

TERRACODEC: COMPRESSING OPTICAL EARTH OBSERVATION DATA

Anonymous authors

Paper under double-blind review

ABSTRACT

Earth observation (EO) satellites produce massive streams of multispectral image time series, posing pressing challenges for storage and transmission. Yet, learned EO compression remains fragmented and lacks publicly available, [large-scale](#) pretrained codecs. [Moreover, prior work has largely focused on image compression, leaving temporal redundancy and EO video codecs underexplored.](#) To address [these gaps](#), we introduce *TerraCodec* (TEC), a family of learned codecs pretrained on [Sentinel-2 EO data](#). TEC includes efficient multispectral image variants and a Temporal Transformer model (TEC-TT) that leverages dependencies across time. To overcome the fixed-rate setting of today’s neural codecs, we present Latent Repacking, a novel method for training flexible-rate transformer models that operate on varying rate-distortion settings. TerraCodec outperforms classical codecs, achieving $3-10\times$ stronger compression at equivalent image quality. Beyond compression, TEC-TT enables zero-shot cloud inpainting, surpassing state-of-the-art methods on the AllClear benchmark. Our results establish [EO-trained neural codecs and temporal compression](#) as a promising direction for Earth observation. Code and model weights will be released under a permissive license.

1 INTRODUCTION

The exponential growth of Earth Observation (EO) data, driven by initiatives such as the Copernicus program, creates critical bottlenecks in storage, transmission, and processing (Guo et al., 2016; Wilkinson et al., 2024). EO imagery also differs fundamentally from natural images. It is multispectral, with up to dozens of channels beyond the visible RGB spectrum; and multi-temporal, with images captured at regular intervals from near-constant viewpoints but subject to seasonal, atmospheric, and cloud cover changes. As a result, EO scenes contain strong spatial and spectral redundancy, while temporal evolution arises from recurring seasonal patterns rather than object motion. These properties create compression challenges distinct from natural images, but make EO well-suited for learned approaches that capture domain-specific priors (Gomes et al., 2025). Despite advances in neural codecs for natural images and video (Ballé et al., 2017; Agustsson et al., 2020), [neural compression for EO remains fragmented, with no available large-scale pretrained codecs for multispectral imagery, and temporal redundancy in satellite time series still largely unexplored.](#)

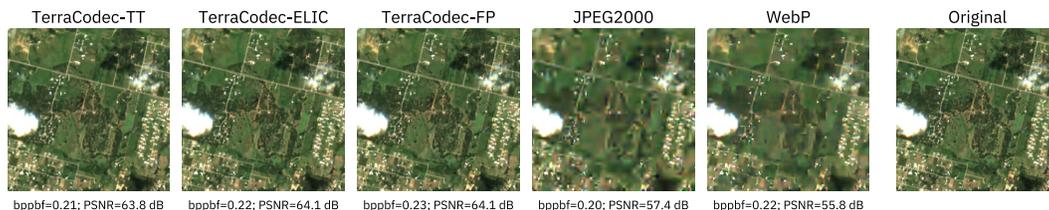


Figure 1: Varying reconstruction quality at a similar compression rate.

We address these gaps with *TerraCodec*, a family of neural codecs tailored to [multispectral EO and pretrained on Sentinel-2 data](#). TerraCodec includes efficient image-based models for multispectral inputs: a lightweight Factorized Prior variant (TEC-FP) and an ELIC-based variant (TEC-ELIC) for optimal rate-distortion. We further introduce a Temporal Transformer (TEC-TT) that captures

054 long-range dependencies across time without relying on hand-crafted motion priors. As shown in
055 Figure 1, these models offer a significantly better reconstruction than standard codecs at the same
056 compression rate. At equal image quality, they reduce storage size by up to an order of magnitude,
057 with TEC-TT providing further reduction on long time series by exploiting temporal structure.

058 Our main contributions are: (1) **TerraCodec**, a suite of [Sentinel-2 pretrained](#) multispectral and
059 multi-temporal codecs that [enable evaluating the benefits of temporal modelling for EO compression](#)
060 and achieve superior rate–distortion performance over classical codecs; (2) **Latent Repacking**, a
061 method to train variable-rate neural codecs which we demonstrate with our FlexTEC model; and
062 (3) **downstream evaluations**, demonstrating the utility of compression models for downstream tasks
063 and zero-shot cloud inpainting. We release code and pretrained weights under a permissive license
064 to support future research and adoption.

065 2 RELATED WORK

066 **Foundations.** Shannon’s source coding theorem bounds lossless compression by the source entropy;
067 practical schemes such as Huffman and arithmetic coding approach this limit (Shannon, 1948; Huff-
068 man, 1952; Rissanen & Langdon, 1979). Lossy compression, in contrast, reduces storage require-
069 ments by discarding information. The rate–distortion function characterizes the minimum bitrate for
070 a given distortion, formalizing the trade-off between rate and fidelity (Shannon, 1948). These prin-
071 ciples underpin transform coding, which applies DCT or wavelets prior to quantization and entropy
072 coding, forming the basis of standards like JPEG, JPEG2000, and HEVC (x265) (Ahmed et al.,
073 1974; Daubechies, 1992; Wallace, 1991; Taubman & Marcellin, 2002; Sullivan et al., 2012).

074 **Neural compression.** Learned codecs replace hand-crafted transforms with autoencoders trained
075 end-to-end under a rate–distortion loss (Ballé et al., 2017; Theis et al., 2017). Inputs are mapped
076 to latents, quantized, and entropy-coded under a learned prior. Recent work extends this beyond
077 convolutional autoencoders, exploring transformer backbones (Zhu et al., 2022; Li et al., 2024a)
078 [generative decoders](#) (Yang & Mandt, 2023; Li et al., 2025) and [latent diffusion models](#) (Zhou et al.,
079 2025), as well as richer perceptual and adversarial objectives (Blau & Michaeli, 2019; Mentzer et al.,
080 2020). A key performance trade-off however is governed by the entropy model. Fully factorized pri-
081 ors offer efficiency (Ballé et al., 2017); hyperpriors (Ballé et al., 2018) introduce side information to
082 capture spatially varying scales; autoregressive priors (Minnen et al., 2018) exploit local context at
083 the cost of sequential decoding; and models such as ELIC (Cheng et al., 2020; He et al., 2022) addi-
084 tionally utilize efficient, parallel space–channel context. However, these image codecs are limited to
085 a single rate–distortion setting per checkpoint. In contrast, flexible-rate models are using approaches
086 such as conditioning on the rate parameter (Choi et al., 2019), spatially adaptive quality maps (Song
087 et al., 2021; Tong et al., 2023), and hierarchical VAEs with quantization-aware priors (Duan et al.,
088 2023).

089 Beyond images, learned video compression targets temporal redundancy across frames (Agustsson
090 et al., 2020; Li et al., 2023; 2024b). While classical approaches rely on motion estimation and com-
091 pensation, transformer-based models remove such priors and model temporal dependencies directly
092 in latent space. The Video Compression Transformer (VCT) (Mentzer et al., 2022) follows this de-
093 sign, encoding frames independently and using a temporal transformer to predict latents from past
094 context, making it better suited to settings with limited or irregular motion.

095 [A complementary line of research investigates Implicit Neural Representations \(INRs\), which fit](#)
096 [a small network to each image or video and store the signal in its weights, achieving strong](#)
097 [rate–distortion performance but requiring per-sample optimization](#) (Kim et al., 2024; Gao et al.,
098 2025; Zhang et al., 2024a; Li et al., 2024c).

099 **Earth Observation data.** Most neural compression targets natural imagery, whereas EO includes
100 multispectral bands, higher bit depth, and long temporal horizons. Compression must preserve
101 spectral and structural cues relevant for downstream analysis (Gomes et al., 2025), aligning with
102 the broader paradigm of task-oriented compression (Torfason et al., 2018; Singh et al., 2020). Op-
103 erational EO pipelines typically rely on the JPEG2000 and CCSDS standards (Yeh et al., 2005;
104 CCS, 2012; 2017) for their robustness and low complexity. Learned models have been explored
105 for optical and SAR (Maharjan & Li, 2023; Di et al., 2022) images, with a focus on reducing on-
106 board complexity (Alves de Oliveira et al., 2021) and spectral grouping (i Verdú et al., 2023). Other
107

works exploit spatial–spectral encoders and mixed hyperpriors to capture redundancy (Kong et al., 2021b;a; Cao et al., 2022; Xiang & Liang, 2023; Fu & Du, 2023; Gao et al., 2023). Recent generative approaches focus on low-bitrate RGB imagery using diffusion models (Ye et al., 2025; Zhang et al., 2024b), and INR–based methods have been explored for multispectral data (Cho et al., 2024). Despite this progress, EO compression research has predominantly focused on single-image settings and often relies on RGB or small-scale datasets, with limited exploration of temporal modeling. To our knowledge, no pretrained models are publicly available for the widely used Sentinel–2 imagery. TerraCodec aims to fill these gaps and offers multispectral neural codecs, a temporal transformer model to capture long-range dependencies, and single-checkpoint, flexible-rate compression.

3 METHODOLOGY

We begin with an overview of our EO compression approach, then detail the architectures of the TerraCodec models 3.1, and finally introduce Latent Repacking for flexible-rate models 3.2.

We study *lossy compression* of multispectral, multi-temporal EO imagery. An EO sequence is a set of images $\mathbf{x}_i \in \mathbb{R}^{H \times W \times C}$, each of size $H \times W$ with C spectral bands. While EO sensors range from a single panchromatic channel to finely sliced hyperspectral imagers, we focus on Sentinel–2 L2A with $C=12$ optical bands in the visible and near infrared, saved as 16-bit radiometry.

A learned codec encodes a frame via an analysis transform $\mathbf{y}_i = g_a(\mathbf{x}_i)$, then quantizes and entropy-codes the latents $\hat{\mathbf{y}}_i = \mathcal{Q}(\mathbf{y}_i)$. The synthesis transform reconstructs the frame, $\hat{\mathbf{x}}_i = g_s(\hat{\mathbf{y}}_i)$. Compression relies on an entropy model $q_\phi(\hat{\mathbf{y}})$ that approximates the unknown latent distribution $p(\hat{\mathbf{y}})$, so arithmetic coding spends, in expectation, the cross-entropy $R \approx \mathbb{E}_{\hat{\mathbf{y}}_i \sim p}[-\log_2 q_\phi(\hat{\mathbf{y}}_i)]$. We train g_a , g_s , and q_ϕ end-to-end with the rate–distortion loss $\mathcal{L} = R + \lambda D$, where D denotes reconstruction error between \mathbf{x}_i and $\hat{\mathbf{x}}_i$, which in our case is measured as MSE in standardized space. As q_ϕ better approximates p , the achieved rate R approaches the entropy $H(p) = \mathbb{E}_p[-\log_2 p]$. Training thereby learns the entropy model while also shaping the latent space to be more predictable under q_ϕ .

EO-specific choices. Our codecs adopt three EO-specific design choices: (i) native support for 12-band, 16-bit inputs; (ii) pretraining on a large-scale global EO dataset; and (iii) per-band standardization rather than global normalization, to stabilize training and preserve band-specific statistics for downstream tasks. Following prior literature (Ballé et al., 2017; Mentzer et al., 2022), we adopt CNN-based encoders for intra-frame compression due to their efficiency and low latency.

3.1 TERRACODEC

We introduce *TerraCodec*, a family of learned codecs for EO, including two image codecs: a lightweight factorized prior (TEC-FP), a stronger space–channel context model (TEC-ELIC), and a temporal transformer (TEC-TT), with a flexible-rate variant (FlexTEC).

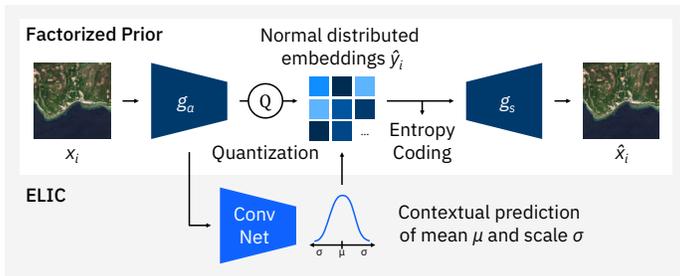


Figure 2: TerraCodec image codecs. Factorized Prior (Ballé et al., 2017) uses a fully factorized prior without context, and assumes a zero-centered normal distribution for the latents. ELIC (He et al., 2022) augments a hyperprior with spatial and channel context to predict per-latent mean/scale.

Factorized Prior (TEC-FP). TEC-FP is a Factorized Prior model (Ballé et al., 2017), our most basic TEC image codec. It employs a fully factorized entropy model, where each element of the quantized latent $\hat{\mathbf{y}}$ is modeled independently by $q_\phi(\hat{y}_j)$, without side information or context. It is illustrated

in the upper part of Figure 2. This yields fast, parallel entropy coding, but limited expressiveness compared to hyperprior- or context-based models.

Efficient Learned Image Compression (TEC-ELIC). TEC-ELIC instantiates ELIC’s space-channel context entropy model (He et al., 2022) for EO inputs. The encoder/decoder networks include residual bottleneck and attention blocks, increasing representational capacity. The entropy model predicts per-latent mean and scale from (i) spatial context via checkerboard convolutions, (ii) channel context from previously decoded latent groups, and (iii) side information from a hyperprior, improving rate–distortion performance at the cost of higher complexity. Figure 2 illustrates the hyperprior-based context model in simplified form.

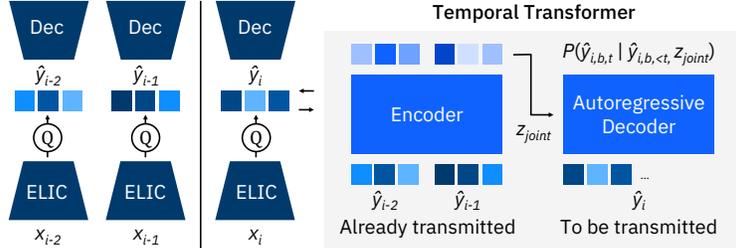


Figure 3: Architecture of the TerraCodec-TT model following Mentzer et al. (2022). Each input image is first encoded into latents by an ELIC image encoder. The per-image latents are tokenized, and a temporal transformer models these tokens autoregressively, predicting the mean and scale parameters for the current frame’s tokens based on past latents.

Temporal Transformer (TEC-TT). TEC-TT builds on the VCT architecture (Mentzer et al., 2022). We train a transformer to model temporal dependencies of seasonal EO data in latent space, predicting the current frame’s latent distribution from past context. Each frame \mathbf{x}_i is encoded to latents \mathbf{y}_i and quantized. We partition the current latent into B non-overlapping spatial blocks $\{\hat{\mathbf{y}}_{i,b}\}_{b=1}^B$ and the two past latents into overlapping context blocks to increase the transformer’s receptive field. Each block b is flattened into a sequence of T tokens $\{\hat{y}_{i,b,t}\}_{t=1}^T$ of channel width d_{lat} . A temporal encoder aggregates the two previous frames into a joint context embedding $z_{\text{joint}} = E(\hat{\mathbf{y}}_{i-2}, \hat{\mathbf{y}}_{i-1})$. As shown in Figure 3, within each current block, a masked autoregressive transformer predicts token-wise prior parameters conditioned on already decoded tokens and z_{joint} following Eq. 1.

$$p(\hat{\mathbf{y}}_{i,b,t} | \hat{\mathbf{y}}_{i,b,<t}, z_{\text{joint}}) = \prod_{d=1}^{d_{\text{lat}}} \mathcal{N}(\hat{y}_{i,b,t}^{(d)}; \mu_{i,b,t}^{(d)}, (\sigma_{i,b,t}^{(d)})^2) \quad (1)$$

We assume conditional independence across blocks given the context, allowing parallel probability estimation during encoding and parallel block decoding. Causal masking prevents attention to undecoded tokens; see Appendix B.3 for details. TEC-TT uses the same CNN analysis–synthesis transforms as TEC-ELIC. Unlike the original VCT, it is trained end-to-end on the rate–distortion objective without image pretraining, using a λ -schedule that emphasizes low-rate regimes early. We further adapt TEC-TT for flexible-rate scaling, introducing the FlexTEC variant in the next section.

3.2 LATENT REPACKING FOR FLEXIBLE-RATE MODELS

Most neural codecs are trained for a fixed rate–distortion tradeoff. This makes deployment inflexible since achieving different bitrates requires retraining separate models. Our goal is to support variable rates at inference. We introduce Latent Repacking, which redistributes latent channels across tokens, and apply token masking with dynamic rate scaling during training so tokens learn an information-based ordering. Early tokens capture global structure, later ones refine detail. Truncating tokens then lowers bitrate while preserving global content. We demonstrate this by adapting TEC-TT, where strong priors allow missing tokens to be predicted, making it well-suited for Latent Repacking.

From spatial tokens to channel slices. A standard transformer codec represents an image block with T spatial tokens, each spanning the full latent dimension d_{lat} . Dropping tokens at inference discards entire regions, causing severe artifacts (see App. F.2.1). Instead, we aim for early tokens to encode information that is *globally useful* across the scene.

Think of the latent block as a 3D tensor $A \in \mathbb{R}^{H \times W \times d_{\text{lat}}}$, with $H \times W = T$ tokens and latent dimension d_{lat} . In the standard layout, each token t is a spatial patch (h, w) containing all d_{lat} channels at that location. Latent repacking instead slices the channel axis into T groups of width $k = \frac{d_{\text{lat}}}{T}$, and redefines tokens so that each spans the full scene but only k channels.

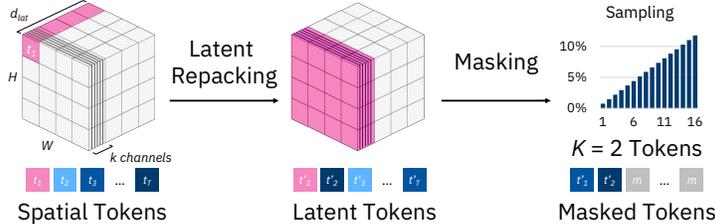


Figure 4: Latent Repacking converts T spatial tokens ($W \cdot H$) into channel-slice tokens so each token carries scene-wide content. During training, we sample a token budget K , mask the rest using a learned token m , and scale the rate. During inference, the user can pick the compression level K .

Formal definition. We define T new tokens $\{t'_1, \dots, t'_T\}$, each formed by a slice of k channels across all spatial positions. Concretely, the u -th repacked token is

$$t'_u = A[:, :, (u - 1) \cdot k : u \cdot k] \in \mathbb{R}^{H \times W \times k}. \tag{2}$$

In other words, t'_u is the u -th slice of k consecutive channels of A , spanning the full spatial field. The procedure is reversible; reapplying the slicing and repacking restores the original layout. After repacking, keeping the first K tokens corresponds to $A[:, :, 0 : K \cdot k]$, i.e. the first $K \cdot k$ latent channels at every spatial location.

Masked training. In order for a flexible-rate model to learn varying rate settings, we mask the repacked tokens during training. Therefore, we sample a token budget $K \in \{1, \dots, T\}$ and mask the last $T - K$ tokens by replacing them with a learned mask token m . Let $M_u \in \{0, 1\}$ indicate masking ($M_u = 1$ if token t'_u is kept). Assuming an autoregressive model with additional context c , the rate is computed only over unmasked tokens as shown in Eq. 3.

$$R(M) = \sum_u M_u \circ \left[-\log_2 q_\phi(t'_u | t'_{<u}, c) \right] \tag{3}$$

To keep later tokens informative, we use *dynamic rate scaling*: when fewer tokens are kept, their rate loss weight is upweighted. It prevents information from collapsing into the first tokens and encourages useful content across all tokens. Budgets K are sampled more frequently at higher values (as in Bachmann et al. (2025)), ensuring that *all* tokens are trained while the model also learns to operate across a range of rates.

We apply the approach on TEC-TT and introduce the **Flexible-Rate TerraCodec (FlexTEC)** model. The model applies Latent Repacking and masking inside the temporal transformer after image-wise compression and restores the original layout before image decoding (details in App. C).

4 EXPERIMENTAL SETUP

This section describes the data and pretraining (4.1), evaluation and baselines (4.2), and downstream tasks (4.3) used to train and assess TerraCodec.

4.1 PRETRAINING

All TEC models are pretrained on SSL4EO-S12 v1.1 (Blumenstiel et al., 2025; Wang et al., 2023), a large-scale Sentinel-2 corpus with 244k globally distributed locations and four seasonal snapshots per location. Each L2A sample consists of 264×264 pixels at 10 m resolution; we crop to 256×256 pixels (random crops for training, center crops for evaluation) to ensure uniform size. Bands at 20 m and 60 m are upsampled to 10 m with nearest-neighbor interpolation for spatial alignment. We

270 follow the official spatial split into training and validation sets. To stabilize multispectral training,
 271 each input band b is standardized by its dataset mean μ_b and standard deviation σ_b . Losses are com-
 272 puted in this standardized space, which balances gradient magnitudes across channels and avoids
 273 overfitting to high-variance bands.

274 **Image codecs.** TEC-FP and TEC-ELIC are optimized with Adam ($lr = 10^{-4}$), using an auxiliary
 275 learning rate of $5 \cdot 10^{-3}$ for the entropy bottleneck, gradient clipping at 1.0, and mixed precision. We
 276 employ a cosine learning-rate schedule with 5% warmup and $\eta_{\min} = 10^{-5}$. Models are trained for
 277 100 epochs with batch size 64 on a single NVIDIA A100 GPU, requiring 20–25 hours. A temporal
 278 index is randomly sampled for each sample in every epoch, so that one epoch covers one quarter of
 279 the dataset. We sweep five λ values to span low- to high-bitrate regimes.

280 **Temporal codec.** TEC-TT is trained with a temporal context of two past frames for 300k steps with
 281 a global batch size of 24 on four NVIDIA A100 GPUs, requiring about 70 hours. We optimize with
 282 AdamW ($lr = 10^{-4}$, weight decay = 10^{-2}) and employ half-cosine decay schedule to $\eta_{\min} = 10^{-6}$
 283 with 15% warmup steps. We sweep six λ values for varying rate–distortion settings. For low-rate
 284 settings ($\lambda \leq 5.0$), we scale λ by 10 during the first 15% of training before annealing back to the
 285 target value. Further pretraining details, including FlexTEC, are given in Appendix B.

287 4.2 EVALUATION

289 **Baselines.** We compare TerraCodec to widely used classical codecs: JPEG (Wallace, 1991),
 290 JPEG2000 (Taubman & Marcellin, 2002), WebP (Zern et al., 2024), and HEVC (x265) (Sullivan
 291 et al., 2012), using the highest bit support for each codec. For JPEG and WebP we use the Pillow
 292 library; for JPEG2000 we use Glymur; and for HEVC we use the ffmpeg x265 implementation with
 293 the *medium* preset tuned for PSNR. Since these codecs are limited to three-channel RGB, we apply
 294 them per band, encoding each spectral channel as an independent grayscale image, following prior
 295 work in EO compression (Radosavljevic et al., 2020; Grassa et al., 2022).

296 **Metrics.** Compression rate is reported as *bits-per-pixel-band-frame* (bppbf), which normalizes by
 297 spatial resolution, number of spectral bands, and sequence length. Unlike the standard *bits-per-pixel*
 298 (bpp) used for three-channel RGB images, bppbf extends to arbitrary channel counts and sequence
 299 lengths. Distortion is quantified using PSNR, SSIM (Wang et al., 2004), MS-SSIM (Wang et al.,
 300 2003), and MSE in the destandardized 16-bit reflectance space. To ensure equal contribution of
 301 all spectral channels, metrics are computed per band and averaged across channels, avoiding bias
 302 toward bands with higher variance.

304 4.3 DOWNSTREAM TASKS

306 **Cloud inpainting.** TEC-TT captures spatiotemporal priors that can be applied beyond compression.
 307 We demonstrate this with zero-shot cloud removal on the AllClear benchmark (Zhou et al., 2024).
 308 Each sample consists of three cloudy observations with masks and one cloud-free target. To apply
 309 TEC-TT, we use the two least cloudy images as a temporal context and extract all cloud-free patches
 310 from the least cloudy image as input x_0 . Patches in x_0 that are covered by clouds are predicted by
 311 TEC-TT. We report PSNR and other metrics, comparing against the official AllClear baselines.
 312 Following the benchmark, metrics are computed in auto mode across all bands, not per-band.

313 **Downstream models on compressed data.** To evaluate the effect of compression on downstream
 314 analysis, we benchmark downstream task models on compressed–reconstructed versus uncom-
 315 pressed inputs. Following standard EO practice, we finetune task-specific encoder–decoder models
 316 on either reconstructed or original inputs. We evaluate on Sen1Floods11 (Bonafilia et al., 2020),
 317 consisting of 512^2 patches from 11 flood events, with binary segmentation masks. The original
 318 dataset includes Sentinel–2 L1C imagery. We, therefore, redownload the L2A version for TEC-FP.
 319 For patchwise multi-label land cover classification, we use reBEN-7k (Marti-Escofet et al., 2025),
 320 which spans eight countries and 19 semantic labels. Images have 120^2 pixels; we apply *reflect*
 321 padding to match the model input sizes. We compress all inputs at three different operating points
 322 using TEC-FP, then fine-tune pretrained models on the reconstructed data. We employ Prithvi 2.0
 323 100M (Szwarcman et al., 2025) and TerraMind base (Jakubik et al., 2025) backbones, with a UNet
 decoder for segmentation and a linear head for classification. All models are trained for 100 epochs
 with AdamW ($lr = 5 * 10^{-5}$) and a *reduce-on-plateau* scheduler.

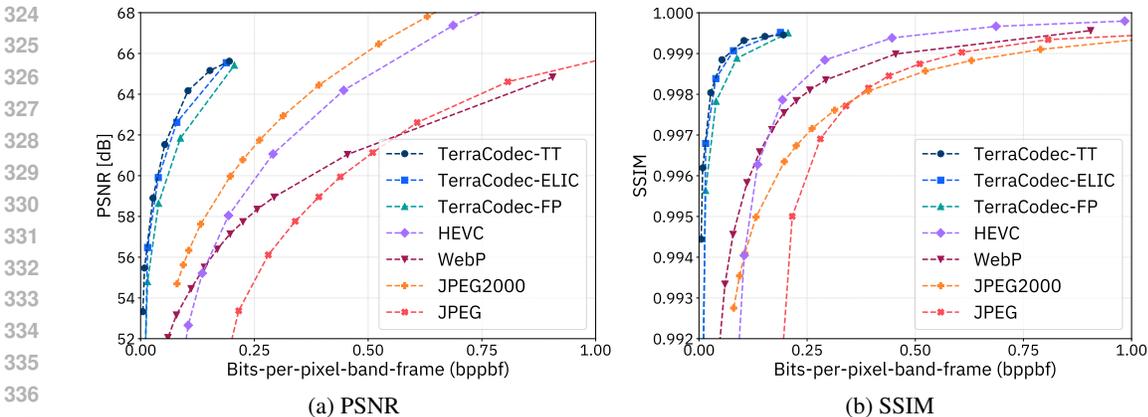


Figure 5: Rate–distortion curves for PSNR (\uparrow) and SSIM (\uparrow) on SSL4EO-S12 v1.1 validation sequences. TerraCodec models consistently outperform standard codecs, with TEC-TT achieving the best overall performance by exploiting temporal context.

5 EXPERIMENTS

We evaluate TerraCodec in terms of rate–distortion (Section 5.1), flexible-rate compression (Section 5.2), and utility for downstream EO tasks (Section 5.3).

5.1 RATE–DISTORTION

Figure 5 reports RD curves (bppbf vs. PSNR and SSIM) on the SSL4EO-S12 v1.1 validation set. TerraCodec consistently outperforms classical image and video codecs in both metrics. The lightweight TEC-FP achieves up to $5\times$ lower rate than the best image codec WebP at an equal SSIM of 0.999, while TEC-ELIC and TEC-TT provide further compression gains. Qualitative reconstructions (App. F.1) highlight that TerraCodec preserves finer structures and details compared to classical codecs. JPEG2000 achieves competitive PSNR with only $3\times$ lower compression rate at similar distortion – surpassing HEVC – but performs poorly in SSIM, especially at high quality. This arises from its tendency to preserve pixel averages (favored by PSNR/MSE) while oversmoothing textures and edges that SSIM is sensitive to. In contrast, TerraCodec models maintain strong performance across both metrics, demonstrating efficient compression without loss of high-frequency details.

TEC-TT improves over the neural image models, although its margin over TEC-ELIC is smaller than the typical video–image gap. While EO sequences differ from natural video, being sampled at daily to seasonal rather than sub-second intervals, the short 4-frame validation setup further departs from typical video settings.

This limits temporal gains since half of the frames are *bootstrap* frames, which lack full previous context, compared to *P-frames*, which are predicted from two past frames. To better understand these effects, we study the role of temporal conditioning and report the RD performance on *P-frames only* in Figure 6. Without context, TEC-TT reduces to an image codec (labeled as *image only*). For a medium setting ($\lambda = 5$), conditioning on one past frame improves compression by 13.6% compared to no context. On *P-frames* with two previous images, TEC-TT achieves a 22.6% rate reduction at equal PSNR, showing that longer EO sequences naturally yield greater efficiency as bootstrap frames are amortized.

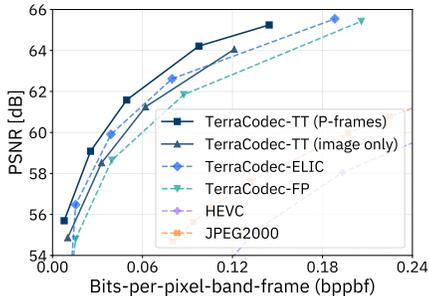
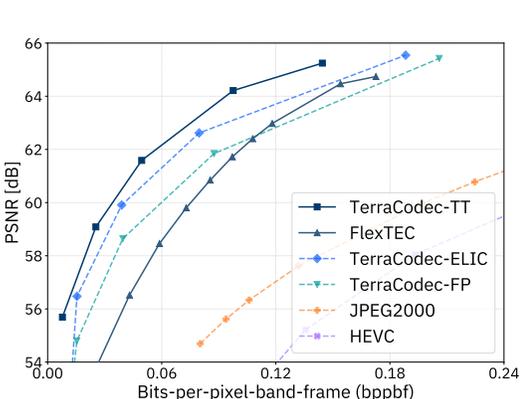


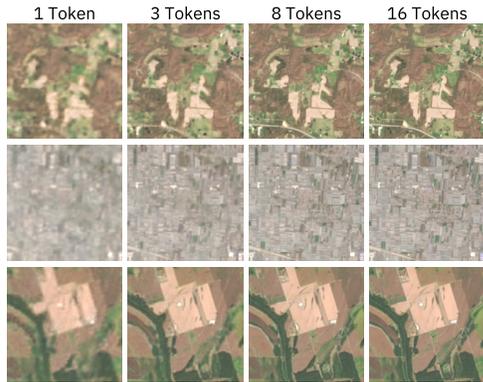
Figure 6: Rate–distortion curves for PSNR (\uparrow) on all evaluation *P-frames*.

378 5.2 FLEXIBLE RATE-DISTORTION
379

380 Our TEC-TT-based FlexTEC model uses Latent Repacking to provide flexible rate-compression
381 from a single checkpoint by transmitting a variable subset of latents and inferring the remainder
382 from the model prior. Figure 7 compares FlexTEC against our fixed-rate models and standard codecs
383 on P-frame compression. While fixed-rate TEC-TT models serve as an upper bound—each being
384 optimally fitted for one specific rate-distortion—they require several separately trained models. In
385 contrast, FlexTEC provides several user-controlled rate settings and performs close to or better than
386 TEC-FP, depending on the setting. Further analysis in Appendix E.3 shows that FlexTEC encodes
387 significant information in bootstrap frames, leading to similar bitrates independent of the token
388 budget. The model then uses this information in the following P-frames to provide efficient rate-
389 distortion settings. Figure 8 provides qualitative examples for different token budgets.



404 Figure 7: RD curves for PSNR (↑) on P-frames. FlexTEC performs close to fixed-rate models and
405 significantly better than standard codecs.
406



409 Figure 8: FlexTEC reconstructions with different token budgets. Early tokens capture coarse struc-
410 tures, while later tokens refine details.
411

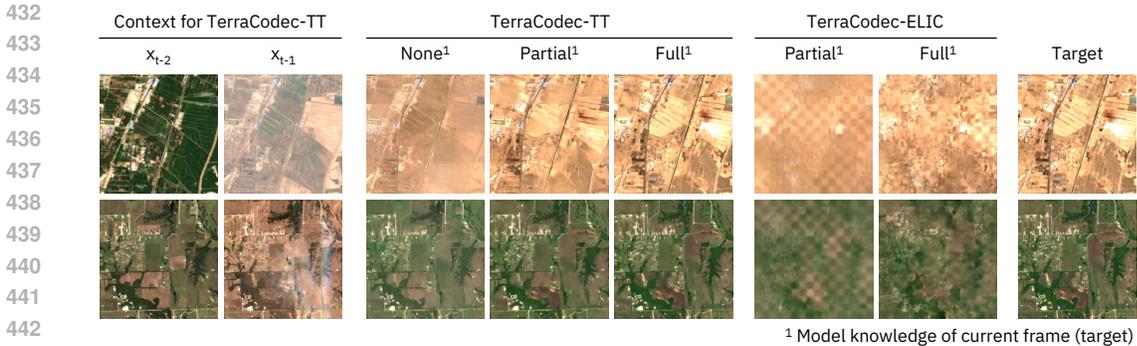
412 5.3 DOWNSTREAM TASKS
413

414 We study the usability of TEC models beyond rate-distortion compression by examining how their
415 learned priors (*model beliefs*) can be leveraged for zero-shot prediction and how compression affects
416 downstream EO applications.
417

418 **Model beliefs.** Neural codecs rely on learned priors to estimate latent distributions for entropy cod-
419 ing. By decoding these priors into the image space, we obtain predictions that expose the model’s
420 implicit knowledge. Figure 9 shows qualitative examples, with additional results in Appendix F.2.2.
421 We compare priors under three conditions: (a) no information from the current frame (only past con-
422 text, TEC-TT), (b) partial information from the current latent (past frames and a subset of tokens for
423 TEC-TT; subsets of channel groups for TEC-ELIC), and (c) full information of the latent. TEC-FP,
424 lacking a hyperprior, cannot adapt to the latent at hand. Results show that TEC-TT already pro-
425 duces plausible forecasts from past frames alone and improves its beliefs when partial information
426 becomes available, yielding the most refined predictions.
427

428 **Cloud inpainting.** Building on TEC-TT’s latent predictions, we evaluate zero-shot cloud removal
429 on the AllClear benchmark (Zhou et al., 2024). TEC-TT is applied without task-specific training (see
430 Sec. 4.3 and Appendix G), with results summarized in Table 1 and qualitative examples in Figure 10.
431 In addition to the full test set, we report performance on subsets ranked by cloud coverage.

The benchmark compares against heuristic baselines (LeastCloudy, Mosaicing) and prior zero-shot
neural approaches. TEC-TT outperforms all heuristic methods and prior zero-shot neural models
on the full test set. The subset analysis, focusing on the most challenging samples in terms of cloud cov-
erage, further underscores the benefit of temporal priors: heuristic approaches perform adequately
in low-cloud cases, but break down under heavy cloud coverage, while TEC-TT maintains high



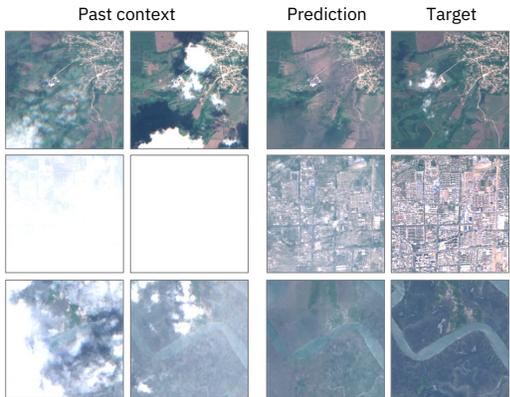
444 Figure 9: Model beliefs obtained by decoding learned priors into the image space. TEC-TT is shown
445 with 0 (past context only), 5 and 16 tokens. TEC-ELIC uses limited and full channel-group context.
446

447
448 PSNR. The hardest 10% of samples correspond to an average cloud coverage of 99%, yet TEC-TT
449 still produces reasonable predictions. Overall, the results demonstrate that the temporal modeling in
450 TEC-TT not only improves compression but also transfers to challenging forecasting tasks.

451 Table 1: Test PSNR on AllClear across difficulty subsets (by average cloudiness). PSNR
452 computed across all bands following AllClear.
453

454
455
456
457
458
459
460
461
462
463
464
465
466

Model	10%	20%	50%	100%
<i>Baseline heuristics</i>				
LeastCloudy	11.07	14.08	24.82	30.61
Mosaicing	16.55	16.70	23.73	29.82
<i>Pre-trained models (zero-shot setting)</i>				
CTGAN	25.58	26.60	27.59	27.79
DiffCR	24.55	25.13	25.50	25.21
PMAA	24.82	25.06	25.02	24.32
U-TILISE	13.20	14.95	18.33	24.67
UnCRtainTS	26.42	26.50	<u>27.97</u>	29.01
TerraCodec-TT	<u>25.97</u>	<u>26.59</u>	30.38	32.86



468 Figure 10: Cloud inpainting examples with
469 TEC-TT on the AllClear benchmark.

470
471
472
473
474
475
476
477

Downstream models. In EO, downstream task models are typically trained on uncompressed data to avoid information loss, requiring to transmit and store large volumes of raw data. We investigate the impact of training and evaluating on data compressed with TerraCodec-FP. Table 2 reports image analysis results on reBEN-7k and Sen1Floods11. Models fine-tuned after moderate compression lead to a performance drop of <1.0pp across all metrics, while reducing data size by up to 380x. At high compression, we observe more pronounced degradations: the F1 score on reBEN-7k decreases by 3.4pp with TerraMind, and the IoU_{Flood} on Sen1Floods11 drops by 2pp with both models. These results suggest that moderate compression can be employed without substantial impact on downstream analysis, whereas higher compression levels entail some performance trade-off.

478 **6 CONCLUSION**
479

480 We introduce and release TerraCodec, a family of learned compression models for Earth observa-
481 tion, pretrained on Sentinel-2 [multispectral time series data](#). Our models outperform classical image
482 and video codecs in rate-distortion, achieving up to an order-of-magnitude reduction at equal qual-
483 ity. Latent Repacking further enables flexible-rate transformer models from a single checkpoint,
484 as demonstrated by FlexTEC. Downstream evaluations show that moderate compression preserves
485 analysis performance, while zero-shot cloud inpainting highlights the strengths of our temporal
transformer TEC-TT beyond compression. Overall, TerraCodec establishes a foundation for [explor-](#)

Table 2: Test performance on reBEN-7k and Sen1Floods11 (\uparrow) when training on compressed inputs (TEC-FP). Numbers in parentheses show the change relative to training on uncompressed data. Performance remains stable at low and mid rates, with clearer degradation for high compression.

Task Model	Compression	reBEN-7k		Sen1Floods11	
		Accuracy	F1	mIoU	IoU _{Flood}
TerraMind base	Original data	88.76	61.99	87.77	78.75
	170 \times	89.05 (+0.29)	63.24 (+1.25)	87.31 (-0.46)	78.02 (-0.73)
	380 \times	88.82 (+0.06)	60.97 (-1.02)	87.27 (-0.50)	77.97 (-0.78)
	940 \times	87.80 (-0.96)	58.60 (-3.39)	86.76 (-1.01)	77.06 (-1.69)
Prithvi 2.0 100M TL	Original data	87.93	59.23	87.27	77.92
	170 \times	87.42 (-0.51)	59.14 (-0.09)	87.06 (-0.21)	77.53 (-0.39)
	380 \times	87.06 (-0.87)	60.15 (+0.92)	86.61 (-0.66)	76.86 (-1.06)
	940 \times	86.86 (-1.07)	58.28 (-0.95)	85.96 (-1.31)	75.81 (-2.11)

[ing the benefits of temporal modelling in EO compression and for advancing multispectral learned compression in Earth observation.](#)

REFERENCES

- CCSDS 123.0-B-1: lossless multispectral and hyperspectral image compression. Technical report, Consultative Committee for Space Data Systems (CCSDS), 2012.
- CCSDS 122.0-B-2: image data compression. Technical report, Consultative Committee for Space Data Systems (CCSDS), 2017.
- Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- Nasir Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 1974.
- Vinicius Alves de Oliveira, Marie Chabert, Thomas Oberlin, Charly Poulliat, Mickael Bruno, Christophe Latry, Mikael Carlavan, Simon Henrot, Frederic Falzon, and Roberto Camarero. Reduced-complexity end-to-end variational autoencoder for on-board satellite image compression. *Remote Sensing*, 2021.
- Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oguzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. FlexTok: resampling images into 1d token sequences of flexible length. *arXiv preprint arXiv:2502.13967*, 2025.
- Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations (ICLR)*, 2017.
- Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations (ICLR)*, 2018.
- Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. CompressAI: a PyTorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.
- Yochai Blau and Tomer Michaeli. Rethinking lossy compression: the rate-distortion-perception tradeoff. In *International Conference on Machine Learning (ICML)*, 2019.
- Benedikt Blumenstiel, Nassim Ait Ali Braham, Conrad M. Albrecht, Stefano Maurogiovanni, and Paolo Fraccaro. SSL4EO-S12 v1.1: a multimodal, multiseasonal dataset for pretraining, updated. *arXiv preprint arXiv:2503.00168*, 2025.

- 540 Derrick Bonafilia, Beth Tellman, Tyler Anderson, and Erica Issenberg. Sen1Floods11: a georefer-
541 enced dataset to train and test deep learning flood algorithms for Sentinel-1. In *Proceedings of*
542 *the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.
- 543 T. Cao, N. Zhang, S. Zhao, K. Hu, and K. Wang. Spectral-spatial feature completely separated
544 extraction with tensor CNN for multispectral image compression. *Lecture Notes in Electrical*
545 *Engineering*, 2022.
- 546 Vincent Chandelier. ELIC implementation. [https://github.com/VincentChandelier/](https://github.com/VincentChandelier/ELiC-ReImplemetation)
547 [ELiC-ReImplemetation](https://github.com/VincentChandelier/ELiC-ReImplemetation), 2023. Accessed: 2025-09-24.
- 548 Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with
549 discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF*
550 *Conference on Computer Vision and Pattern Recognition*, 2020.
- 551 Woojin Cho, Steve Andreas Immanuel, Junhyuk Heo, and Darongsae Kwon. Neural compression
552 for multispectral satellite images. In *Workshop on Machine Learning and Compression, NeurIPS*,
553 2024. URL <https://openreview.net/forum?id=gk2m2OIKMs>.
- 554 Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Variable rate deep image compression with
555 a conditional autoencoder. In *IEEE/CVF International Conference on Computer Vision (ICCV)*,
556 2019.
- 557 Ingrid Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 1992.
- 558 Zhixiong Di, Xuan Chen, Qiang Wu, Jiangyi Shi, Quanyuan Feng, and Yibo Fan. Learned com-
559 pression framework with pyramidal features and quality enhancement for SAR images. *IEEE*
560 *Geoscience and Remote Sensing Letters*, 2022.
- 561 Zhihao Duan, Ming Lu, Zhan Ma, and Fengqing Zhu. Lossy image compression with quantized
562 hierarchical VAEs. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*,
563 2023.
- 564 Chuan Fu and Bo Du. Remote sensing image compression based on multiple prior information.
565 *Remote Sensing*, 2023.
- 566 Ge Gao, Ho Man Kwan, Fan Zhang, and David Bull. Pnvc: Towards practical INR-based video
567 compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. doi: 10.
568 1609/aaai.v39i3.32315. URL <https://doi.org/10.1609/aaai.v39i3.32315>.
- 569 Junjun Gao, Qizhi Teng, Xiaohai He, Zhengxin Chen, and Chao Ren. Mixed entropy model en-
570 hanced residual attention network for remote sensing image compression. *Neural Processing*
571 *Letters*, 2023.
- 572 Carlos Gomes, Isabelle Wittmann, Damien Robert, Johannes Jakubik, Tim Reichelt, et al. Lossy
573 neural compression for geospatial analytics: a review. *IEEE Geoscience and Remote Sensing*
574 *Magazine*, 2025.
- 575 Riccardo La Grassa, Cristina Re, Gabriele Cremonese, and Ignazio Gallo. Hyperspectral data com-
576 pression using fully convolutional autoencoder. *Remote Sensing*, 2022.
- 577 Huadong Guo, Zhen Liu, Hao Jiang, Changlin Wang, Jie Liu, and Dong Liang. Big Earth data: a
578 new challenge and opportunity for Digital Earth’s development. *International Journal of Digital*
579 *Earth*, 2016.
- 580 Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. ELIC: efficient
581 learned image compression with unevenly grouped space-channel contextual adaptive coding. In
582 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- 583 David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of*
584 *the IRE*, 1952.
- 585 Sebastia Mijares i Verdú, Johannes Ballé, Valero Laparra, Joan Bartrina-Rapesta, Miguel
586 Hernández-Cabronero, and Joan Serra-Sagrístà. A scalable reduced-complexity compression of
587 hyperspectral remote sensing images using deep learning. *Remote Sensing*, 2023.
- 588
- 589
- 590
- 591
- 592
- 593

- 594 Johannes Jakubik, Felix Yang, Benedikt Blumenstiel, Erik Scheurer, Rocco Sedona, Stefano Mau-
595 rogiovanni, et al. TerraMind: large-scale generative multimodality for Earth observation. *arXiv*
596 *preprint arXiv:2504.11171*, 2025.
- 597 Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, and Emilien Dupont.
598 C3: High-performance and low-complexity neural compression from a single image or video.
599 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,
600 pp. 9347–9358, 2024. doi: 10.1109/CVPR52733.2024.00893. URL [https://doi.
601 ieeeecomputersociety.org/10.1109/CVPR52733.2024.00893](https://doi.ieeeecomputersociety.org/10.1109/CVPR52733.2024.00893).
- 602 F. Kong, K. Hu, Y. Li, D. Li, and S. Zhao. Spectral–spatial feature partitioned extraction based on
603 CNN for multispectral image compression. *Remote Sensing*, 2021a.
- 604 F. Kong, S. Zhao, Y. Li, and D. Li. End-to-end multispectral image compression framework based
605 on adaptive multiscale feature extraction. *Journal of Electronic Imaging*, 2021b.
- 606 Han Li, Shaohui Li, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Frequency-aware
607 transformer for learned image compression. In *International Conference on Learning Representations (ICLR)*, 2024a.
- 608 Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *Proceedings of*
609 *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
- 610 Jiahao Li, Bin Li, and Yan Lu. Neural video compression with feature modulation. In *Proceedings*
611 *of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.
- 612 Xin Li, Baile Sun, Jixiu Liao, and Xiaofei Zhao. Remote sensing image compression method based
613 on implicit neural representation. In *Proceedings of the 2023 12th International Conference*
614 *on Computing and Pattern Recognition*, pp. 432–439, New York, NY, USA, 2024c. Association
615 for Computing Machinery. doi: 10.1145/3633637.3633706. URL [https://doi.org/10.
616 1145/3633637.3633706](https://doi.org/10.1145/3633637.3633706).
- 617 Zhiyuan Li, Yanhui Zhou, Hao Wei, Chenyang Ge, and Ajmal Saeed Mian. Diffusion-based ex-
618 treme image compression with compressed feature initialization. [https://openreview.
619 net/forum?id=0TSAIUCwpp](https://openreview.net/forum?id=0TSAIUCwpp), 2025. arXiv / OpenReview preprint.
- 620 Paras Maharjan and Zhu Li. Complex-valued SAR image compression: a novel approach for am-
621 plitude and phase recovery. In *IEEE International Conference on Visual Communications and*
622 *Image Processing (VCIP)*, 2023.
- 623 Francesc Marti-Escofet, Benedikt Blumenstiel, Linus Scheibenreif, Paolo Fraccaro, and Konrad
624 Schindler. Fine-tune smarter, not harder: parameter-efficient fine-tuning for geospatial foundation
625 models. *arXiv preprint arXiv:2504.17397*, 2025.
- 626 Fabian Mentzer, George Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity gener-
627 ative image compression. In *Advances in Neural Information Processing Systems*, 2020.
- 628 Fabian Mentzer, George Toderici, David Minnen, Sergi Caelles, Sung Jin Hwang, Mario Lucic, and
629 Eirikur Agustsson. VCT: a video compression transformer. In *Advances in Neural Information*
630 *Processing Systems*, 2022.
- 631 David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors
632 for learned image compression. In *Advances in Neural Information Processing Systems*, 2018.
- 633 Matthew Muckley, Jordan Juravsky, Daniel Severo, Mannat Singh, Quentin Duval, and
634 Karen Ullrich. Neural compression. [https://github.com/facebookresearch/
635 NeuralCompression](https://github.com/facebookresearch/NeuralCompression), 2021.
- 636 Milos Radosavljevic, Branko Brkljac, Predrag Lugonja, Vladimir S. Crnojevic, Zeljen Trpovski,
637 Zixiang Xiong, and Dejan Vukobratovic. Lossy compression of multispectral satellite images
638 with application to crop thematic mapping: a HEVC comparative study. *Remote Sensing*, 2020.
- 639 Jorma Rissanen and Glen G. Langdon. Arithmetic coding. *IBM Journal of Research and Develop-*
640 *ment*, 1979.

- 648 Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*,
649 1948.
- 650
- 651 Saurabh Singh, Sami Abu-El-Haija, Nick Johnston, Johannes Ballé, Abhinav Shrivastava, and
652 George Toderici. End-to-end learning of compressible features. In *IEEE International Conference*
653 *on Image Processing (ICIP)*, 2020.
- 654 Myungseo Song, Jinyoung Choi, and Bohyung Han. Variable-rate deep image compression through
655 spatially-adaptive feature transform. In *IEEE/CVF International Conference on Computer Vision*
656 *(ICCV)*, 2021.
- 657
- 658 Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high
659 efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video*
660 *Technology*, 2012.
- 661 Daniela Szwarcman, Sujit Roy, Paolo Fraccaro, Thorsteinn Eli Gislason, Benedikt Blumenstiel,
662 Rinki Ghosal, Pedro Henrique de Oliveira, Joao Lucas de Sousa Almeida, et al. Prithvi-EO-2.0:
663 a versatile multi-temporal foundation model for Earth observation applications. *arXiv preprint*
664 *arXiv:2412.02732*, 2025.
- 665
- 666 David S. Taubman and Michael W. Marcellin. JPEG2000: standard for interactive imaging. *Pro-*
667 *ceedings of the IEEE*, 2002.
- 668
- 669 Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with
670 compressive autoencoders. In *International Conference on Learning Representations (ICLR)*,
671 2017.
- 672 George Toderici, Sean M. O’Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet
673 Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent
674 neural networks. *arXiv preprint arXiv:1511.06085*, 2016.
- 675
- 676 Kedeng Tong, Yaojun Wu, Yue Li, Kai Zhang, Li Zhang, and Xin Jin. QVRF: a quantization-error-
677 aware variable rate framework for learned image compression. In *IEEE International Conference*
678 *on Image Processing (ICIP)*, 2023.
- 679 Robert Torfason, Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and
680 Luc Van Gool. Towards image understanding from deep compression without decoding. In
681 *International Conference on Learning Representations (ICLR)*, 2018.
- 682
- 683 Gregory K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*,
684 1991.
- 685 Yi Wang, Nassim Ait Ali Braham, Zhitong Xiong, Chenying Liu, Conrad M Albrecht, and Xiao Xi-
686 ang Zhu. SSL4EO-S12: a large-scale multimodal, multitemporal dataset for self-supervised learn-
687 ing in Earth observation. *IEEE Geoscience and Remote Sensing Magazine*, 2023.
- 688
- 689 Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality as-
690 sessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, 2003*,
691 2003.
- 692
- 693 Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment:
694 from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.
- 695
- 696 R. Wilkinson, M.M. Mleczo, R.J.W. Brewin, K.J. Gaston, M. Mueller, J.D. Shutler, X. Yan, and
697 K. Anderson. Environmental impacts of Earth observation data in the constellation and cloud
698 computing era. *Science of the Total Environment*, 2024.
- 699
- 700 Shao Xiang and Qiaokang Liang. Remote sensing image compression with long-range convolution
701 and improved non-local attention. *Signal Processing*, 2023.
- 702
- 703 Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. In
704 *Advances in Neural Information Processing Systems*, 2023.

702 Yixuan Ye, Ce Wang, Wanjie Sun, and Zhenzhong Chen. Map-assisted remote-sensing image
703 compression at extremely low bitrates. *ISPRS Journal of Photogrammetry and Remote*
704 *Sensing*, 223:159–172, 2025. doi: 10.1016/j.isprsjprs.2025.03.005. URL <https://www.sciencedirect.com/science/article/pii/S0924271625001030>.
705
706 Pen-Shu Yeh, P. Armbruster, A. Kiely, B. Masschelein, G. Moury, C. Schaefer, and C. Thiebaut.
707 The new CCSDS image compression recommendation. In *IEEE Aerospace Conference*, 2005.
708
709 James Zern, Pascal Massimino, and Jyrki Alakuijala. WebP image format. RFC 9649, November
710 2024. URL <https://www.rfc-editor.org/info/rfc9649>.
711
712 L. Zhang, T. Pan, J. Liu, and L. Han. Compressing hyperspectral images into multilayer percep-
713 trons using fast-time hyperspectral neural radiance fields. *IEEE Geoscience and Remote Sensing*
714 *Letters*, 21:1–5, 2024a. doi: 10.1109/LGRS.2024.3365106.
715
716 Ziyuan Zhang, Han Qiu, Zhang Maosen, Jun Liu, Bin Chen, Hewu Li, and Tianwei Zhang. COS-
717 MIC: Compress satellite image efficiently via diffusion compensation. In *Advances in Neural*
718 *Information Processing Systems*, 2024b. URL <https://openreview.net/forum?id=itbKmqreqUZ>.
719
720 Chuqin Zhou, Guo Lu, Jiangchuan Li, Xiangyu Chen, Zhengxue Cheng, Li Song, and Wenjun
721 Zhang. Controllable distortion–perception tradeoff through latent diffusion for neural image
722 compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10725–
723 10733, 2025. doi: 10.1609/aaai.v39i10.33165. URL <https://doi.org/10.1609/aaai.v39i10.33165>.
724
725 Hangyu Zhou, Chia-Hsiang Kao, Cheng Perng Phoo, Utkarsh Mall, Bharath Hariharan, and Kavita
726 Bala. AllClear: a comprehensive dataset and benchmark for cloud removal in satellite imagery.
727 *Advances in Neural Information Processing Systems*, 2024.
728
729 Yin hao Zhu, Yang Yang, and Taco Cohen. Transformer-based transform coding. In *International*
730 *Conference on Learning Representations (ICLR)*, 2022.
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

A APPENDIX

A.1 LIMITATIONS

While TerraCodec is pretrained on a large and diverse global dataset, the current models are limited to Sentinel-2 L2A imagery. The underlying architectures and training pipeline are sensor-agnostic, but transferring to other sensors requires finetuning or retraining. From a compression perspective, this reflects a common trade-off: sensor-specific codecs typically achieve higher compression efficiency by modeling the statistics and dynamic range of a particular instrument, whereas sensor-agnostic models offer broader applicability. We see cross-sensor and multisensor pretraining as promising future directions, while also recognizing the practical value of sensor-specific codecs in operational EO pipelines. Our implementation, which will be open-sourced, prioritizes research clarity over optimized inference speed; for instance, efficiency techniques such as KV caching are not yet integrated. Incorporating such improvements could further enhance the models’ practicality for real-world deployment.

Our temporal models operate with a two-frame context and are trained and evaluated on four-frame sequences. Using longer sequences would likely reveal additional temporal gains but would also incur substantially higher computational cost. Our p-frame ablation (Fig. 6) highlights the temporal benefits achievable under longer-sequence evaluation.

Finally, FlexTEC establishes a strong flexible-rate baseline, leveraging a latent inference mechanism in which rate control is achieved by dropping latent tokens and inferring the corresponding missing parts through the temporal latent model at decode time. While FlexTEC performs competitively across many settings, there remains room to further narrow the gap to fixed-rate models, particularly under very high compression ratios. Exploring hybrids with other variable-rate methods is a promising future direction.

A.2 USE OF LLMs

We utilized large language models (LLMs) to refine text, improve readability, and assist with coding. All methods, technical content, experimental design, and analyses were developed by the authors.

B TECHNICAL IMPLEMENTATION DETAILS

This section provides additional TerraCodec implementation details not included in the main paper.

B.1 FRAMEWORK AND ENVIRONMENT

All models are implemented in PyTorch, using CompressAI (Bégaint et al., 2020) for core architectures and entropy bottlenecks, extended to support multispectral inputs, temporal samples, and model-belief analyses. The TEC-TT implementations are based on Muckley et al. (2021). Experiments are run on NVIDIA A100 GPUs with mixed precision training.

B.2 IMAGE MODELS (TEC-FP, TEC-ELIC)

TerraCodec-FP follows the factorized prior baseline (Ballé et al., 2017), with g_a and g_s implemented as four strided 5×5 convolutions combined with GDN/IGDN nonlinearities.

TerraCodec-ELIC builds on the uneven channel-group entropy model (He et al., 2022), using the Chandelier reimplemention (Chandelier, 2023). Latents are divided into groups $[16, 16, 32, 64, M-128]$ and coded sequentially using SCCTX (space-channel context). Relative to He et al. (2022), we omit the preview head for efficiency. Despite the stronger context model, training time remains comparable to TEC-FP since the channel-group autoregression is parallelizable across spatial locations and implemented with efficient masked convolutions.

Checkpoint settings. Hyperparameters N and M denote the channel width of encoder/decoder layers and the latent bottleneck size, respectively. For both codecs, N and M are scaled slightly across the trained λ values, as summarized in Table 3. All checkpoints are trained with identical optimization settings (see Sec. 4.1), varying only N , M , and the rate-distortion trade-off coefficient λ .

Table 3: Architectural specifications for TerraCodec-FP and TerraCodec-ELIC. N : main network channels; M : latent bottleneck channels.

Family	Model (λ)	Analysis / Synthesis	Channels (N/M)
TerraCodec-FP	$\lambda = 0.5$		128 / 128
	$\lambda = 2$		128 / 128
	$\lambda = 10$	Conv layers	128 / 128
	$\lambda = 40$	GDN / IGDN	128 / 192
	$\lambda = 200$	Downsampling $\times 16$	192 / 320
	$\lambda = 800$		192 / 320
TerraCodec-ELIC	$\lambda = 0.5$		128 / 192
	$\lambda = 2$	Conv layers	128 / 192
	$\lambda = 10$	Residual blocks	128 / 192
	$\lambda = 40$	Attention	128 / 192
	$\lambda = 200$	Downsampling $\times 16$	320 / 320

Temporal sampling. For image models, we treat the pretraining data as an image dataset by sampling individual timesteps from the SSL4EO-S12 time series. During training, a single temporal index is randomly drawn for every sample in each epoch, such that one epoch covers one quarter of the temporal data. Over the 100 training epochs, this amounts to about 25 full passes through the complete dataset.

B.3 TEMPORAL MODEL (TEC-TT)

We provide additional details on the TEC-TT architecture, including its tokenization strategy and temporal transformer design, following VCT (Mentzer et al., 2022).

The image encoder–decoder follows the TEC-ELIC backbone, composed of residual bottleneck and attention blocks with total downsampling $\times 16$. For our 256×256 input crops, this yields a 16×16 latent grid with $M=192$ channels.

Tokenization. We tokenize the latent image representations using the scheme introduced in VCT. The latent grid $\hat{\mathbf{y}}_i \in \mathbb{R}^{H_\ell \times W_\ell \times d_{\text{lat}}}$ with $H_\ell = W_\ell = 16$ is divided into spatial blocks. The *current* frame is split into non-overlapping 4×4 blocks, each flattened into a sequence of $T=16$ tokens $\{\hat{\mathbf{y}}_{i,b,t}\}_{t=1}^{16}$. The *two past* frames are partitioned into overlapping 8×8 context blocks using reflect-padding so their grids align with the current frame, producing $T=64$ tokens per block.

Concretely, for block index $b \in \{1, \dots, B\}$:

$$\text{current: } \{\hat{\mathbf{y}}_{i,b,t}\}_{t=1}^{16} \in \mathbb{R}^{16 \times d_{\text{model}}}, \quad \text{past: } \{\hat{\mathbf{y}}_{i-1,b,t}\}_{t=1}^{64}, \{\hat{\mathbf{y}}_{i-2,b,t}\}_{t=1}^{64}.$$

Each block forms an independent short token sequence, and all blocks are processed in parallel. All tokens are linearly projected to embeddings of width $d_{\text{tt}}=768$ before entering the temporal transformer.

Temporal transformer stack. The temporal model follows the VCT design and consists of two *separate encoders* E_{sep} (one per past frame), a *joint encoder* E_{joint} that fuses both contexts, and a *masked decoder* that autoregressively models the current block tokens conditioned on the fused context. We adopt the standard VCT specifications for the number of layers, heads, and embedding size in each transformer (see Table 4).

For token bootstrapping and inference, a learned start-of-sequence (SOS) token seeds masked decoding, while early frames without temporal context use a shared bias as a dummy prior (not entropy-coded).

Training. Training follows the standard uniform-noise quantization surrogate (Toderici et al., 2016; Minnen et al., 2018), while inference applies hard quantization and arithmetic coding. We train six TEC-TT variants at different rate–distortion trade-offs, controlled by $\lambda \in$

Table 4: TEC-TT transformer configuration. All blocks use GELU activations, pre-norm layers, and an MLP expansion factor of $4\times$. Dropout is disabled.

Module	# Layers	# Heads	d_{model}	# Tokens / patch
E_{sep} (per past frame)	6	16	768	64
E_{joint} (fusion)	4	16	768	128 [†]
Masked decoder (current)	5	16	768	16 (causal)
Final heads (μ, σ)	$3\times\text{FC}$	–	768	out: $d_C=192$

[†] Token count after concatenation of past-frame representations.

$\{0.4, 5.0, 20.0, 100.0, 300.0, 700.0\}$, where smaller values enforce higher compression and larger values prioritize reconstruction quality.

C LATENT REPACKING AND FLEXTEC

The main paper (Sec. 3.2) introduces *Latent Repacking*, which slices and reorders latent tokens such that early tokens encode global structure and later tokens refine local detail. Here, we provide additional intuition for introducing Latent Repacking and masked training, along with implementation details for FlexTEC.

Scope and notation. FlexTEC is the flexible-rate variant of TEC-TT, obtained by integrating *Latent Repacking* and *masked training*. It uses the same analysis/synthesis (ELIC) backbone and temporal transformer stack as TEC-TT, with latent channel width $d_{\text{lat}}=192$. After tokenizing the current frame into $T=16$ tokens per patch, repacking groups channels into T *channel-slice* tokens (Sec. 3.2). Consequently, FlexTEC exposes *16 discrete quality levels* via the token budget $K \in \{1, \dots, 16\}$, applied consistently across all patches in an image and frames in a sequence. Each token carries $k = d_{\text{lat}}/T = 12$ channels shared across all spatial positions. For all rate-distortion (RD) visualizations in this paper, we report curves for budgets $K = \{1, 2, 3, 4, 5, 6, 7, 8, 12, 16\}$, spanning from the lowest-rate ($K=1$) to the highest-quality ($K=16$) operating points.

Implementation differences vs. TEC-TT. FlexTEC is architecturally identical to TEC-TT except for: (i) the permutation that repacks tokens (Sec. 3.2); (ii) token masking with a learned mask token $m \in \mathbb{R}^{d_{\text{lat}}}$ used during training to replace dropped tokens; and (iii) the masked-rate objective with budget sampling. All other layers, dimensions, and hyperparameters remain unchanged.

FlexTEC is trained with the *same* hyperparameters as TEC-TT, but for 400k steps (vs. 300k for TEC-TT) to account for the task’s added complexity. A single checkpoint trained at $\lambda=800$ (slightly higher than the TEC-TT maximum of 700) is used to cover the full bitrate range under masked training. Empirically, the T/K scaling in the rate term shifts the effective operating point toward lower rates for the same λ , motivating this increase.

Objective and inference. With mask M , the rate is computed on unmasked tokens and, to prevent information collapse into the earliest tokens, upweighted by T/K :

$$\mathcal{L} = \frac{T}{K} R(M) + \lambda D.$$

At test time we pick a budget K (one of 16 levels), transmit only the first K tokens, and *fill* dropped tokens with the transformer’s predicted means μ before decoding. This yields graceful quality-rate scaling with a single checkpoint.

Masking for variable-rate robustness. Repacking latents *alone* is insufficient: a model trained only with full-token inputs learns to rely on *all* tokens and collapses when some are dropped. We therefore train with *masked budgets*: sample $K \in \{1, \dots, T\}$, replace the last $T-K$ tokens by a learned mask vector $m \in \mathbb{R}^{d_{\text{lat}}}$, and compute the rate only on unmasked tokens $R(M)$ (as defined in Eq. 3). For stability we use *teacher forcing*: masking applies only to the *current* frame’s tokens for the rate term, while the temporal encoder always consumes the *real* quantized past latents $(\hat{y}_{i-2}, \hat{y}_{i-1})$. Inspired by Bachmann et al. (2025), budgets are drawn from a categorical distribution biased toward larger values, $\Pr(K=k) \propto k$ (i.e., the multiset $\{1, 2, 2, \dots, T, \dots, T\}$), which trains

the model frequently near high-rate operation while still exposing it to low-budget regimes. Masked tokens are replaced by a learnable per-channel vector $m \in \mathbb{R}^{d_{lat}}$ ($d_{lat}=192$), initialized uniformly in $[-1, 1]$, and shared across positions. This provides a stable placeholder during training while allowing the image decoder to learn how to interpret missing content.

While we keep the number of tokens K fixed within each sequence, it could also be varied across time steps. One approach is to predict K per sample, allowing the model to allocate tokens dynamically based on content complexity. We tested this by predicting K from the joint latent representation z_{joint} with a simple MLP, but found no improvement at inference, as K appeared to be uncorrelated with perceptual complexity. We thus leave adaptive token rates as a future extension of Latent Repacking.

Inference filling. At inference, dropped tokens are *not transmitted*. By default, we fill them with the transformer’s predicted means, i.e., $\hat{y}_{i,b,t} \leftarrow \mu_{i,b,t}$ for frame i , latent block b , and token t , which leverages the learned temporal prior to improve reconstruction quality at a given bitrate. As a lighter alternative (reduced compute), we can instead substitute the learned mask vector m for all dropped tokens.

D EVALUATION DETAILS AND BASELINE METHODS

Baseline quality settings. We evaluate classical codecs across the following quality grids:

Table 5: Quality settings per codec used in RD evaluation.

Codec	Quality settings
JPEG (Pillow)	0, 1, 5, 15, 25, 35, 45, 55, 65, 75, 85, 95
JPEG2000 (Glymur)	0, 2, 5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 120, 150, 170, 200
WebP (Pillow)	0, 1, 5, 15, 25, 35, 45, 55, 65, 75, 85, 95
HEVC/x265 (FFmpeg)	5, 10, 15, 20, 25, 30, 35, 40, 45, 50

Implementations and bit depth. JPEG and WebP are executed via Pillow, supporting only 8-bit input. JPEG2000 is run per band using Glymur/OpenJPEG, which allows for 8- or 16-bit quantization and lossy compression via target ratios. HEVC encoding is performed with FFmpeg and x265, using raw 12-bit monochrome input (pixel format `yuv400p12le`), `CRF= Q`, `-preset medium`, and `-tune psnr`. Bitstream sizes are measured directly from encoded outputs to compute rate. All codecs operate *per band* (grayscale), and we report bits-per-pixel-band-frame (bppbf) by aggregating bitstream sizes across all bands and frames.

Rate metric (bppbf). We report rate as *bits per pixel–band–frame* (bppbf):

$$\text{bppbf} = \frac{\text{total bits}}{H \cdot W \cdot C \cdot T},$$

where $H \times W$ is the spatial resolution, C the number of spectral bands, and T the number of frames. Unlike standard bpp (suited to 3-channel RGB), bppbf normalizes across arbitrary channel counts and sequence lengths, enabling fair comparisons for multispectral time series.

E QUANTITATIVE RESULTS

Across codecs, we report RD curves under multiple distortion metrics and normalizations, then isolate temporal effects (context window, P-frames) and flexible-rate behavior (masking ablation, amortization). This section complements the main text with analyses that clarify how metric choices and temporal conditioning impact conclusions.

E.1 PSNR RANGE SENSITIVITY

In Fig. 11, we plot RD curves for all codecs using PSNR-per-band under different normalization ranges: PSNR 65k (full 16-bit), PSNR 10k (typical Sentinel-2 reflectance 0–10000), and PSNR auto

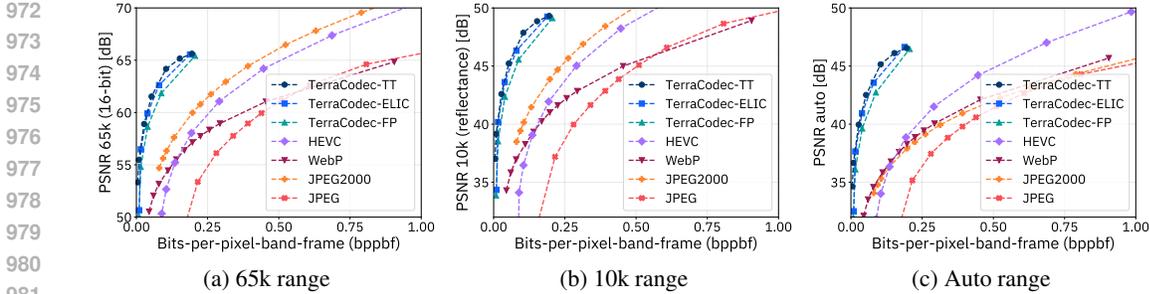


Figure 11: Comparison of the PSNR-per-band metric on different value ranges: 65k covering the full 16-bit range, 10k representing the typical 0–10000 reflectance range of Sentinel-2 data, and auto mode using min–max values of each band.

Table 6: Effect of inference context c on TEC-TT (trained with 2-frame context, $\lambda=5$). We report bits-per-pixel-band-frame (bppbf, \downarrow) and PSNR-per-band (65k, \uparrow). Context $c \in \{0, 1, 1+1, 2\}$ denotes the number of conditioned frames. The P-frames row evaluates only the last two frames (full context) of each 4-frame sequence. Percent changes are relative to $c=0$.

Setting	Context	bppbf \downarrow	PSNR \uparrow
No context (image codec)	0	0.03274	58.522
1 previous frame	1	0.02830 (−13.6%)	58.761 (+0.24dB)
1 previous frame (repeated)	1+1	0.02785 (−14.9%)	58.838 (+0.32dB)
2 previous frames (all frames)	2	0.02722 (−16.9%)	58.902 (+0.38dB)
P-frames only (full context)	2	0.02536 (−22.6%)	59.085 (+0.56dB)

(per-band min–max). We find that PSNR is sensitive to this choice. TerraCodec models (TEC-FP, TEC-ELIC, TEC-TT) remain comparatively stable across ranges, whereas the ranking of classical codecs shifts: JPEG2000 performs best under the PSNR 65k and PSNR 10k ranges, but in the *auto* setting at higher bitrates it is overtaken by WebP and x265.

E.2 TEMPORAL CONDITIONING EFFECTS

We evaluate how temporal context influences fixed-rate TEC-TT models at inference. To isolate genuine temporal gains from the overhead of early bootstrap frames, we vary the number of available past frames and additionally analyze P-frame performance.

Table 6 extends the main paper’s analysis of temporal conditioning. We additionally report results for ($c=1+1$) context, obtained by repeating the same past frame. This configuration yields only a marginal gain over using a single distinct frame (1.6% rate reduction), confirming that improvements stem from meaningful *temporal* information rather than simply longer input sequences. Restricting evaluation to P-frames (full context under $c=2$) further tightens the rate to 0.02536 bppbf at similar PSNR—an additional 6.8% reduction compared to all-frame results including bootstrap frames (0.02722) and 22.6% compared to no context. This quantifies the amortization effect discussed in the main paper and explains why four-frame sequences may understate the full temporal advantage.

Figure 12 shows corresponding RD curves evaluated on *P-frames only*—the last two frames in each four-frame sequence, consistent with TEC-TT’s training context of two past frames. For non-temporal codecs, this selection simply aligns the evaluation set with TEC-TT. The performance gap between TEC-TT and image-only codecs (TEC-FP, TEC-ELIC) widens under this evaluation, reflecting the amortized cost of the initial bootstrap frames. When temporal conditioning is disabled and TEC-TT is run in “image mode,” its performance closely follows TEC-ELIC, consistent with their shared analysis/synthesis backbone.

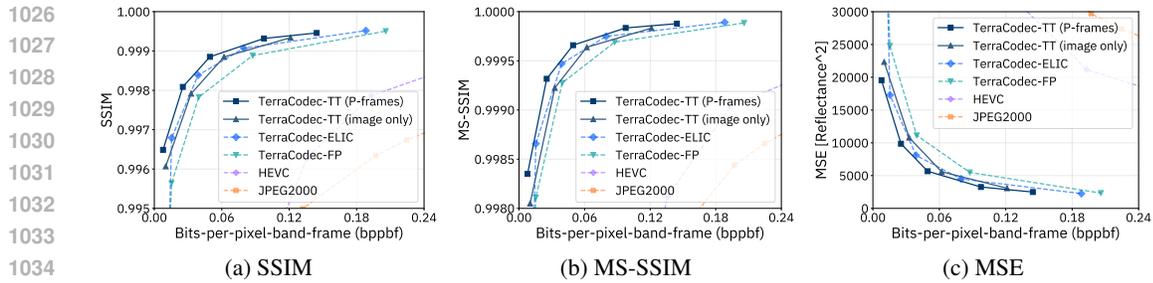
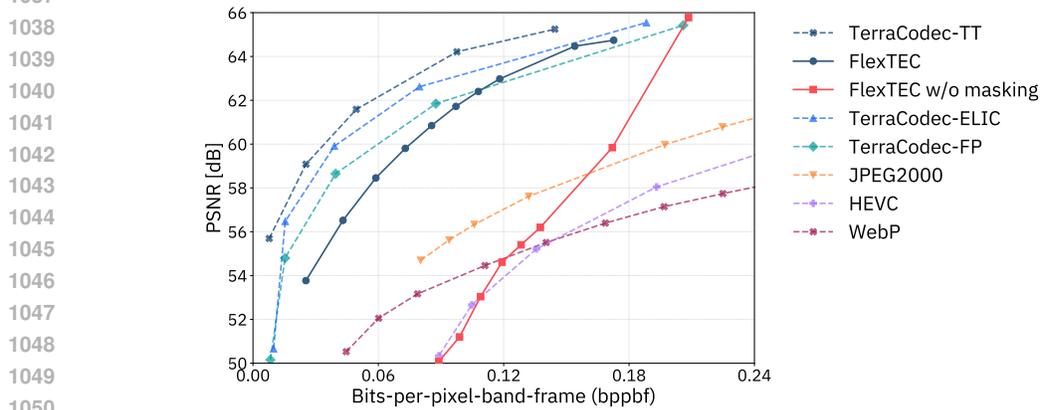


Figure 12: RD curves with additional metrics on the P-frame evaluation set.

Figure 13: Masking ablation for FlexTEC (PSNR-per-band, 65k). We compare FlexTEC (Latent Repacking *with* masking) to a variant trained *without* masking under the same backbone and training setup. FlexTEC curves use token budgets $K = \{1, 2, 3, 4, 5, 6, 7, 8, 12, 16\}$.

E.3 FLEXIBLE-RATE BEHAVIOR

We analyze how masked training impacts FlexTEC’s variable-rate performance. We first ablate masking to verify its necessity, and then compare FlexTEC on P-frames versus all frames to quantify amortization effects.

Fig. 13 shows the effect of masking in flexible-rate training by comparing FlexTEC (Latent Repacking with masking) to an variant trained without masking. The latter deteriorates sharply when tokens are dropped at test time—its RD curve is unstable and substantially below FlexTEC—whereas FlexTEC degrades smoothly and remains roughly parallel to fixed-rate baselines. This confirms that masking is essential for stable variable-rate performance.

Fig. 14 compares FlexTEC on P-frames (last two frames, full context under $c=2$) versus all frames. The gap is notably larger than for fixed-rate TEC-TT, particularly at low rates. We hypothesize two compounding effects: (i) latent repacking with masked training encourages FlexTEC to concentrate scene-wide, high-utility content into the earliest tokens of *bootstrap* frames, lowering their cost relative to fixed-rate models; and (ii) for fully conditioned P-frames, the same curriculum distributes information more evenly across tokens, so truncation retains most of the essentials. Together, these effects yield stronger amortization when excluding bootstrap frames, with the benefit most pronounced in the high-compression regime.

F QUALITATIVE EXAMPLES

This section complements the main paper with additional visual examples. We first compare reconstructions across all codecs on representative SSL4EO-S12 samples (Sec. F.1). We then examine *model beliefs* and forecasting behavior (Sec. F.2) for TEC-TT and FlexTEC. We assess how dropping tokens impacts TEC-TT and the importance of token masking for Latent Repacking. We also

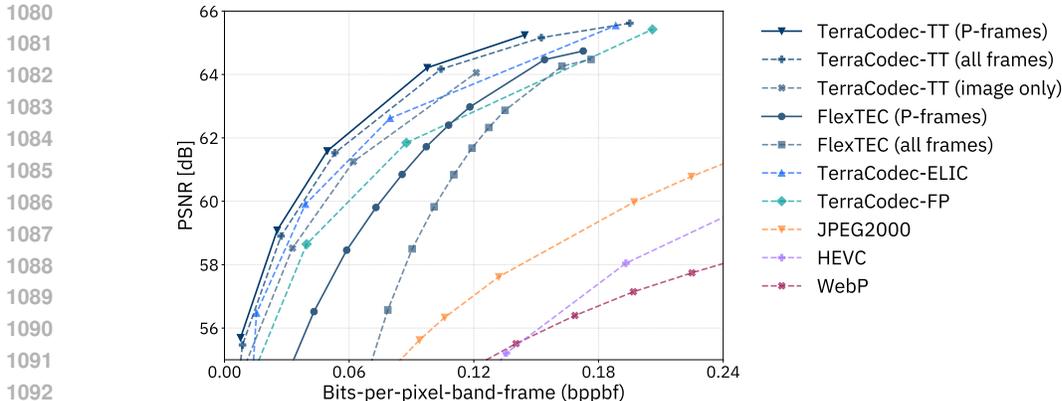


Figure 14: FlexTEC RD on P-frames vs. all frames (PSNR-per-band, 65k). The efficiency gains are much more pronounced for the flexible-rate model than for our fixed-rate TEC-TT models. FlexTEC curves use token budgets $K = \{1, 2, 3, 4, 5, 6, 7, 8, 12, 16\}$.

show how TEC-TT forecasts the next frame from past context, contrasting mean predictions with stochastic samples.

F.1 GENERAL RECONSTRUCTIONS

We compare TerraCodec-TT (TEC-TT), TerraCodec-FP (TEC-FP), and classical codecs (JPEG2000, WebP) on SSL4EO-S12 v1.1 validation samples at matched rate ≈ 0.20 bppbf. Each row in Fig. F.1) shows the original image and reconstructions from each codec, annotated with the average bppbf and PSNR-per-band (65k clipped) across the sequence.

F.2 MODEL BELIEFS AND FORECASTING

We discuss the effect of token budget on different TT model versions and show the TEC-TT forecasts.

F.2.1 TOKEN BUDGET COMPARISON

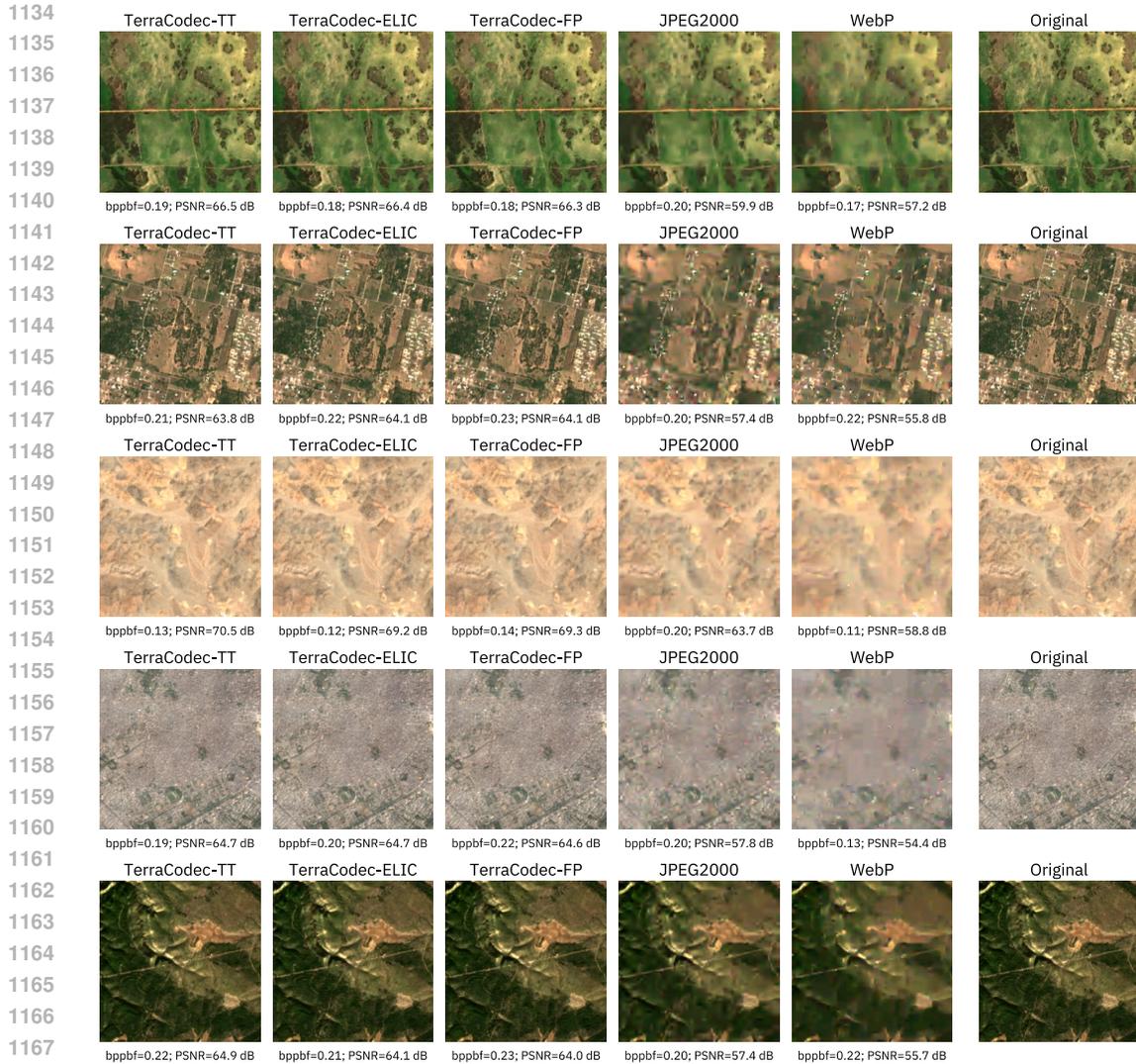
We visualize token budget effects in Fig. 16 using two example sequences under an aggressive token limit, illustrating how models behave when later tokens are dropped. We compare TEC-TT, TEC-TT with Latent Repacking but no masking, and FlexTEC (with both). FlexTEC degrades smoothly and preserves scene-wide structure, whereas TEC-TT exhibits patch erasure and banding; the Latent Repacking w/o masking variant lies in between, confirming that masking with dynamic rate scaling is essential for stable variable-rate performance.

F.2.2 FORECASTS FROM PAST CONTEXT

We probe TEC-TT’s *model beliefs* by predicting the current frame from past context only, using either the predicted mean or samples from the distribution (Fig. 17). The μ -forecast reliably captures large-scale structure, while sampling from the full prior (mean and variance) expresses context-aware uncertainty, primarily reflecting cloud variability. This illustrates the learned distribution rather than a single point estimate. The conservative single-point forecast (μ -forecast) produces a clear-sky prediction for the next frame, enabling TEC-TT to perform cloud removal as evaluated in AllClear.

G ADDITIONAL DETAILS ON CLOUD INPAINTING

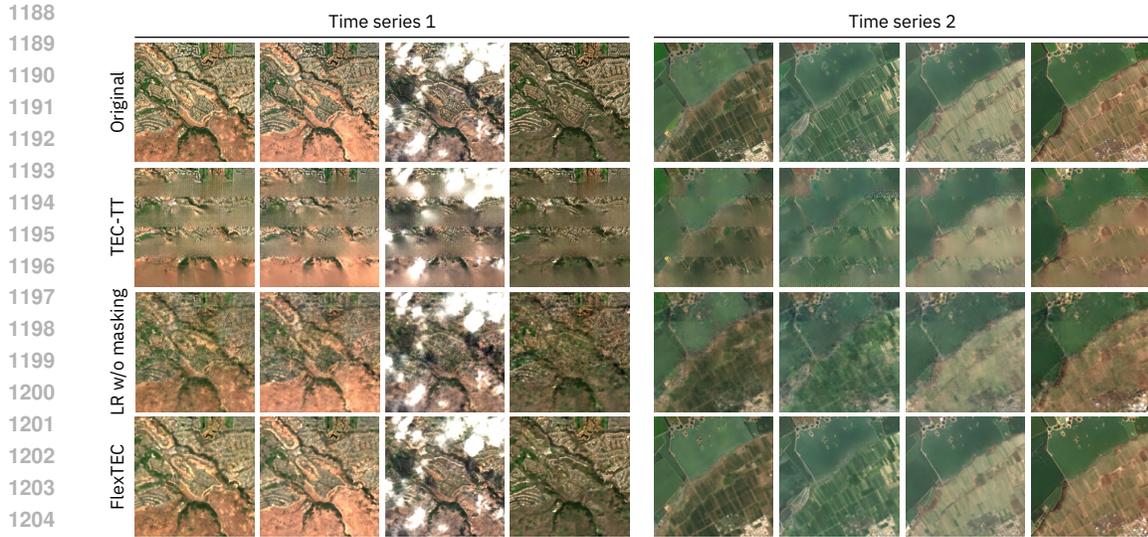
We provide extended results for the AllClear cloud inpainting benchmark (Zhou et al., 2024), complementing Sec. 4.3 and Table 1. TEC-TT is applied without task-specific fine-tuning, leveraging only its latent temporal predictions.

Figure 15: Reconstructions at ≈ 0.20 bppbf on SSL4EO-S12 v1.1.

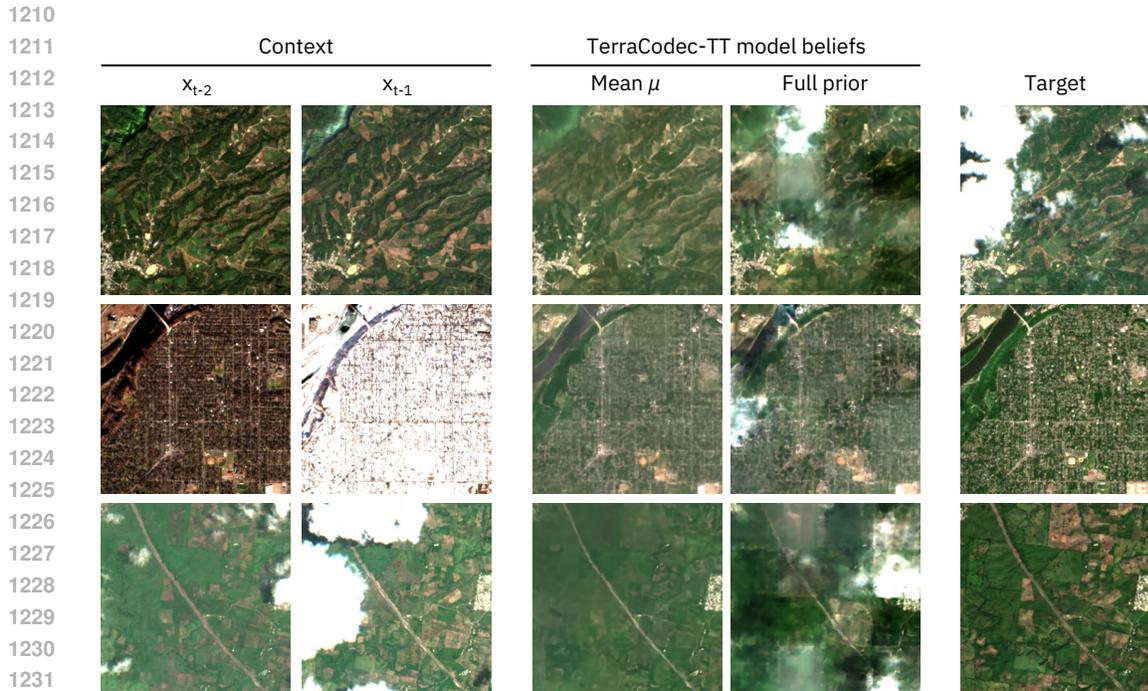
Experimental setup. We follow the AllClear evaluation protocol, reporting metrics across all spectral bands (not per-band). We compare against heuristic baselines (LeastCloudy, Mosaicing) and zero-shot neural methods (CTGAN, DiffCR, PMAA, U-TILISE, UnCRtainTS). LeastCloudy selects the input image with the lowest cloud+shadow coverage, while Mosaicing fills each pixel by copying a single clear value, averaging if multiple are clear, or using 0.5 if none are clear. On AllClear, these heuristics rank among the top three zero-shot methods, outperforming most neural baselines without fine-tuning.

Besides reporting metrics on the full test set, we also evaluate subsets stratified by cloudiness. Difficulty thresholds are defined from the distribution of average cloud cover across the three input frames. Specifically, the 90th percentile (top 10%) corresponds to 0.99 average cloud cover, the 80th percentile (top 20%) to 0.78, and the 50th percentile (top 50%) to 0.49. For each sample, cloud cover is computed as the mean fraction of cloudy pixels across the three timestamps. Cloud masks are generated using the *s2cloudless* algorithm in Google Earth Engine and provided with the dataset.

We use TEC-TT’s prior mean prediction (μ) as a clear-sky estimate of the next frame, capturing large-scale structure while down-weighting transient noise such as clouds (App. F.2.2). Building on this, we adapt TEC-TT for cloud removal by predicting the third input frame from the two previous ones and applying *cloud-aware decoding*: clear regions retain their original tokens, while cloudy



1206 Figure 16: Effect of filling dropped tokens ($K=5$) with model prior predictions at inference for
 1207 different TEC-TT variants. Vanilla TEC-TT exhibits spatial holes and banding when tokens are
 1208 removed. Adding Latent Repacking without masking improves quality but leaves uneven detail,
 1209 while FlexTEC preserves global layout and reduces artifacts.



1233 Figure 17: TEC-TT forecasts using only past context ($K = 0$). Model beliefs are visualized via the
 1234 mean prediction (μ) and full prior sampling from $\mathcal{N}(\mu, \sigma^2)$. The μ -forecast reconstructs coherent
 1235 large-scale structure, while sampling reveals plausible variations (e.g., clouds, surface texture) that
 1236 reflect the model’s uncertainty.

1238 regions are replaced with the transformer’s predicted tokens. Cloud and cloud-shadow masks are
 1239 available and are downsampled to latent resolution using average pooling.

1240 To provide the clean temporal context, we apply a *context reordering* heuristic that duplicates the
 1241 least-cloudy inputs when few clean frames are available. We evaluate 16 variants spanning mask

type, mask threshold (0.0 vs. 0.5), context reordering (on vs. off), and decoding policy: *Interleave*, which predicts only cloudy tokens, and *Propagate*, which predicts from the first cloudy token onward, plus a *pure forecasting* baseline that replaces all tokens with predictions. The best setting uses cloud+shadow masks at threshold 0.0, average-pool downscaling, context reordering, and interleave decoding; all cloud-aware variants outperform pure forecasting, which tends to follow seasonal drift rather than reconstruct the target. Intuitively, conditioning only cloudy regions on predictions while preserving clear tokens reduces seasonal bias and allows TEC–TT to focus its temporal prior on reconstructing the true clear-sky target. Throughout the paper (Sec. 5.3, App. G.1), we report results using this best configuration.

Although our introduced codecs are trained on L2A reflectance data, the AllClear benchmark is defined on L1C inputs. To ensure a fair zero-shot evaluation, we therefore pretrained a TEC–TT model on the SSL4EO-S12 v1.1 L1C data modality using a high $\lambda = 700$ to focus on reconstruction quality. This model uses the same architecture, hyperparameters, and training procedure as the corresponding L2A variant, and is applied without any task-specific fine-tuning on AllClear.

G.1 QUANTITATIVE RESULTS

Table 7: Test SSIM (\uparrow), RMSE (\downarrow), and MAE (\downarrow) on AllClear across difficulty subsets (by average cloudiness). Following the benchmark, the metrics are computed across all bands, not per-band.

Model	10% (hardest)			20%			50%			100% (all)		
	SSIM	RMSE	MAE	SSIM	RMSE	MAE	SSIM	RMSE	MAE	SSIM	RMSE	MAE
<i>Baseline heuristics</i>												
LeastCloudy	0.444	0.348	0.317	0.537	0.279	0.247	0.766	0.135	0.114	0.863	0.078	0.065
Mosaicing	0.107	0.162	0.136	0.183	0.162	0.131	0.558	0.101	0.075	0.755	0.062	0.045
<i>Pre-trained models (zero-shot setting)</i>												
CTGAN	0.765	0.084	0.072	0.794	0.068	0.056	0.822	<u>0.056</u>	0.044	0.840	0.052	0.041
DiffCR	0.716	0.075	0.063	0.739	0.071	0.061	0.758	0.068	0.059	0.744	0.068	0.060
PMAA	0.746	0.071	0.060	0.758	0.076	0.066	0.770	0.080	0.071	0.768	0.086	0.078
U-TILISE	0.546	0.254	0.226	0.598	0.211	0.185	0.693	0.153	0.134	0.807	0.097	0.083
UnCRtainTS	<u>0.813</u>	0.061	0.046	<u>0.826</u>	0.063	0.049	<u>0.865</u>	0.057	<u>0.044</u>	<u>0.898</u>	<u>0.050</u>	<u>0.039</u>
TerraCodec-TT	0.814	<u>0.064</u>	<u>0.050</u>	0.830	<u>0.065</u>	<u>0.050</u>	0.887	0.045	0.034	0.917	0.034	0.025

Table 7 reports SSIM, RMSE, and MAE results for all methods, complementing the PSNR results in the main paper (Table 1). Our zero-shot TEC–TT clearly outperform the heuristics: TEC–TT reaches PSNR ≈ 32.9 dB and SSIM ≈ 0.917 , compared to LeastCloudy (30.61 dB / 0.863) and Mosaicing (29.82 dB / 0.755). Relative to the strongest prior zero-shot neural method on AllClear (UnCRtainTS, 29.01 dB / 0.898 / MAE = 0.039 / RMSE = 0.050), TEC–TT is substantially stronger (32.86 dB / 0.917 / MAE = 0.025 / RMSE = 0.034).

To provide further insight into the stratified evaluation, Figure 18 compares performance vs. cloudiness. While the heuristic baselines (Mosaicing, LeastCloudy) are competitive on less cloudy images, the figures clearly show how they struggle on the highly cloudy subsets, with performance notably degrading. Zero-shot neural methods are more robust, though their performance also declines as past context becomes increasingly obscured.

G.2 QUALITATIVE RESULTS

Figure 19 presents additional TEC–TT inpainting results on the AllClear test set across varying degrees of cloudiness in the three past input frames. While three context images are shown, TEC–TT uses only the two least cloudy as input. Notably, even under heavily clouded conditions, TEC–TT leverages past context to produce relatively accurate predictions of the target frame.

1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349

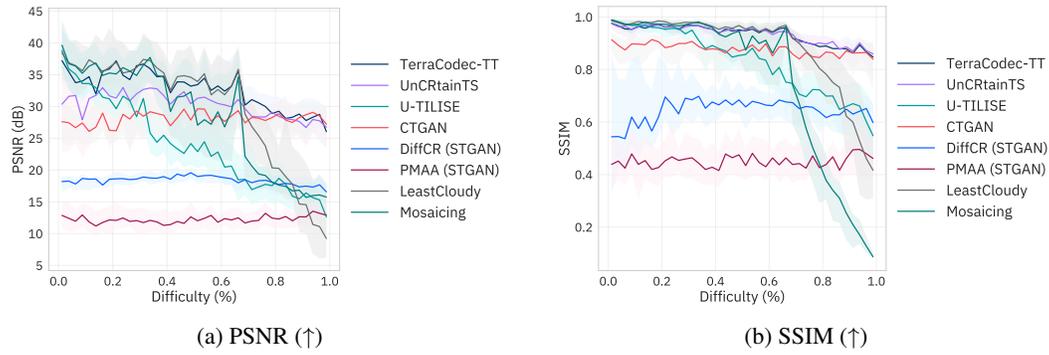


Figure 18: Performance vs. cloudiness on the AllClear test set. Each curve shows the median metric across samples binned by average cloud cover, with shaded ribbons indicating interquartile range (IQR).

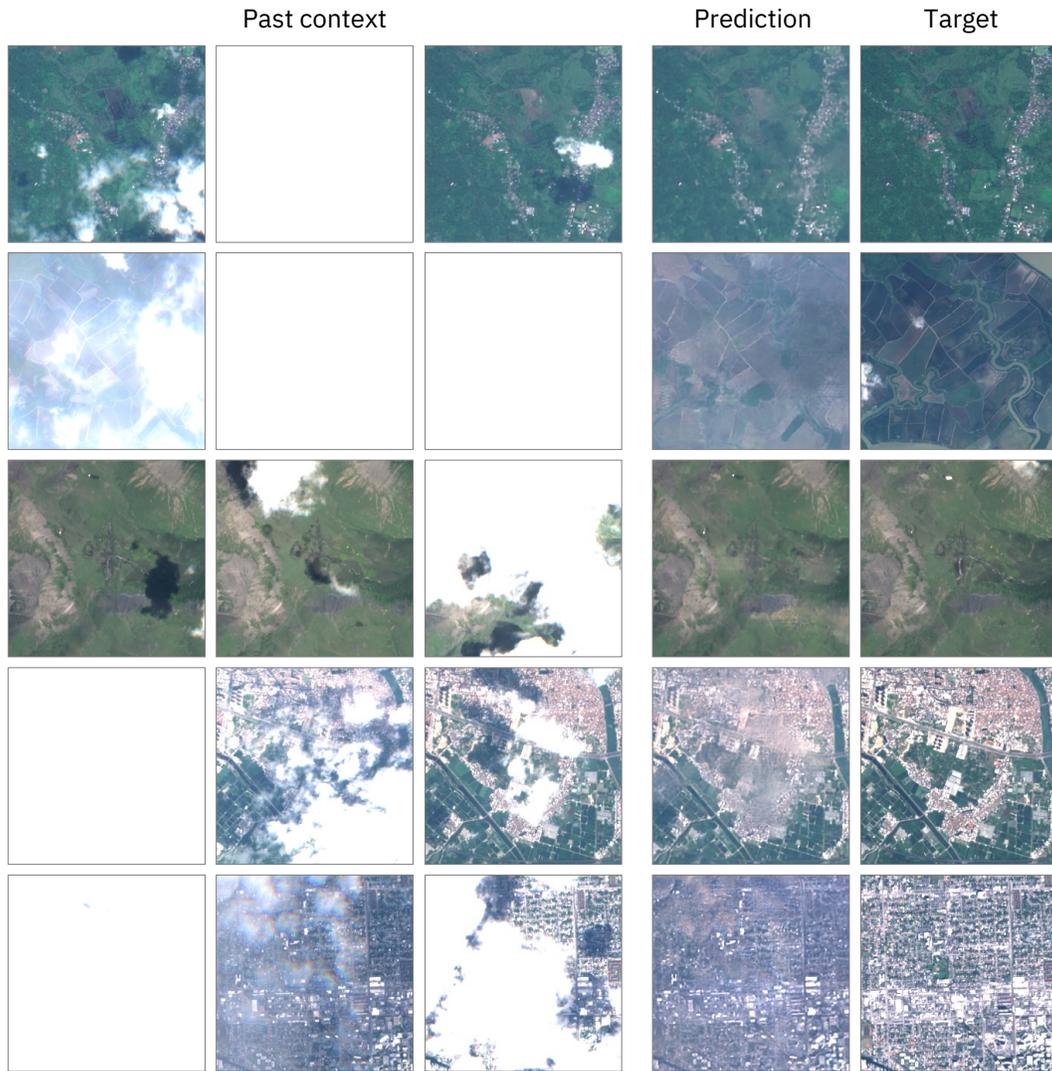


Figure 19: Cloud inpainting examples with TEC-TT on the AllClear benchmark. Reflectance values are clipped to 0–2000, which causes clouds to appear saturated in the visualizations.