AR^2-4FV: Anchored Referring and Re-identification for Long-Term Grounding in Fixed-View Videos

Supplementary Material

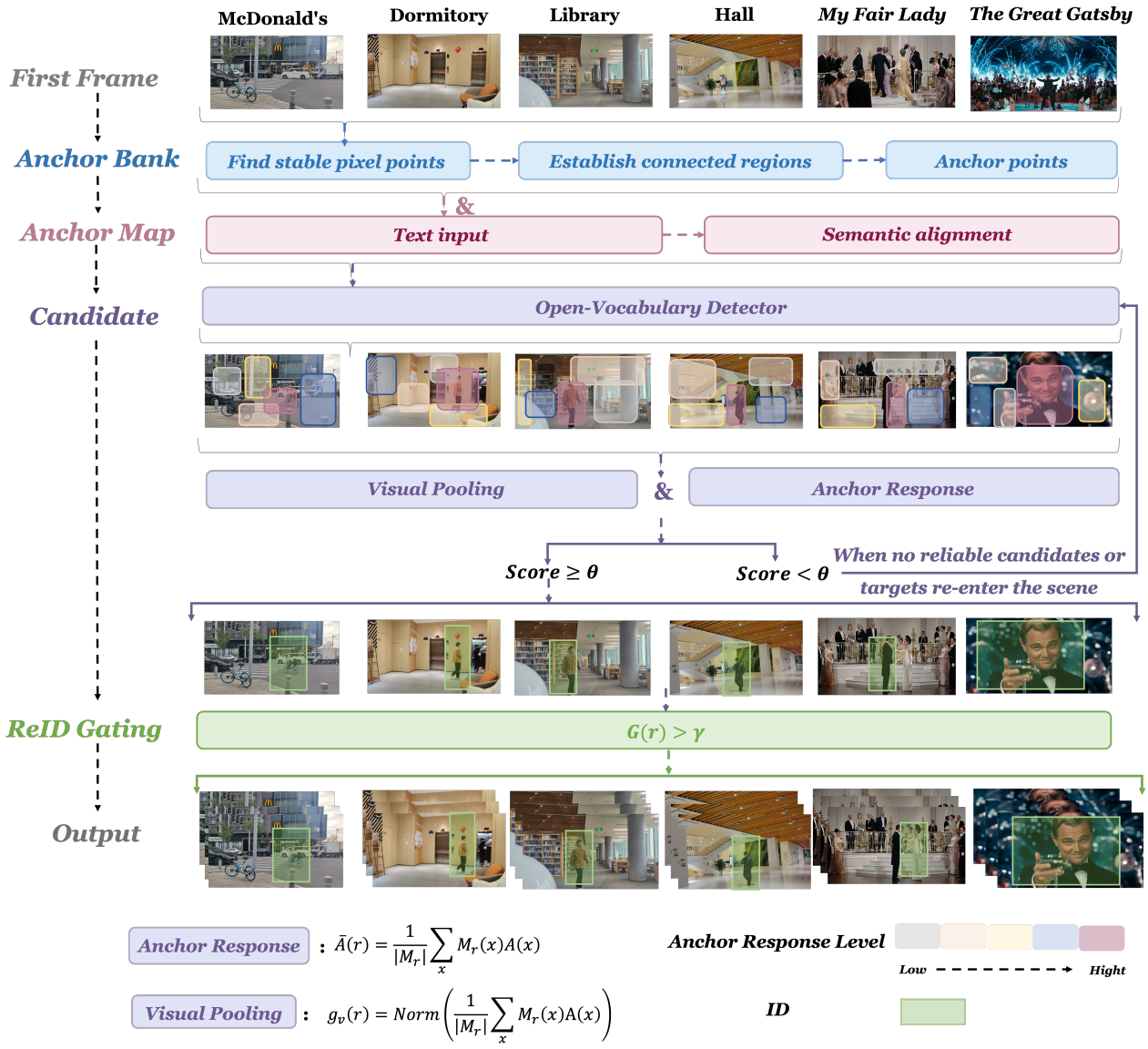

Figure 1. Expanded AR^2-4FV pipeline. Offline, an Anchor Bank distilled from the first T_0 frames generates a query-conditioned Anchor Map $A(x)$. Online, an open-vocabulary detector proposes bounding boxes; within each refined candidate r, mask-aware pooling produces $g_v(r)$ and anchored-response gives $\overline{A}(r)$. The fusion score $\mathrm{Score}(r)$ gates search by threshold θ. If no candidate is reliable, the re-entry prior P_t^{re} is updated (EMA + Gaussian) and used to re-weight proposals. Eligible candidates are validated by ReID-Gating with $G(r) \geq \gamma$; validated targets update the identity queue and redirect P_{t+1}^{re} toward the corresponding anchor. Symbols match Sec 3.2-3.3.

A. More Details of Methodology

A.1. Pipeline

Figure 1 provides an expanded illustration of the AR^2-4FV pipeline. Given the first-frame input, the offline Anchor Bank is constructed by detecting stable background elements and grouping them into connected spatial regions. During inference, the text query is aligned with the Anchor Bank to generate a query-conditioned Anchor Map, which serves as a persistent scene prior even when the referent is invisible. An open-vocabulary detector produces candidate regions, each of which is scored by jointly considering visual pooling features and anchor responses. Candidates with Score $\geq \theta$ proceed to the ReID-Gating stage, where appearance similarity, anchor consistency, and displacement cues are blended into a gate score $G(r)$. If $G(r) \geq \gamma$, the identity is accepted and the momentum queue is updated; otherwise, or when no reliable candidates appear, the re-entry prior guides search. This detailed visualization highlights how AR^2-4FV integrates anchor-based spatial reasoning, candidate scoring, and ReID-gating to maintain long-term identity continuity in fixed-view videos.

A.2. Pseudocode

BUILDANCHORBANK. This module extracts stable background regions from the initial T_0 frames, partitions them into K anchor regions $\{M_k\}$, and computes for each region its feature prototype p_k and centroid c_k to form the Anchor Bank $\mathcal{B}$.

Algorithm 1 BUILDANCHORBANK

Require: Frames $\{I_t\}_{t=1}^{T_0}$, visual encoder f_v, #anchors K
Ensure: Anchor Bank $\mathcal{B} = \{(M_k, p_k, c_k)\}_{k=1}^{K}$

$F_t \leftarrow f_v(I_t)$ for $t = 1, \ldots, T_0$
Compute temporal stability map S (low variance & small optical flow)
Partition S into K connected stable regions $\{M_k\}_{k=1}^{K}$
Pick median-brightness frame index $t^\star$
for $k = 1$ to K **do**
  $p_k \leftarrow \mathrm{Norm}\big(\mathrm{AvgPool}(F_{t^\star}, M_k)\big)$
  $c_k \leftarrow \frac{1}{|M_k|} \sum_x M_k(x)\, x$
end for
return $\{(M_k, p_k, c_k)\}_{k=1}^{K}$

MAKEANCHORMAP. Given a query q, we encode it to obtain $\phi_l(f_l(q))$, compute cosine similarity to anchor prototypes $\{p_k\}$, apply softmax normalization, and aggregate the anchor regions to produce a query-conditioned Anchor Map $A(x)$.

Algorithm 2 MAKEANCHORMAP

Require: Query q, Anchor Bank $\mathcal{B}$, text encoder f_l, heads ϕ_l, ϕ_v, temperature τ
Ensure: Anchor Map $A(x) \in [0, 1]$

for each $(M_k, p_k, c_k) \in \mathcal{B}$ **do**
  $s_k \leftarrow \cos\big(\phi_l(f_l(q)), \phi_v(p_k)\big)$
end for
$w_k \leftarrow \mathrm{softmax}(\tau s_k)$ for $k = 1, \ldots, K$
$A(x) \leftarrow \sum_{k=1}^{K} w_k\, M_k(x)$
return A

UPDATEREENTRYPRIOR. The re-entry prior is updated by smoothing the previous prior with a Gaussian kernel and combining it with the Anchor Map using weight β, followed by ℓ_1 normalization.

Algorithm 3 UPDATEREENTRYPRIOR

Require: Anchor Map A, previous prior P_{t-1}^{re}, Gaussian width σ, EMA weight β,
  (optional) confirmed anchor index $k^\star$ with center $c_{k^\star}$ and redirect weight ρ
Ensure: New prior P_t^{re}

if search mode (no reliable candidate) **then**
  $\tilde{P} \leftarrow \beta\,(G_\sigma * P_{t-1}^{\mathrm{re}}) + (1 - \beta)\,A$
else
  $\tilde{P} \leftarrow \rho\, G_\sigma(\cdot - c_{k^\star}) + (1 - \rho)\,A$
end if
$P_t^{\mathrm{re}} \leftarrow \tilde{P} \big/ \sum_x \tilde{P}(x)$
return P_t^{re}

B. More Details of AR^2-4FV-Bench

B.1. Collect Settings and Device Details

All videos in AR^2-4FV-Bench are captured using fixed-view cameras. We either mount consumer cameras on tripods or rely on pre-installed surveillance cameras, ensuring that the viewpoint, focal length, and optical center remain strictly unchanged throughout each sequence. In terms of hardware, the dataset primarily uses 1080p/30 fps and 720p/25–30 fps consumer video devices, including common surveillance cameras (e.g., HIKVISION, Honeywell) and portable cameras (e.g., iPhone) which provide sufficient resolution for referring and re-identification. Camera height is typically set between $1.3 - 1.8$ meters and kept horizontal or slightly downward, ensuring that the spatial layout remains stable across the entire video. Each sequence has an average duration of over 120 seconds, covering different times of day (morning/afternoon/evening) and diverse weather conditions (sunny, cloudy, light rain) as shown in Figure 2. This preserves realistic variations including pedestrian flow, illumination changes, moving shadows, and temporary occlusions.

Figure 2. Fixed-view sequences in AR^2-4FV-Bench showing the appear $\rightarrow$ disappear $\rightarrow$ re-enter cycle across morning and evening. Each row corresponds to one static camera (restaurant storefront, hospital lobby, sidewalk, courier station). Green boxes denote the referent. Light-green boxes indicate anchor-aligned regions. This figure illustrates the collection protocol and highlights the long-term re-entry characteristics of our dataset.

B.2. Referring Expression Generation

In AR^2-4FV-Bench, we generate referring expressions for each target based on appearance and spatial location attributes. For appearance, we extract attributes such as clothing color and category (e.g., black jacket, white hoodie), body build (e.g., tall, slim), accessories (e.g., backpack, cap, glasses), and basic action states (e.g., walking, standing). These attributes are encoded into structured fields. Examples include "the person in a color clothing," "the build person carrying a accessory," and "the person who is action."

For spatial attributes, we annotate environmental regions for each scene (e.g., entrance, left walkway, corridor, stairs) and generate location expressions such as "near the entrance" or "on the right walkway." Appearance and location templates can be used independently or jointly.

To improve diversity and coherence, we further refine the template-generated expressions using a large language model (LLM). The initial template output is rewritten by the LLM for semantic equivalence, removal of redundancy, grammatical correction, and ambiguity resolution. Final expressions need verification to prevent ambiguous, inconsistent, or unobservable descriptions.

B.3. Annotation Protocol

The annotation process of AR^2-4FV-Bench follows a multi-stage pipeline designed to precisely record the referent's spatial location, visibility state, and re-entry information in long-term fixed-view videos.

Referent Selection. For each video, we first determine a unique referent according to the provided referring expression. When multiple potential candidates appear, we ensure that the chosen referent is semantically distinguishable (e.g., the man in a white shirt, the woman walking a dog). We prefer to select individuals who exhibit appearance-disappearance-re-entry patterns. Only one referent is annotated per video to preserve consistent long-term association.

Visibility Annotation. We annotate the referent's visibility state on every frame using three labels:

- *Visible*: the referent is fully or partially visible;
- *Invisible-Occluded*: the referent is completely occluded by other people or objects but remains inside the scene;
- *Invisible-Absent*: the referent has exited the field of view.

Spatial Annotation. For all visible frames, we annotate the referent with a bounding box. When partial occlusion occurs, the bounding box is required to cover the referent's actual physical extent rather than only the visible part.

Re-entry Event Labeling. For each disappearance and re-entry event, We record:

- the disappearance time t_1, when the referent transitions from *visible* to *invisible*;
- the re-entry time t_2, when the referent transitions from *invisible* back to *visible*;
- an *event ID* used to index and count all re-entry occurrences.

C. Analysis of Experimental Results

Analysis. Compared with existing R-VOS models, our method performs best on the AR^2-4FV-Bench. ReferFormer and MTTR rely on motion cues and fail under minimal appearance changes; The video-level semantic aggregation of SOC is not effective in static scenes; OnlineRefer accumulates errors without stable motion updates; and DsHmp overfits weak temporal cues in fixed views, SSA relys on short-term appearance and query-driven local alignment. Although recent visual-language tracking models (e.g., DUTrack) and SAM-based models (e.g., VideoLISA) show strong detection capabilities, they still struggle in our benchmark due to the lack of persistent spatial memory for long-term re-entry. Overall, these models suit dynamic scenes but fail to exploit the stable spatial anchors and long-term consistency of fixed-view videos.