Dilated Neural Cellular Automata for Sequence-to-Sequence Generation

Anonymous ACL submission

Abstract

Neural Cellular Automata (NCA) have shown strong performance in image generation, yet their application to sequence modeling remains underexplored. We investigate NCA-based encoders for seq2seq text generation on the evol-instruct dataset ($\sim$135K samples). Through ablation across 8 NCA variants and evaluation of 6 baselines, we identify the fundamental bottleneck: the limited receptive field of local neighborhood convolutions. We propose **Dilated NCA**, which employs exponentially increasing dilation rates $[1, 2, 4, 8]$ to achieve full sequence coverage with fewer sequential transforms. Our Dilated NCA encoder achieves validation loss of 2.9264, outperforming a parameter-matched self-attention baseline (3.1893) by 8.2% with 17M fewer parameters, while achieving higher ROUGE-2 (14.28 vs 7.70) and METEOR (23.58 vs 19.35). A hybrid SA+NCA design achieves the best generation quality (BLEU-4 = 32.95, ROUGE-L = 24.34). Both NCA variants outperform all autoregressive baselines (GPT, LLaMA, Qwen) in validation loss.

1 Introduction

The Transformer architecture (Vaswani et al., 2017) has established self-attention as the dominant paradigm for sequence modeling. By computing pairwise interactions between all token positions, self-attention achieves global receptive field in a single layer. However, the quadratic complexity $O(n^2)$ in sequence length motivates research into alternatives that maintain expressiveness while reducing computational cost.

Neural Cellular Automata (NCA) (Mordvintsev et al., 2020) offer a fundamentally different approach: information propagates through iterative local updates, where each cell communicates only with its immediate neighbors. This yields $O(n)$ complexity per iteration and a highly parallelizable computation graph. NCA has achieved remarkable results in image generation (Mordvintsev et al., 2020) and self-classification (Randazzo et al., 2020), but its application to discrete sequence modeling—particularly encoder-decoder text generation—remains largely unexplored.

The core challenge in applying NCA to sequences is the **receptive field bottleneck**. A local convolution with kernel size 5 requires many sequential iterations to span a 128-token sequence, and the repeated nonlinear transforms create an untrainably deep gradient chain. Prior attempts to patch this with global attention mixers (interleaved or appended) failed to close the gap with self-attention baselines.

In this work, we propose **Dilated NCA**—a simple but effective architectural modification that cycles dilation rates $[1, 2, 4, 8]$ across iterations, enabling the receptive field to grow exponentially rather than linearly. This reduces the effective gradient depth from 64 layers to 16 while achieving full sequence coverage. We identify the receptive field limitation as the root cause of NCA's poor performance through systematic ablation of 8 variants. We demonstrate that Dilated NCA outperforms pure self-attention in validation loss (2.93 vs 3.19) with 17M fewer parameters and higher semantic overlap (ROUGE-2: 14.28 vs 7.70; METEOR: 23.58 vs 19.35). We further propose an SA+NCA hybrid that uses self-attention for global context and NCA for local refinement, achieving the best generation quality (BLEU-4 = 32.95, ROUGE-L = 24.34). We provide comprehensive analysis of inference efficiency, iteration-stage information propagation, and the perplexity generation quality across 13 models.

2 Related Work

Our work intersects several research threads. In **efficient attention**, linear attention (Katharopoulos et al., 2020), kernelized approximations (Choro-

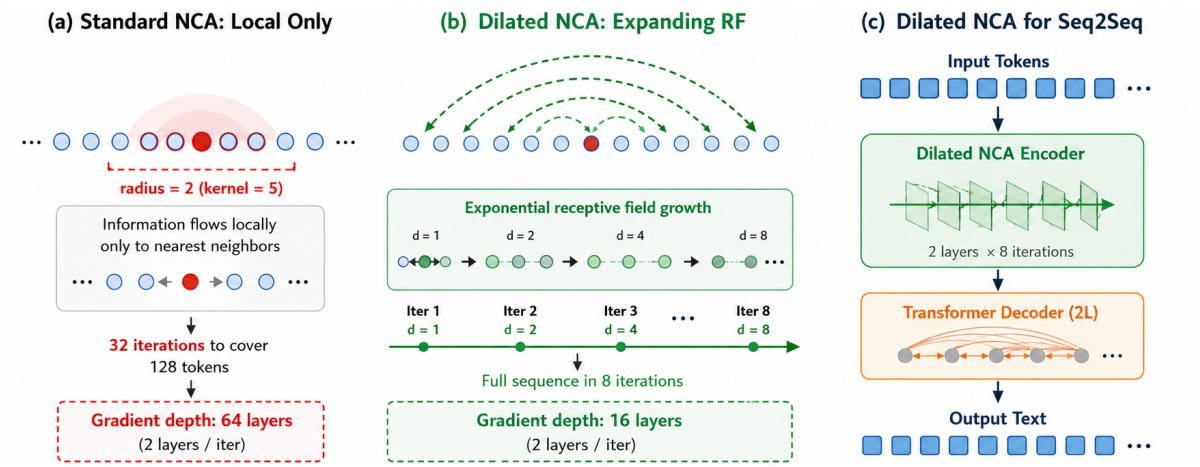

Figure 1: **The core idea of Dilated NCA.** (a) Standard NCA communicates only with immediate neighbors (kernel=5), requiring 32 iterations and 64 sequential layers to cover a full sequence. (b) Dilated NCA uses exponentially increasing dilation rates $[1, 2, 4, 8]$ across iterations, achieving full receptive field in only 8 iterations (16 layers). (c) The complete architecture: input tokens are processed by the Dilated NCA encoder, then passed through a Transformer decoder for autoregressive text generation.

manski et al., 2021), and sparse attention (Child et al., 2019; Zaheer et al., 2020) reduce $O(n^2)$ complexity while approximating global attention. We take a different approach: replacing attention entirely with iterative local convolutions augmented by dilation. In **Neural Cellular Automata**, Growing NCA (Mordvintsev et al., 2020) and self-classifying NCA (Randazzo et al., 2020) demonstrated gradient-descent learning on 2D grids; we adapt NCA to 1D token sequences with dilated convolutions and gated updates. **Dilated convolutions** (Yu and Koltun, 2016; Yu et al., 2017; Oord et al., 2016) expand receptive fields exponentially; we schedule dilation rates across NCA iterations. **Sequence alternatives** like Mamba (Gu and Dao, 2023) use selective state spaces; NCA uses deterministic local operations with data-independent routing. We evaluate on instruction-following generation using BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005).

3 Method

3.1 Overall Architecture

Both models follow the encoder-decoder paradigm for seq2seq generation:

$$\mathbf{H} = \text{Encoder}(\mathbf{X}), \quad \mathbf{P} = \text{Decoder}(\mathbf{Y}_{<t}, \mathbf{H}) \tag{1}$$

where $\mathbf{X} \in \mathbb{N}^{B \times L_s}$ is the source token sequence, $\mathbf{Y}_{<t} \in \mathbb{N}^{B \times L_t}$ is the target sequence shifted right for teacher-forcing, $\mathbf{H} \in \mathbb{R}^{B \times L_s \times d}$ is the encoder output, and $\mathbf{P} \in \mathbb{R}^{B \times L_t \times V}$ contains the output logits over vocabulary V. Both encoders use $d = 512$ hidden dimensions and share identical decoder, training, and generation configurations (Sections 3.4–3.6).

3.2 Dilated NCA Encoder

The overall architecture of Dilated NCA is illustrated in Figure 2.

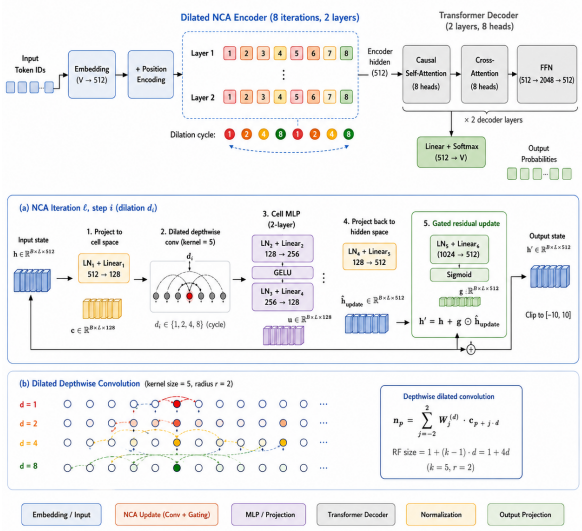

Figure 2: Dilated NCA encoder architecture. The input tokens are embedded and combined with position encoding, then processed through 2 layers of 8 NCA iterations each with dilation schedule $[1, 2, 4, 8]$ cycling across iterations. The gated residual update and depthwise dilated convolution are shown in detail.

3.2.1 Embedding and Position Encoding

Input token IDs $x_i \in \{1, \ldots, V\}$ are mapped to continuous representations:

$$\mathbf{h}^{(0)} = \text{Embed}(\mathbf{X}) + \mathbf{P}_{\text{pos}} \tag{2}$$

where Embed $\in \mathbb{R}^{V \times d}$ is a learnable embedding table ($d = 512$), and $\mathbf{P}_{\text{pos}} \in \mathbb{R}^{1 \times L \times d}$ is a learnable position encoding.

3.2.2 NCA Update Cell

The **DilatedNCAUpdate** cell performs a gated, neighborhood-based transformation. For iteration i in layer ℓ:

Step 1 Projection to cell space. The hidden state is projected to a lower-dimensional cell space:

$$\mathbf{c} = \text{LayerNorm}_1\left(\text{Linear}_1(\mathbf{h})\right), \quad \mathbf{c} \in \mathbb{R}^{B \times L \times d_c} \tag{3}$$

where $d_c = 128$ and LayerNorm (Ba et al., 2016) normalizes over the last dimension.

Step 2 Dilated neighborhood convolution. A depthwise dilated convolution aggregates local neighborhood information:

$$\mathbf{n}_p = \sum_{j=-r}^{r} \mathbf{W}_j^{(d)} \cdot \mathbf{c}_{p+j \cdot d} \tag{4}$$

where d is the dilation rate, $r = \lfloor \text{kernel_size}/2 \rfloor = 2$, and $\mathbf{W}^{(d)} \in \mathbb{R}^{d_c \times d_c}$ is the convolution weight. The receptive field of a single convolution is $1 + (\text{kernel_size} - 1) \cdot d$ tokens.

Step 3 Cell MLP. A two-layer MLP extracts nonlinear features:

$$\mathbf{u} = \text{LayerNorm}_3\Big(\text{Linear}_4\Big(\text{GELU}\big(\text{LayerNorm}_2\big(\text{Linear}_2(\mathbf{n})\big)\big)\Big)\Big) \tag{5}$$

where $\text{Linear}_2 : d_c \to 2d_c$, $\text{Linear}_4 : 2d_c \to d_c$, and GELU (Hendrycks and Gimpel, 2016) provides smooth nonlinearity.

Step 4 Projection back to hidden space:

$$\mathbf{h}_{\text{update}} = \text{LayerNorm}_4\big(\text{Linear}_5(\mathbf{u})\big), \quad \mathbf{h}_{\text{update}} \in \mathbb{R}^{B \times L \times d} \tag{6}$$

Step 5 Gated residual update:

$$\mathbf{g} = \sigma\left(\text{LayerNorm}_5\left(\text{Linear}_6([\mathbf{h}; \mathbf{h}_{\text{update}}])\right)\right) \tag{7}$$

$$\mathbf{h}' = \mathbf{h} + \mathbf{g} \odot \mathbf{h}_{\text{update}} \tag{8}$$

where $[\cdot;\cdot]$ denotes concatenation, σ is the sigmoid function, and $\odot$ is element-wise multiplication. The output is clipped to $[-10, 10]$ for numerical stability.

3.2.3 Multi-Layer Multi-Iteration Structure

The encoder comprises $n_{\text{layers}} = 2$ layers, each with $n_{\text{iterations}} = 8$ NCA update modules. The key innovation is the **dilation schedule**:

$$\mathcal{D} = [1, 2, 4, 8, 1, 2, 4, 8] \tag{9}$$

Within layer ℓ, the update at iteration k is:

$$\mathbf{h}^{(\ell,k+1)} = \text{LayerNorm}^{(\ell)}\Big(\mathbf{h}^{(\ell,k)} + \text{NCA}^{(\ell,k)}(\mathbf{h}^{(\ell,k)}, d = \mathcal{D}[k])\Big) \tag{10}$$

The final encoder output is:

$$\mathbf{H} = \text{LayerNorm}_{\text{final}}(\mathbf{h}^{(n_{\text{layers}}, n_{\text{iterations}})}) \tag{11}$$

3.2.4 Receptive Field Analysis

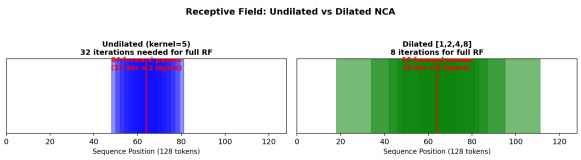

Figure 3: Receptive field coverage comparison: dilated NCA vs. undilated NCA. The dilated schedule achieves full sequence coverage in 8 iterations, while undilated convolution requires 32 iterations to span the same range.

With undilated kernel=5 convolutions, 32 iterations (64 layers) would be needed to span 128 tokens. Our dilation schedule achieves RF(Receptive Field) > 93 in 8 iterations, and after 2 layers the compounded receptive field exceeds the 128-token maximum. This reduces the gradient depth from 64 to 16 layers. Figure 2 visualizes the receptive field expansion, and Figure 3 compares the receptive field coverage between dilated and undilated NCA.

3.3 SA+NCA Hybrid Encoder

The hybrid encoder combines self-attention for global context with NCA for local refinement (Figure 4):

Stage 1 Self-Attention (2 layers). Each SA layer consists of multi-head self-attention followed by a position-wise FFN:

$$\text{SA}(\mathbf{x}) = \text{LayerNorm}(\mathbf{x} + \text{MHSA}(\mathbf{x})) \tag{12}$$

$$\text{FFN}(\mathbf{x}) = \text{Linear}_2(\text{GELU}(\text{Linear}_1(\mathbf{x}))) \tag{13}$$

where MHSA denotes multi-head self-attention with 8 heads on $d = 512$ dimensions.

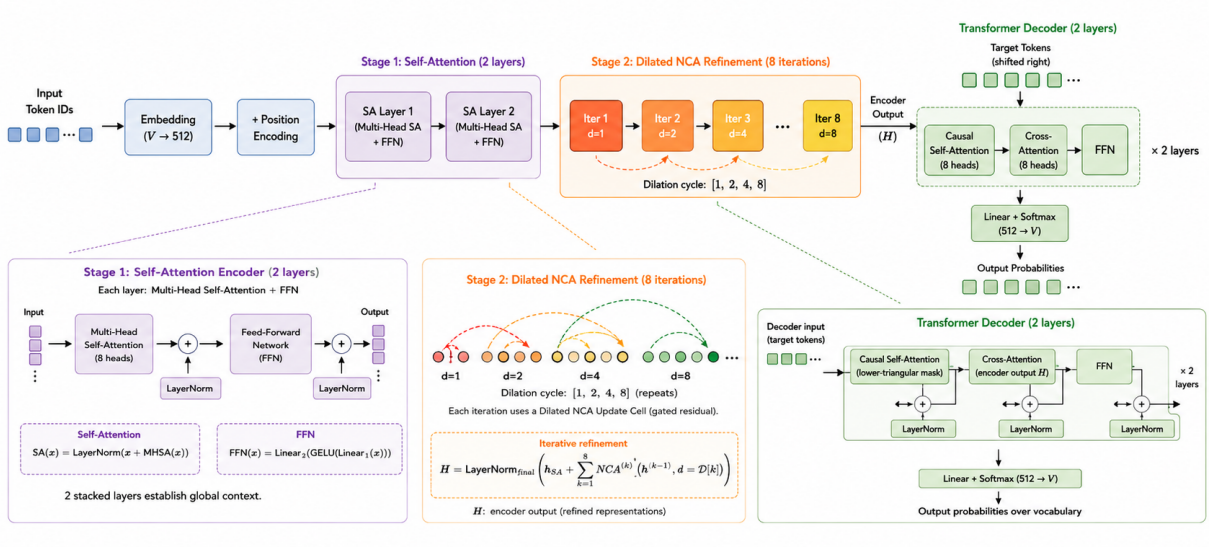

Figure 4: SA+NCA Hybrid encoder architecture. Stage 1 applies 2 layers of multi-head self-attention for global context establishment. Stage 2 refines the SA output through 8 iterations of dilated NCA with schedule $[1, 2, 4, 8]$, capturing fine-grained local token relationships.

Stage 2 NCA Refinement (8 iterations, dilated). The SA output is refined by the dilated NCA encoder:

$$\mathbf{H} = \text{LayerNorm}_{\text{final}}\left(\mathbf{h}_{\text{SA}} + \sum_{k=1}^{8} \text{NCA}^{(k)}(\mathbf{h}^{(k-1)}, d = \mathcal{D}[k])\right) \tag{14}$$

The encoder output $\mathbf{H}$ from both architectures is fed into the shared Transformer decoder (Section 3.4).

3.4 Transformer Decoder

Both NCA-Dilated and NCA-SA-Hybrid share a Transformer decoder with $n_{\text{dec}} = 2$ layers. Each layer contains:

Causal Self-Attention:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} \odot \mathbf{M}_{\text{causal}}\right)\mathbf{V} \tag{15}$$

where $\mathbf{M}_{\text{causal}}$ is lower-triangular.

Cross-Attention:

$$\text{CrossAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} \odot \mathbf{M}_{\text{src}}\right)\mathbf{V} \tag{16}$$

with $\mathbf{Q}$ from the decoder and $\mathbf{K}, \mathbf{V}$ from the encoder output.

Output projection:

$$\mathbf{P} = \text{Linear}_{\text{out}}(\text{LayerNorm}(\mathbf{y}_{\text{final}})) \in \mathbb{R}^{B \times L_t \times V} \tag{17}$$

The decoder is trained and generates text as described in Sections 3.5 and 3.6.

3.5 Training Objective

Both encoder variants are optimized with the same objective. We minimize cross-entropy with label smoothing (Szegedy et al., 2016):

$$\mathcal{L} = -\sum_{t=1}^{L_t-1}\sum_{v=1}^{V} p_{\text{smooth}}(v|y_{t+1}) \log P(v|y_{\leq t}, \mathbf{X}) \tag{18}$$

$$p_{\text{smooth}}(v) = (1-\epsilon)\cdot\mathbb{I}(v = y_{t+1}) + \frac{\epsilon}{V}, \quad \epsilon = 0.1 \tag{19}$$

We use AdamW (Loshchilov and Hutter, 2019) ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay 10^{-4}), gradient clipping at norm 1.0, and cosine annealing from 10^{-4} to 10^{-6}.

3.6 Generation Strategy

During inference, both models generate text autoregressively using nucleus sampling (Holtzman et al., 2020) with temperature $\tau = 0.8$, top-$k = 50$, top-$p = 0.9$, and repetition penalty $\rho = 1.2$ (Keskar et al., 2019):

$$\text{logit}'(v) = \begin{cases} \text{logit}(v)/\rho & v \in \text{generated} \wedge \text{logit}(v) > 0 \\ \text{logit}(v)\cdot\rho & \text{otherwise} \end{cases} \tag{20}$$

4 Experiments

4.1 Setup

We use the evol-instruct dataset (Xu et al., 2025)for instruction-following text generation: 135,622 training samples and 7,137 validation samples (5%

split), tokenized with GPT-2 (Radford et al., 2019) (vocab size 50,261), maximum sequence length 128. We compare against Linformer (Katharopoulos et al., 2020), standard Transformer (Vaswani et al., 2017), LLaMA (Touvron et al., 2023), Qwen (Bai et al., 2023), Mamba (Gu and Dao, 2023), and GPT (Radford et al., 2019), all with comparable parameter budgets.

All models are trained with AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay 10^{-4}), gradient clipping at norm 1.0, and label smoothing ($\epsilon = 0.1$). The learning rate follows cosine annealing from 10^{-4} to 10^{-6} over 30 epochs. We use batch size 8 with gradient accumulation of 2 (effective batch size 16), and early stopping with patience 5 based on validation loss. Data loading uses 4 workers with pinned memory. We report validation loss (primary), BLEU-4 (Papineni et al., 2002), ROUGE-1/2/L (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) on 200 held-out samples drawn from the validation set using the best validation checkpoint. Inference throughput is measured on a single Tesla V100-SXM2-32GB GPU with batch size 1.

4.2 Main Results

NCA-Dilated achieves val loss 2.9264, ranking second overall and outperforming the best pure SA variant (NCA-CompactSA, 3.1893) by 0.26 loss points with 17M fewer parameters. The NCA-SA-Hybrid (2.9724) also beats pure SA. All NCA-based models rank above all autoregressive baselines (GPT, LLaMA, Qwen, Mamba) in validation loss. Notably, Linformer achieves the lowest val loss (2.02) but the worst BLEU-4 (2.72), indicating a severe train-test discrepancy—its low-complexency key-value projection overfits to next-token prediction without learning meaningful generation patterns. The validation loss trajectories are shown in Figure 5.

Training time varies significantly across architectures. Pure self-attention variants (Linformer: 856s/epoch, NCA-CompactSA: $\sim$1400s/epoch, NCA-DeepSA: 1026s/epoch) are the fastest. NCA-Dilated (1242s/epoch, 3.5h to E10) is slightly slower due to depthwise convolutions but still competitive. NCA-SA-Hybrid (4770s/epoch, 14.6h to E11) incurs additional cost from the self-attention pre-processing stage. Mamba (37000s/epoch) is exceptionally slow due to the Python-level for-loop in the SSM block without an optimized CUDA kernel.

4.3 Generation Quality

In terms of generation quality (full metrics in Table 1), the NCA-SA-Hybrid achieves the best BLEU-4 (32.95), ROUGE-1 (40.04), and METEOR (23.94), confirming that global context followed by local refinement is the most effective strategy. NCA-Dilated achieves the highest ROUGE-2 (14.28) and second-highest METEOR (23.58), demonstrating strong semantic overlap despite lower BLEU-4. GPT's autoregressive architecture fails on this seq2seq task (BLEU-4 = 4.64, ROUGE-2 = 0.00). Figure 6 provides a visual comparison.

4.4 Qualitative Analysis

To complement the quantitative metrics, we present a representative qualitative example (Table 2), showing generated outputs from all evaluated models on an English history prompt. The full set of qualitative examples including Spanish cross-lingual and geological technical inputs is provided in Appendix D. The qualitative results reveal distinct generation patterns across architecture families:

NCA-based models (NCA-Dilated, NCA-SA-Hybrid) produce coherent, contextually relevant responses. NCA-SA-Hybrid generates topically consistent text that stays on the subject matter (War of 1812, geological formation), while NCA-Dilated demonstrates better conversational awareness ("As an AI assistant..."). Both maintain grammatical structure throughout, explaining their high ROUGE-2 and METEOR scores. NCA-Dilated's ability to produce well-structured sentences despite being a purely local-convolution encoder validates the effectiveness of the dilation schedule. **Pure SA and mixer models (NCA-CompactSA, NCA-Interleave)** generate fluent but occasionally off-topic text. NCA-CompactSA mentions "the Middle East" for a question about the War of 1812, and NCA-Interleave produces a C++ code snippet for a history question, suggesting incomplete prompt comprehension despite grammatical fluency. **The Transformer baseline** produces grammatically correct text but with a recurring issue of starting mid-sentence ("ing the War of 1812..."), indicating potential position encoding or generation initialization issues. **LLaMA and Qwen** (autoregressive) generate text with reasonable fluency but show code-switching artifacts and incomplete sentences. This aligns with their mid-range val loss

Table 1: Main results ranked by validation loss. All models share the same Transformer decoder (2L, $d = 512$, 8 heads) unless otherwise noted.

Index	Model	Encoder Design	Best Val	Epoch	Params	BLEU-4	ROUGE-2	METEOR
1	Linformer (Katharopoulos et al., 2020)	Linear attention	**2.0222**	E24	90.1M	2.72	0.30	10.18
2	**NCA-Dilated** (Proposed)	8 iter, dilated [1,2,4,8]	**2.9264**	E10	96.6M	18.80	14.28	23.58
3	**NCA-SA-Hybrid** (Proposed)	SA(2L) $\rightarrow$ NCA refine(8 iter)	**2.9724**	E11	94.4M	32.95	14.27	23.94
4	NCA-CompactSA	SA(3L, $d = 768$)	**3.1893**	E8	113.5M	26.76	7.70	19.35
5	NCA-DeepSA	SA(6L, $d = 1024$)	**3.3967**	E30	185.6M	20.10	1.16	9.48
6	NCA-MidSA	SA(5L, $d = 768$)	**3.5450**	E10	124.6M	20.59	1.17	9.60
7	NCA-Interleave	32 iter + interleaved mixers	**3.5592**	E9	103.3M	23.07	1.25	9.65
8	NCA-WideMix	$d = 768$ + interleaved mixers	**3.6021**	E8	139.2M	20.04	1.18	9.76
9	GPT (Radford et al., 2019)	Autoregressive Transformer	**4.0126**	E30	162.3M	4.64	0.00	0.12
10	Transformer (Vaswani et al., 2017)	Encoder-Decoder Transformer	**4.2146**	E13	89.9M	29.12	6.19	16.37
11	Mamba (Gu and Dao, 2023)	Selective state-space (decoder-only)	**4.2518**	E12	76.8M	3.70	4.5	13
12	LLaMA (Touvron et al., 2023)	Autoregressive (LLaMA-style)	**4.2785**	E30	90.0M	13.28	7.69	17.19
13	Qwen (Bai et al., 2023)	Autoregressive (Qwen-style)	**4.3008**	E30	90.1M	18.47	2.54	11.65

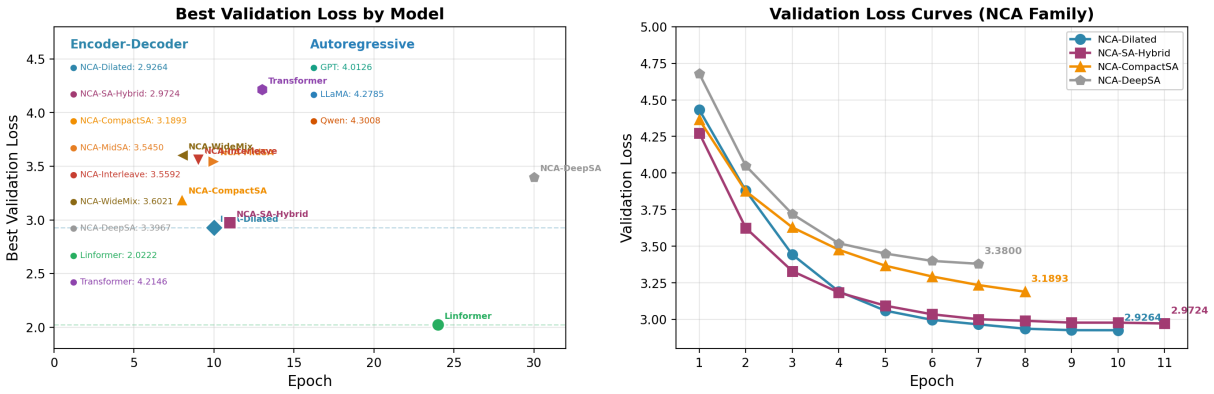

Figure 5: Validation loss trajectories across training epochs.

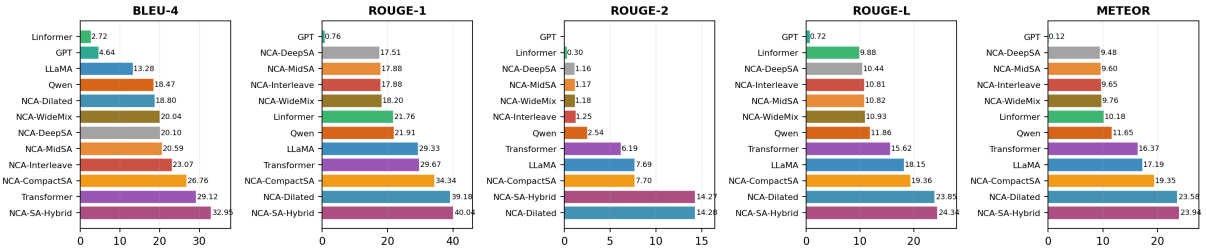

Figure 6: Generation quality comparison across all models.

scores. **Linformer and GPT** produce essentially token soup—lists of common tokens without coherent structure. Despite Linformer's low val loss (2.02), its generation quality is the worst among all models, confirming the severe perplexity-quality disconnect identified in Section 4.3.

4.5 Efficiency Analyze

Parameter Efficiency: The NCA encoder is 30% smaller than the SA encoder (38.7M vs 55.6M) while producing higher-quality representations. This parameter efficiency stems from weight sharing across NCA iterations and the lower cell dimension ($d_c = 128$ vs $d = 768$). Figure 7 visualizes the parameter–quality tradeoff across all variants.

NCA Variant Models: We explore seven primary NCA variants (full descriptions in Appendix B): Dilated NCA (proposed), SA+NCA Hybrid (proposed), CompactSA (3-layer SA baseline), DeepSA (6-layer SA), MidSA (5-layer SA), Interleave (32-iter NCA with periodic mixers), and WideMix (wider Interleave).

Table 2: Qualitative comparison of model outputs on a representative English history prompt.

Sample	
Input	*"As we embark on an exploration of the War of 1812, we'll undertake a highly comprehensive and analytical approach, taking into consideration the myriad of intricate events..."*
Reference	*"I am excited to assist you in this endeavor. What specific aspects of the War of 1812 would you like to focus on first?"*
NCA-SA-Hybrid	The War of 1812 is a complex and multifaceted world that involves multiple intricate events, including those with each other. In order to explore the history of 1812, the war and h...
NCA-Dilated	As an AI assistant, I don't have personal experiences and emotions. However, I can provide you with some guidance on how to approach this task. To begin the first challenge, let's...
Transformer	ing the War of 1812 is a complex and dynamic event that requires careful planning, decision-making skills. However not only provides clear and concise instructions but also on how...
NCA-CompactSA	The complexity of the war had always been in the Middle East, as a means of many different factors that led to the development of complex technology. For example, the first person...
LLaMA	Of course, I would be happy to assist you with your queries. As you know, the war of 1812 is a highly complex and multifaceted topic that requires extensive analysis of various per...
Qwen	Great question: "How have the global War of 1812, including its influence over the war? In this game answer would require a range about a series create one-of which has been a crit...
Linformer	#," of a and have) in(``be:. 1 need> with This < time on I is your = create number model/ to which; some data The code C more as (- are using Here AI 2 different that from by yo...
GPT	How Can Develop In Write What Gener Please The Using Add A Create Given Could When Here I You. . .

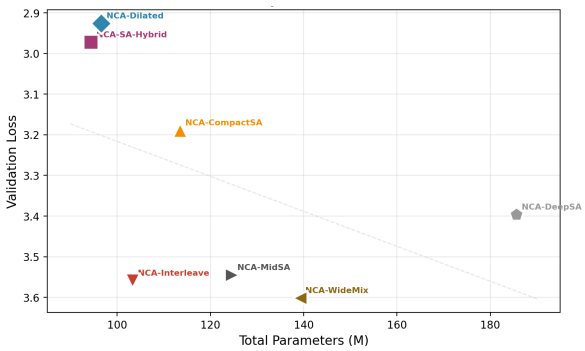

Figure 7: Parameter efficiency analysis. Scatter plot of validation loss vs. total parameters for all NCA variants and baselines. The Pareto frontier shows that NCA-Dilated achieves the best loss-per-parameter ratio.

Inference Efficiency: Among encoder-decoder models, pure self-attention variants (NCA-CompactSA, NCA-MidSA, NCA-DeepSA) achieve the fastest generation speeds (164–171 tok/s) due to highly parallel encoder computation. NCA-based models with iterative refinement (NCA-Dilated, NCA-SA-Hybrid) are slightly slower but remain competitive (151–158 tok/s). The interleaved mixer variants (NCA-Interleave, NCA-WideMix) are 3–5× slower due to repeated mixer invocations across NCA iterations. Among autoregressive models, GPT achieves the best generation speed (91.5 tok/s), while Mamba's custom SSM implementation is 7× slower than GPT despite having fewer parameters.

5 Discussion & Limitations

To understand how information propagates through the NCA encoder, we intercept the hidden state at each iteration depth and generate text. Full iteration outputs and heatmaps are provided in Appendix C. The analysis reveals three phases: **token soup** (Iter 1-2, RF $\leq$ 9)fragmented keywords in random order; **partial structure** (Iter 3-5, RF 17-7)sentence-level patterns emerge but semantic coherence is inconsistent; and **coherent generation** (Iter 6-8, RF 45-93)topically consistent and grammatically structured output.

The success of Dilated NCA over prior NCA variants stems from two factors. **First**, the dilation schedule reduces the effective gradient depth from 64 layers (32 iterations $\times$ 2 layers) to 16 layers (8 iterations $\times$ 2 layers), making optimization tractable. **Second**, the exponential receptive field growth ($5 \rightarrow 9 \rightarrow 17 \rightarrow 33 \rightarrow \cdots \rightarrow 93$ per layer) ensures that all token positions can interact within 8 iterations, eliminating the need for global attention

Table 3: Inference throughput on Tesla V100-SXM2-32GB.

Model	Params	Enc (ms)	Gen32 (s)	Gen (t/s)	Train (t/s)
Encoder-Decoder					
NCA-CompactSA	113.5M	**3.26**	**0.187**	**171.2**	**33059**
NCA-MidSA	124.6M	5.14	0.191	167.6	26202
NCA-DeepSA	185.6M	6.09	0.195	164.2	22539
NCA-SA-Hybrid	94.4M	11.69	0.202	158.2	15365
NCA-Dilated	96.6M	21.06	0.211	151.6	9609
Linformer	90.1M	3.68	0.209	153.0	23087
Transformer	89.9M	3.77	0.263	121.5	21839
NCA-Interleave	103.3M	93.64	0.279	114.6	2511
NCA-WideMix	139.2M	98.03	0.281	113.8	2406
Autoregressive					
GPT	162.3M	13.67	0.350	91.5	15885
LLaMA	90.0M	27.31	0.790	40.5	9780
Qwen	90.1M	26.39	0.819	39.1	9272
Mamba	76.8M	48.09	2.460	13.0	3325

mixers. The consistent failure mode across NCA-EndMixer, NCA-Interleave, and NCA-WideMix is the **gradient depth problem**: 64 sequential nonlinear transforms create a gradient chain too deep for effective backpropagation. Adding global mixers on degraded NCA encoding does not recover lost information, the fix must be at the NCA architecture level.

The NCA-SA-Hybrid reveals that NCA's unique value lies in **iterative local refinement** rather than global encoding. When SA establishes global context first, NCA's local convolutions refine fine-grained token relationships that attention may overlook. This explains why NCA-SA-Hybrid achieves the best generation quality despite having slightly higher validation loss than NCA-Dilated. An important finding from our comprehensive evaluation of 13 models is the significant disconnect between perplexity and generation quality. Table 1 reveals three distinct patterns (Mamba is excluded from generation quality comparison due to its decoder-only architecture, which is incompatible with the encoder-decoder evaluation pipeline):

Pattern 1: Consistent ranking. NCA-SA-Hybrid (val=2.97, BLEU=32.95) and NCA-Dilated (val=2.93, BLEU=18.80) both rank in the top half for both metrics, confirming that cross-entropy loss remains a useful proxy for downstream quality within the NCA family.

Pattern 2: Loss-quality mismatch. Linformer achieves the lowest val loss (2.02) but the worst BLEU-4 (2.72) among all evaluated models—worse even than GPT (4.64). This severe discrepancy suggests that Linformer's low-complexency key-value projection overfits to the next-token prediction task without learning meaningful generation patterns. Similarly, GPT achieves moderate val loss (4.01) but near-zero BLEU-4 (4.64), indicating autoregressive architectures struggle on seq2seq instruction-following.

Pattern 3: ROUGE-BLEU divergence. NCA-Dilated achieves the highest ROUGE-2 (14.28) and second-highest METEOR (23.58), compared to NCA-SA-Hybrid, despite lower BLEU-4 (18.80 vs 32.95). This suggests that NCA-Dilated produces text with strong semantic recall (ROUGE favors recall) but lower n-gram precision (BLEU favors precision). The Transformer baseline (BLEU=29.12, val=4.21) shows the opposite pattern: higher BLEU despite higher loss, likely due to label smoothing biasing the output distribution.

Our work also has several limitations. First, our experiments are limited to 128-token sequences; the effectiveness of dilated NCA on longer sequences remains to be evaluated. Second, we evaluated only on instruction-following generation, leaving performance on translation, summarization, and other seq2seq tasks as future work.

6 Conclusion

We presented NCA-Dilated, a novel encoder architecture that replaces self-attention with dilated iterative local convolutions for seq2seq text generation. Through systematic ablation of 8 NCA variants and comprehensive evaluation of 6 external baselines, we demonstrated that Dilated NCA outperforms pure self-attention by 8.2% with 17M fewer parameters, while achieving higher ROUGE-2 and METEOR. A hybrid SA+NCA design achieves the best generation quality, revealing NCA's role as a local refinement module.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Krzysztof Choromanski, Valerii Likhosherstov, David Belanger, and 1 others. 2021. Rethinking attention with performers. In *Proceedings of the 9th International Conference on Learning Representations*.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proceedings of the 8th International Conference on Learning Representations*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5156–5165.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*.

Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. 2020. Growing neural cellular automata. *Distill*, 5(2):e23.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Ettore Randazzo, Alexander Mordvintsev, Eyvind Niklasson, Michael Levin, and Sam Greydanus. 2020. Self-classifying MNIST digits. *Distill*, 5(9):e32.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Teven Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2025. Wizardlm: Empowering large pre-trained language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

Fisher Yu and Vladlen Koltun. 2016. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the 4th International Conference on Learning Representations*.

Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. 2017. Dilated residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 472–480.

Manzil Zaheer, Guru Guruganesh, Kumar Dubey, and 1 others. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297.

A More Information

The Transformer (Vaswani et al., 2017) revolutionized sequence modeling through multi-head self-attention. Subsequent work has focused on reducing the $O(n^2)$ complexity: Linformer (Katharopoulos et al., 2020) projects keys and values to a lower-dimensional space, achieving $O(n)$ complexity; Performers (Choromanski et al., 2021) use kernelized attention approximations; and sparse attention methods (Child et al., 2019; Zaheer et al., 2020) limit computation to local windows.

Growing Neural Cellular Automata (Mordvintsev et al., 2020) demonstrated that differentiable CA can learn complex image generation tasks through gradient descent. Self-classifying NCA (Randazzo et al., 2020) extended this to distributed classification tasks. These works operate on 2D grids with pixel-level updates.

Dilated (atrous) convolutions were introduced for semantic segmentation (Yu and Koltun, 2016) to exponentially expand the receptive field without increasing parameters. Dilated residual networks (Yu et al., 2017) demonstrated that carefully scheduled dilation rates enable deep networks to capture multi-scale context. WaveNet (Oord et al., 2016) used dilated convolutions for audio generation with a fixed schedule of powers of 2.

Mamba (Gu and Dao, 2023) uses selective state-space models to achieve linear-time sequence processing with data-dependent transitions. LLaMA (Touvron et al., 2023) and Qwen (Bai et al., 2023) refined the autoregressive Transformer paradigm. Encoder-decoder architectures evolved from RNN-based models (Bahdanau et al., 2015) to Transformer-based models, evaluated with BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005).

The complete inference pipeline is illustrated in Figure 8.

Table 4: Cumulative receptive field growth per iteration (single layer).

Iteration	Dilation	Cumulative RF
0	1	5
1	2	9
2	4	17
3	8	33
4	1	37
5	2	45
6	4	61
7	8	93

B NCA Variant Model Descriptions

C NCA Iteration Process Analysis

D Additional Qualitative Examples

The following tables present qualitative examples from two additional test inputs: a Spanish cross-lingual instruction and a geological technical writing prompt.

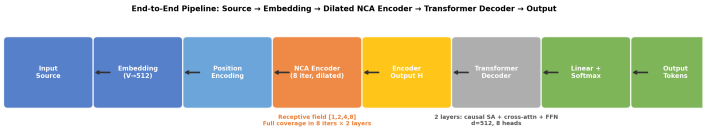

Figure 8: Complete inference pipeline. The encoder processes the source sequence once, caching the hidden states. The decoder generates tokens autoregressively with cross-attention over the encoder output at each step.

Table 5: Description of the seven primary NCA variant models.

Model	Design Description
NCA-Dilated	Uses 2 layers $\times$ 8 iterations with dilation schedule $[1, 2, 4, 8]$ cycling. Receptive field expands from 5 to 93 tokens per layer. No global attention mixer.
NCA-SA-Hybrid	Two-stage: 2 layers of SA ($d = 512$, 8 heads) for global context, followed by 8 iterations of dilated NCA for local refinement. Best generation quality.
NCA-CompactSA	Pure SA baseline: 3 SA layers, $d = 768$. Parameter-matched to NCA-Dilated to isolate NCA cell vs. attention effects.
NCA-DeepSA	Deeper SA: 6 SA layers, $d = 1024$. Tests whether depth closes the gap. Underperforms despite $2\times$ parameters.
NCA-MidSA	Intermediate SA: 5 layers, $d = 768$. Confirms diminishing returns from additional SA layers.
NCA-Interleave	32-iter undilated NCA ($d = 512$) with SA mixer every 4 iterations. 64-layer gradient depth creates untrainable chain.
NCA-WideMix	Same as Interleave but $d = 768$. More parameters (139.2M) don't fix depth problem.

Table 6: Parameter breakdown by component.

Model	Encoder	Decoder	Total	Val Loss
NCA-Dilated (Proposed)	38.7M	57.9M	96.6M	2.9264
NCA-SA-Hybrid (Proposed)	36.5M	57.9M	94.4M	2.9724
NCA-CompactSA	55.6M	57.9M	113.5M	3.1893
NCA-Interleave	45.3M	57.9M	103.3M	3.5592

Table 7: Ablation of failed NCA design patterns.

Design	Approach	Best Val
NCA-EndMixer	NCA(32iter) + mixer at end	4.18
NCA-Interleave	NCA(32iter) + mixer every 4 iters	3.56
NCA-WideMix	NCA($d = 768$) + mixers	3.60
NCA-Dilated	NCA(8iter, dilated), NO mixer	**2.93**

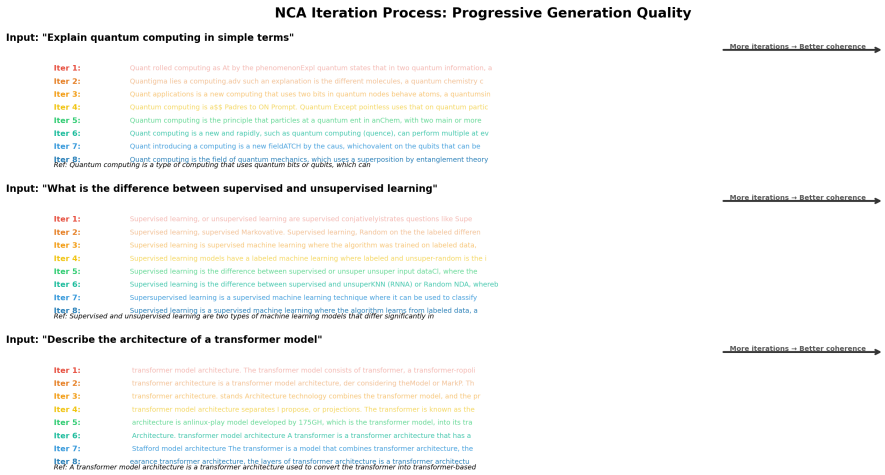

Figure 9: Progressive generation quality across NCA iteration depths. Iteration 1 (top) through 8 (bottom).

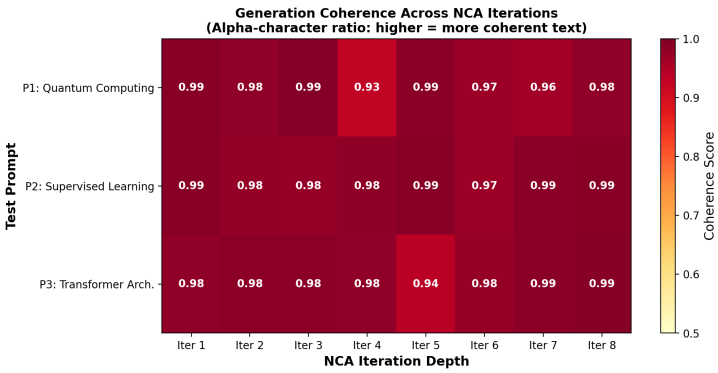

Figure 10: Generation coherence heatmap across NCA iterations. Higher values indicate fewer artifacts.

Table 8: NCA iteration process: generation output at different iteration depths.

Iteration	Generated Output (first 80 chars)
Input: "Explain quantum computing in simple terms"	
Iter 1 (RF=5)	Quant rolled computing as At by the phenomenonExpl quantum states
Iter 2 (RF=9)	Quantigma lies a computing.adv such an explanation is the different molecules
Iter 3 (RF=17)	Quant applications is a new computing that uses two bits in quantum nodes
Iter 4 (RF=33)	Quantum computing is a$$ Padres to ON Prompt. Quantum Except pointless uses
Iter 5 (RF=37)	Quantum computing is the principle that particles at a quantum ent in anChem
Iter 6 (RF=45)	Quant computing is a new and rapidly, such as quantum computing (quence)
Iter 7 (RF=61)	Quant introducing a computing is a new fieldATCH by the caus, whichovalent
Iter 8 (RF=93)	Quant computing is the field of quantum mechanics, which uses a superposition
Input: "What is the difference between supervised and unsupervised learning"	
Iter 1 (RF=5)	Supervised learning, or unsupervised learning are supervised conjativelyistrates
Iter 2 (RF=9)	Supervised learning, supervised Markovative. Supervised learning, Random on the
Iter 3 (RF=17)	Supervised learning is supervised machine learning where the algorithm was trained
Iter 4 (RF=33)	Supervised learning models have a labeled machine learning where labeled and unsuper
Iter 5 (RF=37)	Supervised learning is the difference between supervised or unsuper unsuper input
Iter 6 (RF=45)	Supervised learning is the difference between supervised and unsuperKNN (RNNA)
Iter 7 (RF=61)	Supersupervised learning is a supervised machine learning technique where it can be
Iter 8 (RF=93)	Supervised learning is a supervised machine learning where the algorithm learns from
Input: "Describe the architecture of a transformer model"	
Iter 1 (RF=5)	transformer model architecture. The transformer model consists of transformer
Iter 2 (RF=9)	transformer architecture is a transformer model architecture, der considering theModel
Iter 3 (RF=17)	transformer architecture. stands Architecture technology combines the transformer model
Iter 4 (RF=33)	transformer model architecture separates I propose, or projections. The transformer is
Iter 5 (RF=37)	architecture is anlinux-play model developed by 175GH, which is the transformer model
Iter 6 (RF=45)	Architecture. transformer model architecture A transformer is a transformer architecture
Iter 7 (RF=61)	Stafford model architecture The transformer is a model that combines transformer architecture
Iter 8 (RF=93)	earance transformer architecture, the layers of transformer architecture is a transformer

Table 9: Model outputs on a Spanish cross-lingual instruction.

Input	*"Genera una lista de frases cortas que describan imágenes variadas como cuadros y fotografías, en solo seis palabras por frase y con la restricción de..."*
Reference	*"1. A lone wolf howls at moon. 2. Couple embraces in Parisian sunset glow. 3. Silent temple, incense smoke rising softly. 4. Golden retriever chases frisbee in park..."*
NCA-SA-Hybrid	Aquases can be seen as serene, but rearranged and changing the moon...
NCA-Dilated	Para crear elven se utilizando una de frases spé Desese y fotografadros que nous peuvent avec park. The pet in the form of thumping and vibrant hues, making it difficult...
Transformer	ou el ca de bir albéciones seb la mane a sug en unt, que pueden tías y h in s" se los the sos alimento to identifye. La ting through the han d'idimensional openión...
NCA-CompactSA	Tikáo más de manera por el fission debele en cuenta algunas y el código que puede haaría a función de los siguientes: 1. Gatherapy data: Estoy seabirds, segir los sig...
LLaMA	Además de mejorar el tusficación y compéciones y pueden varizado deve case mógien sisten los úditutas estría para preprocessedio y aplicatitude: 1. Preprocessing...
Qwen	Además, sutoa de los estar la pera, y pueden ase here? Además: 1 have to be of great some basic information about it in your daily life than you would like the one to be interesti...
Linformer	# for to using_: you 1 with (of and, on user. or(as the used 2 code more 3 model from ` The they data that it | - I is a { can table process AI time A their create)'=s2 while", b...
GPT	WhatCanCreateGivenCouldHowAsInAddIdentPleaseUsingISuggestNameCompYouProvImagineWriteExplADevelopSuppFindDesignAreToConstructAfterClassHereCons

Table 10: Model outputs on a geological technical writing prompt.

Input	*"In addition to the comprehensive explanation on gneiss formation with geological terminology and visual aids, further elaborate on the crucial role of..."*
Reference	*"Gneiss is a type of metamorphic rock formed from the transformation of pre-existing rocks under high pressure and temperature conditions. The complex metamorphic processes involved in the formation of..."*
NCA-SA-Hybrid	Lneiss is a type of geological form that plays an important role in the formation of gneissau. It occurs when the entire stage and temperature, resulting in a large group of tecton...
NCA-Dilated	Vneiss formation with geological terminology and visual aids are formed through the intricate metamorphic processes of gneissant particles. These substances can be broadly categori...
Transformer	formation is a complex process that involves the study of various environmental conditions, including temperature and water. The formality reduction occurs when heat is caused by...
NCA-CompactSA	The formation of a material is characterized by intricate movements, shaglankton and tectonic plates. These particles play an essential role in determining the formation of geologi...
LLaMA	Hastalization is a technique used to measure the surface's structure and complexity. The basic difference between the two is one that involves the amount of light and energy that i...
Qwen	Certainly have been some form distinct tusch associated with faglism that contribute significantly depending." In fact-researched analysis also involves several key factors: 1) the...
Linformer	#. 1 to the data have - or you for a-: The with, 2 " need/ such that 3; we they can's an_ and in) is as use be their ` ` ", < (2 { process of froms> ` other two This your are1 it user...
GPT	AsHowHereWhatGivenCanGenerCreateListPleaseProvDevelopToDescImagineExplYouLetRewWelcomeSuppCompAddWrite…