

Article

Residual-Distribution-Aware Capacity Analysis for Reversible Data Hiding in Encrypted 3D Medical Volumes with Integer-Valued Voxels

Yuan-Yu Tsai , Ming-Cheng Chuan, Yi-An Huang, Ya-Zhen Liang and Ching-Ta Lu 

Department of Communications Engineering, Feng Chia University, Taichung City 407102, Taiwan; d1322855@o365.fcu.edu.tw (M.-C.C.); d1256309@o365.fcu.edu.tw (Y.-A.H.); d1255830@o365.fcu.edu.tw (Y.-Z.L.)

* Correspondence: yuanytsai@fcu.edu.tw (Y.-Y.T.); chingtl@fcu.edu.tw (C.-T.L.)

Abstract

Reversible data hiding in encrypted media enables auxiliary information to be embedded into protected content while allowing exact recovery of the processed data. Although this technique has been extensively investigated for images and surface-based 3D models, its application to 3D medical volumes remains insufficiently explored. This study presents a residual-distribution-aware capacity analysis for reversible data hiding in encrypted 3D medical volumes with integer-valued voxels. The current framework is restricted to finite, numerically integer-valued voxel samples stored in single-precision format, which can be mapped losslessly to a nonnegative integer domain without quantization. Rather than treating prediction-error minimization as the sole objective, we analyze how residual occupancy across power-of-two bit-length intervals governs leading-zero-count-based embedding capacity. Because the leading-zero count changes only when a residual crosses an interval boundary, reducing the prediction error within the same interval does not necessarily increase embedding capacity. Experiments on nine categories of the Medical Segmentation Decathlon show that pure embedding capacity is jointly determined by representation-derived redundancy, residual-dependent redundancy, and auxiliary information overhead. The benefit of inter-slice prediction is dataset-dependent rather than universal, while homogeneous zero-valued regions can contribute substantially to favorable residual distributions. End-to-end validation at 99% of the available pure embedding capacity achieved error-free secret-data extraction and exact recovery of the processed voxel arrays at both the numerical and bit levels. These results provide a capacity-oriented interpretation of encrypted volumetric reversible data hiding based on residual distribution rather than prediction-error magnitude alone.



Academic Editor:
Aryya Gangopadhyay

Received: 31 July 2026
Revised: 26 August 2026
Accepted: 1 September 2026
Published: 2 September 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: reversible data hiding; encrypted medical volumes; residual distribution; leading-zero count; capacity analysis; integer-valued voxels

1. Introduction

Volumetric data represent physical or anatomical structures in three-dimensional space using regular voxel grids. They are widely used in medical imaging, scientific visualization, industrial inspection, and biomedical analysis [1–3]. Medical volumetric data, including computed tomography, magnetic resonance imaging, and other three-dimensional clinical scans, often contain sensitive patient-related information and therefore require strict confidentiality protection during storage, transmission, and collaborative

processing. Encryption is an effective mechanism for preventing unauthorized access. However, once the data are encrypted, operations such as annotation, authentication, metadata insertion, and auxiliary information embedding become difficult to perform without first decrypting the content. This limitation reduces the flexibility of encrypted medical-data management, particularly in cloud storage, telemedicine, and third-party processing scenarios.

Reversible data hiding (RDH) in encrypted media [4–10] provides a practical solution by allowing additional information to be embedded into protected content while enabling the receiver to extract the hidden data and recover the original content without loss. This property is particularly important for medical data [11–14], where irreversible distortion may affect clinical interpretation or data integrity. Existing RDH methods for encrypted media have been extensively investigated for two-dimensional images. These methods commonly reserve embedding room before encryption by exploiting spatial redundancy, prediction errors, high-bit-plane correlations, or compressible auxiliary information. Other studies further improve embedding performance through adaptive coding, block rearrangement, or secret-sharing-based mechanisms.

RDH has also been extended to surface-based three-dimensional representations [15–24], including polygonal meshes and point clouds. In these media, embedding capacity is typically related to vertex coordinate similarity, vertex index continuity, connectivity relationships, geometric traversal, spatial subdivision, or path planning. Although these studies demonstrate the potential of encrypted-domain embedding for 3D data, their redundancy sources differ fundamentally from those of volumetric data. Surface-based models are represented by irregular vertices, faces, or point sets, whereas medical volumetric data are organized as dense and regular voxel grids. Therefore, capacity interpretations developed for meshes or point clouds cannot be directly applied to medical volumes.

Compared with encrypted images and surface-based 3D models, reversible data hiding in encrypted 3D medical volumes remains insufficiently explored. Extending a conventional reserving-room-before-encryption (RRBE) framework to volumetric data is technically feasible, but such an extension alone does not explain the factors that determine embedding capacity. Existing RDH methods frequently emphasize prediction accuracy and implicitly assume that reducing prediction errors will improve embedding performance. However, this assumption is not always valid for leading-zero-count-based (LZC-based) embedding mechanisms. In such mechanisms, the number of embeddable bits depends on the effective bit length of each prediction residual rather than on its absolute magnitude alone. The present framework is restricted to finite, numerically integer-valued voxel samples stored in single-precision format that can be mapped losslessly to a nonnegative integer domain without quantization; genuinely non-integer floating-point volumes are outside the scope of the current implementation.

This study presents a residual-distribution-aware capacity analysis for reversible data hiding in encrypted 3D medical volumes with integer-valued voxels. Given a fixed bit-depth representation, the leading-zero count of a residual changes only when its magnitude crosses a power-of-two boundary. Consequently, reducing a residual within the same bit-length interval does not necessarily create additional embedding room. By contrast, a small reduction that shifts a residual into a lower bit-length interval may directly increase the number of embeddable bits. This property indicates that residual interval occupancy provides a more direct interpretation of embedding capacity than prediction-error magnitude alone. The capacity analysis further distinguishes representation-derived redundancy from residual-dependent redundancy and explicitly accounts for auxiliary information overhead, thereby avoiding attribution of the final embedding capacity solely to prediction performance.

Medical volumes may contain multiple sources of redundancy, including homogeneous regions, in-slice spatial structure, and inter-slice correlation. Their relative contributions, however, are dataset-dependent. In particular, inter-slice information does not universally improve pure embedding capacity, and large homogeneous zero-valued regions can substantially affect residual concentration. Therefore, favorable residual distributions should not be attributed solely to three-dimensional spatial correlation. To examine these effects under a common embedding mechanism, this study adopts an implemented RRBE-based framework and evaluates representative causal voxel predictors using consistent residual labeling, auxiliary coding, and capacity calculation procedures. The objective is to clarify how residual distributions and representation characteristics jointly affect pure embedding capacity rather than to optimize prediction error alone.

The main contributions of this study are summarized as follows.

- An RRBE-based analytical and experimental framework is established for encrypted 3D medical volumes with finite, numerically integer-valued voxels stored in single-precision format, with exact voxel recovery verified at both the numerical and bit levels.
- A residual-distribution-aware capacity interpretation is developed by relating LZC-based embedding capacity to residual occupancy across power-of-two intervals and by separating representation-derived, prediction-derived, and auxiliary information contributions.
- Experiments on nine medical-volume categories show that smaller prediction errors do not necessarily yield larger pure embedding capacity and that the benefit of inter-slice information is dataset-dependent.

The remainder of this paper is organized as follows. Section 2 reviews related work on reversible data hiding in encrypted media. Section 3 introduces the adopted RRBE-based analytical framework and explains the relationship between residual distributions and embedding capacity. Section 4 presents the experimental results on medical volumetric datasets. Finally, Section 5 concludes the paper and discusses future research directions.

2. Related Work

RRBE is an important paradigm for reversible data hiding in encrypted media. Under this paradigm, embeddable redundancy is generated from the original content before encryption, and the reserved space is subsequently used for data embedding in the encrypted domain. Although RRBE-based methods share this general principle, their redundancy sources vary across media types. This section reviews RRBE-based methods for encrypted natural and medical images, encrypted 3D models and point clouds, as well as residual-distribution-aware capacity analysis.

2.1. RDH in Encrypted Images and Encrypted Medical Images

Reversible data hiding in encrypted images (RDHEIs) enables additional information to be embedded into protected image content while allowing the receiver to extract the hidden data and reconstruct the original image without loss. Existing RDHEI methods are commonly categorized as vacating-room-after-encryption (VRAE) and RRBE approaches. VRAE methods generate embedding space after image encryption and therefore often rely on block-based encryption or coding mechanisms that preserve partial redundancy. In contrast, RRBE methods reserve room before encryption through additional preprocessing and generally provide greater embedding capacity.

Recent studies have improved RRBE-based embedding performance through more effective prediction and coding strategies. Huang et al. [4] employed hybrid prediction and bidirectional flag encoding to exploit prediction errors concentrated around zero. Ren et al. [5] preserved local redundancy within encrypted blocks and applied adap-

tive Huffman coding to pixel differences. Zhang et al. [6] combined robust regression, deep-feature-guided grouping, and intra-group transformation to improve prediction and entropy-coding efficiency. Although these methods adopt different mechanisms, they share a common objective: converting predictable or statistically concentrated image characteristics into compact auxiliary representations and additional embedding room.

RDHEI has also been extended toward multiparty and distributed-security scenarios. Fang et al. [7] combined secret encrypted image sharing with adaptive coding to support secure image distribution and reversible embedding. Yu et al. [8] introduced threshold secret sharing and multivariate linear equations to enable lossless recovery from multiple image shares. Xiong et al. [9] developed a universal framework based on additive secret sharing and secure multiparty computation for scenarios involving multiple data hiders. Sun et al. [10] proposed dual-phase embedding with multi-key threshold decryption to support authentication before and after decryption. These studies demonstrate that RDHEI can support not only capacity-oriented embedding, but also distributed storage, collaborative processing, authentication, and access control.

Medical images impose stricter requirements because irreversible distortion may affect clinical interpretation, diagnostic reliability, or data integrity. In picture archiving and communication systems (PACSs), telemedicine, and cloud-based medical storage, encrypted-domain embedding can associate protected images with patient-related information, authentication codes, timestamps, or management tags. Compared with ordinary natural images, medical images often exhibit distinctive characteristics, including higher bit depths, large homogeneous regions, background areas, and diagnostically important regions of interest (ROIs). These properties motivate the development of medical-image-oriented RDHEI methods. Kong et al. [11] divided DICOM images into regions of interest and regions of non-interest, embedded data into selected encrypted regions, and used auxiliary information for recovery and tamper detection. Dzwonkowski and Czaplewski [12] proposed a separable RRBE method for encrypted DICOM images based on binary decomposition and sorted binary pixel sequences. Their method exploited selected MSB planes and extended run-length encoding to support high payload and separable recovery. Martyniak and Dzwonkowski [13] further combined fixed and block-wise pixel prediction by decomposing 16-bit DICOM images into two 8-bit components and applying tailored compression strategies. Yang et al. [14] proposed an encrypted medical-image RDH framework based on Huffman coding and count-encryption to improve compression performance and reduce key-management complexity in PACS environments. These studies show that encrypted medical-image RDH must consider not only embedding capacity, but also diagnostic-region protection, recoverability, and secure data management.

2.2. RDH in Encrypted 3D Models and Encrypted Point Clouds

Reversible data hiding has also been investigated for encrypted 3D models. Compared with images, 3D models contain geometric and structural information whose redundancy depends on the underlying representation. For polygonal meshes, existing methods commonly exploit vertex coordinates, spatial relationships, connectivity information, or vertex indices. For point clouds, which do not contain explicit mesh connectivity, spatial subdivision and traversal-path construction are particularly important.

Early methods for encrypted polygonal meshes mainly embedded data into vertex coordinates. Jiang et al. [15] modified the least significant bits of encrypted vertex coordinates, but the achievable capacity was limited and error-free extraction could not always be guaranteed. Subsequent RRBE-based methods improved embedding performance by reserving room before encryption through coordinate transformation, adaptive grouping, integer mapping, and multi-MSB prediction. Representative strategies include parity-based

vertex partitioning and adaptive substitution [16], octree-based spatial grouping [17], and the integration of adaptive grouping and secret sharing [18]. More recent methods have improved the utilization of available vertices through dynamic prediction and spatial organization. Wang et al. [19] introduced a dynamic mechanism that allows vertices to participate more flexibly in both embedding and prediction. Traversal-based and path-based approaches [20–22] arrange spatially related vertices into predictable sequences to increase coordinate similarity and reduce auxiliary information overhead.

Point-cloud methods rely on different redundancy sources because point clouds do not contain explicit face adjacency. For point clouds, spatial organization and path construction can be used to improve local predictability. KD-tree-based path planning [23], for example, connects spatially close points to form more predictable reference sequences. Collectively, these studies demonstrate that the effectiveness of encrypted-domain RDH depends strongly on the structural characteristics of the target media. For irregular 3D data, spatial organization is essential for improving predictability and reversible embedding capacity.

Index-domain embedding [24] provides an alternative strategy for polygonal meshes. Because each polygon is represented by a sequence of vertex indices, suitable face ordering, vertex reindexing, and traversal processing can improve the predictability of neighboring index sequences. The resulting structural redundancy can be used to reserve embedding room without modifying vertex coordinates. Therefore, index-domain methods preserve geometric appearance while exploiting connectivity-related information.

2.3. Residual-Distribution-Based Capacity Analysis

In RRBE-based methods, embedding room is commonly generated by exploiting predictable or compressible structures before encryption. Existing studies improve embedding performance by reducing prediction errors, increasing common high-order bits, selecting suitable reference samples, reorganizing correlated elements, or reducing auxiliary information overhead. Although improved prediction accuracy frequently produces more compact residual representations, prediction-error minimization and embedding-capacity maximization are not mathematically equivalent when the embedding rule is determined by discrete bit-level properties.

In an LZC-based mechanism [24], the number of available positions depends on the effective bit length of the prediction-residual magnitude. Residual magnitudes belonging to the same power-of-two interval have the same effective bit length and therefore provide the same number of leading-zero positions. For example, under an eight-bit representation, reducing a residual from seven, represented as $(00000111)_2$, to six, represented as $(00000110)_2$, does not change its effective bit length. In contrast, reducing a residual from eight, represented as $(00001000)_2$, to seven reduces its effective bit length from four to three bits and creates one additional embeddable position.

Consequently, conventional prediction-error measures such as MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error) do not fully characterize LZC-based embedding potential. Capacity is more directly determined by the occupancy of residuals across bit-length intervals. In addition, pure embedding capacity depends on the coding cost of the corresponding capacity labels. Two predictors may therefore produce similar total capacities but different pure capacities because their label distributions have different entropies and auxiliary information sizes.

Despite these advances, residual distribution is usually treated as an intermediate outcome rather than as an independent factor governing embedding capacity. Existing studies often focus on prediction accuracy, auxiliary information size, embeddable-element ratio, or final payload. Moreover, when a fixed representation width exceeds the effective numerical bit depth of the data, part of the available capacity may originate directly from

the representation itself rather than from prediction. Therefore, the final pure embedding capacity should be interpreted by distinguishing representation-derived redundancy, residual-dependent redundancy, and auxiliary information overhead.

Medical volumes may contain several sources of redundancy, including homogeneous regions, in-slice spatial structure, and inter-slice correlation, but their contributions to embedding capacity are not necessarily uniform across datasets. Accordingly, this study analyzes how residual interval occupancy, representation-derived redundancy, and auxiliary information overhead jointly determine pure embedding capacity under a consistent RRBE-based framework, rather than attributing favorable capacity primarily to prediction-error reduction or three-dimensional spatial correlation.

3. Residual-Distribution-Aware Capacity Analysis Framework

This section presents the complete RRBE-based framework used for residual-distribution-aware capacity analysis and end-to-end reversible data hiding in encrypted 3D medical volumes. As illustrated in Figure 1, the framework consists of three parties: the data owner, the data hider, and the receiver. At the data-owner side, the input voxel values are losslessly mapped to a nonnegative integer domain and processed by causal voxel prediction. The resulting signed residuals are represented by their magnitudes and signs, from which residual labels, embedding capacities, and the auxiliary bitstream are constructed. The residual-magnitude words are then encrypted, and the auxiliary information is embedded into predefined predictable positions of the encrypted representation.

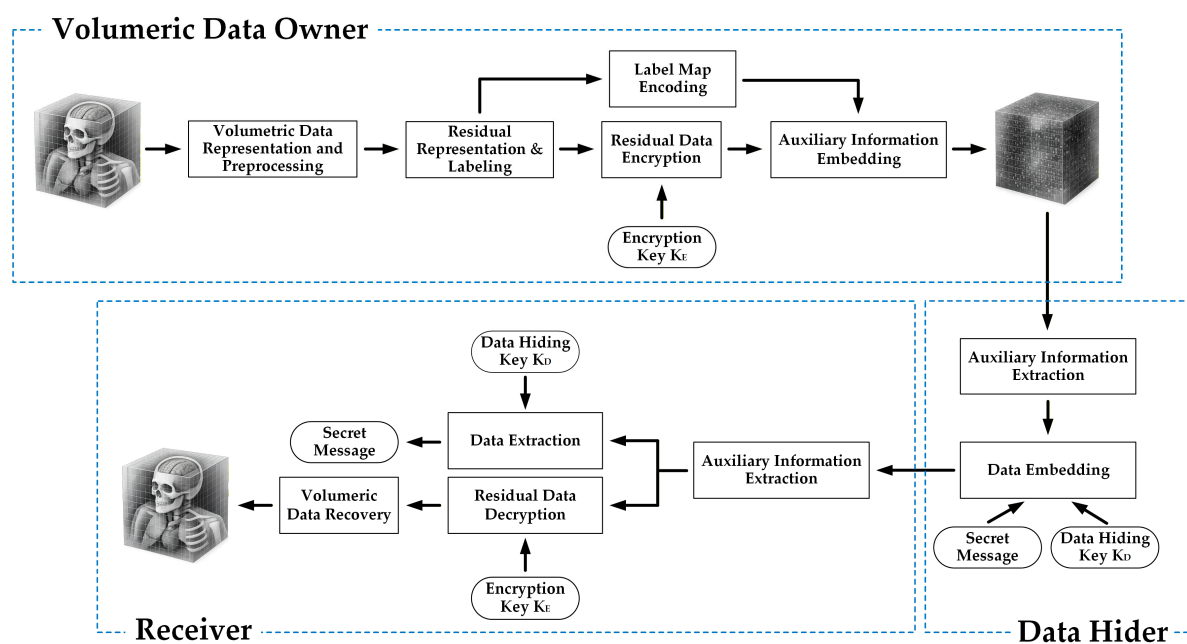


Figure 1. Flowchart of the adopted RRBE-based framework for encrypted 3D medical volumetric data.

The data hider embeds a key-protected secret message into the remaining available positions without accessing the original voxel values or the content-encryption key. At the receiver side, the bootstrap information is extracted first to recover the parameters required for auxiliary decoding. The residual labels are then decoded to determine the remaining embedding positions, after which the secret message can be extracted independently of content recovery. For voxel recovery, the retained ciphertext bits are decrypted, while the overwritten predictable residual-magnitude bits and residual signs are reconstructed from the decoded labels. Causal inverse prediction followed by inverse numerical mapping then reconstructs the processed voxel array exactly. The following subsections describe

the volumetric-data representation and preprocessing, causal prediction and residual labeling, capacity construction, encryption and auxiliary-data organization, embedding and extraction procedures, and exact recovery process.

3.1. Volumetric Data Representation and Preprocessing

A 3D medical volume is represented as a regular voxel grid of dimensions $W \times H \times D$, where W , H , and D denote the width, height, and depth, respectively. The total number of voxels is $N = W \times H \times D$, and the voxel values are denoted by $v_{i,j,k}$, where $1 \leq i \leq W$, $1 \leq j \leq H$, and $1 \leq k \leq D$. When explicit spatial coordinates are not required, the same voxels are equivalently indexed by $p \in \{1, 2, \dots, N\}$ according to a fixed linear scan order. For multidimensional volumetric files containing more than one 3D component, each component is treated as a separate processing unit and undergoes the same preprocessing, prediction, labeling, and embedding procedures.

The current framework is restricted to finite, numerically integer-valued voxel samples stored in single-precision format. Before processing, all voxel values are verified to be finite and to satisfy exact integer-valuedness, defined by equality to their rounded values without applying any numerical tolerance. No rounding or quantization is performed to make non-integer data admissible. Volumes containing non-finite or genuinely non-integer samples are therefore outside the scope of the current framework.

To obtain a nonnegative integer domain without altering the voxel values irreversibly, let $v_{\min} = \min_{i,j,k} v_{i,j,k}$ and $\delta = \min(v_{\min}, 0)$. Each voxel is then mapped as $v'_{i,j,k} = v_{i,j,k} - \delta$. Thus, $\delta = 0$ when all voxel values are nonnegative, whereas a negative minimum value is used as the offset when necessary. Because δ is retained as auxiliary recovery information, the original numerical domain can be restored exactly by $v_{i,j,k} = v'_{i,j,k} + \delta$.

Let $M = \max_{i,j,k} v'_{i,j,k}$. The effective bit depth of the processed 3D component is defined as $b_e = \max(1, \lceil \log_2(M + 1) \rceil)$. The value b_e therefore represents the minimum number of bits required to represent the normalized voxel values of that component. For multidimensional NIfTI files, both δ and b_e are determined separately for each 3D component.

In the implemented framework, the fixed representation width is $b = 32$, whereas b_e is determined by the numerical range of each component. The difference $b - b_e$ forms the representation-derived redundancy that provides the basic embedding capacity. Residual-dependent capacity is subsequently generated from causal prediction and is analyzed separately in the following subsections.

3.2. Residual Representation and Labeling

Three causal voxel predictors, Causal13, MED, and WCP, are evaluated under the same RRBE-based framework. They are used as representative residual generators rather than newly designed prediction models. Prediction is performed on the normalized voxel values $v'_{i,j,k}$, and only causal neighboring voxels are used so that the same prediction process can be reproduced during recovery.

The first predictor is the Causal13 mean predictor. For each voxel $v'_{i,j,k}$, the predictor uses the available 3×3 neighboring voxels from the previous slice $k - 1$ and four causal in-slice neighbors from the current slice. The four in-slice neighbors are $v'_{i,j-1,k}$, $v'_{i-1,j,k}$, $v'_{i-1,j-1,k}$, and $v'_{i-1,j+1,k}$. Let $\Omega_{i,j,k}^{13}$ denote the set of available causal neighbors. The predicted value is computed using Equation (1). For boundary voxels, unavailable neighbors are excluded from the mean. If no causal neighbor is available, the predicted value is set to

zero. This predictor provides a simple reference method that jointly considers in-slice and inter-slice spatial correlations.

$$\hat{v}_{i,j,k}^{13} = \text{round} \left(\frac{1}{|\Omega_{i,j,k}^{13}|} \sum_{(\alpha,\beta,\gamma) \in \Omega_{i,j,k}^{13}} v'_{\alpha,\beta,\gamma} \right) \quad (1)$$

The second predictor is the MED predictor. For voxel $v'_{i,j,k}$, the left, upper, upper-left, and previous-slice voxels are denoted as $A = v'_{i,j-1,k}$, $B = v'_{i-1,j,k}$, $C = v'_{i-1,j-1,k}$, and $D = v'_{i,j,k-1}$, respectively. If any of these neighbors is unavailable because of boundary conditions, its value is set to zero. A linear estimate is first computed as $L = A + B + D - C$. Let $m_{\min} = \min(A, B, D)$ and $m_{\max} = \max(A, B, D)$. The predicted value is then determined by the switching rule shown in Equation (2). This predictor is included as a MED-type reference method for examining whether improved prediction accuracy necessarily leads to greater embedding capacity in volumetric data.

$$\hat{v}_{i,j,k}^{\text{MED}} = \begin{cases} m_{\min}, & \text{if } L \geq m_{\max}, \\ m_{\max}, & \text{if } L \leq m_{\min}, \\ L, & \text{otherwise.} \end{cases} \quad (2)$$

The third predictor is the weighted causal predictor (WCP), which separately considers inter-slice and in-slice causal information. The inter-slice neighborhood consists of the available 3×3 voxels on the previous slice $k - 1$, whereas the in-slice causal neighborhood consists of $v'_{i,j-1,k}$, $v'_{i-1,j,k}$, $v'_{i-1,j-1,k}$, and $v'_{i-1,j+1,k}$. Let $\mu_{i,j,k}^z$ and $\mu_{i,j,k}^{xy}$ denote the mean values of the available inter-slice and in-slice neighbors, respectively. If no available voxel exists in a neighborhood, the corresponding mean is set to zero. The predicted value is computed using Equation (3). The main experiments use the predefined configuration $w_z : w_{xy} = 1 : 2$, which assigns greater weight to the in-slice causal estimate while retaining previous-slice information. This setting is used as a representative configuration rather than an optimized weighting scheme, and its sensitivity is examined experimentally in Section 4.

$$\hat{v}_{i,j,k}^{\text{WCP}} = \text{round} \left(\frac{w_z \mu_{i,j,k}^z + w_{xy} \mu_{i,j,k}^{xy}}{w_z + w_{xy}} \right) \quad (3)$$

For the selected predictor, the signed prediction residual is defined as $d_{i,j,k} = v'_{i,j,k} - \hat{v}_{i,j,k}$. Let $\ell_{i,j,k}$ denote the effective bit length of the residual magnitude shown in Equation (4). Under the fixed $b = 32$ representation, the corresponding leading-zero count is $z_{i,j,k} = b - \ell_{i,j,k}$. The residual sign is represented by $s_{i,j,k} = 1$ when $d_{i,j,k} < 0$ and $s_{i,j,k} = 0$ otherwise. The residual label λ is then defined using Equation (5). The label jointly represents the leading-zero structure used to determine embeddable positions and the sign information required for exact residual reconstruction. Different predictors produce different residual distributions and, consequently, different label distributions and embedding capacities.

$$\ell_{i,j,k} = \begin{cases} \left\lfloor \log_2 |d_{i,j,k}| \right\rfloor + 1, & |d_{i,j,k}| > 0, \\ 0, & |d_{i,j,k}| = 0. \end{cases} \quad (4)$$

$$\lambda_{i,j,k} = 2 \times z_{i,j,k} + s_{i,j,k} \quad (5)$$

3.3. Residual-Distribution-Aware Embedding Capacity Analysis

The embedding capacity of each voxel is determined by the leading-zero structure of its residual magnitude. Using the leading-zero count $z_{i,j,k}$ defined in Section 3.2, the

total embeddable capacity of voxel $v'_{i,j,k}$ is $c_{i,j,k} = \min(z_{i,j,k} + 1, b)$, where $b = 32$. For a nonzero residual, the embeddable positions consist of its leading-zero positions and the first nonzero bit. For a zero residual, all b positions are available.

The total capacity contains two distinct sources of redundancy. Because the effective voxel bit depth is b_e , the upper $b - b_e$ positions are available independently of the prediction residual. The basic embedding capacity (BEC) is therefore $C_{BEC} = N(b - b_e)$. This term represents representation-derived redundancy. The remaining capacity is determined by the residual distribution and is defined as the residual-dependent capacity (RDC), $C_{RDC} = \sum_{i,j,k} c_{i,j,k} - C_{BEC}$. Accordingly, the total embedding capacity is $C_{TEC} = C_{BEC} + C_{RDC} = \sum_{i,j,k} c_{i,j,k}$.

The auxiliary information required for recovery occupies C_{AUX} bits. In the implemented $b = 32$ configuration, the complete auxiliary stream is embedded within the BEC positions for every feasible component. The representation-derived payload capacity and prediction-derived payload capacity are therefore $C_{PBEC} = C_{BEC} - C_{AUX}$ and $C_{PRC} = C_{RDC}$, and the resulting pure embedding capacity is $C_{PEC} = C_{PBEC} + C_{PRC} = C_{TEC} - C_{AUX}$. For comparison across volumes of different sizes, these quantities are normalized by the number of voxels, $A_{BEC} = C_{BEC}/N$, $A_{RDC} = C_{RDC}/N$, $A_{TEC} = C_{TEC}/N$, $A_{AUX} = C_{AUX}/N$, $A_{PBEC} = C_{PBEC}/N$, $A_{PRC} = C_{PRC}/N$, and $A_{PEC} = C_{PEC}/N$.

The residual-dependent component is governed by the occupancy of residuals across power-of-two intervals. For a nonzero residual satisfying $2^{\phi-1} \leq |d_{i,j,k}| \leq 2^\phi - 1$, its effective bit length remains ϕ , and therefore its leading-zero count and embedding capacity remain unchanged throughout the interval. Consequently, reducing the residual magnitude does not increase capacity unless the reduction crosses a power-of-two boundary. For example, reducing a residual magnitude from 15 to 9 decreases the prediction error, but both values remain in the interval $8 \leq |d| \leq 15$ and require four effective bits. Their embedding capacities are therefore identical. By contrast, reducing the magnitude from 8 to 7 moves the residual from the four-bit interval to the three-bit interval, increasing the leading-zero count by one and providing one additional embeddable position. Thus, prediction-error minimization alone does not determine embedding capacity; the distribution of residuals across bit-length intervals is the relevant factor for the prediction-derived component.

3.4. Encryption and Key Usage

After residual generation and labeling, each signed residual $d_{i,j,k}$ is separated into its magnitude $|d_{i,j,k}|$ and sign. The residual magnitude is represented as a 32-bit unsigned word, whereas the sign is incorporated into the residual label defined in Section 3.2. Let w_p denote the 32-bit residual-magnitude word at voxel position p . The complete word is encrypted using AES-256 in counter (CTR) mode with the 256-bit content-encryption key K_E , $w_p^E = w_p \oplus Z_p^E$, where Z_p^E denotes the corresponding 32-bit portion of the AES-CTR keystream generated using K_E and IV_E . Thus, encryption is applied to the complete residual-magnitude representation rather than only to the subsequently embeddable bit positions.

A separate 256-bit data hiding key K_D is used to protect the secret message. Before embedding, the secret bitstream B_D is whitened using a separate AES-CTR keystream, $B_D^E = B_D \oplus Z_D$, where Z_D is generated using K_D and a data hiding IV IV_D . In the implemented framework, IV_D is deterministically derived from IV_E by XORing the first byte of IV_E with 0xA5, while the remaining bytes are unchanged. This construction distinguishes the two CTR invocations while allowing the data hiding IV to be reproduced from the stored content-encryption IV. The original secret message can therefore be recovered using K_D without access to K_E .

The content-encryption IV IV_E is included in the bootstrap information required for recovery. In the experiments, a distinct and reproducible IV_E is generated for each file, 3D

component, and predictor combination. In practical deployment, a unique non-repeating content-encryption IV should be generated for each encryption instance, from which IV_D is then derived as described above. Consequently, secret-data extraction requires K_D but not K_E , whereas exact voxel recovery requires K_E but not K_D .

3.5. Embedding Positions and Auxiliary Information Construction

In the implemented framework, all N voxel positions of a feasible 3D component participate in capacity generation. Therefore, the embeddable voxel set is $\mathcal{E} = \mathcal{V} = \{1, 2, \dots, N\}$. Two types of embeddable bit positions are distinguished. The first type consists of the BEC positions arising from the difference between the fixed representation width $b = 32$ and the effective bit depth b_e . Their position set is $\mathcal{B}_{BEC} = \{(p, q) | p \in \mathcal{V}, q = b, b-1, \dots, b_e+1\}$, which contains $N(b-b_e)$ positions. These positions are ordered first by bit plane from b down to b_e+1 , and then by the fixed linear voxel order within each plane. This ordering is independent of the residual labels and can therefore be determined before the complete label map is decoded. The implementation follows this plane-major ordering explicitly.

The second type consists of residual-dependent positions. For voxel p , let c_p denote its total embeddable capacity defined in Section 3.3. The number of residual-dependent positions is $c_p^{res} = \max(0, c_p - (b - b_e))$. These positions are selected from bit plane b_e downward, using the first c_p^{res} predictable positions of the residual word. Residual-dependent positions are ordered by bit plane and then by the same fixed linear voxel order. Consequently, once b_e and the residual labels are known, the complete embedding-position sequence is deterministic. This corresponds directly to the implementation, which derives residual capacity from the decoded labels and scans bit planes from b_e downward.

The auxiliary bitstream is organized as $B_A = B_H \parallel B_T \parallel B_L$, where B_H is a fixed 512-bit bootstrap header, B_T is the serialized Huffman tree, and B_L is the Huffman-coded residual-label stream. All three components are embedded exclusively in BEC positions before secret-data embedding. The fixed bootstrap header has the following layout shown in Equation (6). The 16-bit format identifier is fixed to 0xA5C3 and is used to verify the bootstrap-header format. The normalization offset is stored using its 32-bit IEEE 754 binary32 representation. The header also records the representation parameters, predictor identifier, voxel count, Huffman-tree and label-stream lengths, secret and auxiliary lengths, and the content-encryption IV. These fields provide the information required to locate and decode the remaining auxiliary stream without prior knowledge of the residual labels. The implemented header contains exactly 512 bits.

$$B_H = \text{formatID}_{16} \parallel \text{version}_8 \parallel b_6 \parallel b_{e,6} \parallel \text{predictorID}_2 \parallel \text{symbolSize}_4 \parallel \text{offset}_{32} \parallel N_{64} \parallel \text{treeLen}_{48} \parallel \text{labelLen}_{48} \parallel \text{secretLen}_{48} \parallel \text{auxLen}_{48} \parallel \text{IV}_{128} \parallel \text{reserved}_{54} \quad (6)$$

The residual labels are Huffman coded in the fixed linear voxel order. The Huffman tree is serialized using preorder traversal: each internal node is represented by a 0, whereas each leaf is represented by a 1 followed by the fixed-width binary representation of its label symbol. For $b = 32$, the symbol width is 7 bits. The serialized tree B_T is immediately followed by the encoded label stream B_L .

To resolve the bootstrap dependency, the first 512 positions of the BEC stream are reserved for B_H . Because the BEC stream begins with bit plane 32 in plane-major order, these positions correspond to the most significant bit of the first 512 residual-magnitude words. The receiver can therefore extract B_H without knowing the residual labels. After obtaining b_e , treeLen , labelLen , and the other required parameters from the header, the receiver reads B_T and B_L from the remaining BEC positions and reconstructs the complete residual-label map. Only then are the residual-dependent positions required. The implementation indeed reads the bootstrap directly from bit 32 of the first 512 words.

A component is considered feasible only if $N \geq 512$, $b_e < 32$, $|B_A| \leq C_{BEC}$. If the complete header, serialized Huffman tree, and encoded label stream cannot be accommodated within the BEC positions, the component is rejected as infeasible; auxiliary information is neither truncated nor partially embedded. This condition is enforced explicitly in the implementation.

3.6. Secret-Data Embedding and Extraction

After the auxiliary bitstream B_A has been embedded into the BEC positions by the data owner, the data hider embeds the protected secret stream B_D^E without accessing the original voxel values or the content-encryption key K_E . Let $L_D = |B_D^E|$ denote the secret length and let $C_{BEC}^{rem} = C_{BEC} - |B_A|$ denote the number of unused BEC positions. The first $L_{BEC} = \min(L_D, C_{BEC}^{rem})$ secret bits are embedded into the remaining BEC positions, following the plane-major and linear-voxel ordering defined in Section 3.5. If $L_D > C_{BEC}^{rem}$, the remaining $L_{RDC} = L_D - L_{BEC}$ bits are embedded into the residual-dependent positions, beginning from bit plane b_e and proceeding toward the lower bit planes in the deterministic order defined by the decoded residual labels. Secret embedding is permitted only when $L_D \leq C_{PEC}$. When residual-dependent positions are required, the data hider first reads the public bootstrap information and decodes the Huffman tree and residual-label stream from the BEC positions. The corresponding residual-dependent positions can therefore be determined without access to the original voxel values or the content-encryption key K_E .

At the receiver side, extraction begins with the fixed 512-bit bootstrap header. The header provides b_e , the auxiliary information lengths, voxel count, predictor identifier, normalization offset, and content-encryption IV. The serialized Huffman tree and encoded residual-label stream are then read from the remaining BEC positions, allowing the complete residual-label map to be reconstructed. Because the BEC positions are independent of the residual labels, this auxiliary information can be recovered before any residual-dependent position is accessed.

Once the residual labels are available, the receiver reconstructs the same embedding-position sequence used by the data hider. The protected secret bits are first extracted from the unused BEC positions and, when necessary, from the residual-dependent positions. The resulting stream B_D^E is then de-whitened using the data hiding key K_D and the derived data hiding IV IV_D , $B_D = B_D^E \oplus Z_D$. Therefore, secret-data extraction requires K_D but does not require the content-encryption key K_E . The embedding and extraction orders are deterministic and identical, ensuring separable secret-data extraction from the encrypted volumetric representation.

The implemented framework assumes a semi-trusted data hider that is authorized to perform encrypted-domain embedding but does not possess the content-encryption key K_E . The confidentiality objective is limited to preventing recovery of the exact voxel values without K_E and protecting the secret payload without K_D . The auxiliary stream required for position determination is intentionally accessible to the data hider and is therefore not treated as confidential metadata. Specifically, the bootstrap header reveals representation and processing parameters, including b_e , the normalization offset, predictor identifier, voxel count, stream lengths, and IV_E . In addition, decoding the spatially ordered residual-label stream reveals the leading-zero count z_p and sign s_p of each residual. An exact-zero residual can therefore be identified directly, while for a nonzero residual, the effective bit length $\ell_p = b - z_p$ reveals that its magnitude lies within $2^{\ell_p-1} \leq |d_p| \leq 2^{\ell_p} - 1$, together with its sign. The lower residual-magnitude bits and exact voxel values remain unavailable without K_E ; however, the exposed labels may reveal spatial or statistical characteristics of the volume, such as residual sparsity, local predictability, smoothness, discontinuities, and

sign patterns. Accordingly, the present implementation does not claim confidentiality of residual-derived metadata or protection against inference from this side information.

3.7. Residual Restoration and Exact Voxel Recovery

After the auxiliary information has been decoded, the residual labels provide the leading-zero count and sign of every residual. Therefore, the bit positions overwritten by the auxiliary and secret streams do not require separate displaced-bit storage. Specifically, embedding is restricted to positions whose original residual-magnitude values are determined by the decoded labels: the leading-zero positions and, for a nonzero residual, its first nonzero bit. The lower unpredictable bits are never overwritten and remain available in encrypted form.

Let w_p^M denote the marked encrypted 32-bit word at voxel position p , and let Z_p^E denote the AES-CTR keystream word generated from K_E and the recovered content-encryption IV. The receiver first computes $\tilde{w}_p = w_p^M \oplus Z_p^E$. Because some predictable positions of w_p^M may contain embedded data, $\tilde{w}_p$ is not directly treated as the complete residual magnitude. Instead, only its lower unpredictable bits are retained. From the decoded label λ_p , the leading-zero count and effective bit length are obtained as $z_p = \lfloor \lambda_p / 2 \rfloor$, $\ell_p = b - z_p$. For a nonzero residual, the first nonzero bit is located at position ℓ_p , whereas the lower $\ell_p - 1$ bits are the unpredictable bits that were never modified during embedding. The residual magnitude is therefore reconstructed as in Equation (7). This operation explicitly restores the overwritten leading zeros and the first nonzero bit from the decoded label while preserving the decrypted lower bits. The residual sign is recovered from $s_p = \lambda_p \bmod 2$, and the signed residual is reconstructed as in Equation (8).

$$|d_p^R| = \begin{cases} 0, & \ell_p = 0, \\ 2^{\ell_p-1} + (\tilde{w}_p \bmod 2^{\ell_p-1}), & \ell_p > 0. \end{cases} \quad (7)$$

$$d_p^R = \begin{cases} -|d_p^R|, & s_p = 1, \\ |d_p^R|, & s_p = 0. \end{cases} \quad (8)$$

The predictor identifier extracted from the bootstrap header then selects the corresponding Causal13, MED, or WCP reconstruction rule. Because all three predictors are causal, normalized voxel values can be recovered sequentially using only previously reconstructed neighbors. For each voxel, $v_{i,j,k}^R = \hat{v}_{i,j,k}^R + d_{i,j,k}^R$, where $\hat{v}_{i,j,k}^R$ is generated using the same predictor and boundary rule employed during residual generation.

Finally, the normalization offset recovered from the bootstrap header is reapplied: $v_{i,j,k}^R = v_{i,j,k}^R + \delta$. The recovered voxel values are then returned to the original single-precision representation. Recovery accuracy is evaluated at both the numerical-value and IEEE 754 binary32 word levels in Section 4. The recovery target is the processed voxel array; reconstruction of the complete NIfTI container, including metadata and file serialization, is outside the embedding domain.

3.8. Overall Embedding, Extraction, and Recovery Procedures

Figures 2–4 summarize the complete RRBE workflow. These algorithms consolidate the operations defined in Sections 3.1–3.7 and make the bootstrap dependency and key-separable extraction and recovery procedures explicit.

```

01  Input
02      3D voxel component  $V$ , selected predictor  $\rho$ , content-encryption key  $K_E$ ,
03      data hiding key  $K_D$ , content-encryption IV  $IV_E$ , secret bitstream  $B_D$ 
04  Output
05      Marked encrypted residual words  $W^M$ 
06  Verify that all voxel values in  $V$  are finite and numerically integer-valued
07  Compute  $\delta = \min(v_{min}, 0)$ , normalize the voxel values as  $v'_{i,j,k} = v_{i,j,k} - \delta$ , and
    determine the effective bit depth  $b_e$ 
08  Apply predictor  $\rho$  to  $V'$  and compute the signed residuals  $d_{i,j,k} = v'_{i,j,k} - \hat{v}_{i,j,k}$ 
09  For each residual, compute its magnitude  $|d_{i,j,k}|$ , sign  $s_{i,j,k}$ , leading-zero count
     $z_{i,j,k}$ , residual label  $\lambda_{i,j,k} = 2z_{i,j,k} + s_{i,j,k}$ , and embeddable capacity  $c_{i,j,k}$ 
10  Huffman encode the residual-label sequence to obtain the serialized tree  $B_T$  and
    encoded label stream  $B_L$ 
11  Construct the fixed 512-bit bootstrap header  $B_H$ , which records  $b_e$ , predictor
    identifier,  $\delta$ ,  $N$ , tree length, label-stream length, secret length, auxiliary length,
     $IV_E$ , and the remaining format fields
12  Form the auxiliary bitstream  $B_A = B_H \parallel B_T \parallel B_L$ 
13  Verify the feasibility conditions  $N \geq 512$ ,  $b_e < 32$ ,  $|B_A| \leq C_{BEC}$ , and  $|B_D| \leq C_{PEC}$ 
14  Represent each residual magnitude as a 32-bit unsigned word  $w_p$ 
15  Encrypt all residual-magnitude words using AES-256-CTR with  $K_E$  and  $IV_E$ ,
     $w_p^E = w_p \oplus Z_p^E$ 
16  Embed  $B_A$  into the BEC positions of  $W^E$  using the predefined plane-major and
    voxel order
17  Derive  $IV_D$  from  $IV_E$ , and whiten the secret bitstream using the AES-CTR key-
    stream generated from  $K_D$  and  $IV_D$ ,  $B_D^E = B_D \oplus Z_D$ 
18  Embed  $B_D^E$  first into the unused BEC positions
19  If the remaining BEC is insufficient, embed the remaining protected secret bits
    into the residual-dependent positions
20  Return the marked encrypted residual words  $W^M$ 

```

Figure 2. Pseudocode for the encrypted-domain embedding procedure.

```

01  Input
02      Marked encrypted residual words  $W^M$ , data hiding key  $K_D$ 
03  Output
04      Recovered secret bitstream  $B_D$ 
05  Read the most significant bit of the first 512 residual words to obtain the bootstrap
    header  $B_H$ 
06  Parse  $B_H$  to recover  $b_e$ ,  $N$ , the Huffman-tree length, label-stream length, secret
    length, auxiliary length,  $IV_E$ , and the other required parameters
07  Read the serialized Huffman tree  $B_T$  and encoded residual-label stream  $B_L$ 
    from the corresponding BEC positions
08  Deserialize  $B_T$  and Huffman decode  $B_L$  to reconstruct the complete residual-la-
    bel sequence  $\{\lambda_p\}_{p=1}^N$ 
09  Recover the leading-zero count of each residual as  $z_p = \lfloor \lambda_p/2 \rfloor$ , and determine
    the residual-dependent embedding positions
10  Compute the remaining BEC  $C_{BEC}^{rem} = C_{BEC} - |B_A|$ 
11  Extract the first  $L_{BEC} = \min(|B_D^E|, C_{BEC}^{rem})$  protected secret bits from the unused
    BEC positions
12  If additional protected secret bits remain, extract them from the residual-depend-
    ent positions in the same deterministic order used during embedding
13  Concatenate the extracted bits to recover  $B_D^E$ 
14  Derive  $IV_D$  from the recovered  $IV_E$ 
15  De-whiten the protected secret stream,  $B_D = B_D^E \oplus Z_D$ , where  $Z_D$  is generated us-
    ing  $K_D$  and  $IV_D$ 
16  Return  $B_D$ 

```

Figure 3. Pseudocode for the secret-data extraction procedure.

```

01  Input
02      Marked encrypted residual words  $W^M$ , content-encryption key  $K_E$ 
03  Output
04      Recovered single-precision voxel component  $V^R$ 
05  Read and parse the fixed 512-bit bootstrap header  $B_H$ 
06  Recover  $B_T$  and  $B_L$  from the BEC positions, deserialize the Huffman tree, and
    decode the complete residual-label sequence  $\{\lambda_p\}_{p=1}^N$ 
07  Generate the AES-CTR keystream using  $K_E$  and the recovered  $IV_E$ 
08  For each marked encrypted residual word  $w_p^M$ , compute  $\tilde{w}_p = w_p^M \oplus Z_p^E$ 
09  For each voxel position  $p$ , recover  $z_p = \lfloor \lambda_p/2 \rfloor$ ,  $\ell_p = 32 - z_p$ 
10  Reconstruct the residual magnitude using (7)
11  Recover the residual sign as  $s_p = \lambda_p \bmod 2$ 
12  Reconstruct the signed residual  $d_p^R$  using (8)
13  Select the causal predictor specified by the predictor identifier recovered from  $B_H$ 
14  Reconstruct the normalized voxel values sequentially according to the same
    causal scan order  $v_{i,j,k}^R = \hat{v}_{i,j,k}^R + d_{i,j,k}^R$ 
15  Restore the original numerical domain,  $v_{i,j,k}^R = v_{i,j,k}^R + \delta$ 
16  Convert the recovered voxel array to the original single-precision representation
17  Return  $V^R$ 

```

Figure 4. Pseudocode for the exact voxel recovery procedure.

4. Experimental Results and Discussion

The residual-distribution-aware capacity analysis was evaluated using the Medical Segmentation Decathlon (MSD) dataset [25]. The dataset contains ten categories of three-dimensional or multidimensional medical volumes covering diverse anatomical structures, spatial resolutions, volume sizes, and voxel-domain characteristics. The current framework is restricted to finite, numerically integer-valued voxel samples stored in single-precision format. Accordingly, Task04_Hippocampus is excluded because it contains genuinely non-integer floating-point samples that cannot be mapped losslessly to the adopted integer domain without quantization. The remaining nine categories satisfy the eligibility condition and are used for evaluation.

Across the nine evaluated categories, 1481 NIfTI files yield 2965 separately processed 3D components. For multidimensional NIfTI files, each 3D component is treated as a separate processing unit because the normalization offset and effective bit depth are determined separately for each component. All evaluated voxel arrays are stored and processed as single-precision values, and their finite and exact integer-valued characteristics are verified directly from the voxel samples rather than inferred from metadata. The fixed representation width $b = 32$ is used throughout the main experiments.

The experimental evaluation proceeds in several stages. First, the evaluation metrics and experimental settings are defined. The three causal predictors are then compared in terms of prediction error, residual interval occupancy, auxiliary information overhead, and embedding capacity. The analysis subsequently separates representation-derived and prediction-derived payload capacities, examines sensitivity to the assumed representation width, and evaluates the contribution of inter-slice information using controlled WCP configurations. Additional analyses investigate zero-valued regions, predictor-specific boundary handling, and file-level variability. Finally, end-to-end embedding, secret-data extraction, exact voxel recovery, computational performance, and cross-media capacity characteristics are evaluated.

4.1. Experimental Settings and Evaluation Metrics

All experiments were implemented in MATLAB R2025b Update 2 and executed under 64-bit Windows 11 on a desktop computer equipped with an AMD Ryzen 7 9800X3D

processor and 64 GB of RAM. NIfTI data were loaded using the MATLAB NIfTI interface, while prediction, residual analysis, Huffman coding, capacity evaluation, embedding, extraction, and voxel reconstruction were implemented in MATLAB. AES-256-CTR operations were performed through the Java Cryptography Architecture using AES/CTR/NoPadding. Dataset-wide processing used eight process-based MATLAB workers with dynamic file-level scheduling.

The main experiments use the fixed representation width $b = 32$. Causal13, MED, and WCP are evaluated using the same preprocessing, residual-labeling, auxiliary-coding, and capacity-calculation procedures. The predefined WCP configuration uses $w_z : w_{xy} = 1 : 2$; alternative weight configurations are examined separately in the sensitivity analysis. For descriptive analyses, measurements are retained for each of the 2965 processed 3D components. For file-level summaries, variability analysis, confidence intervals, and paired predictor comparisons, multiple components originating from the same multidimensional NIfTI file are aggregated into a single file-level observation, yielding 1481 observations per predictor. This avoids treating components from the same file as statistically independent samples.

Prediction accuracy is evaluated using the MAE and RMSE of the signed prediction residuals, as defined in Equations (9) and (10), respectively. These metrics quantify prediction-error magnitude but do not directly indicate whether residuals cross the power-of-two boundaries governing LZC-based embedding capacity. To characterize residual distributions more directly, the magnitude domain is partitioned into eight intervals: $\mathcal{I} = \{\{0\}, \{1\}, [2, 3], [4, 7], [8, 15], [16, 31], [32, 63], [64, \infty)\}$. For residual interval $\mathcal{I}_r$, its occupancy is defined in Equation (11), where $\mathbb{I}(\cdot)$ is the indicator function. The corresponding occupancies are denoted by $P_0, P_1, P_{2-3}, P_{4-7}, P_{8-15}, P_{16-31}, P_{32-63}$, and P_{64+} . These intervals directly reflect changes in effective residual bit length, except that all residual magnitudes of 64 or greater are grouped into the final interval.

$$MAE = \frac{1}{N} \sum_{p=1}^N |d_p| \quad (9)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{p=1}^N d_p^2} \quad (10)$$

$$P_r = \frac{1}{N} \sum_{p=1}^N \mathbb{I}(|d_p| \in \mathcal{I}_r) \quad (11)$$

Embedding performance is evaluated using the capacity components defined in Section 3.3, including A_{BEC} , A_{RDC} , A_{TEC} , A_{AUX} , A_{PBEC} , A_{PRC} , and A_{PEC} , all reported in bits per voxel. In particular, A_{PBEC} and A_{PRC} represent the representation-derived payload capacity and prediction-derived payload capacity, respectively, while A_{PEC} is used as the principal usable-payload metric.

Component-level MAE, RMSE, residual-interval occupancies, and capacity measures are retained for descriptive analyses. Spearman's rank correlation is used descriptively to examine the association between prediction-error metrics and both overall and residual-dependent capacity across the 2965 processed components. Because multiple components may originate from the same NIfTI file, these component-level correlations are not treated as independent-sample significance tests.

For inferential comparisons among predictors, file-level observations are used. Variability is summarized using the mean, standard deviation, median, interquartile range, and 95% bootstrap confidence interval based on 10,000 bootstrap resamples. Paired comparisons of A_{TEC} and A_{PEC} are performed between WCP and Causal13 and between WCP and MED using two-sided Wilcoxon signed-rank tests. Holm adjustment is applied to the resulting p -values, and rank-biserial correlation is reported as the paired effect-size measure.

End-to-end reversibility is evaluated separately using one representative 3D component from each of the nine dataset categories, selected according to the predefined median-volume-size criterion. Each component is tested with Causal13, MED, and WCP, resulting in 27 validation cases. In each case, the secret payload occupies approximately 99% of the available A_{PEC} . Evaluation includes secret-data extraction accuracy, bit-error rate (BER), secret-bit errors, reconstruction RMSE, maximum absolute voxel error, numerical mismatch count, and IEEE 754 binary32 word mismatch count.

For the contextual cross-media comparison, normalized embedding efficiency (NPEE) is defined as Equation (12), where C_{ref} is the source-reported capacity selected for comparison, and b_{ref} is the corresponding representation width expressed using the same sample unit as C_{ref} . For the present framework, $C_{ref} = A_{PEC}$ and $b_{ref} = 32$. Because the referenced studies differ in media type, dataset, embedding mechanism, capacity definition, and auxiliary information accounting, NPEE is used only as a normalized contextual indicator and not as a basis for direct performance ranking.

$$NPEE = \frac{C_{ref}}{b_{ref}} \times 100\% \quad (12)$$

4.2. Dataset and Numerical Characteristics

Table 1 summarizes the structural and numerical characteristics of the nine evaluated MSD categories. The dataset contains 1481 NIfTI files, which yield 2965 separately processed 3D components. Task01_BrainTumour and Task05_Prostate contain multidimensional data and therefore contribute multiple 3D components per file, whereas the remaining categories contain one processed 3D component per file. The evaluated categories also differ substantially in spatial dimensions and original zero-valued voxel ratios. In particular, Task01_BrainTumour contains the largest proportion of original zero-valued voxels at 83.89%, whereas the corresponding ratios for most CT-based categories are close to zero. These differences provide distinct voxel-domain characteristics for examining their influence on residual distributions and embedding capacity.

Table 1. Dataset and numerical characteristics of the evaluated MSD categories.

Class	Files	3D Components	Dim	Avg. Size	Zero-Valued Voxels (%)	Voxel Range	b_e Range	Median b_e
Task01_BrainTumour	484	1936	4	$240 \times 240 \times 155$	83.89%	0–11,737	7–14	11
Task02_Heart	20	20	3	$320 \times 320 \times 114$	33.53%	0–2196	11–12	11
Task03_Liver	131	131	3	$512 \times 512 \times 448$	0.13%	–2048–27,572	11–15	12
Task05_Prostate	32	64	4	$316 \times 316 \times 19$	26.07%	0–3860	11–12	12
Task06_Lung	63	63	3	$512 \times 512 \times 280$	0.11%	–1024–20,349	12–15	12
Task07_Pancreas	281	281	3	$512 \times 512 \times 95$	0.09%	–2048–4009	12–13	12
Task08_HepaticVessel	303	303	3	$512 \times 512 \times 70$	0.09%	–1024–4543	11–13	12
Task09_Spleen	41	41	3	$512 \times 512 \times 89$	0.11%	–1024–3072	12–13	12
Task10_Colon	126	126	3	$512 \times 512 \times 107$	0.09%	–1024–13,438	12–14	12

All evaluated NIfTI files use the single-precision datatype, and the voxel arrays are read as MATLAB single values. Eligibility for the proposed framework is verified directly from the voxel samples rather than inferred from the NIfTI metadata. For each processed 3D component, all voxel values are required to be finite and to satisfy $v = \text{round}(v)$ exactly, without applying a numerical tolerance. Across all 2965 processed components, no NaN or Inf values and no genuinely non-integer voxel samples were detected. The maximum absolute fractional part, $\max|v - \text{round}(v)|$, was therefore zero for all evaluated components. These observations confirm that the nine retained MSD categories satisfy the numerical eligibility condition of the current framework.

The effective bit depth b_e is determined separately for each 3D component after the lossless normalization described in Section 3.1. Across the 2965 components, b_e ranges from 7 to 15 bits. Specifically, 1 component (0.03%) has $b_e = 7$; 10 (0.34%) have $b_e = 8$; 88 (2.97%) have $b_e = 9$; 316 (10.66%) have $b_e = 10$; 892 (30.08%) have $b_e = 11$; 1578 (53.22%) have $b_e = 12$; 61 (2.06%) have $b_e = 13$; 16 (0.54%) have $b_e = 14$; and 3 (0.10%) have $b_e = 15$. Thus, 83.31% of the evaluated components require either 11 or 12 effective bits.

This effective-bit-depth distribution is directly related to the representation-derived embedding capacity. Because the main experiments use the fixed representation width $b = 32$, the basic embedding capacity of the m -th component is $C_{BEC,m} = N_m(32 - b_{e,m})$. Therefore, a one-bit decrease in b_e increases the representation-derived capacity by exactly one bit per voxel. The predominance of $b_e = 11$ and $b_e = 12$ consequently provides substantial representation-derived redundancy before any prediction-dependent capacity is considered. This contribution is separated from the residual-dependent component in the subsequent capacity analysis.

4.3. Residual Distribution and Capacity Analysis

Figures 5–7 present the residual interval occupancy ratios obtained using Causal13, MED, and WCP, respectively. The distributions differ substantially across both predictors and dataset categories. Because each interval corresponds to a range of effective residual bit lengths, moving residuals toward lower intervals can increase the number of predictable leading-zero positions. In contrast, reducing residual magnitude within the same interval does not change the available capacity. The $|d| = 0$ entries in Figures 5–7 are predictor-dependent exact-zero residual occupancies and are distinct from the original zero-valued voxel ratios reported in Table 1.

Class	$ d = 0$	$ d = 1$	$2 \leq d \leq 3$	$4 \leq d \leq 7$	$8 \leq d \leq 15$	$16 \leq d \leq 31$	$32 \leq d \leq 63$	$ d \geq 64$
Task01	83.23%	0.65%	1.24%	2.16%	3.24%	3.71%	2.95%	2.82%
Task02	35.43%	5.20%	6.99%	9.28%	11.91%	13.73%	10.88%	6.58%
Task03	14.73%	5.50%	10.37%	17.01%	20.78%	16.50%	8.60%	6.51%
Task05	22.51%	1.22%	2.27%	4.17%	7.27%	11.32%	14.86%	36.38%
Task06	19.65%	1.78%	3.51%	6.75%	11.73%	16.99%	17.78%	21.81%
Task07	22.51%	4.79%	9.11%	14.80%	17.49%	14.44%	8.94%	7.92%
Task08	22.93%	5.49%	10.26%	15.62%	16.57%	12.64%	8.07%	8.42%
Task09	18.89%	5.42%	10.07%	15.69%	17.88%	14.33%	8.83%	8.89%
Task10	18.87%	5.79%	10.55%	16.26%	17.78%	13.50%	8.47%	8.78%

Figure 5. Heatmap of residual interval occupancy ratios for the Causal13 predictor. Darker shades indicate higher occupancy ratios.

Task01_BrainTumour exhibits the strongest concentration in the lowest residual interval. Its exact-zero residual occupancies are 83.23%, 84.06%, and 83.24% for Causal13, MED, and WCP, respectively, while only 2.82%, 3.19%, and 2.73% of the residuals fall in the $|d| \geq 64$ interval. In contrast, Task05_Prostate contains substantially larger high-magnitude residual tails, with $|d| \geq 64$ occupancies of 36.38%, 32.92%, and 29.15%, respectively. These results demonstrate why the complete residual distribution is more informative for LZC-based capacity analysis than the exact-zero residual ratio alone.

Table 2 reports the category-level mean embedding-capacity results based on file-level observations. For multidimensional NIfTI files, the corresponding 3D components are aggregated within each file before the category-level means are calculated. A_{BEC} is predictor-independent for a given component because it is determined by the fixed representation width and effective bit depth, whereas differences among predictors arise from residual-dependent capacity and auxiliary information overhead. Separate from the

category-level means reported in Table 2, a dataset-wide voxel-weighted pooling was also used to summarize the overall capacity behavior. Under voxel-weighted pooling across all evaluated components, Causal13, MED, and WCP achieve A_{TEC} values of 29.9871, 29.8386, and 30.0318 bits/voxel, respectively. Their corresponding A_{AUX} values are 3.0791, 2.8860, and 3.0663 bits/voxel, resulting in A_{PEC} values of 26.9079, 26.9526, and 26.9655 bits/voxel. Thus, WCP provides the highest pooled A_{TEC} and A_{PEC} , whereas MED requires the lowest auxiliary overhead; however, the A_{PEC} difference between WCP and MED is small. A compact comparison of A_{PEC} and A_{AUX} across the three predictors is provided in Figure 8 to visualize this capacity-overhead trade-off.

Class	$ d = 0$	$ d = 1$	$2 \leq d \leq 3$	$4 \leq d \leq 7$	$8 \leq d \leq 15$	$16 \leq d \leq 31$	$32 \leq d \leq 63$	$ d \geq 64$
Task01	84.06%	0.41%	0.99%	1.80%	2.82%	3.55%	3.18%	3.19%
Task02	34.99%	4.03%	5.56%	7.74%	10.38%	13.07%	13.03%	11.20%
Task03	15.19%	4.23%	8.15%	14.35%	19.92%	18.73%	10.96%	8.47%
Task05	25.53%	1.22%	2.30%	4.24%	7.48%	11.61%	14.70%	32.92%
Task06	21.12%	1.44%	2.85%	5.53%	9.88%	14.77%	17.10%	27.31%
Task07	22.71%	3.92%	7.57%	13.18%	17.54%	15.92%	10.13%	9.03%
Task08	23.30%	4.33%	8.86%	14.64%	17.54%	14.16%	8.42%	8.75%
Task09	19.07%	4.27%	8.60%	14.45%	18.49%	16.24%	9.61%	9.27%
Task10	19.04%	4.81%	9.05%	15.10%	18.72%	15.33%	8.91%	9.04%

Figure 6. Heatmap of residual interval occupancy ratios for the MED predictor. Darker shades indicate higher occupancy ratios.

Class	$ d = 0$	$ d = 1$	$2 \leq d \leq 3$	$4 \leq d \leq 7$	$8 \leq d \leq 15$	$16 \leq d \leq 31$	$32 \leq d \leq 63$	$ d \geq 64$
Task01	83.24%	0.67%	1.26%	2.21%	3.29%	3.73%	2.87%	2.73%
Task02	35.38%	5.13%	6.86%	9.19%	11.95%	13.95%	11.21%	6.33%
Task03	14.91%	5.81%	10.82%	17.51%	20.98%	16.16%	7.82%	5.99%
Task05	22.60%	1.42%	2.69%	5.00%	8.65%	13.45%	17.04%	29.15%
Task06	19.68%	1.84%	3.61%	6.87%	11.79%	16.67%	17.54%	22.00%
Task07	22.67%	5.16%	9.70%	15.53%	17.99%	14.37%	7.82%	6.76%
Task08	23.14%	5.95%	10.94%	16.35%	17.10%	12.64%	6.80%	7.08%
Task09	19.12%	5.91%	10.80%	16.61%	18.62%	14.34%	7.44%	7.16%
Task10	19.27%	6.37%	11.53%	17.41%	18.40%	13.24%	6.96%	6.82%

Figure 7. Heatmap of residual interval occupancy ratios for the WCP. Darker shades indicate higher occupancy ratios.

Table 2. Embedding capacity performance under different predictors.

Class	A_{BEC}	Causal13			MED			WCP		
		A_{AUX}	A_{TEC}	A_{PEC}	A_{AUX}	A_{TEC}	A_{PEC}	A_{AUX}	A_{TEC}	A_{PEC}
Task01	20.90	1.64	31.37	29.73	1.54	31.36	29.82	1.64	31.37	29.74
Task02	20.75	3.45	29.88	26.43	3.28	29.59	26.31	3.45	29.88	26.43
Task03	19.80	3.75	29.42	25.67	3.50	29.17	25.67	3.73	29.49	25.75
Task05	20.41	3.45	27.99	24.54	3.21	28.20	24.99	3.39	28.30	24.91
Task06	19.95	3.75	28.48	24.73	3.58	28.29	24.71	3.75	28.49	24.74
Task07	19.99	3.75	29.54	25.80	3.50	29.38	25.87	3.72	29.64	25.92
Task08	19.96	3.76	29.62	25.86	3.49	29.51	26.02	3.73	29.73	26.01
Task09	19.95	3.82	29.45	25.63	3.53	29.31	25.77	3.76	29.58	25.82
Task10	19.96	3.82	29.49	25.67	3.53	29.37	25.84	3.75	29.66	25.91

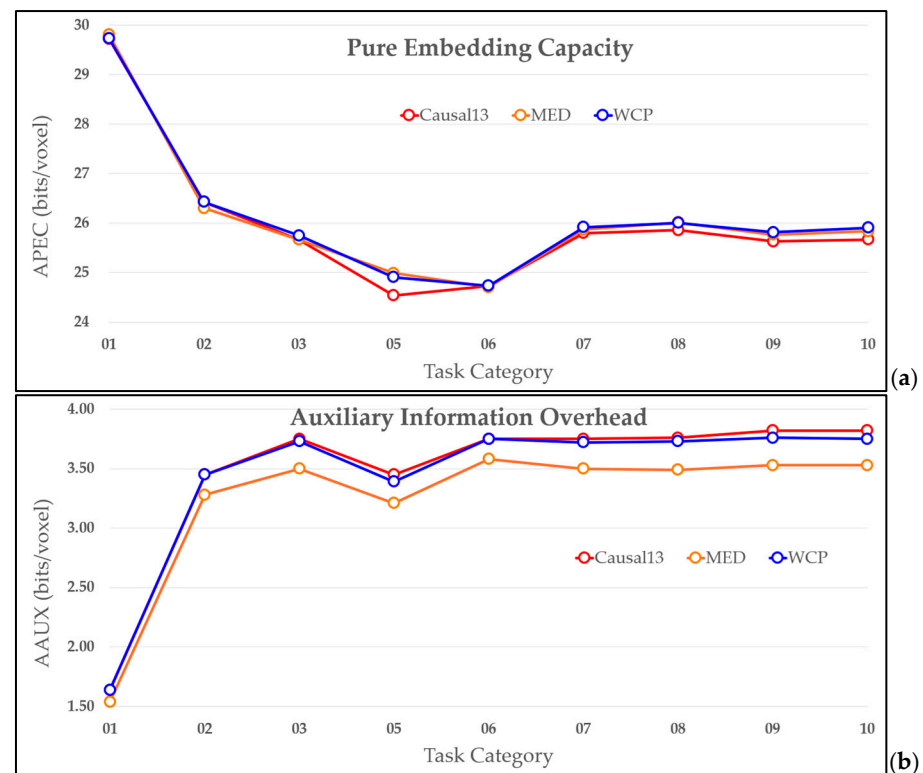


Figure 8. Comparison of (a) A_{PEC} and (b) A_{AUX} for Causal13, MED, and WCP across the evaluated dataset categories.

Prediction-error analysis further shows that lower MAE or RMSE does not necessarily imply higher usable capacity. Table 3 reports the file-level mean MAE and RMSE for each category. WCP achieves the lowest mean MAE and RMSE in eight of the nine categories, with Task06_Lung being the exception. Nevertheless, WCP achieves the highest category-level A_{PEC} in only six categories; MED provides a higher A_{PEC} for Task01_BrainTumour, Task05_Prostate, and Task08_HepaticVessel. The values in Table 3 are file-level means; components belonging to the same multidimensional NIfTI file are aggregated before the category-level statistics are calculated.

Table 3. Prediction-error performance under different predictors.

Class	Causal13 MAE	Causal13 RMSE	MED MAE	MED RMSE	WCP MAE	WCP RMSE
Task01	7.011	32.159	8.299	42.841	6.850	31.379
Task02	17.192	35.503	24.476	51.532	16.949	34.411
Task03	19.776	43.569	27.751	72.449	18.866	42.781
Task05	83.790	158.424	81.428	169.011	56.664	99.457
Task06	46.565	86.887	61.821	126.719	46.781	87.115
Task07	21.864	52.342	28.378	80.432	19.878	48.370
Task08	22.281	54.565	27.890	85.165	20.126	51.113
Task09	24.190	57.941	29.694	85.297	20.747	50.361
Task10	23.956	59.598	28.734	83.613	19.547	48.664

Task01 illustrates the distinction between prediction accuracy and usable capacity at the category level. WCP reduces the mean MAE and RMSE to 6.850 and 31.379, compared

with 8.299 and 42.841 for MED, yet MED achieves a higher A_{PEC} of 29.818 bits/voxel than the 29.736 bits/voxel obtained by WCP. The difference is mainly attributable to MED's smaller auxiliary information overhead. A more direct counterexample occurs in the second 3D component of prostate_00.nii.gz. Relative to MED, WCP reduces MAE from 128.496 to 105.672 and RMSE from 287.619 to 180.590, but its A_{TEC} decreases from 28.4298 to 27.8215 bits/voxel and its A_{PEC} decreases from 24.8555 to 23.9339 bits/voxel. MED produces 35.33% exact-zero residuals compared with 23.53% for WCP, while the $|d| \geq 64$ occupancy is 35.61% for MED and 43.98% for WCP. Thus, the smaller average prediction error of WCP does not translate into a more favorable bit-length distribution in this case.

The descriptive component-level correlation analysis leads to the same interpretation. Across the 2965 processed 3D components, the Spearman correlations between MAE and A_{PEC} are -0.9303 , -0.9270 , and -0.9325 for Causal13, MED, and WCP, respectively, while the corresponding RMSE– A_{PEC} correlations are -0.8191 , -0.8476 , and -0.8235 . However, after isolating the residual-dependent capacity A_{RDC} , the MAE correlations decrease to -0.2670 , -0.3228 , and -0.2228 , and the RMSE correlations decrease to -0.1097 , -0.2105 , and -0.0636 , respectively. These coefficients are used descriptively because multiple components can originate from the same NIfTI file.

The paired file-level analysis further indicates that the small differences in A_{PEC} should not be overstated. For WCP relative to Causal13, the mean paired A_{PEC} difference is $+0.0972$ bits/voxel, with a 95% bootstrap confidence interval of $[0.0917, 0.1026]$ and a rank-biserial effect size of 0.8744. In contrast, the mean WCP–MED difference is only -0.0039 bits/voxel, with a 95% bootstrap confidence interval of $[-0.0082, 0.0003]$ and a rank-biserial effect size of -0.0842 . The bootstrap interval estimates uncertainty in the mean paired difference, whereas the Wilcoxon signed-rank test evaluates the signed ranks of the paired differences and therefore addresses a different aspect of the paired distribution. Accordingly, the confidence interval spanning zero is not inconsistent with the Holm-adjusted Wilcoxon result of $p = 0.0050$. Given the very small mean difference and effect size, WCP and MED are considered to provide essentially comparable file-level A_{PEC} .

These results show that prediction-error magnitude alone does not determine predictor-specific capacity ordering; residual redistribution across effective-bit-length boundaries and auxiliary information overhead are also critical. The representation- and prediction-derived contributions are examined separately in Section 4.4.

4.4. Capacity Decomposition and Representation-Width Sensitivity

The reported embedding capacity originates from two distinct redundancy sources: fixed-representation redundancy determined by the difference between the representation width and effective bit depth, and residual-dependent redundancy generated through prediction. To distinguish their contributions to usable capacity, A_{PEC} is decomposed into the representation-derived payload capacity A_{PBE} and prediction-derived payload capacity A_{PRC} , as defined in Section 3.3.

For WCP under the standard $b = 32$ configuration, Table 4 summarizes the dataset-wide pooled capacity decomposition. The representation-derived and prediction-derived payload capacities account for approximately 63.29% and 36.71% of the final A_{PEC} , respectively. Across the nine evaluated categories, the corresponding contribution ranges are 61.47–68.27% and 31.73–38.53%, confirming that the high usable capacity under the 32-bit representation cannot be attributed to prediction alone.

Table 4. Capacity decomposition and representation-width sensitivity for WCP.

Configuration	A_{BEC}	A_{RDC}	A_{AUX}	A_{PBEC}	A_{PRC}	A_{PEC}	Status
$b = 32$	20.1340	9.8978	3.0663	17.0677	9.8978	26.9655	Feasible
$b = b_e + 8$	8.0000	≈ 9.8978	3.0663	4.9337	≈ 9.8978	14.8314	Feasible
$b = b_e + 4$	4.0000	≈ 9.8978	3.0663	0.9337	≈ 9.8978	10.8314	Feasible
$b = b_e$	0	N/A	N/A	N/A	N/A	N/A	Infeasible

The representation-width sensitivity results further show that A_{PEC} decreases substantially as the representation width approaches the component-specific effective bit depth b_e , whereas the prediction-derived contribution remains approximately stable. The reduction in usable capacity is therefore primarily attributable to the progressive removal of representation-derived redundancy rather than to a loss of prediction-derived redundancy. The limiting configuration $b = b_e$ is infeasible under the current embedding architecture because $C_{BEC} = N(b - b_e) = 0$. No basic-capacity positions are therefore available to store the fixed bootstrap header and residual-label information. Since the complete auxiliary stream must be accommodated before secret-data embedding, feasibility requires $b > b_e$ and sufficient BEC for the auxiliary information.

4.5. Effect of Inter-Slice Information and WCP Weighting

To evaluate the contribution of inter-slice information independently of the embedding mechanism, the predefined WCP configuration $w_z : w_{xy} = 1 : 2$, denoted as WCP-3D, was compared with the WCP variant with $w_z = 0$, denoted as WCP-2D. Residual labeling, Huffman coding, auxiliary information construction, and capacity calculation were kept unchanged. Therefore, the comparison isolates the effect of incorporating previous-slice information without changing the remaining embedding procedure.

Under voxel-weighted pooling across all evaluated components, WCP-2D achieves an A_{TEC} of 30.0403 bits/voxel, an A_{AUX} of 3.0484 bits/voxel, and an A_{PEC} of 26.9919 bits/voxel. The corresponding values for WCP-3D are 30.0318, 3.0663, and 26.9655 bits/voxel, respectively. The pooled exact-zero residual occupancies P_0 are 39.11% for WCP-2D and 38.83% for WCP-3D. Together with the corresponding A_{TEC} , A_{AUX} , and A_{PEC} results, these values indicate that incorporating previous-slice information does not produce a dataset-wide capacity advantage under the predefined 1 : 2 weighting.

The category-level results further show that the effect of inter-slice information is dataset-dependent. Relative to WCP-2D, WCP-3D increases A_{PEC} by approximately 0.1349, 0.0595, and 0.0355 bits/voxel for Task02_Heart, Task03_Liver, and Task06_Lung, respectively, but decreases A_{PEC} for the other six categories. The largest decrease occurs for Task05_Prostate, where A_{PEC} is reduced by approximately 0.5450 bits/voxel. These results indicate that previous-slice information can improve residual concentration and embedding capacity for specific volumetric categories, but its benefit is not universal.

To further examine the effect of predictor weighting, seven WCP configurations were evaluated: $(w_z, w_{xy}) \in \{(0, 1), (1, 3), (1, 2), (1, 1), (2, 1), (3, 1), (1, 0)\}$. Here, (0, 1) represents the in-slice-only configuration, (1, 0) uses only the previous-slice estimate, and (1, 2) is the predefined configuration used in the main experiments. Table 5 reports the corresponding unweighted mean A_{PEC} across the 1481 file-level observations.

Table 5. Effect of WCP weighting on file-level mean A_{PEC} .

$w_z:w_{xy}$	Configuration	File-Level Mean A_{PEC}
0:1	In-slice only	27.2069
1:3	Weighted WCP	27.1323
1:2	Predefined main setting	27.1016
1:1	Equal weighting	27.0374
2:1	Inter-slice emphasized	26.9677
3:1	Inter-slice emphasized	26.9324
1:0	Previous-slice only	26.8205

As shown in Table 5, the WCP variant with $w_z = 0$ provides the highest file-level mean A_{PEC} among the evaluated configurations. The mean A_{PEC} generally decreases as greater relative weight is assigned to the previous-slice estimate, reaching its lowest value when only the previous-slice estimate is used. The predefined 1 : 2 configuration, therefore, does not maximize A_{PEC} and should be regarded as a representative three-dimensional configuration rather than an optimized setting.

4.6. Effects of Zero-Valued Regions and Boundary Handling

The preceding analyses show that the contribution of inter-slice information is dataset-dependent. To further determine whether favorable residual distributions arise from volumetric correlation or from large homogeneous regions, a region-based analysis was performed using Task01_BrainTumour, which has the highest proportion of original zero-valued voxels among the evaluated categories. Because semantic foreground masks are not required by the proposed framework, voxels are separated conservatively into a zero-valued/background-proxy region, defined by an original voxel value of zero, and a nonzero region. These terms describe numerical voxel characteristics rather than anatomical foreground and background.

Approximately 83.89% of the evaluated Task01 voxels belong to the zero-valued region. Under WCP-2D, the exact-zero residual occupancy P_0 is 99.35% in this region, compared with only 1.94% in the nonzero region. The corresponding average residual-dependent capacities are 11.0585 and 7.4141 bits/voxel, respectively. WCP-3D exhibits a similar pattern: the zero-valued region has a P_0 of 98.83% and an average residual-dependent capacity of 11.0367 bits/voxel, whereas the nonzero region has a P_0 of 2.07% and a residual-dependent capacity of 7.5392 bits/voxel.

The large difference between the two regions indicates that the favorable residual distribution of Task01 is strongly associated with its extensive homogeneous zero-valued region. In particular, nearly all residuals within this region are predicted exactly under both WCP configurations, thereby producing long leading-zero prefixes and substantial residual-dependent capacity. The high overall capacity of Task01 should therefore not be attributed primarily to inter-slice correlation. Instead, it reflects the combined effects of homogeneous zero-valued structure, in-slice redundancy, and predictor-dependent spatial correlation. Because auxiliary information is encoded globally rather than separately for the two regions, the regional analysis is restricted to residual occupancy and residual-dependent capacity rather than assigning separate regional A_{AUX} or A_{PEC} values.

A separate analysis was conducted to determine whether the different boundary-handling rules of Causal13, MED, and WCP materially affect their comparison. Rather than modifying the original definitions of the predictors, all three methods were reevaluated on a common interior domain in which the required causal neighborhoods are

fully available: $\mathcal{V}_{int} = \{(i, j, k) | 2 \leq i \leq R - 1, 2 \leq j \leq C - 1, 2 \leq k \leq Z\}$. On this domain, predictor-specific boundary substitutions are not required, allowing the three predictors to be compared using identical spatial support. The dataset-wide ordering remains unchanged after boundary voxels are excluded. For A_{TEC} , the ordering remains $WCP > \text{Causal13} > \text{MED}$, whereas for A_{PEC} , it remains $WCP > \text{MED} > \text{Causal13}$. The difference between WCP and MED remains very small and is therefore not interpreted as evidence of meaningful superiority.

The category-level results are also highly stable. The only observed change involving the leading A_{PEC} methods occurs for Task08_HepaticVessel, where MED is slightly higher than WCP in the full-volume evaluation but WCP becomes slightly higher after restricting the comparison to the common interior domain. The difference is small and does not alter the dataset-wide conclusion. Moreover, the common interior retains most voxels in the evaluated volumes: approximately 99.0% for Task03_Liver, 98.9% for Task06_Lung, and 93.5% even for Task05_Prostate, which has the smallest retained fraction among the evaluated categories.

These analyses show that large homogeneous zero-valued regions can substantially improve residual concentration and prediction-derived capacity, whereas predictor-specific boundary handling has only a limited effect on the overall comparison.

4.7. End-to-End Reversibility and Computational Performance

To verify that the reported embedding capacities correspond to an executable reversible data hiding process, end-to-end embedding, secret-data extraction, and voxel recovery were evaluated under near-capacity conditions. For each of the nine dataset categories, one representative 3D component was selected automatically according to the predefined median-volume-size criterion before any end-to-end reversibility result was examined. The same selected component was then evaluated using Causal13, MED, and WCP, resulting in 27 validation cases. Thus, case selection was independent of extraction or recovery success and was not performed post hoc. In each case, the secret payload occupied approximately 99% of the available component-level A_{PEC} .

As summarized in Table 6, all 27 cases achieved error-free secret-data extraction and exact voxel recovery. Here, exact end-to-end reversibility denotes 100% secret-data extraction accuracy, $\text{BER} = 0$, zero secret-bit errors, zero reconstruction RMSE, zero maximum absolute voxel error, zero numerical voxel mismatches, and zero IEEE 754 binary32 word mismatches. These results confirm that the analytically estimated capacities can be realized under near-capacity embedding rather than only with small payloads.

Table 6. Reversibility and runtime performance of the proposed method.

Predictor	Cases	Recovery	Core Time (s/Mvoxel)	Throughput (Mvox/s)	End-to-End Time (s/Mvoxel)
Causal13	9	Exact	0.0941	10.63	12.09
MED	9	Exact	0.1626	6.15	11.06
WCP	9	Exact	0.1312	7.62	12.53

Exact recovery refers specifically to the processed voxel arrays. Numerical exactness requires each recovered single-precision voxel value to equal its original value, whereas word-level exactness additionally requires the corresponding IEEE 754 binary32 representations to be identical. The latter criterion also detects representation-level differences such as +0.0 and −0.0. The framework does not, however, claim byte-for-byte reconstruction of the complete NIfTI container, because header serialization, metadata representation, compression state, and other container-level information lie outside the embedding domain.

Computational performance was evaluated separately for the core capacity-analysis procedure and the complete near-capacity RRBE workflow. The core timing includes prediction, residual generation, label construction, Huffman-length evaluation, and capacity accounting. As shown in Table 6, Causal13 provides the highest core throughput, whereas the complete workflow requires approximately 11–13 s/Mvoxel for all three predictors. The predictor runtime observed in the core analysis therefore does not directly determine the end-to-end execution time because the complete workflow includes additional bit-level encryption, embedding, extraction, decryption, and reconstruction operations.

The dataset-wide analytical evaluation successfully processed all 1481 NIfTI files, corresponding to 2965 separately processed 3D components and 54,670,761,984 voxels, with no failed files. Using eight process-based MATLAB workers with dynamic file-level scheduling, the complete experimental batch required 15,251.34 s, or approximately 4.24 h, of wall-clock time. This dataset-wide timing refers to the analytical evaluation; the complete embedding-extraction-recovery workflow was evaluated separately on the 27 representative cases.

Memory consumption depends primarily on individual volume dimensions, temporary voxel and bitstream representations, and the number of concurrently processed files. File-level processing and chunked operations were used to limit temporary-memory growth, and the complete evaluation was successfully executed with 64 GB of RAM. Because direct peak-memory profiling was not included in the experimental instrumentation, no quantitative peak-memory requirement is reported.

4.8. Cross-Media Contextual Comparison

Table 7 provides a contextual comparison between the present framework and representative reversible data hiding studies involving conventional images, medical images, HDR images, audio signals, and polygonal models. Because the compared studies use different media representations, datasets, embedding mechanisms, payload definitions, and auxiliary information accounting procedures, Table 7 is intended solely as a contextual normalization and not as a direct performance ranking. The reported NPEE values should therefore not be interpreted as evidence of numerical superiority among the compared methods. Instead, the comparison provides a normalized context for examining how representation characteristics and exploitable redundancy differ across media types.

Table 7. Contextual cross-media comparison of normalized embedding efficiency.

Study	Zhang et al. [6]	Lin et al. [26]	Lin et al. [27]	Tsai et al. [28]	Tsai et al. [21]	Proposed
Method	Regression-based PEE	Pattern-matching RDH	Multi-hider HDR RDH	Interpolation-based RDH	Vertex-traversal RDH	Residual-aware RRBE
Media Type	Images	Medical Images	HDR Images	Audio	Polygonal Models	3D Medical Volumes
b_{ref}	8	16	32	16	96	32
C_{ref}	4.364 bpp	>10 bpp	16.29 bpp	9.46 bits/sample	44.19 bits/vertex	26.9655 bits/voxel
NPEE	54.55%	>62.50%	50.91%	59.13%	46.03%	84.27%

For each referenced study, Table 7 reports the media type, reference representation width b_{ref} , selected source-reported capacity C_{ref} , and the resulting NPEE calculated using (12). For the present framework, $C_{ref} = A_{PEC}$ and $b_{ref} = 32$. For the polygonal-model method in [21], $b_{ref} = 96$ bits/vertex because each vertex contains three 32-bit coordinate components. Because these studies do not define usable payload capacity and auxiliary

information overhead in a uniform manner, the resulting NPEE values are interpreted only as normalized contextual indicators rather than directly equivalent capacity measures.

The cross-media comparison also highlights that different media representations provide fundamentally different sources of redundancy. Conventional image methods primarily exploit two-dimensional spatial correlation, whereas medical-image methods may additionally benefit from homogeneous regions and modality-specific intensity structures. HDR-image methods can exploit redundancy associated with floating-point representation fields, while audio methods primarily rely on temporal correlation. Polygonal-model methods operate on irregular geometric and connectivity structures, where redundancy may arise from coordinate relationships, vertex ordering, connectivity, and traversal organization.

For the present volumetric framework, the high A_{PEC} under the standard 32-bit representation should not be attributed solely to three-dimensional prediction. As shown in Section 4.4, the final usable capacity consists of both representation-derived payload capacity and prediction-derived payload capacity. Under WCP, these contributions account for approximately 63.29% and 36.71% of the pooled A_{PEC} , respectively. The representation-width sensitivity analysis further demonstrates that A_{PEC} decreases substantially as b approaches the component-specific effective bit depth b_e , while the prediction-derived contribution remains comparatively stable. Thus, a substantial portion of the reported capacity arises from fixed-representation redundancy rather than from residual prediction alone.

The prediction-derived contribution is nevertheless governed by the residual distribution. Under the adopted LZC-based rule, prediction improves capacity only when residuals are redistributed across effective-bit-length boundaries. Moreover, as demonstrated in Section 4.5, incorporating inter-slice information does not universally increase A_{PEC} ; its effect depends on the dataset and predictor weighting. Therefore, the proposed framework should be interpreted as exploiting the combined effects of representation-derived redundancy and residual-dependent redundancy rather than as demonstrating intrinsically stronger three-dimensional predictability than other media.

Accordingly, the NPEE values in Table 7 provide only a cross-media contextual reference and should not be interpreted as demonstrating numerical superiority of the proposed framework over the referenced studies. Differences in native representation width, storage format, dataset characteristics, redundancy source, auxiliary information treatment, and capacity definition prevent a rigorous quantitative comparison of embedding efficiency across these studies. A direct performance comparison would require reimplementing of all methods under a unified representation, dataset, and accounting protocol.

4.9. Limitations

The present study has several limitations. First, the current framework is restricted to finite, numerically integer-valued voxel samples stored in single-precision format. Such values can be mapped losslessly to the adopted nonnegative integer domain without quantization. Genuinely non-integer floating-point volumes are not supported by the current implementation; this restriction is the reason for excluding Task04_Hippocampus from the experimental evaluation. Extending the framework to such data would require a representation-aware reversible strategy that directly preserves the floating-point representation or otherwise guarantees exact recovery without numerical quantization. This extension remains future work rather than a capability of the present method.

Second, the demonstrated reversibility applies to the processed voxel arrays rather than to the complete NIfTI file container. The end-to-end validation confirms exact equality at both the numerical single-precision voxel level and the IEEE 754 binary32 word level. However, byte-for-byte reconstruction of the complete NIfTI file, including header serialization, metadata organization, and compression state, is outside the scope of the

current framework. This distinction should be considered when integrating the method into applications requiring preservation of the entire medical-data container.

Third, the separable embedding design intentionally exposes the auxiliary information required for residual-dependent position determination to the data hider. Although the exact residual magnitudes and voxel values cannot be recovered without K_E , the public header and residual-label stream reveal processing metadata and residual-derived information, including the effective bit-length class and sign of each spatially ordered residual. Such information may expose statistical or spatial characteristics of the underlying volume and therefore should not be interpreted as confidential under the present threat model. A stronger privacy model in which residual-derived metadata must also remain hidden would require additional protection of the auxiliary information or a different position-determination mechanism. Possible approaches include trusted or isolated execution for position generation, secure computation, or restricting the data hider to content-independent embedding positions; however, these alternatives may reduce separability or usable embedding capacity and are outside the scope of the current framework.

Fourth, prediction performance depends on both dataset characteristics and predictor configuration. The predefined WCP weighting $w_z : w_{xy} = 1 : 2$ is used as a representative three-dimensional configuration rather than an optimized setting. As demonstrated by the sensitivity analysis, alternative weightings can provide higher APEC, and incorporating inter-slice information does not universally improve embedding capacity. Therefore, the current WCP configuration should not be interpreted as universally optimal, and adaptive predictor or weight selection may provide further improvements for datasets with different spatial characteristics.

Fifth, complete embedding-extraction-recovery validation was performed on 27 representative cases, comprising one representative 3D component from each evaluated category under each of the three predictors, rather than on all 2965 processed components. The dataset-wide experiments establish capacity and residual-distribution behavior over the full dataset, whereas the representative end-to-end tests verify executable near-capacity embedding and exact recovery. A larger-scale end-to-end validation would provide additional evidence regarding practical robustness across the complete dataset.

Finally, although the implementation successfully processed the full dataset using file-level parallelism and chunked operations, direct peak-memory profiling was not incorporated into the experimental instrumentation. Consequently, no quantitative peak-memory requirement is reported. Memory consumption depends on individual volume dimensions, temporary voxel and bitstream representations, and the number of concurrently processed files; systematic profiling across different hardware configurations remains a useful direction for future implementation-oriented evaluation.

5. Conclusions and Future Work

This study presented a residual-distribution-aware capacity analysis for reversible data hiding in encrypted 3D medical volumes with integer-valued voxels. The results show that LZC-based embedding capacity is governed more directly by residual occupancy across power-of-two bit-length intervals than by prediction-error magnitude alone. Therefore, reducing MAE or RMSE does not necessarily increase embedding capacity unless residuals move into more favorable bit-length intervals.

The experiments further show that usable capacity is jointly determined by representation-derived payload capacity, prediction-derived payload capacity, and auxiliary information overhead. Under the 32-bit representation, WCP achieves a pooled A_{PEC} of 26.9655 bits/voxel. However, its advantage over MED is small, and the benefit of inter-slice prediction is dataset-dependent. End-to-end validation on 27 representative near-capacity cases achieved error-free

secret-data extraction and exact recovery of the processed voxel arrays at both the numerical and IEEE 754 binary32 word levels.

Future work will focus on extending the framework to genuinely non-integer floating-point volumetric data, developing adaptive predictor and weighting strategies, and further evaluating memory usage and large-scale end-to-end performance.

Author Contributions: Conceptualization, M.-C.C., Y.-A.H. and Y.-Z.L.; methodology, Y.-Y.T. and C.-T.L.; funding acquisition, Y.-Y.T. and M.-C.C.; software, Y.-Y.T., M.-C.C., Y.-A.H. and Y.-Z.L.; validation, Y.-Y.T. and C.-T.L.; data curation, M.-C.C., Y.-A.H. and Y.-Z.L.; writing—original draft preparation, Y.-Y.T., M.-C.C., Y.-A.H. and Y.-Z.L.; writing—review and editing, Y.-Y.T. and C.-T.L.; supervision, Y.-Y.T. and C.-T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Science and Technology Council of Taiwan under the grant numbers NSTC 115-2813-C-035-121-E and NSTC 115-2221-E-035-080.

Institutional Review Board Statement: This study used publicly available and de-identified data from the Medical Segmentation Decathlon dataset. According to Article 5 of the Human Subjects Research Act of Taiwan and the “Scope of Human Research Projects Exempt from Review by an Ethics Review Committee,” research using lawfully and publicly available information for purposes consistent with its public disclosure is exempt from IRB review. Therefore, additional Institutional Review Board approval was not required for the present study.

Data Availability Statement: The authors used publicly archived datasets in this study. The Medical Segmentation Decathlon (MSD) dataset is publicly available at <https://medicaldecathlon.com/> (accessed on 31 August 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rauniyar, A.; Hagos, D.H.; Jha, D.; Håkegård, J.E.; Bagci, U.; Rawat, D.B.; Vlassov, V. Federated Learning for Medical Applications: A Taxonomy, Current Trends, Challenges, and Future Research Directions. *IEEE Internet Things J.* **2024**, *11*, 7374–7398. [\[CrossRef\]](#)
2. Ren, S.; Li, X. HResFormer: Hybrid Residual Transformer for Volumetric Medical Image Segmentation. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 10558–10566. [\[CrossRef\]](#)
3. Zhou, H.-Y.; Guo, J.; Zhang, Y.; Han, X.; Yu, L.; Wang, L.; Yu, Y. NnFormer: Volumetric Medical Image Segmentation via a 3D Transformer. *IEEE Trans. Image Process.* **2023**, *32*, 4036–4045. [\[CrossRef\]](#)
4. Huang, C.-T.; Dang, T.T.-H.; Weng, C.-Y. Reversible Data Hiding in Encrypted Images Based on Bidirectional Encoding for Payload Enhancing. *J. Inf. Secur. Appl.* **2025**, *93*, 104183. [\[CrossRef\]](#)
5. Ren, H.; Yue, Z.; Li, M. Reversible Data Hiding in Encrypted Images Using Adaptive Block-level Pixel Difference Encoding. *J. Inf. Secur. Appl.* **2026**, *96*, 104316. [\[CrossRef\]](#)
6. Zhang, X.; Zhang, X.; Yu, C.; Yang, J.; He, W.; Tang, Z. High-capacity Reversible Data Hiding in Encrypted Images via Robust Regression and Deep-feature-guided Prediction Error Encoding. *IEEE Trans. Consum. Electron.* **2026**, *72*, 3403–3417. [\[CrossRef\]](#)
7. Fang, G.; Wang, F.; Zhao, C.; Qin, C.; Chang, C.-C.; Chang, C.-C. Reversible Data Hiding with Secret Encrypted Image Sharing and Adaptive Coding. *IEEE Internet Things J.* **2025**, *12*, 23933–23945. [\[CrossRef\]](#)
8. Yu, C.; Zhang, X.; Li, G.; Liu, P.; Zhang, X.; Tang, Z. Reversible Data Hiding in Encrypted Images with Secret Sharing and Multivariate Linear Equation. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 6767–6780. [\[CrossRef\]](#)
9. Xiong, L.; Zhou, W.; Xia, Z.; Weng, J. A Universal Framework for Reversible Data Hiding in Encrypted Images with Multiple Hiders. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 3944–3959. [\[CrossRef\]](#)
10. Sun, J.; Ke, Y.; Zhang, M.; Xiang, S. Reversible Data Hiding in Encrypted Images with Dual-phase Embedding Based on Multi-key Threshold Decryption. *IEEE Trans. Circuits Syst. Video Technol.* **2026**, *36*, 7080–7094. [\[CrossRef\]](#)
11. Kong, P.; Fu, D.; Li, X.; Qin, C. Reversible Data Hiding in Encrypted Medical DICOM Image. *Multimed. Syst.* **2021**, *27*, 303–315. [\[CrossRef\]](#)
12. Dzwonkowski, M.; Czaplewski, B. Reversible Data Hiding in Encrypted DICOM Images Using Sorted Binary Sequences of Pixels. *Signal Process.* **2022**, *199*, 108621. [\[CrossRef\]](#)
13. Martyniak, R.; Dzwonkowski, M. Reversible Data Hiding in Encrypted DICOM Images with Fixed and Block-wise Pixel Prediction. *Signal Process.* **2026**, *239*, 110287. [\[CrossRef\]](#)

14. Yang, Y.; He, H.; Chen, F.; Yuan, Y.; Mao, N.; Li, Y.; Zhao, J. Reversible Data Hiding in Encrypted Medical Images Based on Huffman Tree Coding and Count-encryption. *IEEE Trans. Multimed.* **2025**, *27*, 7419–7431. [[CrossRef](#)]
15. Jiang, R.; Zhou, H.; Zhang, W.; Yu, N. Reversible Data Hiding in Encrypted Three-dimensional Mesh Models. *IEEE Trans. Multimed.* **2018**, *20*, 55–67. [[CrossRef](#)]
16. Lyu, W.-L.; Cheng, L.; Yin, Z. High-capacity Reversible Data Hiding in Encrypted 3D Mesh Models Based on Multi-MSB Prediction. *Signal Process.* **2022**, *201*, 108686. [[CrossRef](#)]
17. Hou, G.; Ou, B.; Long, M.; Peng, F. Separable Reversible Data Hiding for Encrypted 3D Mesh Models Based on Octree Subdivision and Multi-MSB Prediction. *IEEE Trans. Multimed.* **2024**, *26*, 2395–2407. [[CrossRef](#)]
18. Gao, K.; Horng, J.-H.; Chang, C.-C. Reversible Data Hiding for Encrypted 3D Mesh Models with Secret Sharing over Galois Field. *IEEE Trans. Multimed.* **2024**, *26*, 5499–5510. [[CrossRef](#)]
19. Wang, K.; Yao, Y.; Shen, Y.; Xiao, F.; Ren, Y.; Meng, W. High-capacity Reversible Data Hiding in Encrypted 3D Mesh Models Based on Dynamic Prediction and Virtual Connection. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 2996–3010. [[CrossRef](#)]
20. Gao, K.; Horng, J.-H.; Chang, C.-C.; Chang, C.-C. Crypto-space Reversible Data Hiding for 3D Mesh Models with K-degree Neighbor Diffusion. *J. Inf. Secur. Appl.* **2025**, *89*, 103957. [[CrossRef](#)]
21. Tsai, Y.-Y.; Jao, W.-T.; Chen, Y.-H. Authentication-enabled Reversible Data Hiding in Encrypted 3D Meshes via Effective Vertex Traversal and Secret Sharing. *ACM Trans. Multimed. Comput. Commun. Appl.* **2026**, *22*, 86. [[CrossRef](#)]
22. Xu, S.; Horng, J.-H.; Chang, C.-C.; Chang, C.-C. Reversible Data Hiding in Encrypted 3D Mesh Models via Ripple Prediction. *J. Vis. Commun. Image Represent.* **2025**, *111*, 104551. [[CrossRef](#)]
23. Tsai, Y.-Y.; Li, C.-Y.; Ho, C.-Y.; Lu, C.-T. Efficient Reversible Data Hiding in Encrypted Point Clouds via KD Tree-based Path Planning and Dual-model Design. *Mathematics* **2025**, *13*, 2593. [[CrossRef](#)]
24. Tsai, Y.-Y. Reversible Data Hiding in Encrypted Polygonal Faces Using Vertex Index Similarity. *IEEE Trans. Multimed.* **2025**, *27*, 9603–9618. [[CrossRef](#)]
25. Antonelli, M.; Reinke, A.; Bakas, S.; Farahani, K.; Kopp-Schneider, A.; Landman, B.A.; Litjens, G.; Menze, B.; Ronneberger, O.; Summers, R.M.; et al. The Medical Segmentation Decathlon. *Nat. Commun.* **2022**, *13*, 4128. [[CrossRef](#)]
26. Lin, Y.; Liu, J.-C.; Chang, C.-C.; Chang, C.-C. Reversible Data Hiding in Encrypted DICOM Images via Pattern Matching and Dynamic Huffman Coding. *Displays* **2025**, *90*, 103105. [[CrossRef](#)]
27. Lin, A.; Lin, Y.-T.; Jao, W.-T.; Tsai, Y.-Y. High-capacity Reversible Data Hiding in Encrypted HDR Images with Multiple Data Hiders. *Pattern Anal. Appl.* **2025**, *28*, 159. [[CrossRef](#)]
28. Tsai, Y.-Y.; Lin, A.; Jao, W.-T.; Chen, Y.-H. Interpolation-based Reversible Data Hiding in Encrypted Audio with Scalable Embedding Capacity. *Comput. Mater. Contin.* **2025**, *84*, 681–697. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.