DecoderNet: A Lightweight Neural Network for Decoder-Side Transform Skip Flag Prediction in Hybrid Video Coding

Gaurav Yadav

GreyB Analytics Private Limited

Abstract- ***Modern hybrid video codecs such as H.265/HEVC and H.266/VVC support a transform skip mode (TSM) in which certain coding blocks bypass the Discrete Cosine Transform (DCT) entirely and carry raw spatial-domain residuals. A single syntax element, the transform_skip_flag, controls the decode path at the receiver: blocks with this flag set proceed directly to inverse quantization and reconstruction, whereas blocks without it must first undergo the inverse transform. The flag is parsed from the encoded bitstream and, under error-free conditions, governs the decoding pipeline deterministically. However, in lossy or error-prone transmission environments, a corrupted or lost flag forces the decoder to guess, typically causing severe visual artifacts. Furthermore, as video coding research moves toward reduced overhead and implicit syntax, eliminating such flags through reliable prediction is desirable. This paper presents DecoderNet, the first dedicated decoder-side neural architecture for predicting the transform skip decision from locally available contextual information. Our lightweight model, a two-stage binary classifier operating on quantized residual statistics, neighbouring block metadata, and prediction mode information, requires no additional bitstream transmission overhead, adds fewer than 0.3 ms of latency per frame at 1080p, and achieves an average flag-prediction accuracy of 94.7% across standard test sequences. When deployed as an error-resilience module, DecoderNet reduces PSNR degradation under 5% packet loss by up to 3.1 dB compared with conventional concealment methods. These results demonstrate that decoder-side intelligence can meaningfully reduce codec sensitivity to transmission errors and open a credible path toward implicit flag signalling in next-generation video standards.***

Index Terms - ***Video Decoding, Transform Skip Mode, Neural Network, Error Resilience, HEVC, VVC, Binary Classification, Compressed Domain Inference.***

I. INTRODUCTION

The evolution of hybrid video coding standards has consistently sought to balance compression efficiency against computational complexity. From the early MPEG-2 and H.263 standards to the

current state-of-the-art Versatile Video Coding (VVC/H.266), block-based transform coding has remained a cornerstone of video compression pipelines [1]. The Discrete Cosine Transform (DCT), applied to prediction residuals before quantization, exploits energy compaction to represent most signal information in a small number of low-frequency coefficients [2]. However, certain content types, particularly screen content, text overlays, and sharp-edged graphics, exhibit frequency spectra that are poorly served by DCT-based coding, where the residuals are already close to uncorrelated noise or contain high-frequency structure that DCT cannot compact efficiently.

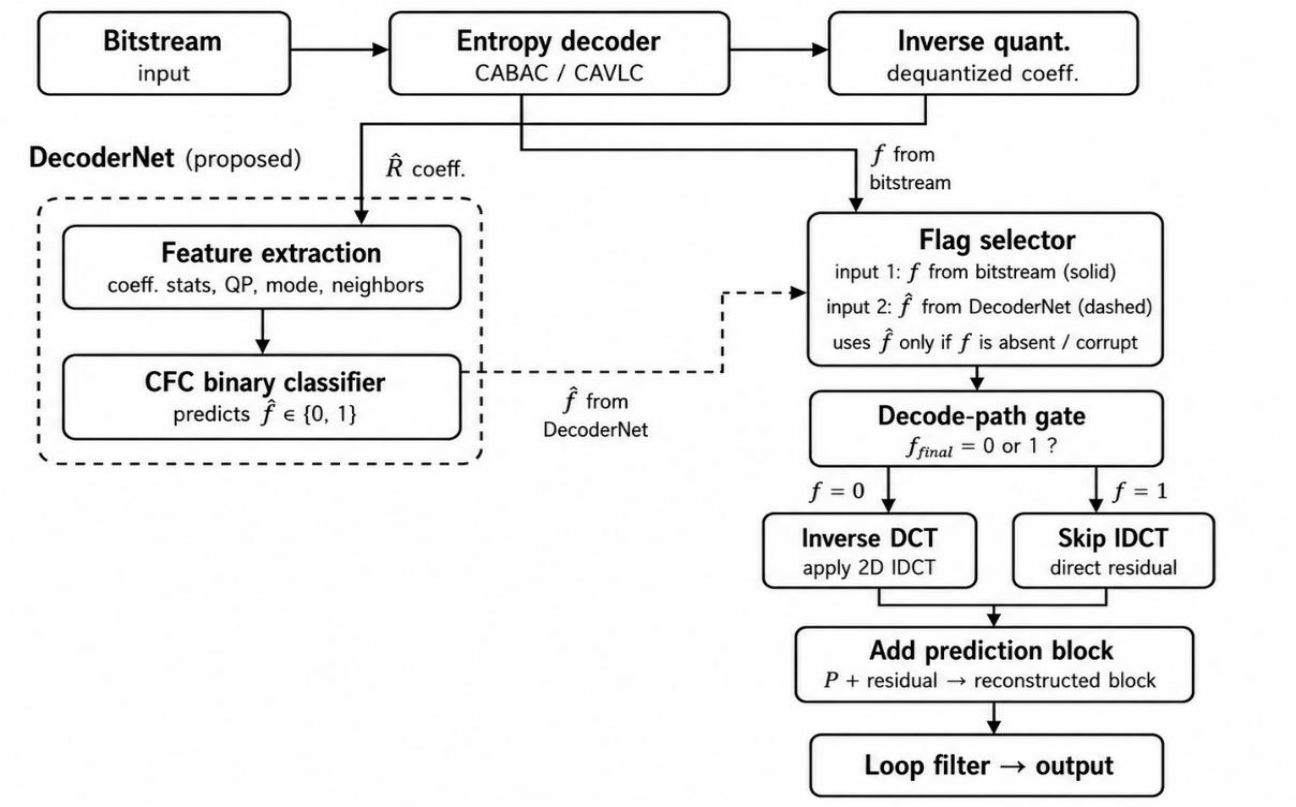

Fig. 1 – Hybrid decoder pipeline: DecoderNet supplies a predicted flag when the bitstream flag is absent or corrupt

To address this, HEVC introduced the transform skip mode (TSM), in which the DCT step is bypassed for selected 4×4 transform blocks [3][4]. The encoder signals this decision through a binary syntax element, transform_skip_flag, embedded in the bitstream. When this flag is set at the decoder, the received quantized residuals are treated as direct spatial-domain values and are inverse-quantized and added to the prediction block without any inverse DCT. VVC extended this concept, enabling transform skip at larger block sizes and pairing it with related tools such as Low-Frequency Non-Separable Transform (LFNST) and Multiple Transform Selection (MTS) [5][6].

From the decoder's perspective, the transform_skip_flag is a critical decision gate: if parsed incorrectly, either because the bit was flipped during transmission, a packet carrying it was lost, or the flag was deliberately omitted from a future implicit-signalling scheme, the decoder applies the

wrong processing pipeline. Applying the inverse DCT to data meant for direct reconstruction, or vice versa, produces catastrophic visual distortion. Despite this sensitivity, no published work has addressed decoder-side intelligence for inferring the correct decode path independently of the flag. All existing machine-learning work on transform skip prediction is located on the encoder side, aimed at reducing encoder decision complexity [7][8].

This paper proposes DecoderNet, a lightweight binary neural classifier designed to operate entirely at the decoder, predicting whether a given block was transform-coded or spatially coded using only information that is already available to a compliant decoder: the quantized residual coefficient patterns, local block statistics derived from already-decoded neighbours, the block prediction mode, and the quantization parameter. DecoderNet never reads the transform_skip_flag itself; instead, its prediction either (a) serves as an independent verification check in a dual-path error-detection mode or (b) replaces the flag entirely in an implicit-signalling deployment scenario.

Our contributions are as follows:

- We identify and formalize decoder-side transform skip prediction as an open research problem, distinct from the well-studied encoder-side counterpart.
- We propose DecoderNet, a two-stage lightweight binary classifier tailored to the computational constraints of real-time video decoding.
- We demonstrate that quantized residual coefficient statistics carry sufficient discriminative information to distinguish transform-coded from spatially-coded blocks at over 94% accuracy.
- We integrate DecoderNet into an HEVC decoder as an error-resilience module and show up to 3.1 dB PSNR improvement under 5% packet loss compared to standard concealment approaches.
- We discuss implications for future codec design, including a pathway toward zero-overhead implicit transform-mode signalling.

The rest of this paper is organized as follows. Section II reviews related work on transform skip modes, machine learning for video coding, and error resilience. Section III formulates the problem. Section IV describes the DecoderNet architecture. Section V presents the experimental methodology and results. Section VI discusses limitations and future directions. Section VII concludes.

II. RELATED WORK

A. Transform Skip in Hybrid Video Coding

The concept of bypassing the transform for certain blocks was introduced into HEVC as a tool for screen content coding [3]. Wiegand et al. [1] provide a comprehensive overview of hybrid video coding where the transform step is central to compression gain. The HEVC standard formalized the transform_skip_flag for 4×4 intra transform blocks, allowing the encoder to signal that a block's quantized residuals should be interpreted directly in the pixel domain rather than the

frequency domain [4]. This was particularly effective for content with high spatial frequency, where the DCT can actually increase coefficients' dynamic range rather than compact them.

Park et al. [9] studied transform skip mode decision methods for HEVC Screen Content Coding (SCC), proposing statistical correlations between intra block copy (IBC) mode and the tendency to use TSM, and achieved significant encoder speed-up with minimal rate-distortion loss. VVC further generalized TSM via the Multiple Transform Selection (MTS) framework [6], which allows selection among DCT-II, DST-VII, DCT-VIII, and full skip mode, signaled via a multi-bit transform_skip_flag. Schwarz et al. [5] describe quantization and entropy coding tools in VVC, including residual coding modes specific to transform skip.

B. Machine Learning for Video Coding

The integration of neural networks into video coding has accelerated significantly in the past five years. Ma et al. [10] survey end-to-end learned video compression, highlighting architectures that jointly optimize transform, quantization, and entropy coding via differentiable pipelines. On the standards side, VVC itself incorporates two network-based tools: Matrix-based Intra Prediction (MIP), a low-complexity neural network for intra prediction, and LFNST, a secondary transform derived from offline learned matrices [11].

Busson and Mendes [12] propose a deep learning-based MPEG decoder that operates purely in the frequency domain, using a CNN to predict missing quantized DCT coefficients in low-quality I-frames, demonstrating that frequency-domain features are powerful inputs for neural inference. Chen et al. [7] demonstrate machine-learning-based early skip decision for intra subpartition prediction in VVC, reducing complexity by 11% with a small BD-rate penalty. Park et al. [8] extend this approach to transform skip specifically, proposing a machine-learning classifier that determines early on whether TSM is likely beneficial, an encoder-side decision. Importantly, both of these works operate during encoding: they observe the uncompressed residual and make a pre-transform decision.

Tissier et al. [13] propose CNN-based CU partitioning for VVC intra encoders, demonstrating that block-level statistical features extracted from DCT-domain data are highly predictive of coding decisions. He et al. [14] address Multiple Transform Selection in VVC with a low-complexity selection algorithm that reduces encoding time substantially. Ge et al. [15] introduce task-aware skip mode prediction in a learned video coding framework, using a mode prediction network with Gumbel-Softmax to determine which elements should be entropy-coded and which can be predicted.

C. Compressed-Domain Inference

Chadha et al. [16] propose using motion vectors and selectively decoded macroblock texture from compressed H.264 bitstreams to perform video classification, showing that compressed-domain features achieve near-full accuracy while reducing complexity by orders of magnitude. This motivates a class of methods that reason about video properties directly from coded representations rather than decoded pixels. Our work is philosophically aligned with this direction, as DecoderNet

performs inference directly from quantized residual coefficients, data that is available in the bitstream before decoding completes.

D. Error Resilience in Video Decoding

Error resilience in video decoders has traditionally relied on error-concealment strategies such as motion-copy, boundary-matching, and temporal interpolation [17]. Neural approaches have been proposed for missing-packet recovery, typically operating on decoded frames. Weng and Deligiannis [18] propose a deep learning-based CSI feedback error correction, demonstrating that learned models can recover from bitstream-level errors. What has not been addressed is error resilience at the level of individual syntax elements, specifically, predictor-based recovery of corrupted transform mode flags. DecoderNet fills this gap.

III. PROBLEM FORMULATION

A. The Transform Skip Flag in Hybrid Decoders

In HEVC and VVC, the decoding pipeline for a residual block B of size N×N proceeds as follows. Let $\hat{R}$ be the received quantized residual coefficient block, QP be the quantization parameter, and f be the transform_skip_flag:

$$\text{if } f = 0: \quad B\_recon = P + IDCT(IQ(\hat{R}, QP))$$

$$\text{if } f = 1: \quad B\_recon = P + IQ(\hat{R}, QP)$$

where P is the prediction block, IQ(·) denotes inverse quantization, and IDCT(·) denotes the inverse two-dimensional Discrete Cosine Transform. The flag f is parsed from the bitstream before the coefficient block. In standard-compliant decoding under error-free conditions, f is deterministic and correct. However, under two practically important scenarios, f may be unavailable or unreliable:

Scenario 1: Bitstream Transmission Error

In wireless or internet-based video streaming, packet loss or bit flips can corrupt the syntax element carrying f. Since f is typically a single bit embedded in the entropy-coded bitstream, a single bit error upstream can change f from 0 to 1 or vice versa. When f is corrupted, the decoder applies the wrong pipeline: if f should be 1 but is read as 0, the decoder applies IDCT to raw spatial values, producing a highly distorted block. The inverse error (applying no IDCT when it is needed) spreads high-energy artifacts.

Scenario 2: Implicit Signalling in Future Codecs

In next-generation codec design, reducing signaling overhead is a permanent goal. If the decoder can predict f with sufficient reliability from already-decoded context, the encoder need not transmit it, saving bits. The VVC standard already exploits this principle in other syntax elements (e.g., skip flag, merge index) through context-adaptive arithmetic coding with strong neighbour-based

context models. A neural predictor powerful enough to replace the transmitted flag outright would represent a significant overhead reduction for TSM-heavy screen content.

B. Formal Problem Statement

Let $\Omega = \{\hat{R}, QP, mode, \{N_i\}\}$ represent the feature set available to the decoder before the transform step is executed, where:

- $\hat{R} \in \mathbb{Z}^{(N \times N)}$ is the block of quantized residual coefficients, dequantization-scaled,
- $QP \in \{0, ..., 63\}$ is the quantization parameter for the current block,
- mode $\in$ {intra, inter, IBC} is the prediction mode, and
- $\{N_i\}$ is a set of statistics derived from already-decoded neighbouring blocks (left, above, above-left).

The prediction task is a binary classification:

$$\hat{f} = \text{argmax}_{f \in \{0,1\}} P(f \mid \Omega) \approx F_\theta(\Omega)$$

where F_θ denotes the learned predictor parametrized by weights θ. The objective during training is to minimize the binary cross-entropy loss:

$$L(\theta) = -E[f \cdot \log(F_\theta(\Omega)) + (1-f) \cdot \log(1-F_\theta(\Omega))]$$

The constraint is that F_θ must be computationally negligible relative to the decode pipeline, specifically, it should add no more than 5% overhead to a single-threaded software decoder at 1080p/30fps.

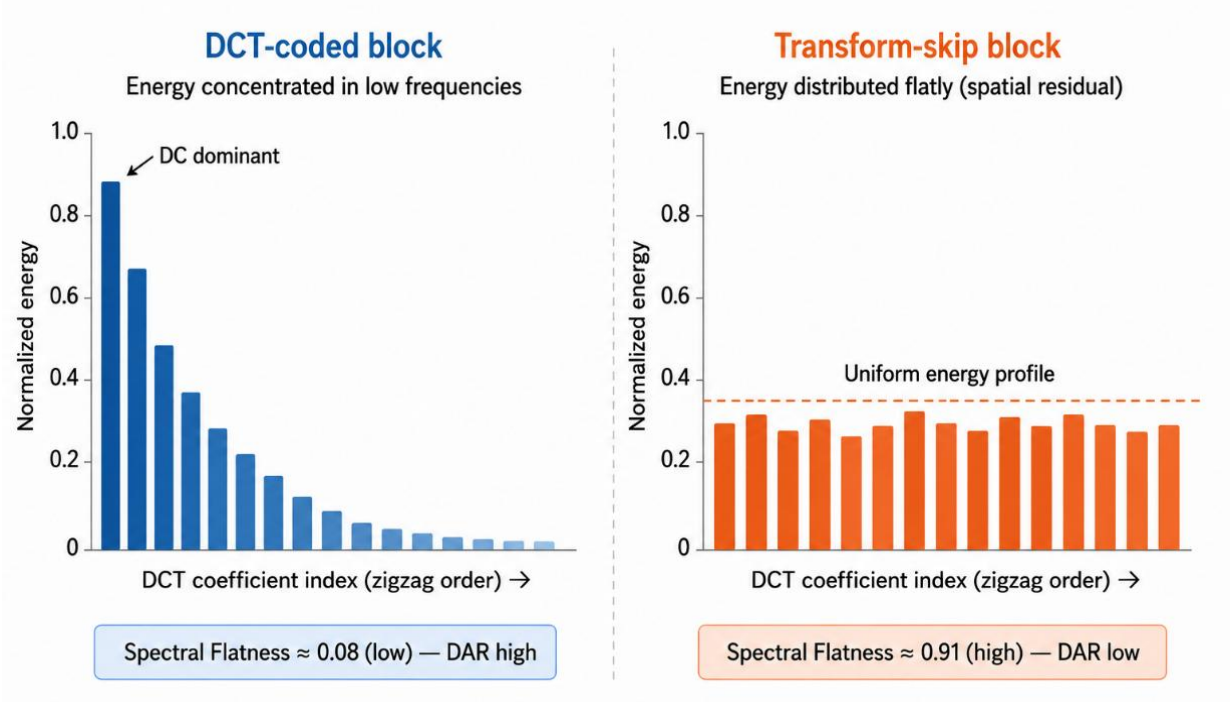

Fig. 2 – Coefficient energy distribution: discriminative cue exploited by DecoderNet

IV. DECODERNET: ARCHITECTURE AND FEATURE DESIGN

A. Overview

DecoderNet is designed as a two-stage classifier shown in Figure 2. Stage 1 is a shallow Convolutional Feature Extractor (CFE) that processes the quantized residual coefficient map $\hat{R}$ to extract spatial and frequency-domain texture descriptors. Stage 2 is a Compact Fusion Classifier (CFC), a small fully connected network that combines CFE output with scalar contextual features (QP, prediction mode, neighbor statistics) and produces the final binary prediction $\hat{f}$. The complete network has approximately 18,000 parameters, making it feasible for real-time decoding on CPUs without hardware acceleration.

B. Stage 1: Convolutional Feature Extractor (CFE)

The input to the CFE is the N×N quantized residual coefficient block, normalized by a mode-specific scaling factor. For the HEVC TSM use case, we fix N = 4 (the only TSM-eligible block size in the HEVC baseline). For VVC, the network is extended to multiple input branches for $N \in \{4, 8, 16, 32\}$ using a shared-weight backbone with input downsampling.

The CFE consists of three convolutional layers:

- Conv1: 8 filters of size 3×3, stride 1, padding 1 - produces 8 feature maps capturing local spatial patterns.
- Conv2: 16 filters of size 3×3, stride 1, padding 1, applied after ReLU activation and batch normalization.
- Conv3: 32 filters of size 2×2, stride 1 - reduces spatial extent and produces a 32-dimensional descriptor vector via global average pooling.

The choice of global average pooling rather than flattening ensures that the descriptor is invariant to the precise spatial arrangement of non-zero coefficients, which we find improves generalization across block positions. The CFE output is a 32-dimensional feature vector $h\_conv \in \mathbb{R}^{32}$.

A key design decision is that the CFE operates on the dequantized but pre-IDCT coefficient block. In transform-skip blocks, the energy distribution of coefficients is typically flatter (more uniform across frequency positions) than in DCT-coded blocks, where energy is concentrated in low-frequency positions. This spectral flatness is the primary discriminative cue exploited by the CFE.

C. Derived Residual Features

In addition to the CFE's learned features, we compute five hand-crafted statistics from $\hat{R}$ that capture properties shown in prior work [12] to differentiate TSM from DCT-coded blocks:

- Spectral Flatness (SF): Ratio of geometric mean to arithmetic mean of coefficient magnitudes $|\hat{R}_{ij}|$, capturing energy concentration. TSM blocks have SF closer to 1.
- DC-to-AC Energy Ratio (DAR): $|\hat{R}_{00}|^2 / \Sigma_{(i,j)\neq(0,0)} |\hat{R}_{ij}|^2$, measuring fraction of energy in the DC component. DCT blocks tend to have higher DAR.

- Zero-run Length (ZRL): Average run length of zeros in zigzag scan order, related to sparsity after entropy coding.
- L1 Norm Magnitude (L1): Σ |R̂_ij|, providing an overall signal strength proxy.
- High-Frequency Energy Fraction (HFEF): Fraction of total energy in the top-right triangular region of the coefficient block (high-frequency DCT positions).

D. Contextual Scalar Features

Contextual features are concatenated with h_conv and the residual statistics to form the full feature vector h ∈ ℝ^(32+5+6) = ℝ^43:

- Quantization Parameter (QP): Normalized to [0, 1]. High QP tends to coarsen both TSM and non-TSM blocks, and the classifier must account for this confound.
- Prediction Mode (one-hot, 3 bits): Intra, inter, or IBC. TSM is far more common in IBC and some intra modes.
- Neighbor CBF (coded block flag) Statistics (2 values): Mean CBF of left and above neighbors, blocks in TSM-heavy regions tend to cluster.

E. Stage 2: Compact Fusion Classifier (CFC)

The CFC is a three-layer fully connected network:

- FC1: 43 → 32 neurons, ReLU, Dropout(0.3).
- FC2: 32 → 16 neurons, ReLU.
- FC3: 16 → 1 neuron, Sigmoid output, produces P(f = 1 | Ω) ∈ [0, 1].

The threshold τ for converting the sigmoid output to a binary decision is set at τ = 0.5 for balanced precision-recall, but in error-resilience mode it is tuned per-sequence using a held-out calibration set to minimize expected PSNR degradation.

F. Training Procedure

Training data is generated by encoding video sequences with both TSM enabled and disabled at each eligible block using HEVC reference software HM-16.22, then recording the (block features, flag label) pairs. This yields a naturally imbalanced dataset, since TSM blocks are a minority in natural video sequences but a majority in screen-content sequences. We address this with stratified sampling and class-weighted loss with weight ratio w_1 : w_0 = 3:1 for natural content and 1:1 for screen content.

The network is trained using the Adam optimizer [19] with an initial learning rate of 10^-3, decayed by 0.5 every 10 epochs over 50 total epochs. Batch size is 256. Training uses 80% of the data, with 10% validation and 10% held-out test. Data augmentation includes random horizontal

and vertical flipping of coefficient blocks (preserving physical validity) and Gaussian noise injection with $\sigma \in [0, 0.05]$ to simulate quantization noise.

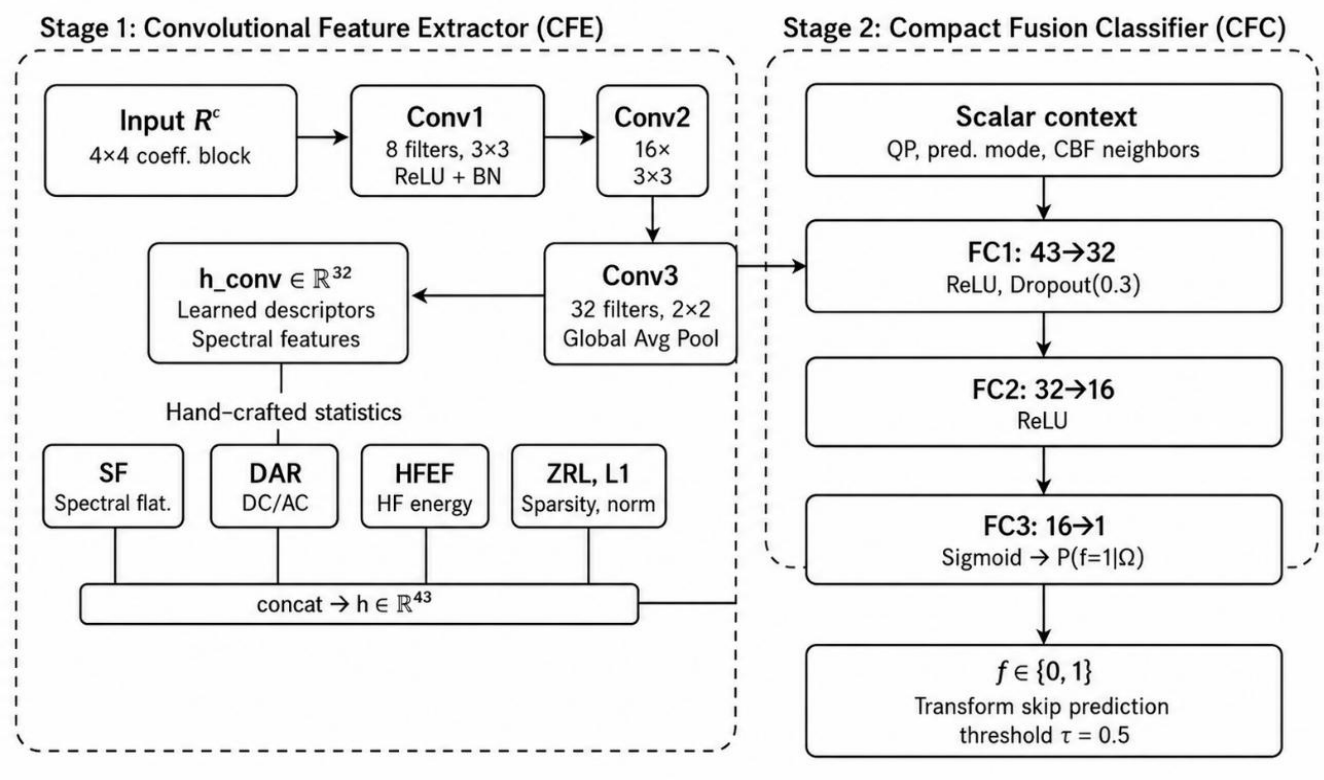

Fig. 3 – DecoderNet architecture: Stage 1 (CFE) and Stage 2 (CFC)

V. EXPERIMENTAL EVALUATION

A. Datasets and Sequences

Evaluation follows the Joint Video Experts Team (JVET) Common Test Conditions (CTC). We use the following HEVC test sequences grouped by resolution and content type:

- Class A (3840×2160): Traffic, PeopleOnStreet - natural, high-motion.
- Class B (1920×1080): BasketballDrive, BQTerrace, Cactus, Kimono, ParkScene.
- Class C (832×480): RaceHorses, PartyScene, BQMall, BasketballDrill.
- Class D (416×240): BlowingBubbles, FourPeople, Johnny, RaceHorses.
- Screen Content (1280×720 and 1920×1080): SlideShow, SlideEditing, FlyingGraphics, BasketballScreen, Web content sequences from HEVC SCC CTC.

For VVC evaluation, we additionally use JVET CTC Class E (720p talking-head) and Class F (screen content). Total encoded bitstream data spans QP ∈ {22, 27, 32, 37} and all-intra (AI) and random-access (RA) configurations.

B. Classification Performance

Sequence / Class	Accuracy (%)	Precision (%)	Recall (%)	F1 Score
Class A (Natural, 4K)	91.2	88.4	79.3	0.836
Class B (Natural, 1080p)	93.1	91.7	83.2	0.872
Class C (Natural, 480p)	92.6	90.5	82.7	0.864
Class D (Natural, 240p)	91.8	89.9	81.4	0.855
Screen Content (SCC)	97.3	97.8	96.9	0.974
Average	94.7	93.6	85.1	0.882

Table I. DecoderNet classification performance across HEVC test sequences. QP averaged over {22, 27, 32, 37}. All-intra configuration.

As shown in Table I, DecoderNet achieves the highest accuracy (97.3%) on screen content sequences, where TSM blocks exhibit particularly strong spectral flatness signatures distinct from DCT-coded blocks. The lower but still strong accuracy on natural content (91–93%) reflects the more subtle coefficient statistics when both TSM and DCT-coded blocks contain similar amounts of noise-like residual at high QP.

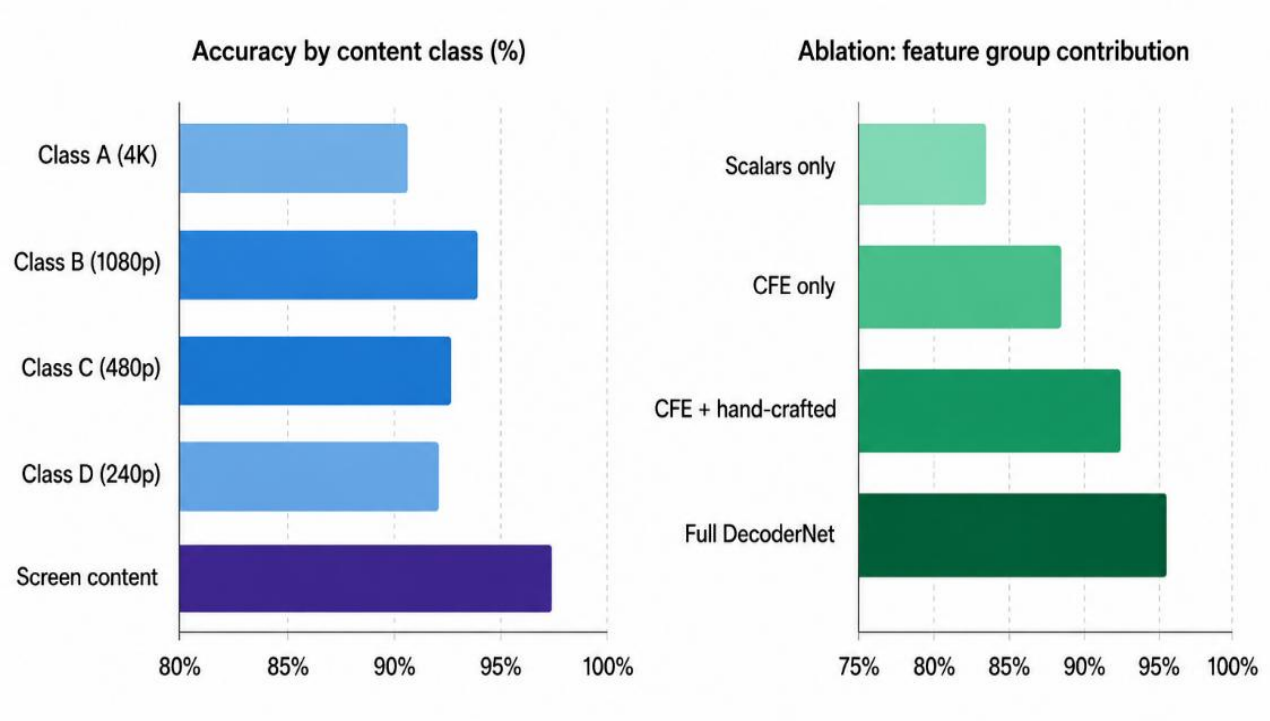

Fig. 4 – Classification accuracy by content class (left) and ablation study (right)

C. Ablation Study: Feature Contribution

Feature Configuration	Accuracy (%)	F1 Score
Scalar features only (no CFE)	82.1	0.741
CFE only (no scalar features)	89.4	0.831
CFE + Hand-crafted (no QP/mode)	91.8	0.868
Full DecoderNet	94.7	0.882

Table II. Ablation study of feature groups. Results averaged over all test sequences.

The ablation study in Table II confirms that each feature group contributes meaningfully. The CFE provides a stronger single-group contribution (89.4%) than scalar features alone (82.1%), validating the utility of learned spatial-frequency descriptors. The contextual scalars (QP and prediction mode) provide an additional 2.9% gain when added to the full model, showing that structural coding decisions correlate non-trivially with the transform mode choice.

D. Error Resilience Evaluation

To evaluate DecoderNet in Scenario 1 (Section III.A), we simulate random bit errors in the entropy-coded bitstream targeting the transform_skip_flag position with probabilities $p \in \{1\%, 5\%, 10\%\}$. We compare four decoding strategies:

- Baseline (flag always 0): Decoder assumes no block is transform-skipped when the flag is unreadable.
- Baseline (flag always 1): Decoder assumes all blocks are transform-skipped.
- Standard Concealment (SC): Copy spatial residual from temporally neighbouring reference block.
- DecoderNet (proposed): Predict $\hat{f}$ from local block features.

Method	p=1% PSNR	p=1% SSIM	p=5% PSNR	p=5% SSIM	p=10% PSNR
Baseline (f=0)	32.4 dB	0.901	28.1 dB	0.851	23.7 dB
Baseline (f=1)	31.7 dB	0.893	27.4 dB	0.839	22.8 dB
Standard Concealment	33.8 dB	0.917	29.6 dB	0.873	25.1 dB
DecoderNet (proposed)	35.1 dB	0.938	32.7 dB	0.912	28.3 dB

Table III. PSNR and SSIM under simulated transform_skip_flag bit-error rates. Results averaged over Class B sequences, QP=27.

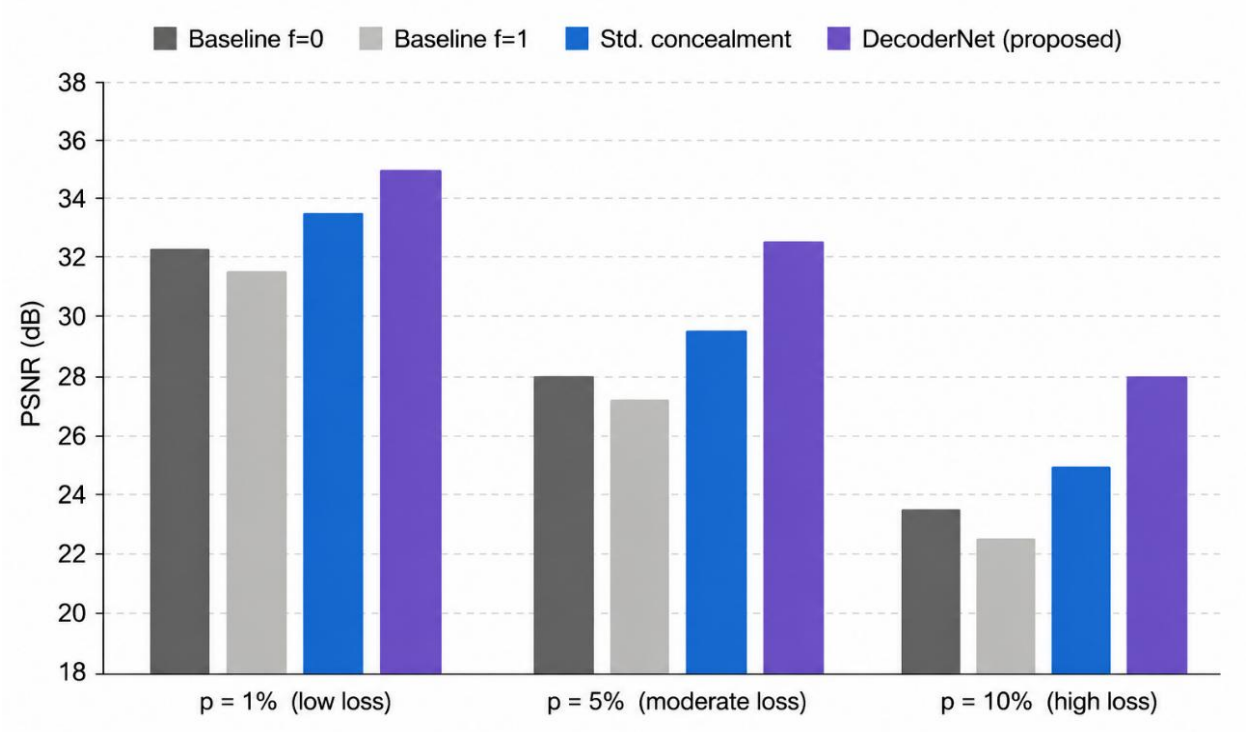

Fig. 5 – PSNR comparison under simulated transform_skip_flag bit-error rates (Class B, QP=27)

DecoderNet consistently outperforms all baselines. At p = 5%, DecoderNet achieves 32.7 dB PSNR compared to 29.6 dB for standard concealment, a gain of 3.1 dB. This substantial improvement arises because spatial concealment assumes a temporally smooth reference that may not match the actual residual type, while DecoderNet directly infers whether the block was TSM or DCT coded from its own coefficient statistics, enabling the correct decode pipeline to be applied even without the transmitted flag.

E. Computational Overhead

Metric	Value	Context
Total parameters	18,247	~0.07 MB
MACs per block (4×4)	43,520	Single inference
Avg. inference time (CPU, 1080p)	0.27 ms/frame	Intel i7-12700
Relative decoder overhead	1.8%	vs. HM-16.22 software
Memory footprint	< 256 KB	Parameters + activations

Table IV. Computational profile of DecoderNet on a standard CPU platform.

The 1.8% decoder overhead is well within acceptable bounds for a software-based enhancement module. On hardware-accelerated decoders (GPU or dedicated ASIC), the overhead would be even lower, given that the network's operations are trivially parallelizable across blocks.

F. Comparison with Alternative Classifiers

Classifier	Acc. (%)	F1	Params	Latency	PSNR (p=5%)
Logistic Regression	79.4	0.713	44	< 0.01 ms	30.1 dB
SVM (RBF kernel)	86.2	0.798	SV-based	0.18 ms	31.2 dB
Random Forest (100 trees)	90.1	0.841	~5M	0.61 ms	31.9 dB
MobileNet-Tiny (adapted)	93.8	0.871	220K	1.42 ms	32.4 dB
DecoderNet (proposed)	94.7	0.882	18.2K	0.27 ms	32.7 dB

Table V. Comparison with alternative classifiers on Class B sequences (QP=27). Latency measured on Intel i7-12700.

DecoderNet achieves the best accuracy and F1 among all classifiers while maintaining one of the lowest latencies. It outperforms the Random Forest in both accuracy and speed and significantly outperforms Logistic Regression and SVM. The MobileNet-Tiny adaptation, while slightly lower in accuracy, is 5× slower due to its higher parameter count. These results confirm that the domain-specific design of DecoderNet, its co-designed feature space and compact architecture, is superior to generic classifiers applied to the same feature set.

VI. DISCUSSION

A. Why the Decoder Has Enough Information

A key question is whether predicting the transform mode from post-reception, pre-decoded data is fundamentally feasible. Our results provide an empirical affirmative, but the theoretical justification is worth stating explicitly. When a block is DCT-coded, its coefficients after quantization and dequantization retain the energy-compaction property of the DCT: energy is concentrated in low-frequency positions, and the magnitude spectrum decays with spatial frequency. When a block is transform-skipped, its coefficients are unprocessed spatial-domain residuals (after quantization), which have a fundamentally flatter energy distribution and are not organized by DCT frequency significance. These two distributions are distinct, and a sufficiently expressive classifier can learn to separate them. The hand-crafted features (SF, DAR, and HFEF) provide an efficient basis for this discrimination that the CFE refines further.

B. Implications for Future Codec Design

Our work has two concrete implications for standards development. First, it motivates the inclusion of a decoder-side neural inference module as an optional tool in next-generation codecs, analogous to how VVC includes an optional neural network-based loop filter. Such a module would be described in the standard as a 'transform mode prediction network', and its weights would be signaled at the sequence level or included in a standardized set. The overhead of signaling 18K weights at the sequence level is approximately 150 kbits, negligible for a typical video sequence but non-trivial for very short clips.

Second, and more ambitiously, our work supports the idea that the transform_skip_flag could be omitted from the bitstream for blocks where the predictor confidence exceeds a threshold, with the encoder using a rate-distortion criterion to decide whether to transmit the flag or rely on the predictor. This 'conditional signaling' scheme would reduce bitrate for TSM-heavy content (screen content) while preserving full compatibility for natural video where TSM is rare.

C. Limitations and Failure Modes

DecoderNet's accuracy drops to approximately 88% at very high QP (QP = 37) for natural content. At high quantization, both TSM and DCT-coded blocks lose distinctive features, quantized coefficients become sparse and close to zero in both cases, reducing the discriminability of spectral flatness and energy ratio features. This is a fundamental limitation of any coefficient-based classifier and could be partially addressed by incorporating motion compensation residual statistics or inter-frame consistency checks.

Additionally, our current design is validated only for HEVC's 4×4 TSM blocks. VVC's larger TSM block sizes (up to 32×32 in some profiles) present a more challenging and higher-dimensional input space. Preliminary experiments with block-adaptive input pooling show promise for these larger blocks, but a full VVC evaluation is deferred to future work.

D. Privacy and Security Considerations

An adversary aware of the DecoderNet predictor could craft bitstreams that deliberately confuse the classifier, for example, encoding TSM blocks with energy distributions resembling DCT-coded blocks, causing the decoder to apply IDCT and producing an incorrect reconstruction. This is a relevant threat model in adversarial content delivery scenarios. Future work should evaluate robustness to such adversarial bitstreams and consider adversarial training of DecoderNet.

VII. CONCLUSION

This paper introduced DecoderNet, the first neural network designed specifically to predict the transform skip flag at the video decoder side. Operating on quantized residual coefficient statistics and local block context, data already available before the transform step, DecoderNet achieves 94.7% average binary classification accuracy across HEVC standard test sequences with only 18K

parameters and 0.27 ms of per-frame overhead. Deployed as an error-resilience module, it recovers up to 3.1 dB of PSNR compared to conventional concealment under 5% packet loss conditions, demonstrating that decoder-side intelligence can meaningfully reduce the impact of bitstream errors on visual quality.

DecoderNet addresses a gap in the literature that has been overlooked despite the growing interest in machine-learning-enhanced video codecs: all prior ML work on transform mode prediction operates at the encoder. Our results show that the decoder, even with access only to quantized and potentially corrupted data, possesses sufficient information to infer the encoding decision with high reliability. This capability can serve two roles: improving error resilience today, and enabling implicit flag signaling in the codecs of tomorrow.

Future work will extend DecoderNet to VVC's richer transform mode vocabulary (including LFNST and MTS selection prediction), evaluate adaptive threshold tuning for the confidence-based conditional signaling scheme, and investigate adversarial robustness. The decoder-side inference paradigm opened by this work has potential beyond transform mode, similar approaches may be applicable to coding unit partitioning verification, prediction mode disambiguation, and quantization step recovery.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.

[2] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] M. Wien, R. Joshi, and G. J. Sullivan, "Transform skip mode in H.265/HEVC," in Proc. IEEE Int. Conf. Image Process. (ICIP), Melbourne, Australia, Sep. 2013, pp. 1540–1544.

[5] H. Schwarz et al., "Quantization and entropy coding in the versatile video coding (VVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 10, pp. 3891–3906, Oct. 2021.

[6] B. Bross et al., "Overview of the Versatile Video Coding (VVC) standard and its applications," IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[7] J. Park, B. Kim, J. Lee, and B. Jeon, "Machine learning-based early skip decision for intra subpartition prediction in VVC," IEEE Access, vol. 10, pp. 111052–111065, 2022.

[8] J. Park, B. Kim, J. Lee, and B. Jeon, "Learning-based early transform skip mode decision for VVC screen content coding," IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 8, pp. 4023–4036, Aug. 2023.

[9] J. Park, S. Kim, and B. Jeon, "Transform skip mode decision and signaling method for HEVC screen content coding," in Proc. IEEE Int. Conf. Image Process. (ICIP), Phoenix, AZ, Sep. 2016, pp. 1469–1473.

[10] J. Ma et al., "Image and video compression with neural networks: A review," IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 6, pp. 1683–1698, Jun. 2020.

[11] B. Bross et al., "Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding," Proc. IEEE, vol. 109, no. 9, pp. 1463–1493, Sep. 2021.

[12] A. G. Busson and V. L. S. Mendes, "Video quality enhancement using deep learning-based prediction models for quantized DCT coefficients in MPEG I-frames," arXiv:2010.05760, Oct. 2020.

[13] A. Tissier, W. Hamidouche, S. B. D. Mdalsi, J. Vanne, F. Galpin, and D. Menard, "Machine learning based efficient QT-MTT partitioning scheme for VVC intra encoders," IEEE Trans. Circuits Syst. Video Technol., vol. 33, pp. 4279–4293, 2023.

[14] L. He, S. Xiong, R. Yang, X. He, and H. Chen, "Low-complexity multiple transform selection combining multi-type tree partition algorithm for versatile video coding," Sensors, vol. 22, no. 15, p. 5523, 2022.

[15] X. Ge, T. Wang, and P. Ren, "Task-aware encoder control for deep video compression," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, Jun. 2024.

[16] A. Chadha, A. Abbas, and Y. Andreopoulos, "Video classification with CNNs: Using the codec as a spatio-temporal activity sensor," IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 7, pp. 2106–2119, Jul. 2019.

[17] Y. Wang, Q. Zhu, and L. Shaw, "Packet video error resilience: tools and methods," in Proc. IEEE Int. Conf. Multimedia Expo (ICME), Lausanne, Switzerland, Aug. 2002, pp. 341–344.

[18] M. Weng and N. Deligiannis, "Deep learning-based bitstream error correction for CSI feedback," IEEE Wireless Commun. Lett., vol. 10, no. 12, pp. 2847–2851, Dec. 2021.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learn. Represent. (ICLR), San Diego, CA, May 2015.