

KEYFRAME INSERTION for Fast Channel Switching & Packet-Loss Repair in Low-Delay Live Streaming

Scan QR to watch video explanation



The GOP size trade-off

Solution: Combine very long + very short GOP

Requirements

Very long GOP

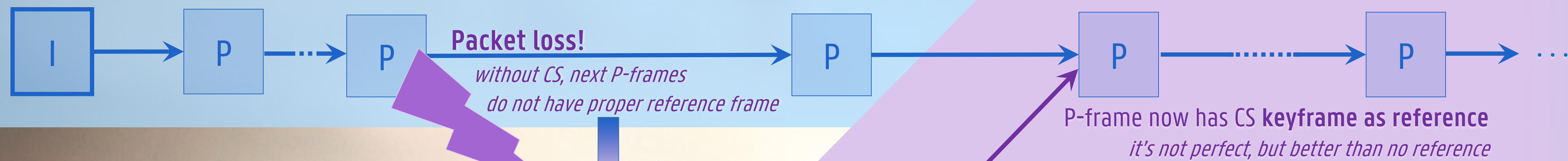
- ✓ efficient compression
- ✗ for staying on one channel
- ✗ only for steady connections

Very short GOP

- ✗ bad compression
- ✓ for fast channel switching
- ✓ for fast packet-loss repair

Normal Stream (NS) with very long GOP

All users receive very compression-efficient video stream by default



Steady-state users are **not** impacted by keyframe insertion of another user

Companion Stream (CS) with only keyframes

Sent only to users when they need it



Channel switch!
without CS, need to wait a long time for keyframe in NS

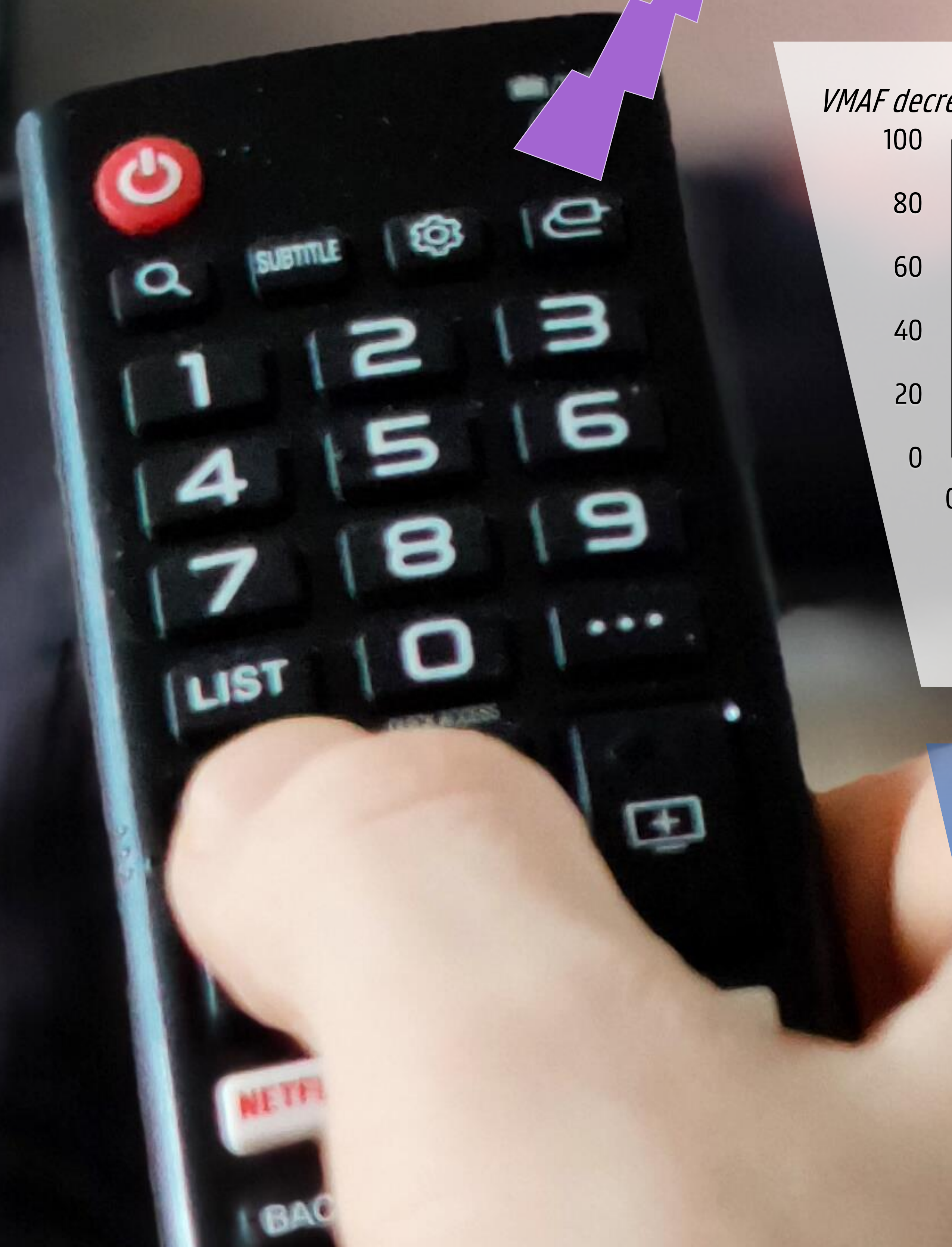
Quick fix:
keyframe inserted
only for this user!

Next CS keyframes not transmitted until needed again

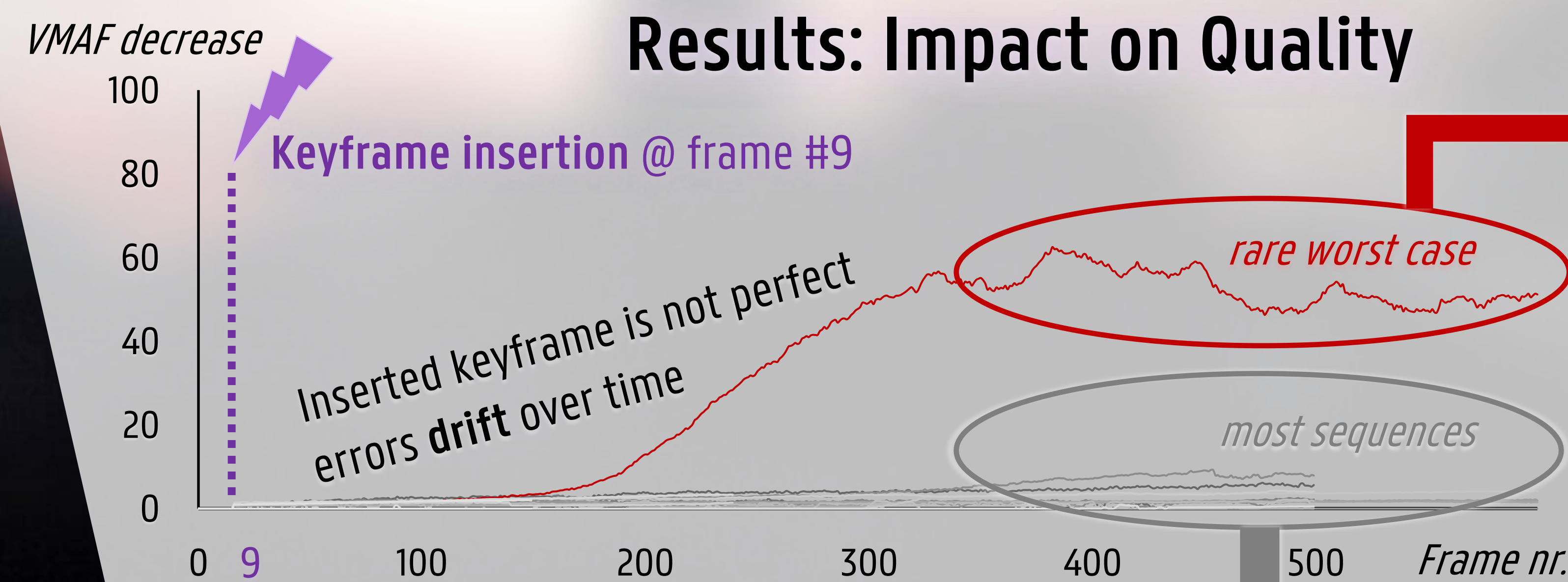
For H.264/AVC, H.265/HEVC & H.266/VVC

- ✓ Disable TMVP
- ✓ Repeat Packet Sets
i.e., VPS, SPS, PPS & APS
- ✓ POC Keyframe = #0
in practice, relaxed at decoder only in H.264/AVC & H.265/HEVC IDR

WHY NOT BOTH?



Results: Impact on Quality

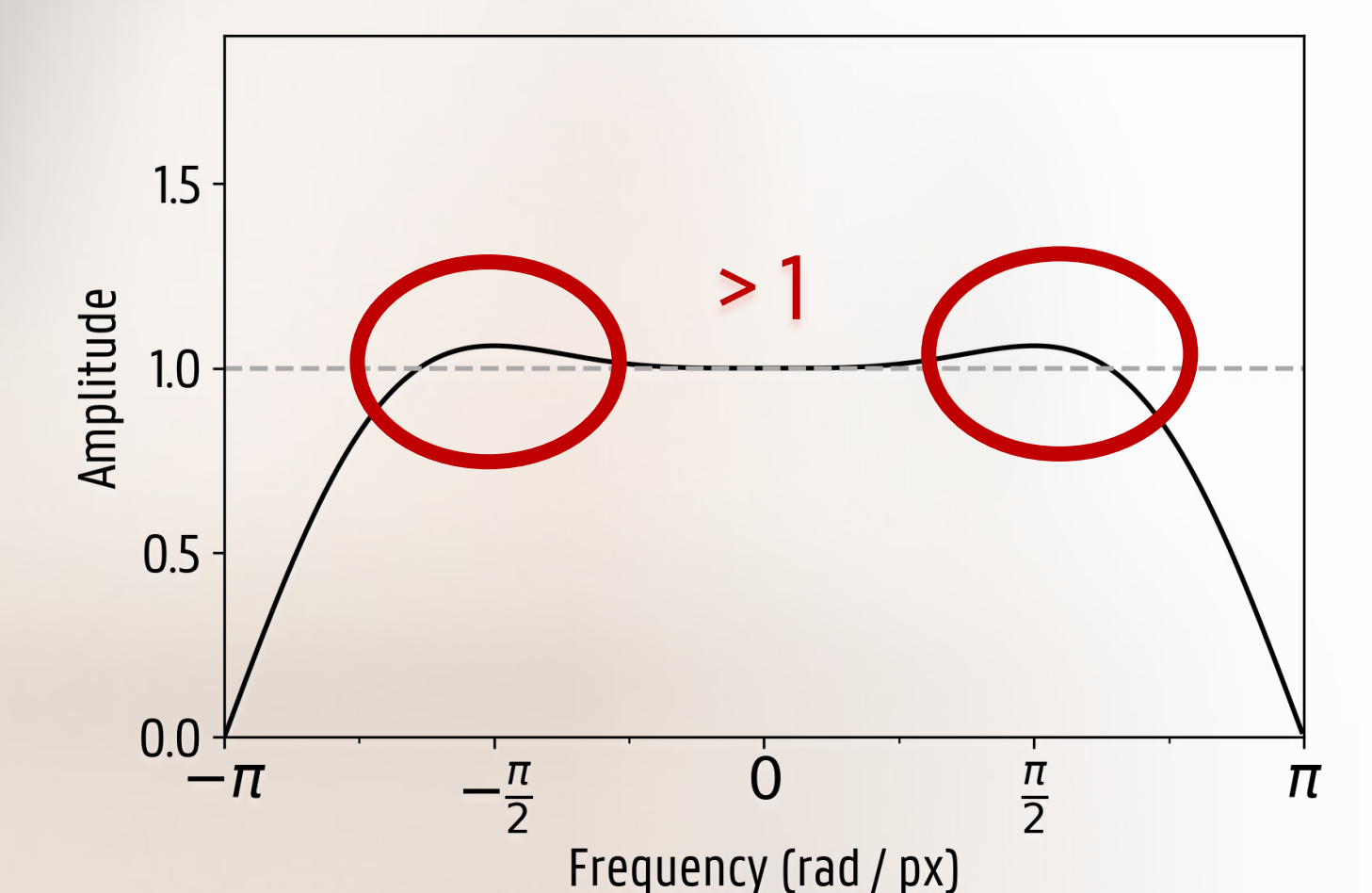


Imperceptible drift in most sequences
 $0.14 \leq \text{median VMAF decrease} \leq 1.69$ (for equal QP in NS & CS)

only when no new keyframe in NS for multiple seconds!
Perceptible drift in rare worst case



Perceptible due to **overshoot** of frequency components in H.264/AVC **halfpel interpolation filter**



Fixed by modifying interpolation coefficients or disabling subpel motion estimation

Conclusion

TL;DR:

insert keyframe in compression-efficient stream to enable fast channel switching & packet-loss repair without impacting the performance of other users

- ✓ Generally imperceptible
- ✗ Sometimes **perceptible** artifacts in worst case
- ✓ Fixed by modifying or disabling subpel interpolation

Future work: fix differently