A. Proof of Theorems and Propositions

A.1. Proof of Theorem 1

Proof. The proof proceeds in two stages. First, we derive a general formula for the amplification of expected noise energy for any given network layer. Then we apply this formula to the key components of a UNet to demonstrate the overall amplification effect.

We conduct the proof from the perspective of the predefined Expected Energy Amplification Factor as shown in Eq 2.

For a standard 2D convolution with a $k \times k$ kernel and c_{in} input channels, the fan-in is $n_{in} = k^2 c_{in}$. Under Kaiming initialization, the expected squared value of any weight is $\mathbb{E}[w^2] = \text{Var}(w) = 2/n_{in}$. An output feature map has d_{out} pixels. The squared Frobenius norm of the full Jacobian is the sum of squared weights, accounting for how many times each weight is applied. For each of the d_{out} output pixels, the sum of squared weights affecting it has an expectation of $n_{in} \cdot \mathbb{E}[w^2] = n_{in} \cdot (2/n_{in}) = 2$. Thus, the expected squared Frobenius norm is $\mathbb{E}[\|\boldsymbol{J}\|_F^2] \approx 2d_{out}$. For a resolution-preserving convolution, $d_{in} \approx d_{out}$. The expected amplification factor is

$$\mathbb{E}[\mathcal{A}] \approx \frac{2d_{out}}{d_{in}} \approx 2, \tag{6}$$

which indicates that even a standard convolutional layer is expected to double the energy of white noise.

Transposed convolutions are a primary source of noise amplification. Consider an upsampling operation with a stride st (typically $st = 2$), which increases the number of pixels by a factor of st^2. Thus, $d_{out} \approx st^2 d_{in}$. The same Kaiming initialization logic applies. The expected amplification factor becomes

$$\mathbb{E}[\mathcal{A}] \approx \frac{2d_{out}}{d_{in}} \approx \frac{2(st^2 d_{in})}{d_{in}} = 2st^2 \tag{7}$$

For a typical UNet with $st = 2$, we have $\mathbb{E}[\mathcal{A}] \approx 8$, demonstrating that **transposed convolutions are highly expansive and are expected to amplify noise energy significantly**.

Consider a max pooling layer with a window size $h \times h$ (typically $h = 2$). The Jacobian for max pooling contains a single '1' for each output pixel, selecting the maximum value from a patch. The squared Frobenius norm is therefore exactly equal to the output dimension as $\|\boldsymbol{J}\|_F^2 = d_{out}$, where the input dimension is $d_{in} \approx h^2 d_{out}$. Thus, the amplification factor is

$$\mathcal{A} = \frac{\|\boldsymbol{J}\|_F^2}{d_{in}} = \frac{d_{out}}{h^2 d_{out}} = \frac{1}{h^2} \tag{8}$$

For $h = 2$, we have $\mathcal{A} = 1/4$. Pooling layers are contractive and reduce noise energy as expected.

For the ReLU activation, the Jacobian $\boldsymbol{J}_{\text{ReLU}}$ is a diagonal matrix with entries:

$$\boldsymbol{J}_{\text{ReLU}} = \frac{\partial f_i}{\partial x_j} = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{if } x_i \leq 0 \end{cases}$$

Its squared Frobenius norm is simply the number of active neurons (positive inputs), denoted as $\|\boldsymbol{J}_{\text{ReLU}}\|_F^2 = \sum_{i=1}^{d_{in}} \mathbb{I}(x_i > 0)$. Assuming inputs are pre-activations from a layer initialized with Kaiming initialization, they are typically symmetric around zero. The probability of a neuron being active is $p = 0.5$. The expected number of active neurons is $\mathbb{E}[\|\boldsymbol{J}_{\text{ReLU}}\|_F^2] = d_{in} \cdot p = 0.5 d_{in}$. Since $d_{out} = d_{in}$, the expected amplification factor is

$$\mathbb{E}[\mathcal{A}] = \frac{0.5 d_{in}}{d_{in}} = 0.5 \tag{9}$$

Consider a simplified linear attention mechanism of the form $\boldsymbol{y} = \boldsymbol{W}_v \boldsymbol{x} \cdot \text{softmax}(\boldsymbol{W}_k \boldsymbol{x})^\top \boldsymbol{W}_q \boldsymbol{x}$. For the purpose of noise propagation analysis, we approximate its core operation as a linear transformation $\boldsymbol{y} \approx \boldsymbol{W}\boldsymbol{x}$, where $\boldsymbol{W}$ encapsulates the combined effect of the key, query, and value projections. Here the Jacobian is approximately $\boldsymbol{J}_{\text{attn}} \approx \boldsymbol{W}$.

Assuming $\boldsymbol{W}$ is initialized with variance $\sigma_w^2 = 2/n_{in}$, where n_{in} is the fan-in (dimension of the input feature vector). The expected squared Frobenius norm is:

$$\begin{aligned} \mathbb{E}[\|\boldsymbol{J}_{\text{attn}}\|_F^2] &\approx \mathbb{E}[\|\boldsymbol{W}\|_F^2] = d_{out} \cdot n_{in} \cdot \sigma_w^2 \\ &= d_{out} \cdot n_{in} \cdot \frac{2}{n_{in}} = 2d_{out} \end{aligned}$$

Thus, the expected amplification factor is

$$\mathbb{E}[\mathcal{A}] \approx \frac{2d_{out}}{d_{in}} \tag{10}$$

If the attention mechanism does not change dimensionality ($d_{out} = d_{in}$), then $\mathbb{E}[\mathcal{A}] \approx 2$.

Our analysis reveals a fundamental imbalance in the UNet architecture regarding noise propagation. The encoder path uses contractive pooling layers ($\mathcal{A} \approx 1/4$), which suppress noise energy. However, the decoder path relies on highly expansive transposed convolutions ($\mathcal{A} \approx 8$) for upsampling. Furthermore, skip connections feed noise from the encoder directly into these expansive decoder blocks.

The potent energy amplification in the decoder path is not counteracted by the encoder's suppression. Instead, the effects compound, leading to a significant net increase in the expected noise energy from the network's input to its output. This rigorous analysis provides a strong theoretical foundation for the proposition that the symmetric, long-path architecture of the UNet is a primary contributor to its sensitivity to noise. □

A.2. Proof of Proposition 1

Proof. Consider a UNet encoder with five layers. Thus, the input image x_0 is processed by a sequence of 2×2 max-pooling and stride=2 convolution operations, yielding five intermediate feature maps denoted $x_0, x_1, \cdots, x_5$ with spatial resolutions $352 \times 352, 176 \times 176, \cdots, 11 \times 11$, respectively, as illustrated in Figure 3a). Each arrow in Figure 3a and Figure 3b represents a single convolution. We denote the noise energy at the inputs of the downsampling stages by $E_0, E_1, \cdots, E_5$.

We use $\mathcal{A}_{\text{conv}}, \mathcal{A}_{\text{up}}, \mathcal{A}_{\text{down}}$ to stand for the EEA of convolutional, transpose convolution, and max-pooling layers. Thus, the transformation of adjacent layers during the downsampling stages, i.e., a 2×2 max-pooling and stride=2 convolution operations, leads to a total EEA of $\mathcal{A}_{\text{down}}\mathcal{A}_{\text{conv}} = \frac{1}{2}$, according to Table 1. Hence $E_k = \frac{1}{2^k}E_0$.

Let $y_{\text{unet}}, y_{\text{cas}}$ denote the output of UNet and CASUNet, and $\Delta\cdot$ indicate the noise of a given tensor. Besides, we use the notation of y_l and y_h to denote the low- and high-level aggregated features, serving as the input of the CPA module.

For the typical UNet, the energy of the output noise $E(\Delta y)$ can be computed as

$$E(\Delta y_{\text{unet}}) = E_5 \cdot (\mathcal{A}_{\text{up}}\mathcal{A}_{\text{conv}})^4 = \frac{1}{2^5}E_0 \cdot 16^4 = 2048E_0$$

As for CASUNet, the energy of the low- and high-level feature noise can be computed as

$$\begin{aligned} E(\Delta y_l) &= E_5 \cdot \mathcal{A}_{\text{conv}}\mathcal{A}_{\text{up}}^2 + E_4 \cdot \mathcal{A}_{\text{up}}\mathcal{A}_{\text{conv}}^2 + E_3 \cdot \mathcal{A}_{\text{conv}} \\ &= \frac{25}{4}E_0 \\ E(\Delta y_h) &= E_2 \cdot \mathcal{A}_{\text{conv}} + E_1 \cdot \mathcal{A}_{\text{conv}}^2\mathcal{A}_{\text{down}} + E_0 \cdot \mathcal{A}_{\text{conv}}^2\mathcal{A}_{\text{down}}^2 \\ &= \frac{5}{4}E_0 \end{aligned}$$

For analytical simplicity, we model the CPA module as if it were linear (in practice, the true nonlinear transform further suppresses noise as shown in Proposition 4). The aggregation carried out in the first half of the module is given by the Eq (3). Since this operation is merely a linear combination of the input vectors and the sigmoid function is bounded by 1, the noise after transformation admits the upper bound

$$\begin{aligned} E(\Delta s_l) &= E(\Delta[\sigma(y_l)y_l + (1-\sigma(y_l))\sigma(y_h)y_h]) \\ E(\Delta s_h) &= E(\Delta[\sigma(y_h)y_h + (1-\sigma(y_h))\sigma(y_l)y_l]) \end{aligned}$$

Hence

$$\begin{aligned} E(\Delta[s_l + s_h]) =& E(\Delta[(2-\sigma(y_h))\sigma(y_l)y_l \\ &+ (2-\sigma(y_l))\sigma(y_h)y_h]) \\ \leq& 2E(\Delta[\sigma(y_l)y_l + \sigma(y_h)y_h]) \\ \leq& 2E(\Delta[y_l + y_h]) \end{aligned}$$

$$\begin{aligned} E(\Delta u_l) &= E(\Delta[y_l + s_l]) \\ E(\Delta u_h) &= E(\Delta[y_h + s_h]) \end{aligned}$$

Thus,

$$\begin{aligned} E(\Delta u) &= E(\Delta[u_l + u_h]) = E(\Delta[y_l + y_h + s_l + s_h]) \\ &\leq 3E(\Delta[y_l + y_h]) \end{aligned}$$

After which this output of the CPA module will pass two transpose convolution layers, implying the final noise energy as:

$$E(\Delta y_{\text{cas}}) = E(\Delta u) \cdot \mathcal{A}_{\text{up}}^2 \leq 1440E_0$$

Therefore, the deduction of the noise energy of CASUNet compared to UNet is at least $(E(\Delta y_{\text{unet}}) - E(\Delta y_{\text{cas}}))/E(\Delta y_{\text{unet}}) = 30\%$.

□

A.3. Proof of Proposition 2

Assuming s_l, s_h as the low- and high-level interaction term, the interaction gain of polynomial aggregation compared to linear aggregation is:

$$\Delta\text{HSIC} = \text{HSIC}_{\text{PA}} - \text{HSIC}_{\text{LA}} \geq \sum_{k=2}^{n} \|\boldsymbol{C}_{s_l^k s_h^k}\|_{\text{HS}}^2 > 0$$

Proof. Consider the feature representations as elements in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with kernel K. The interaction degree is quantified by the Hilbert-Schmidt Independence Criterion (HSIC):

$$\text{HSIC}(u_l, u_h) = \|\boldsymbol{C}_{u_l u_h}\|_{\text{HS}}^2$$

where $\boldsymbol{C}_{u_l u_h}$ is the cross-covariance operator in $\mathcal{H}$.

For linear aggregation (LA):

$$\begin{aligned} u_l^{\text{LA}} &= y_l + s_l \\ u_h^{\text{LA}} &= y_h + s_h \end{aligned}$$

The HSIC decomposes as:

$$\text{HSIC}(u_l^{\text{LA}}, u_h^{\text{LA}}) \leq \underbrace{\text{HSIC}(y_l, y_h)}_{\text{signal}} + \underbrace{\text{HSIC}(s_l, s_h)}_{\text{1st-order interaction}}$$

Since s_l, s_h are first-order features:

$$\text{HSIC}(s_l, s_h) \leq \lambda_{\max}(\Sigma_y)\|K\|^2$$

where $\lambda_{\max}$ is the largest eigenvalue of the covariance matrix Σ_y.

For polynomial aggregation (PA), we consider the feature maps:

$$\phi^{\text{PA}}(u_l) = \left[y_l, s_l, s_l^2, \cdots, s_l^n\right]^T \tag{11}$$

$$\phi^{\text{PA}}(u_h) = \left[y_h, s_h, s_h^2, \cdots, s_h^n\right]^T \tag{12}$$

The cross-covariance operator becomes:

$$\boldsymbol{C}_{u_l u_h}^{\text{PA}} = \begin{bmatrix} \boldsymbol{C}_{y_l y_h} & 0 & \cdots & 0 \\ 0 & \boldsymbol{C}_{s_l s_h} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{C}_{s_l^n s_h^n} \end{bmatrix}$$

The HSIC for PA is:

$$\begin{aligned} \text{HSIC}(u_l^{\text{PA}}, u_h^{\text{PA}}) &= \|\boldsymbol{C}_{u_l u_h}^{\text{PA}}\|_{\text{HS}}^2 \\ &= \sum_{k=0}^{n} \|\boldsymbol{C}_{s_l^k s_h^k}\|_{\text{HS}}^2 \\ &\geq \underbrace{\|\boldsymbol{C}_{y_l y_h}\|_{\text{HS}}^2}_{\text{signal}} + \underbrace{\|\boldsymbol{C}_{s_l s_h}\|_{\text{HS}}^2}_{\text{1st-order}} + \sum_{k=2}^{n} \underbrace{\|\boldsymbol{C}_{s_l^k s_h^k}\|_{\text{HS}}^2}_{\text{k-th order}} \end{aligned}$$

The key inequality holds because:

$$\|\boldsymbol{C}_{s_l^k s_h^k}\|_{\text{HS}}^2 \geq \left(\mathbb{E}[s_l^k s_h^k]\right)^2 > 0 \quad \forall k$$

Thus the interaction gain is:

$$\Delta\text{HSIC} = \text{HSIC}_{\text{PA}} - \text{HSIC}_{\text{LA}} \geq \sum_{k=2}^{n} \|\boldsymbol{C}_{s_l^k s_h^k}\|_{\text{HS}}^2 > 0$$

since at least one $\|\boldsymbol{C}_{s_l^k s_h^k}\|_{\text{HS}} > 0$.

This confirms that polynomial aggregation strictly increases feature interaction. □

A.4. Proof of Proposition 3

Under the Chebyshev polynomial aggregation with polynomial normalization, the upper bound of the noise in the aggregated features is given by $\left(\left[(\frac{\pi}{2}+1)\nabla_y s + \mathbb{I}\right]\epsilon\right.$.

Proof. When using Chebyshev polynomial aggregation, the aggregated feature can be expressed as

$$u = y + \sum_{i=1}^{d} T^{(i)}(s)$$

where each term is $T^{(i)}(s) = \cos(i \arccos s)$. Considering the k-th term in this expansion, the associated noise component is given by the gradient with respect to y, derived via the chain rule as

$$\nabla_y T^{(k)}(s) = \nabla_s T^{(k)}(s) \cdot \nabla_y s$$

The gradient with respect to s is computed element-wise as

$$\nabla_s T^{(k)}(s) = \frac{k \cdot \sin(k \arccos s)}{\sqrt{1-s^2}}$$

Substituting $s = \cos t$ to simplify, we get

$$\nabla_s T^{(k)}(s) = \frac{k \sin(kt)}{|\sin t|}$$

Under the specified polynomial normalization, scaling by a factor of $1/k^2$ yields

$$\nabla_s T^{(k)}(s) = \frac{1}{k^2} \cdot \frac{k \sin(kt)}{|\sin t|} = \frac{\sin(kt)}{k|\sin t|}$$

Consequently, the cumulative noise contribution from all terms up to degree d is formulated as

$$\frac{1}{\sin t} \sum_{i=1}^{d} \frac{\sin it}{i}$$

where the summation $\sum_{i=1}^{d} \frac{\sin it}{i}$ corresponds to the partial sum of the Fourier sine series for the function $f(t) = \frac{\pi - t}{2}$ on the interval $(0, 2\pi)$, represented as $\sum_{k=1}^{\infty} \frac{\sin(kt)}{k}$. This partial sum, denoted $S_d(t) = \sum_{k=1}^{d} \frac{\sin kt}{k}$, can be expressed through the Dirichlet integral:

$$S_d(t) - f(t) = \frac{1}{\pi} \int_{-\pi}^{\pi} \phi_t(\tau) D_d(\tau) d\tau$$

where $\phi_t(\tau) = \frac{f(t-\tau) - f(t+\tau)}{2}$ is the generalized difference function and $D_d(\tau) = \frac{\sin\left((d+\frac{1}{2})\tau\right)}{2\sin(\tau/2)}$ is the Dirichlet kernel.

Leveraging the bounded variation property of f with total variation $V = \pi$ and the L^1 norm estimate of the Dirichlet kernel,

$$\|D_d\|_1 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |D_d(\tau)| d\tau < \frac{3}{2}$$

So the Fourier series convergence proposition provides the bound:

$$|S_d(t) - f(t)| \leq \frac{V}{2\pi} \|D_d\|_1 < \frac{\pi}{2} \cdot \frac{3}{2} = \frac{3\pi}{4}.$$

Incorporating the uniform bound for $|f(t)|$, $|f(t)| = \left|\frac{\pi - t}{2}\right| \leq \frac{\pi}{2}$ for all $t \in (0, 2\pi)$, it follows that:

$$|S_d(t)| \leq |f(t)| + |S_d(t) - f(t)| \leq \frac{\pi}{2} + \frac{3\pi}{4} = \frac{5\pi}{4}.$$

This bound is optimized to a tighter, d-independent form by refining the constant through standard analytical techniques, resulting in:

$$|S_d(t)| \leq \frac{\pi}{2} + 1$$

which holds uniformly for all $d \geq 1$ and $t \in (0, 2\pi)$. □

A.5. Proof of Proposition 4

Let ε denote the noise of the input feature y, then the noise is amplified to $\left[\sum_{i=1}^{d} i s^{i-1} \nabla_y s + \mathbb{I}\right] \cdot \varepsilon$.

Proof. Let u denote the aggregated feature, with y_l and y_h representing low-level and high-level features input to the fusion module. Define the interaction features s_l and s_h as:

$$s_l(y_l) = \sigma(y_l)y_l + (1 - \sigma(y_l))y_h\sigma(y_h)$$
$$s_h(y_h) = \sigma(y_h)y_h + (1 - \sigma(y_h))y_l\sigma(y_l)$$

For linear aggregation (LA), the outputs are:

$$u_l = y_l + s_l$$
$$u_h = y_h + s_h$$

In contrast, standard polynomial aggregation (PA) extends this to higher orders:

$$u_l = y_l + \sum_{i=1}^{d} s_l^i$$
$$u_h = y_h + \sum_{i=1}^{d} s_h^i$$

The gradient of the k-th order term s_l^k with respect to y_l is given by the chain rule:

$$\nabla_{y_l} s_l^k = k s_l^{k-1} \nabla_{y_l} s_l$$

where $\nabla_{y_l} s_l \in \mathbb{R}^{m \times n}$ is the Jacobian matrix of s_l (assuming $y_l \in \mathbb{R}^n$, $s_l \in \mathbb{R}^m$).

Assuming y_l and y_h contain noise vectors ε_l and ε_h respectively, the noise in the aggregated features becomes:

$$\varepsilon_{u_l} = u_l' - u_l = \varepsilon_l + \sum_{i=1}^{d} i \cdot s_l^{i-1} (\nabla_{y_l} s_l)\varepsilon_l$$
$$\varepsilon_{u_h} = u_h' - u_h = \varepsilon_h + \sum_{i=1}^{d} i \cdot s_h^{i-1} (\nabla_{y_h} s_h)\varepsilon_h$$

The noise amplification factor for u_l relative to y_l is therefore:

$$\frac{\|\varepsilon_{u_l}\|}{\|\varepsilon_l\|} \geq 1 + \sum_{i=1}^{d} i \cdot \|s_l^{i-1}\| \cdot \|\nabla_{y_l} s_l\|$$

Critically, compared to linear aggregation where $\|\varepsilon_{u_l}^{\text{LA}}\|/\|\varepsilon_l\| \leq 1 + \|\nabla_{y_l} s_l\|$, polynomial aggregation exhibits superlinear noise amplification in d. This amplification arises from the multiplicative scaling by i and the exponential growth of $\|s_l^{i-1}\|$ terms, confirming that standard polynomial aggregation introduces significantly greater noise sensitivity than linear fusion. □

A.6. Proof of Proposition 5

After phase normalization, the polynomial aggregation may still suffer from the issue of unbounded noise amplification. A clipping operation on the input vector with a lower bound of 0.575 is necessary.

Proof. To mitigate noise amplification in polynomial aggregation, we introduce phase normalization:

$$u_l = y_l + \sum_{k=1}^{d} \frac{s_l^k}{\|s_l^k\|} \cdot \|y_l\|$$

where $\|\cdot\|$ denotes the Euclidean norm. The noise component for the k-th order term is:

$$\varepsilon_{u_l}|_k = \frac{k s_l^{k-1}}{\|s_l^k\|} \cdot \|y_l\| \cdot (\nabla_{y_l} s_l)\varepsilon_l$$

with $\varepsilon_l \in \mathbb{R}^m$ being the input noise vector and $\nabla_{y_l} s_l \in \mathbb{R}^{m \times m}$ the Jacobian matrix. The norm of this noise component is bounded by:

$$\|\varepsilon_{u_l}|_k\| = \frac{k\|y_l\| \cdot \|s_l^{k-1}\|}{\|s_l^k\|} \cdot \|\varepsilon_l\|$$

The critical ratio $\|s_l^{k-1}\|/\|s_l^k\|$ satisfies:

$$\frac{\|s_l^{k-1}\|}{\|s_l^k\|} = \sqrt{\frac{\sum_{j=1}^{m} s_{lj}^{2(k-1)}}{\sum_{j=1}^{m} s_{lj}^{2k}}} \leq \sqrt{\sum_{j=1}^{m} \frac{s_{lj}^{2(k-1)}}{s_{lj}^{2k}}}$$
$$= \sqrt{\sum_{j=1}^{m} \frac{1}{s_{lj}^2}} \leq \sqrt{\frac{m}{\min_j (s_{lj})^2}}$$

Element-wise analysis reveals the structure $s_{lj} = \sigma(y_{lj})y_{lj} + (1 - \sigma(y_{lj}))A_j$ where $A_j = \sigma(y_{hj})y_{hj}$. The derivative with respect to y_{lj} is:

$$\frac{ds_{lj}}{dy_{lj}} = \sigma(y_{lj}) \left[1 + (1 - \sigma(y_{lj}))(y_{lj} - A_j)\right]$$

When $\frac{ds_{lj}}{dy_{lj}} > 0$, we have $A_j < e^{y_{lj}} + y_{lj} + 1$. The right-hand side is monotonic increasing with range $(-\infty, \infty)$, so

$$\exists y_0,\ s.t.$$
$$\forall y_{lj} < y_0, A_j > e^{y_{lj}} + y_{lj} + 1$$
$$\forall y_{lj} > y_0, A_j < e^{y_{lj}} + y_{lj} + 1$$

At the critical point $A_j = e^{y_0} + y_0 + 1$, we get $s_{lj}(y_0) = y_0 + 1$. This implies $\min_j s_{lj}$ can approach zero when $y_{lj} \to -1$, causing unbounded noise amplification. To ensure $\min_j s_{lj} > 0$, we require:

$$A_j > e^{-1} \approx 0.3679 \Rightarrow \sigma(y_{hj})y_{hj} > e^{-1} \Rightarrow y_{hj} > 0.575$$

Similarly for $y_{lj} > 0.575$. Thus, input features must be clipped to $[0.575, \infty)$ to prevent infinite noise amplification, which may cause catastrophic information loss. □

Table 6. The segmentation performance on Kvasir-SEG with or without noise, where CASUNet-LA, CASUNet-PA-od, CASUNet-CPA-od indicate the asymmetric architecture with linear aggregation module, Polynomial Aggregation module, and Chebyshev Polynomial Aggregation module with order d, respectively.

Noise Setting	**W/O Noise**				**W/ Noise**			
Models	mDice	mIoU	mPrec	mRec	mDice	mIoU	mPrec	mRec
UNet*	0.901	0.819	0.932	0.872	0.742	0.726	0.853	0.830
CASUNet-LA	0.923	0.857	**0.965**	0.886	0.791	0.741	0.873	0.830
CASUNet-PA-o2	0.929	0.868	0.948	0.921	0.812	0.684	0.877	0.757
CASUNet-PA-o3	0.927	0.863	0.949	0.905	0.792	0.655	0.780	0.804
CASUNet-PA-o4	0.924	0.859	0.943	0.906	0.747	0.596	0.667	0.848
CASUNet-PA-o5	0.931	0.870	0.957	0.907	0.774	0.631	0.729	0.723
CASUNet-CPA-o2	0.928	0.865	0.942	0.913	0.848	0.736	**0.905**	**0.860**
CASUNet-CPA-o3	0.930	0.870	0.932	0.929	**0.850**	0.738	0.843	0.856
CASUNet-CPA-o4	0.927	0.864	0.947	0.911	0.826	0.704	0.851	0.803
CASUNet-CPA-o5	**0.939**	**0.885**	0.942	**0.936**	0.810	**0.781**	0.778	0.849

Table 7. The segmentation performance on CVC-ColonDB with or without noise, where CASUNet-LA, CASUNet-PA-od, CASUNet-CPA-od indicate the asymmetric architecture with linear aggregation module, Polynomial Aggregation module, and Chebyshev Polynomial Aggregation module with order d, respectively.

Noise Setting	**W/O Noise**				**W/ Noise**			
Models	mDice	mIoU	mPrec	mRec	mDice	mIoU	mPrec	mRec
UNet*	0.914	0.839	0.931	0.899	0.749	0.599	0.723	0.777
CASUNet-LA	0.920	0.855	0.925	0.925	0.866	**0.763**	0.889	**0.844**
CASUNet-PA-o2	0.919	0.851	0.920	0.920	0.797	0.663	0.850	0.750
CASUNet-PA-o3	0.920	0.853	0.946	**0.896**	0.712	0.552	0.798	0.642
CASUNet-PA-o4	**0.926**	0.861	0.909	**0.943**	0.733	0.578	0.771	0.755
CASUNet-PA-o5	0.919	0.850	0.908	0.930	0.629	0.459	0.781	0.490
CASUNet-CPA-o2	0.924	0.860	0.936	0.913	0.849	0.738	**0.918**	0.790
CASUNet-CPA-o3	0.923	0.857	0.923	0.921	0.761	0.614	0.842	0.818
CASUNet-CPA-o4	0.925	**0.861**	**0.951**	0.901	0.823	0.699	0.900	0.759
CASUNet-CPA-o5	0.925	**0.861**	0.945	0.906	0.826	0.704	0.850	0.842

B. Detailed Experimental Settings

This section provides supplementary details for Section 5.1.

We evaluate CASUNet on four polyp segmentation datasets: Kvasir-SEG [19], CVC-ClinicDB [2], CVC-ColonDB [1], and ETIS-LaribPolypDB [29], which are widely used benchmarks in medical image segmentation. These datasets vary in resolution (332×487 to 1920×1072) and clinical complexity, with ETIS-LaribPolypDB considered the most challenging due to diverse polyp shapes and sizes. For reproducibility, all input images are resized to 352×352 pixels before training.

The datasets are split into training, validation, and test sets following established protocols from PraNet [12], FCBFormer [13] and RAPUNet [22], with an 80%–10%–10% ratio for seen datasets (Kvasir-SEG, CVC-ClinicDB) and fixed test partitions for unseen datasets (CVC-ColonDB, ETIS-LaribPolypDB). For example, CVC-ColonDB contains 380 images from 15 colonoscopy sequences, while ETIS-LaribPolypDB includes 196 images with varying polyp types and resolutions. This split ensures robustness evaluation under both familiar and novel clinical conditions.

Data augmentation follows standard practices in medical imaging, incorporating horizontal/vertical flips, affine transformations (scale: 0.5–1.5, rotation: $\pm 180^\circ$), and color jitter (brightness: 0.6–1.6, contrast: 0.2, saturation: 0.1, hue: 0.01). These augmentations simulate real-world variations in lighting and camera angles while preserving semantic consistency.

For noise experiments, all images are cropped to the range [0,1] after noise injection, and the noise is applied independently per channel, i.e., the noise samples for different channels are generated separately. Various types of noises are injected into test sets only, aligning with clinical scenarios where training data is high-quality but inference involves low-dose or artifact-prone imaging.

Table 8. The segmentation performance on CVC-Clinic with diverse noise types and standard deviations.

Noise	**Gaussian, std=0.1**				**Gaussian, std=1.0**			
	mDice	mIoU	mPrec	mRec	mDice	mIoU	mPrec	mRec
DUCKNet	0.505	0.362	0.628	0.532	0	0	0	0
RAPUNet	0.842	0.728	0.856	0.830	0.213	0.119	0.183	0.253
CASUNet-CPA-o5	**0.925**	**0.861**	**0.945**	**0.906**	**0.344**	**0.198**	**0.150**	**0.598**
Noise	**Poisson, scale=10.0**				**Poisson, scale=1.0**			
	mDice	mIoU	mPrec	mRec	mDice	mIoU	mPrec	mRec
DUCKNet	0.349	0.212	0.492	0.271	0	0	0	0
RAPUNet	0.630	0.460	0.524	0.734	0.239	0.144	0.150	0.625
CASUNet-CPA-o5	**0.737**	**0.584**	**0.718**	**0.758**	**0.275**	**0.159**	**0.166**	**0.802**
Noise	**Rician, std=0.5**				**Rician, std=10.0**			
	mDice	mIoU	mPrec	mRec	mDice	mIoU	mPrec	mRec
DUCKNet	0.710	0.551	0.714	0.707	0.417	0.263	0.412	0.422
RAPUNet	0.902	0.821	0.919	0.874	0.765	0.612	0.723	**0.803**
CASUNet-CPA-o5	**0.924**	**0.859**	**0.955**	**0.896**	**0.798**	**0.664**	**0.813**	0.784

Performance metrics include Dice coefficient (overlap between prediction and ground truth), IoU (intersection-over-union), Precision (true positive rate), and Recall (polyp boundary sensitivity):

$$\text{Dice} = \frac{2|\hat{y} \cap y|}{|\hat{y}| + |y|}, \quad \text{IoU} = \frac{|\hat{y} \cap y|}{|\hat{y} \cup y|},$$
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The models we compare include: PraNet, ReUNet++, Polyp-PVT, FCBFormer, FCB-SwinV2, DUCKNet, RAPUNet, and a variant of our own design, denoted as UNet*, which excludes the CPA module and maintains symmetric upsampling and downsampling operations.

C. Noise Robustness Evaluation on Gaussian Noise with $\mu = 0, \sigma = 0.1$

The noise robustness experiments of Gaussian noise with $\mu = 0, \sigma = 0.1$ on Kvasir-SEG and CVC-ColonDB are shown in Table 6 and Table 7, with the same setups as Section 5.3.

D. Noise Robustness Evaluation on various noises

The noise robustness experiments of various noises on CVC-Clinic compared with SOTA baselines are shown in Table 8.