

Analysis of Methods to Improve Underwater Object Detection in Various Environments

Nico Katz

La Jolla Country Day School
San Diego, California, USA
1npk412@gmail.com

Abstract

We evaluate how single-augmentation and aggregate-augmentation training affect the robustness of a Faster R-CNN detector on an underwater target-detection task, and we assess a physics-based dehazing stage built on ten synthetic attenuation levels ordered as the Jerlov water types are. Seven detectors, one of them a control, were tested on a clean split, a composite degradation, each degradation alone, and four degradations held out of every training policy. All seven fall within 0.805–0.841 AP on clean data, so clean validation does not discriminate between augmentation policies. The aggregate model's lead over the control is the same on trained-on and held-out degradations (0.026 AP each) and reappeared in two retrained seeds per policy (0.019 and 0.021 AP), though two seeds cannot establish reproducibility; its composite-condition gap (0.186 AP) did not reproduce on retraining (0.406 and 0.412). The aggregate is not the best of the seven on held-out degradations, an ordering that includes checkpoints predating per-worker data-loader seeding, and its advantage is concentrated in two blur-like conditions on the validation split (83 %) but not on a held-back 97-image partition (44 %). The physics inversion alone raises AP from 0.691 to 0.819 across three conditions and beats a Dark Channel Prior baseline, whereas adding contrast enhancement and white balance lowers it to 0.680; this comparison favors our method by construction, since the inversion is given the true attenuation level and path length, the frames are generated by the equation it inverts, and the coefficients are our own. Across the full sweep, which used the three-stage pipeline, correction gains 0.197 AP in coastal water and costs 0.117 AP in open ocean. Augmentation and correction approximately add for the aggregate policy but overlap for the other five. Clean validation performance may not predict robustness under degraded conditions.

Keywords

Computer Vision, Underwater, Jerlov Water Types, Data Augmentations

ACM Reference Format:

Nico Katz. 2026. Analysis of Methods to Improve Underwater Object Detection in Various Environments. In *Proceedings of International Journal of Secondary Computing and Applications Research (IJSCAR VOL. 3, ISSUE 4)*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.67149/yhjs2024.5/myab98j1>

This paper is published under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International (CC-BY-NC-ND 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

IJSCAR VOL. 3, ISSUE 4

© 2026 The Author(s). Published by IJSCAR.

1 Introduction

Underwater computer vision is affected by two effects that are largely absent from terrestrial imaging [1], [2]. First, water attenuates light wavelength-selectively: red is absorbed within a few meters, while blue persists, producing the characteristic blue-green hue of water and progressive loss of chromatic contrast with distance [1], [2], [3]. Second, suspended particulate matter scatters ambient light directly into the camera path, adding a veiling glow that reduces contrast independently of the subject [1], [2]. Both effects worsen with distance and with the optical properties of the water body itself [1], [2]. These conditions pose a practical problem for detecting targets reliably across widely varying visibility; benchmark collections of real underwater photographs document how wide that variation is [4]. Two mitigation strategies are implemented in this study. One is data augmentation: adjust the training images so the detector learns representations similar to the degradations it will encounter. The data augmentations used in this study will be presented in section three.

The other is image enhancement/dehazing, which attempts to invert the physical degradation before detection, so that the detector receives an approximation of the original image. This report evaluates both methods to answer the following:

- (1) Which individual data augmentations provide the most improvement in underwater object detection where image clarity is reduced?
- (2) How effective is image enhancement/dehazing at improving model performance in underwater computer vision?

2 Related Work

Using Jerlov water types to approach underwater computer vision problems is not a novel approach [5], [6], [7]. Many works use physics based simulation involving Jerlov Water Types for underwater computer vision [5], [6], [7]. For example, Mondal et al. [6], in a preprint, proposed a physics-informed and data-driven framework for simulating underwater images, combining a simplified physical image-formation model with learned residual degradation. Others, such as Uplavikar et al. [5], have used Jerlov Water Types as a way to add nuance to various underwater environments, and Li, Anwar and Porikli [7] synthesize training data across the Jerlov types in the same way.

Two further bodies of work bear on what we do here. The first is data augmentation itself, which Shorten and Khoshgoftaar [8] survey. Hendrycks and Dietterich [9] test a fixed model against a catalog of synthetic corruptions instead of a clean split, which is the idea behind the held-out condition we introduce in Section 3.5. The second is underwater detection benchmarks. DUO [10] and RUOD [11] both provide real underwater imagery with object annotations

and would be the natural outside comparison for a study like this one; we do not evaluate on them here, for the reason given in Section 5. On the enhancement side the standard non-physical baseline is the Dark Channel Prior of He, Sun and Tang [12], which we adopt as a comparison method in Section 4.9; learned alternatives include the analysis-synthesis network of Wang et al. [13], also a preprint, which we do not evaluate here because it would require a trained model we do not have.

The contribution of this study is therefore not the introduction of a new Jerlov-based dehazing method. Instead, this study evaluates two distinct approaches to improving underwater object-detection robustness: detector-side data augmentation and physics-based image correction. We systematically compare individual and aggregate augmentation policies under controlled composite degradation and separately evaluate a Jerlov-parameterized dehazing stage across water types and simulated path lengths. This allows us to examine not only whether these techniques improve detection, but also whether performance on clean validation data is predictive of robustness under degraded conditions.

3 System

3.1 Underwater image formation and the Jerlov classification

The Jerlov Water Type classification system categorizes water according to optical clarity and attenuation characteristics. The two key sub-classifications of Jerlov Water Types are Coastal and Open Ocean Water Types. Open Ocean Types range across I, IA, IB, II, and III, with I being the most clear Water Type. Coastal Types range across 1C, 3C, 5C, 7C, and 9C, with 9C being the most turbid or unclear Water Type. Figure 1 shows one frame from our dataset simulated at each of these water types.

In order to dehaze images using Jerlov Water Types we use the following equations [1], [2], [3]. Let $J_c(x)$ be the true radiance of the scene at pixel x in color channel $c \in \{R, G, B\}$. The observed image $I_c(x)$ is modeled as a spectral attenuation

$$I_c(x) = J_c(x) t_c + B_c (1 - t_c), \quad t_c = e^{-K_c d}, \quad (1)$$

where d is the path length: the distance light travels through water between target and camera, not depth below the surface. K_c is the diffuse attenuation coefficient for RGB channel c , and B_c is

the veiling-light color [2], [14]. The first term is the surviving direct signal; the second is backscatter. As $t_c \rightarrow 0$ the image tends to the flat color B_c , which is why deeply degraded frames lose contrast entirely rather than merely dimming. We note that this single-coefficient formulation is a simplification: Akkaynak and Treibitz [15] show that attenuation and backscatter are governed by different coefficients, each depending on factors this model folds into one constant.

The Jerlov system classifies natural water bodies by their spectral attenuation into open-ocean types (I, IA, IB, II, III, in order of increasing attenuation) and coastal types (1C through 9C, in order of increasing turbidity) [3], [16]. Each type is associated with characteristic wavelength-dependent optical attenuation properties, which we approximate here with an RGB K_c parameterization. The K_c and B_c values used below are our own: we chose them to follow the attenuation ordering that the Jerlov classification implies [1], [3], rather than taking the measured values reported there. The ten settings used below are therefore synthetic attenuation levels ordered as the Jerlov classification orders its water types, rather than the Jerlov water types themselves; we keep the Jerlov labels for readability. So (1) becomes a one-parameter group in path length once a type is chosen. Inverting it gives the restoration [1], [2]

$$\hat{J}_c(x) = \frac{I_c(x) - B_c(1 - t_c)}{t_c}, \quad (2)$$

with t_c clipped below at 0.05 to prevent noise amplification as the denominator approaches zero. Since (2) restores channel means for our water types but not water's overall local contrast, we follow it with CLAHE [17] on the L channel in Lab space and a gray-world white balance giving the end result below.

3.2 Model foundation

The model used here is Faster R-CNN [18] with a ResNet-50 [19] FPN [20] backbone. We measured no training-time benefit from that choice and claim none; what lets a dataset of this size converge in ten epochs is the COCO-pretrained initialization Section 3.4 describes. Our problem can be understood as a multi-classification problem applied many times per image, with an added regression component. A network of region proposal emits candidate regions [18] where for each region a classification head produces a softmax vector over s classes, trained by categorical cross-entropy exactly as in whole-image classification. In our case $s = 2$ (background, target). In parallel, a regression head emits four unbounded box-offset values per class, trained by a smooth- L_1 objective. The Ground-truth class labels for the proposals are not given, but are assigned by intersection-over-union against annotated boxes [18].

Because model detections can include false positives, image confidence alone is not an adequate evaluation metric. Instead, we can use average precision (AP), where a detection counts as a true positive when its intersection over union (IoU) with a ground-truth box exceeds a threshold and its class is correct; sweeping the confidence threshold traces a precision-recall curve whose area is the average precision. We report AP averaged over IoU thresholds 0.50 to 0.95 in steps of 0.05, denoted AP@[.50:.95], which rewards accurate bounding boxes rather than high confidence detection.

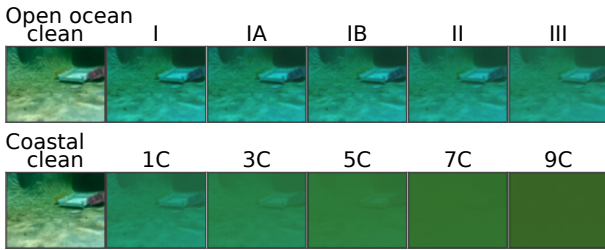


Figure 1: One frame from our dataset, simulated at every attenuation level through Equation (1), labelled by Jerlov type. Top row: the clean frame and the open-ocean types I to III. Bottom row: the same frame and the coastal types 1C to 9C. Contrast and color are lost gradually across the open-ocean types and collapse across the coastal ones.

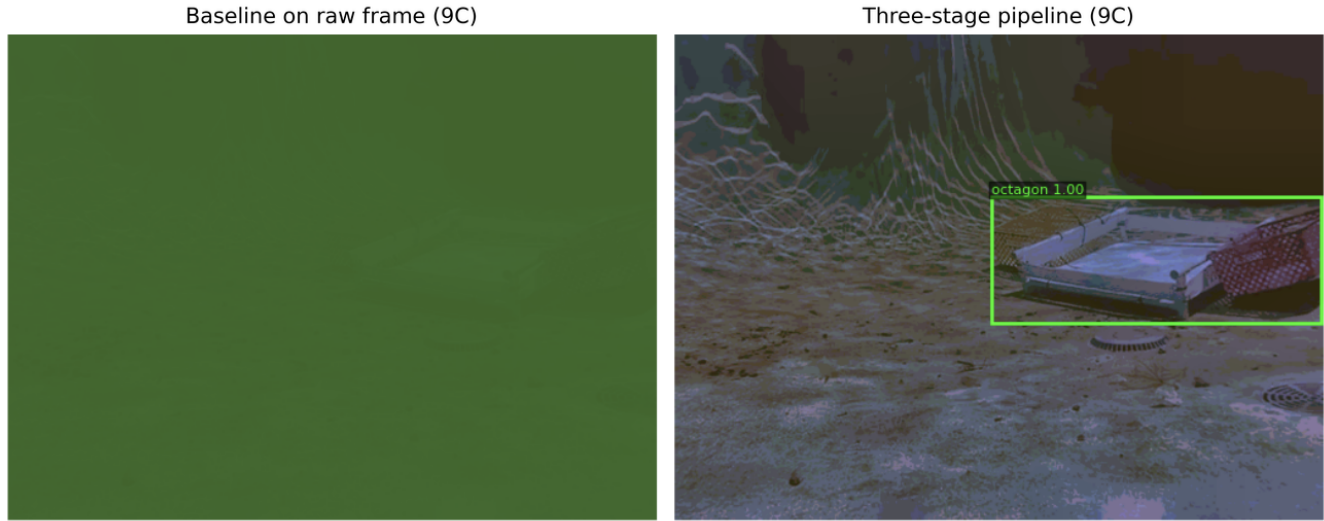


Figure 2: The three-stage correction on one validation frame simulated at 9C, the most turbid coastal type. Left: the raw degraded frame, on which the detector finds nothing. At this turbidity the direct signal has fallen away and the image has tended to the flat veiling-light color B_c , which is the $t_c \rightarrow 0$ limit of (1). Right: the same frame after correction, with the target recovered. The class is labeled octagon for the reason given in Section 3.3.

3.3 Dataset and model

The dataset is a custom single-class target-detection set in COCO format, collected from the camera of an uncrewed underwater vehicle at two swimming pools: a private residential pool and the Woollett Aquatics Center in Irvine, California. The frames were recorded between June and July 2026, while the team was practicing its mission runs for the RoboSub 2026 international competition. It comprises 2,376 images, split into 2,086 training images, a 193-image validation set, and a third partition of 97 images used for neither training nor checkpoint selection, which Section 4.5 evaluates separately. Of the 193 validation images, 150 carry a target box and 43 are negatives in which the target is out of frame, which is why Section 4.6 measures box geometry over 150 annotations. Every AP in this paper is computed over the full 193-image validation split, except where Section 4.5 and Table 4 report the 97-image partition.

The annotation tool wrote more than one output image for some source frames: 79 of the 2,086 training images are second outputs of a frame already in the split, and 59 of those pairs carry different box geometry. The training data therefore already contains tool-transformed copies, for every model including the control, so the control is not trained on wholly untransformed data. No two validation images share a source frame with each other; fourteen share one with a training image (Section 5). Five of the 97 images in the third partition are second outputs of a training frame and one of a validation frame.

Frames were extracted from each recording as consecutive runs, and the filename index is the source video’s frame number. The three partitions were produced by a random split of the images in the nominal ratio 88:8:4, which divides images rather than recordings or source frames; the export bears this out. All sixteen recordings represented in the 97-image partition also appear in training

and none is unique to it, the dataset covering seventeen in all; and of the one hundred source frames for which the annotation tool wrote two output images, twenty have those two outputs in different partitions, which a split grouped on the source frame could not produce. The proportions in which they fall across the partitions are those of an assignment made image by image ($\chi^2 = 5.8$ on five degrees of freedom, $p = 0.33$). Measured against the training split, 5 of the 97 are second annotation outputs of a training frame, 89 are one frame away, 2 are two frames away and 1 is four frames away. No image in the partition was used for training or for checkpoint selection, but its images are not temporally independent of the training footage. Section 5 reports the same measurement for the validation split.

The target is a rectangular equipment table roughly 1 m across, with a flat top plate, two side trays and four legs, resting on the pool floor. It is the same physical object in every frame, photographed from a range of distances and angles.

The class is labeled *octagon* in the annotation files. That name comes from the competition mission the vehicle was built for: ascending inside an octagonal PVC frame floating at the surface. The table sits directly beneath the center of that frame and marks its position from underwater, so the team’s name for the mission carried over into the class label. The octagon itself is at the surface and appears in no frame; the annotated object is always the table.

Annotations are single axis-aligned boxes drawn by hand around the visible extent of the table. Every target fills a moderate to large part of the frame, so no annotation falls in the COCO small-area bucket.

Two properties of the split bear on every result below, so we state them here rather than deferring them to Section 5. First, the partitions divide frames rather than recordings, so an image can

sit one frame from a training image; the paragraph above measures how close, and the very high clean scores in Table 2 may be optimistic for that reason. Second, the same validation images were used both to select the checkpoint and to report results; the 97-image third partition was held back from both, and Section 4.5 reports it separately.

All models are Faster R-CNN ResNet-50 FPN initialized from COCO-pretrained weights [21], with the box predictor replaced by a two-class head (background plus target) and fine-tuned end-to-end for 10 epochs using SGD (learning rate 0.005, momentum 0.9, weight decay 5×10^{-4}) with a step schedule decaying by $\gamma = 0.1$ every 4 epochs. The checkpoint minimizing validation loss was retained.

3.4 Training augmentation policies

Seven models were trained under seven data augmentation policies. Each augmentation is applied stochastically per image with probability p and strength sampled from the stated range. The seventh is the control, which applies no augmentation of its own. It is listed here so that every model appearing in the results has a training row:

Model	Augmentation applied during training	p
Scale+Rotate	rotation $\pm 15^\circ$; scaling $0.8-1.2\times$	0.30
Brightness	uniform brightness $0.6-1.5\times$	0.40
Non-uniform light	directional illumination gradient, intensity $0.2-0.5$	0.25
Noise	additive Gaussian, $\sigma \in [5, 20]$	0.30
Blur	Gaussian Blur radius $0.5-2.5$	0.20
All (aggregate)	all of the above, plus grayscale at $p = 0.15$	–
Baseline (no aug)	–	–

Table 1: Training augmentation policies. Rotation and scaling transform annotation boxes alongside the image; the photometric augmentations leave boxes unchanged.

One further difference was not intended. The original checkpoints were trained before the data-loader workers were seeded: `train_loader` was built without a `worker_init_fn`, so the augmentation draws came from the global NumPy stream and each forked worker inherited the parent’s state and drew the same sequence. The augmented policies may therefore have received less varied augmentation than Table 1 describes, while the control draws no augmentation randomness of its own and is unaffected. The runs retrained in Section 4.3 used per-worker seeding, so the two checkpoint families differ in their data-loader random-number handling as well as in their seeds.

3.5 Evaluation regimes

Each model was evaluated in four regimes. The **clean** regime uses an unmodified validation split. The **composite augmented** regime applies every degradation simultaneously and deterministically to every validation image: rotation 25° , scaling $0.85\times$, brightness $0.6\times$, non-uniform illumination (intensity 0.4, direction 45°), Gaussian noise $\sigma = 25$, and blur radius 1.5. Determinism matters here:

training augmentation is stochastic to generate variety, whereas evaluation must be repeatable so that conditions are comparable across models. Geometric transforms are applied to ground-truth boxes as well, so that AP is computed against correctly transformed annotations.

Two of these composite settings are deliberately outside the ranges any model trained on: rotation is 25° against a training maximum of 15° , and noise is $\sigma = 25$ against a training maximum of 20. The composite condition is therefore out of distribution for every model, including the aggregate one, and its results should be read with that in mind.

The **per-degradation** regime applies each of the seven degradations the aggregate model trains on, one at a time at the composite setting, so that a model trained on one degradation can be tested against that degradation in isolation. Six of them are the ones that make up the composite condition; the seventh is grayscale, which the aggregate policy also uses and which is applied here by the same function with the same settings as in training, so it is a trained-on condition and not a held-out one. This answers the first research question directly, instead of inferring it from the composite result.

The **held-out** regime applies four degradations that appear in no training policy: motion blur, JPEG compression at quality 18, simulated marine snow, and defocus. Because no model has seen them, this regime does not give the aggregate model the built-in advantage it has under the composite condition. Two of the four are nonetheless relatives of a degradation the models did train on: motion blur and defocus both remove high spatial frequencies, as the Gaussian blur in the training policies does, so they test transfer within a family rather than to an unfamiliar kind of corruption. JPEG compression and marine snow are the two unlike anything in training. The marine-snow generator places particles at random positions; its seed is derived from the image itself, so every condition in this paper reproduces exactly across evaluation passes. What that fixes is which particle field each image receives, not how far a different field would move the result, so this column is reproducible rather than averaged over particle seeds.

Separately, a **water-type sweep** evaluates the Jerlov based de-hazing. Our entire validation set is degraded via (1) at all 10 water types $\times$ 6 path lengths (1, 2, 4, 6, 8 and 10 m), and performance is recorded both on the degraded frame (“raw”) and after applying the three-stage pipeline (“corrected”), yielding 60 paired measurements per model. Following the second research question we report this sweep in AP rather than in top-1 confidence, for the reason given in Section 3.2: confidence can rise on a detection that is in the wrong place, and AP cannot.

4 Evaluation

4.1 Validation Test of Models

Here we test every model on the same validation split to ensure we have no discrepancy with our model training or issues with the models themselves. The spread across each of the models on clean data is 0.036 AP, and the ordering is close to uninformative as the aggregate model in the fifth row ranks fifth in $\text{AP}@[.50:.95]$, behind the control. The second data column, $\text{AP}@.50$, is essentially saturated (≥ 0.995 for every model), indicating that on undegraded

Model	AP@ [.50:.95]	AP@ .50	AP@ .75	AP large	AR@ 1	AR@ 100
Non-uniform light	0.841	1.000	0.959	0.843	0.869	0.875
Brightness	0.832	1.000	0.959	0.833	0.859	0.865
Blur	0.831	0.997	0.957	0.832	0.852	0.859
Baseline (no aug)	0.824	0.997	0.957	0.824	0.849	0.855
All (aggregate)	0.823	0.999	0.959	0.823	0.855	0.862
Noise	0.819	0.999	0.959	0.819	0.846	0.851
Scale+Rotate	0.805	0.996	0.955	0.803	0.831	0.839

Table 2: Detection performance (0 to 1) on the clean validation split (193 images). All seven models lie within a 0.036 AP band. AP_{small} is undefined (−1) throughout because the split contains no ground-truth objects in the COCO small-area bucket.

imagery the task is nearly solved and that the $AP@[.50:.95]$ differences reflect localization tightness rather than detection success. This test on clean validation cannot be used to select a data augmentation policy given that it is used to ensure our models work on their own datasets properly and rather acts as a control.

4.2 The composite condition is a ceiling

We first ran every model against the composite degraded validation set, which applies all six training degradations at once. The aggregate model scores 0.277 AP against the control’s 0.091, a gap of 0.186. The best single-augmentation policy, Blur, scores 0.247; we do not read the 0.029 between them as a difference, for the reason Section 4.3 gives.

We report these numbers but do not build on them. The composite test set is assembled from six of the seven degradations the aggregate model trained on; no other policy trained on more than two of them, so a win for the aggregate model here is close to guaranteed by construction. What that gap measures is how far an augmentation policy can carry a detector inside its own training distribution, not whether that policy generalizes. The two experiments that follow avoid this: Section 4.4 tests each degradation separately, so a policy is credited only where it ought to help, and Section 4.5 tests degradations appearing in no training policy at all. Those are the results this paper rests on.

4.3 How much of a gap is training noise

The seven policies in Tables 2, 4 and 6 were each trained once, so a difference between two of them could be a difference between two training runs rather than between two policies. The question has to be put to separate training runs.

We retrained the control and the aggregate policy under two seeds each — four runs of ten epochs, same hyperparameters as the originals — and scored all four on the degradations held out of every training policy. Table 3 reports every run separately. The two runs of one policy differ by 0.003 for the aggregate and 0.005 for the control on the held-out mean. With two seeds per policy these are observations rather than estimates of a spread, and we use them as such.

The held-out comparison is consistent across all three pairs: the two retrained pairs lead by 0.019 and 0.021 and the pair used throughout by 0.026, the same direction each time and within 0.007

of one another. We do not convert that into a multiple of the run-to-run difference, because two seeds on two policies cannot support the spread such a ratio implies. The difference is also specific to the mean over four conditions; Section 4.4 reports margins on single conditions and makes no comparison of this kind. Section 5 gives the qualifications.

Policy	Run	Clean	Composite	Held-out mean
Control	original	0.824	0.091	0.726
Control	seed 1234	0.827	0.087	0.733
Control	seed 5678	0.823	0.053	0.728
Aggregate	original	0.823	0.277	0.752
Aggregate	seed 1234	0.802	0.493	0.751
Aggregate	seed 5678	0.810	0.465	0.748

Table 3: Per-run AP for the two retrained policies. The held-out mean reproduces; the composite score does not.

The two retrained aggregate runs score 0.493 and 0.465 AP under the composite condition, a difference of 0.028. Both exceed the original aggregate checkpoint’s 0.277, by 0.217 and 0.188. The two retrained control runs score 0.087 and 0.053, a difference of 0.034, against 0.091 for the original control. Re-scoring the original checkpoints under the procedure the retrained runs used leaves their composite scores unchanged, so the discrepancy is associated with the training runs rather than with the evaluation. The most likely reason is the data-loader seeding of Section 3.4: the original augmented checkpoints drew a repeated augmentation sequence and may have received less variety than intended, while the control, which draws no augmentation randomness, is unaffected. The runs do not isolate loader seeding as the cause and we do not claim they do. What they do show is a difference in degree consistent with it: the original control sits 0.004 from one of its retrained runs and 0.038 from the other, while the original aggregate sits 0.188 and 0.217 from its two. The held-out comparison is unaffected, both giving the same answer within 0.007, while the composite comparison runs from 0.186 to 0.412. The result this paper rests on survives the change in training procedure; the composite gap, which Section 4.2 already treats as a ceiling, does not, and we quote no composite gap as a measured quantity.

4.4 Each degradation on its own

The composite condition applies six degradations at once, so it cannot say which individual augmentation helps against which individual degradation, which is the first research question. Figure 3 answers that directly, testing all seven models against each degradation separately.

The answer to the first research question is largely negative. Five specialists cover six degradations, since Scale+Rotate trains on both scaling and rotation. Four of those six pairings beat the control: Scale+Rotate on rotation (0.451 against 0.418), Brightness on brightness (0.829 against 0.801), Non-uniform light on non-uniform illumination (0.839 against 0.817) and Blur on blur (0.823 against 0.797). The margins run from 0.022 to 0.033. These are observed differences between single training runs: each specialist policy was trained once, and the retraining in Section 4.3 covers only the control and the aggregate policy, on the mean over four held-out

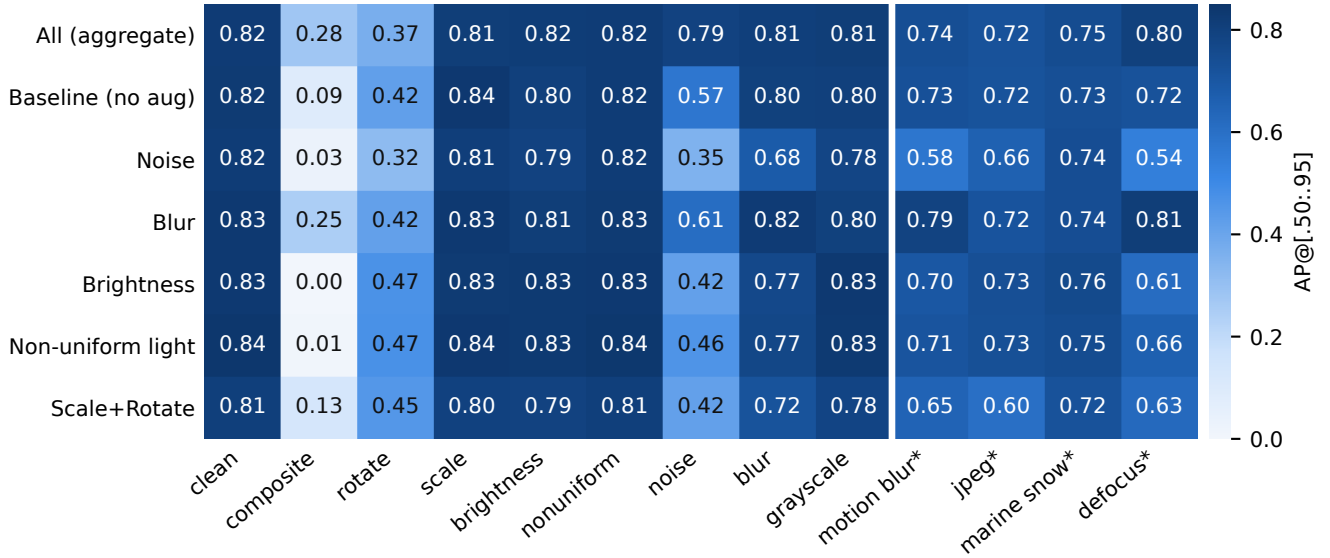


Figure 3: AP@[.50:.95] for all seven models against each condition separately. Columns right of the white rule are the held-out degradations, which appear in no training policy. Darker is better.

conditions rather than on single conditions. We therefore report these four wins as observed and do not claim they are reproducible across training runs, nor rank them against one another. The other two pairings lose: Scale+Rotate scores 0.796 against the control’s 0.836 under scaling, a margin of 0.040 larger than the wins above but from the same single runs, and the Noise model scores 0.353 against the control’s 0.572 under noise. Both policies were trained once for this comparison, so we do not estimate how much of that difference is run-to-run variation; nor do we have an explanation for the Noise result. We report it as observed. The aggregate model, which also trained on noise, is the best model in that column at 0.791.

Rotation and noise are the two degradations that hurt most, and they are much harder than the rest. Rotation is the worst single condition for four of the seven models and noise for the other three; across the field rotation runs from 0.322 to 0.473 and noise from 0.353 to 0.791, with mean scores of 0.417 and 0.517. Every other single degradation (scaling, brightness, non-uniform illumination, blur and grayscale) leaves every model at 0.681 or above. Section 4.6 discusses how rotated boxes are scored, which bears on this figure.

The aggregate policy is broadly but shallowly better. Across all thirteen conditions in Figure 3 it beats the control in ten, which is consistent in sign, though seven of those ten margins are below 0.02 and so inside the range we have said we cannot rank on. It is the single best model in only two of the thirteen, and on the seven individual degradations, all of which its own policy trains on, it is best in only one, noise. Training on everything therefore buys a little robustness in many directions at the cost of specializing in none: a model slightly better than no augmentation almost everywhere, and rarely the best choice anywhere.

Most individual degradations do not separate the models, and no augmentation at all is a middle-of-the-pack policy. Four of the seven

single degradations — scaling, brightness, non-uniform illumination and grayscale — hold all seven models inside bands of 0.033 to 0.051 AP, too narrow to rank on. Only rotation (0.151), blur (0.141) and noise (0.438) spread the field wider, so the matrix holds fewer usable comparisons than its size suggests. Against that background the control is never the best and never the worst model on any of the seven degradations the training policies cover, ranking from second under scaling to fifth under brightness and non-uniform illumination. Averaged over those seven, only two of the six augmented policies beat it: the aggregate policy at 0.746 and Blur-only at 0.731 against the control’s 0.720, while Scale+Rotate (0.680) and the Noise model (0.651) fall clearly below. Tested one degradation at a time, augmentation is not reliably better than no augmentation. These results come from checkpoints predating per-worker data-loader seeding (Section 3.4), so that finding and the unexplained Noise result apply to these checkpoints rather than to the policies in general.

4.5 Degradations no model trained on

The composite regime is built from the aggregate model’s own training degradations, so its result is a ceiling. To test whether augmentation produces robustness that generalizes, we evaluate all seven models against four degradations that appear in no training policy: motion blur, JPEG compression at quality 18, simulated marine snow, and defocus.

This test is the one our main claim rests on. Against degradations it never saw, the aggregate model scores 0.752 against the control’s 0.726: an advantage of 0.026, where the same comparison was larger under the composite condition, by an amount the retrained runs do not reproduce (Section 4.3). It is also no longer the best model: Blur-only leads at 0.765, by 0.013. This ordering includes the five single-augmentation checkpoints that predate per-worker data-loader seeding (Section 3.4). Section 4.3’s run-to-run differences

Model	Motion JPEG		Marine	Defocus	Mean	Mean
	blur	snow	snow		(val.)	(97)
Blur	0.790	0.716	0.742	0.812	0.765	0.813
All (aggregate)	0.738	0.721	0.750	0.798	0.752	0.805
Baseline (no aug)	0.728	0.720	0.734	0.722	0.726	0.779
Non-uniform light	0.711	0.731	0.749	0.663	0.714	0.750
Brightness	0.699	0.734	0.759	0.613	0.701	0.746
Scale+Rotate	0.650	0.601	0.721	0.625	0.649	0.699
Noise	0.576	0.659	0.742	0.538	0.629	0.672

Table 4: AP@[.50:.95] on four degradations held out of every training policy. The four condition columns are the 193-image validation split; Mean (val.) is their mean there and Mean (97) the same mean on the 97-image partition of Section 3.3. The aggregate model’s large advantage under the composite condition does not survive here: it leads the control by 0.026 on validation and 0.025 on the 97, and is not the best model on either.

were observed on the control and the aggregate policy, not on Blur-only, so we do not measure that lead against them. So we only claim that the aggregate model is no longer best, and we do not claim that Blur-only is better than it. The aggregate model does beat the control on all four conditions taken individually, but on this split it is the best model on none of them. Four of the six augmented models score *below* the control. So what the aggregate policy buys is narrow rather than general: it is best on none of the four.

The 97-image partition of Section 3.3 was used for neither training nor checkpoint selection, so it offers an evaluation free of selection pressure. On the clean split the seven models span 0.842 to 0.869 there, against 0.805 to 0.841 on validation; their means over the four held-out degradations are the last column of Table 4. The aggregate model’s held-out mean is 0.805 against the control’s 0.779, a difference of 0.025 where the same comparison on the validation split gives 0.026, and the ordering of all seven models is the same on both partitions, with Blur-only again first. On the four conditions taken individually its lead there is 0.008 on motion blur, 0.027 on JPEG, 0.029 on marine snow and 0.036 on defocus; data/test97.csv carries every model’s figures. The partition carries 83 annotated targets against the validation split’s 150, so its figures warrant more caution. Its images sit within four frames of a training image and it shares every recording with training (Section 3.3), so it does not test generalization to unseen footage.

The little that survives is concentrated rather than broad. The aggregate model’s 0.026 lead is not spread evenly over the four conditions: defocus alone supplies 73% of it, marine snow 16%, motion blur 10% and JPEG 2%, where the two models differ by 0.002. That split follows the family division noted in Section 3.5: defocus and motion blur are the two relatives of the Gaussian blur in training, and account for 83% of the advantage between them. Marine snow supplies most of the rest, a gap of 0.016. The other degradation unlike anything in training, JPEG, contributes almost nothing: the two models differ there by under 0.002. The held-out result is therefore, on this split, better read as transfer within the blur family than as general robustness, which strengthens the negative finding rather than softening it. The 97-image partition agrees on the size of this advantage and not on its composition. Summed over the four conditions it is 0.101 there against 0.104

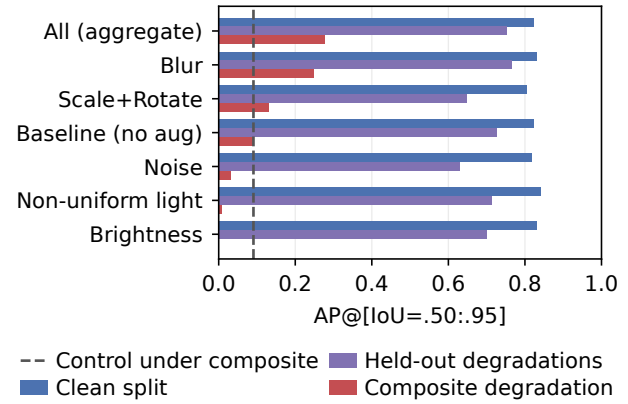


Figure 4: AP@[.50:.95] by model in three regimes. The seven models span 0.036 AP on clean data (blue). That spread widens to 0.276 under composite degradation (red) and narrows again to 0.136 under the held-out degradations (purple) — much narrower than composite, but still close to four times the clean band. Dashed line marks the original control checkpoint under composite degradation, at 0.091; three of the six augmented models fall below it.

here, but defocus supplies 36% of it rather than 73% and JPEG 27% rather than 2%, so the blur-family pair accounts for 44% rather than 83%; the aggregate model also takes first place on JPEG there, by 0.011. With 83 annotated targets spread over four conditions these are small differences on a small sample, and we draw only the negative conclusion they support: the concentration above is a property of the validation split rather than of the policy. The four conditions also differ in how much they discriminate at all: defocus and motion blur spread the seven models by 0.274 and 0.214 and JPEG by 0.133, while marine snow spreads them by only 0.038. We read that column as carrying little signal about augmentation policy: dropping it from the held-out mean leaves the ordering unchanged and widens the gap to 0.029.

Blur-only’s lead follows the same family line, and we offer it only as a conjecture. It wins the two blur-like conditions, motion blur (0.790) and defocus (0.812), and places fifth of seven on JPEG, which chiefly introduces block artifacts; averaged that way it scores 0.801 against 0.729. We do not claim this as a mechanism. We chose the grouping after seeing the results, the control shows no split of its own either way (−0.002), and dropping marine snow turns Scale+Rotate positive, a policy that trained on nothing blur-like. Testing the idea properly needs held-out degradations chosen in advance to sit near and far from each training policy.

4.6 What rotated boxes do to AP@.75

The composite condition rotates each image by 25° and rotates the ground-truth boxes with it. Because an annotation is stored as an axis-aligned rectangle, the rotated box must be re-enclosed in a new axis-aligned rectangle, which is larger than the original. For a box of width w and height h rotated by θ , the enclosure has width $w|\cos \theta| + h|\sin \theta|$ and height $w|\sin \theta| + h|\cos \theta|$. Table 5 evaluates that enclosure across rotation angles.

At 25° the transformed reference box has an area 1.766 times the original for a square annotation, and more for any other rectangle.

Rotation	0°	5°	10°	15°	20°	25°	30°	45°
Area inflation, square annotation × Illustrative IoU, square annotation:								
original box vs. rotated enclosure	1.000	0.852	0.745	0.667	0.609	0.566	0.536	0.500

Table 5: Effect of rotating an axis-aligned ground-truth box and re-enclosing it, shown for a square annotation. The second row is an illustrative limiting case, not a bound established for the actual objects: it assumes the object’s own extent does not change with rotation, and the reciprocal relation it uses holds only while the original box lies inside the enclosure, which it does up to an aspect ratio of about 4.5:1. An object that fills its box reaches IoU 1.0 at every angle.

Across our 150 validation annotations the median area inflation is 1.868×, with an interquartile range of 1.790 to 2.082 and a maximum of 4.323× for the most elongated box.

Area inflation alone does not determine the resulting IoU. If the object fills its annotation box, the tightest axis-aligned box around the rotated object coincides with the transformed reference box, and a perfect prediction still reaches IoU 1.0; the inflation costs nothing. Under other relationships between the object and its box the overlap can be lower. The reciprocal of the median inflation is 0.535, but that is an illustrative area comparison, not a demonstrated upper bound on IoU: it would be the bound only if the object’s own extent were unchanged by rotation, which we did not measure.

We did not compare predicted boxes against alternative ground-truth conventions, so we cannot quantify how much of the measured AP@.75 result is attributable to the box transformation rather than to detector performance. Under the composite condition the aggregate model achieves AP@.75 of 0.199. This confirms that detections above the 0.75 IoU threshold occur under the actual evaluation, but it does not isolate rotation from the five other degradations applied simultaneously, nor quantify the geometric scoring effect. AP@.75 under this condition should therefore be interpreted as a joint measure of detector performance and evaluation geometry, not as a standalone measure of either.

4.7 Effectiveness of the correction across water types

Model	Open ocean (I–III)			Coastal (1C–9C)		
	Raw	Corrected	Δ	Raw	Corrected	Δ
All (aggregate)	0.801	0.729	−0.073	0.502	0.693	+0.191
Blur	0.803	0.697	−0.106	0.440	0.645	+0.205
Brightness	0.815	0.708	−0.107	0.474	0.674	+0.199
Noise	0.776	0.625	−0.151	0.396	0.575	+0.179
Non-uniform light	0.816	0.682	−0.134	0.446	0.647	+0.201
Scale+Rotate	0.777	0.635	−0.142	0.411	0.588	+0.177
Baseline (no aug)	0.795	0.688	−0.108	0.417	0.645	+0.228

Table 6: Mean AP@[.50:.95], split by water-type family, averaged over six path lengths. The correction is not neutral in open-ocean water. It costs every model accuracy there, and gains every model about the same amount in coastal water. Each Δ is computed before rounding, so three of them differ by 0.001 from the rounded figures beside them.

Averaged over the ten water types, six path lengths and seven detectors, the correction, applied throughout this section as the full three-stage pipeline, moves open-ocean AP from 0.798 to 0.680, a loss of 0.117, and coastal AP from 0.441 to 0.638, a gain of 0.197. The aggregate figure conceals the structure, and in AP that structure is sharper than it was in confidence. In open-ocean types the correction now *costs* accuracy, between 0.073 and 0.151 depending on the detector, because those images are barely degraded to begin with and the contrast stages strip contrast that was never haze; Section 4.9 shows the inversion on its own does not do this. The benefit is concentrated in coastal types, where every detector gains between 0.177 and 0.228. The crossover sits between 1C, where the correction still costs 0.041, and 3C, where it gains 0.139. In the most degraded conditions tested (7C and 9C at 6 m and beyond) raw AP is exactly 0.000 in all 42 model-condition cells, while the corrected pipeline recovers 0.440.

Table 6 also tests directly whether augmentation and correction overlap, since both are present in this sweep. The aggregate model leads the control by 0.046 AP on uncorrected frames and by 0.045 AP after correction, so the correction is worth about the same amount whether or not the model was augmented: the two add. Two things limit that. It is an average that hides a reversal, since the augmentation advantage grows by 0.035 in open-ocean water and shrinks by 0.037 in coastal water. And it holds only for the aggregate policy: the same test on the five single-augmentation models gives −0.011, −0.014, −0.027, −0.043 and −0.046, every one negative and at least nine times the aggregate model’s −0.001. For those policies the correction recovers part of what the augmentation already provided.

The open-ocean cost is not an artifact of reporting AP instead of confidence. Across the seven detectors, mean top-1 confidence in open-ocean water falls by 0.045 to 0.141 and the detection rate by 0.025 to 0.107, so all three metrics agree. If a system applies the correction to every frame without checking, it will lose accuracy whenever the water is clear.

4.8 Where the correction fails in extreme turbidity

The dehazing stage does not extend detection indefinitely. Table 7 reports, for the Noise model, the shortest path length at which detection fails completely. Like Section 4.7, this section uses the

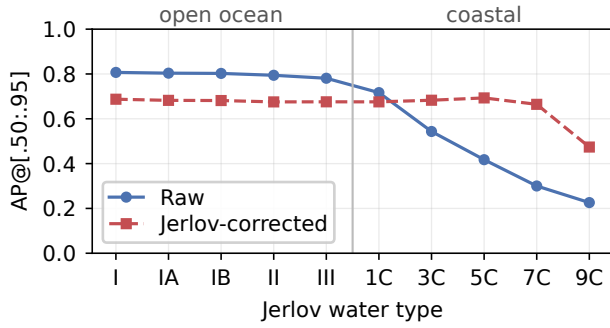


Figure 5: AP@[.50:.95] by water type, averaged over six path lengths and all seven detectors, raw against corrected. Without correction accuracy decays steadily from 1C onward; with correction the curve is far flatter across the coastal types, at the cost of a consistent loss in the open-ocean types where there was little to remove.

three-stage pipeline that Section 4.9 identifies as the weaker configuration, and the boundary has not been re-measured with the inversion alone.

The correction pushes the failure boundary outward substantially — 3C, 5C and 7C are recovered entirely — but at 9C it delays failure only from 4 m to 8 m, and beyond that both pipelines return nothing. This is the expected consequence of the transmission clipping in (2): once t_c falls below the clip floor, the direct signal has been attenuated past the point where inversion can recover it, and amplifying what remains amplifies noise rather than scene content. The correction should therefore be understood as extending the usable envelope, not removing the limit.

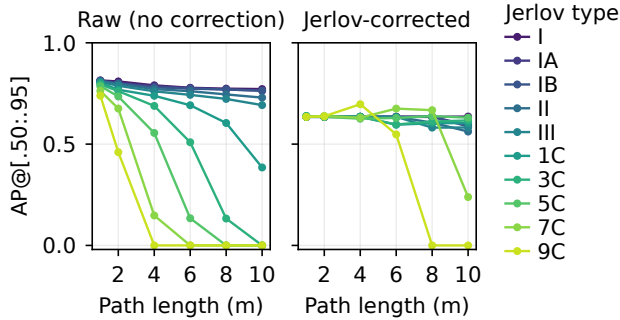


Figure 6: AP@[.50:.95] versus path length for the Noise model, by water type. Raw curves (left) collapse with turbidity; 3C, 5C, 7C and 9C all fail within the sweep. Corrected curves (right) stay between 0.54 and 0.70 out to 6 m for every type, after which 9C reaches zero at 8 m and 7C falls to 0.24 at 10 m.

4.9 Which stage of the pipeline earns the gain

The pipeline described in Section 3.1 has three stages: the Jerlov inversion of (2), then CLAHE on the L channel, then a gray-world white balance. Only the first is physics-based, and the results above cannot separate its contribution from that of the two contrast steps. We therefore ran each stage on its own, over three representative conditions (5C at 4 m, 3C at 6 m, and I at 2 m) across all seven

detectors, and added the Dark Channel Prior [12] as a standard enhancement baseline that uses no water-type information at all.

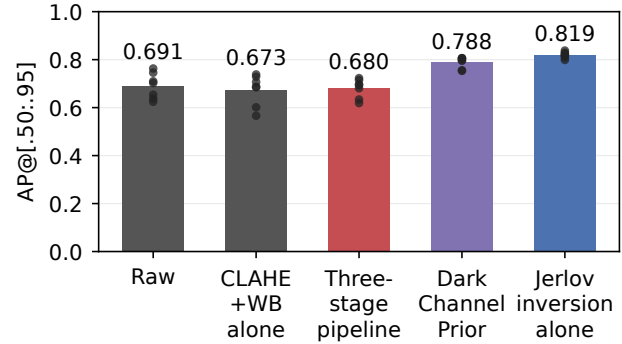


Figure 7: AP@[.50:.95] by enhancement method, averaged over three conditions. Dots are the seven detectors. The Jerlov inversion on its own is the best method tested; adding the contrast stages that follow it in our pipeline costs more than the inversion gained.

This reverses the recommendation implied by the three-stage pipeline we began with. The Jerlov inversion *alone* scores 0.819 against 0.691 for the raw image, a gain of 0.128, and it improves on raw for all seven detectors. The full three-stage pipeline scores 0.680, which is *below* raw. CLAHE and white balance alone score 0.673, worse than raw for five of the seven detectors. The contrast stages are not only redundant after the inversion, they make the result worse, and they are the reason the measured benefit looked smaller than it is.

The same split matters against the external baseline. The full pipeline loses to the Dark Channel Prior (0.680 against 0.788) on all seven detectors, which alone would be a negative result for the physics-based approach. The inversion by itself beats the Dark Channel Prior (0.819 against 0.788) on all seven, by 0.019 to 0.054. If we measure this method as a three-stage pipeline it is worse than a standard baseline, but if we measure only the physics step on its own it is better. We therefore recommend the inversion without the contrast stages.

4.10 What the correction costs when its settings are wrong

Every result above gives the correction the exact water type and path length used to degrade the image. A vehicle has neither. To bound how much that matters we re-ran the correction at 5C and 4 m with deliberately wrong settings, again using the three-stage pipeline rather than the inversion alone that Section 4.9 recommends.

The error is strongly asymmetric, and it is useful to know which direction is the safe one. Assuming a path length 3 m shorter than the truth scores 0.714, slightly *better* than the correct 4 m value of 0.682; assuming 3 m longer scores 0.375, a loss of 0.307. The same holds for water type. Every type clearer than the truth scores between 0.739 and 0.771, above the correct setting's 0.682. Assuming 7C drops it to 0.088, a loss of 0.595, and assuming 9C drops it to 0.001.

Water type	I	IA	IB	II	III	1C	3C	5C	7C	9C
Raw — fails at	—	—	—	—	—	—	10 m	8 m	6 m	4 m
Corrected — fails at	—	—	—	—	—	—	—	—	—	8 m

Table 7: Shortest path length at which AP across the validation split reaches zero (Noise model). “—” denotes no failure within the 10 m sweep. Failure is defined by AP alone rather than jointly with detection rate, since the two need not coincide and reporting both would invite confusion.

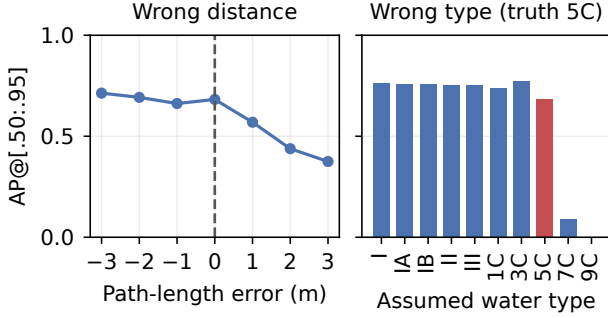


Figure 8: $AP@[.50:.95]$ when the correction is given wrong parameters, averaged over seven detectors. Left: error in assumed path length, true value 4 m. Right: assumed water type, true value 5C in red. Both panels are asymmetric in the same direction: assuming clearer, nearer water changes AP little and usually raises it — every clearer water type scores above the true 5C, and only the 1 m underestimate costs anything, 0.020 — while assuming murkier, further water is very costly.

The mechanism is visible in (2). Over-estimating attenuation makes t_c too small, so the division amplifies both the residual signal and the noise; under-estimating it leaves some haze in place, which the detector tolerates. A practical system that has to guess should therefore guess conservatively. Assuming the water is clearer and nearer than it really is costs us very little. The opposite error is more damaging, and at the extreme it removes detection altogether: over-estimating path length by 3 m still leaves AP at 0.375, while assuming 9C water drops it to 0.001.

4.11 Estimating the parameters from the image

The stronger version of the previous question is whether the settings can be recovered from the frame itself. We implemented a blind variant that searches water type and path length to optimize a no-reference image-quality score, with no access to the true values. This study too used the three-stage pipeline, so it is not directly comparable with the inversion-only result in Section 4.9; we have not measured a blind-estimated inversion on its own.

Blind estimation recovers the water type in 5 of 18 trials, under the diagnostic described below. Despite that, it beats the closed-loop configuration in both coastal conditions (Table 8), for the reason Section 4.10 gives: its errors tend toward under-correction, which is the safe direction. In clear water it fails badly, scoring 0.394 where leaving the image alone scores 0.817, because it infers heavy attenuation from an image that has almost none and then inverts it. Averaged over the three conditions blind estimation is worse than doing nothing.

This is not a problem with our search procedure alone. Within the transmission term, attenuation and path length are coupled through the product $K_c d$, so that term on its own cannot separate them: a clear water type at long range and a turbid one at short range give the same transmission. The full model is not quite so strict. Water type also sets the veiling-light color B_c , which in our implementation differs between types — from $[0.05, 0.35, 0.55]$ at I to $[0.22, 0.40, 0.15]$ at 9C — so two conditions with the same $K_c d$ do not produce exactly the same image. We therefore claim only that the coupling makes the two hard to separate, not that the problem is unsolvable in principle.

The diagnostic that accompanies it is narrower: it evaluates 18 simulated conditions — six water types crossed with three path lengths — on a single input frame, not 18 distinct scenes. Under its criterion, exact recovery of the water type while ignoring the estimated path length, the type was recovered in 5 of 18 conditions (27.8%); the full type-and-path-length pair was recovered in none. Its median optical-depth error, which compares $\tau = K_c d$ under the estimated water type and path length against the true values as $\sum |\tau_{\text{est}} - \tau_{\text{true}}| / \sum |\tau_{\text{true}}|$, is 36.8%. These rates describe one frame under the conditions tested, not identification accuracy in general.

We also looked at the transmission itself, and found we could not use it. The estimator reads transmission off the image as a per-channel dynamic range, which recovers it only up to an unknown factor equal to the scene’s own range. That factor is close to uniform across the three channels for this frame, so it very nearly cancels when we take ratios between channels: a small ratio error would tell us about the frame’s own channel balance, not about how well the water’s color cast was recovered. For the same reason the estimator cannot establish the absolute scale of transmission at all. We therefore do not report transmission-scale or color-ratio errors as evidence of recovery accuracy. Fixing the scale would need a second cue, such as a stereo pair, a known-size reference object, or a depth sensor.

The two interventions studied here act on different parts of the problem, and Section 4.7 shows their effects approximately add for the aggregate policy but overlap for the other five augmented policies. Augmentation acts on the detector, though Section 4.4 shows it does not buy robustness uniformly: the aggregate model falls below the control on rotation and on scaling, the two geometric degradations. Section 4.5 shows that what it does buy is largely confined to the degradations trained on and to their close relatives. The Jerlov correction acts on the image, and buys robustness specifically to wavelength-selective attenuation and backscatter [1], [2], a degradation no amount of the augmentations tested here simulates, since none of them models physical light transport. Neither substitutes for the other. The answer to the second research question is therefore conditional on which configuration is meant: the physics inversion on its own is effective, raising AP from 0.691 to 0.819 and

Condition	Raw	Blind estimate	Known settings
3C @ 6 m	0.604	0.756	0.671
5C @ 4 m	0.652	0.718	0.682
I @ 2 m	0.817	0.394	0.688
Mean	0.691	0.623	0.680

Table 8: AP@[.50:.95] with parameters estimated from the image, against the raw frame and against the closed-loop setting where the true parameters are supplied. Averaged over seven detectors.

beating a standard dehazing baseline on all seven detectors, while the three-stage pipeline we began with scores below the raw image. The methodological finding is arguably the more portable one: clean-split validation was not informative about robustness here, and we do not claim anything stronger than that. All seven models sit within 0.036 AP on the clean split, each was trained once, and the clean ordering does not survive a change of evaluation set. But a spread that small, from single runs, is not enough to show that the ordering reversed rather than that it never meant much. The claim our evidence supports is the weaker one: a clean validation score did not predict which model would hold up under degradation, so robustness must be measured under the conditions of interest.

5 Limitations and Future Work

The attenuation coefficients are uncalibrated. The K_c and B_c values in our equation were constructed to follow the qualitative attenuation ordering associated with the Jerlov classification [1], [3] where red attenuates the fastest and coastal types attenuate more strongly across all bands, but these values are not the same as calculated values. In other words, they match the optical ordering of Jerlov water types but are not calculated for exactitude. Hence, the results involving the water-type sweep should therefore be read as characterizing the approximate behavior of the correction under a plausible environment but not as calculated field predictions of what the water is like.

We cannot separate the rotation scoring effect from the detector. Rotating an image rotates its ground-truth boxes, re-enclosed in new axis-aligned rectangles, inflating the box by a median of $1.868\times$ at 25° . We did not measure how the object’s own extent changes under rotation, so we cannot say what that costs a detector. Separating them means scoring the same predictions against both the inflated box and a tight one around the rotated object, which needs rotated-box or polygon annotations on sample frames. That is inference-only work and the clearest next step for Section 4.6.

This is a single-class study, and the run-to-run evidence rests on two seeds. The task has one target class, so the mean in mean average precision is taken over a single class and we report AP throughout; the results may not extend to multi-class discrimination. The observed run-to-run differences in Section 4.3 cover only the control and the aggregate policy, two seeds each, and refer to the mean over four held-out conditions. They are not a general estimate of training variability and do not apply to individual-condition margins. The specialist results in Section 4.4 come from single training runs and are therefore observed differences rather than effects shown to be reproducible. The retrained models match the originals on the four held-out degradations, within 0.007 on the held-out mean, but not under the composite condition: the

retrained aggregate runs score 0.493 and 0.465 there against the original checkpoint’s 0.277, while the retrained control runs score 0.087 and 0.053 against its 0.091. The discrepancy is real and, as Section 4.3 reports, associated with the training runs rather than with the evaluation; its cause is not isolated, though the most likely reason is the data-loader seeding of Section 3.4, which the original checkpoints predate. We therefore no longer treat the composite gap as a measured quantity. Retraining the five specialist policies under the seeded loader is the clearest next step for Section 4.4, and scoring the retrained runs on the seven single degradations would show which conditions drive it; neither has been run here. The held-out comparison is unaffected, both families agreeing within 0.007.

The models had limited training. While our models were all trained with ten epochs to ensure similarity in the model training process, that is a relatively low number given our dataset having over 2,000 images. This leaves the question of whether increasing the epochs in training would be more effective than augmentations to the dataset of our models.

Neither evaluation partition is temporally independent of the training footage. Frames were extracted from each recording as consecutive runs, and the partitions divide frames rather than recordings, so most held-out images are adjacent in time to a training image. Of the 193 validation images, 14 are second annotation outputs of a training frame, 172 are one frame away and 7 are two frames away; the 97-image partition is distributed the same way (Section 3.3). The consequences differ between the two. The validation split carries both this dependence and the checkpoint-selection pressure, since the images that selected the checkpoints are the images we report on, so its absolute AP values may be optimistic estimates of performance on independently recorded footage. The 97-image partition carries the dependence but no selection pressure, which is what Section 4.5 uses it for.

Comparisons among models on either partition are controlled, since every model sees the same images, but temporal dependence may limit how far the relative differences generalise to unseen recording sessions: a model that memorizes more would gain more from the near-duplicates. The degradations held out of every training policy are a separate axis — they withhold conditions rather than images, and do not remedy the image-level dependence. A recording-level split with a held-out session is therefore the most useful next step. The recording labels are recoverable from the file-names, so what it needs is a redesigned split and retraining every model, not a re-collected dataset; one recording also supplies 32% of the images, which makes a balanced split by session awkward. It is outside the scope of this revision.

The water in the source images is not the water being modeled. The frames were captured in two swimming pools (Section 3.3) and are visibly clear in Figure 1, so they carry very little of the wavelength-selective attenuation and backscatter the correction exists to remove. Every water type reported above is synthetic, applied through (1) to an almost-undegraded image. Whether the correction behaves the same way on frames that already carry real attenuation is untested, and the open-ocean cost reported in Section 4.7 may be specific to inverting haze that was added rather than haze that was there.

The sweep gives the correction the parameters that degraded the image. Sections 4.7 and 4.9 supply the correction with the exact water type and path length used to degrade each frame, so they measure a best case rather than deployment performance. Sections 4.10 and 4.11 measure what that assumption is worth. The figures above should be read as an upper bound.

There is no comparison outside our own models. The Dark Channel Prior in Section 4.9 is the only method here that we did not build. No external detector is evaluated, and no public underwater benchmark such as DUO [10] or RUOD [11] is used, so we cannot say how this detector compares to published results on a shared dataset. Evaluating on one of those benchmarks is the clearest next step for making these results comparable to other work.

6 Conclusions

This study shows that detector-side augmentation and physics-based image correction act on different parts of the problem, and Section 4.7 shows their effects approximately add for the aggregate policy but overlap for the other five augmented policies. Aggregate augmentation gives an advantage of 0.026 AP once it is tested one degradation at a time, and that advantage is no larger on the degradations it trained on than on degradations held out of every training policy. Retraining both policies under two seeds, the advantage appeared again in each pair, at 0.026, 0.019 and 0.021 AP, though two seeds per policy cannot establish its reproducibility. Under the composite condition the original checkpoints give a gap of 0.186 AP; retraining the two policies returned 0.406 and 0.412, so that figure was not reproduced and we do not read it as a stable estimate of what the aggregate policy buys (Section 4.3). The condition is in any case built from the aggregate model's own training degradations, so it measures a ceiling rather than a fair contest: the size of any gap there follows from how the test was built rather than from broader robustness. Against four degradations held out of every training policy the same model leads the control by 0.026 AP and is not the best model tested; this ordering includes the five single-augmentation checkpoints that predate per-worker data-loader seeding (Section 3.4). What advantage remains is concentrated on the validation split, where the two held-out degradations that are relatives of the Gaussian blur in training supply 83% of it; on the 97-image partition the same advantage divides much more evenly, so that attribution is a property of the split rather than of the policy (Section 4.5). Those single-degradation results come from checkpoints predating per-worker data-loader seeding (Section 3.4), so they and the unexplained Noise result apply to these checkpoints rather than to the policies in general; the aggregate-versus-control comparison does not depend on them. Clean-split validation did not predict

either outcome, so robustness evaluation must be conducted under degraded conditions rather than inferred from clean performance, and also under conditions the model was not trained for, since a test built from the training degradations measures the best case rather than what the model can actually do.

The Jerlov parameterized radiative correction is effective but narrowly targeted, and separating its stages changed which configuration we recommend. The physics step alone raises AP from 0.691 to 0.819, measured across three conditions (5C at 4 m, 3C at 6 m and I at 2 m), and beats a Dark Channel Prior baseline on every detector tested; the three-stage pipeline, which follows the inversion with contrast enhancement and white balance, scores 0.680 and loses to that baseline on every detector. That comparison is favorable to us by construction: the inversion is given the attenuation level and path length used to degrade each frame, the frames are generated by the equation it reverses, and the coefficients are our own, while the Dark Channel Prior receives none of this. It is an upper bound on what the inversion can do, not an estimate of what it would do on a vehicle. Across the water-type sweep, which used the three-stage pipeline, the correction gains 0.197 AP in coastal water and costs 0.117 AP in open-ocean water where there was little degradation to remove. The ablation indicates that the open-ocean cost belongs to the contrast stages rather than to the physics: in the one clear-water condition the ablation covers, the three-stage pipeline loses 0.129 AP while the inversion alone gains 0.008. The three-stage pipeline retains a hard limit at extreme turbidity, where transmission falls below the point of recoverable inversion; the inversion alone was not measured there. Its parameters are also hard to recover from a single frame: water type and path length enter the transmission term only through their product, and although the veiling-light color B_c varies with type as well, that extra information was not enough for our blind estimator to separate them. On a single frame it recovered the water type in 5 of 18 simulated conditions and recovered the type and path length together in none. Priorities for further work are calibrated attenuation coefficients [1], re-running the sweep, mismatch and blind-estimation studies with the inversion-only pipeline, evaluation against a public benchmark such as DUO [10] or RUOD [11], session-level splitting with a held-out test partition, and more training seeds, across more policies, to measure run-to-run variation more tightly.

Data and Code Availability

The result tables behind every figure and table ship as CSV files, together with scripts that re-derive each reported number from those tables. The training notebook and the dataset export are available on request.

References

- [1] M. G. Solonenko and C. D. Mobley, "Inherent optical properties of Jerlov water types," *Applied Optics*, vol. 54, no. 17, pp. 5392–5401, 2015. doi: 10.1364/AO.54.005392.
- [2] J. S. Jaffe, "Computer modeling and the design of optimal underwater imaging systems," *IEEE Journal of Oceanic Engineering*, vol. 15, no. 2, pp. 101–111, 1990. doi: 10.1109/48.50695.
- [3] N. G. Jerlov, *Optical Oceanography*. Elsevier Oceanography Series, vol. 5, 1968.

- [4] C. Li et al., "An Underwater Image Enhancement Benchmark Dataset and Beyond," *IEEE Transactions on Image Processing*, vol. 29, pp. 4376–4389, 2020. doi: 10.1109/TIP.2019.2955241.
- [5] P. M. Uplavikar, Z. Wu, and Z. Wang, "All-In-One Underwater Image Enhancement using Domain-Adversarial Learning," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach, CA, USA, 2019.
- [6] T. Mondal, R. Mendoza, and L. Drumetz, "Physics Informed and Data Driven Simulation of Underwater Images via Residual Learning," *arXiv:2402.05281*, 2024.
- [7] C. Li, S. Anwar, and F. Porikli, "Underwater scene prior inspired deep underwater image and video enhancement," *Pattern Recognition*, vol. 98, 107038, 2020. doi: 10.1016/j.patcog.2019.107038.
- [8] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, article 60, 2019. doi: 10.1186/s40537-019-0197-0.
- [9] D. Hendrycks and T. Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations," in *Proc. International Conference on Learning Representations*, 2019.
- [10] C. Liu, H. Li, S. Wang, M. Zhu, D. Wang, X. Fan, and Z. Wang, "A Dataset and Benchmark of Underwater Object Detection for Robot Picking," in *Proc. IEEE International Conference on Multimedia and Expo Workshops*, Shenzhen, China, 2021. doi: 10.1109/ICMEW53276.2021.9455997.
- [11] C. Fu, R. Liu, X. Fan, P. Chen, H. Fu, W. Yuan, M. Zhu, and Z. Luo, "Rethinking general underwater object detection: Datasets, challenges, and solutions," *Neurocomputing*, vol. 517, pp. 243–256, 2023. doi: 10.1016/j.neucom.2022.10.039.
- [12] K. He, J. Sun, and X. Tang, "Single Image Haze Removal Using Dark Channel Prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2011. doi: 10.1109/TPAMI.2010.168.
- [13] Z. Wang, L. Shen, M. Yu, Y. Lin, and Q. Zhu, "Single Underwater Image Enhancement Using an Analysis-Synthesis Network," *arXiv*, 2021, *arXiv:2108.09023*.
- [14] B. L. McGlamery, "A Computer Model for Underwater Camera Systems," *SPIE Proceedings*, vol. 208, pp. 221–231, 1980. doi: 10.1117/12.958279.
- [15] D. Akkaynak and T. Treibitz, "A Revised Underwater Image Formation Model," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 6723–6732. doi: 10.1109/CVPR.2018.00703.
- [16] C. A. Williamson and R. C. Hollins, "Depth profiles of Jerlov water types," *Limnology and Oceanography Letters*, vol. 8, pp. 781–788, 2023. doi: 10.1002/lol2.10338.
- [17] K. J. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization," in *Graphics Gems IV*, pp. 474–485, 1994. doi: 10.1016/B978-0-12-336156-1.50061-6.
- [18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
- [20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 936–944. doi: 10.1109/CVPR.2017.106.
- [21] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," *Computer Vision – ECCV 2014*, pp. 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48.