ZGaming:
Zero-Latency 3D Cloud Gaming by Image Prediction

201933966 양시훈

ACM SIGCOMM 2023

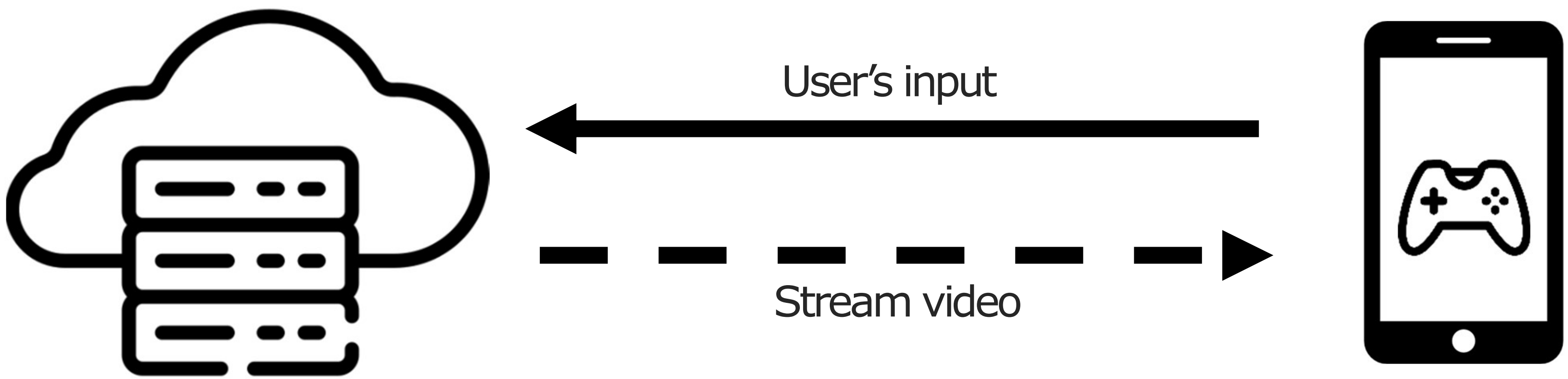
1. INTRODUCTION

Introduction

Server(Cloud)

클라우드 서버에서
resource를 빌려 원격으로
video만 streaming

Client(End Device)



- Run the game
- Computation & Rendering

Introduction

"CLOUD GAMING"

클라우드 게임 서비스 본격화, 게임시장 지각변동 예고

콘솔로 치고 받은 소니·MS, 클라우드 게임 시장 잡아야 진정한 '겜심' 승자

신대륙 꿈꿨던 클라우드 게임 시장...MS만 '맑음'

구글 스타디아 실패 인정 "2023년 1월 서비스 종료"

스토어는 이미 폐쇄…구매한 게임 전액 환불

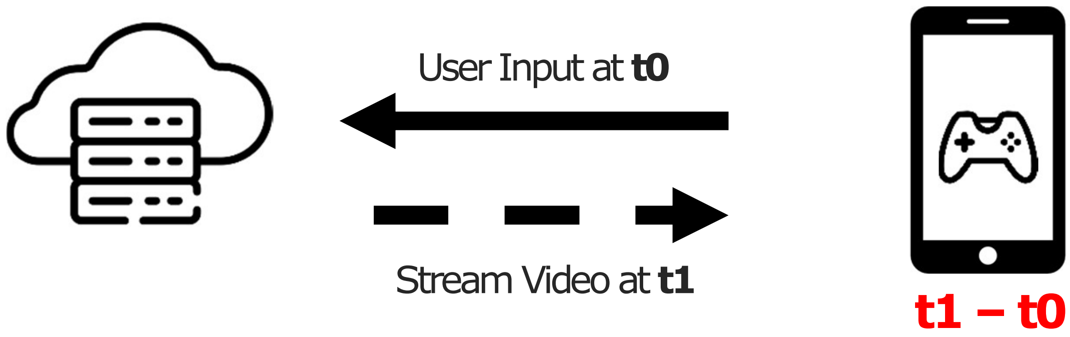
Introduction

HOME > 디지털 혁신 > VR AR & 게임

'갤럭시 노트 20' 시리즈서 'Xbox 게임패스' 클라우드 게임 즐긴다

김형근 게임전문기자 | 승인 2020.08.06 15:21 | 댓글 0

Main Problem : Interactive Latency



- Delay between a user's action and the response on device
- Ideal : <60ms (≈ local game)

✓ Obstacles : Network congestion, Delay variations…

Introduction

Main Problem : Interactive Latency

VS

Introduction

Solution

Image-based Prediction - DIBR

▶ Depth Image Based Rendering (DIBR)

✓ Solution to reduce interactive latency

이전 frame을 기반으로 다음 frame을 예측

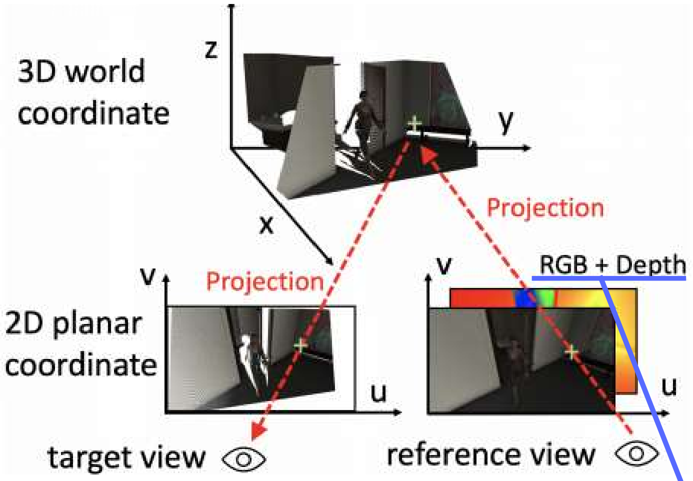
If frame doesn't arrive at the expected time

▶ Client use DIBR to predict the frame based on the latest received frame

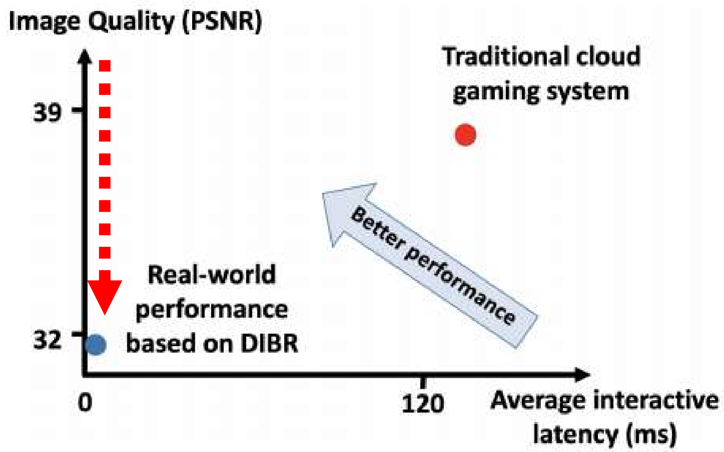
Solution

Image-based Prediction - DIBR

▶ Depth Image Based Rendering (DIBR)



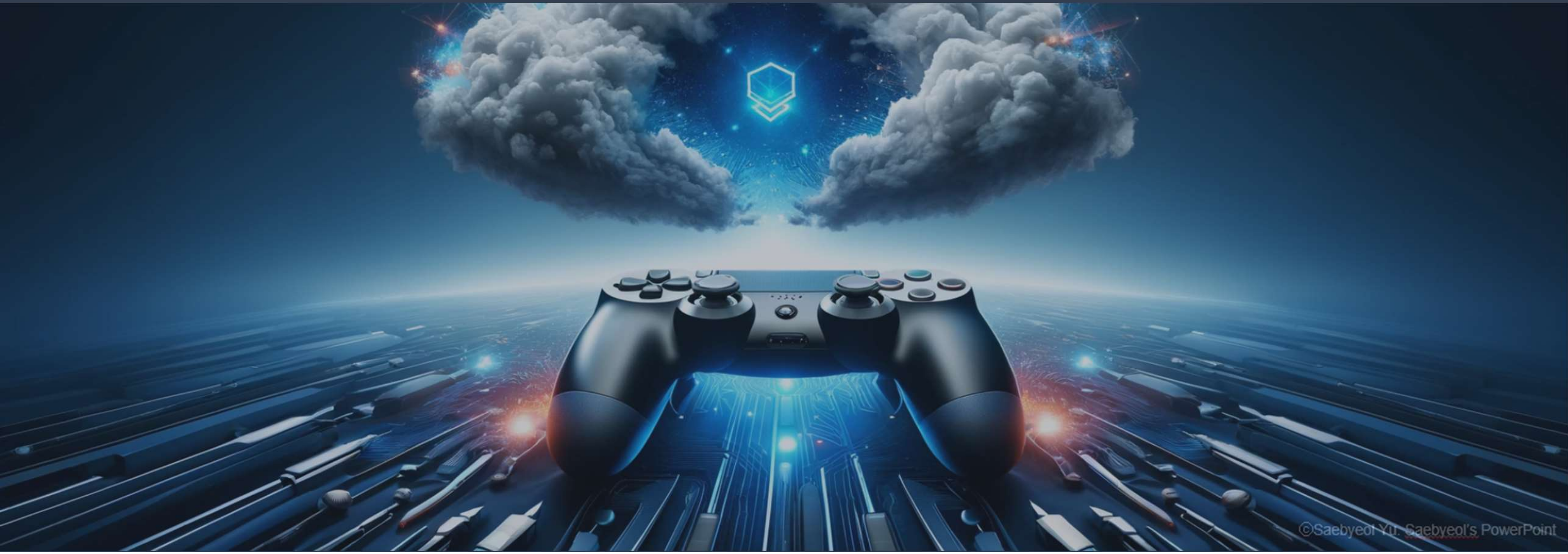
Get from 'Graphics API Hooking'
▶ No need to modify the logic



✓ **Interactive latency ↓**

✓ **But… Image Quality ↓ too!**

2. MOTIVATION



Motivation

" **How to Achieve** **Interactive Latency ↓** **Image Prediction Quality ↑** "

3 technical challenges

1) Artifacts in the predicted images
2) DIBR doesn't work for dynamic objects in the reference frame
3) Trade-off between video bitrate and prediction performance

↓

"Insights of ZGaming"

3 technical challenges

1) Artifacts(holes) in the predicted images
missing fixel
2) DIBR doesn't work for dynamic objects in the reference frame
3) Trade-off between video bitrate and prediction performance

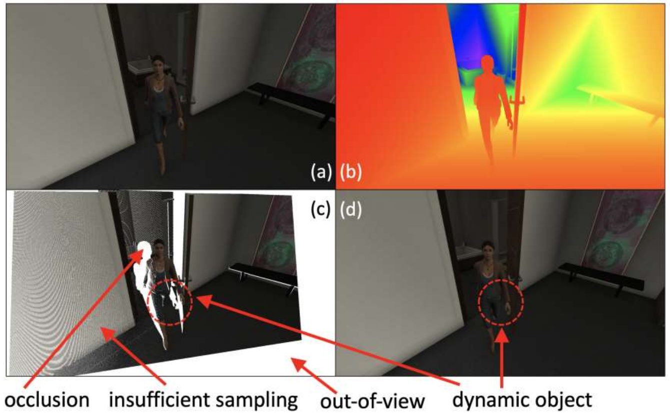


Figure 2- (c)
✓ Artifacts = 'missing pixel'
✓ Due to occlusion, out-of-view…

Insight 1 :

Can be improved by history frames

Motivation

3 technical challenges

1) Artifacts(holes) in the predicted images
2) DIBR doesn't work for dynamic objects in the reference frame
3) Trade-off between video bitrate and prediction performance

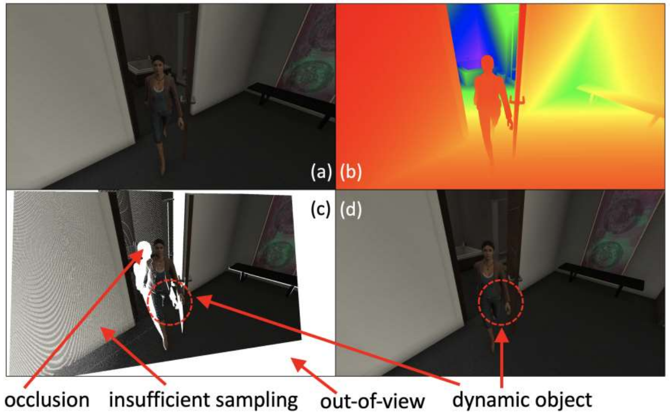


Figure 2- (c),(d)
- ✓ Moving character → different from ground truth
- ✓ Dynamic objects → constantly changing

Insight 2 :

Dynamic objects can be predicted by LSTM

3 technical challenges

1) Artifacts(holes) in the predicted images
2) DIBR doesn't work for dynamic objects in the reference frame
3) Trade-off between video bitrate and prediction performance

[Bitrate ↑ $\xrightarrow[\text{trade-off}]{}$ Prediction performance ↓ / Interactive latency ↑]

Insight 3 :

Can be optimized by adaptive streaming

Motivation

3 technical challenges

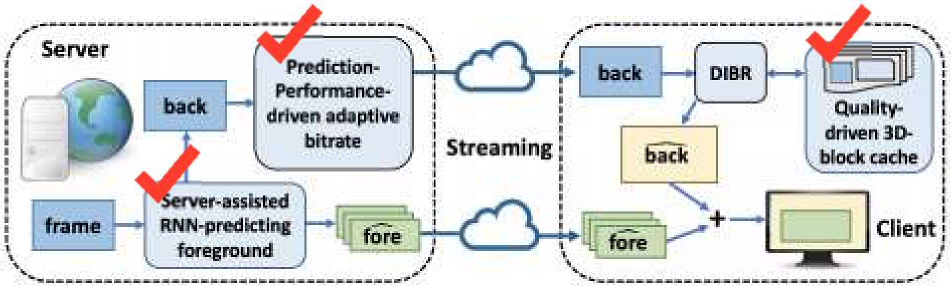
1) Artifacts in the predicted images
2) DIBR doesn't work for dynamic objects in the reference frame
3) Trade-off between video bitrate and prediction performance

ZGaming

I. " Q3B Cache "

II. " Server-assisted LSTM prediction "

III. " Prediction-performance-driven adaptive bitrate "

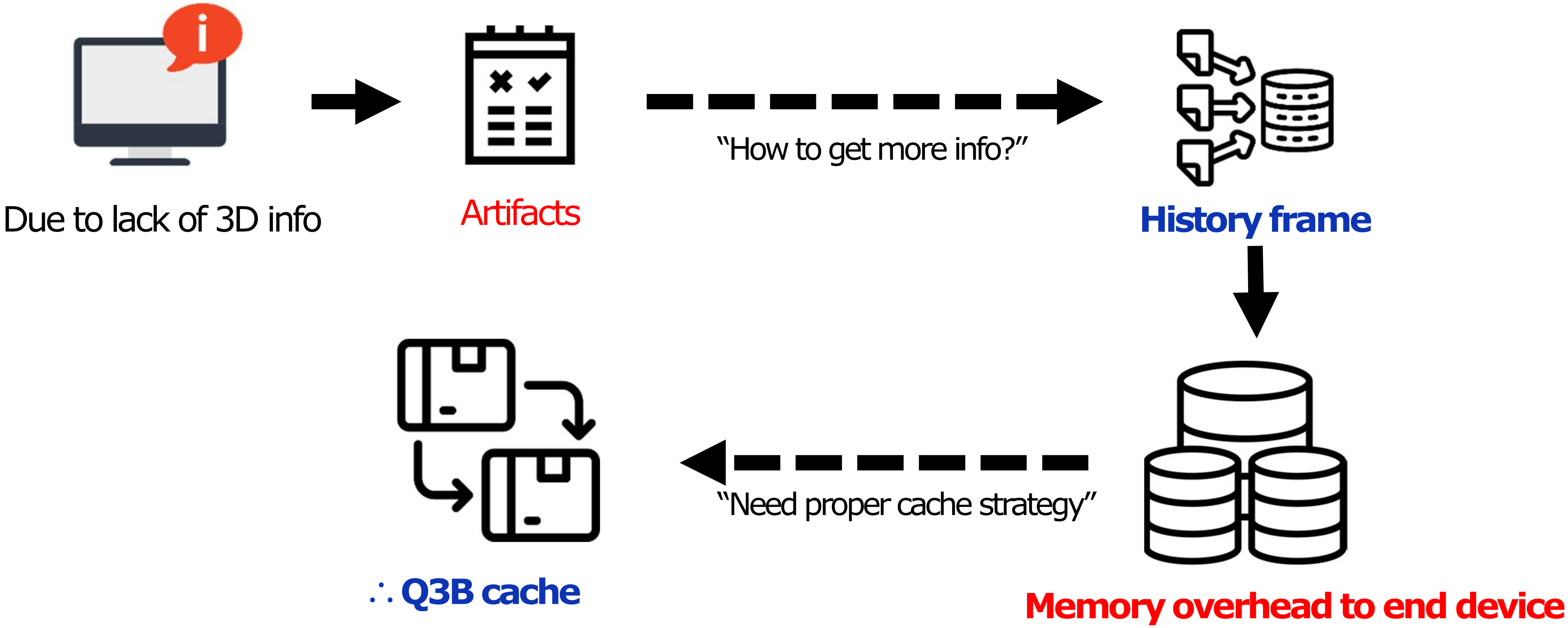


3. ZGAMING

(1) Q3B Cache

Insight 1 :

Prediction performance for DIBR can be improved by history frames



Cache strategy

Legacy methods ?

▶ Reducing Redundancy → "Still too much occupancy"

▶ Compression → "Impose computational pressure"

▶ Traditional Cache Replacement (LRU,FIFO)

→ "Each frames have different utility on prediction quality"

< 3 Factors >

1) The lighting effect of 3D world

2) The competition of inpainting algorithm

3) Influence of user behavior

(1) Q3B Cache

Quality-driven 3D-block caching strategy

Input all video frames and compute the utility value.

→ Replace the low-value contents in real time.

$$U_b = \frac{\sum_{i=1}^{N_b} (PSNR_{b,i} - PSNR'_{b,i})}{T_{now} - t_b}$$

<Block b's utility value in unit time>

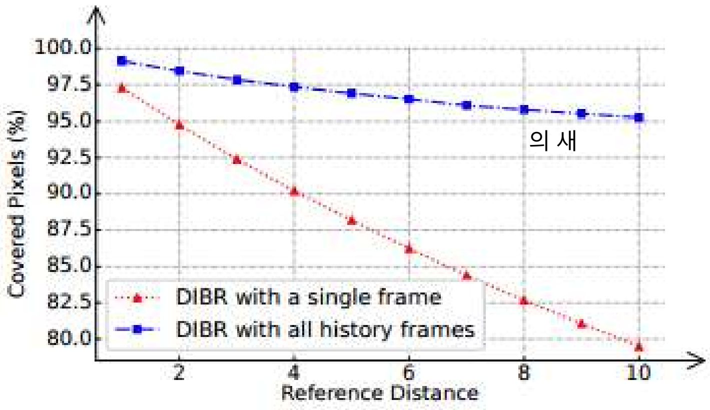
- history frames are split into 64x64 'blocks'
- Ub : utility function for each block b (= block's value)
- Nb : the times of being used since the block b entered the cache
- $PSNRb,i$: the PSNR of b's covered area in ith use
- $PSNR'b,i$: the PSNR of the repaired area by inpainting algorithm(prediction X)
- $PSNRb,i - PSNR'b,i$: the prediction performance gain in this use of b
- $Tnow - tb$: the span of time b is stored in the cache

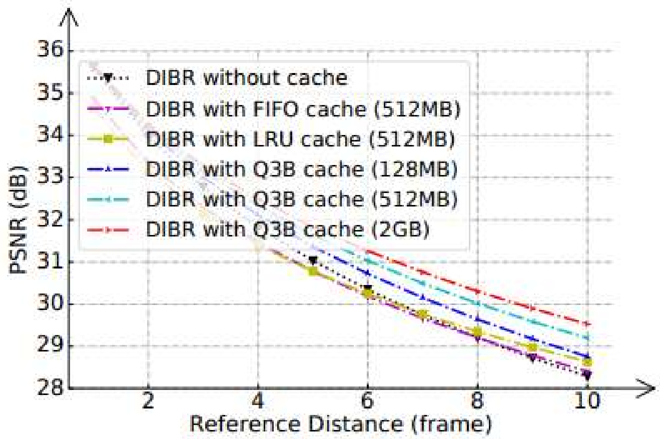
✓ When end device storage is full, Q3B caching strategy evicts 30% blocks

✓ If Ub<0, evict block b even though the storage is not full. → "Inpainting is better!"

(1) Q3B Cache

Performance





✓ History frames provide more 3D information
→ Better prediction performance of DIBR

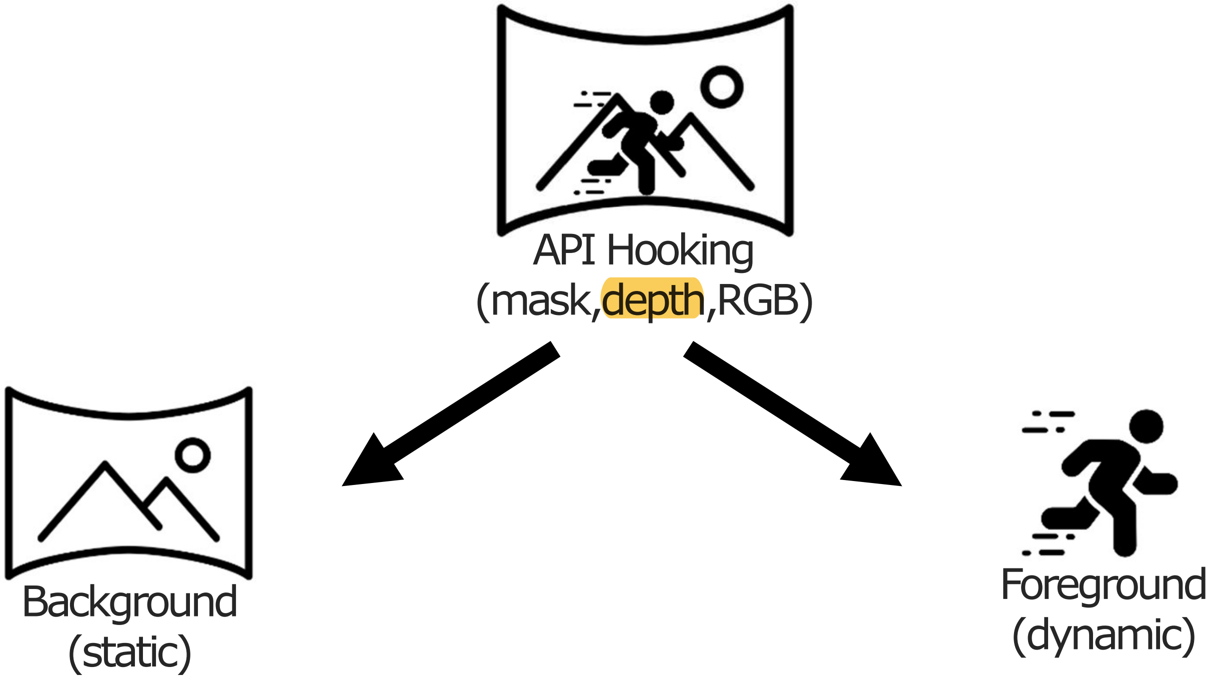
✓ Performance gains from Q3B cache surpass LRU, FIFO

(2) Server-assisted LSTM prediction

Insight 2 :

Prediction performance for foreground objects can be improved by LSTM

"To predict the foreground using LSTM, need to separate foreground from background"



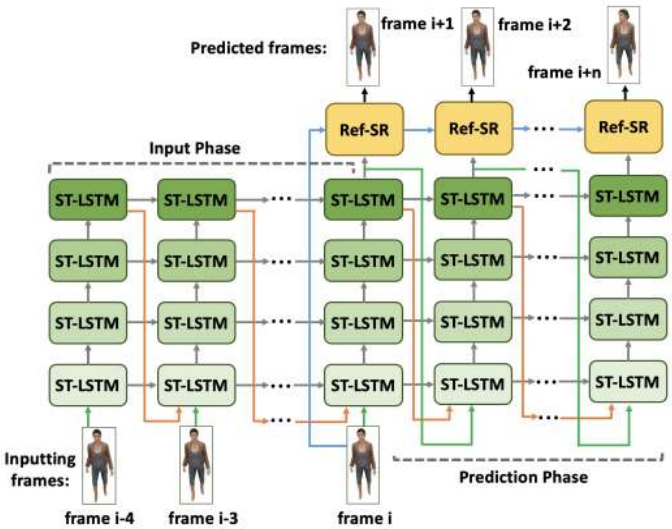
(2) Server-assisted LSTM prediction

LSTM-based foreground prediction

< 2 Factors > → <Method>

1) Motion → **ST-LSTM**

2) Blur → **+ RefSR**

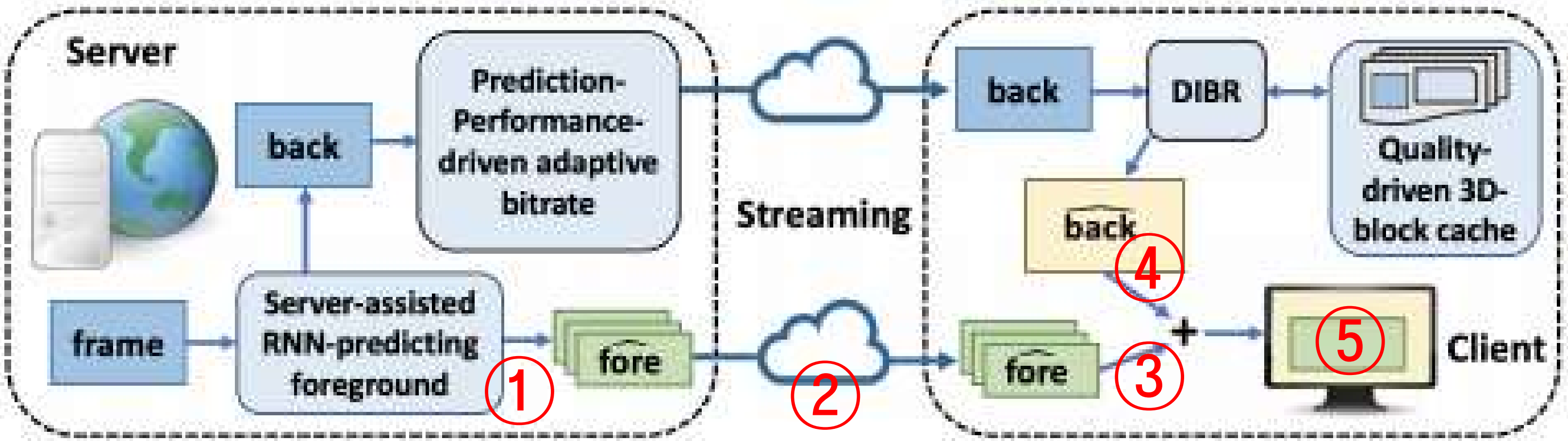


Architecture of LSTM model

1. Input : Load 5 frames into ST-LSTM, processing one by one.
2. Prediction : 4th ST-LSTM unit predicts and outputs the image.
3. Restoration & Output: RefSR refines blurred areas, finalizing the prediction.

(2) Server-assisted LSTM prediction

Server-assisted(Offload)



LSTM model require heavy computation → **Offload to server** (data center)

① Server separates each foreground object from image

② Predicted foreground image from server-LSTM model stream to client

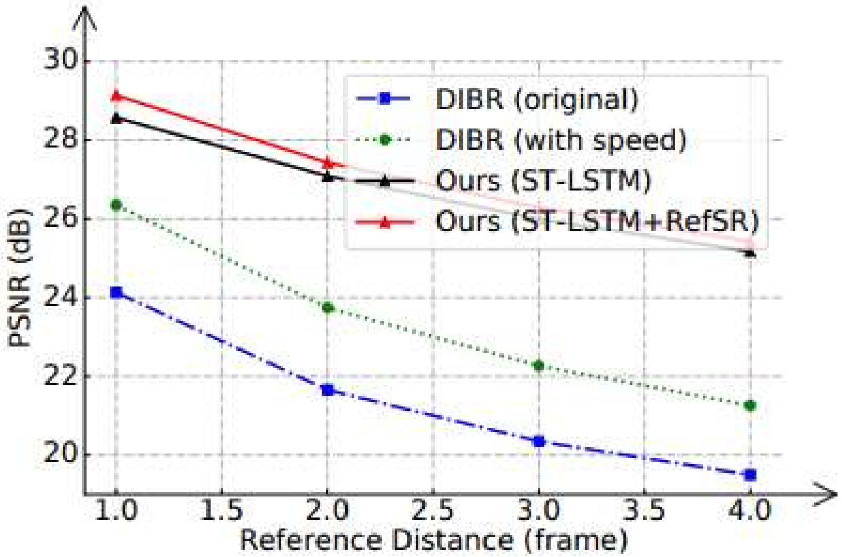
③ Foreground part transmitted as a separate stream to client with high priority

④ Background part from DIBR transmitted to client

⑤ Foreground(server-assisted LSTM) and background(DIBR) complete image

(2) Server-assisted LSTM prediction

Performance



Foreground prediction

- ✓ DIBR vs ST-LSTM vs ST-LSTM + RefSR
- ✓ ST-LSTM : Motion of foreground objects
- ✓ RefSR : Refine blurry result

(3) Prediction-performance-driven adaptive bitrate

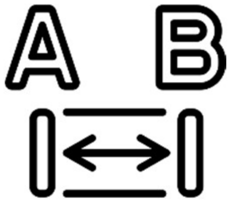
Insight 3 :

Prediction performance for DIBR can be optimized by an adaptive streaming strategy

<**3 Factors** that influence 'Prediction Performance'>

Video content

Reference frame quality

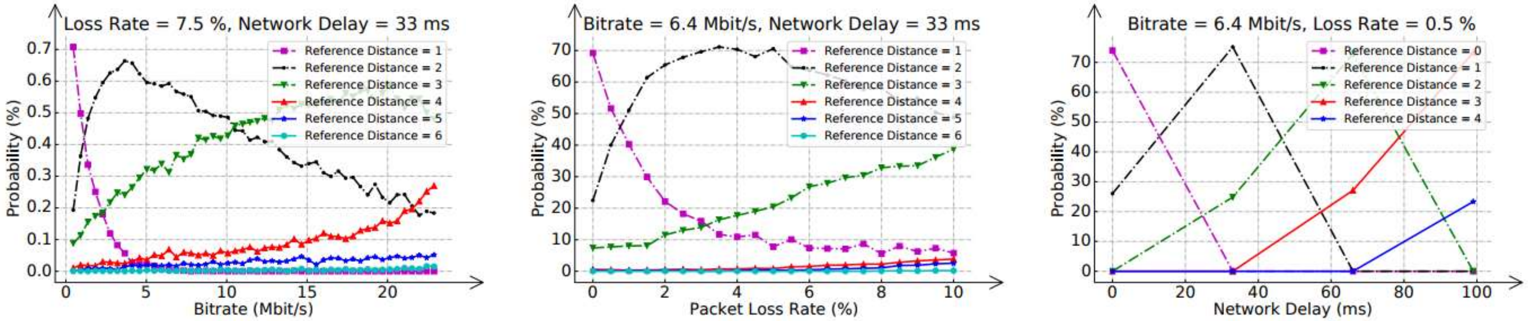


Reference distance
(important)

(3) Prediction-performance-driven adaptive bitrate

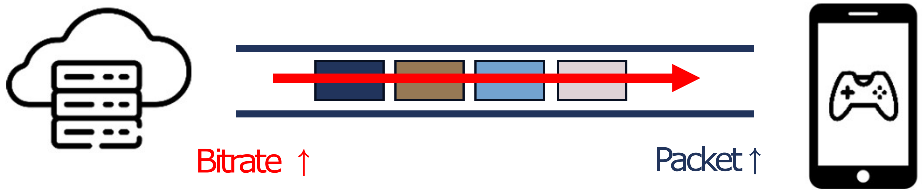
Modeling

KEY : Modeling the '**reference distance**'



(a) Reference distance v.s. bitrate (b) Reference distance v.s. loss rate (c) Reference distance v.s. delay

∴ Reference distance ∝ bitrate, loss rate, delay



(3) Prediction-performance-driven adaptive bitrate

Modeling

$$E(q, B, d, l, i) = \sum_{r \in \{0,\ldots,R\}} P(r, q, B, d, l) * Q(r, q, i)$$

Expected prediction quality under video rate q

- r : reference distance
- q : video quality (= bitrate)
- B : network bandwidth
- d : network delay
- l : loss rate
- i : frame id
- P = Probability distribution of reference distance
- Q = Prediction performance

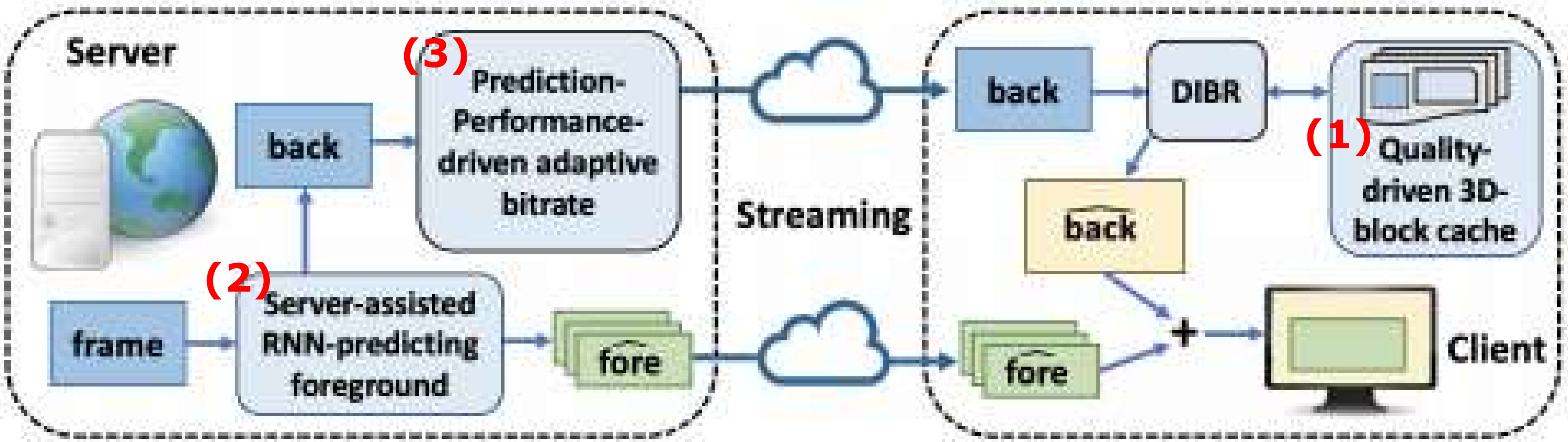
$$q_{chosen} = \arg\max_{q} E(q, B, d, l, i)$$

Server choose next frame's best bitrate ▶ Server adjust bitrate in real time to optimize prediction performance

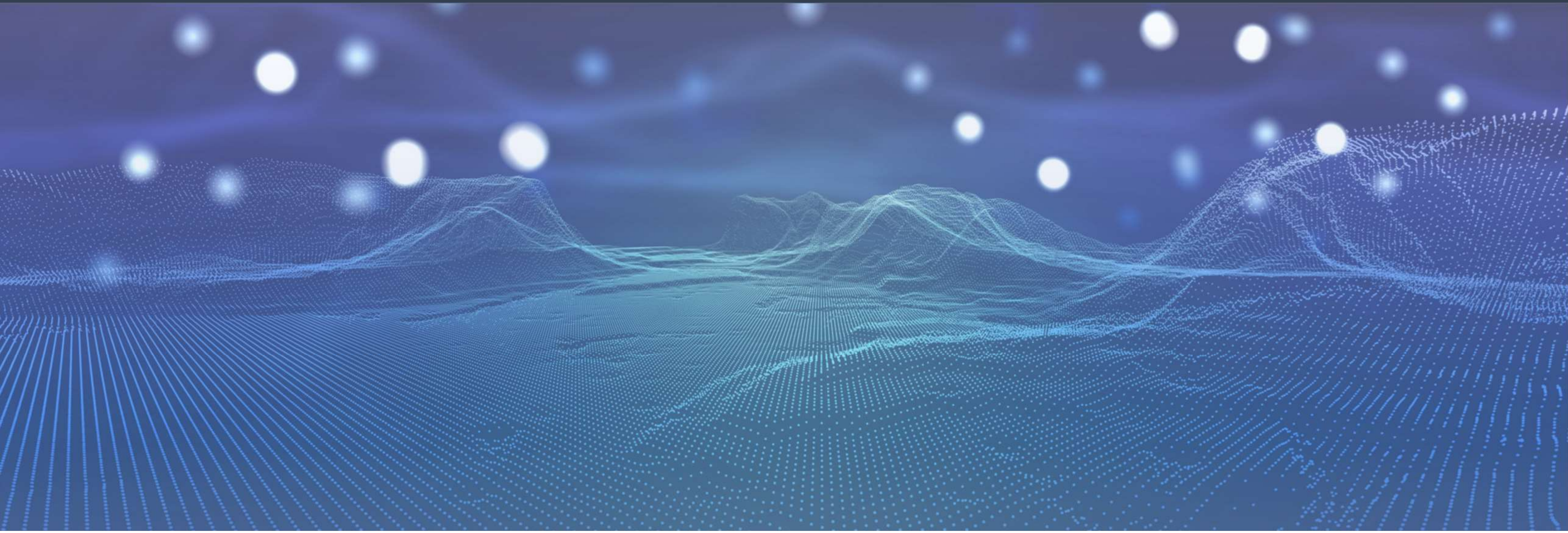
Workflow

"Now we can understand ZGaming's workflow"

1) Q3B Cache
2) Server-assisted LSTM prediction
3) Prediction-performance-driven adaptive bitrate

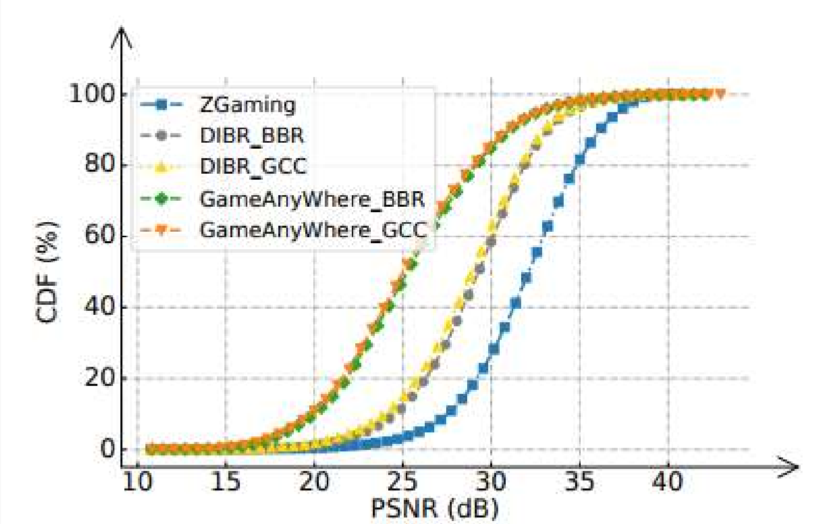


4. EVALUATION



Evaluation

Video Quality



✓ Image similarity(PSNR) between its latest received frame and desired frame

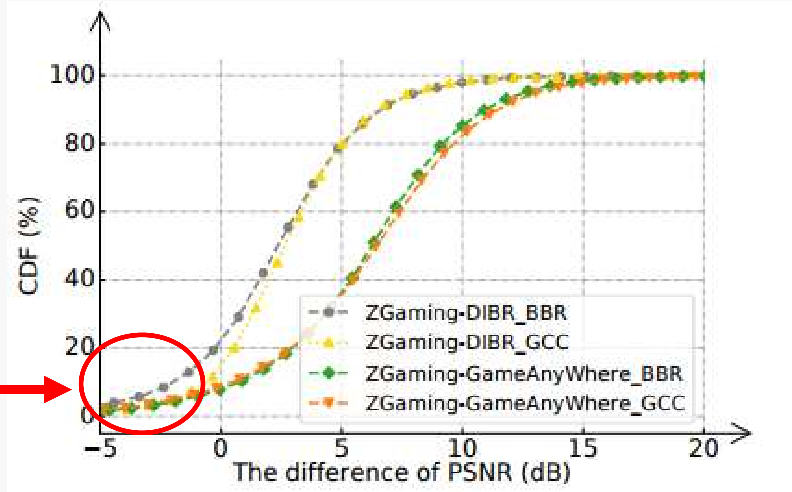
✓ Video Quality : ZGaming > DIBR > GameAnyWhere

※ BBR, GCC : Congestion control algorithms

Evaluation

Robustness

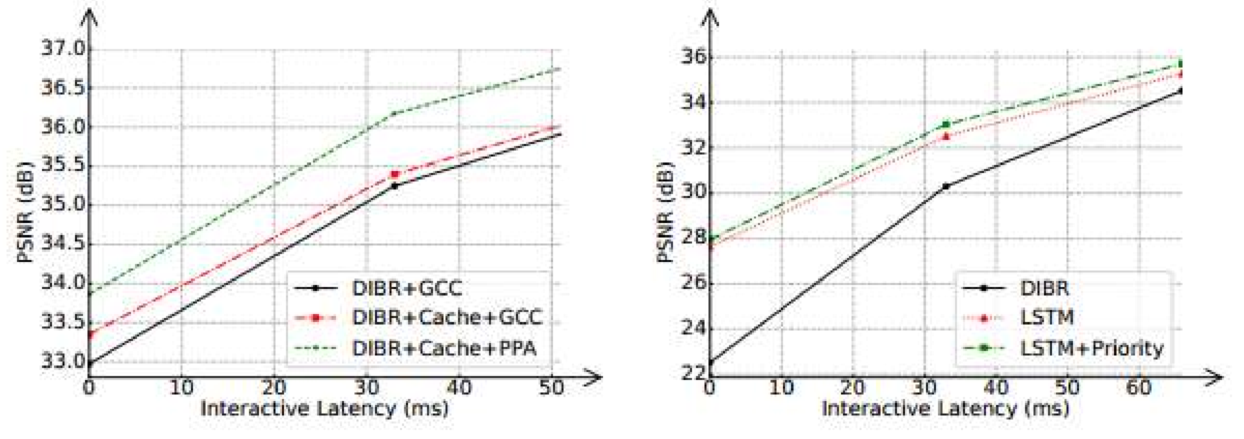
Very low



- ✓ Even if prediction errors occur, the degradation of quality is very limited.
- ✓ Zgaming is robust to prediction errors.

Evaluation

Foreground & Background



(a) background (b) foreground

"Performance gets better with each ZGaming strategies applied!"

THANK YOU