VISUAL MEMORY MEETS GEOMETRIC FILTERING FOR GPS-DENIED UAV LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Localizing an unmanned aerial vehicle (UAV) against a georeferenced satellite map can provide a position estimate when GPS is unavailable. A single aerial observation, however, may match multiple nearby map locations. We present FieldAnchor, a reference-conditioned localization network that integrates evidence at three time scales. A gated recurrent unit (GRU) retains visual information across observations; a Kalman estimator propagates position and uncertainty between observations; and Mean Shift converts cross-view responses into continuous position estimates. The network first extracts a visual measurement with Mean Shift, fuses it with the recurrent features and predicted geometric state, and then applies a second Mean Shift step to refine the updated estimate. On 1,482 observations from eight held-out Bearing-UAV-90K routes, the full system achieves a mean localization error of 3.76 m and a 41.90% success rate within 3 m. Ablations show increased error when recurrent processing or the final refinement stage is removed. Evaluation assumes that a georeferenced map neighborhood is provided.

1 INTRODUCTION

An unmanned aerial vehicle (UAV) can lose reliable Global Positioning System (GPS) reception near structures, under interference, or during deliberate signal denial. Matching an onboard image to a georeferenced satellite map offers a visual route to geographic position (Zheng et al., 2020; Liu et al., 2026). The match is difficult because aerial and satellite images are captured from different sensors and differ in viewpoint, scale, and appearance. Satellite crops also overlap, so several nearby candidates can receive plausible visual scores even when only one location is correct.

Existing cross-view methods often make a decision from one observation, such as selecting a map tile or regressing a position from local satellite context. Bearing-UAV extends beyond tile matching by predicting position and heading from neighboring aerial and satellite features (Liu et al., 2026). Sequential observations provide another source of evidence: a building or road configuration that is ambiguous now may be recognizable from the path of earlier views (Zhang et al., 2023). At the same time, a sequence descriptor alone does not specify how the current image response should be converted into a precise coordinate.

We address these roles with three successive stages. First, a gated recurrent unit (GRU) retains *long-horizon visual context* beyond the three descriptors explicitly supplied at each step. Second, a Kalman estimator carries *medium-horizon geometric state* and uncertainty as the visual measurements change along the route. Third, Mean Shift provides *short-horizon spatial refinement* by following the current response over neighboring satellite candidates. A front Mean Shift decoder converts initial candidate similarities into a visual measurement; the final decoder works on a newly scored neighborhood centered by the geometric estimate. This ordering lets earlier appearance inform the current estimate while leaving the final position sensitive to the present image.

FieldAnchor uses a shared image backbone, separate view projectors, six projected GRU inputs, and four readout heads for correction, measurement variance, motion, and heading. We evaluate one shared model on route-ordered observations from four Bearing-UAV-90K cities. Each evaluation observation comes with a local map reference and a path reference; the experiments therefore measure refinement under those supplied inputs. Under this protocol, the complete system obtains 3.76 m mean localization error. Removing the GRU raises the error to 4.28 m, while removing the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

complete final refinement stage raises it to 6.79 m. We further examine the temporal input length and the spatial extent of the final candidate neighborhood.

2 Related Work

Cross-view geo-localization. Cross-view localization relates UAV imagery to georeferenced aerial or satellite imagery. University-1652, SUES-200, DenseUAV, and GTA-UAV represent retrieval and matching approaches evaluated in different aerial settings (Zheng et al., 2020; Zhu et al., 2023; Dai et al., 2024; Ji et al., 2025). VIGOR studies images that are not centered on their reference tiles (Zhu et al., 2021); Sample4Geo improves cross-view representations with hard-negative sampling (Deuser et al., 2023). Bearing-UAV uses neighboring satellite features to regress position and heading under view misalignment (Liu et al., 2026). Our setting starts from a supplied local reference and examines temporal inference and spatial decoding within that neighborhood.

Visual sequences and geometric estimation. SeqSLAM shows that matching an appearance sequence can resolve ambiguities that remain in individual observations (Milford & Wyeth, 2012). Learned sequence representations also aggregate visual evidence for place recognition and cross-view localization (Facil et al., 2019; Zhang et al., 2023). A recurrent state can retain past appearance beyond a fixed input window. In a separate line of work, Kalman filtering recursively predicts geometric state and corrects it with uncertain measurements (Kalman, 1960); Recurrent Kalman Networks and KalmanNet combine learned representations with structured filtering (Becker et al., 2019; Revach et al., 2022). These ideas motivate the distinct visual-memory and geometric-state stages of our model.

Mean Shift and spatial responses. Mean Shift follows weighted samples toward local response modes. Hsieh et al. (2024) use a Mean Shift procedure to smooth a differentiable architecture-search landscape. Here the shifted variables are map coordinates: candidate centers and cross-view similarity weights define a spatial response from which a continuous location is decoded. The first decoding produces a visual measurement; the second refines the position after temporal estimation.

3 METHOD

Problem setting and overview. At sequence index t, the input UAV image is denoted by I_t, and the georeferenced satellite map is denoted by $\mathcal{M}$. A supplied local reference $\mathbf{r}_t \in \mathbb{R}^2$ selects a neighborhood in map coordinates. The method estimates the current map position $\widehat{\mathbf{x}}_t \in \mathbb{R}^2$ using observations through index t.

Figure 1 shows the processing order. Cross-view similarity supplies response-weighted satellite context and a visual position decoded by front Mean Shift. The GRU receives these quantities and the three-observation appearance features, then produces correction, variance, motion, and heading through four heads. The Kalman stage updates the geometric state. Final Mean Shift decodes a response around its estimated position.

Local cross-view evidence. The local reference $\mathbf{r}_t$ selects the candidate set $\mathcal{A}_t = \{(S_{t,i}, \mathbf{c}_{t,i})\}_{i=1}^{N}$, where $S_{t,i}$ is the ith satellite crop, $\mathbf{c}_{t,i} \in \mathbb{R}^2$ is its map center, and $N = 18$ is the number of scored front candidates. These crops occupy the forward half of a 6×6 lattice aligned with the supplied path reference. Only the retained crops are encoded and scored.

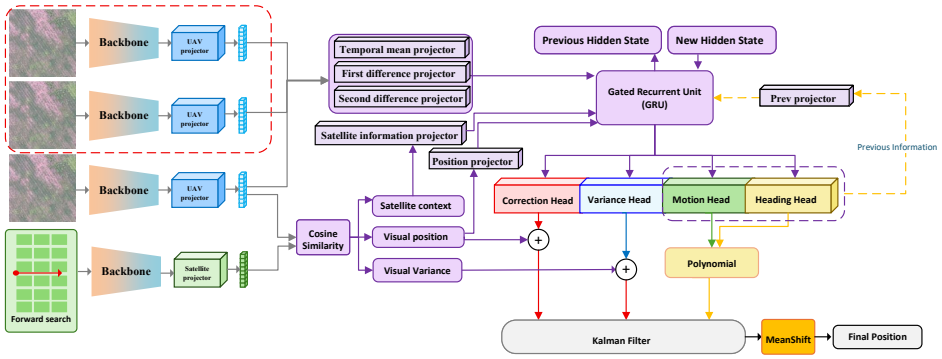

Figure 1: Overview of FieldAnchor. Three UAV observations and nearby satellite candidates provide visual evidence to the GRU. Its four heads supply quantities for geometric state estimation, followed by Mean Shift position refinement.

Visual encoder. Figure 2 shows the shared backbone and the view-specific projection heads.

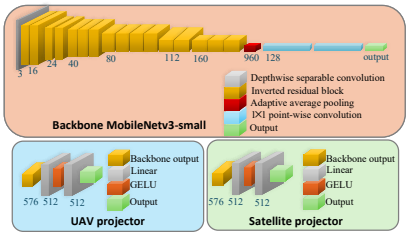

Figure 2: Shared MobileNetV3-Small backbone and view-specific projectors. UAV and satellite features pass through separate projection heads before cosine similarity.

The shared backbone f produces a 576-dimensional feature, placing both views on a common feature basis. The UAV projector g_u and satellite projector g_s account for the appearance difference with independent weights but the same $576 \to 512 \to 512$ structure and a Gaussian Error Linear Unit (GELU) between their linear layers. For any nonzero vector $\mathbf{a}$, define $\mathrm{n}(\mathbf{a}) = \mathbf{a}/\|\mathbf{a}\|_2$. The normalized UAV descriptor $\mathbf{u}_t$ and satellite descriptor $\mathbf{v}_{t,i}$ are

$$\mathbf{u}_t = \mathrm{n}(g_u(f(I_t))), \qquad \mathbf{v}_{t,i} = \mathrm{n}(g_s(f(S_{t,i}))). \tag{1}$$

Cosine similarity converts these descriptors into a response over the candidate set. For candidate i, its similarity $s_{t,i}$ and normalized response weight $p_{t,i}^{\mathrm{vis}}$ are

$$s_{t,i} = \mathbf{u}_t^\top \mathbf{v}_{t,i}, \qquad p_{t,i}^{\mathrm{vis}} = \frac{\exp(s_{t,i})}{\sum_{j=1}^{N} \exp(s_{t,j})}, \tag{2}$$

where the nonnegative weights $p_{t,i}^{\mathrm{vis}}$ sum to one and define the response mass used by Mean Shift.

After similarity scoring, front Mean Shift converts the response over candidate centers into one continuous visual position $\mathbf{x}_t^v$. This is the visual measurement supplied to the GRU and Kalman stages. Figure 3 illustrates seed updates and mode support for the 18 forward candidates; final Mean Shift later applies the same spatial principle after temporal estimation.

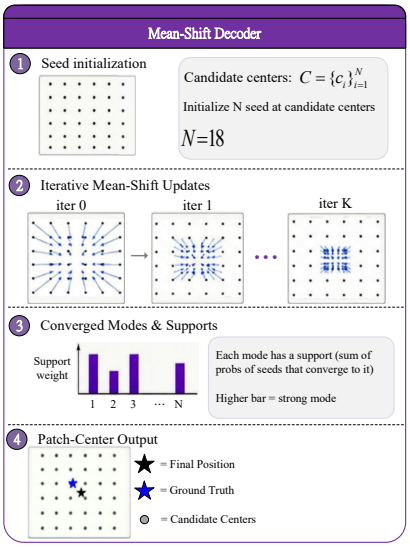

Figure 3: Mean Shift decoding of the forward 18-candidate response. Seeds start at candidate centers, move toward local response modes, and receive support according to the mass converging to each mode. The support-weighted modes yield a continuous position.

The decoder begins with one seed at each candidate center. For seed j, its position at iteration k is $\boldsymbol{\mu}_{t,j}^{(k)} \in \mathbb{R}^2$, initialized by $\boldsymbol{\mu}_{t,j}^{(0)} = \mathbf{c}_{t,j}$. The positive bandwidth h sets the spatial reach of the Gaussian kernel. Using the response masses $p_{t,i}^{\text{vis}}$ already defined in Equation (2), a Mean Shift update is

$$\boldsymbol{\mu}_{t,j}^{(k+1)} = \frac{\sum_{i=1}^{N} p_{t,i}^{\text{vis}} \exp\left[-\left\|\boldsymbol{\mu}_{t,j}^{(k)} - \mathbf{c}_{t,i}\right\|_2^2/(2h^2)\right] \mathbf{c}_{t,i}}{\sum_{i=1}^{N} p_{t,i}^{\text{vis}} \exp\left[-\left\|\boldsymbol{\mu}_{t,j}^{(k)} - \mathbf{c}_{t,i}\right\|_2^2/(2h^2)\right)}, \tag{3}$$

where the numerator computes the response-weighted center of nearby candidates and the denominator normalizes their mass. Seeds initialized at different centers may converge to the same response mode. Each converged mode receives support equal to the response mass of its associated seeds, and the support-weighted average of the modes defines the front visual position $\mathbf{x}_t^v$. This stage preserves the local geometry of the candidate response while yielding a continuous position instead of a discrete crop index.

The same response also supplies satellite information to the temporal stage. The front Mean Shift position $\mathbf{x}_t^v$ carries spatial information, while the response-weighted satellite descriptor $\mathbf{c}_t^v$ carries the appearance associated with the current response. Both quantities are passed to the GRU rather than relying on a single top-scoring crop. With $p_{t,i}^{\text{vis}}$ denoting the normalized similarity of candidate i in Equation (2), the satellite descriptor is

$$\mathbf{c}_t^v = \sum_{i=1}^{N} p_{t,i}^{\text{vis}} \mathbf{v}_{t,i}. \tag{4}$$

GRU input and recurrent state. Figure 4 gives the complete GRU input structure.

The current UAV descriptor $\mathbf{u}_t$ and the descriptors from the two preceding observations, $\mathbf{u}_{t-1}$ and $\mathbf{u}_{t-2}$, form a three-observation window. Its temporal mean $\bar{\mathbf{u}}_t$, first difference $\Delta\mathbf{u}_t$, and second difference $\Delta^2\mathbf{u}_t$ are

$$\bar{\mathbf{u}}_t = \frac{\mathbf{u}_t + \mathbf{u}_{t-1} + \mathbf{u}_{t-2}}{3}, \qquad \Delta\mathbf{u}_t = \mathbf{u}_t - \mathbf{u}_{t-1}, \qquad \Delta^2\mathbf{u}_t = \mathbf{u}_t - 2\mathbf{u}_{t-1} + \mathbf{u}_{t-2}, \tag{5}$$

where $\bar{\mathbf{u}}_t$ summarizes appearance over the window and the two differences expose recent change. All three are feature-space quantities, not measured physical velocity or acceleration. Recurrent context is carried separately from one observation to the next by the hidden state.

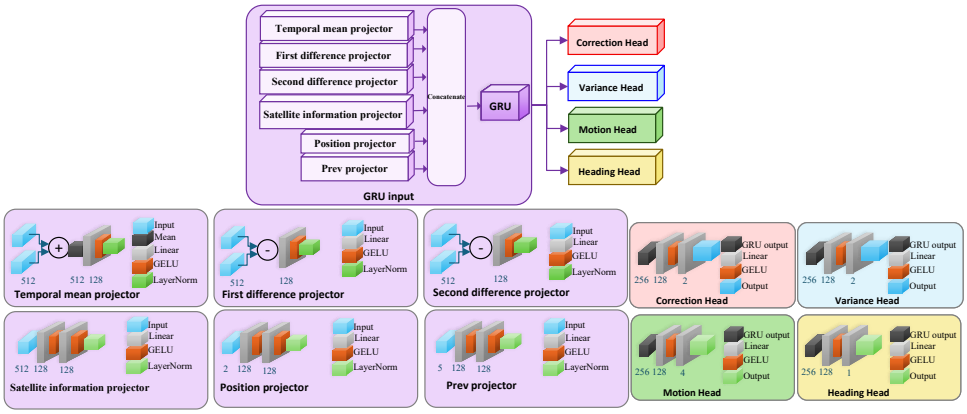

Figure 4: GRU inputs and four readout heads. Six 128-dimensional projected branches are concatenated before the recurrent unit. The lower panels show the projections and the correction, variance, motion, and heading readouts.

The six inputs shown in Figure 4 are the temporal mean, two differences, satellite information $\mathbf{c}_t^v$, the two-dimensional front Mean Shift position, and a five-dimensional summary from the preceding state. Each branch produces a 128-dimensional vector. The temporal mean and differences use a linear projection, GELU, and layer normalization; the satellite, position, and preceding-state branches use the additional linear transformation depicted in the lower panels. Concatenation gives a 768-dimensional vector to a GRU (Cho et al., 2014) with a 256-dimensional hidden state. Four readout heads share this recurrent state and map its visual history to distinct geometric outputs.

Denote the six projectors, in the order drawn in Figure 4, by $E_1, \ldots, E_6$, and denote the five-dimensional preceding-state summary by $\mathbf{g}_{t-1}$. Each projector returns a vector in $\mathbb{R}^{128}$. Only the first three branches use the explicit three-observation window, while $\mathbf{h}_{t-1}$ carries information from earlier observations into the next GRU update. The concatenated input $\mathbf{e}_t \in \mathbb{R}^{768}$ and persistent hidden state $\mathbf{h}_t \in \mathbb{R}^{256}$ obey

$$\begin{aligned} \mathbf{e}_t &= \big[E_1(\bar{\mathbf{u}}_t); E_2(\Delta\mathbf{u}_t); E_3(\Delta^2\mathbf{u}_t); \\ &\qquad E_4(\mathbf{c}_t^v); E_5(\mathbf{x}_t^v); E_6(\mathbf{g}_{t-1})\big], \\ \mathbf{h}_t &= \mathrm{GRU}(\mathbf{e}_t, \mathbf{h}_{t-1}). \end{aligned} \tag{6}$$

Four outputs and geometric state. As shown in Figure 4, the correction, variance, motion, and heading heads share the recurrent state. Each begins with a $256 \to 128$ linear transformation followed by GELU, and their respective output dimensions are 2, 2, 4, and 1. Correction and variance describe the current visual observation, while motion and heading carry geometric change and route-relative orientation to the next observation.

The four head functions are C, V, M, and A in that order. Their outputs are correction $\boldsymbol{\delta}_t \in \mathbb{R}^2$, positive measurement variance $\boldsymbol{\nu}_t \in \mathbb{R}^2_{>0}$, motion $\mathbf{m}_t \in \mathbb{R}^4$, and heading $\phi_t \in \mathbb{R}$. The visual measurement $\mathbf{z}_t$ and its covariance $\mathbf{R}_t$ are

$$\begin{aligned} (\boldsymbol{\delta}_t, \boldsymbol{\nu}_t, \mathbf{m}_t, \phi_t) &= \big(C(\mathbf{h}_t), V(\mathbf{h}_t), M(\mathbf{h}_t), A(\mathbf{h}_t)\big), \\ \mathbf{z}_t = \mathbf{x}_t^v + \boldsymbol{\delta}_t, &\qquad \mathbf{R}_t = \mathrm{diag}(\boldsymbol{\nu}_t). \end{aligned} \tag{7}$$

As summarized in Figure 1, the visual measurement and GRU outputs next enter the Kalman stage. Prediction advances the preceding geometric state using motion information, and the measurement update incorporates the current corrected visual position according to its uncertainty.

The posterior geometric state and covariance at the preceding observation are $\mathbf{s}_{t-1}$ and $\mathbf{P}_{t-1}$. The Kalman transition $\mathbf{F}_t$ carries the state forward, $\boldsymbol{\xi}_{t-1}$ denotes the state-space motion input formed from the preceding motion and heading heads, and $\mathbf{Q}_t$ denotes process uncertainty. The prediction is

$$\mathbf{s}_t^- = \mathbf{F}_t\mathbf{s}_{t-1} + \boldsymbol{\xi}_{t-1}, \qquad \mathbf{P}_t^- = \mathbf{F}_t\mathbf{P}_{t-1}\mathbf{F}_t^\top + \mathbf{Q}_t, \tag{8}$$

where the superscript minus identifies the predicted state and covariance before measurement correction. The observation map $\mathbf{H}_t$ extracts position from the geometric state, and $\mathbf{I}$ is the identity of matching state dimension. Prediction uses the preceding motion output, whereas the current visual measurement appears only in the update, preserving the order of the two Kalman operations. Given $\mathbf{z}_t$ and $\mathbf{R}_t$ from Equation (7), the innovation covariance $\mathbf{S}_t$, gain $\mathbf{K}_t$, posterior state $\mathbf{s}_t$, posterior covariance $\mathbf{P}_t$, and map position $\mathbf{x}_t^K$ passed to final Mean Shift are

$$\begin{aligned}
\mathbf{S}_t &= \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^\top + \mathbf{R}_t, \\
\mathbf{K}_t &= \mathbf{P}_t^- \mathbf{H}_t^\top \mathbf{S}_t^{-1}, \\
\mathbf{s}_t &= \mathbf{s}_t^- + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}_t \mathbf{s}_t^-), \\
\mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t)\mathbf{P}_t^- (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t)^\top + \mathbf{K}_t \mathbf{R}_t \mathbf{K}_t^\top, \\
\mathbf{x}_t^K &= \mathbf{H}_t \mathbf{s}_t.
\end{aligned} \tag{9}$$

Final Mean Shift decoding. The final Mean Shift stage uses the Kalman position to select a candidate neighborhood and scores those candidates against the current UAV image. Seeds follow the resulting spatial response as in Figure 3; their converged modes and supports yield the output. Front Mean Shift produces the visual measurement used by the GRU and filter, while this second application produces the final position.

The Kalman position $\mathbf{x}_t^K$ centers the final candidate neighborhood. Its centers are $\mathbf{a}_{t,j} \in \mathbb{R}^2$, with normalized current-image response weights $w_{t,j} > 0$ satisfying $\sum_j w_{t,j} = 1$. Starting a seed at each center and applying the update in Equation (3) gives converged modes $\boldsymbol{\mu}_{t,m}$ with nonnegative supports $q_{t,m}$; at least one support is positive. The Kalman output therefore selects the neighborhood, while the current response determines the refined position within it. This final location is the support-weighted average

$$\widehat{\mathbf{x}}_t = \frac{\sum_m q_{t,m} \boldsymbol{\mu}_{t,m}}{\sum_m q_{t,m}}. \tag{10}$$

Training objectives. Training first learns the cross-view response and then the recurrent measurement and motion outputs. For training candidate weights $p_{t,j} = p_{t,j}^{\mathrm{vis}}$ from Equation (2) and a target distribution $y_{t,j}$ with label smoothing 0.05, the visual objective combines cross-entropy with coordinate regression. The probability-weighted candidate coordinate is $\widetilde{\mathbf{x}}_t = \sum_j p_{t,j} \mathbf{c}_{t,j}$, and its training reference is $\mathbf{x}_t^\star$. Writing ℓ_{SL1} for the Smooth L1 loss averaged over the two spatial coordinates, the regression term measures the distance between the probability-weighted predicted coordinate $\widetilde{\mathbf{x}}_t$ and the ground-truth position $\mathbf{x}_t^\star$. The coefficient 0.25 is a weighting hyperparameter used to balance the coordinate-regression term with the cross-entropy term, preventing the regression loss from dominating the visual classification objective. The resulting visual objective is

$$\mathcal{L}_{\mathrm{vis},t} = -\sum_j y_{t,j} \log p_{t,j} + 0.25\, \ell_{\mathrm{SL1}}(\widetilde{\mathbf{x}}_t, \mathbf{x}_t^\star). \tag{11}$$

For temporal training, $\mathbf{q}_t^\star \in \mathbb{R}^2$ is the reference position expressed in the measurement coordinate system, and $\mathbf{d}_t^\star \in \mathbb{R}^2$ is the reference displacement from t to $t+1$. The motion and heading outputs determine the predicted next displacement $\mathbf{d}_t$. The indicator a_t equals one when the reference position is captured by the candidate region, and zero otherwise. Measurement regression and uncertainty supervision are applied to captured observations; displacement supervision remains active throughout. The variance objective is a Gaussian negative log-likelihood (NLL). Increasing uncertainty reduces its normalized residual penalty but increases its logarithmic penalty, thereby balancing residual fit and uncertainty magnitude. Smooth L1 penalizes small residuals quadratically and larger residuals linearly, reducing the influence of individual large errors on the coordinate objective. Displacement supervision couples successive reference positions and trains the motion output to predict their change, whereas measurement supervision trains the corrected visual position at the current observation. With $\nu_{t,k} > 0$ denoting the predicted measurement variance in coordinate k and $z_{t,k}$ denoting the corresponding measurement, the objectives are

$$\begin{aligned}
\mathcal{L}_{\mathrm{NLL},t} &= \frac{1}{4} \sum_{k=1}^{2} \left[\frac{(z_{t,k} - q_{t,k}^\star)^2}{\nu_{t,k}} + \log \nu_{t,k} \right], \\
\mathcal{L}_{\mathrm{temp},t} &= 2 a_t \ell_{\mathrm{SL1}}(\mathbf{z}_t, \mathbf{q}_t^\star) + 2.50 \ell_{\mathrm{SL1}}(\mathbf{d}_t, \mathbf{d}_t^\star) + 0.05 a_t \mathcal{L}_{\mathrm{NLL},t}.
\end{aligned} \tag{12}$$

4 EXPERIMENTS

Dataset and controlled reference protocol. We use Bearing-UAV-90K (Liu et al., 2026) with four domains, denoted Cities A–D. Each city contributes one route-ordered training sequence and two held-out evaluation sequences. The shared model is updated by city episodes, and recurrent and filter states are reset at each episode boundary. Across the eight held-out sequences, evaluation contains 1,482 observations.

The UAV observations are ordered along navigation paths; they are not successive frames from a continuously recorded video. We retain their metric coordinates and path geometry. Within each city, the training sequence is divided chronologically into training and validation portions; the two evaluation sequences are never used to select model or filter settings.

The evaluation asks the model to refine a supplied local position. For each observation, the local reference is constructed from its annotated coordinate with a deterministic, smoothly varying offset of at most 6.00 m. The offset varies with route and observation index. A separate path reference is available to the final decoder. All component variants receive identical observations and references, so their differences measure architectural choices under the same reference condition.

Implementation and metrics. The shared backbone is MobileNetV3-Small (Howard et al., 2019). Its 576-dimensional output is mapped to 512-dimensional view descriptors, and the GRU hidden size is 256. The front lattice retains 18 heading-forward candidates, as shown in Figure 3. Three-frame, two-frame, and one-frame variants use separately trained temporal checkpoints.

Training proceeds through ordered episodes from all four cities using one shared set of model parameters. Checkpoints are selected using only the validation portion of each training sequence.

Mean Localization Error (MLE) averages the Euclidean position error in meters. Localization Success Rate (LSR) is the percentage of predicted positions within a specified radius of the evaluated coordinate; LSR@3 and LSR@15 use radii of 3 m and 15 m, respectively. For uncertainty intervals on paired comparisons, we resample the eight routes, rather than individual observations, 20,000 times and report the 2.50% and 97.50% quantiles. These intervals describe route variation for one training run per variant.

4.1 REFERENCE CONDITION

On the same 1,482 observations, the supplied local reference alone has 5.37 m MLE; its median error is 5.59 m and its maximum error is 6.00 m. The evaluated position lies within the front candidate region in 98.65% of cases. The complete model obtains 3.76 m MLE and 41.90% LSR@3. Since it also receives a path reference, this comparison describes the full reference-conditioned pipeline rather than the effect of visual evidence in isolation.

4.2 COMPARISON WITH PUBLISHED METHODS

Table 1 reports localization results for University-1652, SUES-200, DenseUAV, GTA-UAV, and the two VGG-16 variants of Bearing-UAV in Liu et al. (2026). We report LSR@15 and MLE, the location metrics common to every row. The published methods follow their original evaluation protocols; our method receives local and path references. The table places reported values in context, while the matched-input ablations below test the proposed architecture.

Table 1: UAV-view localization results. Baselines are reported by Liu et al. (2026); WA denotes weather augmentation. Protocols differ: FieldAnchor receives supplied local and path references.

Method	LSR@15 (%) ↑	MLE (m) ↓
University-1652	15.11	33.15
SUES-200	15.76	30.83
DenseUAV	16.54	28.79
GTA-UAV	27.96	28.43
Bearing-UAV	89.36	8.61
Bearing-UAV + WA	92.88	7.48
FieldAnchor	**100.00**	**3.76**

4.3 Ablation Studies

Component ablation. Table 2 compares the complete system with three component removals. Removing the GRU raises MLE from 3.76 to 4.28 m and reduces LSR@3 by 9.24 percentage points. The paired MLE difference between the complete model and the variant without GRU is -0.52 m with a route-level 95% interval of $[-0.62, -0.42]$ m. Removing the complete final refinement stage raises MLE to 6.79 m; its paired difference is -3.03 m with interval $[-3.36, -2.63]$ m. For the final-stage ablation, the front visual measurement remains available; the final neighborhood search, rescoring, and Mean Shift decoder are removed together. This row therefore tests the final refinement stage as a whole.

Table 2: Component ablation on 1,482 observations from eight held-out routes. Every variant uses the same evaluation inputs and supplied references. The final-refinement column comprises neighborhood construction, candidate rescoring, and Mean Shift decoding.

GRU	Kalman	Final refinement	LSR@3 (%) ↑	LSR@15 (%) ↑	MLE (m) ↓
×	✓	✓	32.66	99.87	4.28
✓	×	✓	41.30	**100.00**	3.76
✓	✓	×	15.45	98.04	6.79
✓	✓	✓	**41.90**	**100.00**	**3.76**

Adding the Kalman stage increases LSR@3 from 41.30% to 41.90%; both variants reach 100.00% LSR@15 and round to 3.76 m MLE. Beyond this threshold-level difference, the filter carries position and uncertainty between observations and supplies the center for final refinement.

Explicit temporal context. Table 3 compares one-, two-, and three-observation inputs with persistent recurrent state in every variant. The three-observation model has 41.90% LSR@3, compared with 39.81% and 39.14% for one and two observations. Its MLE is lower by 0.12 m and 0.16 m, respectively; the paired route-level intervals for these differences are $[-0.16, -0.10]$ m and $[-0.19, -0.13]$ m. Unlike the GRU ablation in Table 2, this comparison keeps the recurrent state active and changes the explicit input window.

Table 3: Temporal-context ablation with persistent recurrent state. Each input length uses a separately trained checkpoint.

Input	LSR@3 (%) ↑	MLE (m) ↓
1 frame	39.81	3.88
2 frames	39.14	3.92
3 frames	**41.90**	**3.76**

Final-neighborhood support. Table 4 varies final-neighborhood support while retaining the same model and held-out observations. The standard 6×6 lattice scores 36 candidates. Smaller lattices risk excluding nearby evidence; larger lattices expand the response field and its computational cost.

Table 4: Final-neighborhood ablation on the same 1,482 held-out observations. The square candidate lattice is varied while the trained model and evaluation inputs are kept fixed; the main configuration uses 6×6.

Window	LSR@3 (%) ↑	LSR@15 (%) ↑	MLE (m) ↓	Time (ms) ↓
4×4	37.65	99.93	4.17	**4.04**
5×5	41.90	**100.00**	3.77	5.00
6×6	41.90	**100.00**	3.76	7.52
7×7	**42.04**	**100.00**	**3.74**	9.50
8×8	**42.04**	**100.00**	**3.74**	10.27

A 4×4 lattice raises MLE to 4.17 m, indicating that the final response needs sufficient spatial support. Expanding the 6×6 lattice to 8×8 reduces MLE from 3.76 m to 3.74 m and increases LSR@3 from 41.90% to 42.04%, while Mean Shift time rises from 7.52 ms to 10.27 ms. The near-identical MLE of the 5×5 and 6×6 settings shows that moderate neighborhoods already cover most useful nearby responses. The 7×7 and 8×8 settings slightly improve the location metrics, but their additional candidate evaluations increase decoding time. We use 6×6 as the main setting because it captures the observed accuracy trend without paying the full cost of the largest neighborhoods. These results concern the final spatial decoder; they do not change the 18-candidate front measurement used by the recurrent stage.

Qualitative trajectories. Figure 5 shows held-out route predictions from Cities A and C against the waypoint reference paths. The visualization places the final coordinates in geographic context: predictions form route-aligned sequences across different street and building patterns, rather than a collection of independent map-tile labels. Local deviations remain visible along the paths, so the figure complements the aggregate error and success-rate results without replacing them. The two panels illustrate the behavior of the complete model; the component tables provide the controlled comparisons.

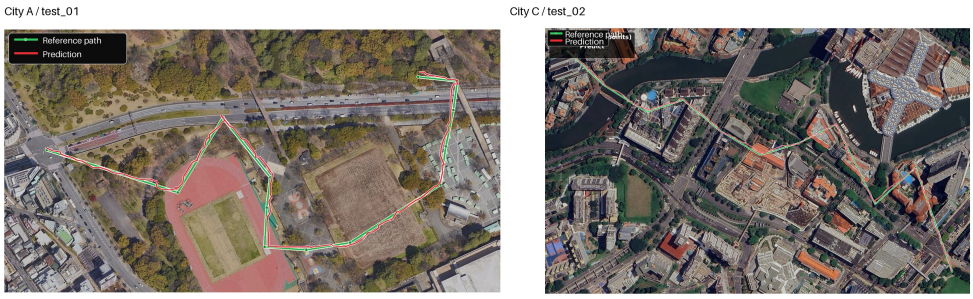

Figure 5: Representative held-out trajectories from Cities A and C. Green denotes the waypoint reference path and red denotes prediction.

5 Conclusion

FieldAnchor combines long-horizon GRU visual memory, medium-horizon Kalman geometric estimation, and short-horizon Mean Shift refinement. Six input branches integrate appearance changes, satellite context, and spatial state; four readout heads connect recurrent memory to geometric estimation. On eight held-out routes, the model achieves 3.76 m MLE and 41.90% LSR@3. The ablations support recurrent processing, three-observation input, and the complete final refinement stage under the supplied-reference protocol. The results connect the architectural roles to measurable behavior: recurrent processing improves localization against its removal, the three-observation input outperforms shorter windows, and the final spatial stage produces a continuous coordinate from the current candidate response. The Kalman component maintains the geometric state used to center that final response. This study establishes the performance of the combined model for local UAV–satellite localization with supplied references.

AI USE STATEMENT

Generative AI tools assisted with drafting, restructuring, translating, and editing the manuscript, including LaTeX formatting. The architecture diagrams were supplied by the authors. The authors are responsible for the technical descriptions, numerical claims, citations, and final manuscript.

ETHICS STATEMENT

This work analyzes an existing research dataset and collects no additional human-subject data. Cross-view UAV localization can support inspection, mapping, and emergency response, but it can also be misused for surveillance or navigation in sensitive settings. We therefore present a controlled local-refinement study rather than a deployment claim; any future release or deployment should follow the source-data terms and applicable privacy, aviation, and safety requirements.

REPRODUCIBILITY STATEMENT

Section 3 describes the model stages. Section 4 states the data split, reference condition, and evaluation metrics. The component, temporal-context, and neighborhood experiments specify their comparison conditions alongside the corresponding tables.

REFERENCES

Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, C. James Taylor, and Gerhard Neumann. Recurrent Kalman networks: Factorized inference in high-dimensional deep feature spaces. In *International Conference on Machine Learning*, pp. 544–552, 2019.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, pp. 1724–1734, 2014.

Jun-Wei Hsieh, Cheng-Han Chou, Ming-Ching Chang, Ping-Yang Chen, Santanu Santra, and Chih-Sheng Huang. Mean-shift based differentiable architecture search. *IEEE Transactions on Artificial Intelligence*, 5(3):1235–1246, 2024. doi:10.1109/TAI.2023.3329792.

M. Dai, E. Zheng, Z. Feng, L. Qi, J. Zhuang, and W. Yang. DenseUAV2: Vision-based UAV self-positioning in low-altitude urban environments. *IEEE Transactions on Image Processing*, 33:493–508, 2024.

Fabian Deuser, Konrad Habel, and Norbert Oswald. Sample4Geo: Hard negative sampling for cross-view geo-localisation. In *IEEE/CVF International Conference on Computer Vision*, pp. 16847–16856, 2023.

Jose M. Facil, Daniel Olid, Luis Montesano, and Javier Civera. Condition-invariant multi-view place recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for MobileNetV3. In *IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019.

Y. Ji, B. He, Z. Tan, and L. Wu. Game4Loc: A UAV geo-localization benchmark from game data. In *AAAI Conference on Artificial Intelligence*, 39(4):3913–3921, 2025.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

Kejia Liu, Haoyang Zhou, Ruoyu Xu, Peicheng Wang, Mingli Song, and Haofei Zhang. Beyond matching to tiles: Bridging unaligned aerial and satellite views for vision-only UAV navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. `https://arxiv.org/abs/2603.22153`.

Michael Milford and Gordon Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation*, pp. 1643–1649, 2012.

Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. KalmanNet: Neural network aided Kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022.

Xiaohan Zhang, Waqas Sultani, and Safwan Wshah. Cross-view image sequence geo-localization. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2914–2923, 2023.

Zhedong Zheng, Yunchao Wei, and Yi Yang. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In *ACM International Conference on Multimedia*, 2020.

Sijie Zhu, Taojiannan Yang, and Chen Chen. VIGOR: Cross-view image geo-localization beyond one-to-one retrieval. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3640–3649, 2021.

R. Zhu, L. Yin, M. Yang, F. Wu, Y. Yang, and W. Hu. SUES-200: A multi-height multi-scene cross-view image benchmark across drone and satellite. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9):4825–4839, 2023.