Visualization-Oriented Progressive Time Series Transformation

Xin Chen*
Renmin University of China
Beijing, China
chenxin199634@gmail.com

Lingyu Zhang*
Shandong University
Qingdao, China
zhanglingyu@mail.sdu.edu.cn

Huaiwei Bao
Shandong University
Qingdao, China
bhuaiwei@gmail.com

Wei Lu
Renmin University of China
Beijing, China
lu-wei@ruc.edu.cn

Eugene Wu
ewu@cs.columbia.edu
Columbia University
New York City, New York, United States
ewu@cs.columbia.edu

Xiaohui Yu
xhyu@yorku.ca
York University
New York City, New York, United States
xhyu@yorku.ca

Yunhai Wang†
Renmin University of China
Beijing, China
wang.yh@ruc.edu.cn

ACM Reference Format:
Xin Chen, Lingyu Zhang, Huaiwei Bao, Wei Lu, Eugene Wu, Xiaohui Yu, and Yunhai Wang. 2026. Visualization-Oriented Progressive Time Series Transformation. In *Proceedings of Special Interest Group on Management of Data (SIGMOD '26).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

A Additional Results

This appendix provides supplementary material for the main paper, including: (i) a detailed illustration of the query mechanism Q, (ii) comprehensive results for initialization overheads, and (iii) additional results for interaction scenarios.

A.1 Detailed Illustrations

While Figure 5 in the main paper demonstrates the core mechanism of the transformation-aware query, the detailed illustrations provided here offer a more comprehensive view of the query process.

Figure 1 shows the full version of the example in Figure 5, expanded to three pixel columns. In this illustration, the first and third pixel columns require three iterations to reach convergence between the valid results (α) and the extreme scores (β). In contrast, the second column (**C2**) converges after only one iteration, leaving twelve nodes unqueried.

To demonstrate the performance of PIVOT on a realistic cross-series transformation, we construct a sophisticated cross-series operator $g(X_1, X_2, X_3, t) = \frac{X_3[t]-X_3[0]}{X_3[0]} - \frac{1}{2}\left(\frac{X_1[t]-X_1[0]}{X_1[0]} + \frac{X_2[t]-X_2[0]}{X_2[0]}\right)$ composed of three component functions:

- **Cumulative Return:** $g_1(X_i, t) = (X_i[t] - X_i[0])/X_i[0]$
- **Cross-Series Average:** $g_2(X_i, \ldots, X_j, t) = \frac{1}{j-i+1}\sum_{k=i}^{j} X_k[t]$
- **Point-wise Subtraction:** $g_3(X_i, X_j, t) = X_i[t] - X_j[t]$

Figure 2 illustrates the query evaluation process for this compositional operator applied to three time series. For the first pixel column (C1), the initial valid result range, computed from its boundary nodes as $[\alpha^1_{\min}, \alpha^1_{\max}] = [-2.40, 0.80]$, already covers the maximum scores of all inner-column nodes (i.e., η_4 and η_{20}). Consequently, the upper bound of the result is already completed, and the subsequent iteration only needs to retrieve nodes to refine the lower bound. This targeted refinement prunes 14 nodes from the query process entirely. This example demonstrates that rapid convergence and effective node pruning are maintained even when evaluating complex, compositional functions.

A.2 Comprehensive Results for Initialization Overheads

Rather than using the original OM^3 preprocessor implemented in JavaScript, PIVOT employs a reimplementation in C++, which significantly improves preprocessing efficiency. Figure 3 presents the initialization overheads for all tested datasets on log-log scales. The results confirm that both initialization time and peak memory usage scale linearly with data size.

Upon closer inspection, a secondary observation emerges for initialization time: real-world datasets tend to require more processing time than synthetic datasets of a comparable point count. This performance difference can be attributed to the greater complexity in real-world data distributions. For instance, the *Soccer* dataset (19.10s) takes approximately 3× longer to initialize than the similarly-sized *Syn40M* dataset (6.39s). Despite these variations, the overall performance remains practical, with the largest dataset

* Xin Chen and Lingyu Zhang are joint first authors.
† Yunhai Wang is the corresponding author.

SIGMOD '26, Bengaluru, India

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/XXXXXXX.XXXXXXX

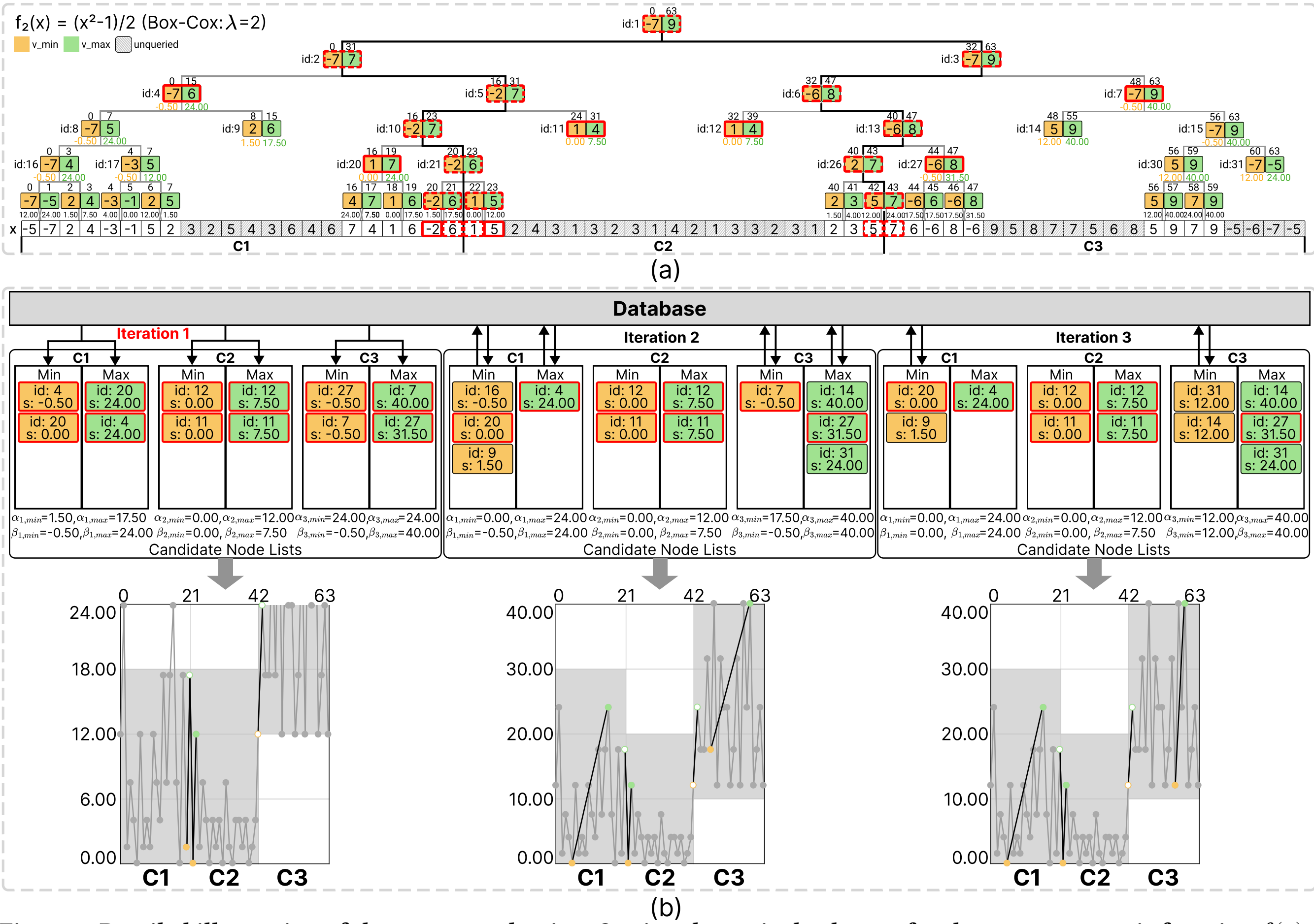

Figure 1: Detailed illustration of the query mechanism Q using three pixel columns for the non-monotonic function $f(x) = (x^2 - 1)/2$. Each column maintains its query status independently. (a) TAT representing the search paths of Q, with node scores indicated below. (b) Evolution of the candidate node lists (α and β) during the query process, shown alongside the corresponding progressive visualizations. The gray ground truth visualization is provided for reference.

(*Syn5B*) completing initialization in approximately 11 minutes. In contrast, peak memory usage is less affected by data complexity, showing no significant disparity between real-world and synthetic datasets. This indicates that memory consumption is primarily driven by data volume rather than its underlying distribution.

A.3 Additional Results for Interaction Scenarios

We present the detailed results for the two datasets used in the interaction scenario. Compared to Figure 10 in the main paper, Figure 4 also includes memory usage. As the query plan progresses, DuckDB [1] gradually consumes more memory, while PIVOT maintains a consistently low and stable memory footprint. Although DuckDB performs active memory release, this process may introduce performance fluctuations, as observed in Figure 4e and 4f.

References

[1] Mark Raasveldt and Hannes Mühleisen. 2019. Duckdb: an embeddable analytical database. In *Proceedings of the 2019 international conference on management of data*. 1981–1984.

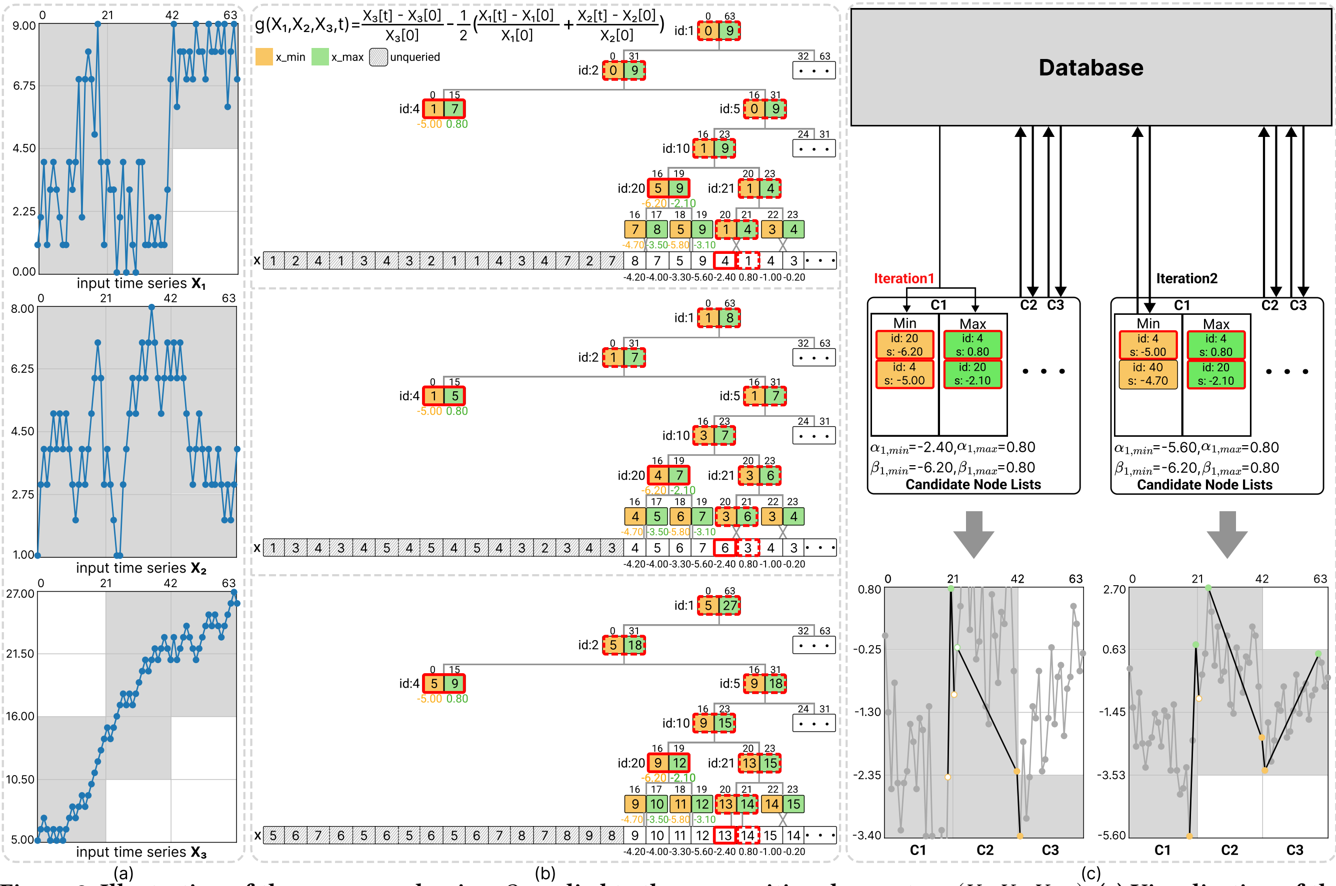

Figure 2: Illustration of the query mechanism Q applied to the compositional operator $g(X_1, X_2, X_3, t)$. (a) Visualization of the input time series. (b) The corresponding TATs for the first pixel column under this operator. (c) Evolution of the candidate node lists (α and β) during the query process.

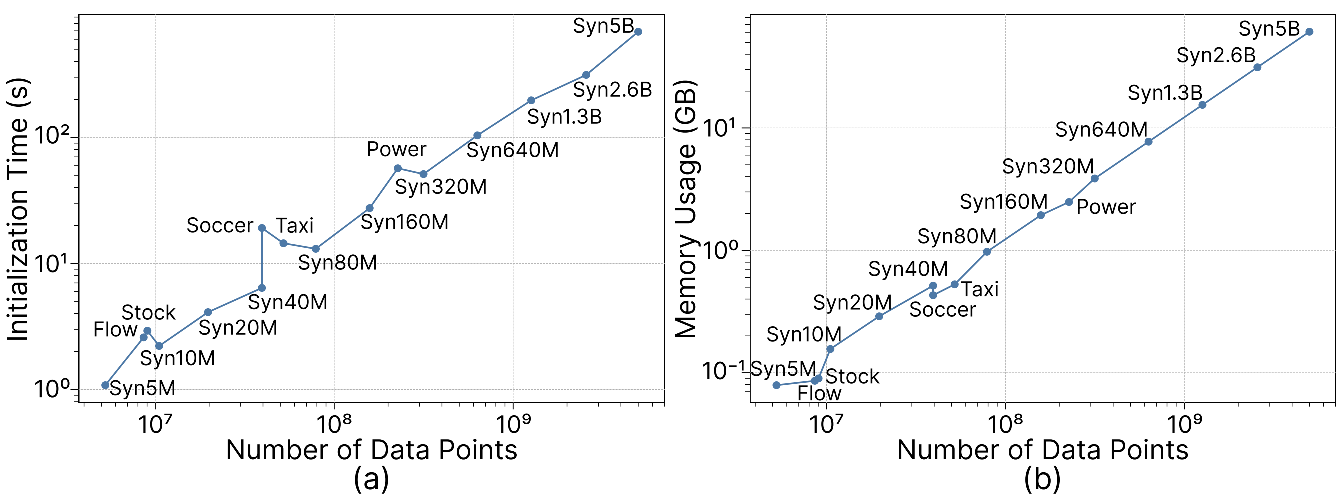

Figure 3: Performance of PIVOT's offline preprocessing stage in terms of (a) initialization time and (b) peak memory usage.

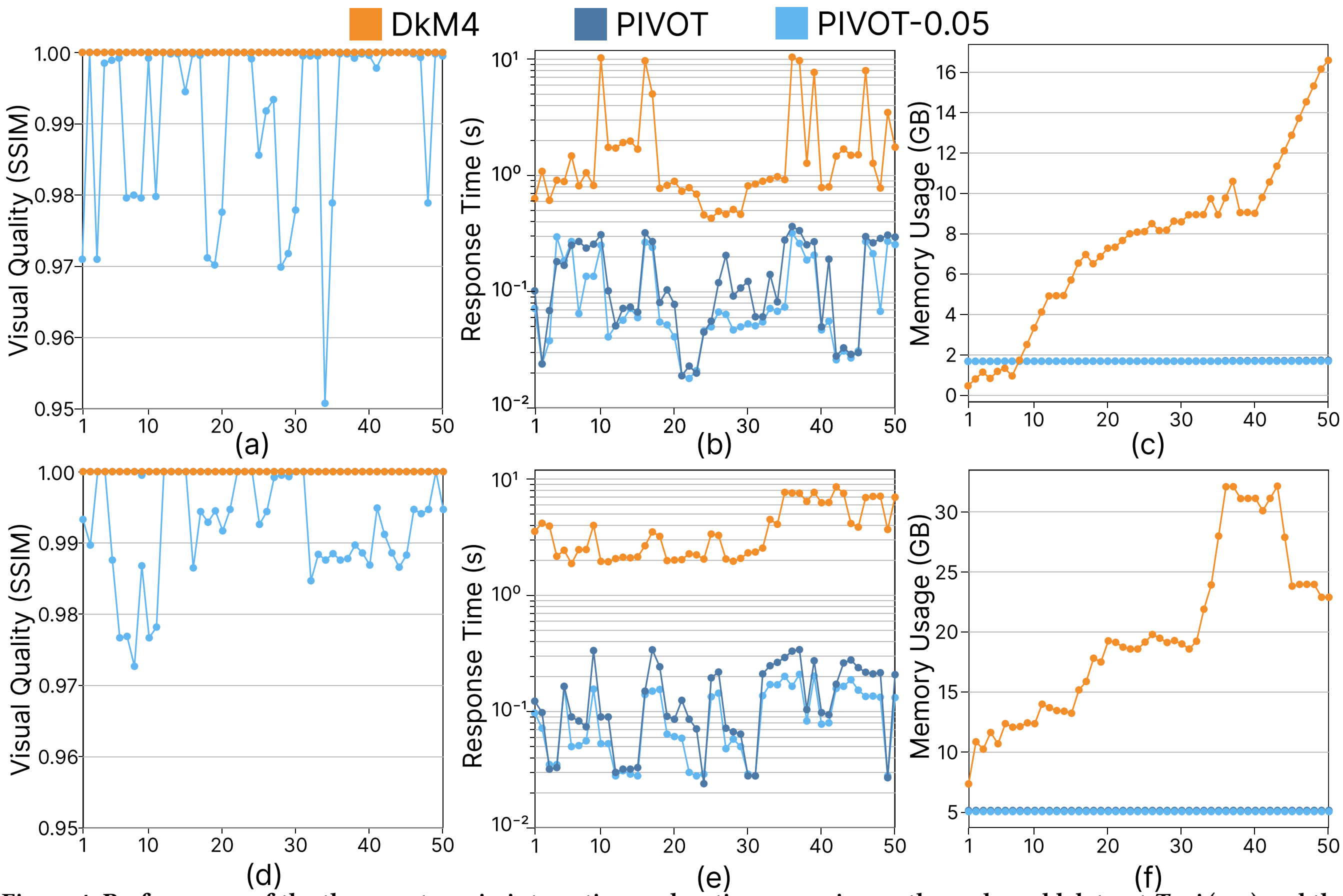

Figure 4: Performance of the three systems in interactive exploration scenarios on the real-world dataset *Taxi* (a–c) and the synthetic dataset *Syn5B* (d–f): (a, d) SSIM scores, (b, e) response times, and (c, f) memory usage.