Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CLARA: Cross-View Landmark Acquisition, Retrieval, and Anchoring as an Edge-Deployed Visual Trigger for Loosely Coupled UAV Navigation

Zakhar Ostrovskyi
Khmelnytskyi National University
Oleksander Barmak
Khmelnytskyi National University
Pavlo Radiuk
radiukp@khmnu.edu.ua

Khmelnytskyi National University
Iurii Krak
Taras Shevchenko National University of Kyiv

Research Article

Keywords: INS drift correction, UAV visual anchoring, cross-view geo-localization, landmark-based localization, GNSS-degraded navigation, edge inference

Posted Date: June 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9958865/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

CLARA: Cross-View Landmark Acquisition, Retrieval, and Anchoring as an Edge-Deployed Visual Trigger for Loosely Coupled UAV Navigation

Zakhar Ostrovskyi[1], Oleksander Barmak[1], Pavlo Radiuk[1*], Iurii Krak[2,3]

[1]Department of Computer Science, Khmelnytskyi National University, 11, Institutes str., Khmelnytskyi, 29016, Ukraine.
[2]Department of Theoretical Cybernetics, Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska str., Kyiv, 01601, Ukraine.
[3]Laboratory of Communicative Information Technologies, V.M. Glushkov Institute of Cybernetics, 40, Glushkov Ave., Kyiv, 03187, Ukraine.

*Corresponding author(s). E-mail(s): radiukp@khmnu.edu.ua;
Contributing authors: ostrovskyiz@khmnu.edu.ua;
barmako@khmnu.edu.ua; iurii.krak@knu.ua;

Abstract

Modern unmanned aerial vehicles in Global Navigation Satellite Systems (GNSS)-degraded environments rely on inertial navigation systems (INSs), which inevitably suffer from cumulative drift. Existing visual localization methods prioritize continuous, dense image retrieval to correct this drift, often ignoring the severe penalties that false visual updates impose on the inertial state. In this work, we propose CLARA (Cross-View Landmark Acquisition, Retrieval, and Anchoring), an edge-oriented visual anchoring layer specifically designed for loosely coupled INS drift correction. CLARA utilizes an AI-agent-annotated pipeline to construct an offline, highly sparse bank of object-level landmarks, verifying scheduled targets online through a strict matching policy. Evaluated on a calibrated VPAIR urban replay, CLARA achieved 0.812 frame-level route-trigger accuracy and successfully recovered 0.723 of eligible landmark events, strictly limiting incorrect updates to 4.37 false accepts per 100 non-landmark frames. Furthermore, CLARA requires only 0.28 MiB per 100 km compared to 477.8 MiB

for AnyLoc-VLAD-DINOv2. While serving as a single-route feasibility study, these results demonstrate that route-conditioned, object-level anchoring yields reliable and memory-efficient correction signals for conservative visual-trigger mechanisms suitable for edge-deployed autonomous-flight pipelines.

Keywords: INS drift correction, UAV visual anchoring, cross-view geo-localization, landmark-based localization, GNSS-degraded navigation, edge inference

1 Introduction

Unmanned aerial vehicles (UAVs) have become integral to complex autonomous operations, facilitating crucial tasks such as dynamic trajectory adaptation during industrial infrastructure inspections [1]. Additionally, their application in real-time, edge-deployed scenarios, such as onboard environmental hazard detection, underscores the necessity of robust autonomous navigation [2]. However, when relying primarily on Global Navigation Satellite Systems (GNSS), UAVs are severely vulnerable in environments where satellite signals are intentionally spoofed, naturally degraded, or structurally unavailable [3]. In these scenarios, maintaining accurate, continuous flight paths inherently requires the deployment of alternative GNSS-independent sensing frameworks [4].

While inertial navigation systems (INS) serve as the indispensable primary propagation backbone for these vehicles, their integrations suffer from unbounded state drift over time, establishing an operational requirement for exteroceptive sensors to deliver absolute map-tied corrections [5]. Simultaneous localization and mapping (SLAM) frameworks have significantly advanced robust perception in real-world conditions [6]. Specific contemporary implementations successfully demonstrate the empirical efficacy of tightly coupled visual-inertial state estimation [7]. Similarly, multi-map SLAM libraries provide resilient local consistency structures [8]. Nevertheless, continuous estimation techniques remain inherently prone to perceptual aliasing and critical loop-closure failures over extended GNSS-denied segments. Addressing these limitations, visual place recognition offers an alternative relocalization mechanism explicitly designed to operate under substantial changes in environmental appearance [9]. Building on this retrieval concept, landmark-based localization introduces map-tied semantic anchors that provide explicit, verifiable correction signals [10].

Executing visual localization from an aerial perspective requires algorithms capable of bridging the significant appearance gap between onboard camera feeds and reference satellite imagery, a problem that originated as a fundamental cross-view geolocalization challenge [11]. The availability of high-quality paired route data continuously exposes algorithms to practical in-plane rotations and geographic shifts [12]. Research across these datasets has progressively transitioned from retrieving overarching whole-image contexts toward isolated matching of distinct physical entities [13]. Object-centric methodologies specifically distinguish local target verification properties from broader holistic scene analysis [14]. Prior investigations into unsupervised distinctive landmark mining demonstrated the feasibility of directly isolating valuable

cross-view structural embeddings from detection networks [15]. Consequently, differentiating visually ambiguous targets at the isolated-object level remains a critical prerequisite for accurate and safe geometric localization [16].

Constructing a robust, usable landmark bank for such cross-view pipelines is frequently obstructed by the immense manual labeling costs, which can be partially mitigated through programmatic weak supervision frameworks [17]. Furthermore, implementing these AI-driven matching capabilities on edge devices necessitates strict architectural alignment with the latency, thermal, and compute constraints inherent to edge devices [18].

The identified problem is that existing visual localization methodologies heavily prioritize generic dense image retrieval, often neglecting the asymmetric risks associated with INS fusion operations where a false visual update is significantly more detrimental than a missed observation. To address this gap, we shift the navigation interface from continuous unconstrained retrieval to sparse scheduled visual anchoring with explicit abstention.

The goal of this study is to improve loosely coupled INS drift correction in edge-deployed UAV navigation by introducing a sparse, route-conditioned object-level visual anchoring framework. The objective of this study is to formulate a robust pipeline that selectively triggers safe map-coordinate updates while operating efficiently within a strict, predefined false-accept budget.

The study introduces three major scientific contributions:

- A CLARA (Cross-View Landmark Acquisition, Retrieval, and Anchoring) interface that systematically integrates route-conditioned object-level landmark verification with a downstream INS backend via sparse map-coordinate correction measurements under a conservative accept/reject/no-match taxonomy.
- A coverage-constrained landmark-bank construction approach that efficiently curates persistent geo-referenced object landmarks, transitioning the operational memory requirement from dense image galleries to highly sparse, low-dimensional descriptors.
- An AI-agent-generated supervision pipeline designed for limited-resource edge deployment, delivering structured object-level correspondence labels to train compact cross-view verification matchers under strict false-positive constraints.

The remainder of this paper is organized as follows. Section 2 reviews current paradigms in visual navigation and object localization. Section 3 formally defines the CLARA framework, bank construction, and AI-agent supervision approach. Section 3.5 details the experimental setup and evaluation protocol. Section 4 presents the empirical findings. Finally, Sections 5 and 6 discuss deployment implications and summarize the conclusions.

2 Related Work

Expanding the operational scope of aerial localization has been significantly driven by the introduction of specialized multi-source drone benchmarks [19]. Trajectory-focused evaluations subsequently shifted the experimental domain toward large-scale visual

place recognition tasks situated precisely along continuous flight paths [20]. To reliably match dynamic UAV captures to static satellite records, practical cross-view frameworks must reconcile extreme viewpoint and illumination disparities [21]. Advanced transformer architectures have recently achieved widespread adoption for handling these intricate spatial relationships across heterogeneous data sources [22]. To further refine the inherent discrimination of these models, hard-negative sampling strategies optimize contrastive boundary learning [23]. Algorithmic enhancements via joint representation learning, tightly coupled with keypoint detection, emphasize structurally invariant features across views [24]. Focusing on granular visual details, local pattern reasoning effectively isolates specific sub-region structures [25]. In operational scenarios where pure downward imagery is strictly unachievable, oblique-view coarse-to-fine localization maintains robust matching accuracy [26]. Advanced partition techniques have also incorporated contrastive attribute mining to efficiently handle position-aware spatial alignment [27]. Moreover, decentralized evaluation protocols provide stringent audits of model reliability across highly non-central target distributions [28]. Refining the feature-matching focus further, segmentation-guided methodologies purposefully restrict similarity computations exclusively to common overlapping semantic regions [29].

Focusing on individual extracted targets rather than full holistic scenes enhances the semantic grounding of the localization system [30]. However, addressing the scarcity of high-quality pixel-level annotations remains a structural barrier to scaling these object-oriented pipelines, underscoring the profound utility of weakly supervised methodologies [31]. Comprehensive theoretical surveys of noisy-label learning demonstrate that highly capable deep models can successfully extract robust decision boundaries despite significant levels of annotation noise [32].

Onboard UAV computing systems operate within strict payload and power envelopes, historically necessitating optimized convolutional network structures [33]. Re-scaling these foundations via mathematically unified width and depth scaling parameters has successfully standardized efficient deep feature extraction [34]. When dense neural inference imposes a hard execution bottleneck, highly scalable gradient-boosting frameworks offer a performant, lightweight alternative for final classification heads [35]. Contemporary edge implementations prominently feature optimized detector backbones expressly tailored for instance segmentation tasks [36]. Deploying these exact network configurations on resource-constrained embedded hardware directly mirror authentic operational deployment conditions [37].

Extremely generalized foundation methodologies increasingly aim for zero-shot capabilities, as illustrated by vision-language-action paradigms targeting robotic control policies [38]. Within the pure perception domain, promptable segmentation architectures aim to isolate arbitrary region masks without prior class categorization [39]. Alternatively, universal visual place recognition systems aggregate massive pre-trained transformer embeddings to bypass localized, task-specific fine-tuning entirely [40]. Utilizing frozen self-supervised features from models trained on massive image corpora provides robust zero-shot geometric feature extraction [41]. Scaling traditional deep classification architectures directly to massive generic geographic datasets represents another primary methodology for creating dense matching baselines [42].

Table 1: How CLARA differs from neighboring localization paradigms

Paradigm	Strength	Limitation for this task	CLARA response
Visual odometry / SLAM	Continuous relative state estimation.	Drift and relocalization failures remain possible over long-denied segments.	Preserve inertial propagation but add sparse map-tied visual anchors.
Image-level cross-view geo-localization	Strong global retrieval across viewpoint changes.	The decision unit is usually the whole image or region; reject/no-match is secondary.	Verify one scheduled object-level landmark and allow abstention.
Object-level geo-localization	Finer spatial units and closer semantic matches to landmarks.	Often assumes a queried object and does not build a route-usable bank.	Acquire the bank automatically and couple it to a schedule.
AI agent/noisy supervision	Reduces manual labeling cost.	Generic label quality does not directly imply safe navigation updates.	Use AI-agent-generated weak labels for training, then calibrate the final route trigger under a false-accept budget.
Edge perception	Enables onboard processing.	Heavy retrieval or two-backbone stacks are costly.	Reuse one compact segmentation backbone and a lightweight acceptance head.

Despite these promising findings in global image retrieval and neural network optimization, severe operational gaps remain in structurally safe, loosely coupled navigation interfaces. The extant literature primarily evaluates generic region-based localization algorithms that implicitly prioritize maximum recall over strict false-positive suppression. Furthermore, dense, continuous retrieval frameworks inherently bypass the critical UAV requirement for route-conditioned, scheduled verification, while current foundation object models remain computationally prohibitive for persistent, onboard edge inference. The proposed research contributions detailed herein directly address these critical omissions by introducing a highly sparse, coverage-constrained visual anchoring methodology engineered explicitly for environments defined by asymmetric penalty landscapes. By enforcing a strict accept/reject/no-match taxonomy calibrated through AI-agent-generated supervision, the proposed architecture successfully converts noisy object-level visual similarities into robust, scheduled correction triggers, thereby safeguarding the downstream inertial state against detrimental false visual updates.

3 Methods

3.1 CLARA formulation

Let Ω denote the territory covered by geo-referenced satellite or satellite-style reference imagery, and let $\mathcal{C} = \{(x_i, e_i)\}_{i=1}^{N_C}$ be the candidate object set extracted from that imagery. Here x_i is the map or route coordinate of candidate object i, and e_i is its embedding. The acquisition stage builds a landmark bank $\mathcal{L} = \{\ell_m\}_{m=1}^{M}$ over the support set $\mathcal{X} = \{x_i\}_{i=1}^{N_C}$. A downstream planner, outside the scope of this paper, later selects a mission path and an ordered landmark schedule $\Lambda = (\lambda_q)_{q=1}^{Q}$.



Fig. 1: CLARA inside an INS-coupled navigation stack. Offline acquisition covers the reference territory with landmarks; a planner selects a route-specific landmark schedule; online recognition verifies the next scheduled landmark and emits sparse visual anchoring measurements

During flight, the INS supplies the propagated state estimate $\hat{s}_t^-$ and the current frame yields detected objects $\mathcal{O}_t = \{o_{t,n}\}_{n=1}^{N_t}$. CLARA receives the next scheduled landmark λ_q and returns one of three decisions: accept the landmark, reject the best visual proposal, or emit no-match. This formulation is deliberately asymmetric. CLARA does not search the entire map at every frame; it tests whether the current evidence is safe enough to emit a map-tied update measurement for the expected landmark.

3.2 Coverage-constrained landmark acquisition

The acquisition stage constructs a landmark bank that is visually distinctive but also spatially useful for route-level visual updates. A pure global outlier ranking can select rare-looking objects that cluster in one route segment and leave long intervals without correction opportunities. CLARA therefore uses a coverage-first greedy acquisition rule over reference-object candidates.

Let $\mathcal{C} = \{(x_i, e_i)\}_{i=1}^{N_C}$ denote the candidate buildings, where x_i is the map or route coordinate and e_i is the embedding. In VPAIR, x_i is the paired-frame index, so ϵ_{cover} and $\epsilon_{\text{exclude}}$ are measured in frames. Multiple buildings can share the same coordinate. We therefore define coordinate groups $\mathcal{G}$, with group coordinate z_g, members $\mathcal{C}_g = \{i \mid x_i = z_g\}$, and support weight w_g. In the final VPAIR route experiments, $w_g = |\mathcal{C}_g|$, so coverage gain is measured in reference-building support while equivalent spatial neighborhoods are evaluated once.

For candidate i, define the coverage and exclusion neighborhoods as

$$\mathcal{N}_{\text{cover}}(i) = \{g \in \mathcal{G} \mid \|z_g - x_i\| \leq \epsilon_{\text{cover}}\}, \tag{1a}$$

$$\mathcal{N}_{\text{exclude}}(i) = \{g \in \mathcal{G} \mid \|z_g - x_i\| \leq \epsilon_{\text{exclude}}\}. \tag{1b}$$

The design target is a separated ϵ_{cover}-cover of the visually usable support:

$$\forall g \in \mathcal{G},\ \exists i \in \mathcal{L} : g \in \mathcal{N}_{\text{cover}}(i), \qquad \|x_i - x_j\| > \epsilon_{\text{exclude}} \quad \forall i, j \in \mathcal{L},\ i \neq j. \tag{2}$$

CLARA does not solve this combinatorial objective exactly. Instead, it uses the greedy heuristic below, which makes coverage primary and uses visual distinctiveness only to resolve coverage-competitive choices.

At iteration k, let $U_k \subseteq \mathcal{G}$ be the uncovered support groups and let $E_k \subseteq \mathcal{C}$ be the candidate buildings whose coordinate groups are not excluded. CLARA fits an IsolationForest on $\{e_i \mid i \in E_k\}$ and defines the iteration-specific distinctiveness score

$$s_k(i) = -d_k(i), \tag{3}$$

where $d_k(i)$ is the IsolationForest decision-function value and lower values indicate more anomalous embeddings.

Recomputing s_k after each exclusion step is intentional: it performs an outlier-peeling procedure over the currently eligible candidate set rather than relying on one fixed global ranking.

For each non-excluded coordinate group, CLARA retains the most distinctive eligible representative,

$$r_k(g) = \arg\max_{i \in E_k \cap \mathcal{C}_g} s_k(i). \tag{4}$$

The representative's absolute coverage gain and novelty fraction are

$$g_{\text{abs}}(g) = \sum_{h \in \mathcal{N}_{\text{cover}}(r_k(g)) \cap U_k} w_h, \tag{5}$$

and

$$g_{\text{nov}}(g) = \frac{g_{\text{abs}}(g)}{\sum_{h \in \mathcal{N}_{\text{cover}}(r_k(g))} w_h}. \tag{6}$$

Representatives whose absolute gain is within relative tolerance δ of the best gain form the coverage-feasible pool. If any representative in that pool satisfies the novelty threshold τ, the pool is restricted to those representatives. CLARA then selects the most distinctive remaining representative. Coverage and exclusion are updated separately: ϵ_{cover} marks support as covered, whereas $\epsilon_{\text{exclude}}$ prevents selecting redundant nearby landmarks. Algorithm 1 summarizes the procedure.

Algorithm 1 Greedy coverage-constrained landmark acquisition

Input: Candidate buildings $\mathcal{C}$, coordinate groups $\mathcal{G}$, weights w_g, ϵ_{cover}, $\epsilon_{\text{exclude}}$, δ, τ, $g_{\min}$
Output: Landmark bank $\mathcal{L}$

covered $\leftarrow \emptyset$; excluded $\leftarrow \emptyset$; $\mathcal{L} \leftarrow \emptyset$
while uncovered groups remain **do**
  $E_k \leftarrow \{i \in \mathcal{C} \mid \text{group}(i) \notin \text{excluded}\}$
  if $E_k = \emptyset$ **then**
    break
  end if
  Fit IsolationForest on $\{e_i \mid i \in E_k\}$ and compute distinctiveness $s_k(i) = -d_k(i)$ (Equation 3)
  for each non-excluded group g **do**
    Set group representative $r_k(g) = \arg\max_{i \in E_k \cap \mathcal{C}_g} s_k(i)$ (Equation 4)
    Compute absolute coverage gain $g_{\text{abs}}(g)$ (Equation 5)
    Compute novelty fraction $g_{\text{nov}}(g)$ (Equation 6)
  end for
  if $\max_g g_{\text{abs}}(g) < g_{\min}$ **then**
    break
  end if
  Determine candidate pool $G \leftarrow \{g \in \mathcal{G} \mid g_{\text{abs}}(g) \geq (1-\delta) \max_h g_{\text{abs}}(h)\}$
  if $\exists g \in G$ such that $g_{\text{nov}}(g) \geq \tau$ **then**
    Restrict pool $G \leftarrow \{g \in G \mid g_{\text{nov}}(g) \geq \tau\}$
  end if
  Select optimal group $g^\star \leftarrow \arg\max_{g \in G} s_k(r_k(g))$
  Append to landmark bank $\mathcal{L} \leftarrow \mathcal{L} \cup \{r_k(g^\star)\}$
  covered $\leftarrow$ covered $\cup\, \mathcal{N}_{\text{cover}}(r_k(g^\star))$ (Equation 1)
  excluded $\leftarrow$ excluded $\cup\, \mathcal{N}_{\text{exclude}}(r_k(g^\star))$ (Equation 1)
end while
return $\mathcal{L}$

Let $N_C = |\mathcal{C}|$ be the number of candidate objects, $G = |\mathcal{G}|$ the number of coordinate groups, and $M = |\mathcal{L}|$ the number of selected landmarks before stopping. If

$I(n)$ denotes the cost of fitting and scoring the IsolationForest on n eligible embeddings, one greedy iteration costs $\mathcal{O}(I(|E_k|) + |E_k| + G_k)$ when coverage and exclusion neighborhoods are precomputed over the sorted one-dimensional route coordinates, where G_k is the number of non-excluded groups. The total offline cost is therefore $\mathcal{O}\Big(\sum_{k=1}^{M}(I(|E_k|) + |E_k| + G_k)\Big)$, upper-bounded by $\mathcal{O}(M(I(N_C) + N_C + G))$. A naive implementation that recomputes all group neighborhoods by pairwise scans would instead add $\mathcal{O}(MG^2)$. This cost is incurred only during offline landmark-bank construction; online visual anchoring evaluates only the scheduled landmark against current-frame object proposals.

The stopping rule is a progress condition. Complete coverage is achieved only if the loop reaches $U_k = \emptyset$ before an empty-candidate, landmark-cap, or $g_{\text{abs}} < g_{\min}$ stop. Thus, any residual uncovered support after minimum-gain stopping or after later saliency and replay-eligibility filtering is an empirical property of the produced bank and is reported with the acquisition results.

3.3 Scheduled retrieval and visual anchoring

At time t, the scheduler exposes λ_q and the current frame provides $\mathcal{O}_t$. A proposal scorer r_θ ranks the current-frame objects and selects the proposal in Equation (7):

$$n_t^\star = \arg\max_n r_\theta(\lambda_q, o_{t,n}). \tag{7}$$

A verification function c_ϕ then evaluates only that proposal and produces the binary acceptance decision in Equation (8):

$$a_t = \mathbf{1}\left[c_\phi(\lambda_q, o_{t,n_t^\star}) \geq \tau_{\text{acc}}\right]. \tag{8}$$

If $a_t = 1$, CLARA emits the scheduled landmark identity and coordinate as a candidate update measurement. If $a_t = 0$, the module returns reject/no-match and the downstream estimator keeps inertial propagation unchanged, as summarized in Equation (9):

$$\hat{s}_t = \begin{cases} \mathcal{U}(\hat{s}_t^-, \lambda_q), & \text{if } a_t = 1, \\ \hat{s}_t^-, & \text{if } a_t = 0, \end{cases} \tag{9}$$

where $\mathcal{U}$ denotes the chosen downstream INS/localization update rule in Equation (9).

This paper evaluates the visual event trigger and does not implement the downstream filter update. The critical property is that the visual module can abstain: no-match is a first-class safety outcome, not a fallback.

3.4 AI-agent-generated object-level supervision

VPAIR does not provide building masks or cross-view building identities. CLARA therefore creates object-level supervision with a budgeted two-stage annotation agent. We use the term weak labels for AI-agent-generated object-level correspondence labels that are used for matcher training but are not treated as manually verified ground-truth identities. Each task contains the paired aerial-query and satellite-style reference images, detected object instances in both views, bounding boxes, object IDs, and

embeddings. In an inspection phase, the agent requests crops around ambiguous regions; in a reporting phase, it emits a structured object-to-object report indexed by aerial-query object ID. Each object receives exactly one of three states: exact match to one reference object, non-exact correspondence to one or more reference objects, or no-match/not-visible. The implementation uses GPT-5.2 for the inspection and reporting stages, temperature 1, one retry for model calls, and strict Building2BuildingAnnotations Pydantic-schema parsing; the full prompt, configuration, messages, inputs, raw output, and parsed schema output are saved per image pair, and only schema-parsed outputs are used as AI-agent-generated weak labels. It should also be noted that the generative-AI system was used only to produce weak object-level annotation artifacts for training/audit.

Because the annotation agent uses a proprietary generative model with stochastic decoding, exact third-party regeneration of the AI-agent-generated weak labels is not guaranteed even with the same prompt and schema. The experiments therefore treat the saved schema-parsed annotation outputs as the experimental AI-agent-generated weak-label artifact. Annotation quality is assessed through the human golden audit subset, and the AI-agent-generated weak-label annotations for this study are available from the corresponding author upon reasonable request.

The learned matcher is a generic object-to-object model, but the deployed policy is easier and safer than generic matching because it verifies only the scheduled landmark within a narrow route context. Exact automatic correspondences train the default positive set; non-exact labels are retained for diagnostics and for an alternative graded-ranker experiment. The final acceptance threshold is selected only after acquisition, retrieval, event-window semantics, and the false-accept budget are fixed.

3.5 Dataset and Experimental Setup

3.5.1 VPAIR route and derived building corpus

All experiments use VPAIR, a paired aerial visual place recognition and localization dataset [12]. It contains 2,706 one-to-one image pairs at 800 x 600 pixels. The query stream was captured from a light aircraft with a downward-facing camera, while the reference view is a satellite-style render generated from public geodata. Each pair includes pose metadata, enabling treatment of the benchmark as a geo-referenced route. The surveyed trajectory spans 106.2 km, with median inter-frame spacing of 38.5 m and approximate geographic footprint of 24.9 x 34.1 km. VPAIR also includes 10,000 distractor renders, but they are not used here because the evaluated task is scheduled building-level landmark verification along the paired route, not large-gallery image retrieval.

The building-level pipeline is restricted to the 1,234 urban frames. We reuse the YOLO11n-seg-family backbone and multi-layer embedding extraction introduced in our previous study [15], without VPAIR-specific fine-tuning. On the urban subset this yields 34,898 aerial-query building instances and 37,603 satellite-style reference building instances. Each retained instance stores detection metadata and a 1,584-D embedding obtained from pooled activations across 11 hooked intermediate layers;

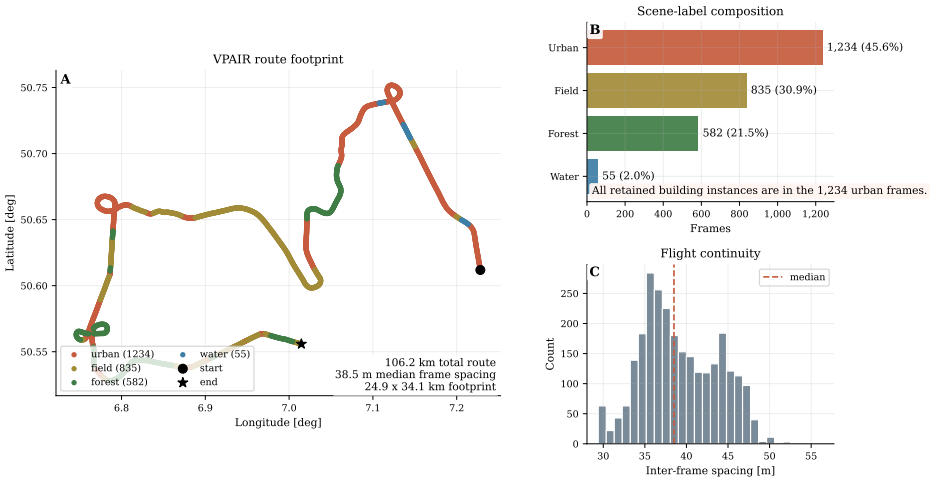


Fig. 2: VPAIR route context: geographic footprint, scene-label composition, and inter-frame spacing. The route is sequential and geo-referenced, with building-rich regions concentrated in urban segments

the deployed matcher uses the [6,9,10] subset with 640 dimensions. Backbone and preprocessing details are summarized in Appendix C.

3.5.2 Splits and matcher-selection protocol

The candidate universe contains 1,217 paired-building frames after benchmark exclusion. Annotation is allocated by cohorts that capture scene density, large-building presence, and within-frame size diversity. Split construction uses contiguous building-presence blocks to avoid treating adjacent route frames as independent. The out-of-distribution (OOD) region is the block spanning frames 850-987; the remaining in-distribution (ID) pool is sampled in 8-frame route groups with a 6/1/1 train/validation/test pattern where available.

In this manuscript, OOD is used in a restricted within-route sense. The OOD block is a contiguous geographic segment of the VPAIR trajectory that is excluded entirely from training and validation and is used only for testing. It therefore evaluates transfer to a held-out regional appearance distribution along the same route, rather than cross-dataset, cross-sensor, or route-universal out-of-distribution generalization.

Table 2: Final frame-level supports

Split / support	Frames	Role
Candidate paired-building pool	1217	Universe for cohort-aware auto-selection
Train pool	126	Weak-label pool; the final matcher uses a 72-frame subset
Auto validation	20	Matcher comparison and threshold selection
ID test	24	In-distribution transfer audit

Split / support	Frames	Role
OOD test (within-route regional block)	30	Contiguous regional-shift audit
Human golden benchmark	10	Fully human-audited exact-label sanity check

Base-matcher evaluation is pool-based. In a match-present pool, one reference object is compared against all aerial-query detections from the paired frame. In a no-match pool, the same reference object is paired with a temporally distant aerial-query frame using —delta frame— ¿= 10. The pool-screening metric reports true-positive rate (TPR) at a pool-level false-positive rate (FPR) not exceeding 1%, denoted TPR@pool-FPR¡=1%; model selection uses this metric on validation, while ID, OOD, and golden results are transfer audits only. The final matcher uses 72 training frames, corresponding to the first twelve contiguous six-frame route chunks.

3.5.3 Landmark-bank and route-replay protocol

The final route study fixes epsilon_cover = 20, epsilon_exclude = 5, delta = 0.7, and tau = 0.6 on the one-dimensional VPAIR frame-index coordinate. Reference candidates below the 25th percentile of satellite-style bounding-box area are removed, and ±3-frame event-window consolidation is applied. This produces 55 route-ordered landmarks before final replay eligibility filtering.

Any landmark whose event window touches a training frame is excluded from final route evaluation. The final calibrated replay contains 47 eligible landmark events and 846 non-landmark frames. An event is recovered if the correct landmark is accepted at least once within [t-3,t+3]; the previous landmark remains active through t+3 so that slightly late detections are not penalized as failures. The operating point is chosen under B1 = 5, meaning at most five false accepts per 100 non-landmark frames. Four contiguous calibration blocks produce fold-optimal thresholds under the B1 = 5 false-accept budget, and the deployed acceptance threshold is their median, $\tau_{\text{acc}} = 0.08646$. Because only one route-level dataset is available, this operating threshold is selected by blockwise calibration on the same route rather than by an independent calibration route. The median-of-blocks rule is used to reduce sensitivity to any single route segment, but the resulting replay metrics should be interpreted as calibrated single-route feasibility results rather than fully independent cross-route generalization estimates.

3.5.4 Same-route dense-VPR baseline protocol

To contextualize CLARA against dense image-level VPR, AnyLoc-VLAD-DINOv2 and CosPlace ResNet101-2048 are evaluated on the same 1,155 calibrated replay frames used for the final CLARA route study. The shared reported quantity, reported below as same-route decision accuracy, is the percentage of correct decisions along the route. For CLARA, a correct decision is a correct non-landmark rejection or a correct scheduled-landmark acceptance under the route-trigger protocol. For the dense VPR baselines, each query frame retrieves a Top-1 reference frame, and the decision is correct when $|\hat{i}_t - i_t| \leq 3$, corresponding to the VPAIR 3-frame localization radius.

The dense VPR baselines are intentionally evaluated without VPAIR-specific supervised adaptation. AnyLoc uses VLAD aggregation from reference/database features without gradient-based VPAIR fine-tuning; CosPlace uses the released ResNet101-2048 descriptor checkpoint without VPAIR fine-tuning. This setting tests off-the-shelf transfer to the same route-replay frames rather than the best possible performance after task-specific adaptation, and is therefore used to assess whether an existing dense image-retrieval system can directly replace CLARA's scheduled visual-anchor policy under the same route geometry.

4 Results

4.1 AI-agent-generated weak-label audit and final landmark-bank selection

The annotation agent was reliable enough to support downstream matcher development, but not reliable enough to treat automatic labels as exact ground truth. On the 10-frame human-audited benchmark, the default confidence threshold produced strict exact precision 0.774, recall 0.624, and $F_1 = 0.691$; the relaxed exact+non-exact criterion produced precision 0.778, recall 0.712, and $F_1 = 0.744$ (Table 3). The differentiated exact/non-exact/no-match taxonomy was important: it reduced the pressure to force one-to-one matches when segmentation differed across views. The crop-inspection budget was also material: strict exact F_1 rose from 0.460 with zero tool calls to 0.691 at 10 calls, while an unlimited budget reached only 0.707; we therefore retain `max_tools` = 10. Confidence sweeps, tool-budget diagnostics, and annotation-agent footprint are moved to Appendix D; Figure 7a, Figure 7b, and Figure 7c report the corresponding precision, recall, and F_1 curves.

Table 3: Annotation-agent quality at the default threshold

Setting	Precision (count; 95% CI)	Recall (count; 95% CI)	F1	Negative recall (count; 95% CI)	Kappa
strict exact	0.774 (113/146; 0.700-0.834)	0.624 (113/181; 0.552-0.692)	0.691	0.794 (54/68; 0.684-0.873)	0.451
relaxed exact+non-exact	0.778 (151/194; 0.715-0.831)	0.712 (151/212; 0.648-0.769)	0.744	0.595 (22/37; 0.435-0.737)	0.366

The final acquisition parameters produced a compact and well-distributed bank. With $\epsilon_{\text{cover}} = 20$ and $\epsilon_{\text{exclude}} = 5$ fixed, δ controlled support most strongly: settings below 0.7 reduced coverage, while more permissive values mainly relaxed density. After fixing $\delta = 0.7$, $\tau = 0.6$ preserved near-complete support without shifting the bank toward small weak landmarks. The resulting 55 landmarks form the route-ordered bank used in the final replay.

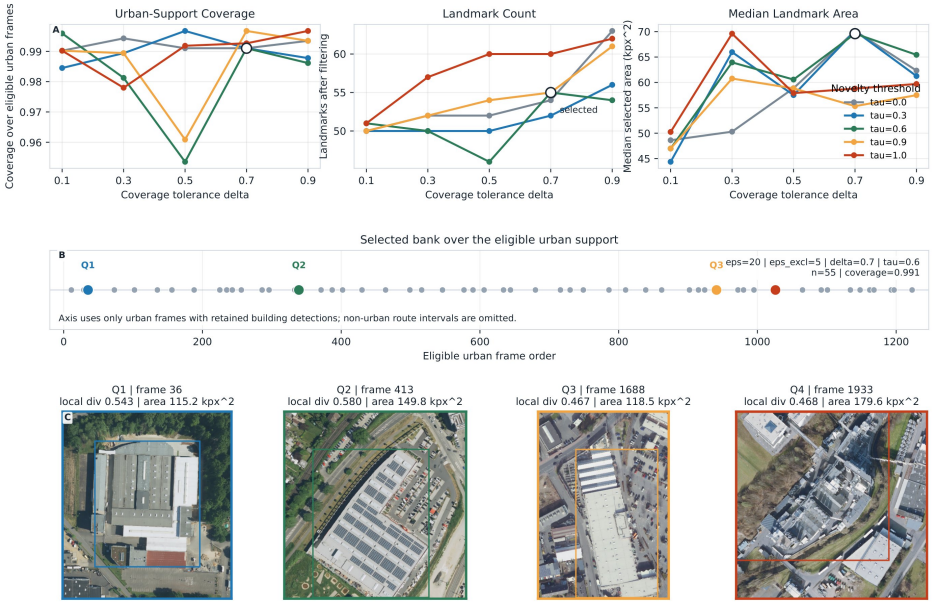


Fig. 3: Final landmark-bank configuration. The parameter trends identify the retained $\delta = 0.7$, $\tau = 0.6$ setting; the route profile shows selected landmark locations over the eligible urban support; representative selected landmarks illustrate the resulting target type

4.2 Lightweight matcher selection

Deterministic cosine similarity with a margin gate retained useful cross-view signal but was not strong enough for the final low-FPR accept/reject role: the best cosine baseline reached validation TPR@pool-FPR $\leq 1\% = 0.078$, with transfer-audit values 0.069 on ID, 0.231 on OOD, and 0.125 on the human golden benchmark. The selected learned representation uses signed embedding differences on the $[6, 9, 10]$ layer subset; absolute differences and principal component analysis (PCA)-compressed variants were rejected because they degraded the validation-side low-FPR metric. Compared with the full 1,584-D embedding, this subset improved validation behavior, reduced storage to 640 dimensions, cut training time from 31.46 s to 6.75 s, and increased CPU-proxy scoring throughput from 1.69M to 2.17M pairs/s (Table 4).

Among standalone model families, the gradient-boosted decision tree (GBDT) classifier gave the strongest practical validation-side operating point under TPR@pool-FPR $\leq 1\%$ (Figure 4). Binary LambdaRank was useful as a proposal stage but less stable as a standalone acceptance head. A graded LambdaRank variant that used non-exact correspondences as weaker relevance labels performed worse for the conservative exact-match acceptance metric. This supports the final cascade design: retrieval can be recall-oriented, but final acceptance remains a conservative classifier threshold.

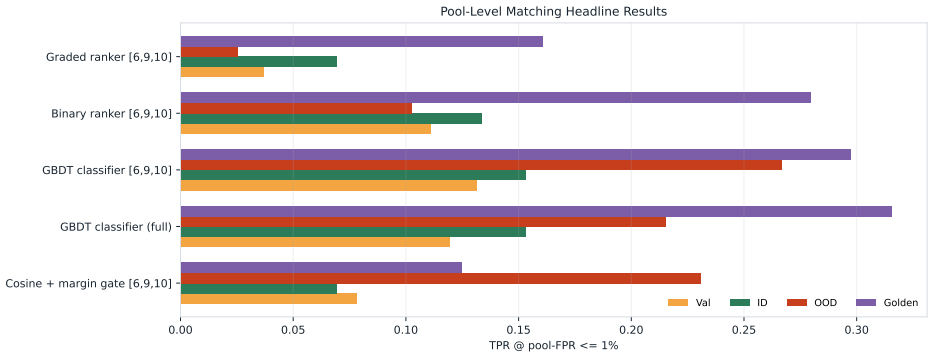


Fig. 4: Pool-level matcher selection and transfer audit. Bars report TPR@pool-FPR $\leq$ 1% for validation-driven selection and ID/OOD/golden post-selection audits. Decomposed split-specific alternatives are provided in Appendix E

Table 4: Deployment-oriented comparison of the selected GBDT variants

Comparison parameter	Common setting	Full embedding	[6,9,10] subset
Pairwise feature construction	signed difference $e_{\text{sat}} - e_{\text{uav}}$	-	-
GBDT hyperparameters	`learning_rate` = 0.03, `num_leaves` = 95, `max_depth` = 10, `min_data_in_leaf` = 10, `subsample` = 0.8, `colsample_bytree` = 0.8, `reg_lambda` = 1.5, `num_boost_round` = 96	-	-
Feature layers	-	all	6, 9, 10
Feature dimension	-	1,584	640
Embedding storage (KiB per building, float32)	-	6.19	2.50
Model size (KiB)	-	1088.7	1026.3
Training time (s)	-	31.46	6.75
Scoring throughput (pairs/s, CPU proxy)	-	1.69M	2.17M

The extended train-size and label-quality diagnostics are moved to Appendix F. They show that high-confidence labels improve classifier sample efficiency on validation and ID (Figure 11a and 11b) and improve retrieval ordering (Figure 12a and 12b), while OOD performance remains limited by threshold transfer and territorial shift. The ranker follows the same retrieval improvement but shows FPR drift when used as a final gate (Figure 13a–13b and Figure 14a–14b).

4.3 Final sequential visual-anchoring evaluation

The deployed cascade on the $\epsilon_{\text{cover}} = 20$ landmark bank attained $938/1155 = 0.812$ frame-level route-trigger accuracy under accept/reject/no-match semantics, recovered $34/47 = 0.723$ eligible landmark events, and correctly rejected $809/846 = 0.956$ non-landmark frames (Table 5). Frame-level route-trigger accuracy is computed over all

evaluated replay frames: each frame contributes one binary decision. A non-landmark frame is correct when CLARA rejects or emits no update, whereas a landmark-window frame is correct when CLARA accepts the scheduled landmark according to the route-trigger protocol. Event-level landmark recall is reported separately: a landmark event is counted as recovered if the correct landmark is accepted at least once within its admissible window $[t-3, t+3]$; the previous landmark remains active through $t+3$ so that slightly late detections are not penalized as failures. The realized false-accept point estimate was $37/846 \times 100 = 4.37$ false accepts per 100 non-landmark frames, below the $B1 = 5$ operating budget; the Wilson interval is reported in Table 5. These metrics evaluate whether the visual perception module triggers candidate landmark updates at correct route locations; they do not measure downstream position error after INS fusion.

Table 5: Final calibrated route-trigger metrics for the deployed cascade.

Metric	Unit / denominator	Count	Value	Wilson 95% CI
Frame-level route-trigger accuracy	all evaluated replay frames	938/1155	0.812	0.789-0.834
Non-landmark rejection accuracy	non-landmark frames	809/846	0.956	0.940-0.968
Landmark-window frame accuracy	landmark-window frames	129/309	0.417	0.364-0.473
Event-level landmark recall	eligible calibrated-replay landmark events	34/47	0.723	0.582-0.831
False accepts per 100 non-landmark frames	$\mathrm{FP}_{\mathrm{nonlandmark}}/N_{\mathrm{nonlandmark}} \times 100$	37/846	4.37 per 100	3.19-5.97 per 100
Longest incorrect streak	consecutive frames	4	-	-

For reference, a degenerate never-update policy would correctly reject all 846 non-landmark frames but recover no landmark events, giving $846/1155 = 0.732$ frame-level route-trigger accuracy. This value is a class-imbalance sanity check, not a meaningful navigation policy.

Figure 5 shows that false accepts are sparse rather than route-wide. Missed events concentrate on a limited set of difficult landmarks. Qualitative examples in Appendix G indicate that successful detections often exploit repeated nearby views, while residual misses mainly come from weak landmark distinctiveness, segmentation split/merge artifacts, and small structures that do not remain stable across views. A manual audit of missed landmarks attributes roughly half of the residual errors to forced low-distinctiveness landmark choices, about two-fifths to segmentation noise and cross-view merging artifacts, and only a small remainder to threshold misses after correct ranking. The final bottleneck is therefore less the lightweight acceptance head and more the quality of the landmark bank and object substrate.

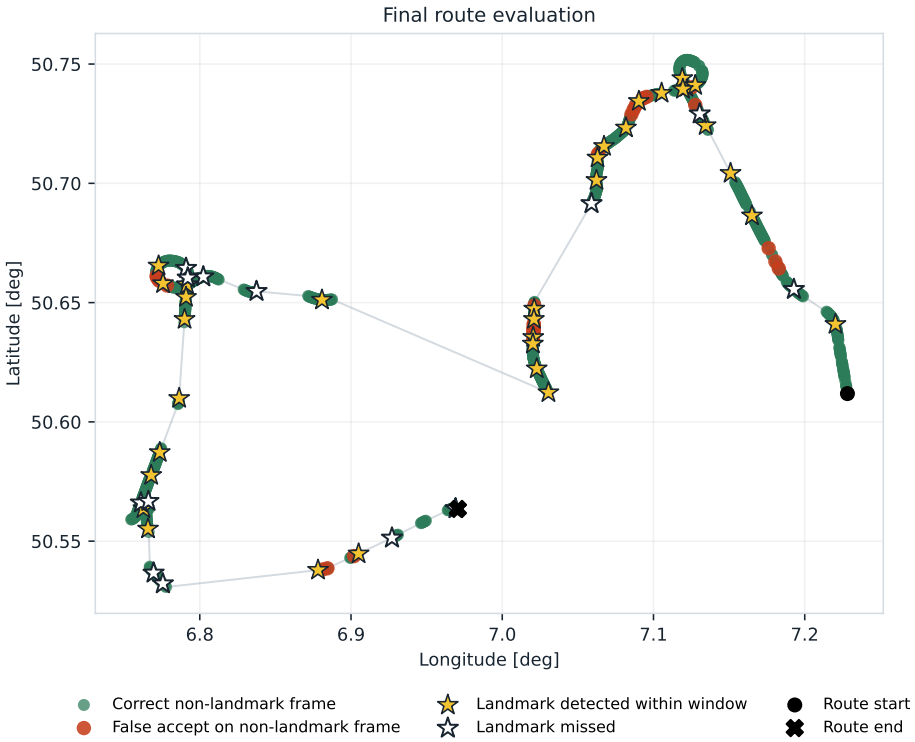


Fig. 5: Final route replay for the deployed cascade. The legend distinguishes correctly rejected non-landmark frames, false accepts on non-landmark frames, recovered landmark events, and missed landmark events

4.4 Same-route comparison with dense image-level VPR baselines

Table 6 compares CLARA with the two dense VPR baselines on the same calibrated replay frames. CLARA's 0.812 value is frame-level route-trigger accuracy under accept/reject/no-match semantics. The AnyLoc-VLAD-DINOv2 and CosPlace ResNet101-2048 values are contextual same-route scores: the dense baselines are evaluated as Top-1 retrieval systems, and a decision is correct when the retrieved reference frame falls within the 3-frame localization radius. The margin over AnyLoc is therefore interpreted conservatively. The main result is not generic dominance over dense VPR, but that the proposed sparse visual-anchoring policy achieves competitive same-route decision performance while solving a stricter accept/reject/no-match anchoring problem and using a far smaller route representation. The CosPlace result highlights a different limitation: a compact CNN descriptor trained through generic VPR place-classification supervision transfers poorly to this VPAIR urban route-replay setting when used off the shelf, without VPAIR-specific adaptation. In contrast, CLARA

exploits route structure explicitly through a sparse landmark bank, object-level evidence, and conservative non-landmark rejection.

Table 6: Same-route comparison with dense image-level VPR baselines

Comparison axis	CLARA	AnyLoc-VLAD-DINOv2	CosPlace ResNet101-2048
Baseline role	Proposed sparse object-level visual anchoring method	High-capacity dense frozen-transformer VPR baseline	CNN VPR-supervised transfer baseline
Online decision unit	Scheduled landmark accept/reject/no-match	Dense Top-1 reference-frame retrieval	Dense Top-1 reference-frame retrieval
Correct-decision criterion	Correct scheduled-landmark acceptance on landmark-window frames or correct reject/no-match on non-landmark frames	Retrieved frame within the 3-frame localization radius	Retrieved frame within the 3-frame localization radius
Evaluated replay frames	1,155	1,155	1,155
Same-route decision accuracy	0.812	0.790	0.140
Training / adaptation character	Route-specific sparse landmark bank; no VPAIR fine-tuning of the segmentation backbone	Frozen self-supervised features; unsupervised VLAD aggregation over reference/database descriptors	Externally VPR-supervised on SF-XL; no VPAIR fine-tuning
Dominant online neural model	YOLO11n-seg-family building segmenter; lightweight GBDT acceptance head	DINOv2 ViT-G/14 backbone with VLAD aggregation	ResNet101-GeM-FC descriptor model
Approximate neural parameters	∼2.9M for YOLO11n-seg-family perception backbone	∼1.1B for ViT-G/14 backbone	∼46.7M for ResNet101-2048 descriptor model
Stored descriptor dimension	640	49,152	2,048
Stored route descriptors per 100 km	114	2,548	2,548
FP32 route-map descriptor memory per 100 km	0.28 MiB	477.8 MiB	19.9 MiB
Edge-compute interpretation	Edge-oriented; online cost is dominated by the YOLO11n-seg-family segmenter, and final target-hardware throughput remains to be measured	Not practical for onboard descriptor extraction in the standard ViT-G/14 configuration	Possible as a low-rate or optimized descriptor baseline, but weak in this transfer setting
Main interpretation	Competitive same-route decision performance while solving a stricter accept/reject/no-match anchoring problem with orders-of-magnitude smaller route storage	Strong contextual retrieval score, but dense, high-dimensional, and computationally heavy	Compact relative to AnyLoc, but poor transfer to the VPAIR urban replay without adaptation

The storage contrast is substantial even before considering dense nearest-neighbor search. CLARA stores about 114 scheduled landmark descriptors per 100 km, whereas dense frame-retrieval baselines store about 2,548 reference-frame descriptors per 100 km. With float32 descriptors, this corresponds to 0.28 MiB for CLARA, 477.8 MiB for AnyLoc-VLAD-DINOv2, and 19.9 MiB for CosPlace ResNet101-2048 per 100 km. Thus the practical advantage is the joint effect of object-level sparsity, conservative

route-triggering, and compact landmark descriptors rather than a small standalone accuracy margin.

5 Discussion

The rapid expansion of UAVs for critical tasks, ranging from dynamic industrial infrastructure inspections [1] to real-time onboard environmental hazard detection [2], imposes strict requirements on autonomous navigation reliability. However, UAVs face severe vulnerabilities when operating in environments where GNSS are intentionally spoofed, naturally degraded, or structurally inaccessible [3]. Addressing the indispensable requirement for GNSS-independent sensing frameworks [4], this study proposed the CLARA framework. CLARA is specifically engineered to mitigate the unbounded state drift inherent to primary INS [5] by providing loosely coupled, highly reliable map-tied visual corrections. As evidenced by the calibrated route replay metrics presented in Table 5, the deployed cascade achieved a 0.812 frame-level route-trigger accuracy and successfully recovered 0.723 of eligible landmark events. Crucially, the system maintained operational safety by strictly adhering to a budget of 4.37 false accepts per 100 non-landmark frames. By shifting the navigational paradigm from continuous global retrieval to a scheduled, sparse visual anchoring architecture defined by a conservative accept/reject/no-match taxonomy, CLARA substantially improves the deployment viability of autonomous aerial systems operating within asymmetric penalty landscapes.

Historically, robust perception in real-world conditions has been dominated by continuous SLAM frameworks [6]. While sophisticated implementations of tightly coupled visual-inertial state estimation have demonstrated significant empirical efficacy [7], and multi-map SLAM libraries provide resilient local consistency structures [8], these techniques inherently suffer from perceptual aliasing and critical loop-closure failures over extended GNSS-denied segments. Visual place recognition (VPR) mitigates some of these limitations by offering a relocalization mechanism explicitly designed to operate under substantial changes in environmental appearance [9]. However, directly translating generic VPR into aerial navigation requires bridging the massive appearance gap between onboard camera feeds and reference satellite imagery, a challenge rooted in cross-view geolocalization [11]. Contemporary datasets expose algorithms to realistic in-plane rotations and geographic shifts [12], yet existing solutions often implicitly prioritize generic dense image retrieval. This unconstrained retrieval often neglects the asymmetric risks associated with INS fusion operations, in which an erroneous visual update exerts a significantly more destructive influence on the state estimate than a missed observation. As demonstrated in Table 6 and evaluated in Section 4.4, while high-capacity dense baselines such as AnyLoc-VLAD-DINOv2 can achieve competitive contextual retrieval scores, they lack the explicit abstain mechanisms required for loosely coupled INS fusion. By integrating a strict accept/reject/no-match taxonomy (Equation 9), CLARA prevents unsafe coordinate updates and converts noisy cross-view similarities into robust, scheduled correction triggers.

To successfully bridge the severe viewpoint and illumination disparities inherent in drone-to-satellite matching, recent research has progressively transitioned from

retrieving overarching whole-image contexts toward the isolated matching of distinct physical entities [13]. Focusing on individual isolated targets rather than full holistic scenes advances the semantic grounding of the localization system, as local target verification properties are cleanly distinguished from broader holistic scene analysis [14]. Prior investigations into unsupervised distinctive landmark mining have demonstrated the feasibility of directly isolating valuable cross-view structural embeddings from generic detection networks [15], indicating that differentiating visually ambiguous targets at the isolated-object level is a critical prerequisite for accurate geometric localization [16]. Building upon these object-centric foundations, the CLARA framework introduces a coverage-constrained landmark acquisition algorithm (Algorithm 1) that systematically curates persistent, geo-referenced building landmarks. This objective, mathematically formalized in Equation 2, purposefully avoids global outlier ranking in favor of a separated epsilon-cover of visually usable support. It dynamically balances absolute coverage gain (Equation 5) with novelty fractions (Equation 6) through a greedy heuristic process. Consequently, the constructed landmark bank, visualized in Figure 3 and detailed in Section 3.2, remains both visually distinctive and spatially distributed. This guarantees scheduled correction opportunities along the flight path without overloading the system with redundant, visually ambiguous targets that could lead to false alignments.

A significant barrier to scaling object-oriented cross-view pipelines is the immense cost of acquiring high-quality pixel-level annotations [31]. While modern deep models can successfully extract robust decision boundaries despite significant levels of annotation noise [32], exhaustive manual labeling of drone-to-satellite structural correspondences remains prohibitive. CLARA effectively mitigates this limitation by employing an AI-agent-generated weak-supervision pipeline, heavily inspired by programmatic weak-supervision frameworks [17]. By directing a sophisticated generative model to analyze paired aerial-query and satellite-style reference images through a multi-stage inspection and structured reporting protocol, the system successfully extracts usable object-level correspondence labels without manual intervention. As detailed in Table 3 and analyzed via the threshold sweeps in Figure 7a, Figure 7b, and Figure 7c, this AI-agent generated supervision proved reliable for downstream matcher development. High-confidence labels extracted via this methodology improved classifier sample efficiency on validation datasets and reliably structured the retrieval ordering necessary for the scheduled visual anchoring stage (Equation 7). Ultimately, this mechanism successfully bypasses a major bottleneck in landmark-based localization [10], validating the utility of weakly supervised intelligent agents in curating training corpora for highly specialized autonomous navigation tasks.

Implementing AI-driven matching capabilities for UAVs requires strict architectural alignment with the latency, thermal, and compute constraints inherent to edge devices [18]. Foundational deep classification architectures, scaled directly to massive, generic geographic datasets [42], or universal visual place recognition systems aggregating massive pre-trained transformer embeddings [40], are computationally exceptional but remain challenging to deploy persistently under the target resource constraints. By contrast, CLARA explicitly addresses edge-deployment requirements by reusing an optimized YOLO11n-seg-family segmentation backbone and pairing it

with a highly scalable GBDT classification head. As summarized in Table 12 and Table 4, the deployment-oriented extraction of the embedding subset cuts training time drastically while maximizing scoring throughput to over two million pairs per second in a CPU proxy environment. Furthermore, the operational memory requirement is minimized. As derived in Table 6, storing the dense descriptors required by AnyLoc-VLAD-DINOv2 demands 477.8 MiB per 100 kilometers of route map memory, and the CosPlace ResNet101-2048 VPR-supervised transfer baseline requires 19.9 MiB. CLARA, leveraging its sparse, coverage-constrained landmark bank, requires only 0.28 MiB of landmark storage per 100 kilometers. This extreme sparsity shifts the navigational paradigm from memory-intensive continuous global retrieval to highly efficient, route-conditioned anchoring (Equation 8), demonstrating that edge-oriented frameworks can outperform generic dense architectures in real-world applications.

The methodological advantages of the proposed CLARA framework are directly observed in the final sequential visual-anchoring evaluation (Figure 5). First, the system inherently incorporates an asymmetric decision threshold that strongly protects against false positives. The explicit integration of a conservative verification function safely maintains inertial propagation when visual evidence is insufficient or contradictory, avoiding catastrophic map-coordinate jumps. Second, the offline greedy coverage-constrained landmark acquisition eliminates the necessity to search the entire map at every frame, drastically reducing online computational overhead while preserving the necessary correction intervals for the INS backend. Third, by focusing exclusively on object-level semantic anchors, CLARA effectively minimizes the influence of transient environmental distractors, such as moving vehicles, vegetation changes, and dynamic shadows, which frequently disrupt traditional whole-image VPR techniques. These design principles ensure that scheduled correction triggers remain precise, verifiable, and strictly constrained within the predefined false-accept budget, enhancing the integration stability of loosely coupled INS frameworks.

Despite the significant empirical success, the proposed framework has distinct disadvantages and operational sensitivities. Primarily, the system's performance is critically bound to the stability and accuracy of the underlying instance segmentation backbone. Qualitative residual failures documented in Figure 16 reveal that cross-view architectural merging artifacts and weak small-landmark detection remain problematic. When the instance segmenter erroneously splits a large building in the drone view but merges it in the satellite view, the downstream verification network receives heavily distorted geometric features, leading directly to threshold-sensitive late-window misses or outright rejection. Furthermore, the reliance on a strict scheduled sequence assumes that the downstream planner and inertial state remain sufficiently bounded to execute the next query within a reasonable temporal window. If the inertial state drifts far beyond the expected schedule or if continuous, dense visual obstructions occur, the framework currently lacks a dense, unconstrained global relocalization fallback mechanism to rapidly reinitialize the tracking sequence.

Ultimately, the research exhibits several notable limitations and potential shortcomings, delineating necessary avenues for future investigation. The current empirical evaluation is strictly confined to a single urban trajectory within the VPAIR dataset, leveraging buildings as the singular dominant landmark class. This geographic and

semantic restriction inherently limits claims of route-universal generalization. Future research challenges include expanding the acquisition principles to highly unstructured rural or agricultural environments, where persistent segmented regions, such as field boundaries, road intersections, or natural water bodies, must replace rigid architectural structures. Additionally, while the framework rigorously evaluates a conservative visual event-trigger metric, evaluating the system inside a fully physical downstream INS-filter simulation is necessary to directly measure the resulting absolute positional error. Finally, an unresolved open question remains regarding the true end-to-end processing latency, thermal throttling effects, and power consumption profiles when the complete cascade is actively deployed on physical onboard robotics platforms operating under realistic atmospheric flight conditions.

6 Conclusions

In this study, we introduced CLARA, a sparse, object-level visual anchoring framework that provides sparse, map-tied visual correction candidates for a loosely coupled INS backend for UAVs operating in GNSS-degraded environments. By transitioning from unconstrained continuous image retrieval to a highly conservative, scheduled landmark verification policy, the proposed architecture significantly optimizes edge-deployment feasibility. Evaluated on a calibrated VPAIR urban replay, CLARA demonstrated robust performance, achieving 0.812 frame-level route-trigger accuracy and recovering 0.723 of eligible landmark events. Most critically, it effectively protected the downstream navigation state by strictly limiting erroneous updates to just 4.37 false accepts per 100 non-landmark frames. Furthermore, the framework required only 0.28 MiB of map storage per 100 kilometres, outperforming high-capacity, dense baselines that demand hundreds of megabytes. Despite these promising findings, the study is currently limited to a single urban trajectory that relies predominantly on building segmentation, which remains susceptible to cross-view architectural merging artefacts.

Future research must address these challenges by expanding landmark acquisition to unstructured rural landscapes utilizing alternative semantic boundaries. Additionally, integrating the visual triggers into a physical downstream inertial filter to formally evaluate absolute positional error and validating real-time processing latency on active onboard hardware remain critical directions for advancing resilient autonomous flight.

Declarations

- **Funding:** The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.
- **Competing interests:** The authors have no relevant financial or non-financial interests to disclose.
- **Ethics approval:** This study did not involve human participants, human data, animals, or biological material. The study is evaluated on public benchmark data and is intended for civil autonomous-navigation robustness and safety research. It does not provide operational guidance for harmful use, evasion, or deployment in restricted environments.
- **Consent to participate:** Not applicable.

- **Consent for publication:** Not applicable.
- **Data availability:** The original VPAIR dataset is not redistributed with this manuscript. It is available through the official VPAIR repository at https://github.com/AerVisLoc/vpair. The VPAIR sample data are available through Zenodo at https://doi.org/10.5281/zenodo.6473989. The building-segmentation training corpus used for the segmentation backbone is the Roboflow Universe Buildings Instance Segmentation dataset, available at https://universe.roboflow.com/roboflow-universe-projects/buildings-instance-segmentation. The AI-agent-generated weak-label annotations for this study are available from the corresponding author upon reasonable request.
- **Materials availability:** Not applicable. This study utilized commercially available software and open-source datasets. No new physical materials or proprietary hardware were developed.
- **Code availability:** The code used for CLARA experiments is available in the public repository https://github.com/Zakhar17/CLARA.
- **Use of generative artificial intelligence:** GPT-5.2 was used as a disclosed research tool for AI-agent-generated object-level annotation under the prompt, retry, tool-use, and Pydantic-schema protocol described in Section 3.4. Because this annotation model is proprietary and uses stochastic decoding, exact regeneration of the AI-agent-generated weak labels by third parties is not guaranteed. The authors therefore treat the saved schema-parsed annotations as the experimental AI-agent-generated weak-label artifact and reviewed annotation quality using the human golden audit subset. No generative-AI system is listed as an author. The authors take responsibility for all experiments, analysis, and conclusions.
- **Author contributions:** All authors contributed to the study conception and design. Zakhar Ostrovskyi and Oleksander Barmak conceived and conducted the experiments, Zakhar Ostrovskyi, Oleksander Barmak, and Pavlo Radiuk validated the results, Oleksander Barmak, Pavlo Radiuk and Iurii Krak made formal analysis, Zakhar Ostrovskyi did investigation, Pavlo Radiuk and Iurii Krak conducted data curation, Zakhar Ostrovskyi prepared the draft of the manuscript, Oleksander Barmak, Pavlo Radiuk and Iurii Krak reviewed and edited the manuscript, Zakhar Ostrovskyi made visualizations, Iurii Krak and Oleksander Barmak provided the supervision. All authors read and approved the final version of the manuscript.

Appendix A Cohort Design and Split Diagnostics

The automatic-annotation cohort grid uses three axes over paired-building frames: scene density, large-building presence, and within-frame size diversity. Scene density is density_min = min(n_query, n_ref), discretized into tertiles: low for density_min ¡= 16, mid for 16 ¡ density_min ¡= 30, and high for density_min ¿ 30. Large-building presence indicates whether the largest detected building in both views exceeds the dataset-level q92 of per-building area. Size diversity uses the minimum cross-view log-area spread, div_spread = min(log(p90/p50)), thresholded at the dataset median 0.955. Alternative diversity measures were rejected when they were too correlated with scene density.

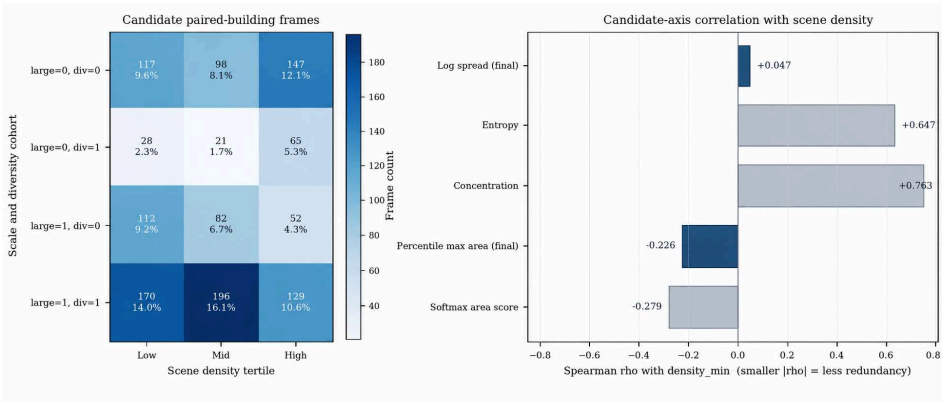


Fig. 6: Cohort design for AI-agent-generated object-level supervision sampling. The retained grid balances scene density, large-building presence, and size diversity while keeping the OOD block contiguous

Appendix B Stage-Wise Metric Map

Table 7: Stage-wise evaluation map

Stage	Primary metric	Secondary metrics	Rationale
Annotation agent	Strict exact-match F1 on the human golden subset	Relaxed F1, no-match rejection, confidence-threshold sensitivity, token/tool footprint	Labels train matchers rather than directly trigger navigation updates.
Base matcher selection	TPR@pool-FPR$_{\text{j}}$=1% with validation-tuned threshold	Top-1, Recall@5, mean reciprocal rank (MRR) / normalized discounted cumulative gain (NDCG@5), model size, CPU throughput	The final system needs rare false accepts on no-match pools.
Calibrated route replay	Event-level landmark recall under B1 = 5 false accepts per 100 non-landmark frames	Frame-level route-trigger accuracy, non-landmark rejection accuracy, landmark-window frame accuracy, longest incorrect streak, qualitative examples	The deployed question is whether visual anchors can be recovered without destabilizing non-landmark stretches.

Table 8: Binary positive/negative contingency for the annotation-agent audit

Setting	Human positive	Human negative	Predicted positive	Predicted negative	Binary TP / FN / FP / TN
strict exact	181	68	146	103	132 / 49 / 14 / 54
relaxed exact+non-exact	212	37	194	55	179 / 33 / 15 / 22

Table 8 supports negative recall and unweighted Cohen's kappa. Table 3 uses stricter satellite-ID correctness, so its true-positive counts are lower when the agent predicts a match but selects the wrong satellite object.

Appendix C Final Protocol Summary

Table 9: Compact summary of the final experimental protocol

Protocol element	Final instantiation	Operational role
Weak-label source	Two-stage GPT-5.2 annotation agent; max_tools = 10; exact automatic matches used as primary positives; human golden subset disjoint	Scalable building-to-building supervision
Split supports	Train 126, validation 20, ID 24, OOD 30, human golden 10 frames	Fixed supports for development and audit
Base matcher pool supports	Validation 243+243; ID 202+184; OOD 195+195; golden 168+168 present/no-match pools	Pool-level evaluation before route deployment
Final training support	First 12 six-frame route chunks from train pool (72 frames)	Training support for deployed matcher
Final landmark bank	epsilon_cover = 20, epsilon_exclude = 5, delta = 0.7, tau = 0.6, reference-area filter ¿= Q1, ±3-frame event-window consolidation; 55 landmarks before final replay filtering	Fixed schedule for retrieval/calibration testing
Final calibrated replay	Exclude landmarks whose windows touch training; evaluate 47 eligible events and 846 non-landmark frames	Route-level visual-trigger audit
Event semantics	Recovered if accepted within [t-3,t+3]; previous landmark active through t+3	Route-usable reacquisition rather than exact-frame detection
Calibration	B1 = 5; four route blocks; deployed threshold is median of fold-optimal thresholds	Conservative operating point

Table 10: Building-segmentation backbone and preprocessing configuration

Item	Configuration
Detector / encoder	Ultralytics YOLO11n-seg-family checkpoint fine-tuned for building instance segmentation
Training corpus	Roboflow Universe Buildings Instance Segmentation dataset
VPAIR adaptation	No additional VPAIR fine-tuning
Training resolution	640 x 640
Training schedule	100 epochs
Optimizer	AdamW, initial learning rate 1e-3, weight decay 0.05
Loss weights	segmentation 7.5, box 7.5, DFL 1.5, classification 0.5
Class mapping	Single foreground class: building
Inference input	VPAIR images resized/letterboxed to YOLO input size
Preprocessing/postprocessing	Ultralytics defaults; no project-specific NMS, IoU, confidence, or mask filtering overrides
Retained metadata	image id, bounding box, segmentation confidence, per-view building id

Item	Configuration
Embedding extraction	Pooled activations from 11 hooked layers; full embedding 1,584-D; deployed subset [6,9,10] is 640-D

C.1 External-baseline resource arithmetic

The descriptor-storage values in Table 6 use a float32 descriptor map and the VPAIR route density. Dense image-level VPR baselines store one descriptor per reference frame, giving

$$N_{\text{dense},100} = \frac{2706}{106.2} \times 100 \approx 2548 \tag{C1}$$

stored descriptors per 100 km. CLARA stores only scheduled landmarks. Using the observed urban-frame fraction gives

$$L_{\text{urban}} = \frac{1234}{2706} \times 106.2 \approx 48.43 \text{ km}, \quad N_{\text{CLARA},100} = \frac{55}{48.43} \times 100 \approx 114. \tag{C2}$$

For descriptor dimension D, the route-map descriptor memory is

$$M_{100\text{km}} = \frac{N_{100} D \cdot 4}{1024^2} \text{ MiB}. \tag{C3}$$

Table 11: Descriptor-storage arithmetic for the same-route comparison

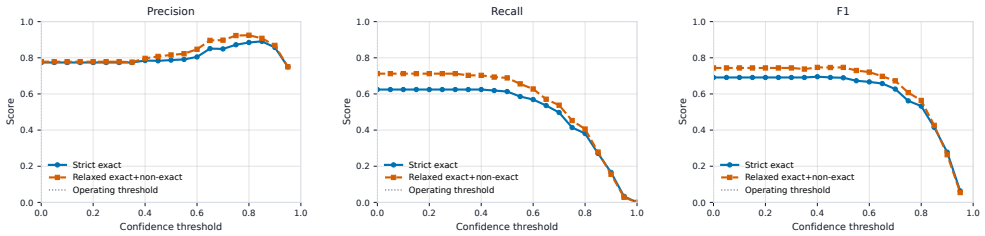
Method	Stored descriptors / 100 km	Descriptor dimension	FP32 memory / 100 km
CLARA	114	640	0.28 MiB
CosPlace ResNet101-2048	2,548	2,048	19.9 MiB
AnyLoc-VLAD-DINOv2	2,548	49,152	477.8 MiB

As a storage-only sensitivity check, if the dense route gallery were idealized as storing every seventh reference frame, the dense map would still require about 364 descriptors per 100 km. Under this aggressive downsampling, CosPlace would require about 2.84 MiB and AnyLoc-VLAD-DINOv2 about 68.25 MiB per 100 km, compared with 0.28 MiB for CLARA. This calculation is not used for the reported accuracy values; it only shows that the descriptor-memory advantage is not an artifact of storing every VPAIR reference frame.

Table 12: Deployment-relevant model-scale summary for the direct baselines

Method	Backbone / head	Approximate online neural parameters	Edge-deployment interpretation
CLARA	YOLO11n-seg-family building segmenter plus GBDT acceptance head	∼2.9M for the segmentation backbone; small non-neural matching head	Edge-computing-oriented design based on compact model scale and Raspberry Pi 5 deployment constraints; final throughput still requires target-hardware validation
CosPlace ResNet101-2048	ResNet101-GeM-FC descriptor model	∼46.7M	Plausible as a low-rate or optimized edge descriptor baseline, but the released checkpoint transfers poorly to this replay task
AnyLoc-VLAD-DINOv2	DINOv2 ViT-G/14 features with VLAD aggregation	∼1.1B for the ViT-G backbone	Strong dense baseline, but standard ViT-G descriptor extraction is not practical for onboard inference under Raspberry Pi 5 deployment constraints

Appendix D Annotation-Agent Diagnostics



(a) Precision-threshold curve (b) Recall-threshold curve (c) F1-threshold curve

Fig. 7: Annotation-agent confidence sweeps: (a) precision, (b) recall, and (c) F1 for strict exact and relaxed exact+non-exact scoring

Figure 7a shows that stricter confidence thresholds mostly improve precision at the cost of retained positives. Figure 7b shows the corresponding recall decline, and Figure 7c summarizes the resulting F1 trade-off. The default operating point for the main AI-agent-generated weak-label corpus is threshold 0.0 because final label quality is later controlled through confidence-filtered ablations and validation-side deployment calibration.

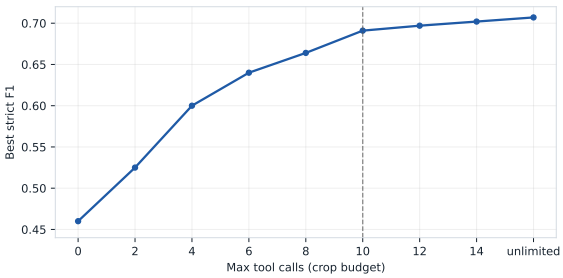
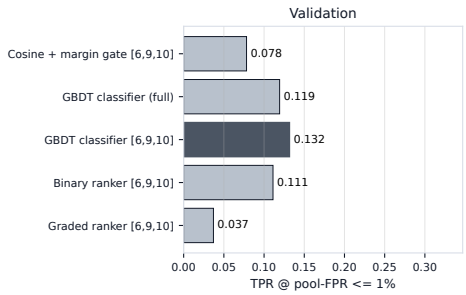


Fig. 8: Annotation-agent tool-call budget. Strict exact F1 improves with crop-inspection budget and shows diminishing returns beyond 10 calls

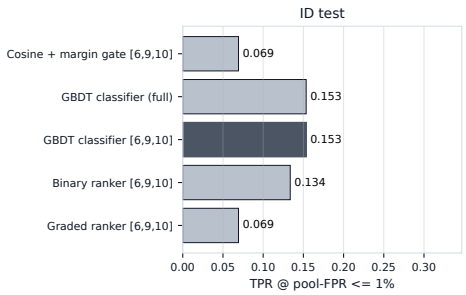
Table 13: Annotation-agent footprint over successful traces

Successful traces	Mean total tokens	Mean input tokens	Mean output tokens	Mean crop calls
210	85,623	81,951	3,672	9.3

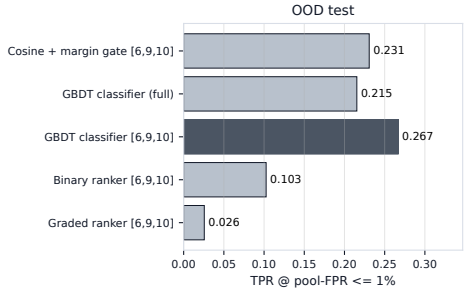
Appendix E Split-Specific Matcher Panels



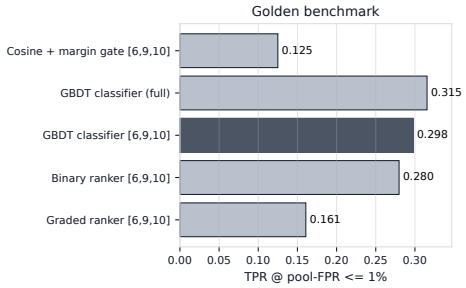
(a) Validation split



(b) ID test split



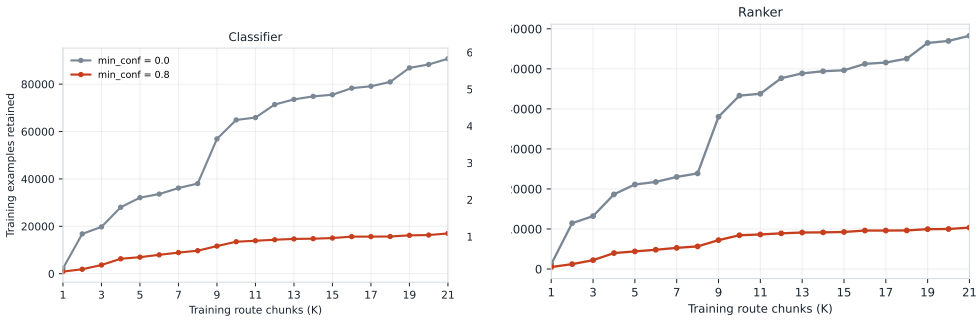
(c) OOD test split



(d) Human golden benchmark

Fig. 9: Split-specific alternatives for the matcher overview: (a) validation, (b) ID test, (c) OOD test, and (d) human golden benchmark

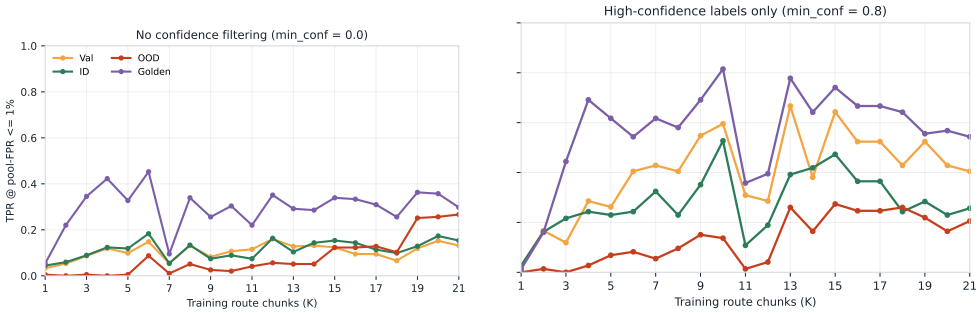
Appendix F Label-Quality and Train-Size Diagnostics



(a) Classifier training examples

(b) Ranker training examples

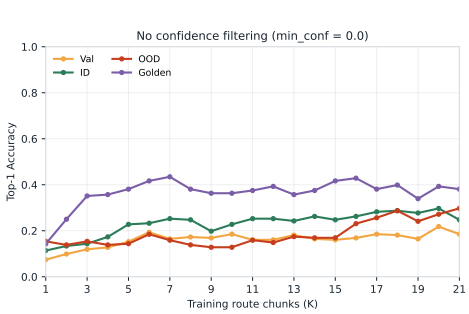
Fig. 10: Training examples retained as route chunks increase: (a) classifier and (b) ranker



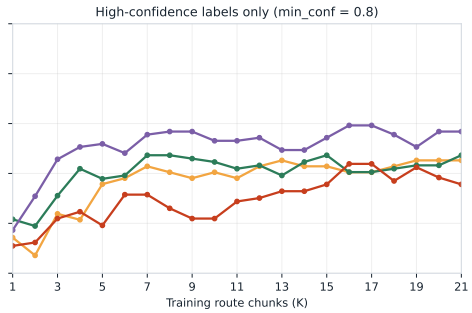
(a) Classifier TPR without confidence filtering

(b) Classifier TPR with high-confidence labels

Fig. 11: Classifier train-size behavior under the low-FPR metric: (a) default AI-agent-generated weak labels and (b) high-confidence AI-agent-generated weak labels

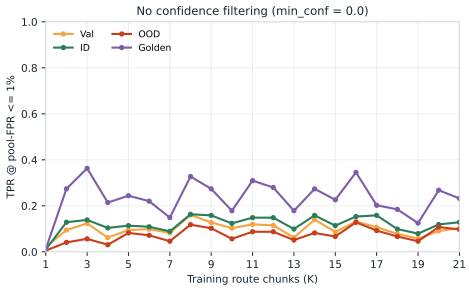


(a) Classifier Top-1 without confidence filtering

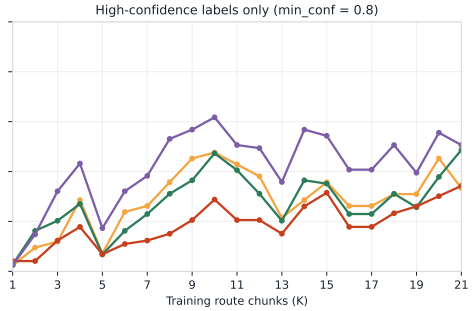


(b) Classifier Top-1 with high-confidence labels

Fig. 12: Classifier retrieval quality: (a) default AI-agent-generated weak labels and (b) high-confidence AI-agent-generated weak labels

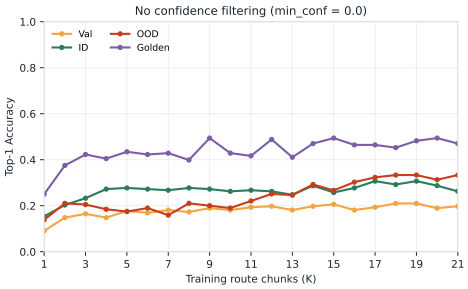


(a) Ranker TPR without confidence filtering

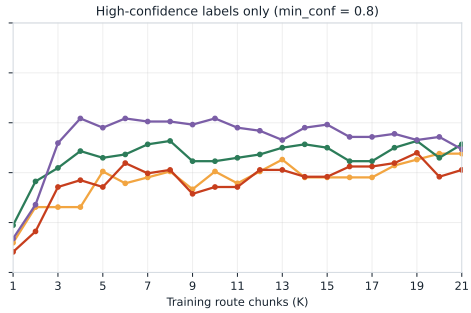


(b) Ranker TPR with high-confidence labels

Fig. 13: Ranker train-size behavior under the low-FPR metric: (a) default AI-agent-generated weak labels and (b) high-confidence AI-agent-generated weak labels



(a) Ranker Top-1 without confidence filtering



(b) Ranker Top-1 with high-confidence labels

Fig. 14: Ranker retrieval quality: (a) default AI-agent-generated weak labels and (b) high-confidence AI-agent-generated weak labels

Table 14: Full-K transfer summary for AI-agent-generated weak-label regimes. Entries are Top-1 / TPR@pool-FPR¡=1% / FPR

Head	min_conf	Train positives	Train pairs	Val	ID	OOD	Golden
Classifier	0.0	1,623	90,724	0.185 / 0.132 / 0.008	0.248 / 0.153 / 0.000	0.297 / 0.267 / 0.041	0.381 / 0.298 / 0.024
Classifier	0.8	332	16,982	0.452 / 0.405 / 0.000	0.473 / 0.257 / 0.000	0.356 / 0.205 / 0.041	0.568 / 0.543 / 0.037
Ranker	0.0	1,620	58,261	0.198 / 0.103 / 0.008	0.262 / 0.129 / 0.005	0.333 / 0.097 / 0.021	0.470 / 0.232 / 0.006
Ranker	0.8	332	10,342	0.476 / 0.333 / 0.000	0.514 / 0.486 / 0.060	0.411 / 0.342 / 0.151	0.494 / 0.506 / 0.111

Appendix G Qualitative Route Examples



Fig. 15: Representative successful detections: (a) dense scene with strong appearance corruption; (b) partially visible unusual landmark; (c) cross-view segmentation inconsistency with preserved dominant geometry; (d) repeated views where a later frame crosses the acceptance threshold



Fig. 16: Representative residual failures: (a) threshold-sensitive late-window views; (b) weak small landmark; (c) landmark absent from compatible online detections; (d) cross-view mismatch from merged small structures

Appendix H Annotation-Agent Prompt and Schema

For reproducibility, this appendix records the inspection-phase prompt and structured output schema. Inside the annotation tool, the labels "drone" and "satellite" denote the first and second input images, corresponding here to the aircraft-borne aerial query image and the satellite-style reference render.

Table 15: Annotation-agent implementation details

Parameter	Value
Inspection model	GPT-5.2
Inspection reasoning effort	low
Reporting model	GPT-5.2
Reporting reasoning effort	medium
Temperature	1
Maximum crop/tool calls	10 per image pair; final report forced at the limit
Tool calling	Sequential tool calls only; parallel tool calls disabled

Parameter	Value
Crop tool	Relative crop coordinates in [0,100], integer zoom factor, optional rotation in {-90, 90, 180}
Final output	Strict Building2BuildingAnnotations Pydantic schema
Retry policy	One retry for model calls
Execution limit	Graph recursion limit 50
Trace handling	Full prompt, configuration, messages, inputs, raw output, and parsed schema output saved per image pair
Dataset inclusion	Only schema-parsed outputs were used as AI-agent-generated weak labels

Inspection-phase prompt

```
You are a building-to-building annotation agent for the VPAIR dataset.

Inputs:
- The user message contains two images with building bounding boxes and
    building ID labels
- Drone image is ALWAYS first, satellite image is ALWAYS second

Goal:
For every building detected with bounding box and with visible ID in the drone
     image, decide which satellite building(s) correspond to it.
You must match the buildings by their visual representation, not IDs - because
     IDs will likely not match.
Return the result as JSON that conforms exactly to the
    Building2BuildingAnnotations schema.

Exact vs non-exact matching rules:
- An exact match is ONLY when the drone building corresponds to a single
    satellite building with the same footprint.
- If segmentation differs, treat as non-exact. Common cases:
  - One view segments a single building while the other segments multiple
    buildings in the same place.
  - One view segments only a part of the building, or merges multiple
    buildings into one.
  - Nested segments (a part of a larger building becomes a separate building).
- For exact matches, fill exact_matching_satellite_building.
    satellite_building_id with the satellite ID.
- If no exact match exists, set exact_matching_satellite_building.
    satellite_building_id to null.
- If ANY non-exact correspondence exists, list them in
    non_exact_matching_satellite_buildings; otherwise set it to null.
- If, for a particular segmented building from drone image its ID is not
    visible, skip this building.
- If the ID of the matching building from the satellite is not visible, put "
    NOT_VISIBLE" in the ‘satellite_building_id‘ field.
- Provide confidence in [0, 1] for each Match; lower it when unsure.

Note, satellite and drone images are already aligned to have almost the same
    scale of buildings and cover almost the same territory.
However, the drone image may be rotated relatively to the satellite image,
    thus you can rotate the images if needed to align their orientation.

Add small summarising notes and observations before calling the tool.

To help you accomplish the task, here is the tool for use:
If details are too small to read, use crop_rectangle_and_scale_it to zoom into
     specific regions.
```

```
- image_source: "drone" or "satellite"
- Coordinates are relative 0..100 for the chosen image
- rotate can be -90, 90, or 180; negative values mean counter-clockwise
Use only one tool call at a time.
Follow the exploration of the drone crop with the exploration of the
    corresponding satellite crop.
Add concise analysis notes for the previous tool call and next intended action
     each time before calling this tool.
You will have only at most 10 tool calls available, so use them wisely.
After you have explored in details all unclear parts and are ready to produce
    the full annotation list, call make_the_final_annotation_report instead of
     outputting JSON directly.
```

Structured output schema

```
from pydantic import BaseModel, Field
from typing import Literal


class Match(BaseModel):
    satellite_building_id: int | Literal["NOT_VISIBLE"] | None = Field(
        description="None␣if␣no␣match␣found."
    )
    confidence: float = Field(
        description="Must␣be␣within␣0␣to␣1␣range.␣Set␣how␣confident␣are␣you␣in
    ␣your␣matching␣decision."
    )


class DroneBuildingAnnotation(BaseModel):
    drone_building_id: int
    exact_matching_satellite_building: Match
    non_exact_matching_satellite_buildings: list[Match] | None


class Building2BuildingAnnotations(BaseModel):
    annotations: list[DroneBuildingAnnotation]
```

References

[1] Svystun, S., Scislo, L., Pawlik, M., Melnychenko, O., Radiuk, P., Savenko, O., Sachenko, A.: DyTAM: Accelerating wind turbine inspections with dynamic UAV trajectory adaptation. Energies **18**(7), 1823 (2025) https://doi.org/10.3390/en18071823

[2] Lysyi, A., Sachenko, A., Radiuk, P., Lysyi, M., Melnychenko, O., Ishchuk, O., Savenko, O.: Enhanced fire hazard detection in solar power plants: An integrated UAV, AI, and SCADA-based approach. Radioelectronic and Computer Systems **2025**(2), 99–117 (2025) https://doi.org/10.32620/reks.2025.2.06

[3] Gyagenda, N., Hatilima, J.V., Roth, H., Zhmud, V.: A review of GNSS-independent UAV navigation techniques. Robotics and Autonomous Systems **152**, 104069 (2022) https://doi.org/10.1016/j.robot.2022.104069

[4] Chang, Y., Cheng, Y., Manzoor, U., Murray, J.: A review of UAV autonomous navigation in GPS-denied environments. Robotics and Autonomous Systems **170**, 104533 (2023) https://doi.org/10.1016/j.robot.2023.104533

[5] Couturier, A., Akhloufi, M.A.: A review on absolute visual localization for UAV. Robotics and Autonomous Systems **135**, 103666 (2021) https://doi.org/10.1016/j.robot.2020.103666

[6] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics **32**(6), 1309–1332 (2016) https://doi.org/10.1109/TRO.2016.2624754

[7] Qin, T., Li, P., Shen, S.: VINS-Mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics **34**(4), 1004–1020 (2018) https://doi.org/10.1109/TRO.2018.2853729

[8] Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. IEEE Transactions on Robotics **37**(6), 1874–1890 (2021) https://doi.org/10.1109/TRO.2021.3075644

[9] Lowry, S., Sünderhauf, N., Newman, P., Leonard, J.J., Cox, D., Corke, P., Milford, M.J.: Visual place recognition: A survey. IEEE Transactions on Robotics **32**(1), 1–19 (2016) https://doi.org/10.1109/TRO.2015.2496823

[10] Qu, X., Soheilian, B., Paparoditis, N.: Landmark based localization in urban environment. ISPRS Journal of Photogrammetry and Remote Sensing **140**, 90–103 (2018) https://doi.org/10.1016/j.isprsjprs.2017.09.010

[11] Lin, T.-Y., Belongie, S., Hays, J.: Cross-view image geolocalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 891–898 (2013). https://doi.org/10.1109/CVPR.2013.120

[12] Schleiss, M., Rouatbi, F., Cremers, D.: VPAIR: Aerial visual place recognition and localization in large-scale outdoor environments. arXiv preprint arXiv:2205.11567 (2022). https://doi.org/10.48550/arXiv.2205.11567

[13] Wilson, D., Zhang, X., Sultani, W., Wshah, S.: Image and object geo-localization. International Journal of Computer Vision **132**(4), 1350–1392 (2024) https://doi.org/10.1007/s11263-023-01942-3

[14] Sun, Y., Ye, Y., Kang, J., Fernandez-Beltran, R., Feng, S., Li, X., Luo, C., Zhang, P., Plaza, A.: Cross-view object geo-localization in a local region with satellite imagery. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–16 (2023) https://doi.org/10.1109/TGRS.2023.3307508

[15] Ostrovskyi, Z., Barmak, O., Radiuk, P., Krak, I.: Unsupervised knowledge extraction of distinctive landmarks from Earth imagery using deep feature outliers for robust UAV geo-localization. Machine Learning and Knowledge Extraction **7**(3), 81 (2025) https://doi.org/10.3390/make7030081

[16] Borovyk, D., Barmak, O., Radiuk, P., Krak, I.: Hierarchical deep learning model for identifying similar targets in UAV imagery. Drones **9**(11), 743 (2025) https://doi.org/10.3390/drones9110743

[17] Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Wu, S., Ré, C.: Snorkel: Rapid training data creation with weak supervision. The VLDB Journal **29**(2), 709–730 (2020) https://doi.org/10.1007/s00778-019-00552-1

[18] Chen, Y., Zheng, B., Zhang, Z., Wang, Q., Shen, C., Zhang, Q.: Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. ACM Computing Surveys **53**(4), 84–18437 (2020) https://doi.org/10.1145/3398209

[19] Zheng, Z., Wei, Y., Yang, Y.: University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In: Proceedings of the 28th ACM International Conference on Multimedia (ACM MM), pp. 1395–1403. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3394171.3413896

[20] Cisneros, I., Yin, P., Zhang, J., Choset, H., Scherer, S.: ALTO: A large-scale dataset for UAV visual place recognition and localization. arXiv preprint arXiv:2207.12317 (2022). https://doi.org/10.48550/arXiv.2207.12317

[21] Ding, L., Zhou, J., Meng, L., Long, Z.: A practical cross-view image matching method between UAV and satellite for UAV-based geo-localization. Remote Sensing **13**(1), 47 (2021) https://doi.org/10.3390/rs13010047

[22] Zhu, S., Shah, M., Chen, C.: TransGeo: Transformer is all you need for cross-view image geo-localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1162–1171 (2022). https://doi.org/10.1109/CVPR52688.2022.00123

[23] Deuser, F., Habel, K., Oswald, N.: Sample4Geo: Hard negative sampling for cross-view geo-localisation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16847–16856 (2023). https://doi.org/10.1109/ICCV51070.2023.01545

[24] Lin, J., Zheng, Z., Zhong, Z., Luo, Z., Li, S., Yang, Y., Sebe, N.: Joint representation learning and keypoint detection for cross-view geo-localization. IEEE Transactions on Image Processing **31**, 3780–3792 (2022) https://doi.org/10.1109/TIP.2022.3175601

[25] Wang, T., Zheng, Z., Yan, C., Zhang, J., Sun, Y., Zheng, B., Yang, Y.: Each part matters: Local patterns facilitate cross-view geo-localization. IEEE Transactions on Circuits and Systems for Video Technology **32**(2), 867–879 (2022) https://doi.org/10.1109/TCSVT.2021.3061265

[26] Ye, Q., Luo, J., Lin, Y.: A coarse-to-fine visual geo-localization method for GNSS-denied UAV with oblique-view imagery. ISPRS Journal of Photogrammetry and Remote Sensing **212**, 306–322 (2024) https://doi.org/10.1016/j.isprsjprs.2024.05.006

[27] Wu, Q., Wan, Y., Zheng, Z., Zhang, Y., Wang, G., Zhao, Z.: CAMP: A cross-view geo-localization method using contrastive attributes mining and position-aware partitioning. IEEE Transactions on Geoscience and Remote Sensing **62**, 1–14 (2024) https://doi.org/10.1109/TGRS.2024.3448499

[28] Xia, P., Yu, L., Wan, Y., Wu, Q., Chen, P., Zhong, L., Yao, Y., Wei, D., Liu, X., Ru, L., Zhang, Y., Lao, J., Chen, J., Yang, M., Zhang, Y.: Cross-view geo-localization with panoramic street-view and VHR satellite imagery in decentrality settings. ISPRS Journal of Photogrammetry and Remote Sensing **227**, 1–11 (2025) https://doi.org/10.1016/j.isprsjprs.2025.04.033

[29] Zeng, Q., Wu, J., Ren, Y., Feng, G.: Cross-view geolocation via segmentation and common region feature matching. ISPRS Journal of Photogrammetry and Remote Sensing **227**, 804–816 (2025) https://doi.org/10.1016/j.isprsjprs.2025.06.020

[30] Huang, Z., Aryal, J., Nahavandi, S., Lu, X., Lim, C.P., Wei, L., Zhou, H.: Object-level cross-view geo-localization with location enhancement and multi-head cross attention. arXiv preprint arXiv:2505.17911 (2025). https://doi.org/10.48550/arXiv.2505.17911

[31] Zhou, Z.-H.: A brief introduction to weakly supervised learning. National Science Review **5**(1), 44–53 (2018) https://doi.org/10.1093/nsr/nwx106

[32] Song, H., Kim, M., Park, D., Shin, Y., Lee, J.-G.: Learning from noisy labels with deep neural networks: A survey. IEEE Transactions on Neural Networks and Learning Systems **34**(11), 8135–8153 (2023) https://doi.org/10.1109/TNNLS.2022.3152527

[33] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017). https://doi.org/10.48550/arXiv.1704.04861

[34] Tan, M., Le, Q.V.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114. PMLR, Long Beach, CA, USA (2019).

https://proceedings.mlr.press/v97/tan19a.html

[35] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 3149–3157 (2017). https://dl.acm.org/doi/abs/10.5555/3294996.3295074

[36] Ultralytics: Ultralytics YOLO11 model documentation (2024). https://docs.ultralytics.com/models/yolo11/

[37] Ultralytics: Quick start guide: Raspberry Pi with Ultralytics YOLO26 (2026). https://docs.ultralytics.com/guides/raspberry-pi/

[38] Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P.R., Salazar, G., Ryoo, M.S., Reymann, K., Rao, K., Pertsch, K., Mordatch, I., Michalewski, H., Lu, Y., Levine, S., Lee, L., Lee, T.-W.E., Leal, I., Kuang, Y., Kalashnikov, D., Julian, R., Joshi, N.J., Irpan, A., Ichter, B.: RT-2: Vision-language-action models transfer web knowledge to robotic control. In: Proceedings of the 7th Conference on Robot Learning (CoRL). Proceedings of Machine Learning Research, vol. 229, pp. 2165–2183. PMLR, Atlanta, GA, USA (2023). https://proceedings.mlr.press/v229/zitkovich23a.html

[39] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R.B.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3992–4003 (2023). https://doi.org/10.1109/ICCV51070.2023.00371

[40] Keetha, N., Mishra, A., Karhade, J., Jatavallabhula, K.M., Scherer, S., Krishna, M., Garg, S.: AnyLoc: Towards universal visual place recognition. IEEE Robotics and Automation Letters **9**(2), 1286–1293 (2024) https://doi.org/10.1109/LRA.2023.3343602

[41] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2024). https://doi.org/10.48550/arXiv.2304.07193

[42] Berton, G., Masone, C., Caputo, B.: Rethinking visual geo-localization for large-scale applications. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4868–4878 (2022). https://doi.org/10.1109/CVPR52688.2022.00483