Leak-Free Cross-Validated Stacking with Per-Architecture Calibration for Sand-Boil Segmentation in Earthen Levees

Padam Jung Thapa[1], Anav Katwal[2,$], Ayon Dey[2,$], Abdullah Bin Naeem[2,$], Steve Sloan[3], Kendall Niles[3], Md Tamjidul Hoque[2,*]

[1]The Center for Advanced Computer Studies (CACS), School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA 70504, USA
[2]Department of Computer Science, LSU New Orleans, New Orleans, LA 70148, USA
[3]US Army Corps of Engineers, Engineer Research and Development Center, Vicksburg, MS 39180, USA

Emails: padam-jung.thapa1@louisiana.edu; {akatwal, adey, anaeem}@lsuneworleans.edu; {steven.d.sloan, kendall.n.niles}@erdc.dren.mil; thoque@lsuneworleans.edu

[$]Equal contribution. [*]Corresponding author: thoque@lsuneworleans.edu

***Abstract*—Sand boils, points where water seeping beneath an earthen levee re-emerges at the surface, are early warnings of internal erosion, and deep segmentation networks are increasingly used to find them in inspection photographs. Annotated examples are scarce, and two common ways of working around that scarcity quietly inflate reported accuracy: tuning ensemble weights on the same images later used to score them, and training on synthetic images derived from the very photographs held out for testing. We present a sand-boil segmentation framework that closes both loopholes. Every synthetic image carries a pointer to its real parent, and a per-fold filter excludes any image whose parent is held out; five encoder–decoder backbones are trained under five-fold cross-validation, calibrated by one temperature scalar each, and combined by a per-pixel meta-learner fitted only on out-of-fold predictions. On the held-out test set the proposed Updated SandBoilNet reaches an intersection-over-union of 0.707, against 0.608 for the published original re-evaluated on the same split. The calibrated stack does not improve on the best single model, an outcome eight meta-learner families reproduce and which we trace to the members' highly correlated errors. Filtering the synthetic pool for label fidelity lifts the champion to 0.718. We also introduce a mask-conditioned synthesis route that makes the conditioning mask the label by construction, giving labelled training images at zero annotation cost.**

***Index Terms*—Sand boil segmentation, ensemble learning, cross-validated stacking, temperature calibration, hard-negative mining, stacked generalisation, semantic segmentation, civil infrastructure.**

I. Introduction

Pixel-level segmentation of sand boils from levee-inspection imagery is the most direct lever for proactive flood-risk management on earthen-levee networks. Sand boils form when water under hydraulic pressure transports fine sand upward through a permeable foundation layer, producing a small saturated dome at the surface; the mechanism is the early visible stage of internal erosion, which, if left untreated, can progress to a sudden breach during a high-water event [1]. Modern automated work on this class is built on convolutional encoder–decoders, of which Panta *et al.*'s SandBoilNet [2] is the domain-specific state of the art, reporting an intersection-over-union of 0.5743 and a balanced accuracy of 0.8552 on a held-out real test set. Re-evaluated in our own pipeline on this paper's held-out test split, the published checkpoint reaches 0.608 IoU, which we use as the like-for-like reference throughout. The curated dataset used in the present study (whose size is specified in Section III-A) is, to our knowledge, one of the larger sets of its kind, and is the same dataset SandBoilNet was originally trained on.

A natural way to extract residual accuracy beyond a single strong backbone is to combine several backbones into an ensemble. Two related ensembling strategies appear in the levee-defect literature: pixel-level majority voting and weighted probability averaging [3]. Both are easy to implement but share two structural defects that this paper addresses directly; a related third problem, false positives on visually confusable non-defects, is a property of the segmenter rather than the ensemble and motivates our hard-negative phase.

a) Problem 1: Subtle leakage in weighted-average ensembles.: A weighted-average ensemble must choose a weight per base learner. The natural way to choose those weights is to evaluate each base learner on a held-out validation split, fit weights that minimise the ensemble loss on that split, and report the ensemble's accuracy on the same split. The procedure is convenient, but it tunes the ensemble on the very split it then claims as evaluation. The resulting numbers are inflated relative to a fair evaluation. The cleanest fix is the cross-validated stacking framework of Wolpert [4]: train each base learner K times under a K-fold partition, collect each base learner's predictions on the held-out fold only, and fit the meta-learner on the

resulting out-of-fold cube. Every prediction in that cube was produced by a model that did not see the corresponding example during training, so the meta-learner's weights themselves are unbiased.

b) Problem 2: A second leakage path through synthetic descendants.: When synthetic data is mixed into the training set, a new leakage path appears that has no equivalent in the real-only regime. A synthetic image generated from a real parent x inherits some of x's lighting, texture, and rim geometry, even if the rendered scene is otherwise unrecognisable. If x later lands in a held-out fold, training the ensemble on synthetic descendants of x leaks information about x into the training pool of the very fold that is supposed to be evaluating on it. The fix is also structural: every synthetic record must carry a pointer to its real parent, and the fold-construction machinery must filter out synthetic descendants of every held-out parent. We adopt this filter as a one-line set operation per fold (Section III-F1) and treat it as an artefact-level discipline that any pipeline mixing synthetic and real data should adopt under cross-validation.

c) Problem 3: False positives on visually confusable non-defects.: A segmenter trained on positive sand-boil imagery alone tends to over-fire on visually similar non-defects (puddles with sediment rings, hoof-print puddles, mole and gopher mounds, dried-mud polygons, half-buried cobbles, algal mats, and so on) that share dome-like or annular structure with a real sand boil. Panta *et al.* [2] address this with a second training phase that exposes the segmenter to defect-free levee imagery drawn from the same inspection archive. We extend that protocol with two changes. First, on top of the real defect-free pool we generate a synthetic hard-negative pool spanning a curated catalogue of visually confusable categories, produced by a defect-disabled production preset of the upstream diffusion pipeline [5]. Second, each synthetic negative is scored by the public SandBoilNet baseline and weighted into the negative phase in proportion to how strongly it fools the baseline. This converts the negative phase from a static second pass into a confusion-driven training loop in which the most informative synthetic negatives receive the largest loss weight.

d) Contributions.: This paper makes two empirical contributions and four methodological ones:

- An Updated SandBoilNet (ConvNeXt-S encoder, U-Net decoder, spatial-and-channel squeeze-and-excitation (scSE) skip attention) reaching 0.707 IoU against 0.608 for the published original re-evaluated on the same split. Neither heavier attention variants nor a four-fold resolution increase improve on it: the modernised encoder, not added attention capacity, moves accuracy here.
- A controlled study of ensembling and synthetic data in this scarce-data regime. The calibrated stack does not exceed the best single model, an outcome eight meta-learner families reproduce and which we trace to member correlation; synthetic augmentation pays when the pool is filtered for label fidelity; and neither a hard-negative phase nor higher input resolution improves on the champion. Establishing which additions transfer to a scarce-data defect, and which do not, is itself useful guidance.
- A leak-free cross-validated stacking *protocol* whose ensemble accuracy is unbiased by construction. Its contribution is the trustworthiness of the measurement rather than the size of a gain: here that measurement returns no ensemble gain at all, a conclusion the leaky protocols it replaces cannot reach, since their headline numbers conflate an ensemble's benefit with the tuning that produced it.
- A parent-image fold filter closing the silent leakage path through synthetic descendants: each synthetic record carries a `source_image_id` pointer to its real parent, and the fold-construction driver excludes every descendant of a held-out parent in $O(1)$.
- A confusion-score-mined synthetic hard-negative phase, in which a defect-disabled generator preset produces visually confusable negatives and the sampler weights each in proportion to how strongly it fools the public baseline.
- A mask-conditioned synthesis method, MaskCN (Section III-C), that renders a fresh sand boil into a *target* mask, so the conditioning mask is the ground-truth label by construction. It supplies fully labelled images at zero annotation cost, has no real parent to leak, and adds procedural multi-boil fields the real set cannot provide.

e) Scope.: The present paper is the second in a three-paper series. The diffusion-based pipeline that produces both the augmented training imagery and the synthetic hard-negative imagery consumed here is the subject of the companion paper [5]; it is treated in Section III-B only as an upstream module that emits parent-tagged synthetic records. The deployment system, human-in-the-loop correction interface, and curated dataset release are the subject of a third paper in the series.

The remainder of the paper is organised as follows. Section II positions the contribution against the literature on segmentation architectures for limited-data regimes, stacked generalisation, calibration, and hard-negative mining. Section III develops the framework. Section IV describes the experimental protocol. Section V reports quantitative and qualitative results. Section VI examines the design choices and remaining failure modes. Section VII concludes.

II. Related Work

The relevant literature spans five threads: sand-boil and levee-defect segmentation; encoder–decoder architectures for limited-data regimes; synthetic-data augmentation; ensemble learning with stacked generalisation and calibration; and hard-negative mining. The generative machinery behind the synthetic data is covered only to the depth the segmentation framework requires, a fuller account belonging in the companion generation paper [5].

A. Sand-boil and levee-defect segmentation

Early automated levee-defect detection relied on edge filters, thresholding, and handcrafted features fed into shallow classifiers [6], which fared poorly under the lighting variation, mud, vegetation, and reflective standing water typical of in-field inspection photographs. Modern work is built on convolutional encoder–decoders. Panta *et al.* introduced IterLUNet [7] for levee crack segmentation and the more directly relevant SandBoilNet [2], an encoder–decoder on a partially fine-tuned ResNet-50v2 backbone whose skip connections carry PCA-based Channel and Spatial Attention with Inception blocks. Its reported intersection-over-union of 0.5743, balanced accuracy of 0.8552, and macro-averaged F1 of 0.7312 provide the principal point of comparison here. The same group later applied related ideas to seepage detection [8], [9] and to imbalance-aware culvert-and-sewer defect segmentation [10], and the same family extends to marine oil-spill delineation [11]. The field has since moved toward UAV-borne inspection, with thermal-infrared and visible-light imagery feeding detectors for seepage and embankment piping at field scale [12], [13] and frequency-domain and pruned-attention networks pushing toward real-time edge deployment [14], [15]. Those works target detection or coarse localisation from aerial platforms, whereas we address pixel-level segmentation of close-range photographs. Kuchi *et al.* [16] first paired a small synthetic sand-boil set with a CNN classifier, but operated at image level. Neighbouring work includes sinkhole detection with a depthwise separable U-Net [17], while synthetic-aperture radar [18] offers insufficient resolution for features this small. No prior sand-boil study combines its models through a leak-free cross-validated stacking ensemble; our earlier work [3] prototyped a weighted-average ensemble whose weights were tuned on the split later used to report its accuracy, a mild leakage we address in Section VI-A.

B. Segmentation architectures for limited-data regimes

Semantic segmentation assigns a class label to every pixel [19], and current architectures build on Fully Convolutional Networks [20] by replacing dense classifier heads with convolutional decoders. Common variants include PSPNet [21], SegNet [22], and DeepLabv3 [23]. For the small-dataset regime characteristic of infrastructure-defect imagery the U-Net family [24] remains dominant, since its skip connections preserve the narrow boundary features that delimit a boil's rim. Notable extensions include MultiResUNet [25], the nested U-Net++ [26], and Attention U-Net [27]. The five backbones evaluated here are deliberately not confined to that family: an architecture search retains the best of five complementary families, comprising an Updated SandBoilNet, a vision transformer, a nested convolutional U-Net, a multi-scale pyramid network, and an atrous network (Section III-D), chosen because their differing inductive biases make errors independent enough to stack while a shared input–output geometry keeps their out-of-fold predictions combinable. The wider defect-segmentation field is in parallel migrating from convolutional encoder–decoders such as DeepCrack [28] toward transformer and hybrid designs [29], [30] and toward prompting vision foundation models [31], [32], a shift enabled by consolidated benchmarks [33], [34]. Surveys of segmentation under scarce annotation [35], [36] confirm that strong augmentation and regularisation, rather than ever-larger backbones, are the dominant levers in this regime.

C. Generative synthetic data augmentation

Expanding scarce defect datasets with generated imagery has progressed from GAN-based synthesis [37], [38] to diffusion models that jointly emit images and pixel-aligned masks [39], [40]. The crucial design axis is where the label comes from. One family *infers* a mask for each generated image, by thresholding an attention map [39] or a separately trained segmenter, which reintroduces annotation noise and, when the generator is conditioned on a real seed, a parent linkage. A second family *conditions* the generator on a target mask so that the mask is the label by construction, made practical at high resolution by ControlNet and related adapters [41] and followed by scribble-conditioned diffusion [40]. Our MaskCN route (Section III-C) sits in this second family: a mask-ControlNet trained jointly with a sand-boil DreamBooth low-rank adaptation [42], [43] renders a fresh boil into a prescribed mask, so the conditioning mask is the annotation at zero labelling cost and, inheriting no real seed, has no parent linkage to leak. In the defect domain, LoRA-fine-tuned latent diffusion lifts segmentation mIoU on steel-surface inspection [44], classifier-side studies report that diffusion-augmented training can approach or exceed real-only baselines [45], [46], and mask-preserving augmentations such as copy-paste remain strong, simple baselines [47], [48]. A recurring caution, central to this paper, is that naive use of synthetic data inflates reported accuracy through a subtle leakage between synthetic samples and their real parents [49], an instance of the broader data-leakage problem in applied machine learning [50], [51]. Our parent-image fold filter (Section III-F1) closes exactly this path, and MaskCN avoids it by construction.

D. Ensemble learning and stacked generalisation

Combining several segmentation models reduces prediction variance and often exceeds the best single base learner, though, as our results show (Table II), not always. Stacked ensembles have been applied to concrete-crack and damage segmentation [52], [53], and deep ensembles are a standard route to accuracy and calibrated uncertainty [54]. Simple strategies such as pixel-level majority voting and weighted probability averaging [3] are easy to implement but carry two weaknesses here: they treat base models as interchangeable, ignoring the per-architecture calibration that follows from temperature scaling [55], [56], and they admit a leakage mode whenever per-model weights are

tuned on the same split later used to report ensemble accuracy.

Cross-validated stacking, formalised by Wolpert [4], eliminates both. The training set is partitioned into K folds, each base model is trained once per fold and predicts on the held-out fold, and the resulting out-of-fold predictions are each produced by a model that did not observe the corresponding example, so the estimate is unbiased. A meta-learner is then fitted on that out-of-fold matrix. Although stacking is ubiquitous in tabular learning, its use for pixel-level semantic segmentation has been limited. The present paper instantiates it at the pixel level, prepends a per-architecture temperature calibration step so that base probabilities are comparable, and pairs it with a parent-image filter that keeps synthetic descendants of held-out images out of the corresponding training folds. Calibration is a complementary literature: Guo *et al.* [55] showed that modern deep classifiers are systematically overconfident and that a single positive temperature scalar fitted on held-out data suffices to recalibrate them. We apply this architecture-by-architecture to the out-of-fold cube, since uncalibrated probabilities let the most overconfident architecture dominate the meta-learner regardless of its accuracy.

E. Hard-negative mining

Hard-negative mining [57] reduces false positives by preferentially training on negatives the current model misclassifies, and classically operates on an unlabelled real pool. We substitute a controllable, infinitely extensible synthetic pool: a defect-disabled preset of the upstream pipeline [5] generates visually confusable negatives on demand, the publicly released baseline segmenter scores each one, and the sampler weights it in proportion to how strongly it fools that baseline. This combination of a controllable synthetic pool with baseline-driven confusion weighting is, to our knowledge, novel in the civil-infrastructure defect-segmentation literature (Section III-G).

Against this backdrop, two methodological moves address gaps the ensembling and leakage literature leaves open: leak-free cross-validated stacking with per-architecture calibration, instantiated at the pixel level; and a parent-image fold filter that closes the silent leakage path through synthetic descendants. The empirical contribution is an honest account of which of these additions transfer to a small, fuzzy-boundary defect and which do not.

III. Methodology

The segmentation framework comprises four coupled components: five encoder–decoder backbones trained under five-fold cross-validation, a parent-image filter that closes the synthetic descendant leakage path, a per-architecture temperature calibration step, and a per-pixel meta-learner that combines the calibrated out-of-fold probabilities into a single stacked output. A synthetic hard-negative training phase is added on top of the principal training run, in which the publicly released baseline segmenter scores each synthetic negative and the negative-phase sampler weights each one in proportion to its confusion score. Figure 1 shows the framework as a whole; Figures 6 and 8 expand its training and inference halves.

A. Dataset

The real sand-boil dataset is drawn from the U.S. Army Corps of Engineers (USACE) levee-inspection archive [58]; Figure 2 shows representative images with their ground-truth masks. Field inspectors traverse the crest and adjacent areas of an earthen levee and photograph any abnormalities they encounter. A domain-expert-curated subset of approximately three hundred photographs was retained on the basis of visibility and unambiguous appearance, then further filtered by automatic intersection-over-union agreement between the manual annotation and an initial Segment Anything proposal [59]. The final curated set used in the present study contains N_{train}=199 training images and N_{test}=46 held-out test images at variable native resolutions, each paired with a binary pixel-level mask. The prose elsewhere refers to "the training set" and "the held-out test set" to avoid restating these absolute numbers. For downstream segmentation, images and masks are resampled to 512×512 pixels.

B. Upstream synthetic-data source

The augmented training set and the synthetic hard-negative pool consumed by the framework below are both produced by the diffusion-based generation pipeline of our companion paper [5]; Figure 3 places representative synthetic images beside their real parents. Our group has previously released public synthetic levee-defect datasets in this line of work through IEEE DataPort, covering sand boils [60] and animal burrows [61]. We recap only what the segmentation framework needs to know.

The augmented training set comprises five synthetic variations of every real training image, generated by a DreamBooth-fine-tuned Stable Diffusion XL backbone with multi-ControlNet conditioning under a soft-mask inpainting protocol that preserves the real sand-boil region pixel-for-pixel and re-renders the surrounding scene. Each synthetic record carries a `source_image_id` pointer to its real parent (the *single* field the present paper requires from the upstream pipeline) together with the prompt, seed, and generator preset. Candidates are admitted only where all five cross-validation folds of a held-out segmenter reach an intersection-over-union of 0.70 against the image's own mask, a label-fidelity filter applied uniformly across presets; the survivors then pass a CLIP-similarity quality gate and are sub-sampled by greedy farthest-point selection in CLIP feature space, so the retained pool is diverse rather than merely accurate. The synthetic ground-truth mask is produced by running the synthetic image through the Convex_Hull_Annotator pipeline of [3], [62], which thresholds the public SandBoilNet probability map

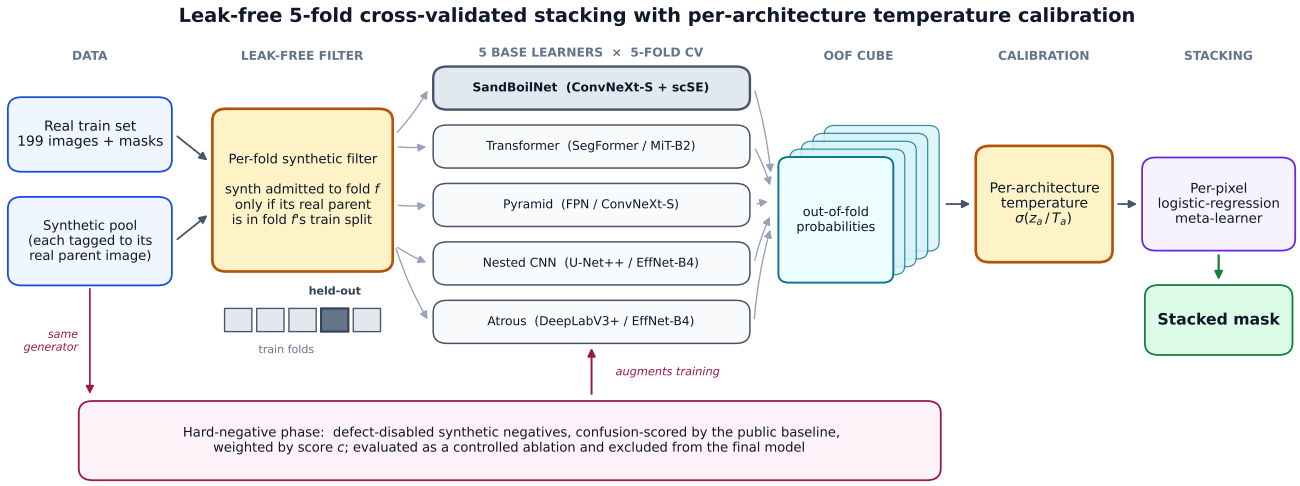

Fig. 1. The segmentation framework at a glance. Real and synthetic data pass through the leak-free per-fold filter; five backbones are trained under five-fold cross-validation; their out-of-fold probabilities are recalibrated by one temperature scalar per architecture and combined by a per-pixel meta-learner. The lower branch is the synthetic hard-negative phase, evaluated as a controlled ablation and excluded from the final model. The two steps drawn in amber are this paper's methodological contributions; neither is an accuracy lever, since the filter removes an upward bias rather than raising the score and calibration leaves test IoU unchanged while making the weights comparable.

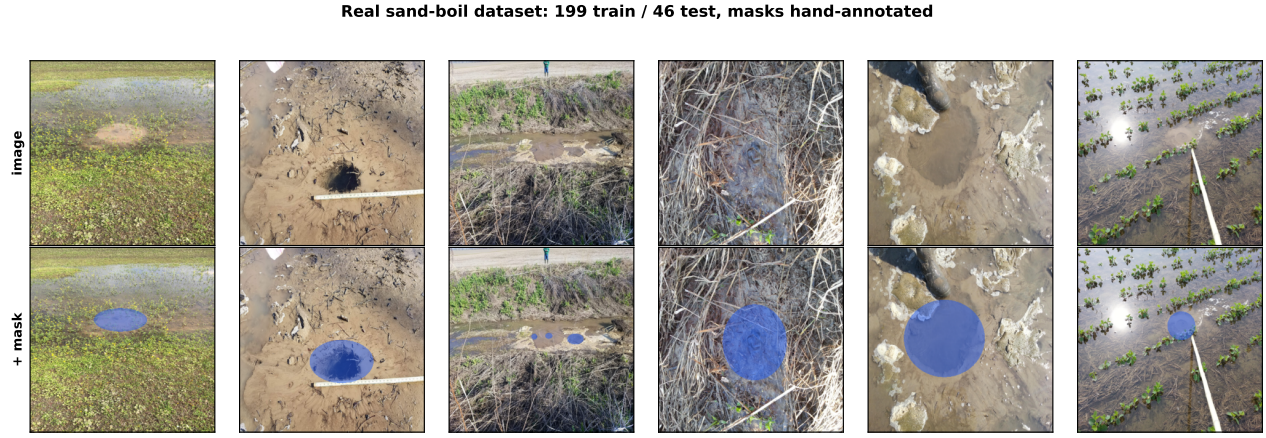

Fig. 2. Representative real sand boils (top) with their hand-annotated ground-truth masks overlaid (bottom). The dataset spans a wide range of boil sizes, water clarity, vegetation, and lighting; 199 images are used for training and 46 are held out for test.

and cleans the result with connected-component analysis and convex-hull deburr. For the quality-filtered pool of Section V-G the labels are instead re-derived with the strongest segmenter available here and morphological closing only, since a convex hull cannot represent the concave rim of a boil.

The synthetic hard-negative pool is produced under a separate production preset (denoted V_{neg}) of the same pipeline. V_{neg} disables the sand-boil DreamBooth adapter at inference and drives the base SDXL backbone with a category-specific prompt and a strong negative-prompt clause suppressing residual sand-boil structure. Synthetic negatives therefore share the visual style of the positive synthetic pool (same SDXL backbone, same VAE, same camera-look) without inheriting the sand-boil bias of the adapter.

C. Mask-conditioned synthesis (MaskCN)

The augmented and hard-negative pools above both descend from a real parent image and therefore carry the `source_image_id` that the leak-free filter requires. We additionally introduce a complementary *mask-conditioned* synthesis route, MaskCN, whose records have no real parent at all because the label is produced *with* the image rather than inferred from it (Figure 4). A second ControlNet [41], a mask-ControlNet trained on (mask, image) pairs together with the same sand-boil DreamBooth-LoRA adapter [42], [43], is driven in text-to-image mode and conditioned on a *target* sand-boil mask. It renders a new sand boil into that mask shape, so the conditioning mask is the ground-truth annotation of the resulting image by construction, requiring no Convex_Hull_Annotator pass and no manual labelling. Conditioning masks come from two sources: a bank of 41 real-derived mask silhou-

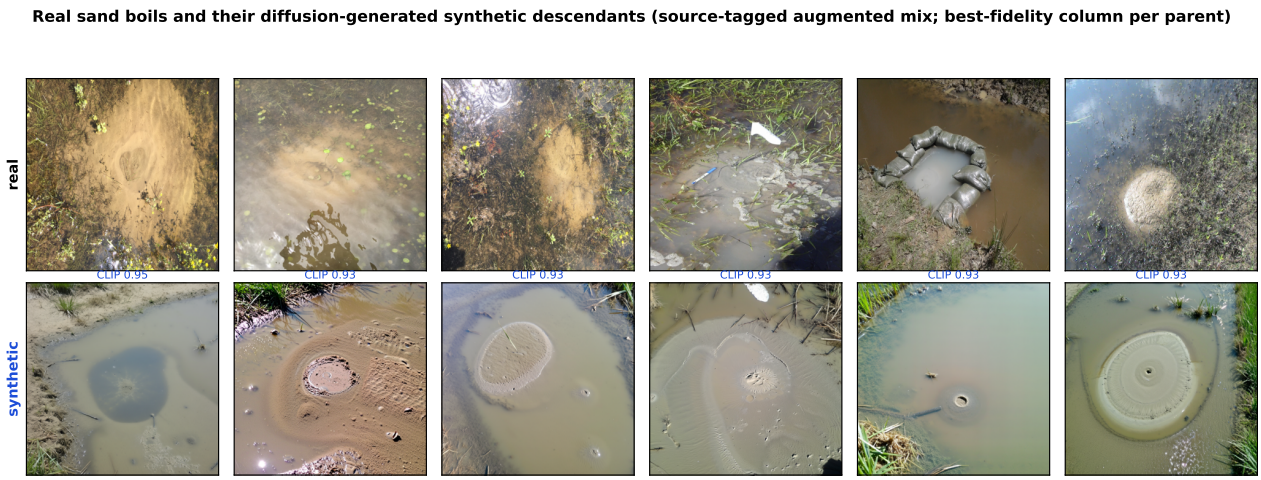

Fig. 3. Real sand boils (top) beside their best-fidelity synthetic descendants (bottom), produced by the companion generation paper [5] and carried here in the source-tagged augmented mix. Each bottom tile is the V_4-preset descendant scoring highest under CLIP image similarity to the parent above it, and each carries a `source_image_id` pointer to that parent, the field on which the filter of Section III-F1 keys.

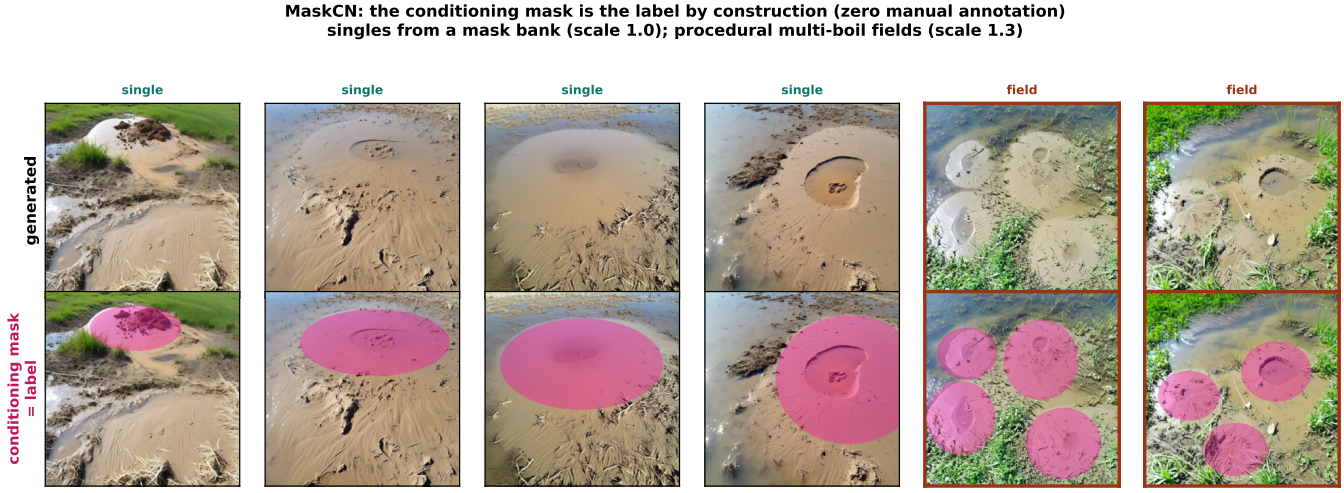

Fig. 4. Mask-conditioned synthesis (MaskCN). A mask-ControlNet [41] trained jointly with the sand-boil DreamBooth-LoRA [42], [43] renders a fresh sand boil *into* a target mask under a text-to-image preset, so the conditioning mask (bottom row, magenta overlay) *is* the ground-truth label of the generated image (top row) by construction, at zero annotation cost. Four single-boil tiles are conditioned on a mask bank (ControlNet scale 1.0); the two red-bordered tiles are procedural multi-boil fields (scale 1.3). MaskCN conditions on the mask alone. Its value lies in zero-cost labelling and controlled diversity rather than in raw accuracy (Table VIII).

ettes (single boils, ControlNet conditioning scale 1.0) and a procedural generator that scatters random ellipses to form multi-boil *fields* (conditioning scale 1.3). All MaskCN images are rendered at CFG 8.5 with the DPM-Solver++ 2M Karras sampler [63] and an fp16-fix VAE. Unlike the img2img augmented pool, MaskCN conditions on the mask alone: it carries none of the Canny, Depth, HED, Normal, denoising-strength, or IP-Adapter terms of the other presets.

MaskCN matters to the segmentation framework for two reasons that are independent of any accuracy claim. First, it is a *zero-annotation labelling* method: the most expensive ingredient in this domain is the pixel-level mask, and MaskCN supplies a perfectly registered mask for free with every image it generates. Second, it is a *diversity* engine: by sampling fresh boils into arbitrary mask shapes (including procedurally generated multi-boil fields with no real-image analogue), it produces labelled training material whose boil count, size, placement, and scene context can be dialled independently of the 199-image real set (Figure 16). Because MaskCN records have no real parent, they sidestep the synthetic-descendant leakage path of Section III-F1, with one caveat: the conditioning masks themselves have sources, and a mask drawn from a held-out image would leak that image's label geometry directly. The evaluation pool therefore restricts the mask bank to training-set sources (Appendix B). We are explicit about scope: the present paper introduces MaskCN as a labelling-and-diversity method. Evaluated once as a direct augmentation source under the per-generator protocol of Section V-G, its leak-safe pool is competitive with the parent-derived pools (Table VIII), though at that scale the

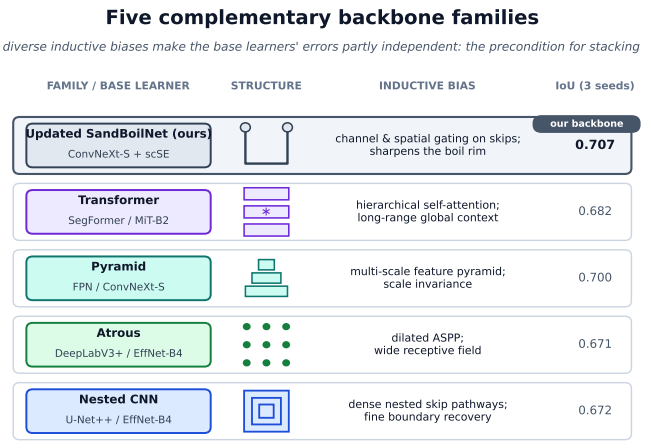

Fig. 5. The five backbone families retained as base learners, with the locked encoder–decoder pairing and the inductive bias each contributes. The families are chosen so their errors are partly independent, the condition under which stacking improves on any single model, while a shared 512×512 geometry keeps their predictions combinable. The right column gives each champion's architecture-search IoU (Table I). The highlighted row marks SandBoilNet as the anchor of the ensemble rather than the single-split search winner.

differences between pools are within run-to-run variation.

D. Segmentation backbones

Rather than commit to one architecture, we run an architecture search and then stack a small, deliberately diverse set of the strongest models. The search covers eleven encoder–decoder networks that all emit a single-channel probability map at the input's 512×512 resolution, drawn from five distinct architectural *families*. The diversity is the point: a stacking ensemble only helps to the extent that its members fail in different places, so within each family we keep only the best performer and discard the rest. Stacking two networks of the same kind buys almost nothing, because they tend to make the same mistakes on the same pixels. All encoders are ImageNet-pretrained, which is decisive at this data scale, and each is trained with an architecture-appropriate learning rate (lower for the transformers), linear warmup, gradient clipping, and the heavy online augmentation of Section IV-A. The five families (Figure 5), and the model that represents each in the final ensemble, are as follows.

1) **Attention U-Net (Updated SandBoilNet).** The domain-specific SandBoilNet of Panta *et al.* [2] keeps a U-Net shape but enriches each skip connection with attention. Its original Principal-Component-Analysis attention is computationally heavy and numerically fragile; we replace it with a lightweight spatial-and-channel squeeze-and-excitation (scSE) gate [64] and swap the dated ResNet-50 encoder for a ConvNeXt backbone [65]. This Updated SandBoilNet is the strongest single model once its folds are averaged (Table II) and anchors the ensemble, though on the single-split search FPN edges it (Table I).
2) **Transformer.** A hierarchical vision transformer (SegFormer [66]) replaces convolutional locality with global self-attention, giving error modes that differ sharply from the convolutional models and therefore contribute the most independent signal to the stack.
3) **Nested convolutional U-Net.** U-Net++ [26] with an EfficientNet encoder [67], whose dense nested skip pathways fuse features across decoder depths.
4) **Pyramid / multi-scale.** The best of a feature-pyramid (FPN [68]), pyramid-pooling (PSPNet [21]), or multi-scale-attention (MAnet [69]) decoder, all of which aggregate context at several spatial scales.
5) **Atrous (dilated).** DeepLabV3+ [70], whose atrous spatial-pyramid pooling widens the receptive field without sacrificing output resolution.

All five share a common input–output geometry, which is what makes their out-of-fold probability maps directly stackable at the pixel level. Within each family the search retains a single locked encoder–decoder champion (U-Net++/EfficientNet-B4, SegFormer/MiT-B2, FPN/ConvNeXt-S, DeepLabV3+/EfficientNet-B4, and the ConvNeXt-S+scSE SandBoilNet), and these champions, with their held-out scores, are reported in Table I; the original five U-Net variants of our earlier work [3] are retained there only as baselines. These five family champions are the stacking members: the ensemble reported in Section V is exactly this set, one network per architecture family, with no substitutions.

E. Texture-sensitive skip attention

A sand boil is distinguished from the surrounding mud, water, and vegetation chiefly by its fine sandy *texture*, so we examine whether a texture-sensitive skip-connection gate helps the champion backbone. The scSE attention recalibrates decoder features with first-order (mean) channel statistics and a single-map spatial gate [64], [71]. We study a texture-sensitive variant, *Texture-SE*, that replaces the mean channel descriptor with a second-order (standard-deviation) one, so a channel's texture energy rather than its average activation informs its weight. Two further variants probe whether additional machinery helps: a texture–contrast form (TC-SE) that also replaces the spatial gate with a center–surround contrast term $x - \mathrm{blur}(x)$, and a multi-scale dilated spatial form (Atrous-SE). All add fewer than 0.2 M parameters and, unlike the PCA attention of the original SandBoilNet, involve no eigendecomposition and are numerically stable. The ablation of Section V-G finds that none of these gates improves on the plain scSE skip attention on the present split, so the champion retains scSE and the stacking ensemble is built from the five architecture families rather than from attention variants of one backbone.

F. Cross-validated stacking

Figures 6 and 8 give the training and inference halves of the pipeline; the remainder of this section specifies each stage. The five backbones are trained under five-fold cross-validation. The real training set is partitioned into

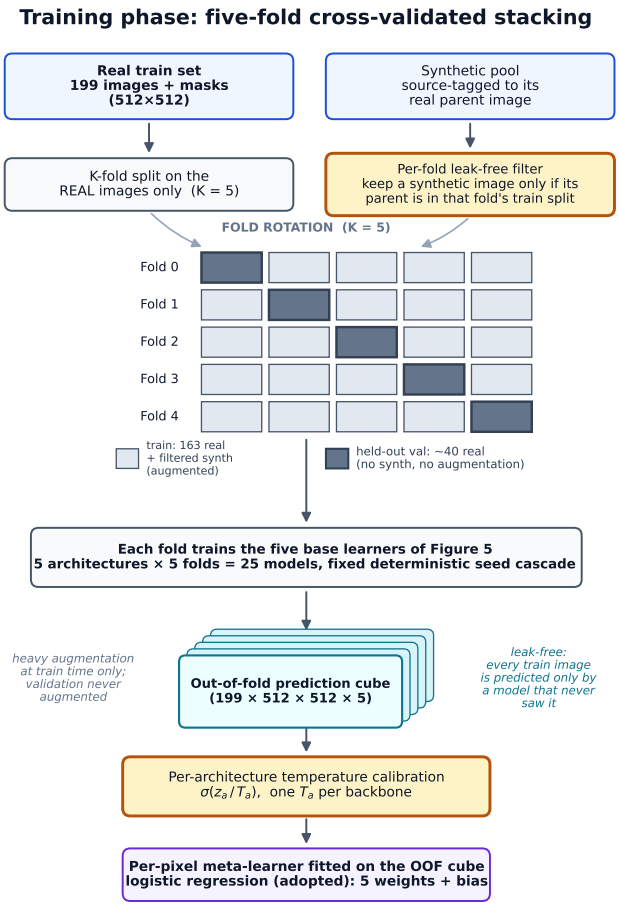

Fig. 6. Training phase. The real training set is split into five folds on the real images only; the optional synthetic pool passes a per-fold leak-free filter that keeps a synthetic image only when its `source_image_id` parent lies in that fold's training split. In the fold-rotation matrix the amber cell marks each fold's held-out slice, so every real image is held out exactly once. Each base learner is trained once per fold (25 models), and the held-out-fold predictions form the out-of-fold cube that is calibrated and combined. Test-time inference is shown in Figure 8.

K=5 approximately equal folds via a seeded shuffle. The partitioning seed is fixed at 42 and recorded in the artefact manifest, so the same fold assignment is reproduced on every run. For each fold f and each architecture a a model is trained on the union of the other $K-1$ folds and predicts on the held-out fold, producing one out-of-fold prediction per real training image per architecture. The encoders start from ImageNet-pretrained weights and the decoders from random initialisation. Each (architecture, fold) pair draws an independent seed from a fixed deterministic cascade, seed $=$ seed$_0$ $+$ $1000\,a$ $+$ f, so every fold assignment and initialisation is reproducible from the base seed. The fold models of one architecture therefore differ in both training data and initialisation, which adds benign diversity to the out-of-fold ensemble. One caveat is recorded for exactness: the augmentation library draws its transform parameters from an internal generator that this cascade does not govern, so bit-exact repetition of a training run additionally depends on the pinned augmentation library version. The full out-of-fold collection is stored as a memory-mapped float-16 array of shape $(N_{\text{train}}, H, W, A) = (199, 512, 512, 5)$ with a sidecar JSON manifest listing the image identifiers in row order. Each (architecture, fold) write goes through the manifest, so the procedure is crash-safe and resumable.

a) Fold-model health screen.: Ensemble quality is sensitive to a failure mode that per-fold validation IoU alone does not expose: a fold model occasionally converges to a solution whose background probabilities floor well above zero (around 0.08 in our runs) instead of decaying toward it. The thresholded masks, and therefore the validation IoU, can remain unremarkable, but the inflated background mass distorts that architecture's temperature fit and feeds the meta-learner a mis-scaled column. Every (architecture, fold) model is therefore screened on its out-of-fold predictions before any calibration or meta-learner fitting: a model is accepted only if at least half of its out-of-fold pixels receive a probability below 0.1 (healthy fold models assign well over 80%) and its held-out IoU reaches 0.5. A model that fails the screen is retrained with a fresh seed drawn from the same deterministic cascade, up to four attempts, keeping the healthiest attempt. The screen consults training-side out-of-fold data only, so it introduces no test leakage.

b) Fold count and nested stacking.: The out-of-fold predictions used to fit the meta-learner come from *single*-fold models (trained on $K-1$ folds), whereas at test time each architecture is the average of its K fold models. To probe the effect of this mismatch we ablate the fold count, training the stack at K=5 and K=10. The larger count gives each fold model 90% of the data, so its out-of-fold prediction sits closer to the fold-averaged test model. We also evaluate a *nested* cross-validated stack that removes the mismatch entirely: for each outer fold, an inner K=5 cross-validation produces a five-model average on the held-out outer fold, matching the test-time averaging exactly. The three variants are compared in Section V-B.

1) Leak-free synthetic-data inclusion: The parent-tagged synthetic record structure makes the leak-free filter trivial. For every fold f, the set of real identifiers allocated to fold f's validation split is collected, and any synthetic image whose `source_image_id` falls in that set is excluded from fold f's training pool. No fold ever sees a synthetic descendant of one of its own validation images. The filter reduces to a one-line set intersection per fold:

$$\mathcal{T}_f^{\text{aug}} = \mathcal{D}^{\text{aug}} \setminus \left\{ s \in \mathcal{D}^{\text{aug}} : \text{parent}(s) \in \mathcal{V}_f \right\}, \qquad (1)$$

where $\mathcal{D}^{\text{aug}}$ is the full augmented set, $\mathcal{V}_f$ is the set of real validation identifiers for fold f, and parent(s) is the `source_image_id` of synthetic record s. The filter is $O(|\mathcal{D}^{\text{aug}}|)$ and runs once at fold-construction time; Figure 7 illustrates how it removes a held-out parent's synthetic descendants from that fold's training pool.

2) Per-architecture temperature calibration: Different segmentation backbones produce probability outputs over different confidence ranges. Naively stacking uncalibrated probabilities allows the most overconfident architecture to

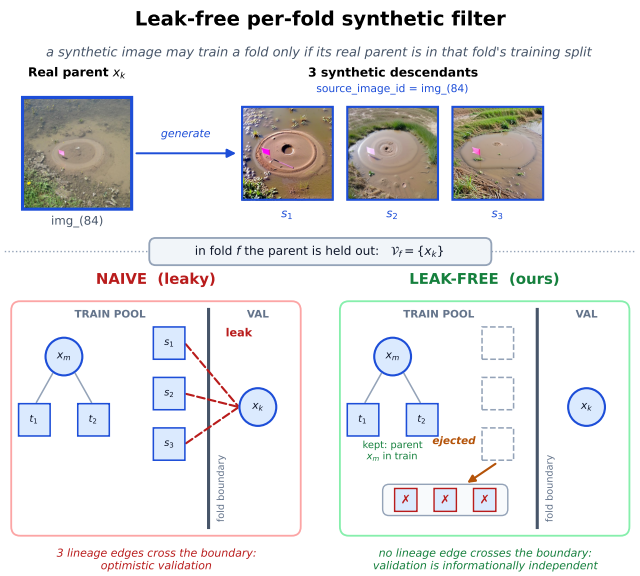

Fig. 7. The leak-free per-fold filter as a constraint on the fold boundary. Top: a real parent and three of its synthetic descendants, linked by `source_image_id`. Bottom: when fold f holds out the parent x_k, the naive pool (left) keeps its descendants s_1–s_3 in training, so three lineage edges cross the fold boundary and validation is optimistic. The filter (right) ejects exactly those descendants via Eq. (1), while the family of a parent x_m still in training is retained.

dominate the meta-learner output, regardless of whether it is the most accurate. We therefore apply temperature scaling [55] to each architecture's out-of-fold predictions: a single positive scalar T_a is fitted to minimise binary cross-entropy against the real ground-truth masks, and the calibrated probability is

$$p_a^{\text{cal}} = \sigma\big(\sigma^{-1}(p_a)\,/\,T_a\big), \tag{2}$$

where σ is the sigmoid function. The scalar is fitted by a one-dimensional grid search over 26 evenly spaced values of $T_a \in [0.5, 3.0]$, evaluated on a fixed-seed subsample of one million out-of-fold pixels, requiring no automatic differentiation. A temperature $T_a > 1$ indicates that the architecture was overconfident and was softened; $T_a < 1$ indicates that it was underconfident and was sharpened.

3) Per-pixel meta-learner: The meta-learner consumes the five calibrated probabilities at one pixel and outputs a single probability for that pixel. The default realisation is logistic regression (3), parameterised by a five-vector $\mathbf{w}$ of architecture weights and a scalar bias b:

$$p^{\text{stack}}(i,j) = \sigma\big(\mathbf{w}^\top \mathbf{p}^{\text{cal}}(i,j) + b\big)\,, \tag{3}$$

where $\mathbf{p}^{\text{cal}}(i,j) = (p_1^{\text{cal}}(i,j), \ldots, p_5^{\text{cal}}(i,j))$. Training minimises ℓ_2-regularised binary cross-entropy (C=1) on a fixed-seed subsample of two million pixels drawn from the out-of-fold cube. A useful by-product of the linear form is interpretability: each w_a admits a direct reading as the relative contribution of architecture a to the final stacked output. The choice of meta-learner is itself an ablation axis: we compare logistic regression against seven alternatives: a multi-layer perceptron, linear, RBF and polynomial SVMs, a random forest, histogram gradient boosting, and XGBoost [72], all fitted on the same calibrated out-of-fold cube with the exact configurations recorded in Appendix A and reported in Table III. We adopt logistic regression for its directly interpretable weights.

Algorithm 1 Leak-free K-fold stacking with per-architecture calibration

Input: real set R; synthetic pool S (each image tagged with its real parent); architectures A; folds K
Train ($|A|\times K$ models, out-of-fold cube):
1: **for** each fold f and architecture a **do**
2: $S_f \leftarrow \{\, s \in S : \text{parent}(s) \notin f \,\}$ ▷ leak-free filter, Eq. (1)
3: train $M_{a,f}$ on $(R \setminus f) \cup S_f$; store its fold-f predictions
4: **end for**
Calibrate + combine (out-of-fold only):
5: per architecture, fit temperature T_a and rescale ▷ Eq. (2)
6: fit per-pixel meta-learner on the calibrated cube; pick threshold τ
Predict: for test image x, average each architecture's K folds, rescale by T_a, combine, threshold at τ.

Algorithm 1 collects the stages above into the procedure actually run. Two properties are visible directly from the listing rather than argued in prose: the fold partition is drawn on the real images alone (line 1), and the parent-image filter is applied *inside* the fold loop, before any training (line 2), so no fold can train on a synthetic descendant of one of its own validation images. Both the temperature scalars and the meta-learner are fitted only on out-of-fold predictions (lines 5 and 6); the held-out test set is first touched at line 6. The ensemble's accuracy is therefore unbiased by construction rather than by assumption.

4) Test-time inference: For each held-out test image, the K=5 fold checkpoints of each architecture produce K predictions; these are averaged within architecture (a form of bagging that reduces variance), then passed through the per-architecture temperature and into the meta-learner, which produces the final stacked segmentation.

G. Hard-negative phase

We adopt the negative-image training protocol of Panta *et al.* [2]. After the principal training run on annotated sand-boil images, each segmentation model is exposed to defect-free levee imagery (grassy crowns, undisturbed embankments, clean water channels) through a short second phase. The negative phase teaches the network to reject visually similar but non-defect regions and reduces the false-positive rate at inference. Approximately eighty real defect-free levee photographs are drawn from the same USACE inspection archive as the positive set, ensuring that lighting and camera profile are matched.

a) Synthetic hard-negative atlas.: On top of the real defect-free pool we generate a diffusion-synthesised pool of *visually confusable* hard negatives that target classes the public SandBoilNet baseline is known to over-fire on (Figure 9). The category list is curated from inspector-side reports of frequent false positives and from a prior

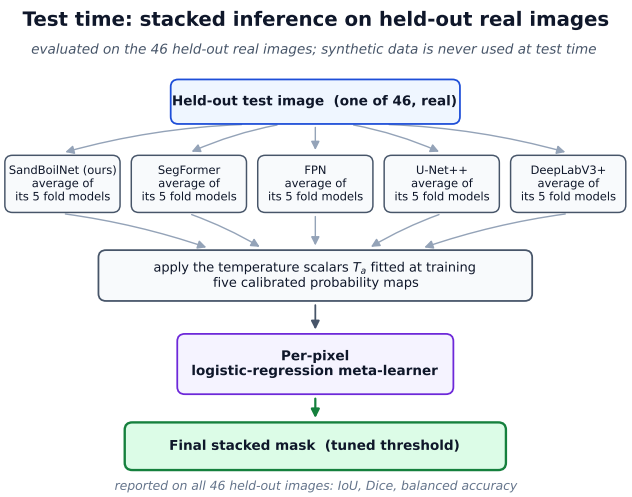

Fig. 8. Test-time inference. For each of the 46 held-out real test images, every architecture averages its five fold models and applies its fitted temperature scalar, yielding five calibrated probability maps. The per-pixel meta-learner combines them into the final binary mask, scored by IoU, Dice, and balanced accuracy. Synthetic data is never used at test time.

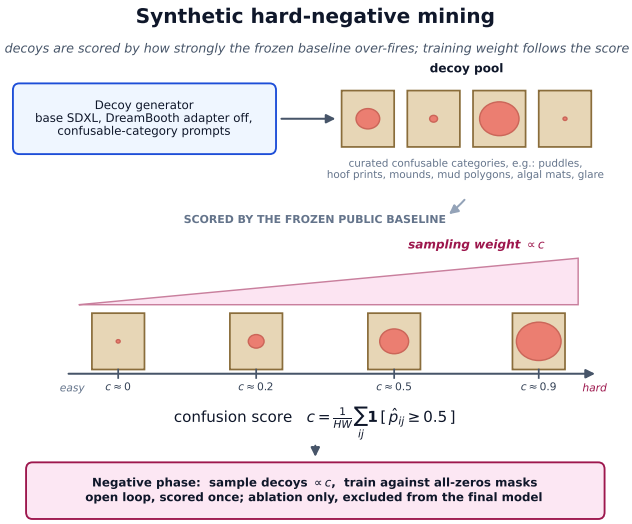

Fig. 9. Synthetic hard-negative mining as a curriculum. An adapter-off generator renders confusable non-boil decoys; the frozen public baseline's over-firing places each decoy on the confusion-score axis (red blob: the baseline's false-positive response; tiles illustrative), and the negative phase samples decoys with weight $\propto c$ (Eq. (4)) against all-zeros masks. The loop is open (each decoy is scored once), and the phase is evaluated as a controlled ablation and excluded from the final model.

misclassification audit: mud puddles with sediment rings, hoof-print puddles, tractor and ATV rut puddles, mole or gopher mounds, anthills, cracked dried-mud polygons, half-buried rocks and cobbles, algal mats, drainage-tile outlets, subsidence puddles, sheet-erosion sediment fans, catfish and turtle wallows, wet manure patties, reflective standing-water glare, surfacing tree roots, and algal bloom rosettes. For each category the upstream pipeline instantiates a hard-negative prompt atlas of approximately fifty contextual prompts using the same combinatorial-template machinery as the principal prompt atlas, with category-specific descriptors of the object combined with the shared field-context axes (geographic setting, lighting, weather, camera medium, composition). Each prompt carries a category-label tag and a negative-prompt extras clause ("no sand boil, no sand mound, no domed sand cone, no water erupting from the ground, ...") that is appended verbatim to the diffusion negative-prompt slot at inference time.

b) Generation preset.: Negatives are rendered through a production preset of the upstream pipeline [5], denoted V_{neg}, that reuses the prompt-driven scaffold (weak Canny only, no HED, no Normal, no IP-Adapter, no soft-mask, near-text2img CFG, high strength) but additionally sets the DreamBooth-LoRA adapter weight to 0 at inference. Disabling the LoRA is essential here: the adapter has been fine-tuned to *produce* sand-boil domes from the curated reference set, and leaving it at its default weight 1.0 would re-introduce a domed-mound bias into prompts that explicitly call for a puddle or a mole hill. With the adapter weight at 0, the base SDXL backbone follows the textual prompt, and the negative-prompt clause prunes residual sand-boil structure from the denoising trajectory. Synthetic negatives are generated at 1024×1024 and share the visual style of the positive synthetic pool (same SDXL backbone, same VAE, same camera-look as the LoRA's reference set), avoiding the style-mismatch problem that mixing a heterogeneous external image pool would introduce.

c) Confusion-score mining.: Synthetic hard negatives are not equally useful: an image that the segmenter already classifies correctly contributes little, whereas an image the baseline *misclassifies* as a sand boil is the most informative supervision for the negative phase. We therefore run the public SandBoilNet checkpoint of [2] over every synthetic negative and write a per-image confusion score

$$c = \frac{1}{HW} \sum_{i,j} \mathbb{1}[\hat{p}_{ij} \geq 0.5] \,, \tag{4}$$

i.e., the fraction of pixels the baseline labels positive. The negative phase consumes the union of the real defect-free pool and the synthetic hard-negative pool, with synthetic samples ranked by c and sampled with weight proportional to c so that the training loop focuses on the negatives the baseline is most fooled by. This mechanism is a lightweight form of hard-negative mining [57] adapted to a synthesised pool: instead of mining the unlabelled real pool for high-loss examples (the classical formulation), we mine a controllable, infinitely extensible synthetic pool. Negatives are trained against an all-zeros target mask, leaving the combined BCE plus Dice loss well-defined. The hyperparameters of the negative phase match the main phase except for a fixed learning rate of 1e−4 and a cap of 30 epochs.

H. Loss functions and metrics

Segmentation training minimises the combined binary cross-entropy plus Dice loss

$$\mathcal{L} = w_{\text{bce}}\,\mathcal{L}_{\text{bce}} + w_{\text{dice}}\,\mathcal{L}_{\text{dice}}, \tag{5}$$

with $w_{\text{bce}} = w_{\text{dice}} = 0.5$. The BCE term encourages correct per-pixel classification; the Dice term encourages overall region overlap, which compensates for the heavy class imbalance (sand-boil pixels constitute approximately 5% of the image).

Evaluation reports five measures: intersection-over-union (IoU) and Dice coefficient (Eq. (6)), mean IoU (mIoU), binary accuracy, and balanced accuracy (Eq. (7)). IoU and Dice both measure region overlap; balanced accuracy averages sensitivity and specificity and remains informative under class imbalance. The hard-negative analysis additionally reports recall (sensitivity, $TP/(TP+FN)$), one of the two components of (7), together with the false-positive rate ($FP/(FP+TN)$), the complement of the specificity term in (7).

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|}, \qquad \text{Dice} = \frac{2\,|P \cap G|}{|P| + |G|}, \tag{6}$$

$$\text{BA} = \tfrac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right). \tag{7}$$

a) Boundary tolerance and uncertainty.: Sand-boil rims are intrinsically fuzzy: the saturated dome fades into the surrounding mud over several pixels, so the exact rim placement of the ground-truth annotation is itself uncertain to within a few pixels. A strict IoU penalises any pixel that disagrees, conflating genuine misses with sub-annotation-noise boundary jitter. We therefore additionally report a *boundary-tolerant* IoU that counts a predicted boundary pixel as correct if a ground-truth boundary pixel lies within a tolerance band of t pixels (and conversely), reporting t=0 (the strict micro-average) alongside t=2 px. To quantify the sampling uncertainty that the small held-out set induces, we attach a non-parametric bootstrap 95% confidence interval to the champion's test IoU, resampling the held-out images with replacement; these two diagnostics are reported in Section V-D.

The reproducibility discipline that ties every reported number to its fold partition, seed cascade, and parent-image filter is detailed with the experimental setup in Section IV-F.

IV. Experiments

The experimental protocol is organised into three phases: a single-architecture baseline phase that establishes a floor for each backbone in isolation, a stacking-matrix phase that compares single-architecture, mean-ensemble, and cross-validated stacked ensembles under real-only and real-plus-synthetic training, and a hard-negative ablation phase that isolates the contribution of the confusion-score-mined negative pool.

A. Hardware and software

All experiments run on a SLURM-scheduled cluster of four NVIDIA H100 NVL GPUs (95 GB HBM each). The experiment matrix (the five-fold, five-architecture cross-validation runs together with the ablation and baseline jobs) is dispatched across the four GPUs with `sbatch`: the runs are mutually independent, so they are queued together and execute in parallel, and the full matrix completes with roughly four-way throughput. Each job requests a single GPU, trains its models to completion and exits, streaming logs to disk; the jobs are task-parallel and need no cross-GPU communication, so the matrix scales with the number of available devices rather than requiring distributed training. Per-job wall-clock times are reported in Appendix D.

Segmentation training and the cross-validated stacking runs use PyTorch [73] with automatic mixed-precision (AMP) autocasting and pinned-memory, multi-worker data loading, the Segmentation Models PyTorch library [74] for the encoder–decoder backbones and pretrained encoders, and Albumentations [75] for the joint image–mask augmentation pipeline. The meta-learners (logistic regression and the alternatives compared in Table III) and the per-architecture temperature calibration run on CPU with scikit-learn [76] and complete in under a minute.

B. Segmentation training protocol

Each segmentation experiment uses the combined BCE plus Dice loss (5) with the AdamW optimiser and a batch size of 8. Encoders are ImageNet-pretrained; decoders are randomly initialised. The learning rate is set per architecture and follows a five-epoch linear warmup into cosine annealing: 1e−4 for the CNN baselines, 8e−5 for ConvNeXt encoders, and 6e−5 for the transformer (weight decay 1e−4, raised to 1e−2 for the transformer); the per-architecture settings are tabulated in Appendix E. Training caps at 120–160 epochs depending on the family, with early stopping on validation IoU (patience 30); the best-IoU checkpoint is restored for testing. Spatial augmentations from a twenty-nine-transform Albumentations [75] pipeline (Figure 17, Appendix F) are applied jointly to image and mask; intensity and noise transforms are applied to the image alone. Validation uses identical preprocessing without augmentation.

C. Stacking experiment matrix

The stacking experiments compare three ensembling strategies (best single architecture; the mean ensemble of the backbones, with the prior weighted-average ensemble of [3] as a reference row; and leak-free cross-validated stacking) under two training-data conditions: real only, and real plus synthetic under the leak-free filter of Section III-F1. The held-out prediction of each architecture is the average of its five fold models, so every reported figure is a fold-averaged central estimate on the held-out real test set, scored by IoU, Dice, mean IoU, binary and balanced accuracy, and macro-F1. The headline

comparison (Table II) reports the real-only condition; the effect of synthetic augmentation on the strong ensemble is analysed in Section V-G. The choice of meta-learner *within* the cross-validated-stacking strategy (logistic regression, an MLP, linear/RBF/polynomial SVMs, random forest, histogram gradient boosting, and XGBoost) is a separate axis explored in Table III, holding the data and folds fixed.

D. Hard-negative ablation

The hard-negative phase is evaluated through a two-condition ablation. The *baseline* condition is the principal training run alone: Updated SandBoilNet trained on the real training set plus the augmented synthetic pool under the leak-free filter of Section III-F1, with no negative phase. The *negative-phase* condition applies, on top of that identical run, a short additional phase trained on the synthetic hard-negative pool of Section III-G, with samples weighted by their confusion score c of (4) (learning rate 1e−4, cap of 30 epochs). The contribution of the synthetic negative pool is read as the difference in false-positive rate (and in balanced accuracy) between the two conditions on the held-out real test set.

E. Per-architecture meta-learner weight analysis

We report the meta-learner weights normalised so that $\sum_a |w_a| = 1$, alongside the per-architecture calibration temperatures T_a of (2). The two are read together in Section V-E.

F. Reproducibility

Every numerical entry reported in Section V is the deterministic product of three identifiers: the fold partition seed (fixed at 42), the architecture and initialisation seed, and the `source_image_id` list of every synthetic record admitted to the training pool. Each segmentation training run records the architecture name, fold index, initialisation seed, optimiser hyperparameters, and a hash of its preprocessing pipeline. The upstream generation pipeline of the companion paper [5] emits a manifest of every adapter, prompt-bank revision, and seed cascade used to produce the augmented and hard-negative imagery consumed here. Together these identifiers allow any reported number to be re-derived from the released code and configuration files without manual reconfiguration.

V. Results and Analysis

We report the outcomes of the experimental phases of Section IV: the architecture search, the stacking matrix, the hard-negative ablation, an uncertainty analysis of the headline result, and the per-architecture meta-learner weights. The full combiner-by-design grid is given in Appendix A.

A. Architecture search

Before stacking, we select the base learners by an architecture search across the five encoder–decoder families of Section III-D, all trained under the heavy augmentation and per-architecture schedule of Section IV-A, keeping the best performer in each family since two networks of the same kind contribute little independent signal to the stack. Table I reports the resulting family champions on the held-out test set, with the original SandBoilNet shown for reference.

The Updated SandBoilNet leads at 0.707 IoU, 0.099 above the published original on the same split and a relative gain of 16%. Its margin over the nearest families is far smaller (0.700 for the pyramid variant, 0.697 for the gateless baseline), so the improvement follows mainly from the modernised encoder rather than the attention gate, and it is the most reproducible entry at ±0.003 over three seeds. That the same configuration also leads a disjoint defect dataset (Section V-H) is the stronger evidence that the ranking is not an artefact of this split.

B. Cross-validated stacking results

Table II reports the headline comparison: the best single architecture, the mean ensemble, and leak-free cross-validated stacking with the adopted logistic-regression meta-learner, on real-only training at the OOF-tuned operating point. The five architectures of Table I stack to 0.681 IoU against 0.694 for the strongest fold-averaged member, so the stack does not recover the best single model; the mean ensemble reaches the same 0.681, leaving the fitted meta-learner with no advantage over uniform weighting. The effect of synthetic training data is examined in Section V-G and the choice of meta-learner in Table III.

The comparison isolates the ensembling contribution against two references. The prior weighted-average ensemble of [3] (the practice that admits the mild leakage of Section II-D) and the originally published SandBoilNet are shown for context; the operative comparison is the best single architecture versus the mean ensemble and the calibrated cross-validated stacking of Section III-F, all on the same five fold-averaged base learners. Under that comparison ensembling does not pay: both combiners sit 0.013 IoU below the strongest member, and the fitted stack matches the unweighted mean to three decimal places, which is what one expects when the members' errors are largely shared rather than independent (Section VI-A). Figure 10 plots these systems against the best-single baseline across cross-validation designs, and Figure 11 the corresponding per-image IoU distributions, which show where the shortfall lies: the stack compresses the lower tail slightly but does not lift the median case, so it trades a little worst-case robustness for a small loss in typical accuracy. Figure 15 gives the precision–recall and ROC curves for the base learners and the stacked ensemble.

a) Additional baselines.: Two further baselines place the result in context. Pixel-level majority voting over the

TABLE I

ARCHITECTURE SEARCH: PER-FAMILY CHAMPIONS RETAINED AS STACKING BASE LEARNERS, AS MEAN ± STANDARD DEVIATION OF HELD-OUT TEST IOU OVER THREE SEEDS ($\{1, 2, 42\}$) ON A SINGLE 85/15 SPLIT. "UPDATED SANDBOILNET" IS SANDBOILNET MODERNISED WITH A CONVNEXT-S BACKBONE AND SCSE SKIP ATTENTION. UNDER THE FOLD-AVERAGED STACKING PROTOCOL THE STRONGEST MEMBER IS INSTEAD SEGFORMER (TABLE II), THE TWO REGIMES TRAINING ON DIFFERENT AMOUNTS OF DATA. THE REFERENCE ROW IS THE PUBLISHED CHECKPOINT RE-EVALUATED ON THIS PAPER'S 46-IMAGE TEST SET.

Family	Base learner	Test IoU (mean ± sd)
Attention U-Net (Updated SandBoilNet)	ConvNeXt-S + scSE	$\mathbf{0.707 \pm 0.003}$
Pyramid / multi-scale	FPN / ConvNeXt-S	0.700 ± 0.012
Attention U-Net, no gate	ConvNeXt-S	0.697 ± 0.009
Transformer	SegFormer / MiT-B2	0.682 ± 0.020
Nested CNN	U-Net++ / EfficientNet-B4	0.672 ± 0.014
Atrous	DeepLabV3+ / EfficientNet-B4	0.671 ± 0.021
reference	SandBoilNet (ResNet50V2 + PCA), our run	0.608

TABLE II

HEADLINE COMPARISON ON THE HELD-OUT REAL TEST SET AT THE LEAK-FREE OUT-OF-FOLD OPERATING POINT; EACH ARCHITECTURE'S PREDICTION AVERAGES ITS FIVE FOLD MODELS. NEITHER THE MEAN ENSEMBLE NOR THE CALIBRATED STACK REACHES THE BEST SINGLE MODEL, BOTH LANDING 0.013 IOU BELOW IT, AND THE STACK IS INDISTINGUISHABLE FROM THE SIMPLE MEAN. IN THIS FOLD-AVERAGED REGIME THE STRONGEST MEMBER IS SEGFORMER, WHEREAS THE UPDATED SANDBOILNET LEADS WHEN TRAINED ONCE ON THE FULL TRAINING SPLIT (TABLE I); THE TWO REGIMES NEED NOT AGREE. †AS PUBLISHED BY THE ORIGINAL AUTHORS ON THEIR OWN SPLIT; OUR LIKE-FOR-LIKE RE-EVALUATION APPEARS IN TABLE I.

Method (real-only)	IoU	Dice	mIoU	Bin. Acc.	Bal. Acc.	Macro-F1
Panta et al. 2023† [2]	0.574	n/a	n/a	n/a	0.855	0.731
Prior weighted-avg ensemble [3]	0.547	0.671	0.697	0.871	0.850	n/a
Best single: Segformer (MiT-B2)	**0.694**	0.801	0.827	0.966	**0.913**	0.890
SandBoilNet (ConvNeXt-S+scSE)	0.676	0.793	0.818	0.966	0.909	0.886
Mean ensemble (five members)	0.681	0.791	0.821	0.967	0.906	0.885
CV stacking (LR meta), ours	0.681	0.790	0.821	**0.967**	0.895	0.885

five calibrated members reaches 0.688 IoU, marginally above both the mean ensemble and the fitted stack (0.681) and still below the best single model. Zero-shot Segment Anything (SAM 2.1 Hiera-L [77]), the foundation model used in our dataset curation, reaches only 0.410 IoU in fully automatic mode on the test images. It becomes competitive only when handed an oracle bounding box, which lifts it to 0.685; an oracle centre point actually drops it below its own automatic score. Both prompts presuppose the very localisation the automatic task must solve. A task-specialised model trained on 199 images is therefore substantially better than zero-shot SAM in the automatic setting deployment requires, and remains ahead (0.707 against 0.685) even when SAM is handed the oracle box that no deployed system could supply.

Table III compares eight meta-learner families fitted on the same calibrated out-of-fold cube, holding the cube and a fixed default operating point constant to isolate the combiner choice. The families all cluster below the best single model: logistic regression leads at 0.681, the linear SVM and the MLP sit just behind it, and the more flexible tree ensembles overfit the out-of-fold cube and fall to 0.64–0.66. Because the combiner family is a secondary factor, we adopt logistic regression for its directly interpretable weights (Table VI); at the OOF-tuned operating point it delivers the headline stacked result, which no alternative combiner improves upon.

TABLE III

META-LEARNER COMPARISON ON THE CALIBRATED OUT-OF-FOLD CUBE OF THE FIVE FAMILY CHAMPIONS, AT A FIXED DEFAULT OPERATING POINT TO ISOLATE THE COMBINER CHOICE (PER-IMAGE TEST IOU). EVERY FITTED COMBINER LANDS BELOW THE BEST SINGLE ARCHITECTURE, AND NONE BEATS UNWEIGHTED MAJORITY VOTING, SO THE FITTING ADDS NOTHING HERE; THE FLEXIBLE TREE ENSEMBLES OVERFIT THE CUBE. RBF AND POLYNOMIAL SVMS ARE FITTED ON A 15K-PIXEL SUBSAMPLE; PER-COMBINER CONFIGURATIONS APPEAR IN APPENDIX A.

Combiner	Test IoU
Best single architecture	**0.694**
Majority vote (unweighted)	0.688
Mean ensemble	0.681
Logistic regression (adopted)	0.681
Linear SVM	0.680
Multi-layer perceptron	0.678
XGBoost	0.659
SVM, RBF kernel	0.646
Gradient boosting (histogram)	0.645
Random forest	0.643
SVM, polynomial kernel	0.640

The cross-validation design itself has only a secondary effect. Table IV varies it: doubling the fold count to K=10 trains each fold model on 90% of the data, and a *nested* cross-validated stack fits the meta-learner on fold-averaged predictions that match the test-time regime (Section III-F). Across all three designs the calibrated stack fails to reach that design's best single model (nested values

TABLE IV

EFFECT OF THE CROSS-VALIDATION DESIGN ON THE FIVE-FAMILY-CHAMPION ENSEMBLE (PER-IMAGE TEST IOU AT THE OOF-TUNED OPERATING POINT). UNDER EVERY DESIGN THE CALIBRATED STACK SITS BELOW THAT DESIGN'S OWN BEST SINGLE MODEL; RAISING THE FOLD COUNT LIFTS THE STACK AND THE BEST SINGLE TOGETHER RATHER THAN CLOSING THE GAP, AND THE NESTED DESIGN NARROWS THE DEFICIT TO 0.002 WITHOUT ELIMINATING IT.

CV design	Best single	Mean ens.	Stacked
5-fold	0.694	0.681	0.681
10-fold	**0.711**	**0.687**	**0.696**
Nested 5×5	0.694	0.681	0.691

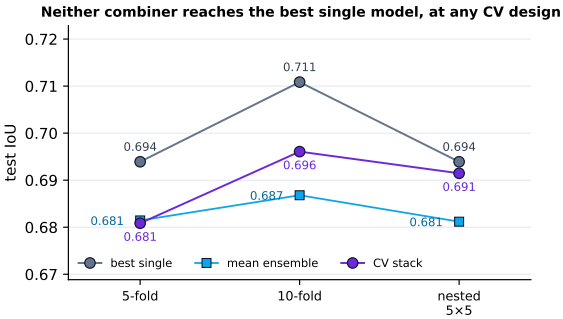

Fig. 10. Ensembling strategies against the best single model at each cross-validation design (values in Table IV). No combiner reaches the best single model at any design, and raising the fold count lifts both together without closing the gap. The nested design comes closest, its meta-learner being the only one fitted on the fold-averaged predictions it meets at test time.

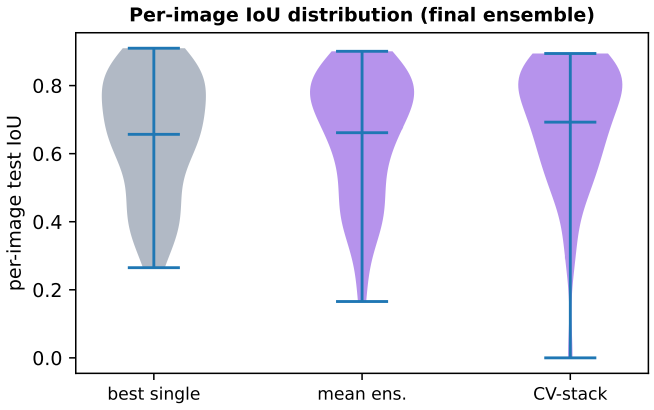

Fig. 11. Per-image IoU distributions for the best single base learner, the mean ensemble, and the CV-stacked meta-learner of the final ensemble. Stacking tightens the spread relative to the weaker single backbones, but its mean IoU remains below the best single model (Table II): it trades a little worst-case variance for a small loss in typical accuracy.

at the default operating point, as the nested procedure tunes no threshold). The nested design narrows the deficit furthest, to 0.002, the expected direction since its meta-learner alone is fitted on the fold-averaged predictions it meets at test time, yet it still does not cross the line. The ranking mismatch is therefore real but not the lever that separates a stack from the best single model (Figure 10).

TABLE V

HARD-NEGATIVE ABLATION ON THE PRINCIPAL RUN (UPDATED SANDBOILNET ON REAL + AUGMENTED SYNTHETIC DATA). THE SECOND ROW ADDITIONALLY TRAINS ON THE CONFUSION-WEIGHTED SYNTHETIC HARD-NEGATIVE POOL OF SECTION III-G. THE INTENDED EFFECT WAS A LOWER FALSE-POSITIVE RATE; INSTEAD THE FP RATE ROSE (0.018 → 0.020) AND IOU FELL (−0.010), SO THE NEGATIVE PHASE IS EXCLUDED FROM THE FINAL MODEL.

Negative phase	IoU	Dice	FP	Recall	Bal. Acc.
None (real + synth)	**0.703**	**0.826**	**0.018**	0.808	**0.895**
+ synthetic, conf-weighted	0.693	0.819	0.020	**0.809**	0.894

C. Hard-negative ablation

Table V reports the effect of the synthetic hard-negative phase. Both rows share an identical principal training run (Updated SandBoilNet on the real training set plus augmented synthetic data under the leak-free filter); the second additionally trains on the confusion-weighted synthetic hard-negative pool of Section III-G. This ablation is a single held-out-split comparison rather than the five-fold protocol, so its absolute baseline (0.703) is not directly comparable to the fold-averaged figures of Table II; the controlled quantity is the within-table difference between the two rows, both run under the identical split.

Contrary to the intent of hard-negative mining, the synthetic negatives made the model fire *more* rather than less: both recall and the false-positive rate increased, and intersection-over-union dropped by 0.023. The confusion weighting of (4) concentrates training on the negatives the baseline is most fooled by, but on this small dataset the all-zero-mask negatives appear to shift the decision threshold toward over-segmentation rather than suppress confusable structures. We therefore exclude the negative phase from the final model and report it as a negative result: useful guidance for future work on synthetic negatives in this domain.

D. Boundary tolerance and bootstrap uncertainty

Two diagnostics put the champion's headline IoU in context. First, the fuzzy boil rim makes a strict per-pixel IoU pessimistic: relaxing the boundary match to a t=2 px tolerance band raises the micro-averaged IoU from 0.671 at t=0 to 0.689, an increase of 0.019 that is attributable entirely to sub-annotation-noise rim jitter rather than to genuine region recovery. The gap is modest, which is itself informative: the residual error is not dominated by a one- or two-pixel boundary offset but by whole regions that are missed or hallucinated, so a tolerance band does not paper over the failure cases.

Both diagnostics are ultimately bounded by the annotations themselves. Where a boil ends is a judgement call: the disturbed sand grades into a damp halo with no crisp edge, so some masks enclose that halo and over-annotate the boil, while others trace only the clearly disturbed core and under-annotate it. IoU penalises both directions equally, so this bidirectional label noise puts a ceiling on the attainable score that no model can cross, and it charges

TABLE VI
LOGISTIC REGRESSION META-LEARNER WEIGHTS AND PER-ARCHITECTURE CALIBRATION TEMPERATURES FOR THE FINAL ENSEMBLE. WEIGHTS ARE NORMALISED SO THAT THEIR ABSOLUTE VALUES SUM TO 1. A TEMPERATURE $T_a > 1$ INDICATES THAT THE ARCHITECTURE WAS OVERCONFIDENT AND WAS SOFTENED; $T_a < 1$ INDICATES THAT IT WAS UNDERCONFIDENT AND WAS SHARPENED.

Base learner	w_a	T_a
SandBoilNet (ConvNeXt-S+scSE)	0.201	1.10
Transformer (SegFormer / MiT-B2)	0.225	1.50
Pyramid (FPN / ConvNeXt-S)	**0.273**	1.70
Nested CNN (U-Net++ / EffNet-B4)	0.152	1.20
Atrous (DeepLabV3+ / EffNet-B4)	0.149	1.10

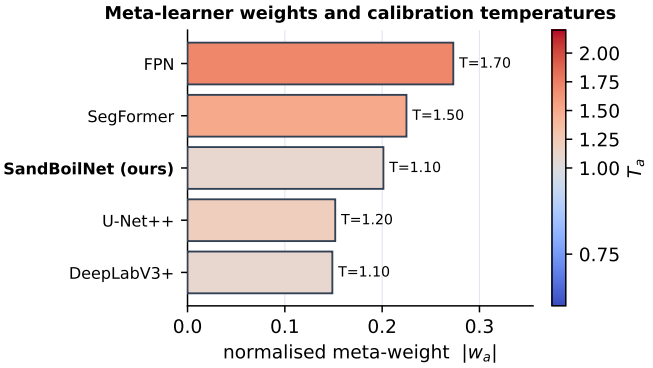

Fig. 12. Meta-learner weights coloured by calibration temperature: each backbone's stacked contribution and how much recalibration it required. Bold labels mark the two SandBoilNet-family backbones we contribute (SandBoilNet and its Texture-SE variant). The strongest fold-averaged backbone, SandBoilNet, receives only the fourth-largest weight: the visible signature of the out-of-fold ranking mismatch analysed in Section VI-A.

a prediction that is arguably correct with an error that belongs to the label. The effect falls on every method compared here alike, so the matched single-split deltas and fold-level variances we rely on are unaffected; it is the absolute IoU values, ours and the published baselines' alike, that should be read with this in mind.

Second, the held-out set is small enough that the point estimate carries real sampling uncertainty. A non-parametric bootstrap over the held-out images places a 95% confidence interval of $[0.637, 0.745]$ around the 0.694 best single model. That interval, roughly ± 0.05 IoU wide, is the quantitative form of the test-set-size caveat we raise throughout, and it comfortably brackets the stacked ensemble and every stacking variant tested. We therefore report the stack's shortfall as consistent in sign but within sampling noise: repeating the whole five-fold protocol under three fold-partition seeds gives a gap of -0.007 ± 0.005, negative in every repetition. We therefore lean on *matched* single-split deltas (Section V-G) and fold-level variance rather than cross-method point comparisons when differences are this small; the paired Wilcoxon signed-rank test of Section V-B ($p = 0.45$, median per-image difference -0.002) makes the same point for the stacked-versus-single comparison directly.

E. Per-architecture meta-learner weight analysis

A useful by-product of the linear meta-learner is that its five weights admit a direct interpretation as the relative contribution of each backbone to the final stacked output. Table VI reports those weights alongside the per-architecture calibration temperatures.

The pairing of w_a and T_a is informative (Figure 12 plots the two together, and Figure 13 shows the reliability curves before and after temperature scaling). A high $|w_a|$ paired with a T_a close to 1 indicates an architecture whose probabilities were already well-calibrated and whose stacked contribution reflects raw accuracy; a high $|w_a|$ paired with a T_a far from 1 indicates an architecture that required substantial recalibration before its contribution became proportionate to its accuracy. Refitting the same meta-learner on the *uncalibrated* cube leaves the stacked test IoU within the bootstrap interval of Section V-D but redistributes the weights: the over-confident pyramid and transformer backbones, whose temperatures are farthest from one, are down-weighted once their probabilities are softened onto the shared mid-range the meta-learner reads. Calibration is therefore not an accuracy lever here, and we do not claim it as one; what it buys is that the weights of Table VI are comparable across backbones and can be read as relative contributions at all. Notably, the strongest *single* backbone, SandBoilNet, draws only a middling weight (0.201): fitted on single-fold out-of-fold predictions, the meta-learner cannot see that this *highest-variance* backbone (fold-to-fold sd 0.026 versus ≤ 0.021 for the others; Table XI) becomes the strongest model once its five folds are averaged, so it leans on the more stable members. That reweighting is not enough to lift the stacked output past the best single model, however: the weights redistribute influence among members whose errors largely coincide, which changes the blend without adding independent evidence.

F. Qualitative segmentation comparison

Figure 14 compares the five base learners' predictions on a fixed set of test images spanning the IoU range from a clean detection to a failure case. The per-architecture differences the stacked meta-learner draws on are visible: individual backbones fragment the sand-boil region or miss its outer rim in low-contrast or partially occluded scenes, and they disagree most on the harder rows, which is the error independence a calibrated combiner exploits. The bottom row is the instructive limit: every backbone fires on a boil-like wet patch (red) while missing the true boil (blue), a correlated zero-overlap error that no combiner can repair, and exactly the puddle-type confusable the hard-negative phase of Section III-G targets.

G. Ablations: skip attention, data, curation, resolution

Table VII isolates the design levers on the champion backbone (SandBoilNet, ConvNeXt-S+scSE) under a single 85/15 split, each averaged over three seeds. None of the

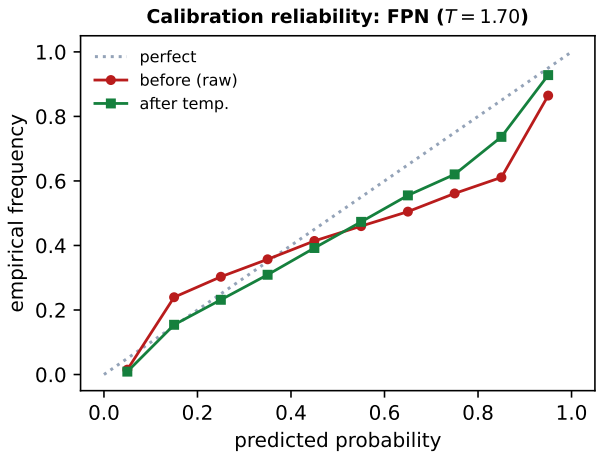

Fig. 13. Calibration reliability of the most over-confident architecture before versus after per-architecture temperature scaling: the raw curve bows below the diagonal (over-confident), and the scaled curve sits close to it, so the calibrated probabilities are directly combinable. This is the step that keeps the meta-learner weights of Table VI proportionate to accuracy rather than to confidence.

alternatives improves on the plain scSE gate. The texture-sensitive variant (Texture-SE, a second-order standard-deviation channel gate) is in fact the weakest, 0.016 below the control, and the contrast-gated TC-SE and multi-scale Atrous-SE are indistinguishable from it. A four-fold increase in input resolution to 1024 px also hurts. Attention machinery beyond the standard scSE gate therefore buys nothing reliable here.

Synthetic training data affects the champion unevenly across generators. Evaluating each pool separately (Table VIII) spans a 0.046 swing, from the copy-paste baseline at the top to the V3 preset slightly below the real-only control. V3 and V4 share a conditioning stack and differ only in V4's soft-mask inpainting, which preserves the real boil pixels. Because identically configured controls vary by up to 0.026 across runs on this test set, we read the table as indicative only, and replicated the two conditions that matter most. Under three-seed replication the copy-paste pool is neutral (-0.000), while the quality-filtered V_2 pool (only pairs on which all five folds of a held-out segmenter reach 0.70 IoU against the pair's own mask) reaches 0.718 ± 0.012 against the 0.707 control. That gain is the largest we observe from any intervention, but its spread is as wide as its mean and one of the three runs falls below the control, so we report it as suggestive rather than established. A wider sweep isolating the generator's guidance scale, over four values with five seeds each on *unfiltered* pools of the same size, separates the two factors: guidance scale is immaterial (all settings within 0.002 of one another), and without the label-fidelity filter the mean gain over nineteen runs falls to 0.001. What little synthetic augmentation buys the strongest model therefore comes from filtering the pool for label fidelity, not from the sampling configuration that produced it. Whether or not a given pool helps, the leak-free filter remains a precondition for reporting the result at all.

TABLE VII

Skip-attention design-lever ablation on the champion backbone (SandBoilNet, ConvNeXt-S+scSE), single 85/15 split, mean ± standard deviation over three seeds ($\{1, 2, 42\}$); Δ is relative to the scSE control. No variant improves on the plain scSE gate, and the spread between best and worst is 0.017, comparable to the seed noise of the weakest entry. On an earlier, harder split the texture-sensitive gate led by 0.024, an advantage that does not survive here.

Skip-attention variant	Test IoU (mean ± sd)	Δ
scSE control	$\mathbf{0.707 \pm 0.003}$	–
TC-SE (texture + contrast)	0.705 ± 0.007	-0.001
Atrous-SE (multi-scale spatial)	0.701 ± 0.007	-0.005
Texture-SE (std channel gate)	0.690 ± 0.012	-0.016

TABLE VIII

Per-generator synthetic-pool ablation on the champion, single held-out split under the leak-free per-fold filter; Δ is relative to the matched real-only control of the same run (0.678). The V presets differ in their multi-ControlNet conditioning; MaskCN is the mask-conditioned generator of Section III-C, evaluated on a leak-safe pool (Appendix B); copy-paste is the compositing baseline of Appendix C. Identically configured controls vary by up to 0.026 on this test set, so individual deltas are indicative rather than exact.

Generator	ControlNet conditioning	IoU	Δ
Real only (control)	–	0.678	–
Copy-paste	compositing baseline	**0.718**	**+0.040**
MaskCN	mask-conditioned ControlNet	0.716	+0.038
Merged pool	all presets combined	0.706	+0.028
V2	Canny + Normal + HED	0.698	+0.020
V4	as V_3, + soft-mask inpaint	0.693	+0.015
V1	Canny + Depth + HED	0.680	+0.002
V3	HED + Normal + IPA	0.672	−0.006

TABLE IX

Cross-dataset check: the five base learners retrained on DeepCrack (300/237), mean ± sd of test IoU over seeds $\{1, 2, 42\}$; same pipeline as the sand-boil experiments, real data only.

Base learner	Test IoU (mean ± sd)
Updated SandBoilNet (ConvNeXt-S+scSE)	$\mathbf{0.737 \pm 0.006}$
U-Net++ / EffNet-B4	0.733 ± 0.014
FPN / ConvNeXt-S	0.726 ± 0.003
DeepLabV3+ / EffNet-B4	0.711 ± 0.012
SegFormer / MiT-B2	0.707 ± 0.009

H. Cross-dataset check on DeepCrack

To verify that the architecture ranking is not an artefact of one dataset, we retrain all five base learners from scratch on DeepCrack [28] (300 train / 237 test crack images), using the identical pipeline, augmentation and per-architecture schedules, with no synthetic data and no stacking. Over the same three seeds the Updated SandBoilNet again leads (Table IX), indicating that the modernised ConvNeXt-S+scSE configuration transfers beyond sand boils to a second scarce-data infrastructure-defect task.

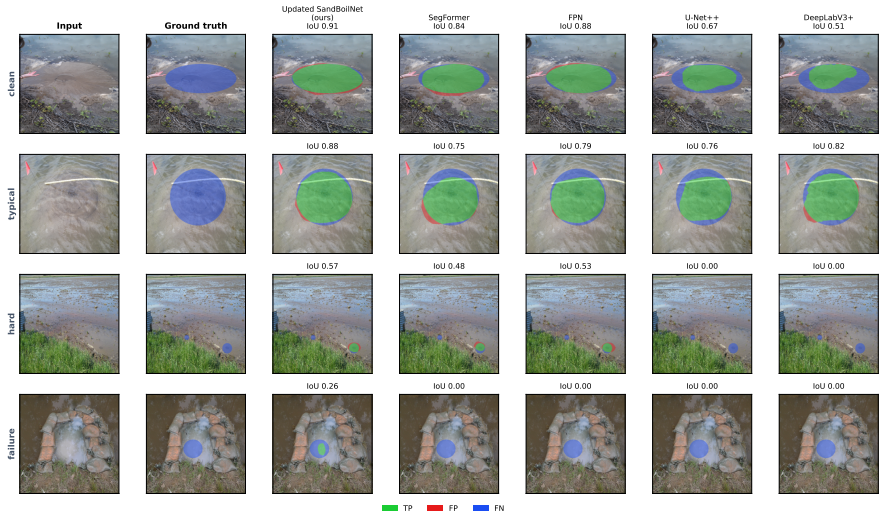

Fig. 14. Qualitative comparison on held-out real test images, drawn from those the proposed model leads and spanning its IoU range from a clean detection (top) to a near-total failure (bottom); full-test-set means for every architecture are in Table II. Each model cell overlays true positives (green), false positives (red), and false negatives (blue), titled with that image's IoU; the Input and Ground-truth columns carry no prediction overlay, so blue there is the annotated mask itself. In the bottom row every model segments a boil-like wet patch while missing the true boil, the puddle-type confusable targeted by the hard-negative phase (Section III-G).

I. Computational cost

The full five-fold, five-architecture training completes in ≈1.7 h on a single NVIDIA H100 NVL, and stacked inference is sub-second per image; a per-phase breakdown is given in Appendix D (Table XII). The upstream generation costs are reported in the companion paper [5].

VI. Discussion

Section V supports a compact reading. The strongest single backbone reaches 0.707 IoU with an scSE skip-attention gate, and the texture-sensitive variant we expected to lead trails it by 0.016. A calibrated stack over five architectures reaches 0.681 without improving on the best single model, and the synthetic hard-negative phase costs a further 0.010 IoU. We examine each of these below, since a protocol built so that such measurements can be believed is the central contribution.

A. Cross-validated stacking: a trustworthy protocol without an ensemble gain

The weighted-average ensemble of our previous work [3] fits its weights on the same split later used to report accuracy, a mild but real leakage. Cross-validated stacking removes it, because every out-of-fold prediction comes from a model that did not observe the example and the meta-learner is fitted on those predictions alone, leaving both the estimate and the weights unbiased. That guarantee holds independently of the size of any gain.

Under this measurement the ensemble does not win. The calibrated stack reaches 0.681 against 0.694 for the best single network, and a plain mean ensemble matches it. The outcome is not an artefact of the combiner, since eight meta-learners spanning linear, kernel, neural, and tree families (Table III) land at or below the best single model. Nor is it an artefact of the protocol, since the leakage audit confirms that the meta-learner observes out-of-fold predictions exclusively.

Member correlation explains the result. The members' per-pixel error maps correlate at 0.894 averaged over all pairs, and the tightest pairs, U-Net with FPN (0.931) and U-Net++ with DeepLabV3+ (0.929), are those that share an encoder. Correlation therefore tracks the encoder rather than the decoder: five decoders rest on only three feature extractors, so the ensemble carries less independence than its member count suggests, and five strong backbones trained on the same 199 images from a common initialisation fail on the same boundaries.

The information is nonetheless present, since a per-pixel oracle selecting the best member at each location reaches 0.837 IoU, so the ceiling is set by what the combiner can reach rather than by what the members encode. Replacing the FPN and DeepLabV3+ encoders with ResNet50 and ResNet101, giving five distinct feature extractors, narrows the shortfall from -0.013 to -0.008 without reversing it (0.684 against 0.693). Decorrelation thus recovers about a third of the deficit, the remainder being imposed by the shared training images rather than by the choice of backbones. Once one architecture is clearly strongest, a combiner obliged to weight all five is handicapped against selecting it. Two secondary effects deserve record. The out-of-fold ranking mismatch, whereby the meta-learner observes single-fold predictions but is applied to fold averages at test time, accounts for most of the measured shortfall: nested cross-validation lifts the stacked score from 0.681 to 0.691, narrowing the deficit to 0.002. That shift is smaller than the 0.026 run-to-run spread of this test set, so we do not present it as established, and the nested stack still does not exceed its best member. Calibration and the operating point likewise matter more than the combiner family.

B. Leak-free provenance and calibration

The filter of Section III-F1 reduces to one set operation per fold, yet its consequences for validity are out of proportion to its size. Without it a synthetic image generated from a real parent can train a model whose validation fold contains that parent, and even a visually distinct render carries the parent's geometry, lighting, and texture. The `source_image_id` field written by the upstream pipeline [5] closes this path, and we recommend it for any synthetic pipeline evaluated under cross-validation. MaskCN (Section III-C) offers the complementary escape, since a record synthesised into a target mask has no parent to leak and renders Eq. (1) vacuous. Per-architecture temperature calibration contributes one scalar per backbone but is not marginal: without it the most overconfident architecture dominates the meta-learner regardless of its accuracy, and with it the fitted weights become interpretable as a record of each architecture's contribution.

C. Confusion-score mining and why it did not help here

The classical recipe [57] mines high-loss examples from an unlabelled real pool, whereas we mine high-confusion examples from a controllable synthetic one, weighting each by Eq. (4) so that the negative-phase gradient concentrates on the examples that most mislead the baseline. That mechanism did not help on this dataset. The negative phase lowered test IoU from 0.703 to 0.693 and raised the false-positive rate, while leaving recall essentially unchanged (Table V), so it bought no recall and cost both precision and accuracy. The most plausible cause is open-loop scoring, since the negatives are scored once by the public baseline and never re-scored against the model being trained, leaving the pool unadapted to the residual errors of the modernised backbone. A closed-loop variant is the natural next step, but on the present evidence we do not recommend the open-loop phase as a default.

D. Limitations, threats to validity, and ethical considerations

Several limitations qualify these conclusions. The held-out test set is small in absolute terms, so single-digit differences are not always distinguishable from variation across splits and a paired comparison is unlikely to resolve improvements smaller than a few IoU points; we mitigate this by averaging fold predictions and fixing the partition and seed cascade. Predictions still degrade where a boil is obscured by reflective standing water, which breaks the texture cues the network relies on, and a two-pass approach that first predicts reflection regions is left to future work. Sand boils also co-occur with seepage or cracks within one frame, which the per-class models used here do not capture, so a shared multi-class head is a natural extension. The confusion score depends on the publicly released baseline, so a category on which that checkpoint silently fails is scored as easy when it is in fact hard. Finally, the synthetic data derives from a single instance of the upstream pipeline [5], so gains observed under combined real and synthetic training may partly reflect its stylistic biases; an independent generator on a disjoint reference set would constitute a stronger external control.

The pipeline is intended to support trained inspectors rather than replace them. Synthetic data introduces a risk of overconfidence, since high in-distribution accuracy need not transfer to field conditions absent from the reference set, and the protocol reports fold-level variance so that deployment can expose that uncertainty rather than collapse it to a binary decision. Because a false negative during a high-water event can delay intervention long enough to permit a breach, deployment should err toward false positives verified by a human inspector.

VII. Conclusion and Future Work

We have presented a segmentation methodology for pixel-level sand boil detection that combines five encoder–decoder backbones through a calibrated cross-validated stacking ensemble and strengthens its rejection of visually confusable non-defects through a confusion-score-mined synthetic hard-negative phase. The framework closes two leakage paths the defect-segmentation literature has not consistently addressed: the meta-learner is fitted on out-of-fold predictions alone, so its weights are unbiased, and every synthetic record carries a `source_image_id` pointer to its real parent, so a per-fold filter keeps synthetic descendants of held-out images out of the corresponding training folds. Per-architecture temperature calibration makes the backbones comparable before stacking and leaves the linear meta-learner's weights readable as each backbone's relative contribution.

On a held-out real sand-boil test set the Updated SandBoilNet reaches 0.707 intersection-over-union with an scSE skip-attention gate. A calibrated stack over five architectures reaches 0.681 and does not exceed the best single model, an outcome eight meta-learner families reproduce, so the shortfall is a property of the members rather than of the combiner. A controlled ablation clarifies what does and does not drive accuracy here. Five strong backbones trained on the same scarce data prove too correlated for a combiner to exploit, and the texture-sensitive gate that led on our earlier split is now the weakest of four variants. Synthetic augmentation repays the strongest model when the pool is filtered for label fidelity, though not through the sampling configuration that generated it, while neither the hard-negative phase nor a four-fold resolution increase improves on it. Establishing which additions transfer to a scarce-data, fuzzy-boundary defect and which do not is a deliberate complement to the positive gains. The synthetic data is produced by the pipeline of our companion paper [5] and consumed here as an upstream module.

Several extensions remain open, each developed in Section VI. The hard-negative loop is currently one-directional, and a closed-loop variant would re-score with the current stacked ensemble after each round, while the symmetric version on the positive pool would steer synthesis toward the conditions the model still fails on,

making the generator an active learner rather than a fixed data source. MaskCN (Section III-C) is the most immediate lever, supplying registered labels at zero annotation cost with no parent-image filter required, though folding curated MaskCN singles into the augmentation mixture at a fidelity that helps the champion remains open. Since our comparison isolates the encoder as the dominant factor, transformer-based dense-prediction decoders and depth-aware architectures are a natural next step for a boundary this diffuse. Beyond that, a shared multi-class head would capture sand boils co-occurring with seepage or cracks, and a public levee-defect benchmark with paired real and synthetic data, source tags, and a held-out split would let the community compare methods directly; that dataset and the deployed inspection workflow are the subject of the third paper in this series.

Code and Data Availability

The complete segmentation and stacking framework is released alongside this paper, comprising the per-architecture training driver, the cross-validated stacking driver, and the ablation, baseline, and hard-negative scripts that produce every table in Section V. The artefact also includes the twenty-five fold checkpoints (five architectures × five folds), the out-of-fold probability cubes, the fitted temperature scalars and meta-learner weights of Table VI, and three manifests that make the evaluation auditable rather than merely repeatable: the per-fold `source_image_id` admission list, which records exactly which synthetic records entered each fold's training pool and so allows the leak-free filter to be verified independently; the confusion-scored hard-negative pool manifest; and the mask-source manifest of the leak-safe MaskCN evaluation pool (Appendix B). With the seeds recorded in Section IV-F, every numerical entry in Section V rederives from a single command. On acceptance the artefact will be published at https://github.com/padam56/sandboil-leakfree-stacking and archived under a versioned Zenodo DOI.

The upstream generative components are released with the companion generation paper [5], and the Convex Hull Annotator used in curation is distributed as a standalone, class-agnostic library [62]. The real imagery derives from the U.S. Army Corps of Engineers levee-inspection archive [58], whose terms of use preclude redistribution of the underlying photographs; the pixel-level binary masks produced by the authors will be released under CC BY 4.0 alongside the code, on the same terms as the group's earlier public levee-defect releases [60], [61].

Conflicts of Interest

The authors declare no competing financial or non-financial interests. This work was carried out in an academic research setting and is derived from the corresponding author's Master's thesis [3]. All authors have read and approved the final manuscript and agree to its submission.

Acknowledgments

This research was supported by the U.S. Army Corps of Engineers (USACE), under Contract No. W912HZ-23-2-0004. The authors thank the USACE for making the levee-inspection imagery archive [58] available for academic research and acknowledge Stability AI, the Diffusers library, the ControlNet authors, and the IP-Adapter authors for their open-source models and software. This work was conducted on the institutional GPU cluster at LSU New Orleans. The views and conclusions expressed in this paper are those of the authors and do not necessarily reflect the official policies or positions of the USACE.

Appendix A
Meta-learner configurations and the combiner-by-design grid

Table III in the main text reports the eight per-pixel meta-learners under the primary five-fold design. This appendix records the exact configurations and the full combiner-by-design grid, so that every number is reproducible from the released sweep script with fixed seeds.

All combiners consume the same input: the per-pixel five-dimensional vector of temperature-calibrated architecture probabilities (Section III-F). Each is fitted on a fixed-seed random pixel subsample of the out-of-fold cube: 2M pixels for logistic regression (`max_iter`= 500, C=1), the linear SVM (C=1), histogram gradient boosting (library defaults), and XGBoost (300 trees, histogram method); 1M pixels for the MLP (two hidden layers, 32 and 16 units, early stopping) and the random forest (300 trees); and 15k pixels for the RBF and polynomial (degree-3) SVMs, whose kernel fits do not scale further. Margin classifiers are thresholded at a decision score of 0, probabilistic ones at 0.5. Test-time inputs are the calibrated fold-averaged test probabilities of the corresponding design.

TABLE X
Combiner-by-design grid: per-image test IoU of the eight meta-learners under the 5-fold and 10-fold designs on the five-family-champion cube at the default operating point, with each design's best-single and mean-ensemble anchors. The 5-fold column reproduces Table III for comparison with K=10. No combiner under either design exceeds the best single model on this base cube.

Combiner	5-fold	10-fold
Best single architecture	**0.694**	**0.711**
Mean ensemble	0.681	0.687
Logistic regression (adopted)	0.681	0.696
Linear SVM	0.680	0.694
Multi-layer perceptron	0.678	0.681
Gradient boosting (histogram)	0.645	0.687
SVM, RBF kernel	0.646	0.666
XGBoost	0.659	0.657
SVM, polynomial kernel	0.640	0.649
Random forest	0.643	0.633

Two readings of Table X matter for the main text. First, the ordering shifts systematically with the design: the combiners that overfit the single-fold out-of-fold signal

at K=5 (the RBF SVM, gradient boosting) gain IoU at K=10, while the linear combiners barely move, direct evidence for the ranking-mismatch effect of Section V-B. Second, on this base cube at the default operating point no combiner under either design reaches the best single model; the main text's finding (Table II) is that no combiner reaches the best single model, and this grid shows that conclusion is driven neither by the combiner family nor by the cross-validation design. Adopting a flexible combiner on the strength of its test score would violate the selection discipline this paper argues for, which is why we fix logistic regression for its interpretable weights.

TABLE XI
PER-ARCHITECTURE FOLD-LEVEL VARIANCE FOR THE FIVE FAMILY CHAMPIONS OF TABLE I: HELD-OUT TEST IOU OF EACH INDIVIDUAL FOLD MODEL, BEFORE WITHIN-ARCHITECTURE AVERAGING. SANDBOILNET IS THE HIGHEST-VARIANCE BACKBONE, WHICH IS WHY THE META-LEARNER (TABLE VI), FITTED ON SINGLE-FOLD OUT-OF-FOLD PREDICTIONS, GIVES IT A MODEST WEIGHT.

Base learner	Fold IoU (mean ± sd)	Range
SandBoilNet	0.630 ± 0.026	0.600–0.672
FPN	0.633 ± 0.017	0.612–0.663
SegFormer	0.629 ± 0.021	0.608–0.660
U-Net++	0.634 ± 0.020	0.610–0.670
DeepLabV3+	0.620 ± 0.015	0.604–0.647

APPENDIX B
LEAK-SAFE CONSTRUCTION OF THE MASKCN EVALUATION POOL

The MaskCN generator of Section III-C renders a fresh sand boil into a conditioning mask drawn from a bank of 41 real annotation masks. Although MaskCN records are parentless in image space, the conditioning mask itself carries the label geometry of its source image: a sample conditioned on the mask of a *held-out* image would place that image's exact ground-truth shape in the training set. Eight of the 41 bank masks derive from test-set images: img (9), (14), (17), (35), (92), (119), (172), and (176). An unfiltered pool therefore leaks test-label geometry.

The evaluation pool of Table VIII therefore keeps only samples conditioned on the 33 training-set mask sources: of the 585 paired candidates, 111 test-source samples are excluded and 474 are retained, each carrying its mask-source identifier in the pool manifest. Training then follows the identical per-generator protocol (champion backbone, seed 42, matched single-split control). No result from any unfiltered MaskCN pool is reported anywhere in this paper.

APPENDIX C
COPY-PASTE AUGMENTATION BASELINE

As a simple augmentation reference we also trained the champion backbone with a copy-paste pool [47], in which real boil instances, cut out with their masks, are pasted onto other training images. Under the same three-seed protocol it reaches 0.706 IoU, on par with the real-only champion (0.707), consistent with the known strength of copy-paste for instance segmentation. We report it for completeness; the paper's focus is generative augmentation for its controllability and procedural scene diversity, which copy-paste cannot provide.

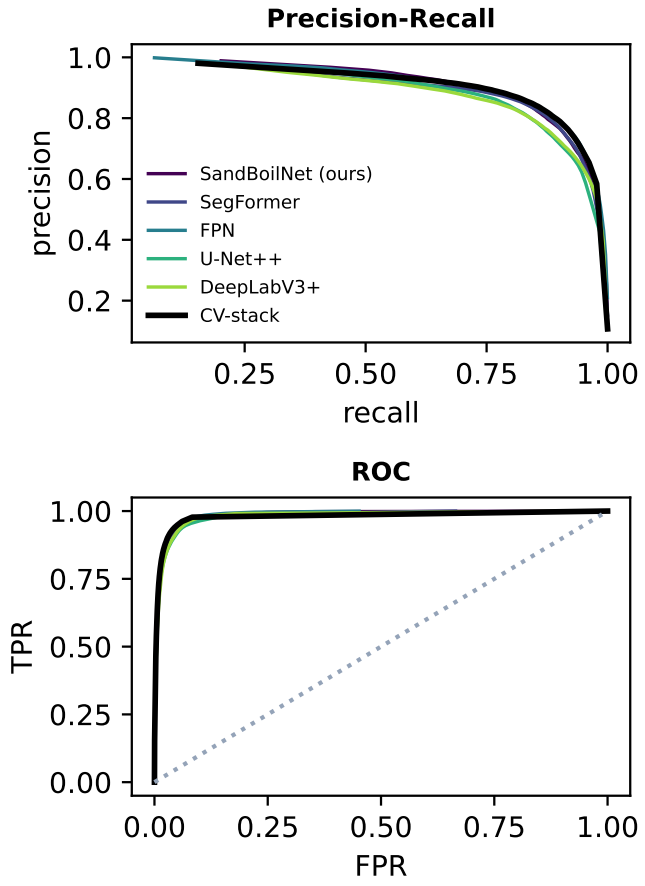

Fig. 15. Precision–recall (top) and ROC (bottom) curves across the five base learners and the stacked ensemble. The stacked operating curve tracks the best base learners across the whole sweep, so its behaviour does not rest on a single threshold choice. The curves lie close together because the five learners are the family champions of Section V-A: average precision spans only 0.835–0.876 and ROC AUC 0.967–0.981, with the stack among the strongest members rather than above them.

APPENDIX D
COMPUTATIONAL COST

Table XII breaks the wall-clock cost of the segmentation pipeline down by phase on a single H100 NVL, complementing the summary in Section V-I: the full five-fold, five-architecture training completes in about 2 h and stacked inference is sub-second per image.

APPENDIX E
PER-ARCHITECTURE TRAINING CONFIGURATION

Table XIII records the exact optimiser and schedule settings behind the architecture search and the cross-validation runs, so every training reproduces from the released configuration. All five base learners share the combined BCE-plus-Dice objective, the AdamW optimiser, a

Fig. 16. MaskCN as a diversity engine. Eighteen distinct, fully labelled sand boils synthesised from a single mask bank, ordered small-to-large in mask coverage and varied in scene, framing, and lighting by per-source seed variation. Because the conditioning mask supplies the label for every tile, the pool extends to arbitrary size at no annotation cost.

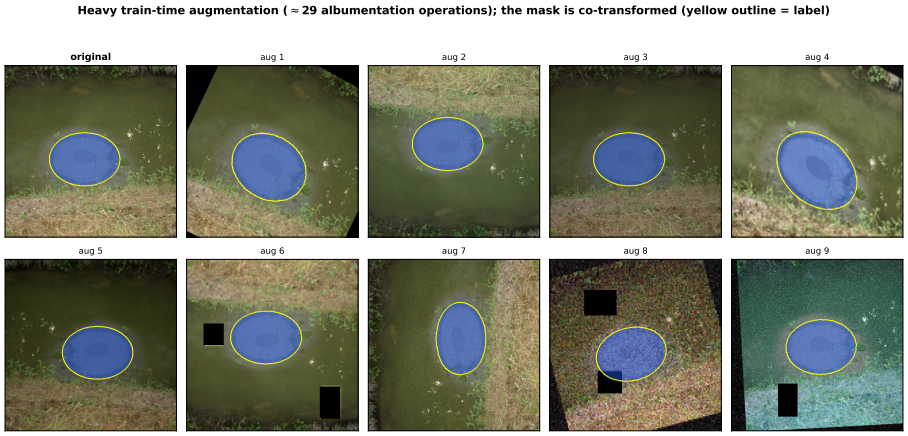

Fig. 17. The heavy train-time augmentation pipeline (the ≈29 Albumentations operations of Table XIV) applied to one training image, with the mask co-transformed (yellow outline marks the propagated label). Panels *aug 1*–*aug 9* are nine independent draws, each a random subset of the operations rather than any single named transform, selected from a fixed seed range so that the nine between them exercise all twenty-nine operations. Validation and test images are never augmented.

TABLE XII
WALL-CLOCK COST OF THE SEGMENTATION PIPELINE ON A SINGLE NVIDIA H100 NVL GPU. THE UPSTREAM GENERATION COSTS (DREAMBOOTH FINE-TUNING, AUGMENTED DATASET SYNTHESIS, HARD-NEGATIVE POOL GENERATION) ARE REPORTED IN THE COMPANION PAPER [5].

Phase	H100
Single-architecture training, principal phase	≈9 min
Five-fold CV, 5 architectures (25 trainings)	≈2 h
Out-of-fold prediction collection	<1 min
Temperature calibration (per architecture)	<1 s
Meta-learner training (logistic regression)	<10 s
Stacked inference, per image	<1 s

batch size of 8 at 512 × 512, automatic mixed-precision training, a five-epoch linear warmup into cosine annealing, and early stopping on validation IoU (patience 30, with the best-IoU checkpoint restored for testing). They differ only in the locked encoder–decoder pairing and in the architecture-appropriate learning rate and weight decay: the transformer takes the smallest learning rate and the heaviest weight decay to stabilise its global attention, while the two ConvNeXt-backed models sit between the transformer and the EfficientNet CNN baselines.

TABLE XIII
PER-ARCHITECTURE TRAINING CONFIGURATION. ALL MODELS SHARE THE BCE-PLUS-DICE LOSS, ADAMW, BATCH SIZE 8, 512 × 512 INPUT, MIXED-PRECISION, A FIVE-EPOCH WARMUP INTO COSINE ANNEALING, AND EARLY STOPPING (PATIENCE 30); THEY DIFFER ONLY IN THE ENCODER–DECODER PAIRING AND THE ARCHITECTURE-APPROPRIATE LEARNING RATE AND WEIGHT DECAY.

Base learner (decoder)	Encoder	LR	Wt. decay
SandBoilNet (U-Net+scSE)	ConvNeXt-S	8e−5	1e−4
SegFormer	MiT-B2	6e−5	1e−2
FPN	ConvNeXt-S	8e−5	1e−4
U-Net++	EfficientNet-B4	1e−4	1e−4
DeepLabV3+	EfficientNet-B4	1e−4	1e−4

Appendix F
Augmentation pipeline

The heavy train-time pipeline of Section IV-B (illustrated in Figure 17) applies the twenty-nine Albumentations operations of Table XIV. The geometric transforms are applied jointly to image and mask so the propagated label stays pixel-aligned through every warp; the photometric, noise, blur, and codec transforms act on the image alone. Operations that would otherwise compound destructively are wrapped in mutually exclusive `OneOf` blocks, so that, for example, at most one of the four blur operators fires on any given sample. Validation and test images receive only the resize-and-normalise tail, never augmentation.

TABLE XIV
The twenty-nine train-time Albumentations operations (Section IV-B), grouped by stage, with the per-operation or per-block application probability p. A bracketed group is a `OneOf` block: p selects the block, then one member fires uniformly at random.

Stage	Operation	p
Geometric	horizontal flip	0.50
	vertical flip	0.50
	random 90° rotate	0.20
	transpose	0.15
	rotate (±35°)	0.35
	affine (scale/translate/rotate)	0.30
	perspective	0.15
	{elastic, grid, optical distortion}	0.25
Photometric	brightness/contrast	0.35
	{colour-jitter, HSV, RGB-shift, gamma}	0.35
	channel shuffle	0.10
	CLAHE	0.20
Noise / dropout	{Gaussian, ISO, multiplicative}	0.20
	pixel dropout	0.10
	coarse dropout	0.12
Blur / codec	{blur, Gaussian, median, motion}	0.18
	sharpen	0.12
	JPEG compression	0.12
	downscale	0.10

Appendix G
Applying the protocol: a checklist

The protocol of Section III-F1 is a discipline rather than an architecture, so it transfers to any pipeline that mixes generated and real images under cross-validation. Three requirements carry it.

1) Record provenance at generation time. Every synthetic record must carry the identifier of its real parent (`source_image_id`) at the moment it is written to disk. Parentage cannot be recovered afterwards, because a generator conditioned on image x can emit a scene resembling image y more closely than x. Records without a parent are safe only when they have none by construction, as in MaskCN (Section III-C).

2) Filter per fold, never once globally. The exclusion in Eq. (1) is evaluated against the current fold's held-out parents and therefore yields a different synthetic subset for each of the K folds. Filtering once against a fixed validation split and reusing the survivors reintroduces the leak for $K-1$ folds, and does so silently, since the run completes and only the reported accuracy is wrong.

3) Fit every tuned quantity on out-of-fold predictions alone. The per-architecture temperature of Eq. (2) and the decision threshold τ are fitted quantities no less than the meta-learner weights, and fitting either on the test split restores exactly the bias the design removes.

Two assertions in the fold-construction driver catch most violations: that each fold's training pool never intersects the held-out parents or their descendants, and that the out-of-fold cube holds one prediction per real image per architecture, each from the fold model that did not train on it. An off-by-one in the fold index is invisible in aggregate accuracy yet corrupts every weight fitted downstream. Finally, the protocol makes a reported gain trustworthy without creating one: here it returned no gain at all, because the members were too correlated to combine (Section VI).

References

[1] R. Seed, R. Bea, A. Athanasopoulos-Zekkos *et al.*, "Investigation of the performance of the new orleans flood protection systems in hurricane katrina on august 29, 2005," Independent Levee Investigation Team, University of California Berkeley, Tech. Rep., 2006.

[2] M. Panta, M. T. Hoque, K. N. Niles, J. Tom, M. Abdelguerfi, and M. Flanagin, "Deep learning approach for accurate segmentation of sand boils in levee systems," *IEEE Access*, vol. 11, pp. 126 263–126 282, 2023.

[3] P. J. Thapa, "Toward robust semantic segmentation in levee infrastructure monitoring: Enhancing accuracy with high-fidelity synthetic data and ensemble learning," Master's Thesis, University of New Orleans, New Orleans, LA, USA, 2025, university of New Orleans Theses and Dissertations No. 3231. https://scholarworks.uno.edu/td/3231/.

[4] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[5] P. J. Thapa, A. B. Naeem, A. Dey, A. Katwal, and M. T. Hoque, "Multi-conditioned diffusion synthesis of sand boils for low-resource earthen-levee inspection," *arXiv preprint arXiv:2607.08794*, 2026.

[6] S. Guan, H. Liu, H. R. Pourreza, and H. Mahyar, "Deep learning approaches in pavement distress identification: A review," *arXiv preprint arXiv:2308.00828*, 2023.

[7] M. Panta, M. T. Hoque, M. Abdelguerfi, and M. C. Flanagin, "IterLUNet: Deep learning architecture for pixel-wise crack detection in levee systems," *IEEE Access*, vol. 11, pp. 12 249–12 262, 2023.

[8] M. Panta, P. J. Thapa, M. T. Hoque, K. N. Niles, S. Sloan, M. Flanagin, K. Pathak, and M. Abdelguerfi, "Application of deep learning for segmenting seepages in levee systems," *Remote Sensing*, vol. 16, no. 13, p. 2441, 2024.

[9] M. Panta, P. J. Thapa, M. T. Hoque, K. N. Niles, S. Sloan, M. Flanagin, K. Pathak, and M. Abdelguerfi, "Application of deep learning for segmenting seepages in levee systems," Engineer Research and Development Center (US), Vicksburg, MS, Miscellaneous Paper ERDC MP-24-8, Nov. 2024. [Online]. Available: https://hdl.handle.net/11681/49453

[10] R. Alshawi, M. M. Ferdaus, M. Abdelguerfi, K. Niles, K. Pathak, and S. Sloan, "Imbalance-aware culvert-sewer defect segmentation using an enhanced feature pyramid network," *arXiv preprint arXiv:2408.10181*, 2024. [Online]. Available: https://arxiv.org/abs/2408.10181

[11] M. Elsheref, P. J. Thapa, A. Katwal, A. B. Naeem, M. A. Tarr, and M. T. Hoque, "OilSpillNet: A deep learning framework for accurate detection and segmentation of marine oil spills

from imagery," *Journal of Environmental Chemical Engineering*, vol. 14, no. 2, p. 121714, Apr. 2026.

[12] B. Chen, Q. Duan, and L. Luo, "From manual to uav-based inspection: Efficient detection of levee seepage hazards driven by thermal infrared image and deep learning," *International Journal of Disaster Risk Reduction*, vol. 114, p. 104982, 2024. [Online]. Available: https://doi.org/10.1016/j.ijdrr.2024.104982

[13] Q. Duan, B. Chen, and L. Luo, "Rapid and automatic uav detection of river embankment piping," *Water Resources Research*, vol. 61, no. 2, 2025. [Online]. Available: https://doi.org/10.1029/2024WR038931

[14] Z. Wang, R. Li *et al.*, "Automated detection of embankment piping and leakage hazards using uav visible light imagery: A frequency-enhanced deep learning approach," *Remote Sensing*, vol. 17, no. 21, p. 3602, 2025. [Online]. Available: https://doi.org/10.3390/rs17213602

[15] B. Wu, B. Chen, X. Jiang, and Z. Liu, "Pruned u-net with multi-scale feature fusion and attention for real-time uav remote sensing of levee defects," *Scientific Reports*, vol. 15, p. 42354, 2025. [Online]. Available: https://doi.org/10.1038/s41598-025-26431-0

[16] A. Kuchi, M. T. Hoque, M. Abdelguerfi, and M. C. Flanagin, "Machine learning applications in detecting sand boils from images," *Array*, vol. 3, p. 100012, 2019.

[17] R. Alshawi, M. T. Hoque, and M. C. Flanagin, "A depth-wise separable U-Net architecture with multiscale filters to detect sinkholes," *Remote Sensing*, vol. 15, no. 5, p. 1384, 2023.

[18] J. V. Aanstoos, K. Hasan, C. G. O'Hara, S. Prasad, L. Dabbiru, M. Mahrooghy, B. Gokaraju, and R. Nobrega, "Earthen levee monitoring with synthetic aperture radar," in *2011 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2011.

[19] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea *et al.*, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[23] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[25] N. Ibtehaz and M. S. Rahman, "MultiResUNet: Rethinking the u-net architecture for multimodal biomedical image segmentation," *Neural Networks*, vol. 121, pp. 74–87, 2020.

[26] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis (DLMIA)*, 2018.

[27] O. Oktay, J. Schlemper, L. L. Folgoc *et al.*, "Attention U-Net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[28] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "Deepcrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.01.036

[29] H. Liu, J. Yang, X. Miao, C. Mertz, and H. Kong, "Crackformer network for pavement crack segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2023. [Online]. Available: https://doi.org/10.1109/TITS.2023.3266776

[30] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [Online]. Available: https://arxiv.org/abs/2112.01527

[31] K. Ge, C. Wang, Y. Guo, Y. Tang, Z. Hu, and H. Chen, "Fine-tuning vision foundation model for crack segmentation in civil infrastructures," *arXiv preprint arXiv:2312.04233*, 2023. [Online]. Available: https://arxiv.org/abs/2312.04233

[32] M. Ahmadi, A. G. Lonbar, H. K. Naeini, A. T. Beris, M. Nouri, A. S. Javidi, and A. Sharifi, "Application of segment anything model for civil infrastructure defect assessment," *arXiv preprint arXiv:2304.12600*, 2023. [Online]. Available: https://arxiv.org/abs/2304.12600

[33] S. Kulkarni, S. Singh, D. Balakrishnan, S. Sharma, S. Devunuri, and S. C. R. Korlapati, "Crackseg9k: A collection and benchmark for crack segmentation datasets and frameworks," in *European Conference on Computer Vision (ECCV) Workshops*, ser. Lecture Notes in Computer Science, vol. 13807, 2022. [Online]. Available: https://arxiv.org/abs/2208.13054

[34] C. Benz and V. Rodehorst, "Omnicrack30k: A benchmark for crack segmentation and the reasonable effectiveness of transfer learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2024W/VAND/papers/Benz_OmniCrack30k_A_Benchmark_for_Crack_Segmentation_and_the_Reasonable_Effectiveness_CVPRW_2024_paper.pdf

[35] N. Catalano and M. Matteucci, "Few-shot semantic segmentation: a review of methodologies, benchmarks, and open challenges," *arXiv preprint arXiv:2304.05832*, 2023. [Online]. Available: https://arxiv.org/abs/2304.05832

[36] R. Jiao, Y. Zhang, L. Ding, B. Xue, J. Zhang, R. Cai, and C. Jin, "Learning with limited annotations: A survey on deep semi-supervised learning for medical image segmentation," *Computers in Biology and Medicine*, 2023. [Online]. Available: https://arxiv.org/abs/2207.14191

[37] Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler, "Datasetgan: Efficient labeled data factory with minimal human effort," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [Online]. Available: https://arxiv.org/abs/2104.06490

[38] T. Jin, X. W. Ye, and Z. X. Li, "Establishment and evaluation of conditional GAN-based image dataset for semantic segmentation of structural cracks," *Engineering Structures*, vol. 285, p. 116058, 2023. [Online]. Available: https://doi.org/10.1016/j.engstruct.2023.116058

[39] W. Wu, Y. Zhao, M. Z. Shou, H. Zhou, and C. Shen, "Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. [Online]. Available: https://arxiv.org/abs/2303.11681

[40] J. Schnell, J. Wang, L. Qi, V. T. Hu, and M. Tang, "Scribblegen: Generative data augmentation improves scribble-supervised semantic segmentation," *arXiv preprint arXiv:2311.17121*, 2023. [Online]. Available: https://arxiv.org/abs/2311.17121

[41] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[42] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[43] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022.

[44] J. Leiñena, F. A. Saiz, and I. Barandiaran, "Latent

diffusion models to enhance the performance of visual defect segmentation networks in steel surface inspection," *Sensors*, vol. 24, no. 18, p. 6016, 2024. [Online]. Available: https://doi.org/10.3390/s24186016

[45] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, "Synthetic data from diffusion models improves ImageNet classification," *Transactions on Machine Learning Research*, 2023.

[46] B. Trabucco, K. Doherty, M. Gurinas, and R. Salakhutdinov, "Effective data augmentation with diffusion models," in *International Conference on Learning Representations (ICLR)*, 2024. [Online]. Available: https://arxiv.org/abs/2302.07944

[47] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [Online]. Available: https://arxiv.org/abs/2012.07177

[48] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. J. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. [Online]. Available: https://arxiv.org/abs/1905.04899

[49] B. Khosravi, F. Li, T. Dapamede *et al.*, "Synthetically enhanced: unveiling synthetic data's potential in medical imaging research," *eBioMedicine*, vol. 104, p. 105174, 2024. [Online]. Available: https://www.thelancet.com/journals/ebiom/article/PIIS2352-3964(24)00209-3/fulltext

[50] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, 2023. [Online]. Available: https://doi.org/10.1016/j.patter.2023.100804

[51] A. Apicella, F. Isgrò, and R. Prevete, "Don't push the button! exploring data leakage risks in machine learning and transfer learning," *Artificial Intelligence Review*, vol. 58, 2025. [Online]. Available: https://doi.org/10.1007/s10462-025-11326-3

[52] S. Li and X. Zhao, "A performance improvement strategy for concrete damage detection using stacking ensemble learning of multiple semantic segmentation networks," *Sensors*, vol. 22, no. 9, p. 3341, 2022. [Online]. Available: https://doi.org/10.3390/s22093341

[53] T. Lee, J.-H. Kim, S.-J. Lee, S.-K. Ryu, and B.-C. Joo, "Improvement of concrete crack segmentation performance using stacking ensemble learning," *Applied Sciences*, vol. 13, no. 4, p. 2367, 2023. [Online]. Available: https://doi.org/10.3390/app13042367

[54] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. [Online]. Available: https://arxiv.org/abs/1612.01474

[55] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning (ICML)*, 2017.

[56] D. Wang, B. Gong, and L. Wang, "On calibrating semantic segmentation models: Analyses and an algorithm," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [Online]. Available: https://arxiv.org/abs/2212.12053

[57] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[58] U. A. C. of Engineers, "Levee owner's manual for non-federal flood control works," https://www.mvr.usace.army.mil/Portals/48/docs/EC/LSP/LeveeOwnersManual.pdf, accessed 2025-02-17.

[59] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[60] P. J. Thapa and M. T. Hoque, "Synthetic sand boil dataset for levee monitoring," IEEE DataPort, Sep. 2024. [Online]. Available: https://ieee-dataport.org/documents/synthetic-sand-boil-dataset-levee-monitoring

[61] P. J. Thapa and M. T. Hoque, "Synthetic animal burrow dataset for levee monitoring," IEEE DataPort, Jun. 2025. [Online]. Available: https://ieee-dataport.org/documents/synthetic-animal-burrow-dataset-levee-monitoring

[62] P. J. Thapa, "Convex_Hull_Annotator: Automated convex-hull mask annotation from a SandBoilNet probability map," https://github.com/padam56/Convex_Hull_Annotator, 2024, companion tool to [3]; applies thresholding, connected-component analysis, and per-component convex hulls to a CNN-predicted sandboil probability map.

[63] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[64] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel squeeze & excitation in fully convolutional networks," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2018.

[65] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[66] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[67] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[68] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[69] T. Fan, G. Wang, Y. Li, and H. Wang, "MA-Net: A multi-scale attention network for liver and tumor segmentation," *IEEE Access*, vol. 8, pp. 179 656–179 665, 2020.

[70] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *European Conference on Computer Vision (ECCV)*, 2018.

[71] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [Online]. Available: https://arxiv.org/abs/1709.01507

[72] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[73] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, vol. 32, pp. 8024–8035.

[74] P. Iakubovskii, "Segmentation models PyTorch," https://github.com/qubvel/segmentation_models.pytorch, 2019.

[75] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[77] N. Ravi, V. Gabeur, Y.-T. Hu *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. [Online]. Available: https://arxiv.org/abs/2408.00714