SELVA: A Reliable and Fast Selectivity Estimation Method for Query Plan Optimization in Video Analytics

Guang Lu[1], Jing Zhang[2], Ji Li[1], Jiaquan Li[1], and Chuanwen Li[1](✉)

[1] Northeastern University, Shenyang, China
lichuanwen@mail.neu.edu.cn, 2010656@stu.neu.edu.cn
[2] Shenyang Aerospace University, Shenyang, China

Abstract. Video Database Management Systems (VDMS) face a critical challenge in optimizing query plans: accurate selectivity estimation, which predicts the proportion of data satisfying a predicate. Existing methods rely on fixed sampling rates (e.g., 10% of video data) to approximate selectivity, but this incurs substantial computational overhead for video data transformation (26-37% of query time) and lacks statistical guarantees for estimation reliability. The reliance on manual, static sampling further limits efficiency, as it fails to adapt to varying query complexities. We propose a dynamic sampling framework that reframes selectivity estimation under the statistical framework. Our approach introduces two key innovations. First, we design confidence sequences using an adaptive strategy, which dynamically adjusts sample sizes to meet user-defined error bounds and confidence levels, eliminating the need for fixed sampling rates. Second, we integrate an optimized betting algorithm with geometric checkpoint scheduling to minimize sample size while preserving statistical guarantees. Additionally, we leverage control variates, a variance reduction technique, to exploit correlations between transformation algorithms, further reducing sampling requirements. Experiments demonstrate that our framework reduces selectivity estimation overhead by 31% compared to conventional fixed sampling, while achieving equivalent accuracy (95% confidence, ±5% error margin). By replacing heuristic sampling with rigorous statistical guarantees, our method enhances the efficiency and scalability of VDMS query optimization, particularly for large-scale video analytics applications.

Keywords: Video analytics · Query plan optimization · Selectivity estimation · Confidence sequence

1 Introduction

Recent advances in neural networks have enabled structured analysis of unstructured video data, yet real-time querying remains hindered by the computational intensity of video transformations. For instance, processing a one-month surveillance video with YOLOv8 requires over 12 days on a T4 GPU [7]. While initial systems optimized single-predicate queries (e.g., "find frames with buses") by minimizing redundant transformations [9], real-world applications demand scalable solutions for *multi-predicate queries*, complex tasks requiring coordinated transformations like object detection, face recognition, and attribute classification (e.g., identifying "two men riding a motorcycle through an intersection").

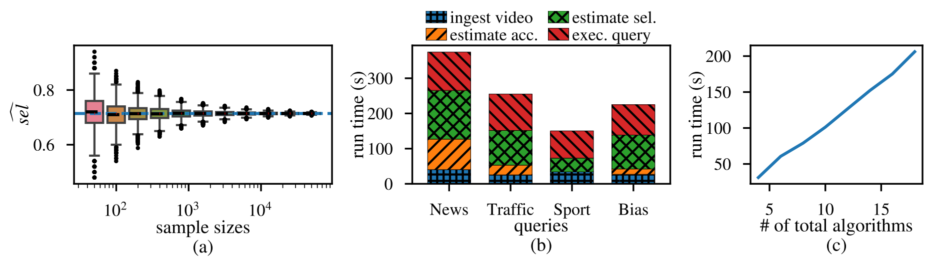

Fig. 1. (a) The estimated selectivity distribution for a TV news analysis query. The blue dotted line denotes the true selectivity. (b) Query processing time breakdown for the query. (c) The selectivity estimation cost under various sample sizes.

Existing systems like VIVA [15] and Chao et al. [2] optimize such queries via cost models, but their efficacy hinges on accurate selectivity estimation, i.e., predicting the fraction of video data satisfying a predicate. Traditional databases achieve up to 1.47× speedups with precise selectivity estimation [5], but video database management systems (VDMS) face unique challenges. Current methods manually fix sample sizes, risking inaccuracy with small samples or inefficiency with large ones. For example, using 10% of video data consumes 26-37% of query time (Fig. 1 b) and scales linearly with transformations (Fig. 1 c), yet still incurs errors exceeding 0.1 in worst cases (Fig. 1 a). Without statistical guarantees, these approaches falter for long videos or queries with dozens of predicates, stifling real-time analytics in critical domains like surveillance.

We propose **SELVA**, a selectivity estimation method that dynamically adjusts sample sizes while ensuring statistical reliability (as defined in Eq. (2)). SELVA integrates three key innovations: First, *Confidence Sequences* construct time-uniform confidence intervals using a martingale betting strategy. Sampling stops once the confidence interval width falls below the error tolerance with the desired confidence level Section 4.1. Second, *Geometric Checkpoint Scheduling* and *Adaptive Betting* combine exponentially spaced checkpoints with an adaptive betting algorithm to minimize CI width and reduce computation Section 4.2. Finally, *Control Variates* leverage correlations between transformation algorithms (e.g., YOLOv8m vs. YOLOv8l detections [7]) to reduce estimator variance, enabling fewer samples as detailed in Section 4.2.

Experiments show SELVA reduces estimation time by 31% compared to the conventional 10% fixed-sampling approach, while maintaining the same accuracy guarantees. To the best of our knowledge, SELVA is the first method to integrate confidence sequences into VDMS. Our contributions are as follows:

- We frame the selectivity estimation problem in VDMS through a probabilistic framework, using confidence sequences to provide statistical guarantees on error margins.
- We propose three optimizations to enhance CS performance, including a geometric checkpoint scheduling, an adaptive betting algorithm, and control variates that collectively enhance the tightness of confidence bounds and reduce sampling overhead.
- We thoroughly evaluate our techniques to demonstrate their efficiency and effectiveness on real-world queries.

The rest of this paper is organized as follows: In Section 2, we define the problem after presenting related work. Section 3 shows the motivation of our study and SELVA's workflow. We introduce the proposed method in Section 4. Section 5 evaluates all proposed techniques, and Section 6 concludes the paper.

2 Preliminary

2.1 Related Work

Video Analytics Optimization. Early work focused on single-predicate queries using proxy models [9, 8, 20], feature indexing [12, 6, 1], model cascades [1, 3], or advanced sampling [14]. While effective for simple tasks (e.g., "detect cars"), these methods lack scalability for multi-predicate queries requiring coordinated transformations.

Multi-Predicate Systems. EVA [19] materializes intermediate results to accelerate repeated queries but struggles with exploratory tasks. VIVA [15] optimizes query plans via cost models but relies on heuristic sampling for selectivity estimation. Chao et al. [2] leverage user-provided video metadata, assuming impractical prior knowledge. All share a critical limitation: selectivity estimation lacks statistical rigor, leading to suboptimal plans or excessive sampling.

Probabilistic Methods. SUPG [10] and ABAE [11] use confidence intervals for single-predicate queries but ignore inter-algorithm correlations. InQuest [16] extends these to streaming data but remains confined to single transformations.

2.2 Problem Definition

In this paper, we model the video as a series of consecutive segments $\mathcal{V} = \{s_i\}$, where each segment is either a fixed-length clip or a single frame, depending on the input signature of a transformation T.

$$T_R : \mathcal{V} \to R := \langle fid, l_1, \ldots, l_k, score \rangle \tag{1}$$

where fid identifies the frame/clip, $\{l_i\}$ represent transformation outputs (e.g., object labels), and *score* denotes algorithm confidence. Let $\mathbb{A}^{(R)}$ denote the set of algorithms implementing transformation T_R - for example, $\mathbb{A}^{(R)}$ for object detection might include YOLOv8m and YOLOv8l models [7].

Definition 1 (Selectivity). *The selectivity of a predicate p over relation r_j (materialized by algorithm $A_j^{(R)}$) is defined as:*

$$\mathrm{sel}(p^{r_j}) := \frac{|\{t \in r_j \mid p(t)\}|}{|r_j|}$$

Our goal is to estimate $\mathrm{sel}(p^{r_j})$ with (ϵ, δ)-guarantees: for user-specified error tolerance $\epsilon > 0$ and confidence level $\delta \in (0, 1)$, we require:

$$Pr\left(\left|\widehat{\mathrm{sel}}(p^{r_j}) - \mathrm{sel}(p^{r_j})\right| < \epsilon\right) \geq 1 - \delta. \tag{2}$$

3 The SELVA Selectivity Estimation Components

Current video database management systems face critical bottlenecks in query optimization due to costly selectivity estimation. As shown in Fig. 1(b), selectivity estimation consumes $26 - 37\%$ of total query time, scaling linearly with the number of transformations (Fig. 1c). While reducing sample sizes could alleviate costs, naive approaches risk unacceptable errors. For example, a 10% sample yields absolute errors exceeding 0.1 in worst cases (Fig. 1 a). This tension motivates **SELVA**, which provides selectivity estimates with *probabilistic guarantees* on error bounds while minimizing sampling overhead.

SELVA Workflow. Given inputs $\mathcal{V}$ (video), $\mathbb{A}$ (algorithms), $\mathcal{P}$ (predicates), and constraints (ϵ, δ), SELVA operates as follows:

1. **Initialization**: Group predicates by transformation family R_i, prioritizing fast algorithms. Initialize samplers, schedulers, and CS executors per predicate.
2. **Sampling**: For each video segment $s \in \mathcal{V}$, evaluate predicates using p_j^i to obtain scores $w_j^i \in W_j^i$. Apply control variates if correlated estimates exist.
3. **Scheduling**: Schedule checkpoints geometrically (e.g., $t = 1, 2, 4, 8, \ldots$), collecting samples and updating statistics $\hat{\mu}_t$, $\hat{\sigma}_t^2$.
4. **CS Execution**: At each checkpoint t, compute CI bounds via adaptive betting.

By integrating statistical rigor with system-aware optimizations, SELVA reduces estimation overhead by 31% versus fixed sampling while maintaining (ϵ, δ)-guarantees (Fig. 2), making it practical for large-scale video analytics.

4 Selectivity Estimation

In this section, we formalize selectivity estimation within a probabilistic framework and introduce confidence sequences with a betting strategy. We then present three optimization techniques to improve sample size efficiency and reduce computational overhead.

4.1 Confidence Sequence for Selectivity Estimation

We assume the randomly drawn samples S are independent and identically distributed (i.i.d.). For classification-based transformation algorithms, which estimate the conditional probability $Pr(Y|X)$ (where Y is a class label and X is input data), we derive equality predicates p_e by thresholding scores score$(s) = Pr(Y = y \mid X = s)$:

$$p_e(s) = \begin{cases} 1, & \text{if score(s)} \geq h. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Inequality predicates p_n require aggregating confidence scores across multiple records per segment. Suppose algorithm A_i generates n records from a segment, with m satisfying p_n. The probability of k records meeting p_n is:

$$P = \prod_{i=1}^{k} Pr(y \mid x_i) \cdot \prod_{i=k+1}^{n} (1 - Pr(y \mid x_i)). \tag{4}$$

Summing over all valid k values (dependent on the operator $\leq$ or $\geq$) gives the segment's confidence score:

$$\text{score}(s) = \sum_{k=\in K} \sum_{I \in \binom{[n]}{k}} \left(\prod_{i \in I} Pr(y \mid x_i) \cdot \prod_{j \notin I} (1 - Pr(y \mid x_j)) \right), \tag{5}$$

where $[n] = \{1, 2, \ldots, n\}$ and $\binom{[n]}{k}$ denotes all k-combinations of n records.

Let Z model the Bernoulli-distributed random variable for $p_e(s)$, with selectivity estimation framed as a mean estimation problem: $\widehat{\text{sel}}(p_j^{(R)}) = \frac{1}{N} \sum_{i=1}^{N} Z_i$, where N is the sample size. We seek probabilistic guarantees that $\hat{\mu} = \frac{1}{N} \sum Z_i$ lies within ϵ of μ with confidence $1 - \delta$.

To achieve this, we adopt confidence sequences (CS), a sequence of intervals $\{CI_t\}_{t \in \mathcal{T}}$ where $Pr(\forall t \in \mathcal{T} : \mu \in CI_t) \geq 1 - \delta$. A betting strategy [18] refines these bounds: for each candidate mean $m \in [0, 1]$, we wager against the hypothesis $H_0^m : \mu = m$. If H_0^m holds, the "capita" (a likelihood ratio statistic) remains stable; otherwise, it grows until H_0^m is rejected at confidence $1 - \delta$. This strategy uses a sequence $(\lambda_t(m))_{t \in \mathcal{T}}$, where λ_t depends on observed Z_t.

4.2 Optimizing Confidence Sequences

Geometric Checkpoint Scheduling (GCS) While effective, computing λ_t at every t incurs overhead. Geometric checkpoint scheduling (GCS) optimizes this by evaluating CS only at checkpoints $t = \lceil \beta^k \rceil$ for $\beta > 1$. Between checkpoints, we fix λ to a constant, minimizing the expected CI half-width $\mathbb{E}[M_t]$ derived from:

$$\mathbb{E}[M_t] \lesssim \frac{\log(2/\delta) + \frac{\sigma^2}{2} \sum_{i=1}^{t} \lambda_i^2}{\sum_{i=1}^{t} \lambda_i}. \tag{6}$$

Let $a = \sum_{i=1}^{t'} \lambda_i^2$ and $b = \sum_{i=1}^{t'} \lambda_i$ at the last checkpoint t'. For the next checkpoint $t = t' + \Delta t$, fixing $\lambda_{t'+1}^{t} = \lambda$, we minimize:

$$\mathbb{E}[M_t] \lesssim \frac{\log(2/\delta) + \frac{\sigma^2}{2}(a + \Delta t \lambda^2)}{b + \Delta t \lambda}. \tag{7}$$

Taking the derivative of (7) with respect to λ and solving yields the optimal λ^*:

$$\lambda^* = \sqrt{\left(\frac{b}{\Delta t}\right)^2 + \frac{2\log(2/\delta) + a\sigma^2}{\Delta t \sigma^2}} - \frac{b}{\Delta t}. \tag{8}$$

At each checkpoint, we estimate σ^2 as $\hat{\sigma}^2 = \frac{1}{t}\sum_{i=1}^{t}(Z_i - \hat{\mu}_{t'})^2$, where $\hat{\mu}_{t'}$ is the mean at t'. This balances tight bounds with reduced computation, as checkpoints grow geometrically.

Adaptive Betting Algorithm (AB) SELVA employs an adaptive betting algorithm (AB) to optimize the capital process $\mathcal{K}_t(m)$ for $m \in [0, 1]$. By splitting the capital into two processes $\mathcal{K}_t^+(m)$ (betting $\mu > m$) and $\mathcal{K}_t^-(m)$ (betting $\mu < m$), we track opposing hypotheses simultaneously:

$$\mathcal{K}_t^{\pm}(m) := \prod_{i=1}^{t} \left(1 \pm \lambda_i^{\pm}(m) \cdot (Z_i - m)\right), \tag{9}$$

where Z_i is the i-th observation. The combined capital $\mathcal{K}_t(m) = \max\left\{\frac{1}{2}\mathcal{K}_t^+(m), \frac{1}{2}\mathcal{K}_t^-(m)\right\}$ ensures robustness. When $\mu \neq m$, one process grows exponentially, allowing confident rejection of $H_0^m : \mu = m$ via Ville's inequality[17] once $\mathcal{K}_t^{\pm}(m) > 1/\delta$.

To balance precision and computation, we discretize m into a grid $[0, 1/g, \ldots, 1]$ with $g > 1/\epsilon$. This yields confidence bounds $\mathfrak{B}_t^{\pm} := \{m \mid \mathcal{K}_t^{\pm}(m) < 1/\delta\}$ that tighten over time. Critically, confidence intervals never widen, allowing us to retain only the tightest historical bounds. For instance, if $\mathcal{K}_t^-(m') > 1/\delta$, all $m'' < m'$ are automatically excluded, reducing computational effort.

Algorithm 1 implements this adaptively. We first set initial bounds $[L_0, U_0]$ around a prior estimate o of μ (Line 3). Then, we iteratively narrow L and U by testing directional hypotheses (e.g., betting $\mu > L$) in lines 4-9. After finding one bound, we focus capital on refining the remaining bound (e.g., upper bound if $\mu > L$ is confirmed) in lines 10-18. Finally, we compute $\hat{\mu} = \arg\min_m \mathcal{K}_t^{\pm}(m)$ within the final bounds(Line 19).

The algorithm reduces computation by 1) omitting invalidated hypotheses and 2) concentrating capital on active bounds. Experiments show this achieves tighter confidence sequences with 31% fewer samples than baseline methods Fig. 2.

Control Variates for Sample Variance Reduction For inequality predicates, variance reduction becomes critical. The half-width of confidence intervals in Eq. (6) depends on σ^2—the variance of the estimator. We employ control

Algorithm 1: Adaptive Betting

Input: Prior o, error constraint ϵ, confidence δ, grid size g
Output: Confidence interval $[L_t, U_t]$, estimator $\hat{\mu}$

Initialize $t \leftarrow 1$, C_0 with δ, g;
$L_0, U_0 \leftarrow$ `Initialize(`o, ϵ`)` ; `// Initial bounds`
repeat
  Update $t \leftarrow t+1$, observe Z_t;
  Update confidence state $C_t \leftarrow C_{t-1}(Z_t)$;
  $L_t, U_t, d \leftarrow$ `ProbeDirection(`C_t, L_{t-1}, U_{t-1}`)`;
until *direction d detected*;
repeat
  Update $t \leftarrow t+1$, observe Z_t;
  Update confidence state $C_t \leftarrow C_{t-1}(Z_t)$;
  $L_t, U_t, f \leftarrow$ `RefineAnotherBound(`C_t, L_{t-1}, U_{t-1}, d`)`;
until *stopping condition f met*;
return $[L_t, U_t]$, `GetEstimator(`C_t, L_t, U_t`)`;

variates [4], leveraging correlations between transformation algorithms (e.g., different object detectors) to reduce σ^2. Let Z represent the target algorithm's selectivity estimate and Y a correlated control variate (e.g., estimates from a cheaper algorithm). We construct an adjusted estimator:

$$\hat{\mu}_c = Z + c(Y - \mathbb{E}[Y]), \tag{10}$$

where $c^* = -\operatorname{Cov}(Z, Y)/\sigma_Y^2$ minimizes variance.

We prioritize candidate algorithms by computational cost, first evaluating the cheapest to leverage its results as control variates for correlated algorithms in the same family. For each subsequent algorithm, we select maximally correlated predecessors via pilot sampling to optimize variance reduction. Iterative correlation analysis progressively reduces sample requirements for later stages, forming an efficiency feedback loop that optimizes overall sampling costs.

5 Experiments

We evaluate SELVA's three core innovations, i.e., geometric checkpoint scheduling (GCS), adaptive betting (AB), and control variates, across varying error (ϵ), confidence (δ), and correlation settings. Our experiments validate that SELVA provides (ϵ, δ)-guarantees while reducing sample sizes by 25-50% versus baseline methods, and cuts estimation runtime by up to 31% compared to prior systems like VIVA.

5.1 Setup

We implemented SELVA within VIVA's query optimizer [13], replacing its heuristic sampling with our probabilistic framework, implemented in Python with

Numba JIT compilation. Our implementation is 10x faster than original NumPy version [18]. Following the configuration by [13], we evaluated a 1-hour *News* video with predicate `count('person')` ≥ 2, $\epsilon = 0.01$, $\delta = 0.05$, and 10^3 rounds by default. To validate the efficiency of GCS and AB, we evaluated four CS construction algorithms:

- **VanillaSeq**: Sequential checkpoints, grid search (baseline from [18]).
- **VanillaGeo**: Replace sequential checkpoints with GCS.
- **AdaSeq**: Adaptive betting with sequential checkpoints.
- **AdaGeo**: Adaptive betting with GCS.

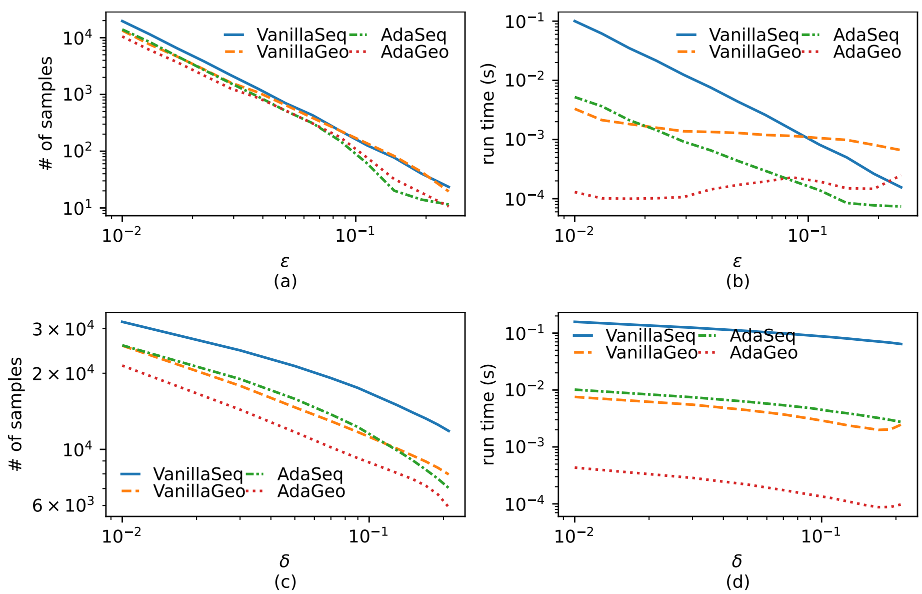

Fig. 2. The required sample sizes as a function of ϵ (a) and δ (c), respectively. The run time of algorithms as a function of ϵ (b) and δ (d), respectively.

5.2 Results and Analysis

Impact of Error constraint (ϵ). As ϵ tightens, all methods except AdaSeq (which under-samples at $\epsilon \geq 0.1$) satisfy (ϵ, δ)-guarantees. AdaGeo's GCS mitigates this by optimizing λ at checkpoints, halting sampling 25-50% earlier than VanillaSeq (Fig. 2 a). Runtime scales exponentially with $1/\epsilon$ for sequential methods but remains stable for GCS variants (Fig. 2 b)—AdaGeo achieves three-order speedups over VanillaSeq by bulk-processing samples.

Impact of confidence level (δ). Lower δ (stricter confidence) increases sample sizes exponentially across methods (Fig. 2 c), but error distributions remain stable. AdaGeo again outperforms, requiring 32-50% fewer samples than VanillaSeq and achieving 360-810x runtime gains (Fig. 2 d).

Control variates. We report the normalized error drop (Δ_e) in Fig. 3 with the formulas $\Delta_e = \text{error rate}/\delta$. For strict error constraint ($\epsilon < 0.1$), c.v. reduce error by 12-88% and sample sizes by over 50% when $m \geq 200$. For smaller m ($\epsilon > 0.1$), c.v. provides more robust results and fixes the unreliable problem for AdaSeq but requires a slightly larger sample size to estimate $\mathrm{Cov}(Z, Y)$, as in Fig. 3 (a).

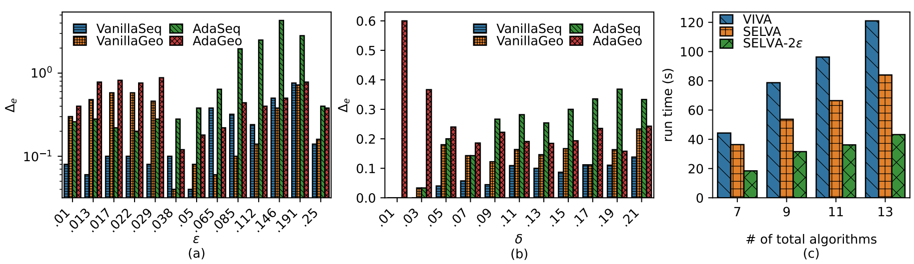

Fig. 3. The error drop after utilizing control variates as a function of ϵ (a) and δ (b), respectively. (c) Comparison of run time between SELVA and VIVA.

Comparison with previous estimation method. Under identical $\epsilon = 0.05$ (VIVA's fixed 10% sampling), SELVA reduces runtime by 18-31% (Fig. 3 c). Additionally, we relax the error constraint to 2ϵ (SELVA-2ϵ) to explore further reductions in estimation overhead. SELVA cuts run time by 18-31% compared to VIVA with the same δ. Doubling the error tolerance allowed SELVA to reduce run time by 58-64%, demonstrating its ability to flexibly trade off precision for efficiency when needed.

6 Conclusions

Multi-predicate video analytics queries require efficient selectivity estimation, yet existing methods struggle with reliability and computational overhead. We propose SELVA that leverages confidence sequences with two key innovations: (1) adaptive betting and geometric checkpointing to tighten confidence bounds, and (2) control variates to reduce sample variance. Experiments demonstrate SELVA reduces query runtime by up to 31% compared to traditional sampling methods while maintaining probabilistic guarantees, with doubled error tolerance yielding 2-3× faster convergence at equivalent confidence levels.

References

1. Anderson, M.R., Cafarella, M., et al.: Physical representation-based predicate optimization for a visual analytics database. 2019 Ieee 35th International Conference on Data Engineering pp. 1466–1477 (2019)
2. Chao, D.R., Chen, Y.T., et al.: Optimizing video queries with declarative clues. Proceedings of the VLDB Endowment pp. 3256–3268 (2024)
3. Chunduri, P., Bang, J., et al.: Zeus: Efficiently localizing actions in videos using reinforcement learning. Proceedings of the 2022 International Conference on Management of Data pp. 545–558 (2022)
4. Hammersley, J.: Monte carlo methods (2013)
5. Han, Y., Wu, Z., Wu, P., Zhu, R., Yang, J., Tan, L.W., Zeng, K., Cong, G., Qin, Y., Pfadler, A., et al.: Cardinality estimation in dbms: A comprehensive benchmark evaluation. Proceedings VLDB Endowment pp. 752–765 (2021)
6. Hsieh, K., Ananthanarayanan, G., et al.: Focus: Querying large video datasets with low latency and low cost. Proceedings of the 13th Usenix Symposium on Operating Systems Design and Implementation pp. 269–286 (2018)
7. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLOv8 (2023), https://github.com/ultralytics/ultralytics
8. Kang, D., Bailis, P., et al.: Blazeit: Optimizing declarative aggregation and limit queries for neural network-based video analytics. Proceedings of the VLDB Endowment pp. 533–546 (2019)
9. Kang, D., Emmons, J., et al.: Noscope: Optimizing neural network queries over video at scale. Proceedings of the VLDB Endowment pp. 1586–1597 (2017)
10. Kang, D., Gan, E., et al.: Approximate selection with guarantees using proxies. Proceedings of the VLDB Endowment pp. 1990–2003 (2020)
11. Kang, D., Guibas, J., et al.: Accelerating approximate aggregation queries with expensive predicates. Proceedings of the VLDB Endowment pp. 2341–2354 (2021)
12. Kang, D., Guibas, J., et al.: Tasti: Semantic indexes for machine learning-based queries over unstructured data. Proceedings of the 2022 International Conference on Management of Data pp. 1934–1947 (2022)
13. Kang, D., Romero, F., et al.: Viva: An end-to-end system for interactive video analytics. In: CIDR (2022)
14. Moll, O., Bastani, F., et al.: Exsample: Efficient searches on video repositories through adaptive sampling. 2022 Ieee 38th International Conference on Data Engineering pp. 3065–3077 (2022)
15. Romero, F., Hauswald, J., et al.: Optimizing video analytics with declarative model relationships. Proceedings of the VLDB Endowment pp. 447–460 (2022)
16. Russo, M., Hashimoto, T., et al.: Accelerating aggregation queries on unstructured streams of data. Proceedings of the VLDB Endowment pp. 2897–2910 (2023)
17. Ville, J.: Etude critique de la notion de collectif (1939)
18. Waudby-Smith, I., Ramdas, A.: Estimating means of bounded random variables by betting. Journal of the Royal Statistical Society Series B: Statistical Methodology pp. 1–27 (2023)
19. Xu, Z.D., Kakkar, G.T., et al.: Eva: A symbolic approach to accelerating exploratory video analytics with materialized views. Proceedings of the 2022 International Conference on Management of Data pp. 602–616 (2022)
20. Zhang, C., Cao, Q., et al.: A fast filtering mechanism to improve efficiency of large-scale video analytics. Ieee Transactions on Computers pp. 914–928 (2020)