate point cloud. Examples of categorical hyperparameters are Velodyne LiDAR sensor return modes that configure internal wavefront peak selection algorithms for point cloud formation;

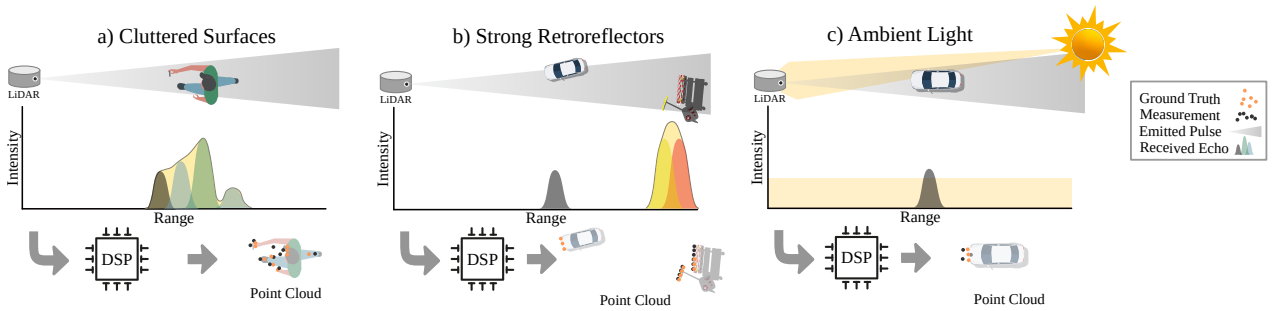

Figure 1. LiDAR Point Cloud Formation. Typical LiDAR sensor PC measurements are produced by a multi-stage measurement and signal processing chain: The LiDAR emits a laser pulse. This signal travels through the scene and returns to the detector after single or multiple reflections. Cluttered surfaces (a), strong retroreflectors (b) and ambient light are introduced (c) in the returned signals. Thus, the full transient waveform read by sensors is the superposition of multiple return paths. The DSP, itself a chain of processing blocks, processes all temporal waveforms and extracts a continuous stream of 3D points that forms the final point cloud (bottom).

rotation velocity, which impacts angular resolution, is an example of a continuous hyperparameter [4].

Grid search optimization is impractical here because of combinatorial explosion. Orthogonally to LiDAR, it was recently shown that 0th order solvers can find camera DSP hyperparameters that improve downstream 2D object detectors [36, 48]. We propose an optimization method for LiDAR sensing and DSP hyperparameters that minimizes end-to-end domain-specific losses such as RMSE of the measured depth against ground truth and IoU measured on downstream 3D object detection. To assess the proposed method, we devised a LiDAR simulation method based on the CARLA engine [13] that models a LiDAR DSP as well as the full transient noisy waveform formed by multiple laser echoes. We optimize sensing and DSP hyperparameters by solving a Multi-Objective black-box Optimization (MOO) problem with a novel CMA-ES (Covariance Matrix Adaptation-Evolution Strategy [21]) that relies on a max-rank multi-objective scalarization loss [36] to dynamically improve scale matching between different loss components. In combination with a novel champion selection method, it finds a *balanced Pareto-optimal* solution for which no loss component has a comparatively poor value for LiDAR optimization with multiple objectives. We validate the proposed optimization method for 3D object detection and point cloud depth estimation, both in simulation and using an off-the-shelf experimental LiDAR sensor.

Specifically, we make the following contributions:

- We introduce a LiDAR wavefront simulation for the CARLA simulation environment that models realistic transient scene responses.
- We devise a novel multi-objective optimization method for balanced MOO of LiDAR parameters.
- We validate end-to-end LiDAR sensing and DSP optimization for 3D object detection and depth estimation through simulation and with a real system. For all applications, our approach outperforms existing state-of-the-art methods, including expert tuning.

Simulator code and DSP models are published here[1].

Limitations Because commercial LiDAR units are IP-protected black boxes, interfacing their DSP hyperparameters is not straightforward. While the off-the-shelf LiDAR system used in this work makes some DSP hyperparameters accessible, most LiDAR systems are completely closed. We hope that these findings spur LiDAR vendors to follow the lead of digital camera and ISP (Image Signal Processor) vendors and open their processing pipelines.

2. Related Work

DSP and Sensor Hyperparameter Optimization. A growing body of work explores the optimization of sensors and DSPs for downstream vision tasks. Existing methods target camera ISPs and optics [36, 48, 59, 60, 71]. Departing from manual hyperparameter tuning by experts, these approaches optimize them automatically, driven by one or more downstream performance metric. As DSP and sensor hyperparameters can be categorical and losses are often non-convex and noisy, diverse optimization methods have been proposed. Early methods targeted specific processing blocks [35, 40, 44] as differentiable programs or in a reduced parameter space. Recent methods rely on differentiable pipeline proxies [60] or 0th-order optimization [36], alone or in combination with block coordinate descent [48]. One advantage of 0th-order optimizers is that they handle black box hardware and DSPs. 0th-order solvers used to optimize camera systems include MOEA/D [40, 73] and CMA-ES [21, 36, 48]. These approaches successfully tackle camera pipeline optimization, from the optics [59] to downstream detectors [36, 48]. To our knowledge, however, no loss-driven method has been proposed for end-to-end LiDAR system optimization: existing LiDAR sensors are designed in isolation from downstream depth and detection

[1] https://light.princeton.edu/LITL-Optimization/

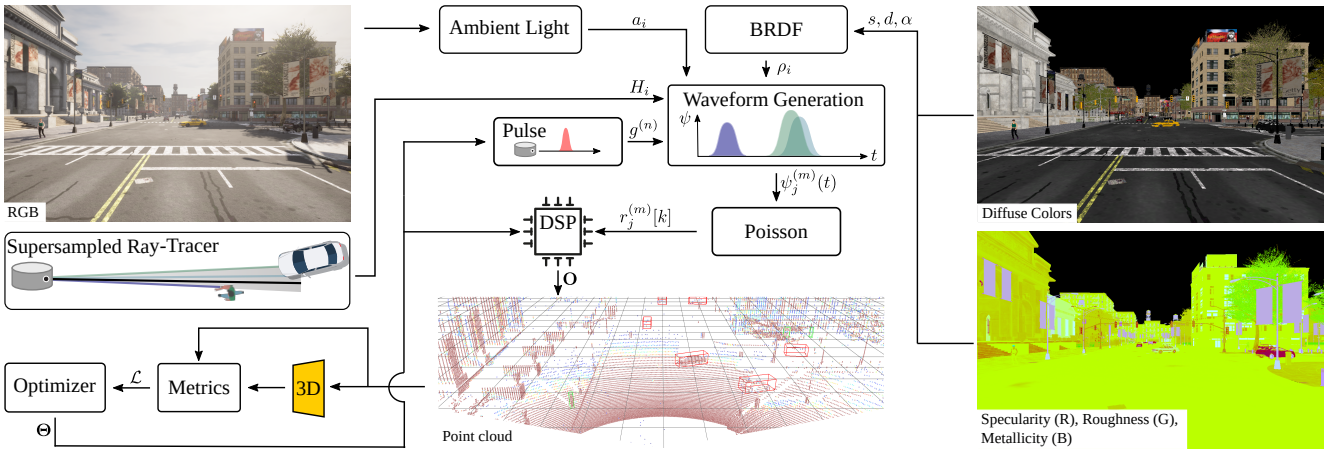

Figure 2. LiDAR Simulation Method. We propose a parameterizable LiDAR simulation model that generates full transient waveforms by extracting scene response H, ambient light a and object reflectances s, d, α from CARLA. End-to-end loss functions drive an MOO solver toward an optimal vector $\mathbf{\Theta}$ that includes both pulse and DSP hyperparameters: Wavefronts are processed by the DSP, resulting in a point cloud $\mathbf{O}$. The "loop is closed" by feeding $\mathbf{O}$ to 3D object detection and depth estimation methods, the output of which is "rated" by loss functions $\boldsymbol{\mathcal{L}}$. Validation datasets cycle through the loop until $\mathbf{\Theta}$ converges.

tasks. Aiming to close this gap, this work proposes automated LiDAR hyperparameter optimization.

LiDAR Sensing and Point Cloud Generation. LiDAR sensors produce point clouds by emitting pulses of light into a scene and measuring the round trip time of sensor returns. Extracting a point cloud from time-of-flight measurements is a complex process that depends on measurement procedure specifics like beam formation, scanning, pulse/continuous wave generation and peak finding within the acquired temporal laser response [1,46]. LiDAR sensors differ in their scanning pattern, beam steering technology, wavelength, pulse profile, coherence of the measurement step, detector technology and DSP capabilities to process the measurement echos [1, 31].

Existing LiDAR sensors can extract single or multiple peaks resulting from multi-path interreflections in the scene. Assuming a single Lambertian reflector in the scene, the temporal resolution and signal-to-noise ratio (SNR) of the measurement are tied to laser power. Accordingly, some existing optimization methods [43] use automated runtime laser power adjustment to *maximize SNR while preventing oversaturation*. Other approaches have been proposed for adaptive beam steering [38, 45]. Recently, Vodisch *et al*. [65] proposed beam configuration optimization via reinforcement learning methods driven by a downstream 3D detection loss. This method is the most similar to the proposed approach; however, it only predicts beam patterns, i.e., where to place sparse samples. In contrast, we optimize DSP hyperparameters in addition to sensing ones, pulse power and scanning parameters included.

LiDAR Simulation. To assess and validate the proposed method, we introduce a new LiDAR simulation method that plugs directly into the CARLA simulator. Existing work [5, 14, 37, 47, 52] proposes several simulation environments; the open-source CARLA [14] simulator is broadly adopted in academia. Simulation frameworks have enabled the creation of multimodal synthetic datasets used successfully in academia and industry, *e.g.*, PreSIL [24], SHIFT [57], AIODrive [32, 69] and SYNTHIA [49]. However, underlying simulation methods employ heuristic forward models; in addition, none of the datasets include full waveform returns that would allow simulating LiDAR point cloud generation. Even the AIODrive dataset [69], in which multiple peaks are returned via depth image convolution and Single Photon Avalanche Diode (SPAD) quantization, bakes transients into SPAD image formation, hence falls short of enabling realistic transient simulation. Similarly, real PC datasets augmentation methods have been employed to tackle rare occuring events like obstacles [16, 33], traffic [68], rain [20], fog [20] or snow [19, 26]. However, since the underlying datasets do not include the raw wavefronts, none of these approach facilitate modeling the DSP pipeline. We address these challenges with our simulation method that simulates full wavefront signals. Combined with a realistic DSP model, this produces PC data representative of real systems.

3. Transient LiDAR Forward Model

Consider a single laser pulse emitted by a LiDAR unit into a 3D scene, from which a returned signal is detected by a SPAD detector which sends temporal histograms to the sensor DSP. For channel n at time t, we model the sensor-incident temporal photon flux ψ [22, 41, 54] as

$$\psi^{(n)}(t) = (H * g^{(n)})(t) + a(t), \tag{1}$$

where $g^{(n)}$ is the temporally varying photon flux emitted by the laser for channel n, H is the transient response from the

scene, $a(t)$ is the ambient photon flux, and $*$ is the temporal convolution operator. By definition, the transient scene response H includes multipath returns from scene interreflections and scattering. The detector measures the returned signal and digitizes the temporal measurement into temporal wavefronts processed by the DSP. For low photon counts or path lengths above a few meters typical in automotive scenes, the binning process can be modeled as a Poisson random process [22, 41, 43, 54]. We consequently model the wavefront's number of photons $r^{(n)}$ detected within the integration time Δ in channel n's time bin k as

$$r^{(n)}[k] \sim \text{Poisson}\left(\int_{k\Delta}^{(k+1)\Delta} \psi^{(n)}(t)dt\right). \quad (2)$$

3.1. Transient Scene and Pulse Model

Rasshofer *et al.* [46] introduced a linear model for direct laser reflections in the LiDAR context for the incident transient response $H * g^{(n)}$ of Eq. (1). With this model, we formalize the transient response as

$$(H * g^{(n)})(R) = C \int_0^{2\tau^{(n)}} g^{(n)}(t) H\left(R - \frac{ct}{2}\right) dt, \quad (3)$$

where R is the distance between the sensor and the observed point, c is the speed of light, C is a proportionality constant, independent of t and R, describing the system, and $2\tau^{(n)}$ is the total pulse duration for channel n. We convert path length to time with $t = R/c$. Following Hahner *et al.* [19] and Rasshofer *et al.* [46], we adopt the following pulse shape

$$g^{(n)}(t) = \begin{cases} P_0^{(n)} \sin^2\left(\frac{\pi t}{2\tau^{(n)}}\right) & \text{if } 0 \le t \le 2\tau^{(n)} \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

where $P_0^{(n)}$ is channel n's pulse power magnitude. The transient scene response H embedded in Eq. (3) consists of the geometric attenuation of the light, proportional to $1/(2R)^2$, and the scene response. For a single opaque point object i, the latter is proportional to its reflectance ρ_i and a Dirac function $\delta(R - R_i)$, R_i being the object distance to the sensor. Reformulating Eq. (3) for a single echo from i yields

$$(H * g^{(n)})_i(R) = \begin{cases} f_i^{(n)}(R) & \text{if } R_i \le R \le R_i + c\tau^{(n)} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$\text{with } f_i^{(n)}(R) = \frac{C P_0^{(n)} \rho_i}{4R_i^2} \sin^2\left(\frac{\pi}{c\tau^{(n)}}(R - R_i)\right). \quad (6)$$

3.2. Object Reflectance Model

The object reflectance ρ_i depends on the material BRDF and the angle of incidence θ. We model reflectance using specular and diffuse components of the retroreflected portion of the Cook-Terrance model [9], that is,

$$\rho_i = \frac{\alpha^4 s \cos\theta}{4[\cos^2\theta(\alpha^4 - 1) + 1]^2 [\cos\theta(1-k) + k]^2} + d\cos\theta, \quad (7)$$

where s, d and $\alpha \in [0, 1]$ refer to the specular, diffuse and roughness properties of a surface material and $k = (\alpha + 1)^2/8$, see Supplemental Material. To render realistic textures without a large texture database, we approximate s, d, α through CARLA's Phong-like parameters s, d and α. These parameters are not directly accessible, so we extract them by projecting targeted hit points onto custom camera images encoding their values, encoding illustrated at the bottom right of Fig. 2. Specifically, we define a function $\mathcal{P}_i$ that, for a rendered image $I_{sd\alpha}$, returns pixel information at the location of projected point i, that is, $[s, d, \alpha] = \mathcal{P}_i(I_{sd\alpha})$.

3.3. Ambient Illumination

The ambient light $a(t)$ in Eq. (1) is modeled following Weng *et al.* [69]. A location i is projected on the red channel of a rendered RGB camera image in which shadows and reflections are properly accounted for; denote this image by I_{red}. We approximate $a(t)$ as a constant over waveform time bins, that is, $a(t) \equiv a_i = \mathcal{P}_i(I_{\text{red}})$, where a_i is independent of t. See Supplemental Material for details.

3.4. Multipath Transients

Multipath transients for laser beams hitting object discontinuities are taken as primary artifact sources in automotive scenarios. Multipath transients are modeled as linear combinations of neighboring waveforms. Specifically, a supersampled collection of $\{R_i\}$ and channels is computed using direct illumination only; then, for each LiDAR channel and each horizontal angle, a downsampled waveform $\psi_j^{(m)}$ is obtained:

$$\psi_j^{(m)}(R) = \sum_{i \in \mathcal{N}(j),\ n \in \mathcal{N}(m)} (K_i^{(n)} (H * g^{(n)})_i(R) + a_i), \quad (8)$$

where $\mathcal{N}(j)$ and $\mathcal{N}(m)$ define respectively the spatial neighborhood of the target point j and the channel m, and the $K_i^{(n)}$s are normalized weights that can be interpreted as a beam spatial intensity profile, see Supplemental Material.

3.5. LiDAR Sensing and DSP Model

A sensor-incident waveform $\psi_j^{(m)}$ is measured according to Eq. (2) and, along with sensor noise, subjected to clipping at a saturation level that depends on detector type. The DSP converts noisy and saturated waveforms $r_j^{(m)}$ into a point cloud $\mathbf{O}$ that includes range and intensity information. The entire process, from the emision of a laser pulse $g^{(m)}$ to the output of a processed point cloud $\mathbf{O}$ is shown in Fig. 3.

With compressed notation, we model LiDAR sensing as a function $\mathbf{\Phi}(\mathbf{\Theta})$ with hyperparameters $\mathbf{\Theta} = (P_0^{(m)}, \tau^{(m)}, V^{(m)})$. Laser power $P_0^{(m)}$ and pulse duration $\tau^{(m)}$ are functions of the channel m; they determine the emitted pulse $g^{(m)}$. First, the DSP denoises by convolving the measured waveform $r_j^{(m)}$ with the emitted pulse $g^{(m)}$, see [61]. Next, ambient light is estimated by removing the waveform's median from $r_j^{(m)}$. This allows the DSP

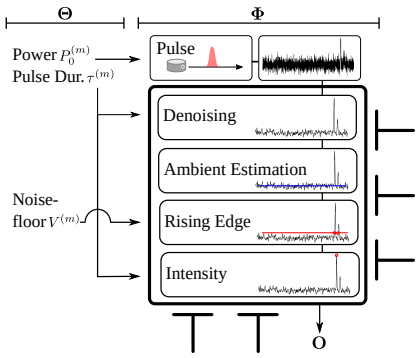

Figure 3. LiDAR Sensing and DSP Model $\mathbf{\Phi}$. The LiDAR sensing and DSP hyperparameters $\mathbf{\Theta}$, including pulse power, duration, and parameters for rising edge detection in a transient wavefront, affect the quality of the generated point cloud $\mathbf{O}$.

to find adequate noise thresholds $V^{(m)}$ since ambient light varies strongly throughout the scene. Typical DSPs use a rising edge detector [30] that finds peaks along the denoised waveform by identifying intersections with $V^{(m)}$. Multiple peaks may arise; the peak with the highest intensity is added to the point cloud $\mathbf{O}$. Finally, the maximum intensity is compensated for the emitted laser pulse: The pulse half width $\tau^{(m)}/2$ and power level $P_0^{(m)}$ are used as scaling factors to recover the true intensities, see Eq. 3.

Modern consumer LiDAR sensors have upward of 128 channels. Optimizing every channel hyperparameter individually is prohibitive. The Velodyne HDL-64 bundles lasers in two groups [62]; our LiDAR model likewise groups lower and upper lasers. Within each group, hyperparameter modulation is an affine function of the channel number with tuneable slope and bias. The edge threshold is modeled as a continuous parameter $V^{(m)} \in [0, 2]$. In contrast, power levels $P_0^{(m)}$ can only take one of 11 values, the lowest making the peak almost indistinguishable from ambient noise and the highest likely to saturate at close range. Pulse duration $\tau^{(m)}$ takes discrete values ranging from 3 to 15 ns in 1 ns increments. See Supplemental Material.

4. Optimization

We propose a multi-objective optimization (MOO) method that finds Pareto-optimal (see the Supplemental Material for a definition) LiDAR hyperparameters. Specifically, optimization returns high-quality point clouds when optimized for depth precision and optimal mAP when point clouds are fed to a downstream object detection module. We first formulate the optimization problem, then we present the proposed optimization method.

4.1. LiDAR Hyperparameter Optimization

We optimize hyperparameters with loss-driven end-to-end optimization. Like Robidoux *et al.* [48], we optimize the system as a whole, bundling hardware functionality, for example the laser beam power of each channel, with DSP functionality. The operation of the LiDAR imaging pipeline $\mathbf{\Phi}$ is modulated by P hyperparameters $\mathbf{\Theta} = (\Theta_1, \ldots, \Theta_\mathrm{P})$ with ranges of values normalized to the unit interval $[0, 1]$. With $\mathrm{T} \gg 2\tau$, we model each of the J channels Φ_j of $\mathbf{\Phi} = (\Phi_1, \ldots, \Phi_\mathrm{J})$ as

$$\Phi_j : [-\mathrm{T}, 0]^{[0,\infty)} \times [0, 1]^\mathrm{P} \to \mathbb{S}^{[0,\infty)}, \quad (\mathrm{r}_j, \mathbf{\Theta}) \mapsto \mathrm{O}_j, \tag{9}$$

O_j being a mapping from the unit sphere $\mathbb{S}$ (proxy for projective geometry) to nonnegative distances with 0 understood as "undefined" so that each Φ_j reconstructs a portion O_j of the overall point cloud $\mathbf{O}$ from a waveform r_j truncated to the time interval $[-\mathrm{T}, 0]$. The overall $\mathbf{\Theta}$-modulated LiDAR pipeline

$$\mathbf{\Phi} : (\mathbf{r}(H, \mathbf{\Theta}), \mathbf{\Theta}) \to \mathbf{O} \tag{10}$$

maps the set of truncated wave forms $\mathbf{r} = (\mathrm{r}_1, \ldots, \mathrm{r}_\mathrm{J})$ to the point cloud $\mathbf{O}$ that contains the compressed information available to downstream detectors about the changing scene H. We seek solutions of

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta} \in [0,1]^\mathrm{P}}{\arg\min}\, \boldsymbol{\mathcal{L}}(\mathbf{\Theta}), \tag{11}$$

which are Pareto-optimal with respect to the MOO loss vector $\boldsymbol{\mathcal{L}} = (\mathcal{L}_1, \ldots, \mathcal{L}_\mathrm{L})$. Loss components may not directly use $\mathbf{O}$; for example, $\mathcal{L}_l$ may use data tapped out of the pipeline (*e.g.*, a channel's waveform r_j) or the output of a downstream detector (*e.g.*, a deep CNN) which ingests $\mathbf{O}$ (*e.g.*, mAP). The *Pareto front* [28], which contains Pareto-optimal compromises between losses, is the solution set of Eq. (11), from which a single "champion" is selected using additional criteria.

4.2. Optimization Algorithm

To solve the MOO problem, we propose a variant of the Covariance Matrix Adaptation-Evolution Strategy methods of Mosleh *et al.* [36] and Robidoux *et al.* [48] that differs from the state of the art in a number of ways. It is listed in Algorithm 1. We use centroid weights that change between generations; to obtain better transients, we greedily override the CMA-ES centroid whenever a new Best So Far (BSF) is found, isolating the computation of CMA-ES statistics from the resulting "jump" as well as from the novel inclusion of centroids as BSF candidates; and we formulate a stable dynamic max-rank loss scalarization which we use to drive optimization and as *champion selection criterion*.

Stable Dynamic Max-Rank Loss Scalarization *Scalarizations* like the well-known *convex combination* $\sum_{l=1}^{\mathrm{L}} \mathrm{w}_l \mathcal{L}_l$ which boils down to the ℓ_1-norm of the loss vector $\boldsymbol{\mathcal{L}}$ with unit weights w_l, are used to combine multiple objectives so that a single objective optimizer yield MOO solutions [15]. Scalarization weights are difficult to choose when loss variations are not commensurate. The max-rank loss addresses this issue. In the context of a generation-based algorithm like Algorithm 1, Mosleh *et al.* [36] define a max-rank loss as follows. Let

$$\mathcal{R}_l^{q,m,n} = \text{rank of } \mathcal{L}_l^{q,m} \text{ within } \{\mathcal{L}_l^{r,o}\}_{r \in \{0,\ldots,4\mathrm{P}\}}^{o \in \{1,\ldots,n\}} \tag{12}$$

with ranks counted from 0 and loss component value ties resolved by left bisection. Then, the *weighted (left-bisection) max-rank loss* $\mathcal{M}^{q,m,n}$ *of the hyperparameter vector* $\boldsymbol{\Theta}^{q,m}$ *at the end of generation* n is

$$\mathcal{M}^{q,m,n} = \max_{l \in \{1,\dots,\mathrm{L}\}} \left(\mathrm{w}_l \cdot \mathcal{R}_l^{q,m,n} \right). \tag{13}$$

The max-rank loss is dynamic. For a given $\boldsymbol{\Theta}^{q,m}$ its values are monotone non-decreasing with respect to the addition of data. Because weights multiply ranks, they are non-dimensional, hence dial in the *relative importance* of loss components. While Mosleh *et al*. [36] scale each w_l by the (damped) running proportion of individuals that "fail" to pass a user-defined threshold, we find that such adaptive weigths break monotonicity. Instead, we keep w_ls fixed. We also improve on Mosleh *et al*. in that we average left and right bisection ranks (exception: if the left bisection rank is 0, the "average" is set to 0) to stabilize $\mathcal{M}^{q,m,n}$ with respect to loss value tie breaking (from, *e.g*., noise) and creation (quantization); this defines our *stable (dynamically monotone) max-rank loss scalarization*.

Dual-Weight CMA-ES Besides more refined seatbelting (see Supplemental Material), Algorithm 1 differs from earlier CMA-ES in its use of non-constant (hyperparameter, not loss) *centroid* weights. Although variable CMA-ES generation sizes are common [39], the formula used to derive centroid weights is invariably kept fixed; in contrast, the proposed CMA-ES alternates between gradient-seeking centroid weights [36], which assign zero weight to the worst quartile of each generation instead of the usual half to exploit the symmetry of the second and third quartiles of Gaussian distributions to get a more accurate gradient approximation, and boundary-stabilizing centroid weights [48] *with no discard* so that further exploration not go in the wrong direction near generic local minima.

Other novel components are that the loss of the weighted centroid of every generation is evaluated (standard CMA-ES only generate Gaussian clouds with them) and the greedy branch in Algorithm 1 (Lines 18–20): If any individual of the generation, weighted centroid included, turns out to be a strict minimizer, it becomes the next generation Gaussian cloud center.

5. Experiments

In the following, we first validate the proposed approach with the LiDAR simulation model, jointly optimizing depth estimation and downstream 3D object detection within simulated scenes. Then, we compare the proposed optimization algorithm to recent 0th-order solvers. Finally, we present experiments with an off-the-shelf hardware LiDAR unit.

Experimental Setup As described in Sec. (3.5), hyperparameters affect the wavefront *and* DSP. We optimize the DSP rising edge threshold $V^{(m)}$ and 10 LiDAR hyperparameters that control low-level sensing parameters including the laser power $P_0^{(m)}$ and laser pulse width $\tau^{(m)}$ for each of the 128 channels.

Algorithm 1 LiDAR Hyperparameter Optimization.

Require: LiDAR $\boldsymbol{\Phi}$, $\boldsymbol{\Theta} \in [0,1]^{\mathrm{P}}$ (initial hyperparameter vector), $\mathrm{N} \in \mathbb{N}^*$ (number of generations), $\varepsilon \in \left(0, \frac{1}{3}\right)$ (small bound), $\mathbf{C} \in \mathbb{R}^{\mathrm{P}\times\mathrm{P}}$ (CMA-ES "directional" covariance matrix factor), $\sigma \in \left[\varepsilon, \frac{1}{3}\right]$ (square root of covariance matrix "scale" factor)

1: $\mathbf{p} \leftarrow \mathbf{0}$, $\mathbf{c} \leftarrow \mathbf{0}$ (CMA-ES path vectors), $\boldsymbol{\Theta}_{\text{center}} \leftarrow \boldsymbol{\Theta}$
2: **for** $n = 1$ **to** N **do**
3: $\quad\boldsymbol{\Theta}^{0,n} \leftarrow \boldsymbol{\Theta}$
4: $\quad\boldsymbol{\mathcal{L}}^{0,n} \leftarrow$ losses for LiDAR $\boldsymbol{\Phi}$ modulated by $\boldsymbol{\Theta}^{0,n}$
5: $\quad$**for** $p = 1$ **to** 4P **do**
6: $\quad\quad\boldsymbol{\Theta}^{p,n} \leftarrow$ random draw from Gaussian distribution with covariance matrix $\sigma^2\mathbf{C}$ centered at $\boldsymbol{\Theta}_{\text{center}}$
7: $\quad\quad\boldsymbol{\Theta}^{p,n} \leftarrow \boldsymbol{\Theta}^{p,n} +$ Gaussian distribution with diagonal cov. matrix proportional to square of quantization grain [48]
8: $\quad\quad\boldsymbol{\Theta}^{p,n} \leftarrow \boldsymbol{\Theta}^{p,n}$ reflected back into $[0,1]^{\mathrm{P}}$
9: $\quad\quad\boldsymbol{\mathcal{L}}^{p,n} \leftarrow$ losses for LiDAR $\boldsymbol{\Phi}$ modulated by $\boldsymbol{\Theta}^{p,n}$
10: $\quad$**end for**
11: $\quad$Compute $\{\mathcal{M}^{q,m,n}\}_{q\in\{0,\dots,4\mathrm{P}\},m\in\{1,\dots,n\}}$ by including $\{\boldsymbol{\mathcal{L}}^{p,n}\}_{p\in\{0,\dots,4\mathrm{P}\}}$ in rank computations
12: $\quad$Use "eager" [36] centroid weights with $\lambda = 4\mathrm{P}$, $\mu = 3\mathrm{P}$
13: $\quad$**if** n is odd **then**
14: $\quad\quad$Use "stable" [48] centroid weights with $\lambda = \mu = 4\mathrm{P}$
15: $\quad$**end if**
16: $\quad$Standard CMA-ES update [21] of $\boldsymbol{\Theta}$, σ, $\mathbf{C}$, $\mathbf{p}$, $\mathbf{c}$ based on $\{\boldsymbol{\Theta}^{p,n}\}_{p\in\{1,\dots,4\mathrm{P}\}}$ and $\{\mathcal{M}^{p,n,n}\}_{p\in\{1,\dots,4\mathrm{P}\}}$
17: $\quad\boldsymbol{\Theta}_{\text{center}} \leftarrow \boldsymbol{\Theta}$
18: $\quad$**if** $\min_{p\in\{0,\dots,4\mathrm{P}\}} \mathcal{M}^{p,n,n} < \min_{q\in\{0,\dots,4\mathrm{P}\},m\in\{1,\dots,n\}} \mathcal{M}^{q,m,n}$ **then**
19: $\quad\quad\boldsymbol{\Theta}_{\text{center}} \leftarrow$ minimizer closest to centroid of minimizers
20: $\quad$**end if**
21: **end for**
22: **return** $\boldsymbol{\Theta}^{p,n}$ in the (guaranteed nonempty) intersection of the Pareto front and the set of minimizers of $\mathcal{M}^{q,m,N}$, with ties resolved by choosing the one closest to their centroid and remaining ties resolved by maximizing n, then p

5.1. Optimization for Depth and Intensity

Next, we assess the method by optimizing point clouds for depth and intensity with the proposed optimizer. We minimize the average RMSE of the depth and the average RMSE of the intensity, specifically

$$\mathcal{L}_{\text{depth}}(\boldsymbol{\Theta}) = \tfrac{1}{\mathrm{F}} \textstyle\sum_{f=1}^{\mathrm{F}} \mathrm{RMSE}\left(\mathrm{R}_f, \Phi_f^{(\mathrm{R})}(\boldsymbol{\Theta})\right), \text{ and} \tag{14}$$

$$\mathcal{L}_{\text{int.}}(\boldsymbol{\Theta}) = \tfrac{1}{\mathrm{F}} \textstyle\sum_{f=1}^{\mathrm{F}} \mathrm{RMSE}\left(\mathrm{I}_f, \Phi_f^{(\mathrm{I})}(\boldsymbol{\Theta})\right), \tag{15}$$

with F the number of frames in the validation set, see Supplemental Material. The depth loss $\mathcal{L}_{\text{depth}}$ rewards accurate point cloud depth estimates over the full range, whereas $\mathcal{L}_{\text{int.}}$ ensures that accurate intensities are measured with the pulse power $P_0^{(j)}$. High output power generally results in more accurate point clouds at farther distances but also leads to excessive saturation; $\mathcal{L}_{\text{int.}}$ penalizes saturation.

Fig. 4 shows a comparison between ground-truth, expert-tuned and optimized PCs for a Pareto point that mini-

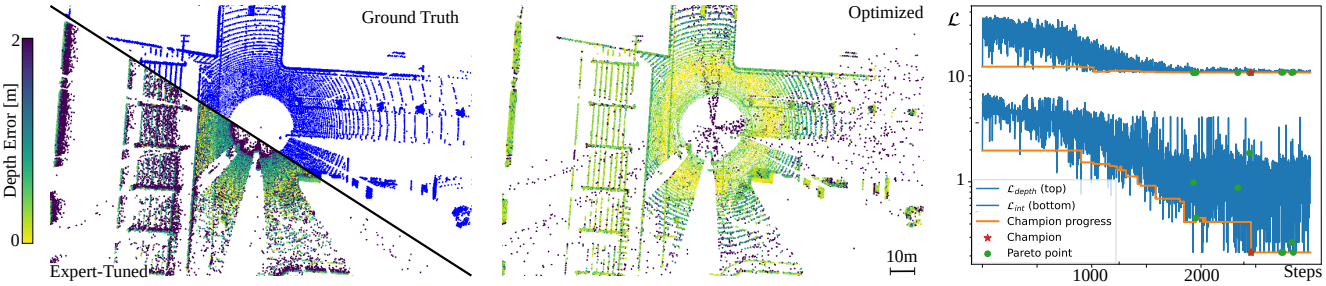

Figure 4. Comparison between expert-tuned and optimized point clouds. Colors encode the individual depth error of each point (clipped at 2 m). Convergence plot shows the loss components $\mathcal{L}_{\text{depth}}$ and $\mathcal{L}_{\text{int.}}$ (Eq. (14)-(15)) against optimization step (3000 loss evaluations total), the evolution of the champion and the final Pareto front, champion included.

mizes the depth error. Compared to the optimized point cloud measurement, the expert-tuned point cloud suffers from clutter, and its depth error increases with distance. The results reported in Table 1 validate the optimization method: The expert-tuned configuration has a loss vector $\boldsymbol{\mathcal{L}} = (\mathcal{L}_{\text{depth}}, \mathcal{L}_{\text{int.}})$ equal to $(12.195, 1.971)$; after 3000 evaluations of the objectives, the champion of the proposed method has a loss vector of $(10.754, 0.216)$, improving depth and intensity metrics by 12% and 89%, respectively. We note that biasing optimization toward reducing $\text{RMSE}_{\text{depth}}$ more than $\text{RMSE}_{\text{int.}}$ can be done by changing Eq. (13) loss weights w_l. In this work, loss component weights were kept at their default 1.

Table 1 also shows a comparison of the proposed algorithm with other state-of-the-art MOO optimizers. With both champion selection criteria—the one proposed by Mosleh *et al.* [36], namely "last Pareto point of the run", and our own, "Pareto point that minimizes the stable (weighted) max-rank loss"—Algorithm (1) is the best performer.

Solver	ℓ_1-norm (↓) last Pareto point [36]	ℓ_1-norm (↓) proposed champion
Expert-Tuned	14.091	14.091
Proposed	**11.089**	**10.995**
NSGA-III [10, 11]	***11.408***	11.574
Mosleh *et al.* [36]	*11.497*	11.497
AGE-MOEA [42]	11.547	***11.427***
C-TAEA [29]	11.622	11.461
RVEA [8]	11.851	*11.437*
R-NSGA-III [63]	12.306	11.619
U-NSGA-III [51]	13.423	11.512
SMS-EMOA [2]	13.658	11.750

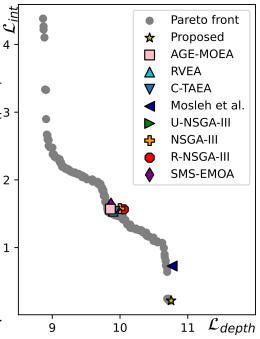

Table 1. LiDAR Depth and Intensity MOO. The Pareto front (see figure) is approximately a straight line with slope -1. So, the variations of $\mathcal{L}_{\text{depth}}$ and $\mathcal{L}_{\text{int.}}$ are approximately balanced and the ℓ_1-norm of the loss vector is a reasonable evaluation metric. For all solvers, the first column shows the ℓ_1-norm of the last Pareto point of the run (champion per Mosleh *et al.* [36]); the last column instead uses the proposed champion selection method (Line 22 of Algorithm (1)). The proposed solver performs favorably; in addition, the proposed champion selection method improves results for all but two solvers (there is one tie). Median of three runs with 3000 loss evaluations; see the Supplemental Material for details.

5.2. Optimization for Object Detection

Next, we investigate optimization for object detection and classification. We use Average Precision (AP) as additional optimization objective. AP is maximized for cars and pedestrians at 40 recall positions [55] over an optimization set with $\text{F} = 100$ frames. With standard KITTI IoU thresholds [17], the CV loss is

$$\mathcal{L}_{\text{obj,\{car, ped.\}}}(\boldsymbol{\Theta}) = -\text{AP}_{\text{\{car, ped.\}}}(\boldsymbol{\Phi}(\boldsymbol{\Theta})), \qquad (16)$$

evaluated over a 0-80 m range. We train a PV-RCNN [53] for 3D object detection on 5900 full range point clouds collected from the simulation environment with 8 different expert-tuned parameterizations $\boldsymbol{\Theta}$, see training and dataset details in the Supplemental Material.

Table 2 contains a quantitative comparison between the proposed method and expert tuning. Compared to expert tuning, the proposed method increases AP by 52.9% and 27.2% for cars and pedestrians, respectively, on the full 0-80 m range, with large increases predominately at close range. Lower AP values in the 0-30 m range compared to 30-50 m are due to a lower object occurrence (3.2 per frame vs. 3.6) and a less varied distribution of object yaw angles (std. dev. of 0.63 rad vs. 1.25 rad) within the training set. Consequently, the detector can better adapt within the 30-50 m bin. The quantitative findings agree with the qualitative results of Fig. 5, where optimized and expert-tuned point clouds are compared to the ground truth. Here, the clutter in the expert-tuned point cloud results in false positives and missed detections. In these examples, the proposed method misses only one pedestrian, whereas on the expert-tuned PC the detector misses them all. We also compare the average depth RMSE before and after filtering suppressed points; the low loss of the filtered PC suggests that clutter has been removed and that the detector prefers an accurate if sparser point cloud. Note however that a DSP optimized for the removal of clutter, like the proposed method, also suppresses ground points, with a small impact on object detection results, see Table 2 last column.

5.3. Off-the-shelf LiDAR Optimization

In this section, we adapt the proposed optimization algorithm to an off-the-shelf Baraja Spectrum LiDAR sensor. Only a handful of hyperparameters are user-accessible: the

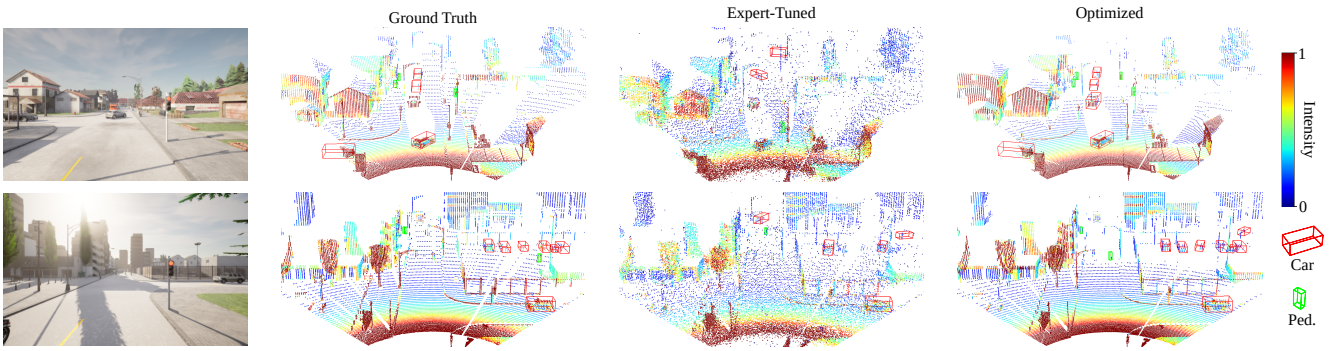

Figure 5. Comparison between ground-truth, expert-tuned and optimized point cloud for 3D object detection. Optimization results in clutter-free point clouds which helps reduce false positive detections. For visualization purposes only the camera field-of-view is shown.

	mAP (↑)				AP_{car} (↑)				$AP_{ped.}$ (↑)				$RMSE_{depth}$ (↓)	
Method	0-80 m	0-30 m	30-50 m	50-80 m	0-80 m	0-30 m	30-50 m	50-80 m	0-80 m	0-30 m	30-50 m	50-80 m		filtered
Expert	7.33	10.88	10.18	1.28	14.58	21.75	20.19	2.53	0.07	0.02	0.17	0.04	**9.12**	6.17
Proposed	**46.86**	**52.14**	**58.70**	**27.08**	**66.47**	**75.06**	**84.86**	**41.20**	**27.25**	**29.22**	**32.53**	**12.95**	12.87	**5.13**
Margin	+39.53	+41.26	+48.52	+25.80	+51.89	+53.31	+64.67	+38.67	+27.18	+29.20	+32.36	+12.91	+3.75	-1.04

Table 2. Optimization for 3D Object Detection. We report mAP and AP in different distance range bins, and unfiltered and filtered RMSE, for expert-tuned and proposed. Filtered $RMSE_{depth}$ excludes all points suppressed by the DSP. Best in **bold**.

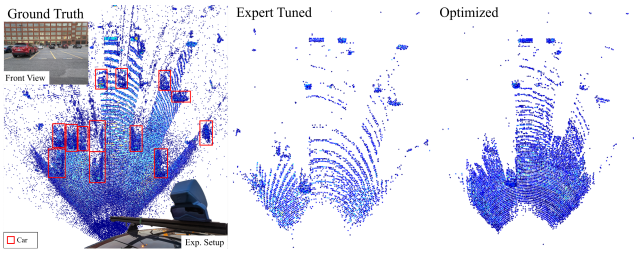

Figure 6. Left: ground truth 3D histogram used to compute losses, see text, red bounding boxes denoting cars, with experimental setup zoom-ins: captured static scene (top) and LiDAR installation (bottom); the LiDAR detector is the blue unit. Middle: loss contributions for the expert-tuned parameters and Right: for optimized parameters. Brighter color means a smaller loss.

return mode that selects the waveform peaks used to generate the point cloud, the sensor heads scanning pattern, and the two sensor head motor frequencies which determine the PC angular resolution vs. sensor noise balance. Because raw wavefronts are inaccessible, we optimize 3D point cloud histograms taken over an (intermittently) static scene such as a workplace parking lot. Fig. 6 shows the experimental setup along with a photograph of the scene over which optimization was performed. The ground truth histogram is shown on the left of Fig. 6; it was generated by averaging 100 captures with a uniform angular resolution scanning pattern. Expert-tuned and optimized histogram contributions to the fitness function (described in the Supplemental Material) are respectively shown in the middle and right of Fig. 6; brighter bins indicate larger contributions. Optimized hyperparameters achieved better results by focusing laser scans in regions of interests (ROIs) where the ground truth was denser; expert-tuned ones produced more uniform point clouds. Using the RMSE of PC histograms weighted by distance and number of points per bin, we estimate a 10.1 cm error for the expert-tuned configuration vs. 3.0 cm with optimized hyperparameters.

6. Conclusion

We propose an in-the-loop black box 0th-order optimization method for LiDAR sensing pipelines that finds optimal parameters for depth and intensity estimation and 3D object detection and classification. To assess our method, we propose a novel LiDAR simulation method integrated into the CARLA simulator. We find that optimizing the LiDAR sensing pipeline can significantly improve depth, intensity and object detection and classification compared to manual expert tuning. Specifically, for 3D object detection our optimization method results in a major increase of 51.9% AP for cars and 27.2% AP for pedestrians, compared to fine-tuning a detector on expert-optimized LiDAR vendor parameters. An interesting direction for future research is real-time scene-dependent optimization of LiDAR scanning parameters, potentially leading to adaptive sensing in adverse weather in urban and highway scenarios.

Acknowledgments This work was supported by the AI-SEE project with funding from the FFG, BMBF, and NRC-IRA. We thank the Federal Ministry for Economic Affairs and Energy for support through the PEGASUS-family project *VVM-Verification and Validation Methods for Automated Vehicles Level 4 and 5*. Felix Heide was supported by an NSF CAREER Award (2047359), a Packard Foundation Fellowship, a Sloan Research Fellowship, a Sony Young Faculty Award, a Project X Innovation Award, and an Amazon Science Research Award.

References

[1] Behnam Behroozpour, Phillip A. M. Sandborn, Ming C. Wu, and Bernhard E. Boser. Lidar System Architectures and Circuits. *IEEE Communications Magazine*, 55(10):135–142, Oct. 2017. Conference Name: IEEE Communications Magazine. 3

[2] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, Sept. 2007. 7

[3] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[4] Mario Bijelic, Tobias Gruber, and Werner Ritter. A benchmark for LiDAR sensors in fog: Is detection breaking down? In *IEEE Intelligent Vehicles Symposium (IV)*, 2018. 1, 2

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars, Apr. 2016. arXiv:1604.07316 [cs]. 3

[6] Alexander Carballo, Jacob Lambert, Abraham Monrroy Cano, David Robert Wong, Patiphon Narksri, Yuki Kitsukawa, Eijiro Takeuchi, Shinpei Kato, and Kazuya Takeda. LIBRE: The multiple 3D LiDAR dataset. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020. 1

[7] M. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3D tracking and forecasting with rich maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[8] Ran Cheng, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791, Oct. 2016. Conference Name: IEEE Transactions on Evolutionary Computation. 7

[9] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM SIGGRAPH Computer Graphics*, 15(3):307–316, Aug. 1981. 4

[10] K. Deb and J. Sundar. Reference point based multi-objective optimization using evolutionary algorithms | Proceedings of the 8th annual conference on Genetic and evolutionary computation. 7

[11] Kalyanmoy Deb and J. Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 635–642, New York, NY, USA, July 2006. Association for Computing Machinery. 7

[12] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Motion-based detection and tracking in 3D LiDAR scans. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016. 1

[13] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 1–16. PMLR, 13–15 Nov 2017. 2

[14] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *1st Annual Conference on Robot Learning*, 2017. 3

[15] Michael T. M. Emmerich and André H. Deutz. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing*, 17(3):585–609, Sept. 2018. 5

[16] Jin Fang, Dingfu Zhou, Feilong Yan, Tongtong Zhao, Feihu Zhang, Yu Ma, Liang Wang, and Ruigang Yang. Augmented LiDAR Simulator for Autonomous Driving. *IEEE Robotics and Automation Letters*, 5(2):1931–1938, Apr. 2020. arXiv:1811.07112 [cs]. 3

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 7

[18] Christopher Goodin, Daniel Carruth, Matthew Doude, and Christopher Hudson. Predicting the influence of rain on LiDAR in ADAS. *Electronics*, 8, 2019. 1

[19] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. LiDAR Snowfall Simulation for Robust 3D Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3, 4

[20] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real LiDAR point clouds for 3D object detection in adverse weather. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 3

[21] Nikolaus Hansen. The CMA evolution strategy: A tutorial, 2016. 2, 6

[22] Felix Heide, Steven Diamond, David B Lindell, and Gordon Wetzstein. Sub-picosecond photon-efficient 3d imaging using single-photon sensors. *Scientific reports*, 8(1):1–8, 2018. 3, 4

[23] Robin Heinzler, Philipp Schindler, Jürgen Seekircher, Werner Ritter, and Wilhelm Stork. Weather influence and classification with automotive LiDAR sensors. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019. 1

[24] Braden Hurl, Krzysztof Czarnecki, and Steven Waslander. Precise Synthetic Image and LiDAR (PreSIL) Dataset for Autonomous Vehicle Perception, May 2019. arXiv:1905.00160 [cs]. 1, 3

[25] Maria Jokela, Matti Kutila, and Pasi Pyykönen. Testing and validation of automotive point-cloud sensors in adverse weather conditions. *Applied Sciences*, 9, 2019. 1

[26] Velat Kilic, Deepti Hegde, Vishwanath Sindagi, A. Brinton Cooper, Mark Foster, and Vishal Patel. LiDAR light scattering augmentation (LISA): physics-based simulation of adverse weather conditions for 3D object detection. *arXiv preprint 2107.07004*, 2021. 1, 3

[27] Matti Kutila, Pasi Pyykönen, Maria Jokela, Tobias Gruber, Mario Bijelic, and Werner Ritter. Benchmarking automotive LiDAR performance in arctic conditions. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020. 1

[28] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282, 2002. 5

[29] Ke Li, Renzhi Chen, Guangtao Fu, and Xin Yao. Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 23(2):303–315, Apr. 2019. Conference Name: IEEE Transactions on Evolutionary Computation. 7

[30] Xiaolu Li, Bingwei Yang, Xinhao Xie, Duan Li, and Lijun Xu. Influence of waveform characteristics on lidar ranging accuracy and precision. *Sensors*, 18(4), 2018. 5

[31] Ievgeniia Maksymova, Christian Steger, and Norbert Druml. Review of LiDAR Sensor Data Acquisition and Compression for Automotive Applications. *Proceedings*, 2(13):852, 2018. Number: 13 Publisher: Multidisciplinary Digital Publishing Institute. 3

[32] Yunze Man, Xinshuo Weng, Prasanna Kumar Sivakumar, Matthew O'Toole, and Kris Kitani. Multi-echo lidar for 3d object detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3743–3752, 2021. 3

[33] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11164–11173, 2020. 3

[34] Sebastien Michaud, Jean-François Lalonde, and Philippe Giguère. Towards characterizing the behavior of LiDARs in snowy conditions. In *Workshop on Planning, Perception and Navigation for Intelligent Vehicles, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. 1

[35] Anish Mittal, Anush K. Moorthy, and Alan C. Bovik. Automatic parameter prediction for image denoising algorithms using perceptual quality features. In *Human Vision and Electronic Imaging XVII*, volume 8291, pages 110–116. SPIE, Feb. 2012. 2

[36] Ali Mosleh, Avinash Sharma, Emmanuel Onzon, Fahim Mannan, Nicolas Robidoux, and Felix Heide. Hardware-in-the-Loop End-to-End Optimization of Camera Image Processing Pipelines. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7526–7535, Seattle, WA, USA, June 2020. IEEE. 2, 5, 6, 7

[37] Matthias Müller, Vincent Casser, Jean Lahoud, Neil Smith, and Bernard Ghanem. Sim4CV: A Photo-Realistic Simulator for Computer Vision Applications, Mar. 2018. arXiv:1708.05869 [cs]. 3

[38] Kazuki Nakamura, Kenji Narumi, Kohei Kikuchi, and Yasuhisa Inada. Liquid crystal-tunable optical phased array for LiDAR applications. In *Smart Photonic and Optoelectronic Integrated Circuits XXIII*, volume 11690, pages 94–99. SPIE, Mar. 2021. 3

[39] Kouhei Nishida and Youhei Akimoto. Population Size Adaptation for the CMA-ES Based on the Estimation Accuracy of the Natural Gradient. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, pages 237–244, New York, NY, USA, July 2016. Association for Computing Machinery. 6

[40] Jun Nishimura, Timo Gerasimow, Sushma Rao, Aleksandar Sutic, Chyuan-Tyng Wu, and Gilad Michael. Automatic ISP image quality tuning using non-linear optimization, Feb. 2019. arXiv:1902.09023 [cs]. 2

[41] Matthew O'Toole, Felix Heide, David B. Lindell, Kai Zang, Steven Diamond, and Gordon Wetzstein. Reconstructing Transient Images from Single-Photon Sensors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2289–2297, Honolulu, HI, July 2017. IEEE. 3, 4

[42] Annibale Panichella. An adaptive evolutionary algorithm based on non-euclidean geometry for many-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, pages 595–603, New York, NY, USA, July 2019. Association for Computing Machinery. 7

[43] Adithya K. Pediredla, Aswin C. Sankaranarayanan, Mauro Buttafava, Alberto Tosi, and Ashok Veeraraghavan. Signal Processing Based Pile-up Compensation for Gated Single-Photon Avalanche Diodes, June 2018. arXiv:1806.07437 [physics]. 3, 4

[44] Luke Pfister and Yoram Bresler. Learning Filter Bank Sparsifying Transforms. *IEEE Transactions on Signal Processing*, 67(2):504–519, Jan. 2019. Conference Name: IEEE Transactions on Signal Processing. 2

[45] Francesco Pittaluga, Zaid Tasneem, Justin Folden, Brevin Tilmon, Ayan Chakrabarti, and Sanjeev J. Koppal. Towards a MEMS-based Adaptive LIDAR, Oct. 2020. arXiv:2003.09545 [cs, eess]. 3

[46] Ralph Rasshofer, Martin Spies, and Hans Spies. Influences of weather phenomena on automotive laser radar systems. *Advances in Radio Science*, 9, 2011. 3, 4

[47] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for Benchmarks, Sept. 2017. arXiv:1709.07322 [cs]. 3

[48] Nicolas Robidoux, Luis E. Garcia Capel, Dong-eun Seo, Avinash Sharma, Federico Ariza, and Felix Heide. End-to-End High Dynamic Range Camera Pipeline Optimization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6297–6307, 2021. 2, 5, 6

[49] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, June 2016. ISSN: 1063-6919. 3

[50] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. 1

[51] Haitham Seada and Kalyanmoy Deb. A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives. *IEEE Transactions on Evolutionary Computation*, 20(3):358–369, June 2016. Conference Name: IEEE Transactions on Evolutionary Computation. 7

[52] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles, July 2017. arXiv:1705.05065 [cs]. 3

[53] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-voxel feature set abstraction for 3D object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 7

[54] Dongeek Shin. *Computational imaging with small numbers of photons*. Thesis, Massachusetts Institute of Technology, 2016. Accepted: 2016-07-18T20:05:44Z. 3, 4

[55] Andrea Simonelli, Samuel Rota Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3D object detection. *arXiv preprint arXiv:1905.12365*, 2019. 7

[56] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[57] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. SHIFT: a synthetic driving dataset for continuous multi-task domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21371–21382, June 2022. 1, 3

[58] Maxime Tremblay, Shirsendu S. Halder, Raoul de Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision (IJCV)*, 126, 2021. 1

[59] Ethan Tseng, Ali Mosleh, Fahim Mannan, Karl St-Arnaud, Avinash Sharma, Yifan Peng, Alexander Braun, Derek Nowrouzezahrai, Jean-Francois Lalonde, and Felix Heide. Differentiable compound optics and processing pipeline optimization for end-to-end camera design. *ACM Transactions on Graphics (TOG)*, 40(2):1–19, 2021. 2

[60] Ethan Tseng, Felix Yu, Yuting Yang, Fahim Mannan, Karl ST. Arnaud, Derek Nowrouzezahrai, Jean-François Lalonde, and Felix Heide. Hyperparameter optimization in black-box image processing using differentiable proxies. *ACM Transactions on Graphics*, 38(4):1–14, Aug. 2019. 2

[61] G. Turin. An introduction to matched filters. *IRE Transactions on Information Theory*, 6(3):311–329, 1960. 4

[62] Velodyne. Hdl-64e user's manual, 2007. 5

[63] Yash Vesikar, Kalyanmoy Deb, and Julian Blank. Reference Point Based NSGA-III for Preferred Solutions. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1587–1594, Nov. 2018. 7

[64] F Villa, B Markovic, S Bellisai, D Bronzi, A Tosi, F Zappa, S Tisa, D Durini, S Weyers, U Paschen, et al. SPAD smart pixel for time-of-flight and time-correlated single-photon counting measurements. *IEEE Photonics Journal*, 4(3):795–804, 2012. 1

[65] Niclas Vödisch, Ozan Unal, Ke Li, Luc Van Gool, and Dengxin Dai. End-to-End Optimization of LiDAR Beam Configuration for 3D Object Detection and Localization. *IEEE Robotics and Automation Letters*, 7(2):2242–2249, Apr. 2022. Conference Name: IEEE Robotics and Automation Letters. 3

[66] Andrew Wallace, Abderrahim Halimi, and Gerald Buller. Full waveform LiDAR for adverse weather conditions. *IEEE Transactions on Vehicular Technology*, 69, 2020. 1

[67] Dingkang Wang, Connor Watkins, and Huikai Xie. MEMS mirrors for LiDAR: A review. *Micromachines*, 11(5), 2020. 1

[68] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. AdvSim: Generating Safety-Critical Scenarios for Self-Driving Vehicles, Jan. 2022. arXiv:2101.06549 [cs]. 3

[69] Xinshuo Weng, Yunze Man, Jinhyung Park, Ye Yuan, Dazhi Cheng, Matthew O'Toole, and Kris Kitani. All-In-One Drive: A Large-Scale Comprehensive Perception Dataset with High-Density Long-Range Point Clouds. *arXiv*, 2021. 1, 3, 4

[70] George Williams. Optimization of eyesafe avalanche photodiode LiDAR for automobile safety and autonomous navigation systems. *Optical Engineering*, 56(3):1 – 9, 2017. 1

[71] Chyuan-Tyng Wu, Leo F. Isikdogan, Sushma Rao, Bhavin Nayak, Timo Gerasimow, Aleksandar Sutic, Liron Ainkedem, and Gilad Michael. VisionISP: Repurposing the image signal processor for computer vision applications. In *IEEE International Conference on Image Processing (ICIP)*, pages 4624–4628, 2019. 2

[72] Ji Zhang and Sanjiv Singh. LOAM : LiDAR odometry and mapping in real-time. *Robotics: Science and Systems Conference (RSS)*, 2014. 1

[73] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11:712–731, 2007. 2