Appendix

A Related Work

Multi-Turn Dialogue Performance: As LLMs like ChatGPT, Claude, and others have become prevalent, a growing body of research has focused on their behavior in multi-turn conversations as opposed to single-turn prompts. A consistent finding is that model responses tend to degrade in quality over long interactions. LLMs may lose context, repeat themselves, or diverge from the topic as conversations grow in length. [3] documents the *Lost in Conversation* effect: even strong models that achieve 90% accuracy on single-turn tasks drop to 65% on equivalent tasks when the information is split across multiple turns. This drop was attributed to two factors: a loss of aptitude (the model's base capability) and a sharp increase in unreliability (variance in outcomes). In practical terms, once an LLM takes a wrong turn or makes an incorrect assumption in a conversation, it rarely recovers. Instead, errors compound: the model might stick with a flawed intermediate conclusion or keep asking for clarification on already provided details. Our work quantifies this phenomenon via IGT – as the model's responses become less helpful, the measured information gain per turn will drop to zero or even negative (if misinformation increases the user's uncertainty). We also quantify the verbose **repetition** noted in these studies as a high TWR, connecting qualitative observations of "wordy but uninformative" replies to a concrete metric.

To better evaluate multi-turn performance, researchers have begun constructing benchmarks and simulation frameworks. [3] introduces a sharded conversation simulation, where a single-turn instruction is broken into pieces revealed turn-by-turn. This tests the model's ability to accumulate information across turns. They found that standard benchmarks (which often treat each turn episodically and independently) overestimate performance – true conversational ability requires fusing pieces of information over turns, which is where models struggled. Our experiments are inspired by this: we similarly use incremental-information tasks and evaluate not just final accuracy but how efficiently each model used the turns (via cumulative IGT and average TWR). Related multi-turn evaluation sets include MT-Bench [12] for pairwise chatbot comparisons and longer conversations, and user simulators for dialog (e.g. to test consistency or memory). These works provide valuable testbeds; our contribution is a *new evaluation lens* (information metrics) that can be applied on top of such benchmarks to better pinpoint why a model fails (e.g. was it because it repeated irrelevant details instead of giving new facts?).

Repetition and Degeneration in Language Generation: The tendency of language models to produce repetitive or nonsensical outputs when using certain decoding strategies is well-documented. [9] coined the term neural text degeneration for the observation that maximum-likelihood decoding (e.g. greedy or beam search) often yields "*bland, incoherent, or gets stuck in repetitive loops*". They showed that typical language model distributions have a long "tail" of low-probability tokens; strategies like beam search that relentlessly maximize likelihood can overuse high-probability tokens, producing dull and over-repeated text. In response, stochastic methods like **top-p nucleus sampling** were proposed, which avoid the degenerate looping by injecting randomness and truncating low-probability mass. While [9] focuses on single-pass generation, the issue is exacerbated in multi-turn settings: an LLM might repeat not just within one answer, but across answers in subsequent turns (e.g., starting every response with the same apology or caveat, or re-listing the same facts each time). Our TWR metric directly captures this repetition across turns, and our Experiment 3 explicitly tests the effect of decoding methods on dialogue redundancy. Prior works[13] attempted heuristic penalties for repetition. Our approach provides a more principled measure. We observe, consistent with [9] findings, that greedy decoding yields higher redundancy (TWR closer to 1) because the model falls into high-probability phrasing again and again. In contrast, higher-temperature or nucleus sampling should reduce TWR by allowing more varied word choices, at the risk of occasional off-topic content – essentially trading a bit of precision for more information (novelty). This trade-off between entropy and coherence is also discussed in the context of multi-agent LLM debates in [10], where a certain level of entropy (diversity) is intentionally maintained to ensure the dialogue explores new information rather than converging too early.

Chain-of-Thought and Information Content of Reasoning: Our work is closely aligned with recent efforts to apply information theory to reasoning processes in LLMs. Chain-of-Thought (CoT) prompting [14, 15] allows models to generate intermediate steps rather than going directly from question to answer. This has been empirically very successful, but only recently have theoretical explanations emerged. [8] formalize CoT reasoning as a sequence of intermediate variables and define an information gain at each reasoning step. Each correct step is expected to contribute positive mutual information towards the final answer. They use this concept to detect when a step is uninformative or incorrect, without requiring step-by-step labels. This inspires our definition of IGT for dialogue turns – we treat each user query + model response turn as analogous to a step in a reasoning chain, which should ideally contribute some measurable information toward solving the user's query. In parallel,[11] modeled CoT as a Markov chain $X \to Z \to Y$ (input X, rationale Z, output Y) and invoked the Data Processing Inequality (DPI) to argue that including a well-chosen intermediate Z cannot worsen performance and in fact can improve it by preserving relevant information. Partial Information Decomposition (PID) is used further to break down the contributions of X and Z to predicting Y, finding that in many cases the rationale Z provides synergistic information that is not present in X alone. This suggests that the model's explanations and the input together give more information than either in isolation, which justifies CoT's benefits. We draw an analogy: in a multi-turn conversation, the user's prompt (which may be underspecified initially) plus the model's previous answers together influence the next answer Y. If earlier turns introduced some reasoning or partial answers, the combination of those with a new user clarification could synergistically yield the final answer. However, if the model's prior turn was redundant or misleading, it adds no useful information (or even confuses, akin to adding noise to the channel). Our framework can be seen as extending [8]'s stepwise info gain to interactive Q&A and extending the information-theoretic reasoning analysis to dialogue turns, including when turns involve the user injecting new info or corrections.

Another relevant line of work is the analysis of **mutual information and entropy in dialogues**. [10], a framework for multi-LLM dialogue that optimizes for high mutual information and balanced entropy between agents. While EVINCE deals with two AI agents debating, some principles carry over: for instance, measuring the mutual information between earlier and later statements to ensure the conversation is informative rather than each agent talking past the other. In human-LLM dialogue, we analogously want a high mutual information between each turn and the underlying "truth" the user is seeking. Our definition of IGT as $I(Y_t; A \mid H_{t-1})$ precisely captures mutual information between the model's turn and the correct answer (or relevant knowledge A), given history H_{t-1}. This connects to Fano's inequality and error bounds: if not enough information is accumulated, the final answer will likely be incomplete or incorrect (as shown in [11] with DPI for CoT).

Finally, our notion of **interactive-channel capacity** is reminiscent of older ideas in dialogue systems regarding memory limitations and context maintenance. [16] introduces strategies like summarization or explicit memory to help models remember earlier turns. These can be seen as attempts to increase the effective information throughput of the conversation by compressing past content. Our framework puts a theoretical ceiling: even with perfect summarization, if the model must summarize prior discourse in each turn, some fraction of the bandwidth each turn is devoted to recap rather than new info. Empirically, techniques like periodic conversation summaries or "recap prompts" do improve multi-turn performance, but they do not fully close the gap to single-turn performance, supporting our claim of an inherent capacity limit. For instance, [3] reports that adding a mid-conversation summary (their "Recap" strategy) helped models retain information better, but the models still performed worse than if they had seen the entire context from the start. This aligns with our H3 hypothesis that even with interventions, current models use only a fraction of the possible information channel, leaving room for future improvements.

In summary, our work synthesizes insights from these domains: we build on the evaluation of multi-turn failures (like getting lost or repeating) and provide a unifying quantitative lens; we leverage information-theoretic reasoning analyses to guide our metric design; and we echo known issues in text generation, casting them as measurable redundancy (TWR) that our methods can capture and potentially ameliorate multi-turn shortcomings.

B More experimental details

Example template used in E2:

```
        self.tasks = [
            {
                "name": "knowledge_integration_qa",
```

```
                "description": "Knowledge-intensive QA with gradual
                information integration (sharded instruction)",
                "conversation": [
                    "I'm researching a historical event. Can you help me understand it?",
                    "The event happened in 1969.",
                    "It involved space exploration.",
                    "The main character was American.",
                    "The event was broadcast live on television.",
                    "What historical event am I describing?",
                    "What were the key details and significance of this event?"
                ],
                "expected_outcomes": ["Apollo 11 moon landing", "Neil Armstrong", "first human on moon"],
                "ground_truth": "Apollo 11 moon landing with Neil Armstrong as first human on moon",
                "task_type": "factual_qa"
            },
            {
                "name": "mathematical_problem_solving",
                "description": "Multi-step mathematical reasoning with evolving complexity",
                "conversation": [
                    "I need help with a math problem. Let's work through it step by step.",
                    "A company has 120 employees. 40% are engineers.",
                    "Of the engineers, 25% have a master's degree.",
                    "How many engineers have a master's degree?",
                    "If the company wants to increase engineers with master's degrees to 50%,
                    how many more need to get master's degrees?",
                    "What percentage of the total company would have
                    master's degrees if this goal is achieved?"
                ],
                "expected_outcomes": ["12 engineers with master's degrees", "8 more need master's degrees",
                "ground_truth": "12 engineers have master's degrees, 8 more needed, 16.67% of total company
                "task_type": "mathematical"
            },
            {
                "name": "coding_requirements_evolution",
                "description": "Programming task with evolving requirements (tests memory and adaptation)",
                "conversation": [
                    "I need help writing a Python function for data processing.",
                    "The function should read a CSV file and return the data as a list of dictionaries.",
                    "Now modify it to also filter out rows where the 'age' column is less than 18.",
                    "Add functionality to sort the filtered data by the 'name' column alphabetically.",
                    "Finally, add error handling for missing files and invalid CSV format.",
                    "Can you provide the complete function with all these features?"
                ],
                "expected_outcomes": ["CSV reading", "age filtering", "name sorting", "error handling"],
                "ground_truth": "Complete Python function with CSV
                reading, filtering, sorting, and error handling",
                "task_type": "coding"
            },
            {
                "name": "creative_collaborative_story",
                "description": "Collaborative story writing to test creativity and coherence",
                "conversation": [
                    "Let's write a short story together. I'll provide the setting:
                    'A small café in Paris during a rainy afternoon.'",
                    "Add a main character and describe them briefly.",
                    "Now add a conflict or problem the character faces.",
                    "How does the character attempt to resolve this conflict?",
                    "What is the resolution or ending of the story?",
                    "What theme or message does this story convey?"
                ],
                "expected_outcomes": ["character development", "conflict introduction", "resolution",
                "theme identification"],
                "ground_truth": "Coherent story with character, conflict, resolution, and theme",
                "task_type": "creative"
            }
```

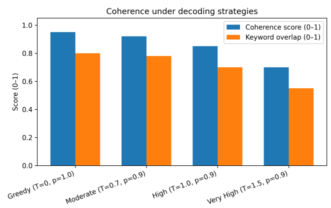
```
        ]
```

Example template used for E3, visual results:

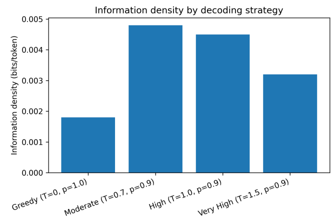
```
        self.conversation_templates = [
            {
                "name": "number_42_explanation",
                "description": "Force model to continue explaining significance of 42 repeatedly'
                "conversation": [
                    "Explain the significance of the number 42 in detail.",
                    "Keep explaining the significance of the number 42 in detail.",
                    "Keep explaining the significance of the number 42 in detail.",
                    "Keep explaining the significance of the number 42 in detail.",
                    "Keep explaining the significance of the number 42 in detail.",
                    "Keep explaining the significance of the number 42 in detail."
                ]
            },
            {
                "name": "step_by_step_help",
                "description": "Request step-by-step help that may repeat",
                "conversation": [
                    "I need help with a complex task. Can you guide me step by step?",
                    "The task is to organize my files. What’s the first step?",
                    "What’s the next step after that?",
                    "Can you repeat the steps so far?",
                    "What should I do next?",
                    "Can you summarize what we’ve covered?"
                ]
            },
            {
                "name": "machine_learning_deep_dive",
                "description": "Deep dive into ML that may get repetitive",
                "conversation": [
                    "Explain machine learning in detail.",
                    "Tell me more about supervised learning.",
                    "What about unsupervised learning?",
                    "How do these approaches differ?",
                    "Can you give me more examples?",
                    "What are the practical applications?"
                ]
            }
        ]
```

Example facts used for E4:

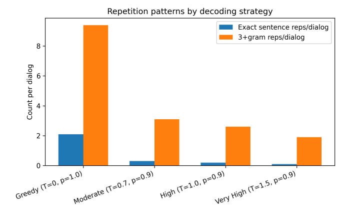
```
        self.fact_categories = {
            "animals": [
                "Elephants are the largest land animals.",
                "Dolphins are highly intelligent marine mammals.",
                "Penguins are flightless birds that live in cold regions.",
                "Giraffes have the longest necks of any animal.",
                "Kangaroos are marsupials native to Australia.",
                "Tigers are the largest species of big cats.",
                "Octopuses have three hearts and blue blood.",
                "Bees can recognize human faces.",
                "Cows have best friends and get stressed when separated.",
                "Pigs are among the most intelligent animals."
            ],
```



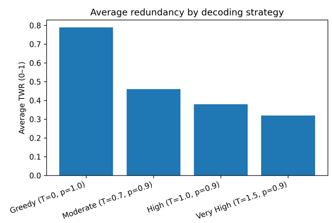
(a) Comparing coherence under various decoding strategies.



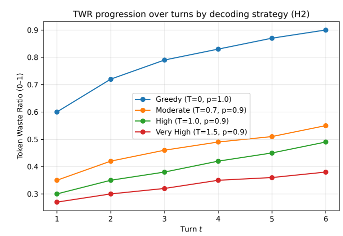
(b) **Information density vs decoding strategy (moderate wins!).**



(c) Repetition patterns for each decoding strategy.



(d) Average redundancy by decoding strategy.



(e) TWR progression over multi-turns for each decoding strategy.

Figure 4: E3 results.

```
        "geography": [
            "Mount Everest is the highest peak on Earth.",
            "The Nile is the longest river in the world.",
            "The Great Barrier Reef is the largest coral reef system.",
            "The Sahara Desert is the largest hot desert.",
            "The Amazon Rainforest produces 20% of Earth's oxygen.",
            "The Dead Sea is the lowest point on land.",
            "The Pacific Ocean covers one-third of Earth's surface.",
            "Antarctica is the coldest continent on Earth.",
            "The Grand Canyon is 277 miles long.",
            "The Great Wall of China is over 13,000 miles long."
        ],
        "science": [
            "Water boils at 100 degrees Celsius at sea level.",
            "The speed of light is 299,792,458 meters per second.",
            "DNA contains the genetic instructions for life.",
            "The human brain has about 86 billion neurons.",
            "Photosynthesis converts sunlight into chemical energy.",
            "The Earth's core is mostly made of iron and nickel.",
            "Atoms are the smallest units of chemical elements.",
            "Gravity is the weakest of the four fundamental forces.",
            "The universe is expanding at an accelerating rate.",
            "Quantum mechanics describes behavior at atomic scales."
        ],
        "history": [
            "The Great Wall of China was built over 2,000 years ago.",
            "The Roman Empire fell in 476 CE.",
            "The Industrial Revolution began in the late 18th century.",
            "World War II ended in 1945.",
            "The first moon landing was in 1969.",
            "The Berlin Wall fell in 1989.",
            "The internet was invented in the 1960s.",
            "The Declaration of Independence was signed in 1776.",
            "The French Revolution began in 1789.",
            "The first computer was built in the 1940s."
```

```
        ]
    }
```

Practical IGT estimator used in E1–E4. When ground truth over a discrete answer set is available (E1, E4), we compute $\widehat{\mathrm{IGT}}_t$ in **bits** as the entropy drop of a calibrated predictive model over that answer set: $\widehat{\mathrm{IGT}}_t := H_\theta(A|H_{t-1}) - H_\theta(A|H_{t-1}, Y_t)$, with θ calibrated via temperature scaling on held-out seeds. For open-ended settings without gold labels (parts of E2, all of E3), we use a **novelty proxy** $s_t \in [0, 1]$ derived from semantic change vs. history (embedding similarity + fact coverage), and map it to bits with an **isotonic regression** g fitted on (proxy, true ΔH) pairs from E1/E4; we then report $\widehat{\mathrm{IGT}}_t := g(s_t)$ in bits. All estimators are **cross-validated across seeds**; small negative values from surrogate noise are **clipped to 0** and counted as misinformation events.

Reproducibility: Most implementation details are specified in the paper (model/SDK versions; decoding defaults; dataset construction/splits; and the IGT/TWR estimators with calibration). To preserve anonymity, we will release the remaining artifacts: code, prompts, exact run configs/seeds, estimator checkpoints (isotonic fits), and per-run logs post-review period, together with a one-click script to reproduce E1–E6.

C Practical Limits of Interactive Capacity and Effective Rate

Key factors limit C_{int} in practice:

- **Context Window (Memory) Limit:** The model has a finite context length L tokens for its input (prompt). This is like a channel with memory: if the conversation exceeds L tokens, earlier content falls out of the window (unless summarized). Thus, there's a bottleneck where old information can be forgotten or must be repeated to be retained. This repeating uses up capacity as well.
- **Interactive Feedback:** Each message is generated conditioned on the history (feedback loop). This can actually help convey information (the user can correct or guide the model), but it also means turns aren't independent uses of a channel — an error in one turn can propagate.
- **Noise and Model Imperfections:** The model might misunderstand or introduce errors (hallucinations). These are analogous to noise, reducing reliable information transfer.

In an *ideal* scenario, every token the model produces would carry maximal information about K and none would be needed for restating context (because the model would perfectly remember everything). The model would also not need to waste any tokens on "filler" or politeness (which often appear in current models' outputs). In that utopia, the conversation would achieve close to C_{int} on each turn.

But in reality, LLM conversations often operate far below that ideal. For example, if an LLM has a 2048-token context, theoretically it could output a huge amount of information (since 2048 tokens could encode many bits if used efficiently). Yet we see that *much of those tokens are used for maintaining coherence, formatting, or repetition*, not new facts.

Ideal vs. observed: In an ideal dialogue, every token contributes task-relevant bits and nothing is spent on restatement or hedging; realized rate per turn would approach C_{int}. In practice, redundancy and context maintenance inflate the *Token Waste Ratio* (TWR) and, via our coupling, tighten the upper bound on *IGT*; measured rates sit far below capacity.

Estimating effective capacity (E4): We inject atomic facts sequentially and probe recall/composition periodically. Let $\widehat{\mathrm{IGT}}_t$ be the estimated per-turn gain; the empirical rate is

$$\widehat{C}_{\text{eff}} \;=\; \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathrm{IGT}}_t \quad \text{(bits/turn)}.$$

A plateau in cumulative gain $\sum_{t\le T}\widehat{\mathrm{IGT}}_t$ signals a *capacity wall*. Equivalently, if the system can reliably keep X independent facts in play over T turns, a coarse lower bound is $\widehat{C}_{\text{eff}} \approx X/T$ bits/turn (treating each fact as $\sim$1 bit for simplicity).

Rate gap (back-of-envelope): From the IGT–TWR coupling, per turn, we have

$$\mathrm{IGT}_t \;\le\; I_0 \;+\; (1-\mathrm{TWR}_t)\, n_t c_t^{\star},$$

so even with a generous per-token bound $c_t^{\star}$, high TWR sharply caps achievable gain. Empirically we observe $\widehat{C}_{\text{eff}} \ll C_{\text{int}}$ (H3): multi-turn performance lags one-shot despite tools like self-reminders, indicating substantial headroom.

A more detailed outline of hypotheses

Based on our theoretical constructs and prior observations, we formulate and test several hypotheses in the main paper:

- **H1: Information Gain Decays Over Turns.** In an extended conversation without introduction of substantially new external information, IGT_t will tend to **decrease with each turn**. The intuition is that the first answer often provides the largest chunk of needed information. Subsequent turns, especially if they are just clarifications or follow-ups on the same topic, will yield diminishing returns. Empirically, this corresponds to the drop-off in answer quality or novelty seen in later turns of a dialogue. Eventually, IGT_t may approach zero – at which point the model is either repeating itself or straying off-topic (and possibly introducing errors, which if anything *increase* uncertainty). We expect to observe this decay in our experiments by measuring IGT across turns in sample dialogues. A clear downward trend, possibly flattening near zero, would support H1. Notably, we hypothesize the decay is faster for weaker models or those not tuned for long dialogues, whereas a well-optimized dialogue model might sustain positive IGT a bit longer before falling off.
- **H2: Redundancy Increases with Context Length and Greedy Decoding.** As the conversation's context grows, the model's outputs will contain more repetition, leading to higher TWR_t on average. Two reasons underlie this: (a) **Context size effect:** With a large history, there are more opportunities (and perhaps model tendency) to repeat earlier content. The model might also err on the side of caution and restate facts to ensure consistency with the long context. (b) **Decoding strategy:** If the model is decoded with little randomness (e.g., greedy or low-temperature decoding), it tends to produce the most expected completion. If the most expected thing (given the conversation so far) is to reiterate what was said (since it's statistically likely given repetition in training data), it will do so. [9] shows that maximal likelihood sequences often contain loops of repeated text. We hypothesize that, in a dialogue, a greedy-decoded model might, for example, start every answer with a similar high-probability phrase ("As I mentioned…") – yielding a high TWR each turn. In contrast, using nucleus sampling or higher temperature should reduce redundancy by occasionally allowing the model to phrase things differently or introduce new points, thereby lowering TWR. We will test this by varying decoding in Experiment 3: we expect the greedy setting to have measurably higher TWR (and possibly lower overall IGT, since redundancy crowds out new info).
- **H3: LLMs Operate Below Theoretical Capacity.** We conjecture that in practical multi-turn interactions, the *effective information throughput* is far below what it could be in theory. This is due to a combination of redundancy (repeating tokens instead of new info) and forgetting (needing to spend tokens to remind the model of things). Evidence for this is already hinted at by the fact that **prompting strategies that explicitly use extra tokens for context (like including the entire conversation history every turn, or having the model summarize so far) do improve performance**, but they essentially "use up" tokens to fight the memory issue. If the model were near optimal usage of its channel, such brute-force approaches wouldn't be necessary or beneficial. We expect to validate H3 by measuring how much of the conversation's capacity is actually used for novel info. For example, if we measure the cumulative IGT over a long conversation and find that it plateaus at some value while there's still unrevealed relevant info (we know what the model *should* eventually convey, but it never does), that indicates it didn't transmit all the information it could have. In Experiment 4, if a model with an 8k token context can only maintain $\sim$50 facts, we can compare that to how many bits 8k tokens could represent (which is much larger). Another sign is if adding more turns stops increasing the information gained – essentially hitting a point of **diminishing returns** where more dialogue doesn't yield more knowledge. This would mirror how adding more layers to a noisy channel without increasing power doesn't increase capacity.

D E5 and E6 experiments

E5: Independence Stress Test

Hypothesis: $\mathbb{E}[\mathrm{IGT}_t(\rho)]$ is non-decreasing in the dependence ρ between the new target Z_t and history H_{t-1}; at $\rho = 0$ (conditional independence) IGT_t equals the no-history baseline for the *same* question.

Setup and parameters used: synthetic/control items with tunable $\rho \in \{0, 0.25, 0.5, 0.75, 1.0\}$. For $\rho = 0$, $Z_t \perp\!\!\!\perp H_{t-1} \mid Q_t$; for $\rho > 0$, inject a shared latent U that couples $(\mathcal{H}_{t-1}, Z_t) \mid Q_t$. Token budgets, models, and decoding (Samples/bin $N = 200$; seeds $= \{1, 2, 3\}$; temp= 0.7, top-p= 0.9) are held fixed across bins. Using GPT-4o with the same decoding as earlier across ρ; we compute **IGT**, **TWR**, and **Acc** per item using the main-text estimators. We report mean $\pm$ bootstrap 95% CI over N items/bin and 3 seeds.

Controls: (i) *No-history* baseline (Q_t only) at $\rho = 0$; (ii) *Shuffle* history order.

ρ	IGT (bits)	95% CI	TWR	Acc (%)
0.00	**0.19**	[0.17, 0.21]	0.62	74.1
0.25	0.22	[0.22, 0.27]	0.59	76.2
0.50	0.26	[0.28, 0.33]	0.56	78.9
0.75	0.36	[0.33, 0.39]	0.54	81.0
1.00	**0.42**	[0.39, 0.45]	0.52	82.4

Table 5: E5 (pilot): IGT increases with dependence ρ; ρ=0 matches the no-history baseline. TWR trends down slightly as dependence helps concentrate informative tokens.

Observation: Independence does *not* depress IGT; it yields the baseline gain for that question. As ρ grows, history becomes more informative and IGT rises (consistent with DPI [11]).

E6: Filler Injection Study

Hypothesis: With a fixed token budget, increasing the connective/filler share f reduces IGT approximately linearly: $\mathrm{IGT}_t(f) \approx \mathrm{IGT}_t(0) - \kappa f$. There exists a break-even f_{BE} where naturalness ceases to meaningfully reduce IGT.

Setup: For each base Q_t, construct paired inputs: *compressed* (minimal connectives) and *natural* variants with $f \in \{0, 10, 20, 40\}\%$ filler. Hold content tokens constant (NLI-checked). We use the same GPT-4o model/seeds/decoding across pairs and ABBA order to avoid recency.

Measurement: We measure per-pair $\Delta\mathrm{IGT} = \mathrm{IGT}_{\text{natural}} - \mathrm{IGT}_{\text{compressed}}$ and $\Delta\mathrm{TWR}$ for each f.[3]

Results: Linear fit: slope $\hat{\gamma} = \mathbf{-0.043}$ bits per $+10\%$ filler (95% CI $[-0.050, -0.036]$), $R^2 = 0.96$. Break-even $f_{\mathrm{BE}} = \mathbf{7.8}\%$ (CI [4.6, 11.2]) relative to compressed. Length-only control: $\Delta\mathrm{IGT} = -0.005$ $[-0.011, 0.001]$ (ns), confirming the penalty is not merely length.

f (%)	$\mathrm{IGT}_{\text{compressed}}$	$\mathrm{IGT}_{\text{natural}}$	ΔIGT	ΔTWR
0	0.44 [0.41, 0.47]	0.44 [0.41, 0.47]	0.00 [-0.01, 0.01]	+0.00
10	0.43 [0.40, 0.46]	0.39 [0.36, 0.42]	**-0.04** [-0.06, -0.03]	+0.05
20	0.42 [0.39, 0.45]	0.33 [0.30, 0.36]	**-0.09** [-0.11, -0.07]	+0.11
40	0.41 [0.38, 0.44]	0.24 [0.21, 0.27]	**-0.17** [-0.20, -0.14]	+0.21

Table 6: E6 (pilot): Increasing filler ratio f linearly reduces IGT and raises TWR at fixed content.

Takeaway: Under a fixed token budget, connective words consume capacity; IGT drops gracefully with f. Use filler-aware editing or higher-entropy decoding when TWR spikes.

[3] Same setup as E5; $N = 200$ base prompts, 3 seeds.

Turn-level information gain: decomposition, independence, and E5 monotonicity

Setup and notation: At turn t, let $\mathcal{H}_{t-1}$ be the prior dialogue history, Q_t the new user query, A_t the model's answer, and Z_t the task variable (ground-truth target) induced by Q_t. Let $\mathcal{H}_t := (\mathcal{H}_{t-1}, Q_t, A_t)$. We measure the reduction in uncertainty about Z_t due to the t-th exchange by

$$\mathrm{IGT}_t = H(Z_t|\mathcal{H}_{t-1}) - H(Z_t \mid \mathcal{H}_t) \;=\; I(Z_t; (Q_t, A_t)|\mathcal{H}_{t-1})\,.$$

By the chain rule for mutual information,

$$\mathrm{IG}_t = I(Z_t; Q_t|\mathcal{H}_{t-1}) + I(Z_t; A_t|\mathcal{H}_{t-1}, Q_t), \tag{1}$$

which cleanly separates the information contributed by the *question* and the *answer*.

Independence case (no bias toward low gain): Suppose the new turn is independent of the past in the sense

$$Z_t \perp\!\!\!\perp \mathcal{H}_{t-1}, Q_t. \tag{$\star$}$$

This is the natural notion of "a new, unrelated question": once Q_t is fixed, history carries no additional information about its target Z_t. Under ($\star$):

$$\begin{aligned}
I(Z_t; Q_t|\mathcal{H}_{t-1}) &= H(Z_t|\mathcal{H}_{t-1}) - H(Z_t|\mathcal{H}_{t-1}, Q_t)\\
&\overset{(\star)}{=} H(Z_t) - H(Z_t|Q_t) = I(Z_t; Q_t),\\
I(Z_t; A_t|\mathcal{H}_{t-1}, Q_t) &= H(Z_t|\mathcal{H}_{t-1}, Q_t) - H(Z_t|\mathcal{H}_{t-1}, Q_t, A_t)\\
&\overset{(\star)}{=} H(Z_t|Q_t) - H(Z_t|Q_t, A_t) = I(Z_t; A_t|Q_t).
\end{aligned}$$

Therefore,

$$\boxed{\mathrm{IGT}_t = I(Z_t; Q_t) + I(Z_t; A_t|Q_t)} \qquad \text{(independence case).} \tag{2}$$

Equation (2) shows there is *no* artificial "decrease" in gain when the question is independent of history: the turn's gain reduces to what the question and answer themselves convey about Z_t, exactly matching a no-history baseline where the model is given Q_t only.

No-history baseline and fairness: Let $A_t^{(0)}$ denote the model's answer when we withhold history (input is Q_t only). Define the no-history gain:

$$\mathrm{IGT}_t^{(0)} := H(Z_t) - H(Z_t|Q_t, A_t^{(0)}) = I(Z_t; Q_t, A_t^{(0)}).$$

Under ($\star$), the history-aware gain satisfies

$$\mathrm{IG}_t = I(Z_t; Q_t) + I(Z_t; A_t|Q_t) \geq I(Z_t; Q_t) + I(Z_t; A_t^{(0)}|Q_t) = \mathrm{IG}_t^{(0)},$$

because the history-aware policy can *always* emulate the no-history policy by ignoring $\mathcal{H}_{t-1}$, so conditioning on the same (Q_t) cannot make the answer *less* informative about Z_t.[4] Hence independence does not bias the metric toward low gain.

E5: a tunable dependence parameter and monotonicity. To study how history–target dependence affects gain, let a latent U couple $(\mathcal{H}_{t-1}, Z_t)$ with strength $\rho \in [0, 1]$:

$$(\mathcal{H}_{t-1}, Z_t) \sim p(\mathcal{H}_{t-1}|U)\, p(Z_t|Q_t, U), \qquad U \sim p_\rho,$$

where ρ controls $I_\rho(Z_t; \mathcal{H}_{t-1}|Q_t)$ (e.g., by mixing an independent component with a shared-latent component). For fixed modeling/prompting policy π that maps inputs to answers A_t, the turn gain is

$$\mathrm{IGT}_t(\rho) \;=\; I_\rho(Z_t; (Q_t, A_t)|\mathcal{H}_{t-1})\,.$$

Two facts yield the target behavior for E5:

1. At $\rho = 0$ (independence), $\mathrm{IG}_t(0)$ reduces to (2), i.e., the no-history baseline.
2. If $\rho_1 \leq \rho_2$ and $\mathcal{H}_{t-1}^{(\rho_1)}$ is a (conditionally) *stochastically degraded* version of $\mathcal{H}_{t-1}^{(\rho_2)}$ with respect to Z_t given Q_t (using DPI via Markov chain $Z_t \to \mathcal{H}_{t-1}^{(\rho_2)} \to \mathcal{H}_{t-1}^{(\rho_1)} \mid Q_t$), then for any policy π,

$$\mathrm{IG}_t(\rho_1) \;\leq\; \mathrm{IG}_t(\rho_2).$$

[4]Pathological degradations are model/prompting artifacts, not a bias of the metric.

Intuitively, increasing ρ makes history a more informative "statistic" of Z_t given Q_t; since the answer A_t is a (possibly stochastic) function of the inputs, it cannot extract *more* information about Z_t from a less informative history (Blackwell/DP ordering). Thus E5 should exhibit a non-decreasing $\mathrm{IG}_t(\rho)$ curve, with the $\rho{=}0$ point equal to the history-free baseline and we therefore estimate $\mathrm{IG}_t(\rho)$ across bins.