applied sciences

MDPI

Article

Beyond Perplexity: A Multi-Faceted Analysis of a Novel Densely Connected Transformer

Enrico De Santis [1], Alessio Martino [2,*] and Antonello Rizzi [1]

[1] Department of Information Engineering, Electronics and Telecommunications, University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy; enrico.desantis@uniroma1.it (E.D.S.); antonello.rizzi@uniroma1.it (A.R.)
[2] Department of AI, Data and Decision Sciences, LUISS University, Viale Romania 32, 00197 Rome, Italy
* Correspondence: amartino@luiss.it; Tel.: +39-06-85225957

Abstract

Background: Dense cross-layer connectivity can shorten gradient paths and promote feature reuse, potentially improving optimization under fixed training budgets. **Objective:** We test whether concatenation-based dense historical connectivity improves decoder-only autoregressive language modeling under controlled comparison protocols. **Methods:** We compare a standard Transformer decoder and a dense decoder on Penn Treebank and WikiText-2 under two fairness regimes: (i) a *same training recipe* setting with a fixed baseline and a bounded dense architectural search, and (ii) a *same parameter budget* setting where the dense model is resized to not exceed the baseline parameter count. **Results:** Dense connectivity does not consistently reduce test perplexity; on WikiText-2, the baseline remains better in both regimes, while gains on Penn Treebank are small and regime-dependent. Ablations within the dense family show that depth and feed-forward capacity are the most reliable drivers of perplexity improvements. **Conclusions:** Probes and attention diagnostics do not reveal a clear advantage for dense connectivity in our limited probe set, while Zipf–RQA analysis of long-form generations reveals systematic structural differences between baseline and dense outputs. Specifically, Zipf–RQA is used here as a descriptive structural probe rather than a performance metric.

Keywords: transformer; dense connectivity; decoder-only language modeling; perplexity; causal masking; parameter budget; ablation study; probing tasks; Zipf–RQA

1. Introduction

Generative AI systems leveraging Transformer-based Large Language Models (LLMs) have shown exceptional performance in language production and understanding, although important limitations remain [1]. These limitations matter for researchers and practitioners who train, deploy, or depend on LLMs, because they can translate into higher computational and energy costs and into brittle failures in downstream applications; without improvements in architectures and evaluation, progress risks remaining dominated by expensive scaling with limited insight into how information propagates through depth. Transformers are the de facto backbone for sequence modeling through neural networks, with progress driven more by scale and computation than by changes in intra-stack information flow [2–4]. Modern LLM architectures typically consist of a stack of decoder layers, each combining self-attention and a feed-forward network, and are designed to model conceptual spaces [5,6] and long-range correlations that support coherent generation [7–11].

Academic Editor: Douglas O'Shaughnessy

Received: 29 January 2026
Revised: 5 March 2026
Accepted: 8 March 2026
Published: 12 March 2026

The stack is hierarchical, that is, each layer receives the representation produced by the *preceding* layer and refines it, while information from deeper layers influences a given layer only through this sequential transformation (unless additional cross-layer connections are introduced).

From a methodological standpoint, we ask whether granting each decoder layer broader access to the stack's evolving representations translates into better next-token prediction under controlled experimental budgets. We study decoder-only stacks endowed with decoder-style causal masking and fixed context, adopting perplexity (PPL) as the primary metric [12], and complement it with learning dynamics and targeted probes. To separate the effect of connectivity from confounding factors, we report results under two fairness regimes: *same training recipe* (same training protocol and learning-rate grid, with the baseline architecture fixed and tuned only over the learning-rate grid, while dense is explored within a bounded architectural space) and *same parameter budget* (dense resized to not exceed the baseline parameter count). To probe beyond likelihood, we also analyze long-form generations through corpus-agnostic structural descriptors derived from Zipf-rank encoding and Recurrence Quantification Analysis (RQA), i.e., Zipf–RQA, providing a complementary view of global organization in generated text [7]. The objective of this study is to isolate and evaluate the impact of concatenation-based dense historical connectivity in decoder-only Transformers under controlled comparison regimes, considering both perplexity and complementary behavioral/structural diagnostics. In this work, we address the following research questions:

RQ1. Under a controlled training and model-selection protocol, does concatenation-based densification with learned projection yield lower test PPL than a standard Transformer decoder, when comparing (i) a fixed baseline against a bounded dense architectural search under the *same training recipe* (*hyperparameters regime*, i.e., shared optimization settings with the baseline architecture fixed and dense selected from a limited architectural grid), and (ii) a parameter-budget constrained dense model against the fixed baseline under the *same parameter budget* (*parameters regime*)?

RQ2. Within the explored design space, how does test PPL vary as a function of architectural capacity allocation (d_{model}, d_{ff}, L, h), and do these factors explain the observed baseline–dense gap more strongly than the presence of dense historical connectivity itself?

RQ3. Do long-form generations from baseline vs. dense models exhibit different long-range structural signatures, as quantified by Zipf–RQA descriptors, and how do these differences relate to the PPL gaps observed under the hyperparameters regime?

To provide answers to the above RQs, we run a controlled set of experiments on Penn Treebank (PTB) and WikiText-2 (WT2) under two fairness regimes: *same training recipe* (the *hyperparameters regime*) and *same parameter budget* (the *parameters regime*), reporting test PPL as the primary readout. We complement PPL with learning dynamics and targeted analyses of internal behavior (attention maps and cloze-style probes), and we analyze long-form generations via Zipf–RQA. Our main takeaway is that, within the explored experimental envelope, dense historical connectivity through concatenation and projection does not consistently improve test perplexity over a standard baseline decoder, while depth and feed-forward capacity emerge as the most reliable drivers of PPL within the dense family.

This paper is organized as follows.
Section 2 reviews related work on dense connectivity and cross-layer aggregation. Section 3 describes the language modeling task, the two Transformer variants, the training protocol, the fairness criteria, and the quantitative and qualitative evaluation setup. Section 4

presents the experimental results on PTB and WT2, including the main grid search, the ablation study, and the probe-based and long-text generation analyses. Section 5 discusses the implications of these findings, the limitations of our study, and several directions for further investigation. Finally, the Conclusions Section summarizes the main takeaways and outline possible extensions, while Appendix A provides additional experimental details.

2. Literature Review

The idea of dense connectivity through concatenation originates in computer vision with DenseNet [13], where each block receives the feature maps of all preceding layers and compresses them via bottleneck projections, yielding shorter gradient paths and extensive feature reuse. In parallel, Highway Networks [14] demonstrated that learnable gating along depth can stabilize optimization and generalize residual connections, offering an additive alternative to concatenation.

Transposed to Transformers, cross-layer aggregation appears in "DenseFormer" variants [15,16], which aggregate multiple intermediate representations and project them back to a fixed hidden size, often using shared or selective fusion to control parameter growth. Other works, such as TransReID [17], augment Vision Transformers with task-specific side information rather than altering intra-decoder connectivity.

Our dense decoder lies within this design space by concatenating the full historical sequence of layer outputs at each depth and applying a learned projection to maintain a constant d_{model}, while operating under causal masking for autoregressive language modeling with fixed context length. More selective strategies, including gating, rank-constrained projections, or selective historical aggregation, remain promising directions for future exploration [18,19].

3. Materials and Methods

This work is a controlled empirical, predominantly quantitative study that evaluates a decoder-only Transformer architectural variant under a fixed, reproducible training protocol. The scope is a budget-aware comparative analysis on two public benchmark datasets (PTB and WT2), with two fairness regimes designed to control confounders (same training recipe vs. same parameter budget) and with targeted ablations and control checks to isolate the effect of connectivity within the explored envelope. Evidence is derived from next-token language modeling runs and is analyzed through standard likelihood-based metrics (cross-entropy and PPL) complemented by diagnostic probes, attention visualizations, and long-form generation analysis via Zipf–RQA.

3.1. Problem Statement and Data

In the following, we will concentrate on autoregressive language modeling. For a sequence of token IDs $X_{1:t}$ at timestamp t, with next token x_{t+1}, we seek to maximize the log-likelihood of x_{t+1} given $X_{1:t}$. The training loss reads as the Cross-Entropy (CE):

$$\mathcal{L} = \text{CE}\big(\text{logits}(X_{1:t}),\, x_{2:t+1}\big), \tag{1}$$

where $\mathcal{L}$ is understood as the mean negative log-likelihood per predicted token. We report PPL as the corresponding exponentiated loss:

$$\text{PPL} = \exp(\mathcal{L}). \tag{2}$$

As anticipated, experiments feature WT2 and PTB as datasets. Each split is concatenated into a single token stream, then reshaped into matrices of shape $[T, B]$ (sequence

length T by batch size B). Training employs sliding windows of fixed context length T_c, without padding.

Concerning tokenization and vocabulary, we adopt a word-level modeling setup. We use Hugging Face Datasets for loading the official train/validation/test splits for both datasets (WT2 raw and PTB text-only), and apply the `basic_english` tokenizer from TorchText. Tokens are mapped to integer IDs via a vocabulary built from the training split; out-of-vocabulary tokens are mapped to a dedicated ⟨unk⟩ symbol. Accordingly, all perplexity values reported in this work are computed at the word level for this specific vocabulary and tokenization, and their absolute scale is not directly comparable to subword/BPE-based perplexity values commonly reported for WT2/PTB in the literature.

3.2. Architectures

This subsection outlines the two decoder-only Transformer architectures examined in our study. Both models operate on token sequences $\{x_t\}_{t=1}^{T}$ drawn from a vocabulary $\mathcal{V}$ and share a common structural scaffold, that are learned token embeddings, positional encodings, causal self-attention, and a linear prediction head. The primary distinction between the Baseline Transformer decoder and the Dense Transformer decoder lies in the internal organization of their decoder layers, the former following the canonical residual formulation and the latter integrating dense historical connections inspired by progressively aggregated feature reuse. In Figure 1, we compare the standard Transformer decoder layer with the proposed dense decoder layer, highlighting the concatenation-and-projection path and the residual connection that originates from $h^{(\ell-1)}$.

Regarding token and positional embeddings, let $x = (x_1, \ldots, x_T)$ denote the input sequence. Each token is mapped to a d_{model}-dimensional vector via an embedding matrix $E \in \mathbb{R}^{|\mathcal{V}| \times d_{\text{model}}}$. Positions are encoded by a fixed sinusoidal scheme in the baseline decoder and by learned absolute positional embeddings in the dense decoder. For notational simplicity, we denote positional encodings in both cases by a matrix $P \in \mathbb{R}^{T \times d_{\text{model}}}$ and write

$$h_t^{(0)} = E[x_t] + P[H]. \tag{3}$$

A dropout layer is applied to $h^{(0)}$ before entering the decoder stack.

For causal masking, all attention layers make use of a strict causal mask that prevents information leakage from future tokens. Given a sequence of length T, the mask $M \in \{0,1\}^{T \times T}$ is defined as

$$M_{ij} = \begin{cases} 1 & \text{if } j > i, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where entries with $M_{ij} = 1$ are masked out in the attention logits. This constraint is consistently enforced in every self-attention computation and aligns with the decoder-style masking customary in autoregressive language modeling.

Concerning the Baseline Transformer decoder, the Baseline Transformer adheres to the standard decoder block design. For layer ℓ, attention is computed over the input $h^{(\ell-1)}$ according to

$$\tilde{h}^{(\ell)} = \text{MHA}\big(h^{(\ell-1)}, h^{(\ell-1)}, h^{(\ell-1)}; M\big), \tag{5}$$

where MHA is the Multi-Head Attention operator with shared query, key, and value projections. Residual integration and normalization yield

$$z^{(\ell)} = \text{LayerNorm}\Big(h^{(\ell-1)} + \tilde{h}^{(\ell)}\Big). \tag{6}$$

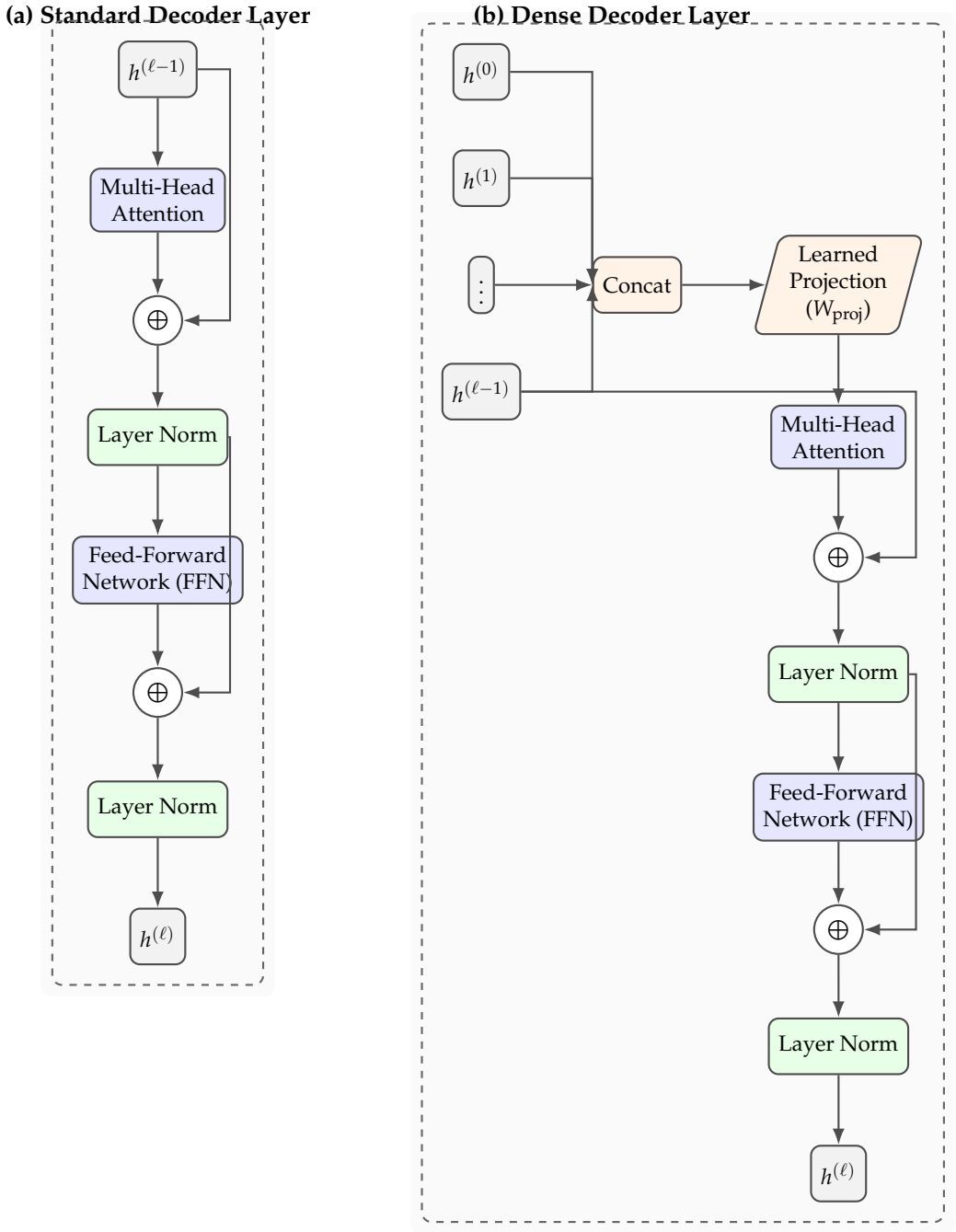

Figure 1. Architectural comparison between (**a**) the Baseline Transformer Decoder layer and (**b**) the proposed Dense Transformer Decoder layer. The dense layer concatenates historical representations ($h^{(0)}$ to $h^{(\ell-1)}$) and projects them before the self-attention mechanism, while the residual connection originates exclusively from $h^{(\ell-1)}$, preserving compatibility with the standard decoder formulation.

The feed-forward network (FFN) is a two-layer multi-layer perceptron (MLP) with hidden size d_{ff},

$$\text{FFN}(u) = W_2\,\text{ReLU}\big(W_1\,u\big), \tag{7}$$

where W_1 and W_2 are learned projection matrices and ReLU (rectified linear unit) being the non-linear activation function. The resulting transformation is integrated through a residual pathway

$$h^{(\ell)} = \text{LayerNorm}\Big(z^{(\ell)} + \text{FFN}\big(z^{(\ell)}\big)\Big). \tag{8}$$

Consistent with the current implementation, only the output of the final decoder layer is instrumented for attention visualization and diagnostic analysis.

On the other hand, the Dense Transformer Decoder introduces a structured form of feature accumulation across depth. At layer ℓ, the model aggregates all prior representations by concatenating them along the channel dimension

$$c^{(\ell)} = \text{Concat}\Big(h^{(0)}, h^{(1)}, \ldots, h^{(\ell-1)}\Big), \tag{9}$$

thus producing a tensor with dimensionality $(B, T, \ell d_{\text{model}})$. A learned projection maps this expanded multi-level representation back to the model dimension:

$$p^{(\ell)} = W^{(\ell)}_{\text{proj}}\, c^{(\ell)}. \tag{10}$$

Self-attention is then computed directly on $p^{(\ell)}$,

$$\tilde{h}^{(\ell)} = \text{MHA}\big(p^{(\ell)}, p^{(\ell)}, p^{(\ell)}; M\big), \tag{11}$$

while the residual pathway preserves the original layer input $h^{(\ell-1)}$,

$$z^{(\ell)} = \text{LayerNorm}\Big(h^{(\ell-1)} + \tilde{h}^{(\ell)}\Big). \tag{12}$$

The feed-forward stage mirrors the baseline:

$$h^{(\ell)} = \text{LayerNorm}\Big(z^{(\ell)} + \text{FFN}\big(z^{(\ell)}\big)\Big). \tag{13}$$

Through this mechanism, each layer receives a progressively enriched historical summary of the network's internal evolution. This dense connectivity promotes feature reuse and mitigates representational attenuation across depth, providing a principled architectural alternative to the standard residual-only formulation.

For assessing output projections and language modeling heads, the final decoder output $h^{(L)}$ is transformed into token logits by a linear prediction head with parameters $W_{\text{out}} \in \mathbb{R}^{d_{\text{model}} \times |\mathcal{V}|}$,

$$\text{logits}_t = W_{\text{out}}^{\top} h_t^{(L)}. \tag{14}$$

Training proceeds through standard next-token cross-entropy loss, applied independently at each position of the sequence.

3.3. Training and Fair Comparison Settings

Unless otherwise stated, optimization uses AdamW with global gradient clipping (L2 norm) at 0.5, multiplicative learning rate (LR) decay $\gamma = 0.95$ per epoch, and fixed context length T_c. Training batches are contiguous token windows $[i{:}i + T_c]$. The best validation-loss checkpoint is selected for testing. Training is carried out with automatic mixed precision to improve throughput, while evaluation metrics are computed in standard precision. The full optimization and reproducibility configuration is summarized in Table 2 (see later Section 4.1).

To separate the effect of connectivity from other confounding factors, we report results under two fairness regimes. In the same training recipe regime (hyperparameters regime), the baseline architecture is held fixed, while the dense architecture is explored within a bounded design space; in both cases, the optimization settings (epochs, batch size, dropout, and a small learning-rate grid) are shared and the best checkpoint is selected by validation loss. In the same parameter budget regime (parameters regime), the baseline remains fixed and the dense model is resized to respect the baseline parameter budget by reducing

$(d_{\text{model}}, d_{\text{ff}})$ under the constraints $d_{\text{model}} \bmod h = 0$ and $d_{\text{ff}} \approx 4d_{\text{model}}$, until its parameter count does not exceed the baseline. The regime nomenclature used throughout the paper is summarized in Table 1.

Table 1. Regime nomenclature and definition used throughout the paper. The two regimes differ in what is controlled across model families (optimization settings vs. parameter budget) while holding the evaluation protocol and model-selection criterion fixed (best-by-validation checkpoint). Exact grids and configurations are reported in Section 4.1.

Regime	Comparison Setup	Key References
Same training recipe (*hyperparameters*)	Baseline: fixed architecture, tuned only over the shared optimization grid. Dense: selected from a bounded architectural search under the same optimization settings and model-selection criterion.	Architectures: Section 3.2 (baseline: Equations (5)–(8); dense: Equations (9)–(13)). Shared optimization recipe: Table 2. Grid/search details: Section 4.1.
Same parameter budget (*parameters*)	Baseline: fixed architecture; its parameter count defines the budget. Dense: resized to not exceed the baseline parameter budget (by reducing d_{model} and d_{ff} under divisibility/ratio constraints), then trained under the same optimization settings and model-selection criterion.	Budget constraint and resizing: Section 3.3. Architecture equations: Section 3.2. Shared optimization recipe: Table 2. Grid/search details: Section 4.1.

3.4. Evaluation

The performed primary evaluation is both quantitative and qualitative.

Concerning the qualitative one, the selected checkpoint is evaluated on the test split using T_c, reporting CE loss:

$$\mathcal{L} = -\frac{1}{BT} \sum_{b=1}^{B} \sum_{t=1}^{T} \log p_\theta(y_{b,t} \mid y_{b,<t}), \tag{15}$$

and PPL:

$$\text{PPL} = \exp(\mathcal{L}). \tag{16}$$

For qualitative evaluation and diagnostic purposes, beyond likelihood-based evaluation, we analyze model behavior via: (i) last-layer attention heatmaps, (ii) cloze-style syntactic/semantic probes, and (iii) long-form text generation with greedy, top-k, and nucleus (top-p) decoding. For generation, we use a sliding context of length T_c: at step t, the model conditions on the most recent $\min(t, T_c)$ tokens, recomputing positions and masks as in Equation (4).

Probing tasks: Beyond aggregate likelihood, we use targeted next-token probes to obtain a more fine-grained diagnostic signal about specific syntactic and semantic cues that may not be visible from PPL alone. We use a small set of cloze-style probes that test targeted capabilities by querying the model with short prompts and inspecting the probability assigned to an expected next token [20,21]. We consider three probe families: (i) subject–verb agreement (SVA), which probes syntax-sensitive dependencies [22]; (ii) short factual prompts (FACT), which probe whether a specific entity-relation completion is preferred; and (iii) a simple contextual continuation (CTX), which probes whether the expected continuation is supported by local context. For each probe instance, let $y^\star$ denote the expected next token under a given prompt and let $p_\theta(\cdot)$ be the model distribution over the vocabulary at that position. We report Top-1 and Top-5 success (indicator that $y^\star$ is

the most likely token or appears among the 5 most likely tokens); the rank of $y^\star$ when tokens are sorted by $p_\theta(\cdot)$ in descending order; and $\log p_\theta(y^\star)$. To reduce sensitivity to a single wording, we evaluate multiple prompt variants per family and aggregate results by reporting Top-1/Top-5 rates, the median rank, and the mean log-probability across variants and instances.

Long-text structural analysis (Zipf–RQA): To complement these local probes with a global view of generative structure, we also analyze long-form samples through Zipf–RQA descriptors, which summarize long-range recurrence patterns in the generated sequences. To characterize the structural properties of long-form generations beyond likelihood, we adopt a complex-systems perspective on text by transforming each sample into a numeric series and analyzing its recurrence patterns [7]. Specifically, each text is mapped to a Zipf-rank time series [23]: tokens are replaced by their rank in the text's empirical frequency distribution (rank 1 = most frequent), yielding a rank-based trajectory over positions. We then compute windowed RQA [24–26], estimating delay and embedding dimension from the series and fixing the recurrence rate to standardize comparisons across samples. We report standard RQA descriptors, including determinism (DET), laminarity (LAM), entropy (ENTR), trapping time (TT), and diagonal/vertical line statistics (e.g., $L_{\max}$, L_{mean}, $V_{\max}$, V_{entr}). Intuitively, these descriptors summarize how often the series revisits similar states and whether such recurrences organize into structured patterns. Higher DET indicates that recurrences concentrate along diagonal lines, consistent with more predictable, temporally coherent dynamics; LAM and TT quantify vertical-line structures, capturing laminar/intermittent behavior and the typical duration of quasi-stationary episodes. ENTR measures the diversity/complexity of diagonal-line lengths, while $L_{\max}$ and L_{mean} summarize the strength and typical extent of repeated trajectories; analogously, $V_{\max}$ and V_{entr} summarize the extent and heterogeneity of laminar episodes.

3.5. Intuition and Qualitative Rationale

Our comparison is best understood as two different ways of letting depth accumulate and reuse information. In the standard decoder formulation (Section 3.2), each layer refines only the representation received from the layer immediately below, as in Equations (5)–(8). This creates a single, narrow conduit through which early cues must travel, and any signal that fades along the way is costly to recover.

The dense decoder formulation (Section 3.2) instead exposes each layer to the full representational history accumulated so far, as formalized in Equation (9). Early lexical hints, mid-level phrase structure, and emerging long-range dependencies are concatenated and then projected back to d_{model} via Equation (10) before self-attention. Later layers are therefore not forced to rely on a single compressed summary of the past; they can attend directly to whichever abstraction is most useful at a given position. This shortens both information and gradient paths and encourages feature reuse rather than repeated relearning.

Without the learned projection $W_{\mathrm{proj}}^{(\ell)}$ in Equation (10), concatenating all previous outputs would cause the width to grow linearly with depth, quickly becoming unwieldy. The projection acts as a depth-aware bottleneck that blends the historical channels into a compact state, effectively deciding how much of the early versus intermediate representations to carry forward. This transforms a naive width explosion into a stable input for the masked MHA in Equation (11), at the cost of an additional set of projection parameters. Ignoring biases, the extra parameters contributed by the dense projections alone scale as

$$\sum_{\ell=1}^{L} \ell \, d_{\mathrm{model}}^2 \;=\; \frac{L(L+1)}{2} \, d_{\mathrm{model}}^2, \tag{17}$$

since at depth ℓ the model introduces a matrix $W_{\text{proj}}^{(\ell)} \in \mathbb{R}^{\ell d_{\text{model}} \times d_{\text{model}}}$. In practice, the implementation uses linear layers with bias, so the total parameter count includes an additional $\mathcal{O}(Ld_{\text{model}})$ term from the projection biases, which does not affect the leading-order scaling in Equation (17).

Because our goal is autoregressive language modeling, causality must be enforced not just during generation but also during training. The strictly upper-triangular mask M in Equation (4) guarantees that every attention pattern respects temporal order, so the likelihood in Equations (1) and (2) is optimized under the same constraints faced at test time. Dense connections do not undermine this contract, since attention always operates on masked, time-aligned histories.

Concerning positional encodings, in both baseline and dense models embeddings are kept deliberately simple so that the connectivity difference remains the main experimental variable. In all cases, it is the mask—not the positional parameterization—that guarantees the arrow of time; generation mirrors training by conditioning on a sliding window of the most recent T_c tokens.

These design choices come with trade-offs. Self-attention remains the dominant $\mathcal{O}(T^2)$ cost in sequence length, while the dense projection introduces a linear-in-depth parameter overhead and a modest runtime increase. This motivates the two evaluation regimes in Section 3.3—equal hyperparameters and equal parameter count—so that any improvements are not merely attributable to extra capacity. In practice, the dense topology tends to help most in deeper stacks and in settings where heterogeneous cues must be combined (e.g., local lexical markers and broader syntactic scaffolds). Potential drawbacks mirror those of richer parameterizations, such as overfitting in small-data regimes and occasional co-adaptation among layers, mitigated by regularization and careful parameter matching.

Finally, regarding interpretability, in the dense model, last-layer attention often highlights a blend of early and mid-level signals, which can blur a simple layer–abstraction correspondence. This is by design, in that late decisions are informed by a palette of earlier views rather than a single compressed stream. Our claims are therefore scoped precisely: fixed-context, decoder-only, causal language modeling, comparing a progressive refinement stack to one that deliberately re-exposes its history at every depth.

4. Results

This section is structured around the three research questions stated in the Introduction. For RQ1, under both fairness regimes, dense historical connectivity does not yield a consistent perplexity improvement over the baseline decoder: on WT2 the baseline remains better, while on PTB the gap is small and regime-dependent. For RQ2, the ablations indicate that most of the performance variation is explained by capacity allocation (in particular depth and feed-forward width) rather than by connectivity alone. For RQ3, Zipf–RQA of long-form generations reveals systematic family-wise shifts in long-range structural descriptors, even when likelihood-based metrics do not improve.

4.1. Experimental Setting

The experimental campaign was designed to systematically assess the performance of the proposed Dense Transformer Decoder against a standard Transformer Decoder baseline. General architectural details, training procedures, and dataset preprocessing steps are provided in Section 3. Here, we summarize the specific configurations, search protocols, and evaluation criteria relevant to the two grid search setups adopted in this study (Table 2).

Table 2. Compact training configuration (shared across runs unless stated otherwise). This table summarizes the optimization and reproducibility settings used for both baseline and dense models.

Item	Value
Loss	Next-token cross-entropy (Equations (1) and (2)).
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay $= 0.01$).
Learning rate	Grid $\{3 \times 10^{-4}, 1 \times 10^{-4}\}$; best selected by validation loss.
LR schedule	StepLR (step size $= 1$ epoch, $\gamma = 0.95$); no warmup.
Gradient clipping	Global L2 norm clipped at 0.5.
Dropout	0.1 (applied to embedding+positional sum and within attention/FFN sublayers).
Context length	$T_c = 128$ tokens (fixed sliding window; no padding).
Batching	Batch size $= 128$ (total); contiguous windows $[i{:}i + T_c]$ from a concatenated token stream.
Epochs	15.
Random seed	42 (controls shuffling and initialization).
Precision	AMP for training (autocast + GradScaler); evaluation in standard precision.

We evaluated the models on two widely used benchmark datasets for language modeling:

- PTB [27]: a syntactically annotated corpus of American English, preprocessed following standard tokenization and rare-word handling conventions [28].
- WT2 [29]: a corpus extracted from Wikipedia articles, characterized by a larger vocabulary and longer-range dependencies compared to PTB.

As anticipated in Section 3.1, in both cases, the official training, validation, and test splits provided by the dataset authors were used without modification.

Performance was measured in terms of test PPL, the standard metric for autoregressive language modeling [12], where lower values indicate better generalization. Each model was trained and evaluated under both the *hyperparameters* and *parameters* fairness regimes (Section 3.3). In addition to PPL, we assess linguistic generalization through targeted probes [30,31].

For the long-form analysis on WT2 (hyperparameters regime), we generate $N = 150$ texts per family from a fixed prompt (*"The future of language models is"*) using top-k sampling ($k = 50$, temperature 0.9), with a target length of 8000 words per sample. We use the best WT2 checkpoints under this regime, baseline `run_005` and dense `run_059` (Table 3). Zipf–RQA is computed on word-level Zipf-rank series with windowed RQA over four overlapping windows (20% overlap), fixing the recurrence rate to 20% for comparability. Representative recurrence plots in Section 4.5 (window 0) are selected by a median-case criterion: within each family, we choose the sample whose DET/LAM ratio is closest to the family median.

All experiments share the same optimization settings: LR $\in \{3 \times 10^{-4}, 10^{-4}\}$, dropout 0.1, epochs 15, batch size 128, and context length $T_c = 128$ (BPTT truncation length). Architectural parameters vary by grid and fairness regime as detailed below, with the regime nomenclature summarized in Table 1 (Section 3.3).

For both regimes, baseline models in Grid A use a fixed architecture:

$$(d_{\text{model}}, L, h, d_{\text{ff}}) = (768, 8, 12, 3072). \tag{18}$$

The baseline is tuned only over the LR grid above.

Under the hyperparameters regime in Grid A, dense models explore only a limited architectural subset. All configurations shared $d_{\text{model}} = 768$ and $h = 8$, while depth and feed-forward width were varied according to

$$L \in \{6, 8, 10\}, \qquad d_{\text{ff}} \in \{2560, 3072, 3584\}. \tag{19}$$

This search space is intentionally narrow by design, in that fixing the representation scale and the head setting reduces degrees of freedom and computational cost, and aims to isolate the effect of dense historical connectivity under a controlled training recipe rather than to perform an unconstrained architecture search. Potential confounds introduced by these fixed architectural choices (e.g., head count and positional encoding differences in Grid A) are addressed via explicit control checks in the WT2 results (see in the following). Grid C then provides a broader capacity-reallocation ablation that also varies d_{model} and h to contextualize the Grid A findings.

In the *parameters* regime in Grid A, dense models are resized to not exceed the baseline parameter count (Section 3.3). We iteratively reduce $(d_{\text{model}}, d_{\text{ff}})$ subject to d_{model} mod h = 0 and $d_{\text{ff}} \approx 4d_{\text{model}}$ until the dense budget constraint is satisfied.

Two configurations structure the experimental campaign. Grid A (primary) includes both fairness regimes using the fixed baseline configuration in Equation (18) and the dense settings described above. Grid C (ablation) is a broader capacity-reallocation study under the *hyperparameters* regime, and its architectural search space is defined by

$$d_{\text{model}} \in \{512, 640, 768, 832\}, \tag{20}$$

$$L \in \{4, 6, 8, 10\}, \tag{21}$$

$$h \in \{8, 12\}, \tag{22}$$

$$d_{\text{ff}} \in \{2048, 2560, 3072, 3584\}. \tag{23}$$

All optimization hyperparameters coincide with those used in Grid A. For the depth–PPL summary in Grid C (Figure 2), the $h = 12$ subset uses $d_{\text{model}} = 768$, while the $h = 8$ subset spans Equation (20).

Concerning the random seed control, to ensure reproducibility, all runs were initialized with fixed random seeds, controlling for sources of stochasticity in dataset shuffling, parameter initialization, and parallel computation, following best practices for deterministic deep learning experiments [18]. This overall setup ensured a controlled and reproducible framework for comparing the two architectures under both the *same training recipe* and *same parameter budget* regimes, providing a consistent basis for the analyses presented in the following sections.

4.2. Quantitative Results (Grid A)

This subsection addresses RQ1 by comparing baseline and dense decoders under the two fairness regimes, and it supports RQ2 by inspecting how outcomes vary across the explored dense configurations. We report test PPL, comparing the proposed dense decoder against the baseline across datasets and fairness regimes (see Section 3.3 and Table 1). These results correspond to the hyperparameter exploration of Grid A, our primary experimental setup, which systematically explores both *hyperparameters regime* and *parameters regime* (parameter-budget) fairness regimes, using the search space and selection criteria in Section 4.1. Unless stated otherwise, all figures and tables in this subsection refer to the best checkpoint per configuration, selected on validation loss according to Section 3.4.

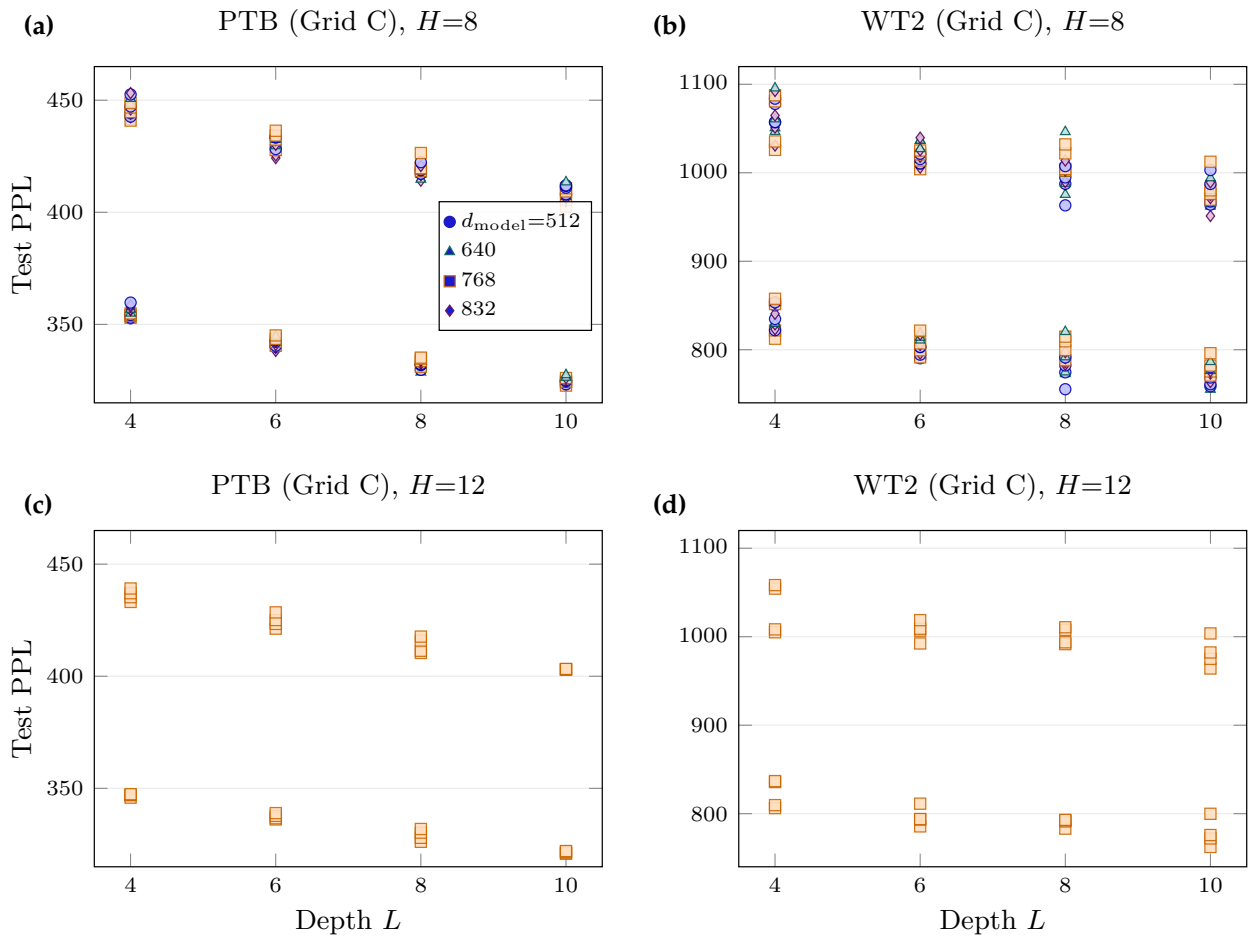

Figure 2. Grid C (dense, hyperparameters): test PPL vs. depth L for PTB and WT2, stratified by number of heads ($H \in \{8, 12\}$). Panels: (**a**) PTB, $H = 8$; (**b**) WT2, $H = 8$; (**c**) PTB, $H = 12$; (**d**) WT2, $H = 12$. Points are colored/marked by model width d_{model} in the $H = 8$ panels; in the $H = 12$ panels, all points correspond to $d_{\text{model}} = 768$. Each point corresponds to one trained configuration in the grid.

For an overview across datasets and regimes, Table 3 summarizes the best PPL achieved on PTB and WT2 under both the *hyperparameters regime* and the *parameters regime* (see Section 3.3). On PTB, the dense variant yields a small improvement under the *hyperparameters* regime, while the baseline remains better under the parameter-budget constraint. On WT2, the baseline is consistently better in both regimes, indicating that dense historical connections do not straightforwardly translate into lower perplexity in this decoder-only, fixed-context setting.

From the perspective of RQ2, the Grid A ablations provide a simple causal takeaway. Specifically, within our controlled comparisons, *capacity-allocation knobs* such as depth (L) and feed-forward width (d_{ff}) are the factors that reliably shift perplexity, whereas introducing dense historical connectivity by itself does not yield a consistent perplexity gain across datasets and fairness regimes. This is visible not only in the best-checkpoint comparisons (Table 3), but also in the distribution of outcomes across the explored dense configurations (Figure 3), which shows that WT2 remains worse than the tuned baseline throughout the explored design space.

Table 3. Best test PPL per dataset, fairness regime, and architecture. Full configurations are reported in Appendix A.

Dataset	Regime	Model	Params	PPL	Best Run ID
PTB	hyperparams	baseline	71.95 M	325.13	run_001
PTB	hyperparams	dense	118.68 M	322.69	run_023
PTB	params	baseline	71.95 M	325.13	run_003
PTB	params	dense	46.87 M	328.61	run_029
WT2	hyperparams	baseline	158.23 M	726.14	run_005
WT2	hyperparams	dense	204.96 M	770.74	run_059
WT2	params	baseline	158.23 M	726.14	run_007
WT2	params	dense	134.69 M	766.54	run_071

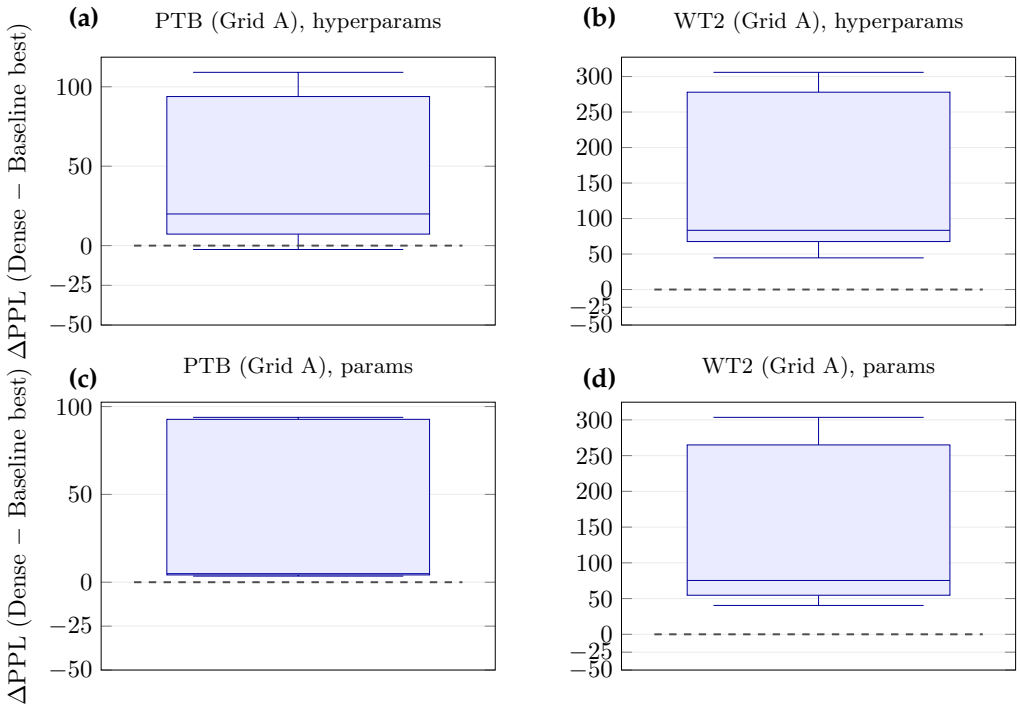

Figure 3. Grid A: distribution of test ΔPPL across the explored dense configurations, where $\Delta\text{PPL} = \text{PPL}_{\text{dense}} - \text{PPL}_{\text{baseline(best)}}$ is computed within each dataset and fairness regime. Panels: (**a**) PTB, hyperparameters; (**b**) WT2, hyperparameters; (**c**) PTB, parameters; (**d**) WT2, parameters. Here, $\text{PPL}_{\text{baseline(best)}}$ denotes the best baseline result within Grid A for that dataset/regime (baseline architecture fixed; only learning rate is varied in the grid). Values above zero indicate worse perplexity than the best baseline in the same setting.

Importantly, Figure 3 complements the best-checkpoint comparison by reporting the full distribution of dense outcomes over the explored design space. Because ΔPPL is referenced to the best baseline setting within Grid A (baseline fixed architecture with a small LR grid), the figure directly tests whether the dense topology can outperform a tuned baseline under the same dataset and regime. On WT2, ΔPPL remains positive across the explored dense design space in both regimes, indicating that the observed gap is not driven by a single outlier configuration. On PTB, a small subset of dense configurations approaches (or slightly improves upon) the tuned baseline under the *hyperparameters* regime, consistent with the narrower gap observed in Table 3.

Given its larger vocabulary and longer-range dependencies, WT2 is our primary testbed for figures and probes. Since our setup uses word-level tokenization (Section 3.1), the resulting perplexity values are expected to be numerically higher than subword/BPE-based reports and should be interpreted within this evaluation setup. Table 4 reports

absolute and relative differences between dense and baseline under both regimes. The dense decoder is worse in PPL by +44.60 points (+6.14%) in the *hyperparameters regime*, and by +40.40 points (+5.56%) under the *parameters regime* (parameter—budget constraint). These deltas suggest that, within the decoder-only causal setting, exposing every layer to the full representational history (as described in Section 3.2) does not yield a lower language-modeling perplexity on WT2 when controlling either for optimization settings or for a fixed parameter budget.

Control checks for architectural confounds (WT2, hyperparameters): Grid A comparisons may be confounded by architectural differences beyond connectivity, notably the attention-head count (baseline $H = 12$ vs. dense $H = 8$) and the positional encoding choice (baseline sinusoidal vs. dense learned absolute embeddings). To quantify the magnitude of these effects in isolation, we ran two minimal baseline controls under the same WT2 *hyperparameters regime* recipe as the original baseline (Table 4): (i) a head-matched baseline with $H = 8$, achieving 723.41 test PPL, and (ii) a positional-encoding-matched baseline using learned absolute positional embeddings, achieving 727.06 test PPL. Both controls are close to the original baseline (726.14) and remain substantially below the dense Grid A WT2 perplexity (770.74), suggesting that these specific confounds do not explain the WT2 gap observed for the dense configuration.

Table 4. WT2 deltas (Dense minus Baseline) under both fairness regimes.

Regime	Baseline PPL	Dense PPL	ΔPPL	Δ %
hyperparams	726.14	770.74	44.60	6.14
params	726.14	766.54	40.40	5.56

To complement likelihood-based evaluation, we also report cloze-style probes on WT2 under the *hyperparameters* regime (see Section 3.4). These probes are intended as lightweight behavioral diagnostics, not as definitive measures of linguistic competence. In particular, Top-k success (Top-1/Top-5) is a strict criterion in a word-level setup with a large vocabulary, and small lexical or contextual variations can move the expected continuation outside the top ranks even when it receives non-negligible probability mass. Accordingly, we emphasize rank and $\log p(y^\star)$ as the more informative signals for these probes. As summarized in Tables 5 and 6, Top-1/Top-5 success is zero across the reported probe families at context length T_c, which is consistent with the rank statistics (median ranks remain above 5). Within this evaluation setting, the baseline tends to assign higher probability to the expected tokens in FACT and CTX, consistent with the PPL gap in Table 4.

Table 5. WT2 probes (hyperparameters regime): aggregated Top-1/Top-5 rates, median rank, and mean log-probability of the expected token across prompt variants for each probe family. Top-k is a strict success criterion, while rank and log-probability provide a graded view of whether the expected continuation is preferred even when it is not among the top candidates.

	Baseline (run_005)					Dense (run_059)				
Probe	***N***	**Top-1**	**Top-5**	**Median Rank**	$\overline{\log p}$	***N***	**Top-1**	**Top-5**	**Median Rank**	$\overline{\log p}$
SVA	16	0.0	0.0	26	−5.65	16	0.0	0.0	24	−5.52
FACT	5	0.0	0.0	1572	−10.00	5	0.0	0.0	2823	−10.64
CTX	3	0.0	0.0	354	−8.98	3	0.0	0.0	717	−9.71

Table 6. Representative probe examples (WT2, hyperparameters regime). Reported ranks and log-probabilities refer to the expected continuation token $y^\star$ (or the best-matching token within the target set, when multiple targets are allowed). These examples illustrate why Top-k accuracy is a stringent criterion in our word-level setup and why rank/log-probability provides a more graded diagnostic signal.

			Baseline		Dense	
Probe	**Prompt (Next Token)**	$y^\star$	**Rank**	**log p**	**Rank**	**log p**
SVA	`the key to the cabinets on the wall far from the door`	`is`	9	−3.90	21	−4.95
FACT	`the capital of italy is`	`rome`	1355	−9.80	2823	−10.67
CTX	`after she dropped the glass, it`	`{broke, shattered, cracked, smashed}`	340	−8.89	704	−9.69

Figure 4 compares evaluation loss/PPL summaries of the best baseline (run_005) and dense (run_059) models on WT2 under the *same training recipe* (hyperparameters) regime. The baseline achieves a lower test PPL.

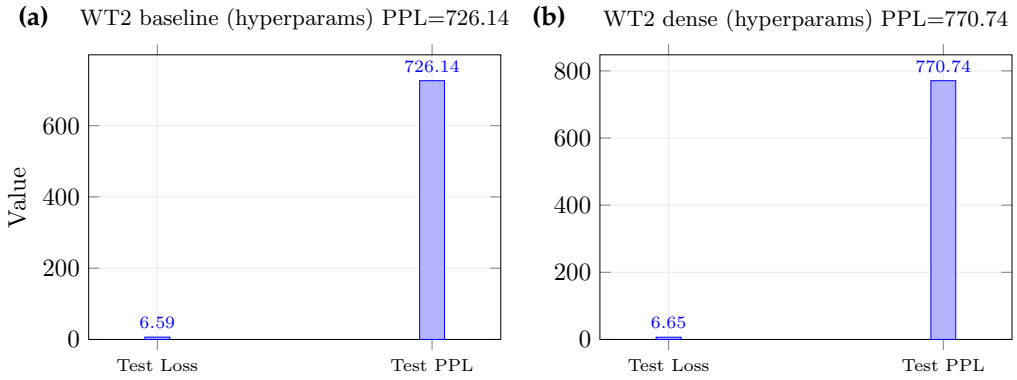

Figure 4. WT2: evaluation summaries under the *same training recipe* (hyperparameters) regime. Panels: (**a**) baseline (run_005); (**b**) dense (run_059). Each panel reports the test loss and test PPL.

Figure 5 qualitatively compares last-layer self-attention heatmaps on WT2 between the two models on the same fixed input and context length. Both allocate substantial mass to recent tokens (consistent with causal masking and local predictive cues). However, the dense model exhibits a more heterogeneous pattern with sharper off-diagonal hotspots, whereas the baseline shows a more regular near-diagonal decay. To complement the illustrative heatmaps with a lightweight summary statistic, Table 7 reports two batch-averaged metrics computed on the WT2 test split: normalized attention entropy (lower indicates a more peaked distribution) and diagonal mass (higher indicates stronger self-token attention). Under this setting, the dense model shows lower entropy and higher diagonal mass, consistent with a more concentrated attention profile.

Table 7. WT2 attention summary metrics (hyperparameters regime). Metrics are averaged over 25 batches from the test split ($T_c = 128$). Normalized entropy is computed per query position and normalized by $\log(t+1)$ under causal masking.

Model	**Normalized Entropy (Mean)**	**Diagonal Mass (Mean)**
Baseline (run_005)	0.9345	0.2209
Dense (run_059)	0.6345	0.3385

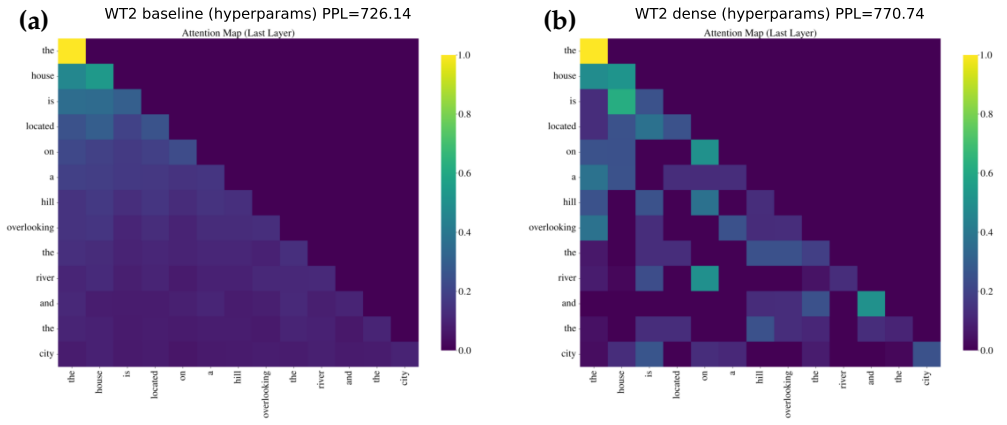

Figure 5. WT2: qualitative comparison of last-layer attention for the best baseline (run_005) and dense (run_059) models. Panels: (**a**) baseline (run_005); (**b**) dense (run_059).

Figure 6 shows the training/validation learning curves (loss and perplexity) for the best baseline and dense models on WT2, highlighting the convergence dynamics and stability across epochs.

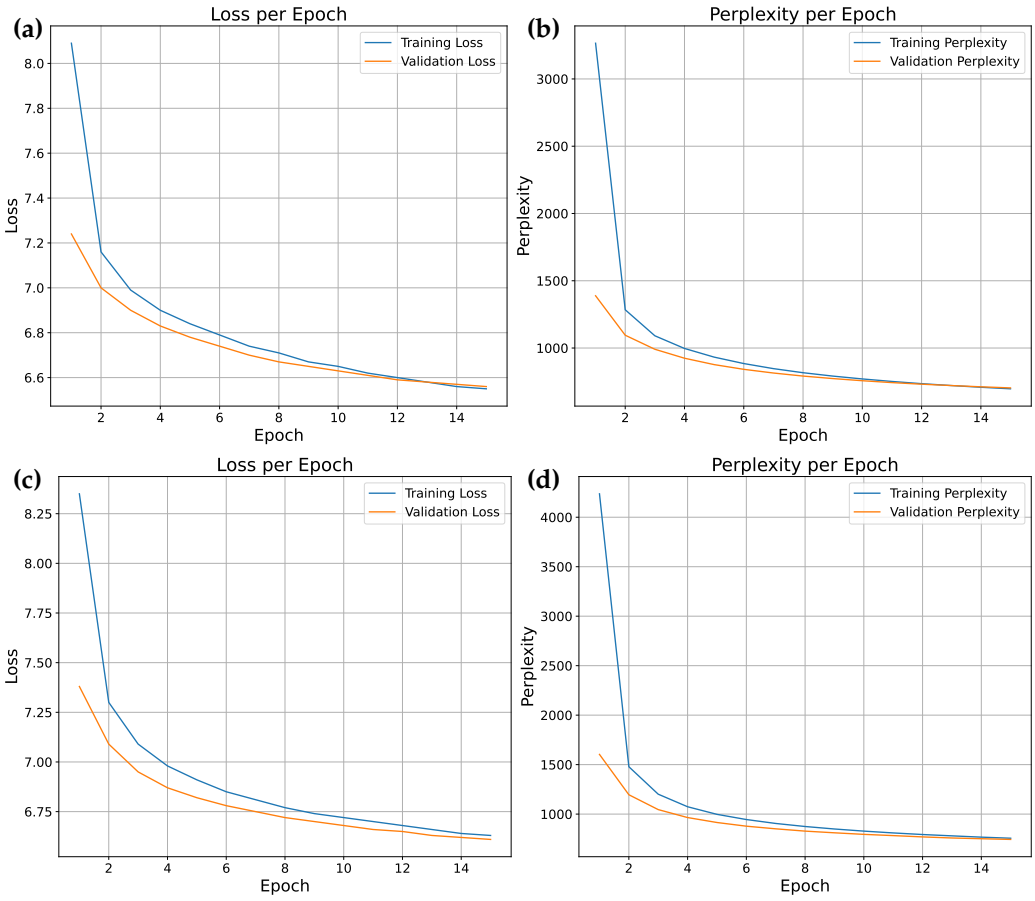

Figure 6. WT2: learning curves under the *same training recipe* (hyperparameters) regime. Panels: (**a**) baseline (run_005) loss; (**b**) baseline (run_005) perplexity; (**c**) dense (run_059) loss; (**d**) dense (run_059) perplexity. In each panel, training and validation curves are shown across epochs.

Across datasets and regimes, dense historical connectivity does not consistently improve perplexity. Under the *hyperparameters regime*, dense yields a small gain on PTB but degrades on WT2; under the *parameters regime* (parameter—budget constraint), the baseline remains better on both datasets (Table 3). On WT2, where we focus figures and probes, we find consistent PPL gaps in favor of the baseline (Table 4) and no evidence of improved cloze-style probe behavior in our limited probe set (Tables 5 and 6). Taken together, these findings from Grid A motivate a more targeted architectural ablation in Grid C to identify which capacity-allocation dimensions most affect dense-decoder performance.

4.3. Ablation Study (Grid C)

This subsection addresses RQ2 by isolating how dense-decoder perplexity varies with capacity-allocation factors (depth, width, feed-forward size, and head count) under a controlled training recipe. We now ablate architectural factors within the Dense Transformer Decoder using the alternative search configuration of Grid C (see Section 4.1). To isolate the effect of single factors, we vary one dimension at a time under the *hyperparameters* regime, keeping all other settings fixed. Test PPL is reported for each sweep. PTB one-factor tables are reported in detail below, and we additionally provide a compact WT2 one-factor summary to better localize the WT2 mechanism under matched controls. Figure 2 complements these tables with a cross-dataset depth–PPL view over the full grid.

Figure 2 further supports a robustness perspective, specifically, across the full grid (rather than only the best configuration), deeper models tend to yield lower PPL, with a clear dispersion that reflects interactions with width and head count. This motivates emphasizing depth and feed-forward capacity as primary levers for dense-decoder performance under the tested optimization envelope, while treating width-only scaling as less predictive in isolation.

Grid C yields the best models summarized in Table 8. The top configuration uses $d_{\text{model}} = 768$ with a deeper stack ($L = 10$), more heads ($h = 12$), and a wider feed-forward block ($d_{\text{ff}} = 3584$), achieving 320.85 PPL. Close variants that reduce d_{ff} or depth remain competitive, indicating that (within this regime) increasing depth and feed-forward capacity is a robust driver of better PTB perplexity for the dense decoder.

Table 8. Grid C (PTB, hyperparameters): top configurations by test PPL.

Rank	d_{model}	L	H	d_{ff}	PPL
1	768	10	12	3584	320.85
2	768	10	12	3072	321.06
3	768	10	12	2560	321.68

Varying width at fixed $L = 8$, $h = 8$, $d_{\text{ff}} = 3072$ yields only modest and non-monotonic changes (Table 9). Increasing d_{model} from 512 to 640 slightly improves PPL (331.84 to 330.93), while larger widths do not further improve under this setting. This suggests that, for the dense decoder under this optimization envelope, depth and feed-forward scaling are more reliable levers than width alone.

Table 9. Width sweep at $L = 8$, $h = 8$, $d_{\text{ff}} = 3072$.

d_{model}	PPL
512	331.84
640	330.93
768	333.64
832	332.85

Depth produces the strongest and most consistent improvements at $d_{\text{model}} = 768$, $h = 8$, $d_{\text{ff}} = 3072$ (Table 10). Increasing layers from 4 to 6 to 8 to 10 steadily lowers PPL from 354.87 to 343.54 to 333.64 to 322.69, supporting the view that dense historical access benefits from deeper stacks where representations can be progressively composed and reused.

Table 10. Depth sweep at $d_{\text{model}} = 768$, $h = 8$, $d_{\text{ff}} = 3072$.

n. layers	PPL
4	354.87
6	343.54
8	333.64
10	322.69

Changing the number of heads while keeping $d_{\text{model}} = 768$, $L = 8$, $d_{\text{ff}} = 3072$ yields an improvement when moving from $h = 8$ to $h = 12$ (Table 11). Despite reducing per-head dimensionality, additional heads may help the dense model represent diverse token dependencies more effectively under this setting.

Table 11. Heads sweep at $d_{\text{model}} = 768$, $L = 8$, $d_{\text{ff}} = 3072$.

n. heads	PPL
8	333.64
12	327.93

Finally, scaling the feed-forward block at $d_{\text{model}} = 768$, $L = 8$, $h = 8$ yields steady improvements (Table 12). Increasing d_{ff} from 2048 to 3584 lowers PPL from 335.17 to 331.13, consistent with the MLP being a primary channel for token-wise nonlinear transformation in word-level language modeling.

Table 12. Feed-forward sweep at $d_{\text{model}} = 768$, $L = 8$, $h = 8$.

d_{ff}	PPL
2048	335.17
2560	334.74
3072	333.64
3584	331.13

On WT2, Table 13 helps localize the failure mechanism more precisely. Depth remains the strongest improving factor within the dense family (from 857.62 at $L = 4$ to 770.74 at $L = 10$ under matched controls), whereas width and feed-forward sweeps are non-monotonic in this setting and increasing heads from 8 to 12 does not improve PPL. This indicates that WT2 behavior is still primarily governed by capacity-allocation effects rather than by dense connectivity alone. Importantly, even the best one-factor WT2 setting in this controlled slice (770.74) remains above the baseline WT2 result (726.14, Table 3).

Within Grid C, depth is the dominant driver of improved PPL for the dense decoder, with dataset-dependent contributions from feed-forward width and head count, while width-only scaling is weaker and often non-monotonic. These ablations contextualize Grid A by showing that denser cross-layer access can benefit from deeper composition, but does not by itself close the baseline–dense gap on WT2 under comparable training regimes.

Table 13. WT2 one-factor ablation summary (Grid C, dense, hyperparameters). Each row reports the best PPL obtained for each tested value while fixing the remaining architectural factors as indicated.

Factor	Fixed Setting	Best PPL by Tested Value
Width (d_{model})	$L = 8$, $h = 8$, $d_{\text{ff}} = 3072$	512: 774.51; 640: 774.75; 768: 788.07; 832: 790.55
Depth (L)	$d_{\text{model}} = 768$, $h = 8$, $d_{\text{ff}} = 3072$	4: 857.62; 6: 807.13; 8: 788.07; 10: 770.74
Heads (h)	$d_{\text{model}} = 768$, $L = 8$, $d_{\text{ff}} = 3072$	8: 788.07; 12: 792.50
Feed-forward (d_{ff})	$d_{\text{model}} = 768$, $L = 8$, $h = 8$	2048: 814.79; 2560: 799.59; 3072: 788.07; 3584: 809.54

4.4. Extended-Training Robustness Check

This subsection supports RQ1 by checking whether the baseline–dense conclusions remain stable when extending the training schedule, while keeping the training recipe fixed. We additionally reran the primary grid under an extended training schedule, keeping the protocol unchanged and increasing only the number of training epochs. As expected, longer training improves perplexity in absolute terms across datasets. Importantly, the qualitative conclusions remain stable on WT2. Particularly, the baseline continues to outperform the dense variant under both the *same training recipe* and *same parameter budget* regimes (Table 14), and cloze-style probes in the SVA/FACT/CTX families remain near-zero in Top-1/Top-5 accuracy at the selected context length.

Table 14. Extended-training robustness check (best test PPL). Δ denotes dense minus baseline (positive values favor the baseline).

Dataset	Regime	Baseline (15 ep)	Dense (15 ep)	Δ	Baseline (30 ep)	Dense (30 ep)	Δ
ptb	hyperparams	325.13	322.69	−2.44	291.07	296.24	+5.17
ptb	params	325.13	328.61	+3.48	291.30	301.40	+10.10
wikitext2	hyperparams	726.14	770.74	+44.60	660.79	706.39	+45.60
wikitext2	params	726.14	766.54	+40.40	660.79	710.40	+49.61

On PTB under the same-training-recipe (hyperparameters) regime, Table 14 shows a small reversal when extending training from 15 to 30 epochs: the dense decoder slightly improves over the baseline at 15 epochs, while the baseline becomes better at 30 epochs. Since the optimization recipe is held fixed when extending the schedule (i.e., we do not retune regularization or learning-rate schedules for the longer horizon), this behavior suggests that the dense parameterization can be more sensitive to late-training dynamics on the smaller PTB corpus, e.g., increased susceptibility to overfitting or co-adaptation among layers. We therefore treat the extended-training outcome as a robustness diagnostic rather than a definitive statement about convergence, and we complement the table with the corresponding validation curves for the best PTB runs at 30 epochs (Figure 7).

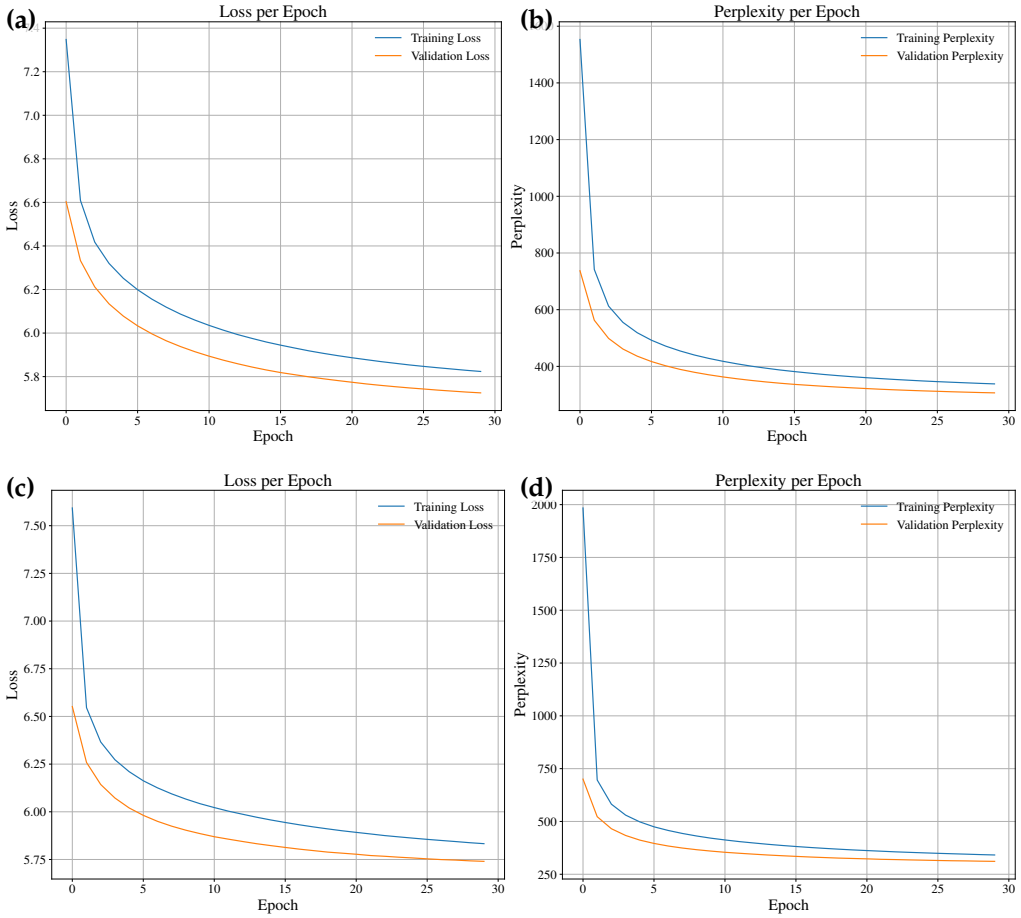

Figure 7. PTB extended training (30 epochs) under the *same training recipe* (hyperparameters) regime. Panels: (**a**) baseline (run_001) loss; (**b**) baseline (run_001) perplexity; (**c**) dense (run_023) loss; (**d**) dense (run_023) perplexity. In each panel, training and validation curves are shown across epochs.

4.5. Long-Form Structural Analysis (Zipf–RQA)

This subsection addresses RQ3 by comparing long-range structural descriptors of long-form generations from baseline vs. dense models under the WT2 hyperparameters regime. We complement likelihood-based results with a structure-oriented analysis of long-form generations. Using the WT2 hyperparameters-regime models `run_005` (baseline) and `run_059` (dense), we compute Zipf–RQA descriptors on $N = 150$ long texts per model (length 8000 words each). Figures 8 and 9 summarize the resulting distributions (standardized for comparability) and visualize the two families in a 2D MDS projection of the RQA feature space. Across the 12 descriptors, family-wise marginal distributions differ consistently, with Jensen–Shannon divergences in the range $[0.24, 0.32]$ (largest for LAM, T_1/T_2, $V_{\max}$, and DET), indicating systematic shifts in long-range structural organization rather than isolated outliers. As a complementary multivariate check, MANOVA on the first three PCA scores (explaining $\approx 87\%$ of total variance) confirms a strong family effect (Wilks' $\lambda = 0.382$, Pillai's trace 0.618; $F(3, 296) = 159.4$, $p < 0.001$), consistent with the distributional evidence above. Figure 10 provides representative recurrence plots (window 0) selected by the median-case criterion described in Section 3.4, with panel (a) for baseline and panel (b) for dense.

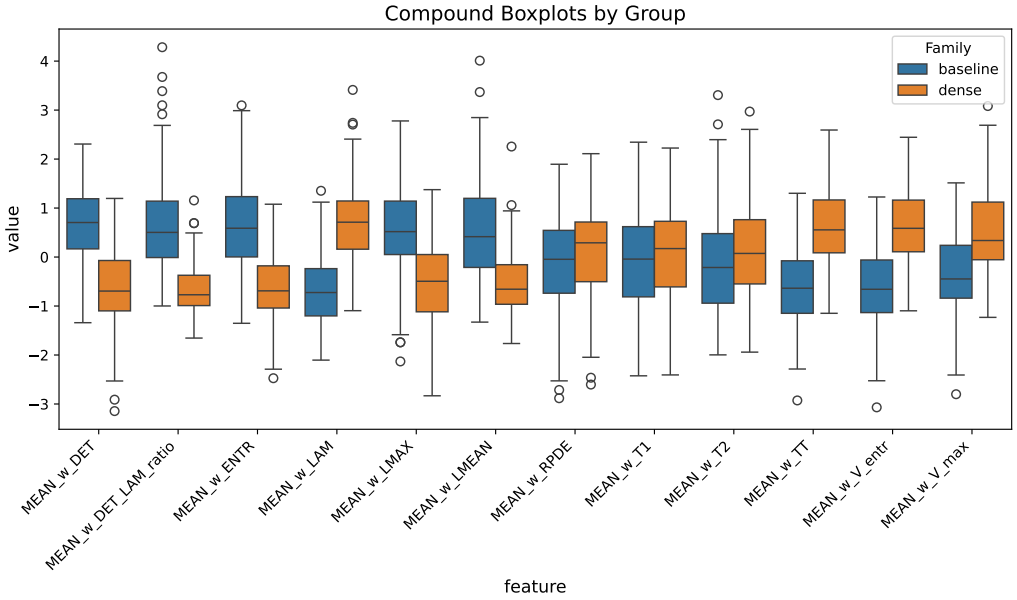

Figure 8. Long-form generations (WT2, $N = 150$ per family): compound boxplots of standardized Zipf–RQA descriptors. The distributions show consistent family-wise shifts across descriptors, indicating systematic differences in long-range structural organization between baseline and dense generations.

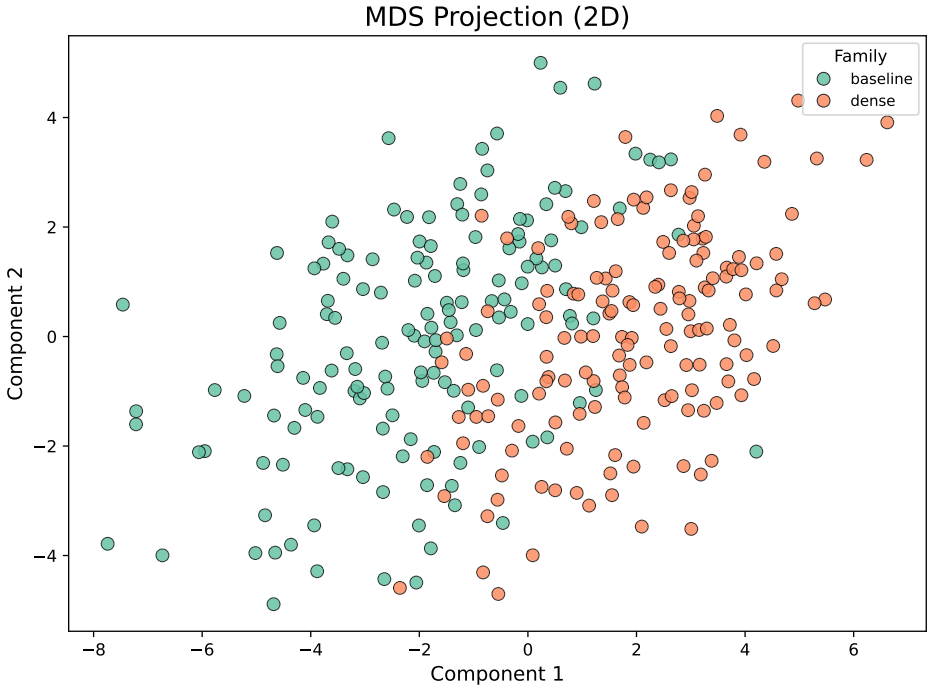

Figure 9. Long-form generations (WT2, $N = 150$ per family): 2D MDS projection of the Zipf–RQA descriptor space (distances computed in the standardized descriptor space). The separation between the two families provides a multivariate view consistent with the marginal distribution shifts.

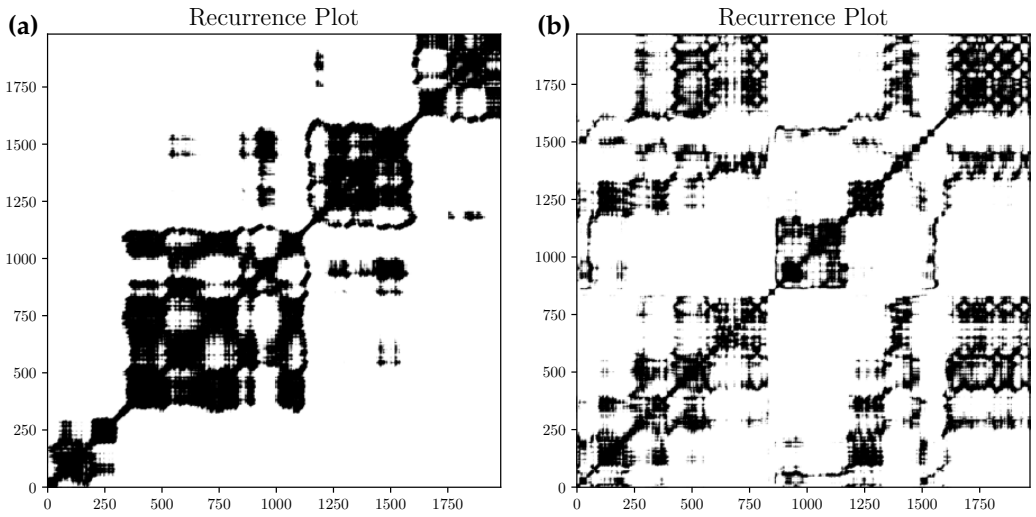

Figure 10. Representative recurrence plots (word-level Zipf-rank series, window 0) for WT2 long-form generations, selected as closest-to-median instances under the DET/LAM ratio to avoid cherry-picking while enabling qualitative inspection. Panels: (**a**) baseline (`run_005`); (**b**) dense (`run_059`).

5. Discussion

This study isolates the effect of dense historical connectivity in a decoder-only, causally masked Transformer setting under controlled optimization and comparison protocols [13–16]. Across datasets and fairness regimes (RQ1), densification does not yield consistent perplexity improvements over a fixed baseline decoder. We can state that any gains are dataset- and regime-dependent, and on WT2 the baseline remains competitive or better even when dense configurations are tuned within the same learning-rate grid. These results suggest that, at the tested scales, increasing cross-layer access via concatenation and projection is not by itself a reliable route to improved next-token prediction. Given that densification mechanisms for Transformers have been explored in prior works [15,16], the contribution of this study lies in a controlled, budget-aware evaluation (two fairness regimes) complemented by long-form structural analysis (Zipf–RQA), which together disentangle connectivity effects from capacity-allocation drivers. That said, further investigation still needs to be carried out to reach definitive conclusions.

The Grid C ablations clarify that much of the observed variation is explained by capacity allocation rather than connectivity alone (RQ2). In particular, increasing depth and feed-forward capacity provides the most consistent reductions in test perplexity for dense models, while width-only scaling is weaker and non-monotonic and head count yields smaller but measurable effects in specific settings. Consistently, the WT2 one-factor summary (Table 13) shows that depth improves dense-model PPL most clearly, whereas width/FF trends are non-monotonic and raising heads from 8 to 12 does not improve PPL, leaving a substantial gap to the WT2 baseline. This supports an interpretation in which optimization and representational capacity along depth/MLP channels dominate performance within the explored envelope, and dense connectivity interacts with (rather than replaces) these primary levers [3,4].

Beyond perplexity, we report diagnostics that help contextualize model behavior. Probing results show that both families struggle on the limited cloze-style probes considered, with Top-1/Top-5 rates at or near zero and differences mainly visible in rank and log-probability, indicating that such micro-tests are sensitive to prompt formulation and tokenization and should be interpreted as lightweight behavioral checks rather than definitive measures of linguistic competence [20,21]. Qualitative attention maps and learning curves further suggest broadly similar training dynamics, with the dense model showing a

more concentrated attention profile in the batch-averaged WT2 summary metrics (Table 7), while still yielding higher perplexity.

Finally, we extend the analysis to long-form generation (RQ3) using Zipf–RQA descriptors computed on long samples from the WT2 hyperparameters-regime best runs [7,23–26]. The descriptor distributions exhibit consistent family-wise shifts and a clear multivariate effect (Section 4.5), indicating that densification changes long-range structural signatures in generated text. Importantly, these structural differences do not automatically translate into improved likelihood. We know that in our setting, the dense model shows distinct recurrence/laminarity statistics while still exhibiting higher perplexity. This highlights a potential trade-off between altering global structural organization and improving predictive accuracy, motivating future work that jointly optimizes for likelihood and structure-aware objectives and that broadens the evaluation to multiple prompts, decoding strategies, and stronger human-validated quality measures.

Our conclusions are bounded by a controlled but limited experimental scope (PTB/WT2, word-level tokenization, and decoder-only causal masking), which we adopt to isolate architectural connectivity effects under fixed training budgets. In addition, we report results for a single fixed random seed to ensure strict run-to-run comparability under the same protocol; a multi-seed evaluation reporting mean $\pm$ std for the key comparisons is an important direction for future work [18]. We also did not include standardized efficiency measurements (e.g., throughput in tokens/s, peak memory footprint, or inference latency) under a fixed hardware and benchmarking protocol; quantifying these practical trade-offs is left for future work. Extending to larger corpora and scales and to subword tokenization is also an important direction for future validation. In the same spirit, while Zipf–RQA provides a descriptive view of long-range structural organization, we do not directly tie these shifts to external text-quality or degeneracy measures in the present study. Future work should therefore complement Zipf–RQA with a broader battery of inexpensive but informative indices (e.g., repetition rate, distinct-n, self-BLEU, and related diversity measures), ideally under multiple prompts and decoding strategies and, when feasible, with human-validated assessments.

6. Conclusions

We presented a controlled empirical study of concatenation-based dense historical connectivity in causally masked Transformer decoders, evaluated under two fairness regimes that separate optimization-recipe effects from parameter-budget effects. Within the explored regimes, dense connectivity does not consistently improve test perplexity over a standard baseline, while the ablation results indicate that depth and feed-forward scaling are the most reliable drivers of performance within the tested envelope. Complementary Zipf–RQA of long-form generations reveals systematic differences in long-range structural descriptors between baseline and dense models, indicating that connectivity can alter global text structure even when likelihood-based metrics do not improve.

Overall, the results suggest that dense historical connectivity, by itself, is not a reliable lever for reducing word-level PPL at the tested scales, and that progress is more strongly governed by capacity allocation and training protocol choices. Future work should extend these findings to broader datasets and scales, include multi-seed reporting, and refine structure-aware evaluation by complementing Zipf–RQA with multiple text-quality and degeneracy indices (e.g., repetition rate, distinct-n, self-BLEU, and related measures) to better translate structural shifts into directly interpretable quality signals.

Author Contributions: Conceptualization, E.D.S.; methodology, E.D.S.; software, E.D.S.; validation, E.D.S. and A.M.; formal analysis, E.D.S. and A.M.; investigation, E.D.S. and A.M.; resources, A.M. and A.R.; data curation, E.D.S.; writing—original draft preparation, E.D.S., A.M. and A.R.;

writing—review and editing, E.D.S., A.M. and A.R.; visualization, E.D.S.; supervision, A.R.; project administration, E.D.S. and A.R.; funding acquisition, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

BPTT	Backpropagation Through Time
CE	Cross-Entropy
CTX	Contextual Continuation
DET	Determinism
ENTR	Entropy
FACT	Factual Prompts
FFN	Feed-Forward Network
LAM	Laminarity
LLM	Large Language Model
LR	Learning Rate
MHA	Multi-Head Attention
MLP	Multi-Layer Perceptron
PPL	Perplexity
PTB	Penn Treebank
ReLU	Rectified Linear Unit
RQA	Recurrence Quantification Analysis
SVA	Subject–Verb Agreement
TT	Trapping Time
WT2	WikiText-2

Appendix A. Full Experimental Results

For completeness, this appendix reports the full Grid A summary of the best-by-validation checkpoints for each dataset, regime, and model family. Each row lists the evaluation metric (test PPL; lower is better), the optimization settings, and the corresponding run identifier used to locate the trained checkpoint and logs.

Table A1. Full Grid A summary (best checkpoints): dataset, model family, regime, and configuration details. Columns: PPL = test perplexity; LR = learning rate; Ep = epochs; BS = batch size; Seq = context length (T_c); d_{model}, L, H, d_{ff} = architecture; Run = run identifier; Params = parameter count.

Dataset	Model	Mode	PPL	LR	Ep	BS	Seq	d_{model}	L	H	d_{ff}	Run	Params
ptb	baseline	hyperparams	325.13	3×10^{-4}	15	128	128	768	8	12	3072	run_001	71.95 M
ptb	baseline	params	325.13	3×10^{-4}	15	128	128	768	8	12	3072	run_003	71.95 M
ptb	dense	hyperparams	322.69	3×10^{-4}	15	128	128	768	10	8	3072	run_023	118.68 M
ptb	dense	params	328.61	3×10^{-4}	15	128	128	528	8	12	2048	run_029	46.87 M
wikitext2	baseline	hyperparams	726.14	3×10^{-4}	15	128	128	768	8	12	3072	run_005	158.23 M
wikitext2	baseline	params	726.14	3×10^{-4}	15	128	128	768	8	12	3072	run_007	158.23 M
wikitext2	dense	hyperparams	770.74	3×10^{-4}	15	128	128	768	10	8	3072	run_059	204.96 M
wikitext2	dense	params	766.54	3×10^{-4}	15	128	128	624	8	12	2560	run_071	134.69 M

References

1. Shojaee, P.; Mirzadeh, I.; Alizadeh, K.; Horton, M.; Bengio, S.; Farajtabar, M. The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity. *arXiv* **2025**, arXiv:2506.06941. [CrossRef]
2. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
3. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling Laws for Neural Language Models. *arXiv* **2020**, arXiv:2001.08361. [CrossRef]
4. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; de Las Casas, D.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training Compute-Optimal Large Language Models. *arXiv* **2022**, arXiv:2203.15556. [CrossRef]
5. De Santis, E.; Martino, A.; Ronci, F.; Rizzi, A. From Bag-of-Words to Transformers: A Comparative Study for Text Classification in Healthcare Discussions in Social Media. *IEEE Trans. Emerg. Top. Comput. Intell.* **2025**, *9*, 1063–1077. [CrossRef]
6. De Santis, E.; Rizzi, A. Prototype Theory Meets Word Embedding: A Novel Approach for Text Categorization via Granular Computing. *Cogn. Comput.* **2023**, *15*, 976–997. [CrossRef]
7. De Santis, E.; Martino, A.; Rizzi, A. Human Versus Machine Intelligence: Assessing Natural Language Generation Models Through Complex Systems Theory. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 4812–4829. [CrossRef]
8. Traxler, M.J.; Boudewyn, M.; Loudermilk, J. What is Special About Human Language? The Contents of the "Narrow Language Faculty" Revisited. *Lang. Linguist. Compass* **2012**, *6*, 611–621. [CrossRef]
9. Altmann, E.G.; Cristadoro, G.; Esposti, M.D. On the origin of long-range correlations in texts. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 11582–11587. [CrossRef]
10. De Santis, E.; Martino, A.; Bruno, E.; Rizzi, A. 2025: A GPT Odyssey. Deconstructing Intelligence by Gradual Dissolution of a Transformer. In *Proceedings of the 2025 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–10. [CrossRef]
11. De Santis, E.; Martino, A.; Ronci, F.; Rizzi, A. LSTM in Recursive Feedback Loops: A Study on Textual Evolution and Complexity. In *Proceedings of the 2025 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–10. [CrossRef]
12. Jelinek, F.; Mercer, R.L.; Bahl, L.R.; Baker, J.K. Perplexity—A Measure of the Difficulty of Speech Recognition Tasks. *J. Acoust. Soc. Am.* **1977**, *62*, S63–S63. [CrossRef]
13. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2017; pp. 2261–2269. [CrossRef]
14. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway Networks. *arXiv* **2015**, arXiv:1505.00387. [CrossRef]
15. Ma, H.; Li, X.; Yuan, X.; Zhao, C. Denseformer: A dense transformer framework for person re-identification. *IET Comput. Vis.* **2023**, *17*, 527–536. [CrossRef]
16. Pagliardini, M.; Mohtashami, A.; Fleuret, F.; Jaggi, M. DenseFormer: Enhancing information flow in transformers via depth weighted averaging. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 136479–136508.
17. He, S.; Luo, H.; Wang, P.; Wang, F.; Li, H.; Jiang, W. TransReID: Transformer-based Object Re-Identification. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2021; pp. 14993–15002. [CrossRef]
18. Reimers, N.; Gurevych, I. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*; Palmer, M., Hwa, R., Riedel, S., Eds.; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 338–348. [CrossRef]
19. Tay, Y.; Dehghani, M.; Bahri, D.; Metzler, D. Efficient Transformers: A Survey. *arXiv* **2020**, arXiv:2009.06732. [CrossRef]
20. Ettinger, A. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 34–48. [CrossRef]
21. Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 2463–2473.
22. Linzen, T.; Dupoux, E.; Goldberg, Y. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 521–535. [CrossRef]
23. Zipf, G.K. *Human Behavior and the Principle of Least Effort*; Addison-Wesley: Boston, MA, USA, 1949.
24. Eckmann, J.P.; Kamphorst, S.O.; Ruelle, D. Recurrence Plots of Dynamical Systems. *Europhys. Lett.* **1987**, *4*, 973. [CrossRef]
25. Webber, C.L.; Zbilut, J.P. Dynamical Assessment of Physiological Systems and States Using Recurrence Plot Strategies. *J. Appl. Physiol.* **1994**, *76*, 965–973. [CrossRef]
26. Marwan, N.; Carmen Romano, M.; Thiel, M.; Kurths, J. Recurrence plots for the analysis of complex systems. *Phys. Rep.* **2007**, *438*, 237–329. [CrossRef]

27. Marcus, M.P.; Santorini, B.; Marcinkiewicz, M.A. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.* **1993**, *19*, 313–330.
28. Merity, S.; Keskar, N.S.; Socher, R. Regularizing and Optimizing LSTM Language Models. *arXiv* **2018**, arXiv:1708.02182.
29. Merity, S.; Xiong, C.; Bradbury, J.; Socher, R. Pointer Sentinel Mixture Models. *arXiv* **2016**, arXiv:1609.07843. [CrossRef]
30. Conneau, A.; Kruszewski, G.; Lample, G.; Barrault, L.; Baroni, M. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Gurevych, I., Miyao, Y., Eds.; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 2126–2136. [CrossRef]
31. Tenney, I.; Das, D.; Pavlick, E. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*; Korhonen, A., Traum, D., Màrquez, L., Eds.; Association for Computational Linguistics: Florence, Italy, 2019; pp. 4593–4601. [CrossRef]