VideoChef: Efficient Approximation for Streaming Video Processing Pipelines

Ran Xu[α]

Jinkyu Koo[α], Rakesh Kumar[α], Peter Bai[α]

Subrata Mitra[β], Sasa Misailovic[γ], Saurabh Bagchi[α]

[α] PURDUE UNIVERSITY [β] Adobe Research [γ] ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Why approximate computing in video streaming apps?

- Video streaming applications require low-latency processing
- Devices are resource constrained
- Human perception can tolerate slight errors in videos

Typically 30FPS → 33 ms for each frame

Background: Approximation techniques and parameters

- Loop perforation:

```
for (i = 0; i<n; i = i + approx_level)
  result = compute_result();
```

- Loop memorization:

```
for (i = 0; i<n; i = i ++)
  if(i % approx_level == 0)
    cached_result = result = compute_result();
  else
    result = cached_result;
```

Approximation parameters = approx_level

- 1 = Exact execution
- Higher value => More approximate

Execution saving $\approx 1 - \frac{1}{approx\_level}$

6 -> up tp 83%

Quality degradation is unknown

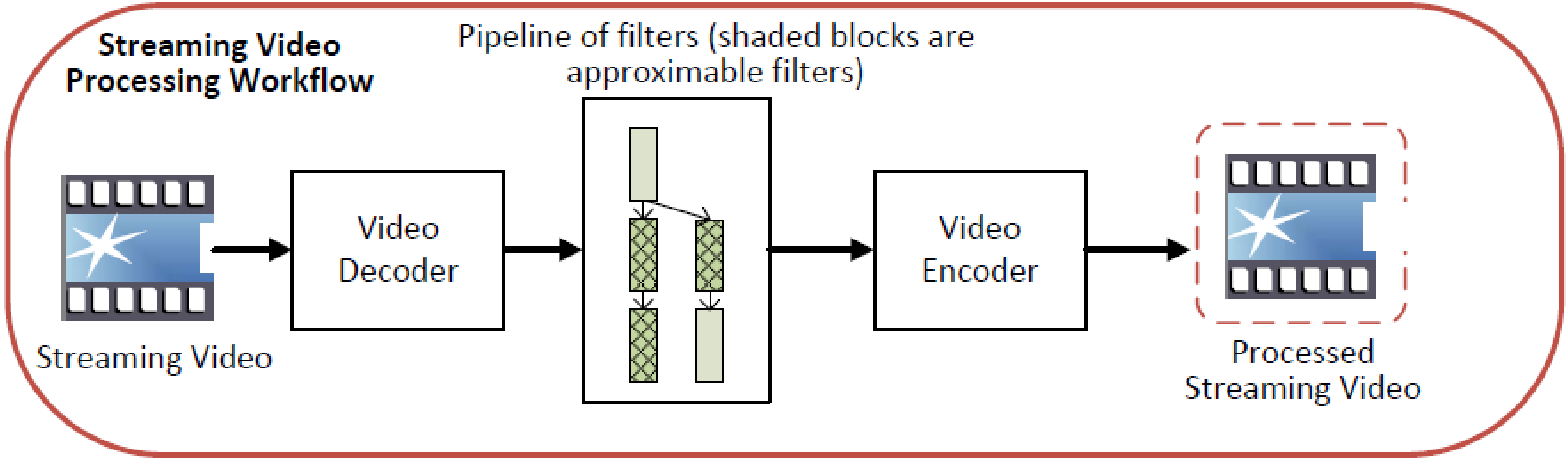
Quality metric for videos

- PSNR (Peak Signal to Noise Ratio)
  - **Higher PSNR means higher quality/lower error**
  - The approximate output with regard to the exact output
  - 30dB means RMSE is 6% of the mean pixel value and 20dB means 20%.
  - With easy-to-understand meaning and easy-to-choose threshold

$$PSNR = \frac{1}{K}\sum_{k=0}^{K-1} 20 \times log_{10}\frac{MaxValue}{\sqrt{MSE(Z_k, Y_k)}}$$

- SSIM, FSIM
  - Guarantee the quality ordering but lacking obvious meaning and threshold.
  - Slow to compute

A video processing workflow



Research questions

1) Does one approximation level apply to all frames?
2) How to determine optimal approximation level in a data-aware manner?
3) How to control online overhead of determining optimal approximation level?

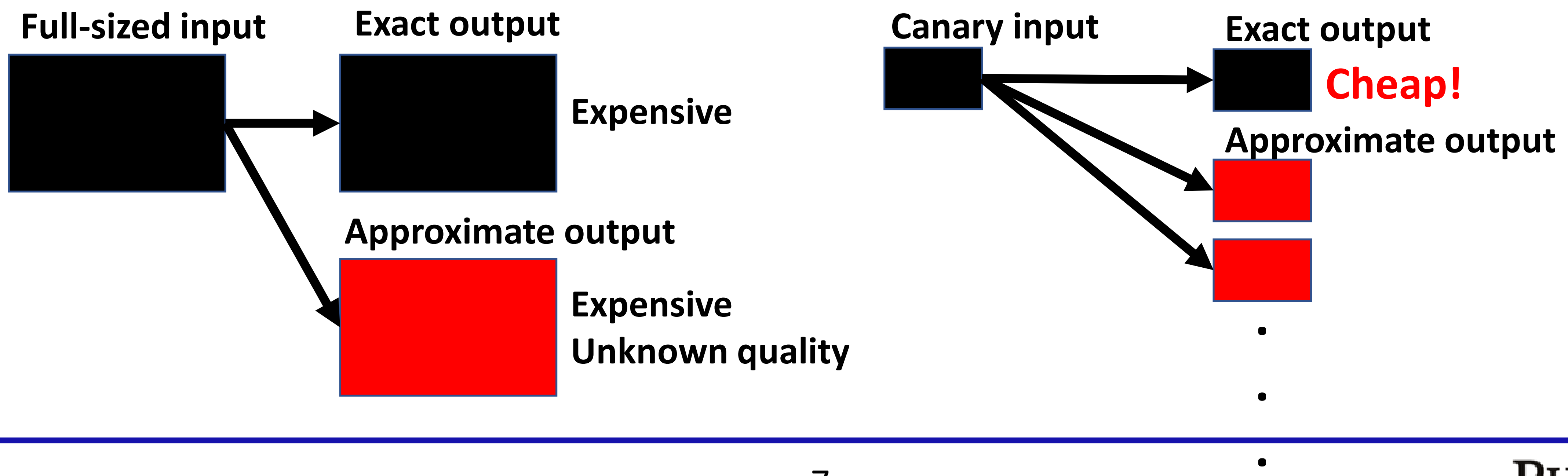
Prior Work

IRA[1]	Video proc. w/ approx. Canary input to search	• (+) Parameters for each input • (-) Biased error metric • (-) Not for streaming application
Static approx	Video proc. w/ approximation	• (-) Too conservative para. for all input.
Exact	Video processing	• (-) Slow

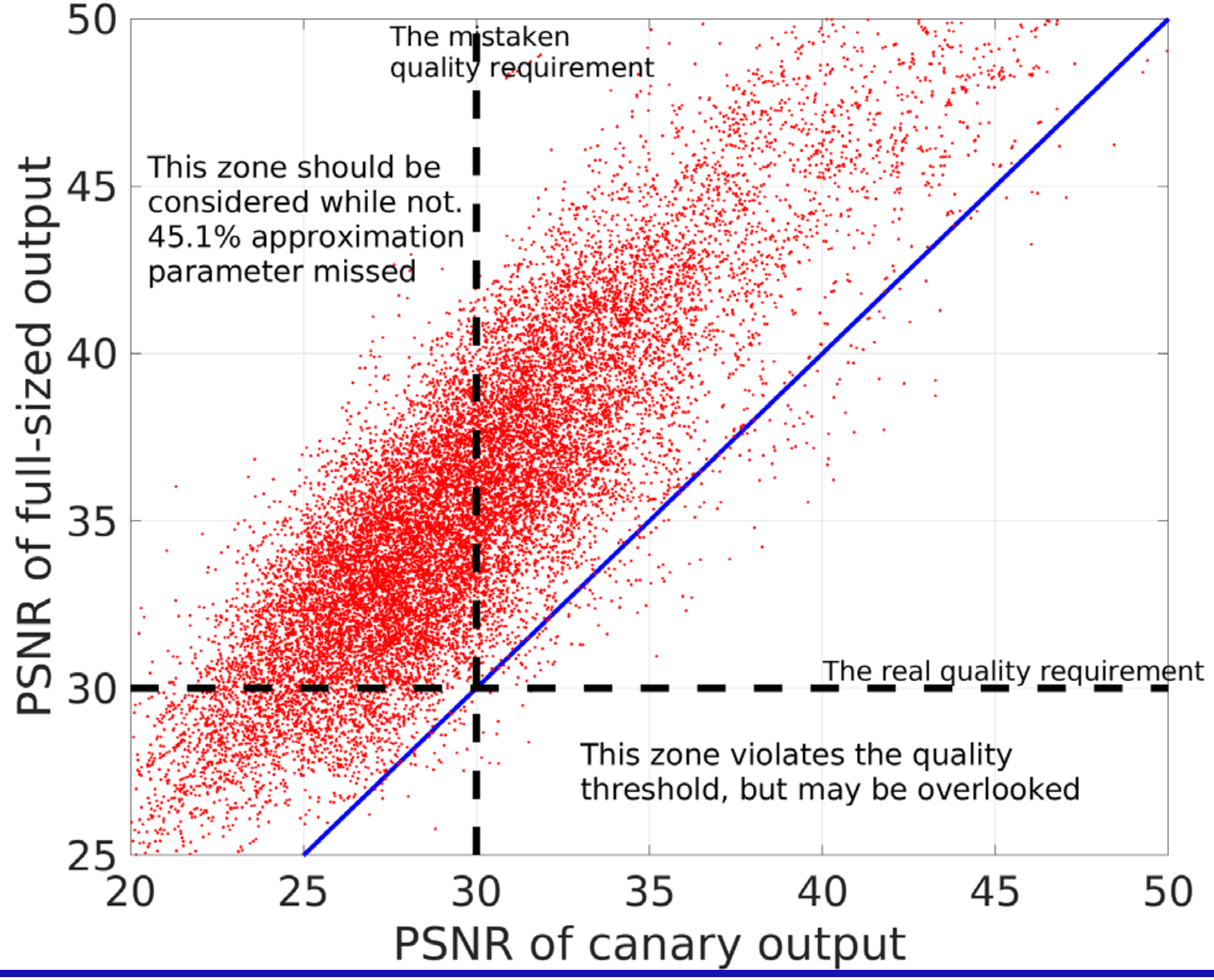
[1] Laurenzano, M. A., Hill, P., Samadi, M., Mahlke, S., Mars, J., & Tang, L. (2016). Input responsiveness: using canary inputs to dynamically steer approximation. *ACM SIGPLAN Notices*, *51*(6), 161-176.

Why use a canary input

- Provides an estimate of the output quality
- Enables data-aware approximation



Problem 1 – Canary output quality is biased



- Full-sized output quality is higher than canary one for over 98% approximation setting.
- 45.1% approximation setting is ignored due to the mistaken quality threshold.

Problem 2 – Online overhead really matters

Sources of online overhead

1) Generating canary input
2) Searching approximation parameters
3) Calculating quality metric (PSNR)
4) Correcting quality bias

- Bottom line: online overhead should never outweigh the savings of approximation

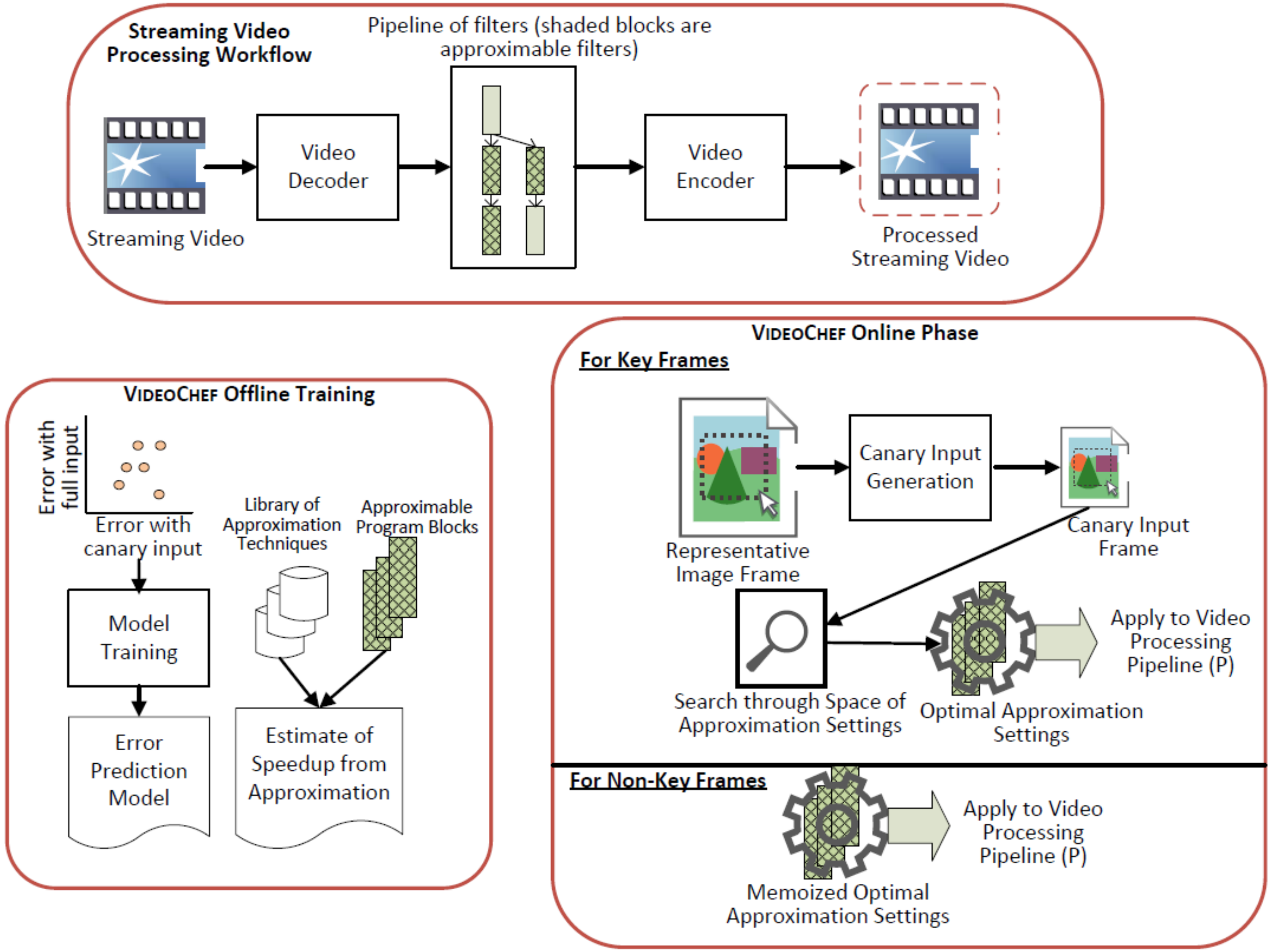
Progress of approximation in video processing

Oracle	Video proc. w/ approx. Optimal parameters	• (+) Never violate quality threshold • (+) Low processing time • (+) Low overhead
VideoChef[2]	**Video proc. w/ approx.** **Canary + Error mapping + Sampling**	• **(+) Unbiased error metric** • **(+) Close to optimal parameters** • **(+) Overhead controlled**
IRA[1]	Video proc. w/ approx. Canary input to search	• (+) Parameters for each input • (-) Biased error metric
Static approx	Video proc. w/ approximation	• (-) Too conservative para. for all input.
Exact	Video processing	• (-) Slow

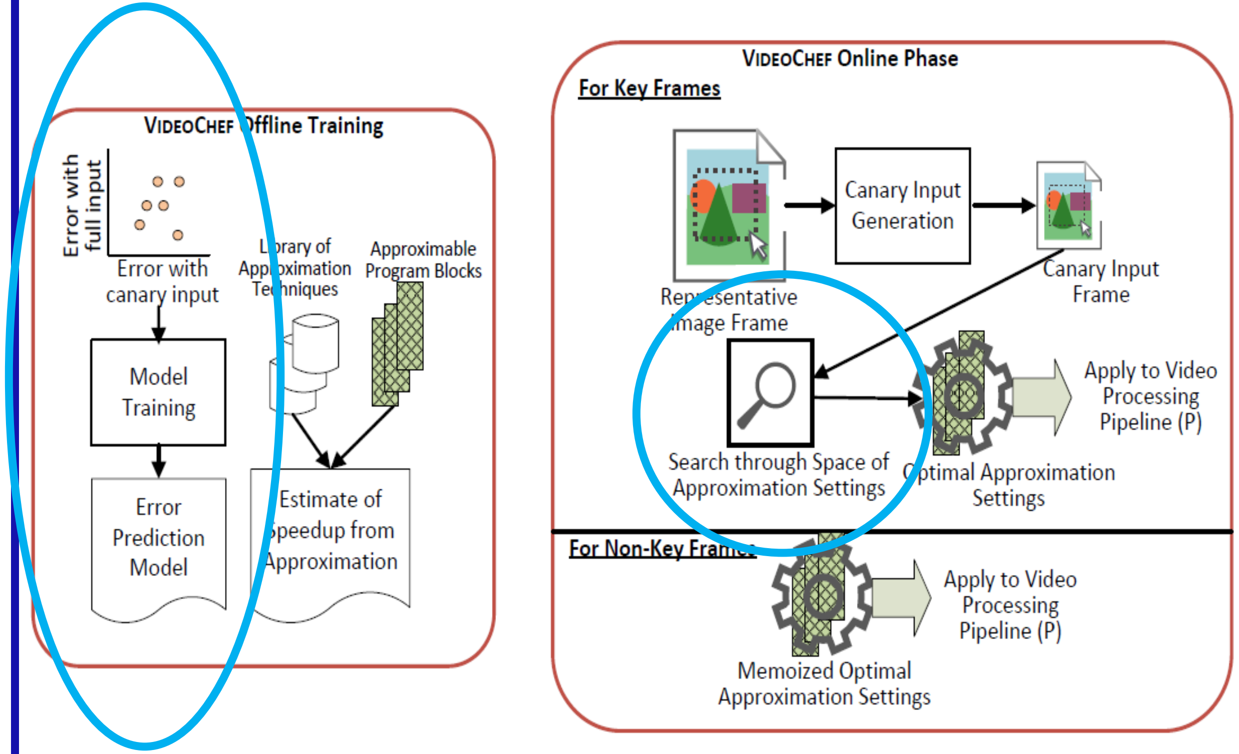
[1] Laurenzano, M. A., Hill, P., Samadi, M., Mahlke, S., Mars, J., & Tang, L. (2016). Input responsiveness: using canary inputs to dynamically steer approximation. *ACM SIGPLAN Notices*, *51*(6), 161-176.

[2] Xu, R., Koo, J., Kumar, R., Bai, P., Mitra, S., Misailovic, S., & Bagchi, S. (2018, July). VideoChef: Efficient Approximation for Streaming Video Processing Pipelines. In 2018 USENIX Annual Technical Conference (USENIX ATC 18). USENIX Association}.

End-to-end system workflow

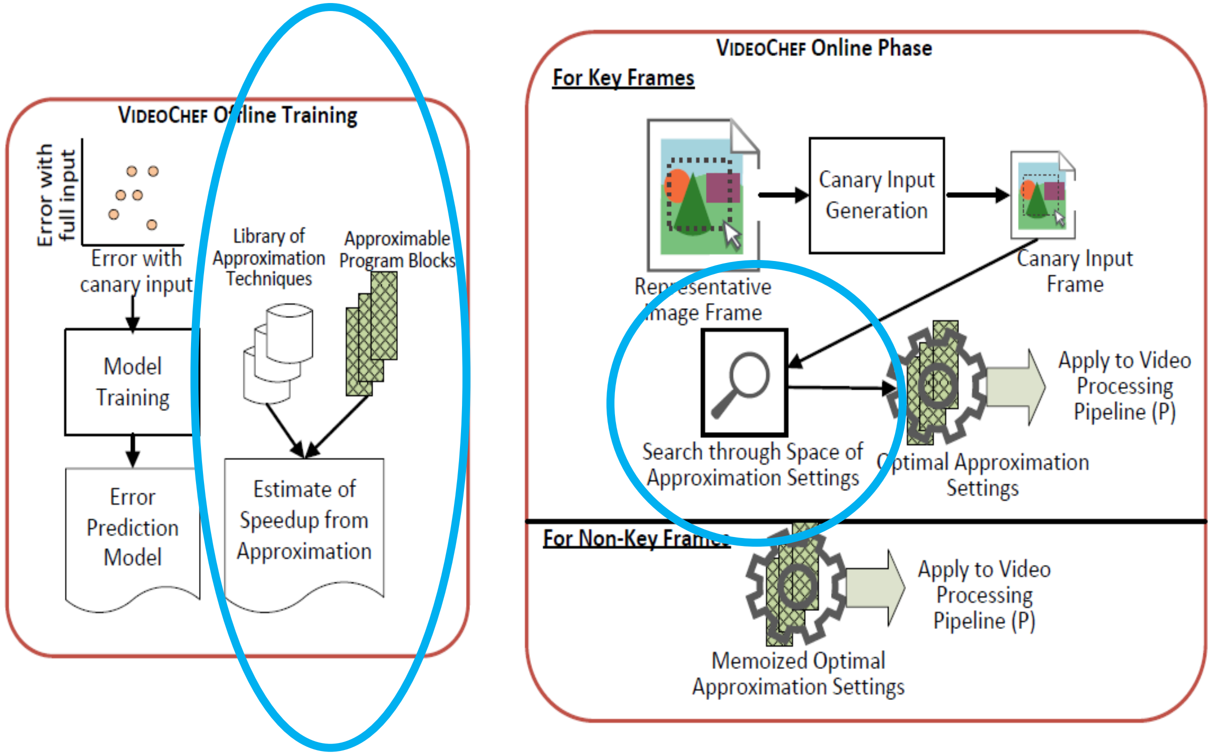


Key Designs



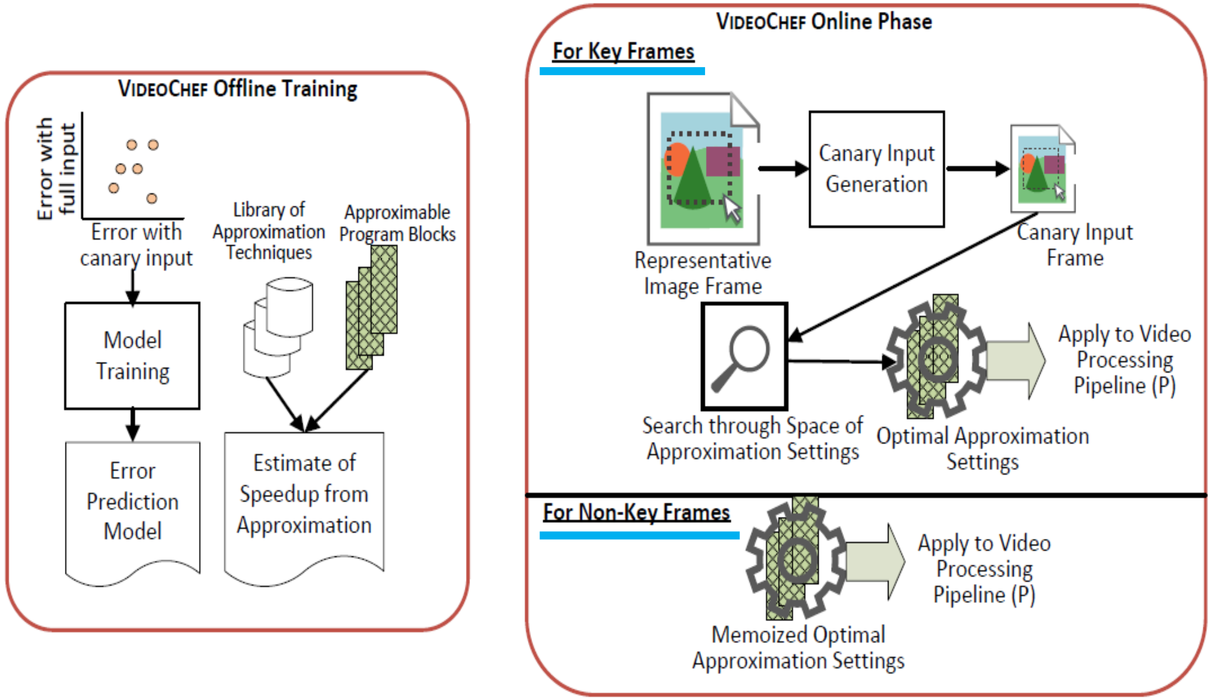
- Error mapping model – to map the quality metric of canary output to that of full-sized output
- Searching policy – to approach the optimal approximation setting that achieve lowest execution time while guaranteeing quality
- Sampling policy – to identify the key frames that redo the searching for approximation parameters.

Key Designs



- Error mapping model – to map the quality metric of canary output to that of full-sized output
- Searching policy – to approach the optimal approximation setting that achieves the lowest execution time while guaranteeing quality
- Sampling policy – to identify the key frames that redo the searching for approximation parameters.

Key Designs



- Error mapping model – to map the quality metric of canary output to that of full-sized output
- Searching policy – to approach the optimal approximation setting that achieves lowest execution time while guaranteeing quality
- Sampling policy – to identify the key frames that redo the searching for approximation parameters.

PURDUE UNIVERSITY

Error mapping model

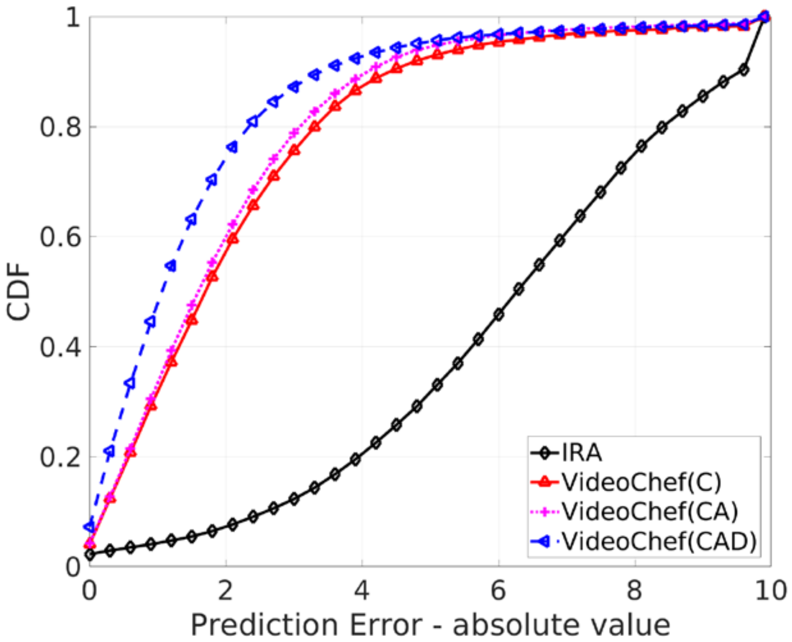
- Given a full-sized frame X^F, the canary frame X^C, the canary output quality C and a set of approximation parameter A.
- We want to predict the full-sized output quality F

- No prediction: $F = C$ (IRA)
- C model – aware of canary quality

  $F = w_0 + w_1 \times C + w_2 \times C^2$
- CA model – C model plus approximation parameters

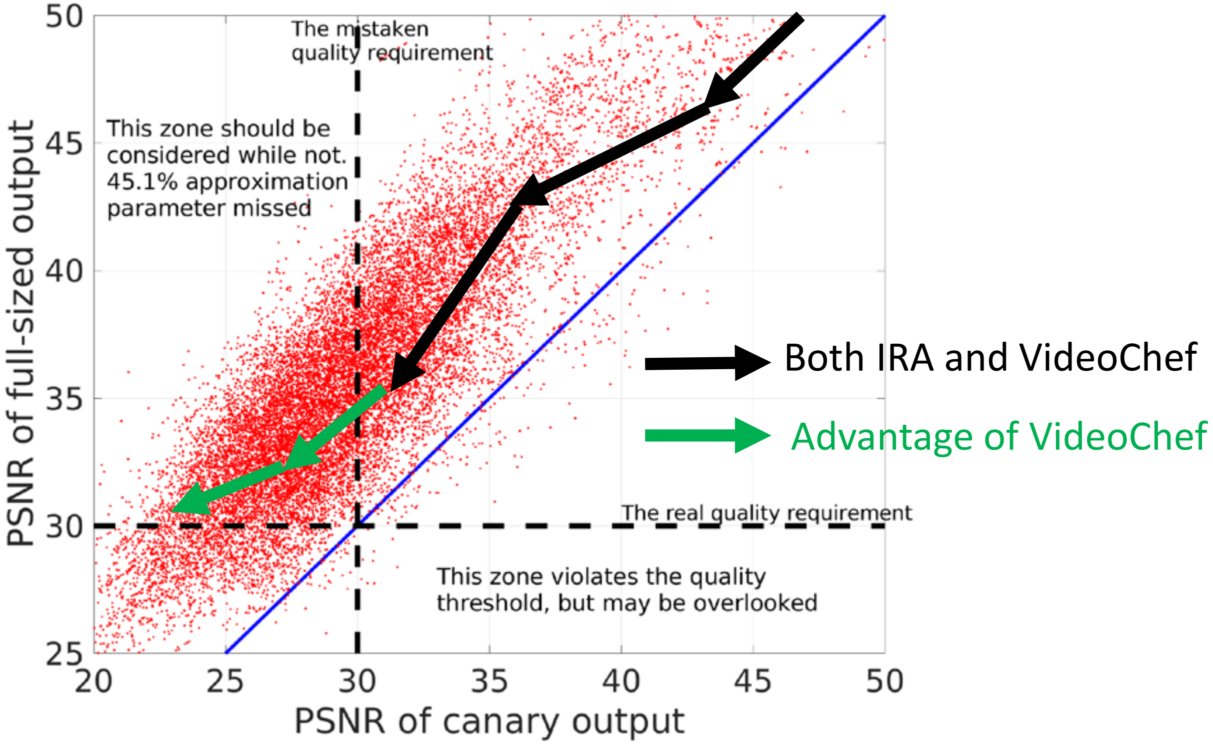
  $F = \vec{w} \cdot (1, C, \vec{A})$
- CAD model – CA model plus feature vectors (row diff.)

  $F = \vec{w} \cdot (1, C, \vec{A}, \vec{D})$



Searching policy

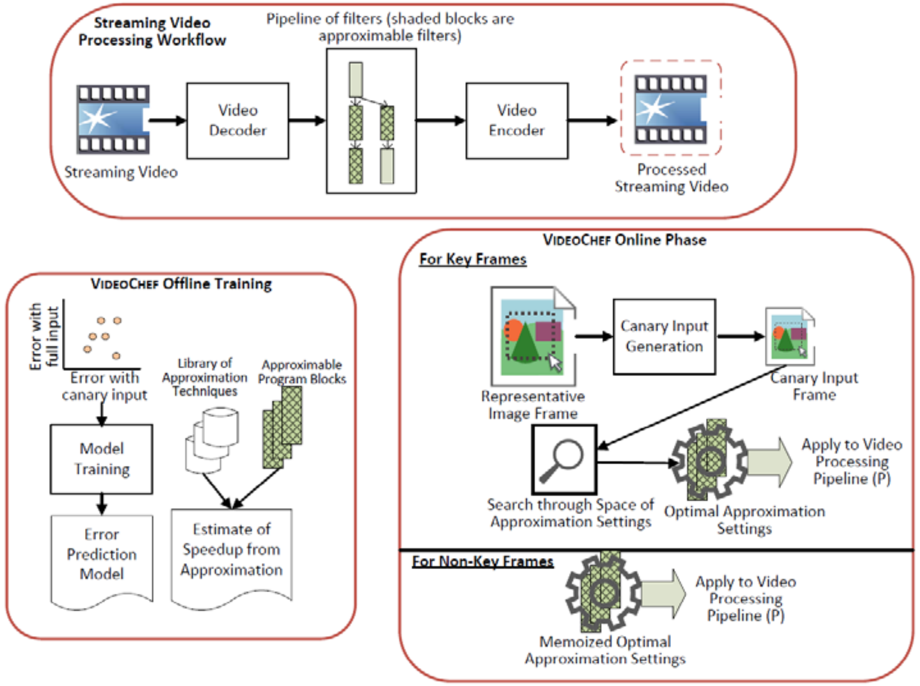
- Start from (1,1,1), then increase by 1 in each dimension and follow the least-error path until the predicted quality of full output reaches the threshold.



PURDUE UNIVERSITY

Sampling policy to reinitiate search for optimal settings

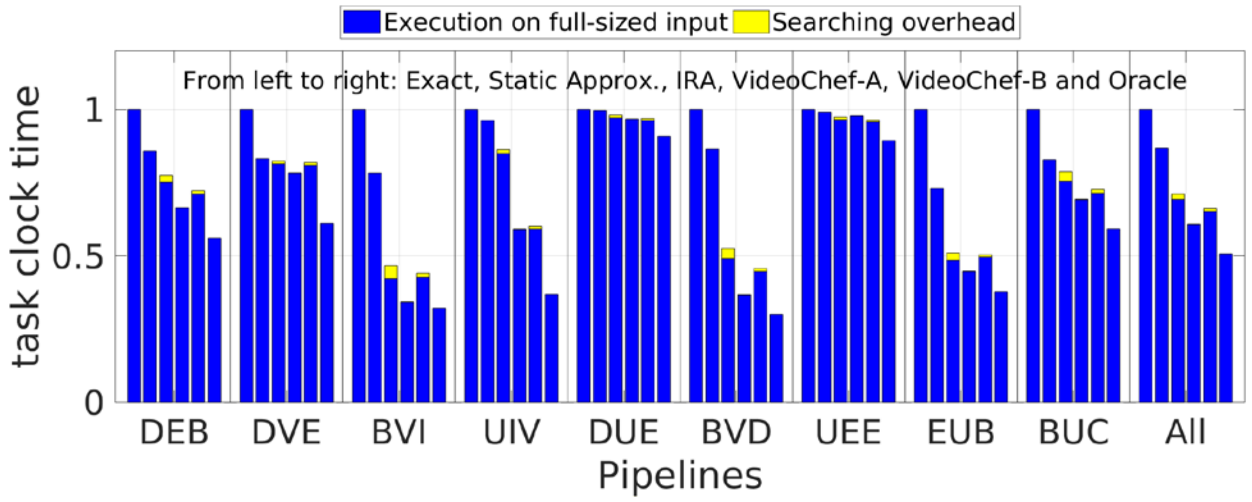
- I-frames in MPEG-4 videos
- Scene change detector (lightweight frame-difference based classifier)



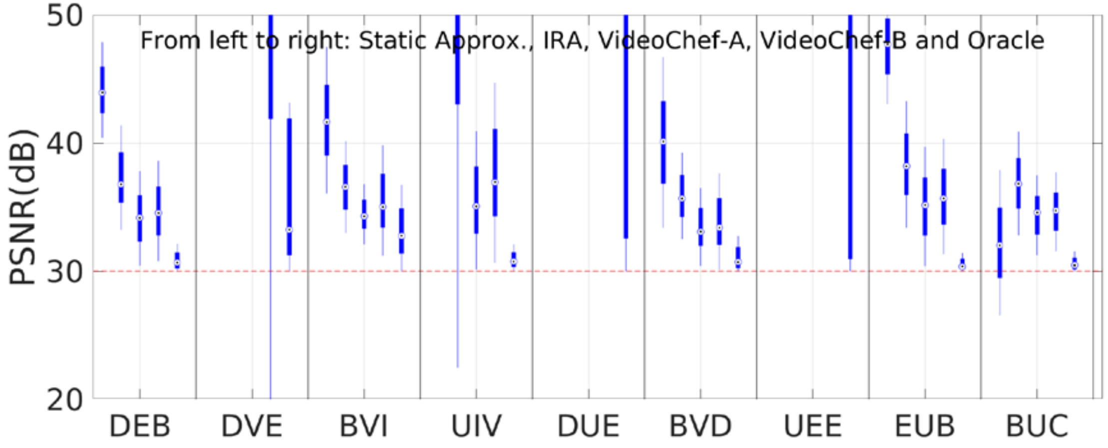
Evaluation

- 106 Youtube videos w/ 10 video filters and 9 3-stage filter pipelines
- Loop perforation and memoization, each w/ 6 approximation levels
- Comparing 6 configurations (2 variants of VideoChef) and 2 PSNR thresholds (30dB and 20dB)
  1) Exact execution
  2) Static approximation
  3) IRA
  4) VideoChef – I-frame sampling
  5) VideoChef – Scene change detector
  6) Oracle

Evaluation – 30dB tight quality constraint



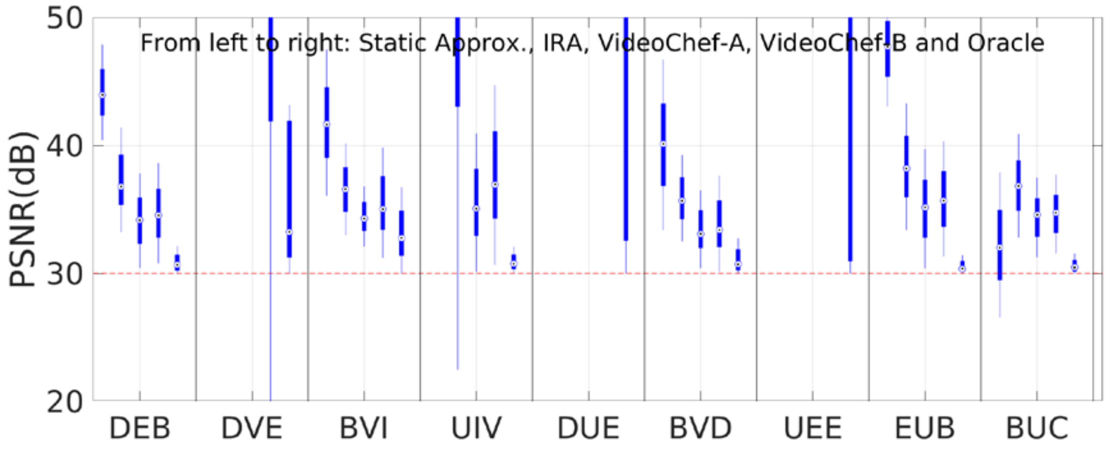
Execution time is reduced by
39.1% over exact execution
29.9% over static approximation
14.6% over IRA and
within 20% of Oracle

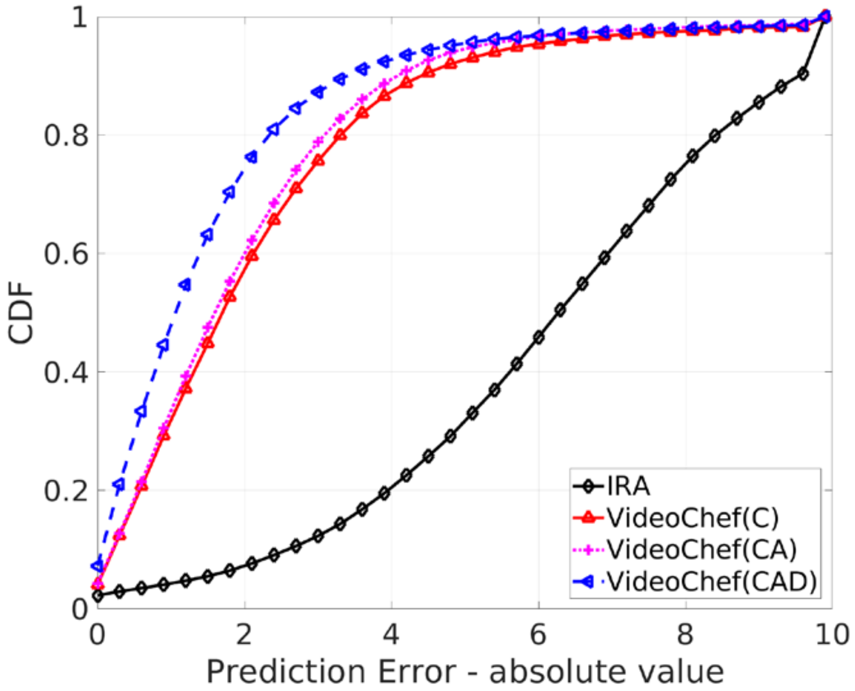


Tracks the Oracle quality and the user specified quality threshold, violation < 5%

Evaluation – 30dB tight quality constraint

The CDF of prediction error helps to choose a good in-application threshold on top of user's hard threshold.





Tracks the Oracle quality and the user specified quality threshold, violation < 5%

User Perception Study

We asked 16 users to watch 16 side-by-side video pairs and tell difference between them.

VideoChef video

Oracle video

Degree of difference	Percentage
No difference	58.59%
Little difference	34.77%
Large difference	6.64%
Total difference	0

Conclusion

- VideoChef: A system for performance and accuracy optimization of video streaming pipelines in a data-dependent manner
- Predictive model to accurately estimate the quality degradation in the full-sized output from the canary output
- Efficient and incremental search technique for the optimal approximation setting to reduce the overhead of the search process
- Quantitative evaluation and user study

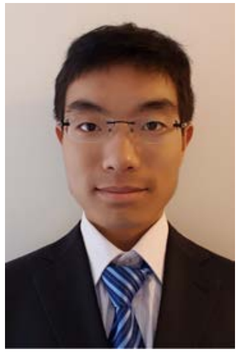
Insights

- Determination of optimal approximation setting in a streaming application is challenging because the setting may change during the stream. It is important to ensure that the cost of searching for the optimal parameter does not outweigh the benefit of the approximate execution.
- Quality difference between canary output and full-sized output is not negligible.
- Bringing in domain knowledge (I-frames for MPEG video) can be a great help to reduce the overhead of the approximation technique.

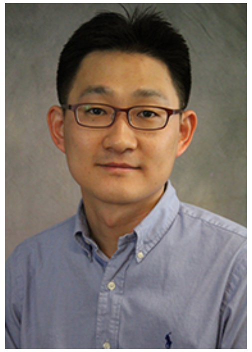
Questions?

- Thank you!

  --- All authors

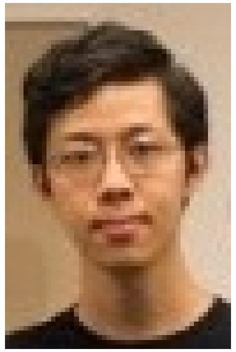


Ran Xu



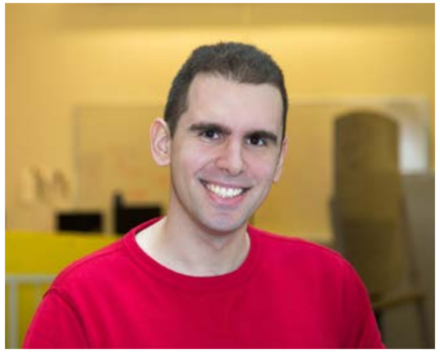
Jinkyu Koo

Rakesh Kumar



Peter Bai

Subrata Mitra



Sasa Misailovic

Saurabh Bagchi

PURDUE UNIVERSITY — PURDUE UNIVERSITY — Adobe Research — ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN — PURDUE UNIVERSITY