ONLY THE STORY COMPLETES THE GRID: HAND-AUTHORED AND GRAMMAR-GENERATED PUZZLES FOR NARRATIVE AUGMENTED REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

ARC-style benchmarks exclude language by design; language-annotated variants show a written description can stand in for visual examples. Neither tests the widely made claim that narrative scaffolds reasoning through novel problems. We introduce NARC, Narrative Augmented Reasoning Challenges: sequences of colour grids, one masked, with a short story as the clue. The story never mentions grids: it helps only when read figuratively, a character as a bar, a mood as its height. A puzzle has the NARC property for a model when neither channel alone yields the masked grid and both together do. We evaluate 633 puzzles on eight open-weight models with an ablation battery separating lexical, order and form dependence. In 69% of narrative-necessary cases the clue's keywords are not enough; grids sufficing is a puzzle defect for 2% and a model fact for 32%. A stronger closed model keeps the property on 47% of 226 human-authored puzzles, depending less on the story's form and more on its words. We then present the Narrative Hierarchy Model, a dual-surface grammar whose every event rewrites into a sentence and a grid transition. It generates 21,864 puzzles from TinyStories, the property certified by construction, plus a control text faithful to the grids; data and generator will be released. Stating the rendering conventions outright does not substitute for the story: what the models lack is the event chain, not the code. Models trained from scratch learn the control text from four to five times fewer puzzles than the story, and paraphrases with clean labels train like the story: the cost is wording, not labels. What a narrative contributes is event structure; what it costs a learner is the variety of the surface that carries it.

1 INTRODUCTION

Suppose you are shown the three grids of Figure 1 and asked to draw the fourth. Nothing in the grids says what the fourth should be: the grey bar could climb again, the orange bar could fall, either could leave. Now read the clue. If the grey bar is Ally and a bar's height is her mood, the fourth grid follows. The story never mentions grids, bars or heights. It fixes the picture only for a reader who can take a character for a bar and a mood for its height. Other puzzles demand harder abstractions, but the capacities they call on are the same. That is the idea of a Narrative Augmented Reasoning Challenge (NARC) puzzle.

NARC puzzles serve two research purposes. One is for narrative theorists, who have argued for decades that stories are a technology for reasoning, in science as elsewhere (Morgan et al., 2022; Johnson et al., 2023), but on a case-study basis that invites the objection that the story was fitted to the answer afterwards. NARC serves as a testing ground: we write a puzzle, then shuffle the grids or strip the clue to its keywords and read off what the narrative was doing.

Similarly, computer scientists surmise that narratives (language) scaffold reasoning through novel problems. The family that defines abstract visual reasoning for machines, ARC and its successors, excludes language by design (Chollet, 2019; ARC Prize Foundation, 2026), yet human subjects report using narratives when solving those tasks, especially ARC-AGI-3 (Machine Learning Street Talk, 2026), and language-annotated variants show that natural-language instructions can stand in for visual examples: in LARC, a description written by one person lets another complete most ARC

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

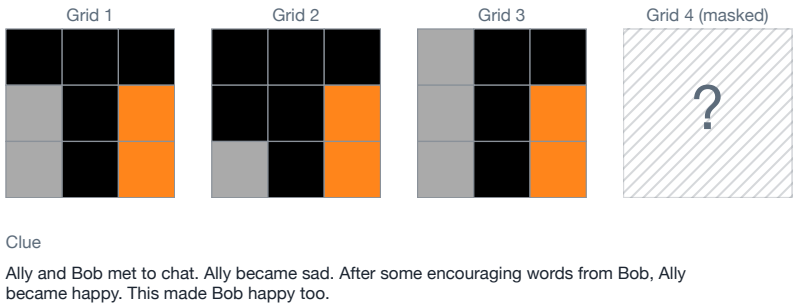


Figure 1: A hand-authored NARC puzzle (*Chat*); the fourth grid is masked. Read figuratively, the grey bar is Ally, the orange bar is Bob, and a bar's height is its mood. The grids alone leave the fourth grid open; the clue's last sentence fixes it. Two of eight open models solve it only with the clue, and both fail every shuffle and every keyword version of it. Answer in Appendix A.

tasks (Acquaviva et al., 2022). But why? What is it about language, and narratives in particular, that purportedly enable reasoning through novel problems? And how can we test that?

The second research purpose is to make progress on testing an answer to these questions: narratives utilize the hierarchical structure of language to facilitate metaphorical transfer of knowledge across domains via synonymy. In related work, the Random Hierarchy Model describes compositional data as a latent tree with synonymous productions at every level, and predicts the data a learner needs from a variable's depth and the synonymy below it (Cagnetta et al., 2024; Cagnetta & Wyart, 2024). Our extension, the Narrative Hierarchy Model (NHM), writes stories under one grammar with two surfaces, text and grid-pictures, so that what a story asserts is known in advance and what the pictures leave open under the mask is certified by design. It also separates two capacities a solver needs: the narrative supplies a structure of events, and figurative reading maps that structure onto coloured cells. Hand-authored puzzles risk confounding them; generated puzzles fix the pictorial mapping and vary only the text.

In this paper we make four contributions. (i) We introduce NARC and the NARC property as a relation between a puzzle and a model, with a diagnostic battery that separates dependence on the clue's words from dependence on its order and its form (i.e. dependence on the *narrative*). (ii) On 633 hand-authored puzzles and 8 open-weight models, the clue's own words are not enough in 69% of the cells that needed the clue, and grids sufficing is a defect of the puzzle for 2% of them and a fact about the model for 32%. A stronger closed model keeps the property on 47% of 226 human-authored puzzles but leans less on the clue's form and more on its words: what the narrative says still matters, how it is said less so. Together, (i) and (ii) demonstrate a method for generating the phenomenon we wish to study. (iii) We present the NHM, a dual-surface grammar in which each event rewrites into both a sentence and a grid transition, and generate 21,864 puzzles from TinyStories with the property certified by construction and a control text faithful to the grids. The extractor behind them is measured and not assumed: 92% of its events and 68% of its puzzles are judged faithful, and three open-weight judges agree on 11,049, the clean set later experiments draw from. (iv) Training small models from scratch, we measure what each wording of the same events demands. A plain rendering of the event chain, the control text, is learned from four to five times fewer puzzles than the TinyStory version. A paraphrase of that rendering in TinyStories style trains like the story, and puzzles whose answer grids the judges call wrong train no worse than the ones they call right: what the story costs in data is the variety of its wording, not errors in its (latent) labels. Position matters separately. A masked middle grid, whose neighbours are both visible, asks which of two implied events came first, and it stays harder than a masked ending at every budget and after its share of training is matched, though probes recover the masked event equally well in both. Hence, difficulty follows an event's position in the story, not how often that position is trained. Indeed, both the diagnostic battery on hand-authored puzzles and the learned representations trained on grammar-generated ones tell the same story: narratives supply the (event) structure that enables solutions to novel puzzles.

Section 2 defines the format, the relation and the battery, and reports what the models do with the hand-authored puzzles. Section 3 then builds the grammar to automatically generate a set of puzzles, Section 4 NARC selection mechanism to select from the generated set, and Section 5 trains from scratch and asks what those models represent. Section 7 takes stock.

2 NARC: THE HAND-AUTHORED BENCHMARK

Figure 1 is an example NARC puzzle: three to four grids on the ten-colour ARC palette tell a small story, one of them is masked, and a short story clue accompanies the grids. The solver produces the masked grid, and grading is exact. The grids are a sequence (not a set) so the clue is read against their order, and whether the order does any work is itself tested below. We run every puzzle under three conditions: grids only, narrative only, and both. A (puzzle, model) pair has the *NARC property* when the model reconstructs the masked grid from both channels and fails with either alone. Each condition rules out one explanation of a success with both. If the grids alone suffice the clue was decoration; if the clue alone suffices it was a solution in words; the NARC property holds only when both are necessary. Note, the NARC property belongs to the puzzle-model pair. Section 6 places NARC among ARC and its language-annotated variants. The corpus holds 633 puzzles (319 public): 229 written by people alone, and 404 with a language model somewhere in the making, every one of them reviewed by a person (Appendix A).

"The grids alone suffice" is a fact about the puzzle or a fact about the model. If every model in a pool of eight solves a puzzle from its grids, the defect is likely the puzzle's; if three do and five do not, what varies is the capabiliteis of the models. We call the vector of outcomes across the pool the puzzle's *cross-model profile*. When grids sufficient on none, narrative sufficient on none too, and both necessary on at least one model, we have reason to believe that the underdetermination the NARC format is built for holds across the pool. Otherwise, the clue may under-specify the answer, or the puzzle may simply be too hard for these readers (the story is fine and its grids fair, with the property waiting on a better model, or human). Absence of the property is therefore never by itself evidence against a puzzle, and Section 2.2 brings in a stronger reader to help tell the two apart in hand-authored puzzles. The benchmark's first job is to establish the attribution of the NARC property. In doing so, we demonstrate the viability of this method for selecting among automatically-generated puzzles (since a profile picks out the pairs in which the narrative was essential), and those are the puzzle-model pairs an interpretability study can open (which is why the pool is open-weight).

The property says that this model needed this narrative to complete this sequence. It does not say why, and four questions are open. Was it the story's order that did the work, or only its content? Its words, which might have primed the answer, or its form, which might have constrained it? Would any story have done, or only this one? And does the property belong to the puzzle or to the cell, the position we masked and the clue we wrote? Table 3 lists one test per question, each re-running the winning *both* condition with one thing changed. Two of them carry the results below. Shuffling the grids while keeping the clue asks about order: a cell is *strong* when no shuffle solves it and *weak* when every shuffle does. Replacing the clue by its alphabetised content words asks about form: a cell is *narrative* when no keyword list solves it and *lexical* when every one does. The others vary the clue, move the masked position, re-encode the grids, and drop the mask altogether to ask for the story order, the grid analogue of sorting jumbled photographs (Agrawal et al., 2016). Every trial is one sample at temperature zero, parsed in two passes with a fixed extractor, with $K = 3$ for the order and form tests; the protocol is in Appendix B.

2.1 IDENTIFYING THE NARC PROPERTY ACROSS MODELS

Across the eight models the NARC rate over all classification cells runs from 17% to 29%, and size does not order it: the 27B Qwen 3.8 has the highest rate, ahead of models four times its size (Table 5, Appendix C). Read the cross-model profiles of the 617 puzzles with a verdict from all eight (Table 4, Appendix C): grids suffice on all eight for 15 puzzles (2%), on one to seven for 197 (32%), and on none for 405 (66%), while 203 (33%) are unsolved with both channels by every model. Roughly put, two puzzles in a hundred are broken, a third are beyond this pool, and for the rest whether the story is needed depends on who is reading it.

A diagnostic battery adds further understanding of the dependence on narrative. Table 1 sorts the 2,163 narrative-necessary cells by whether and how they break. The clue's mere key words (i.e. 'lexical') are not enough in 68.7% of cells and are enough in 8.8%: in three of four cells where the story was needed, its words were not. Grid order was necessary in 41% and incidental in 19%; fewer than half the cells that need the clue's narrative form also fail every grid shuffle. That is, the structure that constrained the answer was usually not the sequence. Rather, the finding this paper carries forward is that it's the clue's (latent) narrative form that explains the solving ability, and is what Section 5 meets again as the surface variety a learner pays data for. Appendix C adds the per-model table, the 34-puzzle core that is NARC for every model, and repeats at temperature zero showing that the NARC rate is stable across runs while single-cell verdicts flip about one in six; Appendix D adds three smaller design studies, on ordering, on narrative facets and on grid re-encoding.

Table 1: Ablation typology over the 2,163 narrative-necessary cells of the eight open models, as percentages. Rows: order sensitivity under grid shuffling (strong = no shuffle solves). Columns: form sensitivity under keyword ablation (narrative = no keyword list solves). The top-left corner is the deepest tier, where both order and form were necessary; the bottom-right is keyword priming. Counts in Table 6; the stronger-model comparison in Table 10.

	narrative	partial	lexical	total
strong	28%	10%	2%	41%
partial	29%	8%	3%	40%
weak	12%	4%	3%	19%
total	69%	23%	9%	100%

2.2 A STRONGER CLOSED MODEL: THE PROPERTY PERSISTS, FORM DEPENDENCE FALLS

Is the NARC property an artefact of a weak pool? That is, would a "smarter" model successfully infer the masked grid from the other grids along, or from the narrative alone? We ran one stronger closed model, Claude Fable 5.1, closed book, on all 226 human-authored puzzles (Table 9, Appendix D). It keeps the property on 107 of them (47%), and on the 157 drafts that were never public at the same rate as on the public set, which is what bounds contamination. Of the 79 puzzles unsolved by all eight open models it makes 32 NARC, so part of that row was the pool's ceiling and part the clue.

The property persists, then, but its typology moves (Table 10, Appendix D). On the same puzzles, form-necessary cells fall from 72% to 47%, lexical cells rise from 8% to 27%, and the deepest tier falls from 50% to 24%. So this model still needs what the story says, but less often how it says it. Human solvers on the public site, an exploratory comparison over 821 anonymous sessions, solve 55% of attempts with the clue and 29% without (Appendix E).

3 THE NARRATIVE HIERARCHY MODEL

The hand-authored puzzles establish a phenomenon: for these models, on these puzzles, the story is crucial, and its narrative form is more than its words. But, they cannot say why, for three related reasons. First, there is no ground truth for what a clue asserts, so a story the model failed to read cannot be told apart from one that never said enough. Second, the typology of Section 2.1 sorts outcomes without controlling their causes; we must infer these. And third, 633 puzzles cannot train a model from nothing, so every result so far is about readers that already read. The Narrative Hierarchy Model addresses each of these concerns. It is a grammar whose every production writes a sentence and a scene change together, so a generated puzzle carries its own latent structure, its property is certified before any model sees it, and its text can be varied at a scale that supports training from scratch. Here our question shifts. We no longer ask whether a narrative resolves underdetermination, which Section 2 showed. Rather, we identify puzzles that have this NARC property, and then ask what a learner must recover from a narrative to solve them, and how much data that takes as the textual surface varies.

3.1 A LATENT TREE WITH TWO SURFACES

The Random Hierarchy Model (Cagnetta et al., 2024) is an account of what makes language learnable. Data are generated by a tree of depth L: a class at the root expands, level by level, into s sub-symbols, each chosen among m synonymous productions over a vocabulary of size v. Deep networks learn such data level by level, the examples needed to recover a variable grow with its depth and with the synonymy below it, roughly as $v\,m^L$, and a model trained on more text represents deeper levels (Cagnetta & Wyart, 2024). The idea of the latent tree itself is old, and not new to narratives either: story grammars decomposed stories into episodes and events decades earlier (Rumelhart, 1975; Mandler & Johnson, 1977; Thorndyke, 1977), and to comprehend a text, on the situation-model account, is to build a model of what it describes (Kintsch, 1988; Zwaan & Radvansky, 1998). To that end, the Narrative Hierarchy Model is a tree over a story, story type at the root, then episodes, events and clauses, with one change: every event production has two right-hand sides, a clause template (the text surface) and a transition on a small scene (the grid surface). Figure 5, in Appendix F, shows one. We call this a *dual-surface grammar*.

On the dual-surface grammar, comprehension gets an operational reading: to understand the text is to recover enough of the tree to render the other surface, which makes the grid a situation model, external and gradable. Mask position is tree position: a masked ending reads out the story's outcome, a masked middle grid one event inside an episode, the grid analogues of story-cloze and story-infilling (Mostafazadeh et al., 2016; Ippolito et al., 2019). And the synonymy m is a knob, set by the corpus for real stories and by hand in a synthetic lane (Appendix F). Building this on real stories needs recoverable event structure and scale enough to train from nothing. Here we select TinyStories (Eldan & Li, 2023), two million short stories in a small child's vocabulary.

3.2 EVENT PRIMITIVES AND RENDERING CONVENTIONS

Eleven primitives have a grid denotation: appear, vanish, move, ascend, descend, acquire, drop, lose, transfer, emote and transform. Together they cover about a third of the finite clauses in TinyStories. What has no denotation (e.g. dialogue, desire, knowledge) is exactly what the text clue can carry but the grids cannot. The rendering conventions are few and fixed (Figure 4, Appendix C): seven rows, a ground row coloured by the location, a character as a bar whose height is its mood, an object as one cell held or on the ground, an elevated character drawn higher, a broken object grey, and nothing else. They are this sparse so that each can be exhibited in a visible grid, which Section 3.3 requires. Extraction is neo-Davidsonian (Parsons, 1990): each finite verb introduces an event, and the participants attach to that event as roles (agent, theme, recipient, location) rather than to the verb, so that "Lucy handed Mia the toy" and "the toy was handed to Mia by Lucy" yield the same event. Roles come from the dependency parse; pronouns are resolved behind a confidence gate, and hedged clauses (wants, purposes, habits, negations) are skipped, since they do not assert that the event happened. A narrative grammar induced by Re-Pair (Larsson & Moffat, 1999) over the event alphabet supplies the episode chunks and the estimates of v, s, m and L (Appendix F).

3.3 NARC BY CONSTRUCTION

A puzzle is a story's grid sequence with one atomic transition masked. We perform three checks to certify the property. Ambiguity: we enumerate every one-primitive completion of the state before the mask, keep those that differ from the answer (and, for a middle mask, from which the next visible state is reachable in one step), and require at least two. Determinacy: the extracted event chain must yield the answer. Self-teaching: every convention the answer uses must appear in a visible grid of the same puzzle, so the solver is never asked to know a convention it has not been shown. Long chains are compressed to keyframes, only atomic transitions are maskable, and every valid mask position is stored as a mask variant; Figure 10 (Appendix F) shows one generated puzzle with its alternatives. We acknowledge two limitations of this approach. The checks hold for an ideal reader of the chain: they certify that the grids underdetermine the answer and the story determines it, not that a given solver needs the story, which is the empirical relation of Section 4. Second, determinacy is relative to the extracted chain, since an extraction error makes a puzzle whose answer is right for the grids and wrong for the story, which is why we measure fidelity next.

3.4 THE EXTRACTOR, ITS FIDELITY, AND THE GRAMMAR TEXT

We iterated on an extractor and froze it after seven rounds: in each, a strong closed model judged on forty fresh puzzles whether every extracted event is asserted by the story. Event fidelity rose from 78% to 92%, and puzzle fidelity, every event right, from 50% to 68% (Table 11, Appendix F). At scale, three open-weight judges (Zheng et al., 2023) read all 21,864 puzzles; they call 57%, 69% and 65% fully faithful, and we keep the 11,049 on which all three agree as the clean set. The judges select as well as filter, since the puzzles they pass are shorter and simpler, and Section 5 controls for that. Because text and grids are two surfaces of one tree, every puzzle also comes with a second text: its event chain rendered back into plain sentences by the grammar's own templates. We call it the *grammar text*. Faithful to the grids by construction, it is the clean supervision and the ceiling for what the grids can be learned from, while the original story from TinyStory is the generalisation target. Every experiment below reports both.

4 NARC-TINY: GENERATED NARC AT SCALE

The construction certifies the property for an ideal reader of the event chain. From 98,113 TinyStories the generator yields 21,864 puzzles, roughly one story in four or five; the rest fail on too few grid-visible events, on no mask position that is both ambiguous and self-taught, or on low-confidence pronoun resolution. The corpus grammar also returns a fact about the stories: the root of the tree shows in the last primitive, which is what makes a final-grid mask a probe of the root (Appendix F). We also generate puzzles with one of the middle grids masked. This section asks two more things of the generated set before Section 5 trains on it: whether a real reader handed the events can draw the pictures under conventions it was never told, and how the eight open models fare under the protocol of Section 2.

4.1 THE GRAMMAR-TEXT CONTROL

The construction certifies determinacy for an ideal reader of the event chain; whether a real reader can do it is a separate question. The grammar text is how we ask. It states the same events in plain sentences, so a solver handed it supplies only the rendering, drawing the masked grid from the visible ones and the conventions they exhibit. We ran forty stratified puzzles through one strong closed model under four conditions: the grids alone, the story alone, the story with the grids, and the grammar text with the grids (Table 12, Appendix F). The numbers below are from the round with the full breakdown; the frozen extractor scores higher on each condition but the story alone, which is zero throughout.

Four things follow. First, the events are enough: the grammar text with the grids solves 27 of 40 against 10 from the grids alone, and the story alone solves none. The grid side of the construction is sound for a real reader, not only an ideal one. Second, the story is much weaker, at 14 of 40, and the 2×2 over the 30 puzzles the grids do not settle shows how nearly the grammar text contains it: 12 are solved only with the grammar text, 6 with either text, 1 only with the story and 11 with neither. Third, that gap is not merely extraction error. Among the 13 of those 30 whose events the judge called fully faithful, 5 still solve with the grammar text and not with the story, and among the 17 carrying a judged-wrong event the grammar text solves 9 anyway. Finding the drawable events among dialogue, wishes and restatements is hard in its own right. Fourth, when the grammar text does miss, the grid it draws is right in 99% of its cells, so its misses are wrong events rather than misread conventions, which is the self-teaching check doing its job. The grid side is sound, then, and the set already separates one text from another while the extractor is still imperfect.

4.2 EIGHT OPEN MODELS ON 200 GENERATED PUZZLES

We then ran the same eight models on 200 puzzles from the clean set, stratified by masked primitive, under the same protocol (Table 13 and Figure 9, Appendix F). The story alone solves 0 of 200 for every model. The story with the grids solves 1 to 12, next to 0 to 16 from the grids alone; the grammar text with the grids solves 4 to 99, in order of model strength. NARC cells run 1 to 11 with the story and 4 to 89 with the grammar text; 174 of 200 puzzles are grids-sufficient for no model, and 31 are NARC for at least one model with the story against 133 with the grammar text. The

generated set is underdetermined for the whole pool by construction, so what separates models is how much latent structure they recover from which surface. The order checks transfer (Table 16): 85% of grammar-text NARC cells and 70% of story NARC cells are strong and none is weak, and in the ordering task the grammar text lifts mean τ from 0.15 to 0.50 while the story leaves it at 0.17. A final control separates two readings of the story's work, teaching the code or supplying the events. We replaced the story by a plain statement of the rendering conventions, with no events, and also appended it to the story and to the grammar text (Table 17, Appendix F). The conventions alone solve 37 puzzle-model pairs against 53 from the grids alone (27 NARC cells against the story's 47); added to the story they raise its 56 solved pairs to 66. Stating the code is not what the story does for these models; what they lack is the event chain.

5 Learning from scratch: the data each text surface needs

Here the two surfaces do what the hand-authored puzzles could not: with the latent structure known and the surface varied on purpose, we can ask how much data a learner needs to render each text into pictures, and what about the text sets that amount. We train a 4M-parameter GPT from scratch on (text, grid sequence) records, loss on the answer tokens only, and score exact match on held-out puzzles; the text is the grammar text or the story, the budget is 1K, 3K, 10K or all ($\sim$19K) training puzzles, and every model is trained to its plateau (18,000 steps; Figure 2; Tables 14 and 15, Appendix F). Three predictions were fixed before the runs, the second and third in the RHM's terms. (i) Position sets difficulty: a masked middle grid should stay harder than a masked ending once the two positions are matched in training frequency. (ii) The low-synonymy surface is learned from less data: the grammar text should reach a given accuracy at a much smaller budget than the story. (iii) Label cleaning should help the story more, since only its labels carry the extractor's errors.

Prediction (ii) holds cleanly. The grammar text at 1K puzzles (24.9 ± 0.8) beats the story at 3K (19.8 ± 2.2), and at full data the gap is 25 points (66.0 ± 0.7 against 40.7 ± 3.5). Read off the curves, the story needs four to five times the data of the grammar text for the same accuracy: about 5K puzzles to match it at 1K and about 13K at 3K. That ratio is a plateau number, not a fixed-step one, since the earlier short-training sweeps put it at ten. Prediction (i) holds and survives frequency matching: at plateau, middle masks reach 46% against 70% for final masks on the grammar text and 29% against 43% on the story, and balancing mask positions in training moves the middle from 25% to 30% without closing the gap. It is the position of the question in the story, not its rarity. Our 12M runs stopped short of plateau and are reported in Appendix F; we draw no capacity claim from them.

Prediction (iii) fails, and the failure is the informative result. On difficulty-matched sets of 6,824 story puzzles, judge-clean and judge-rejected puzzles train to the same accuracy (21.8% against 19.1%, three seeds, spread ± 4), and injecting grammar-consistent wrong answers into 15 or 30% of the training set does not lower exact match (25.0 and 25.5%). Then the decisive control: a language model rewrote each grammar text as a TinyStories-style paraphrase, same events, varied wording, labels clean by construction and 90% of events confirmed by a second judge. Trained on the paraphrase, the model lands beside the story (45.1 ± 0.6 against 40.7), not beside the grammar text (66.0). What the story needs extra data for is its surface variety, not its labels: in the RHM's terms, data in proportion to m. Training on all three realisations of a puzzle at once, the grammar text and two paraphrases, prices $m{=}3$ at matched compute: the grammar surface loses 4.8 points (61.2 against 66.0), equally on final and middle masks, while the paraphrase surface gains 12 (56.8 against 45.1 alone). Synonymy has a price, and a direct description of the events is cheap supervision for a varied one.

5.1 What the model represents

Does the network build the tree? The question has been put to a synthetic board game (Li et al., 2023) and to entity states in text (Li et al., 2021); here the latent variables come with the data. At every layer of seventeen checkpoints we read the residual stream at the answer token and fit logistic-regression probes for the masked event, its episode chunk and the story-level label, plus a first-word control, against majority, shuffled-label and layer-zero baselines on 578 held-out puzzles (Figure 7, Appendix F). At full data the 4M grammar-text model decodes the event at 53% (chance 7), the episode at 36% (chance 5), the root at 28% (chance 14) and the control at 19% (chance 5); the story

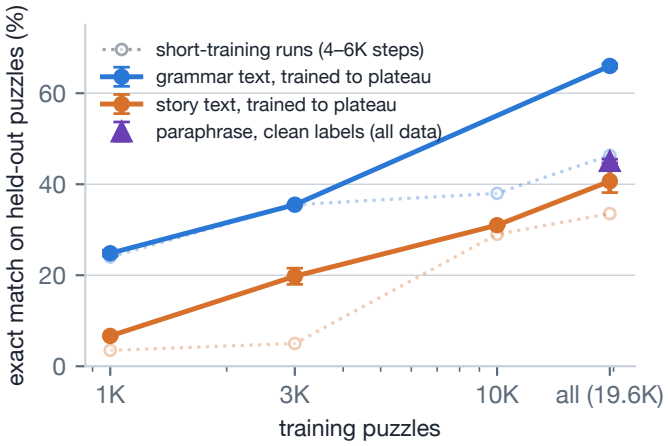


Figure 2: Learning from scratch: exact match on held-out puzzles against the number of training puzzles, 4M models trained to plateau (solid, with the range over seeds) and the earlier short-training runs (faded). The horizontal gap between the two solid curves is the claim: the story needs four to five times the data of the control text for the same accuracy.

model, 32, 29, 34 and 20. Of four predictions fixed in advance, the first fails: we expected the levels to become decodable at successively deeper layers, but every target is decodable from the first block and all peak by the second, so the layer-ordered pipeline of large pretrained encoders (Tenney et al., 2019) does not appear at four to six blocks. The second, depth in data, holds for the two levels with grid denotations: at 1K puzzles the grammar model's event probe is already at 89% of its final value and the episode at 78%, while the root never rises and the story model's root is read off the words. The third holds for event and episode: the story model never reaches the grammar model's 1K event value, and reaches its 1K episode value only at full data, a delay of more than twenty times the data. The fourth, capacity, is partial: 12M lifts event and episode, not the root. Two results bear on Section 5: the masked event is as decodable on middle masks as on final ones although middle masks solve at half the rate, so the middle-mask deficit lies in executing the transition, not recognising it; and the story surface delays the representation more than it delays the score.

6 Related work

ARC and language. ARC (Chollet, 2019) was built to exclude language, and its successors keep that choice (Chollet et al., 2024; 2026; ARC Prize Foundation, 2026); human baselines (LeGris et al., 2024), concept-level variants (Moskvichev et al., 2023), generators (Moffitt, 2025; Lehmann et al., 2026) and reasoning traces (Kim et al., 2025) all stay inside the grids. LARC (Acquaviva et al., 2022) showed that a written description can stand in for visual examples, but pairs it with example transformations rather than a masked sequence; MARC (Anonymous, 2026) attached one figurative sentence to a single transformation, where NARC's sequence lets the clue carry order and causation; cross-modal studies compare humans and models on the same abstractions (Mitchell et al., 2023; Beger et al., 2025). Explanations in context help beyond examples (Lampinen et al., 2022), and test-time training is the field's answer to underdetermination (Akyürek et al., 2025); ours is a story. Grounded grid worlds with language (Ruis et al., 2020; Chevalier-Boisvert et al., 2019) test systematic generalisation, not a masked continuation under a figurative clue.

Hierarchical generative models of data. The Random Hierarchy Model (Cagnetta et al., 2024) and its language variant (Cagnetta & Wyart, 2024) predict level-by-level learning and data requirements that grow with depth and synonymy; scaling laws (Cagnetta et al., 2025), sparse variants (Tomasini & Wyart, 2024), diffusion-model probes (Sclocchi et al., 2025b;a; Favero et al., 2025) and transformer studies on synthetic grammars (Allen-Zhu & Li, 2025; Kallini et al., 2024) develop the picture on synthetic data. The NHM instantiates the RHM with two leaf alphabets on a real corpus.

Narrative. Story grammars (Rumelhart, 1975; Mandler & Johnson, 1977; Thorndyke, 1977) and situation models (Kintsch, 1988; Zwaan & Radvansky, 1998) are the cognitive lineage of the tree and of the reading of comprehension we adopt; narrative as a mode of thought and of reasoning is argued by Bruner (1986), Morgan et al. (2022) and Johnson et al. (2023), and for science by Olson (2015) and Doxiadis & Mazur (2012). In NLP, story cloze (Mostafazadeh et al., 2016), infilling (Ippolito et al., 2019), story ordering (Agrawal et al., 2016) and narrative event chains (Chambers & Jurafsky, 2008) are the text-side analogues of our masks, ordering task and extractor; figurative-language tests (Liu et al., 2022) and text-to-scene systems (Coyne & Sproat, 2001) are the nearest work on the clue and on the back-renderer. TinyStories (Eldan & Li, 2023) supplies the corpus; Re-Pair (Larsson & Moffat, 1999) and Sequitur (Nevill-Manning & Witten, 1997) the grammar induction; neo-Davidsonian event semantics (Parsons, 1990) the extractor's representation.

Probing and evaluation hygiene. World-state probes on from-scratch sequence models (Li et al., 2023; 2021) and the layer-ordered pipeline of pretrained encoders (Tenney et al., 2019; Hewitt & Manning, 2019) frame Section 5.1; contamination (Sainz et al., 2023) and LLM-as-judge (Zheng et al., 2023) frame the drafts-as-bound and the fidelity judges.

7 LET'S TAKE STOCK

What sorts of things have we learned? First, it is possible to both identify and generate the NARC property, a model-puzzle relation, which we can increasingly attribute to the puzzle through a diagnostic battery and cross-model profile. The battery establishes that both the narrative and the grids are necessary, and how far that necessity survives stripping the clue to its keywords and shuffling the grids. The cross-model profile then helps us identify which failures belong to the puzzle and which to the model. Grids that suffice for every model mark a defect in the puzzle; grids that suffice for none, while at least one model has the property, mark a puzzle that works.

Second, the NHM-constructed puzzles establish what the battery cannot: that the grids underdetermine the answer and the story determines it, before any model is consulted. Neither establishes sufficiency for reasoning in general, nor that a solver needs a story rather than the event chain it carries: a model may be too weak, or may have met the conventions elsewhere. But for the open models we test, the grammar text and the conventions control agree that the chain is what is missing.

Several limitations bound these claims. Our extractor is imperfect, so we measured it, and the judges select simpler puzzles as they filter, so the learning experiments matched difficulty. Both texts are synthetic: the grammar text is stilted by design, and TinyStories are themselves written by a language model. Contamination is bounded by drafts that were never public and score the same, not by a held-out set. And the models are stochastic: single-cell verdicts in the battery flip about one run in six, so our claims are rates over cells rather than lists of puzzles.

Despite these limitations, we think this is a productive line of research that we and others can build on. Narratives do not always tell events in the order they happened, but our NHM grammar renders them in order: how much does discourse order matter against story order? That is a knob worth turning. Another set of questions concerns the conventions we used for the grids, since every rendering code is, in a sense, a different metaphor that has to be learned. How much data a new one takes can be measured the way synonymy was here, and how much it takes before the capacity generalises to in-context learning of new metaphors is open too.

We have made a small contribution to a large question. A narrative resolves the underdetermination of a visual sequence by supplying its latent event structure; what a learner needs data for is recovering that structure from a varied surface; and with one grammar writing both the story and the pictures, the resolution and the data requirement can both be measured. Whether human minds resolve such underdetermination the same way remains open.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks that require disclosure. *Synthetic data generation.* Of the 633 puzzles in the hand-authored benchmark, 229 were written by people with no model involvement. The other 404 had a language model in the loop, in one of two ways: 252 were composed directly by a language model under a human-written recipe, and 152

were produced by an autonomous agent run as an evolutionary loop in which a person judged at intervals which candidates to keep and develop, 59 of those being the April 2026 agent series that converged on a fixed design recipe. Every puzzle that was made public passed the platform's review gate, owner or reviewer approval and the automated eight-model battery. Narrative variants were generated by a language model to a domain or facet specification and read by an author before use. In the generated benchmark, the source stories are TinyStories, themselves written by GPT-3.5 and GPT-4; the paraphrase text was written by an open-weight model (qwen3.8-27b) and judged by a second (gemma-4-31b); extractor fidelity was judged by a closed model during development and by three open-weight models at scale. The subject models' outputs are the data of this paper, not assistance, and the second-pass extractor (gpt-oss-120b) is part of the measured pipeline, reported as such in Appendix B. *Conceptual framework and experimental design.* The research questions, the NARC property, the Narrative Hierarchy Model and the four predictions were specified by the authors. An LLM coding assistant (Claude Code) drafted several designs in working sessions from the authors' specifications, including the order- and form-sensitivity tests, the mask-variant schema, the puzzle-by-model decomposition, the stability repeats and the explicit-conventions control; the authors approved every design before it ran. *Experiments and implementation.* The collection, classification, generation, training and analysis code was written by the assistant under author direction; all code was run by the authors, and every headline number was recomputed from the stored trial records by scripts the authors inspected. We have not used generative AI tools to formulate mathematical claims, and no literature claim was taken from a model without checking the source. Additionally, we used generative AI tools for tasks with recommended disclosure: figure scripts were written by the assistant and the figures checked by the authors; the first drafts of most sections of this paper, of its captions and of these statements were produced by the assistant from the authors' outlines, results tables and instruction notes, and the authors revised every paragraph. We have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Human solver data. The public site records anonymous solve attempts: a random session identifier, no login and no e-mail, the grids submitted, correctness, time on task and whether the clue had been revealed. No demographic information was collected and no address is stored. Institutional review is in progress, so we report these data only as exploratory descriptive statistics, draw no inference about any person, and release them in aggregate alone: counts by puzzle and by condition, not per-session records. *Dataset content and release.* The hand-authored puzzles are original grids with original short clues; clues that allude to literary or philosophical sources paraphrase them and quote nothing. The generated puzzles are derived from TinyStories, a synthetic corpus released under the CDLA-Sharing-1.0 licence, and contain no personal data. We release the puzzles, variants, trials and verdicts under CC BY 4.0 and the code under the MIT licence. No set is withheld; the never-public drafts serve as the contamination bound (Appendix A). *Generated content.* Part of the corpus was authored by language models or an autonomous agent, as the AI use statement details; every active puzzle passed human review and the automated battery before activation. The moral-dilemma puzzles describe generic scenarios, and a matched control showed that their moral content is not what the models respond to. *Compute and interests.* All open-model inference ran on a university-hosted cluster of open-weight models, about 42,000 masking trials plus their extraction passes for the hand-authored benchmark and about 170,000 model calls for the generated benchmark, its judges and paraphrases; the stronger closed model was run through its vendor's command-line tool. From-scratch training comprised about sixty runs of 16 to 90 minutes each on a single GPU. This work received no external funding; the inference and the training ran on computing resources provided by our institution. We declare no competing interests.

REPRODUCIBILITY STATEMENT

The supplementary archive contains the anonymized source of the collection pipeline, the classifier, the sensitivity tests and the ordering task; the generator of the Narrative Hierarchy Model with its lexicon, extractor, renderer and back-renderer; the training and probing code; and the scripts that produce every table and figure from the stored records. It also contains a sample of the hand-

authored puzzles with their variants, trials and verdicts, a sample of the generated puzzles with their event chains, mask analyses, grammar texts and paraphrases, the three judges' records, the training configurations with per-puzzle predictions, and the probe results; the full puzzle sets and trial records will be released as a public dataset on acceptance. Verbatim prompts, the grid text encoding, the two-pass extraction and its grading rule, the shuffle seeding, the keyword rule with its stopword list and the ordering protocol are in Appendix B; the eight open models are named with their sizes in Table 5 and were run at temperature zero through an OpenAI-compatible endpoint, so the battery runs against any such endpoint. Served open-weight models can differ across backends and versions, which we did not control; the repeats of Appendix C bound the run-to-run variance for the pool used here. Training details, model sizes and every sweep configuration are in Appendix F.

REFERENCES

Samuel Acquaviva, Yewen Pu, Marta Kryven, Theodoros Sechopoulos, Catherine Wong, Gabrielle E. Ecanow, Maxwell Nye, Michael Henry Tessler, and Joshua B. Tenenbaum. Communicating natural programs to humans and machines. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2022. arXiv:2106.07824; NeurIPS 2022 Datasets and Benchmarks. Verified Sep 17, 2026.

Harsh Agrawal, Arjun Chandrasekaran, Dhruv Batra, Devi Parikh, and Mohit Bansal. Sort story: Sorting jumbled images and captions into stories. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 925–931, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1091.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 942–963. PMLR, 2025. arXiv:2411.07279.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 1, learning hierarchical language structures. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=mPQKyzkA1K. arXiv:2305.13673.

Anonymous. MARC: Metaphor abstraction and reasoning corpus. Public dataset; citation withheld for double-blind review, 2026.

ARC Prize Foundation. Arc-agi-3: A new challenge for frontier agentic intelligence. Technical report, ARC Prize Foundation, 2026. arXiv:2603.24621; also released as the ARC Prize Foundation technical report, April 22, 2026.

Claas Beger, Ryan Yi, Shuhao Fu, Kaleda Denton, Arseny Moskvichev, Sarah W. Tsai, Sivasankaran Rajamanickam, and Melanie Mitchell. Do ai models perform human-like abstract reasoning across modalities? *arXiv preprint arXiv:2510.02125*, 2025. arXiv:2603.24621; also released as the ARC Prize Foundation technical report, April 22, 2026.

Jerome Bruner. *Actual Minds, Possible Worlds*. Harvard University Press, Cambridge, MA, 1986.

Francesco Cagnetta and Matthieu Wyart. Towards a theory of how the structure of language is acquired by deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2406.00048.

Francesco Cagnetta, Leonardo Petrini, Umberto M. Tomasini, Alessandro Favero, and Matthieu Wyart. How deep neural networks learn compositional data: The random hierarchy model. *Physical Review X*, 14:031001, 2024. arXiv:2307.02129.

Francesco Cagnetta, Alessandro Favero, Antonio Sclocchi, and Matthieu Wyart. Scaling laws and representation learning in simple hierarchical languages. *Physical Review E*, 112:065312, 2025. arXiv:2505.07070.

Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pp. 789–797, Columbus, Ohio, 2008. Association for Computational Linguistics. URL https://aclanthology.org/P08-1090/.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJeXCo0cYX.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

François Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. Arc prize 2024: Technical report. *arXiv preprint arXiv:2412.04604*, 2024. arXiv:2412.04604. Verified Sep 17, 2026.

François Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. Arc prize 2025: Technical report. *arXiv preprint arXiv:2601.10904*, 2026. arXiv:2601.10904. Verified Sep 17, 2026 (arXiv abs page).

Bob Coyne and Richard Sproat. WordsEye: An automatic text-to-scene conversion system. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 487–496, New York, NY, USA, 2001. ACM. doi: 10.1145/383259.383316.

Apostolos Doxiadis and Barry Mazur (eds.). *Circles Disturbed: The Interplay of Mathematics and Narrative*. Princeton University Press, 2012.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

Alessandro Favero, Antonio Sclocchi, Francesco Cagnetta, Pascal Frossard, and Matthieu Wyart. How compositional generalization and creativity improve as diffusion models are trained. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.12089.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419.

Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck. Unsupervised hierarchical story infilling. In *Proceedings of the First Workshop on Narrative Understanding*, pp. 37–43, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-2405.

Samuel G. B. Johnson, Avri Bilovich, and David Tuckett. Conviction narrative theory: A theory of choice under radical uncertainty. *Behavioral and Brain Sciences*, 46:e82, 2023. doi: 10.1017/S0140525X22001157. Verified Sep 17, 2026 (Cambridge Core).

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14691–14714, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.787.

Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

Sejin Kim, Hayan Choi, Seokki Lee, and Sundong Kim. Arctraj: A dataset and benchmark of human reasoning trajectories for abstract problem solving. *arXiv preprint arXiv:2511.11079*, 2025. arXiv:2603.24621; also released as the ARC Prize Foundation technical report, April 22, 2026.

Walter Kintsch. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95(2):163–182, 1988. doi: 10.1037/0033-295X.95.2.163.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 537–563, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.38.

N. Jesper Larsson and Alistair Moffat. Offline dictionary-based compression. In *Proceedings of the Data Compression Conference (DCC)*, pp. 296–305, 1999. doi: 10.1109/DCC.1999.755679. Re-Pair. DCC 1999, pp. 296–305; journal version Proc. IEEE 88(11):1722–1732, 2000, doi 10.1109/5.892708. Verified Sep 17, 2026 (Crossref).

Solim LeGris, Wai Keen Vong, Brenden M. Lake, and Todd M. Gureckis. H-arc: A robust estimate of human performance on the abstraction and reasoning corpus benchmark. *arXiv preprint arXiv:2409.01374*, 2024. arXiv:2409.01374. Verified Sep 17, 2026.

Jens Lehmann, Syeda Khushbakht, Nikoo Salehfard, Nur A Zarin Nishat, Dhananjay Bhandiwad, Andrei Aioanei, and Sahar Vahdati. Arc-tgi: Human-validated task generators with reasoning chain templates for arc-agi. *arXiv preprint arXiv:2603.24621*, 2026. arXiv:2603.24621; also released as the ARC Prize Foundation technical report, April 22, 2026.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1813–1827, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *International Conference on Learning Representations*, 2023. arXiv:2210.13382.

Emmy Liu, Chenxuan Cui, Kenneth Zheng, and Graham Neubig. Testing the ability of language models to interpret figurative language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4437–4452, Seattle, United States, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.330.

Machine Learning Street Talk. ARC-AGI-3 winning team: Millennia of minds, compressed into words. Podcast interview with the Tufa Labs team (Benjamin Crouzier, Jeroen Cottaar, Dries Smit, Stefano Viel and Michal Tesnar), hosted by Tim Scarfe; video, 1 h 24 min, 2026. Published July 1, 2026. Chapters "Stripping language away, then priors leak back" (1:07:53) and "Representation and whether language is necessary" (1:14:06). Verified September 25, 2026 from the episode page; no transcript available.

Jean M. Mandler and Nancy S. Johnson. Remembrance of things parsed: Story structure and recall. *Cognitive Psychology*, 9(1):111–151, 1977. doi: 10.1016/0010-0285(77)90006-8.

Melanie Mitchell, Alessandro B. Palmarini, and Arseny Moskvichev. Comparing humans, gpt-4, and gpt-4v on abstraction and reasoning tasks. *arXiv preprint arXiv:2311.09247*, 2023.

Michael D. Moffitt. Arc-gen: A mimetic procedural benchmark generator for the abstraction and reasoning corpus. *arXiv preprint arXiv:2511.00162*, 2025. arXiv:2603.24621; also released as the ARC Prize Foundation technical report, April 22, 2026.

Mary S. Morgan, Kim M. Hajek, and Dominic J. Berry (eds.). *Narrative Science: Reasoning, Representing and Knowing since 1800*. Cambridge University Press, 2022.

Arseny Moskvichev, Victor Vikram Odouard, and Melanie Mitchell. The conceptarc benchmark: Evaluating understanding and generalization in the arc domain. *Transactions on Machine Learning Research*, 2023. arXiv:2305.07141; TMLR 2023. Verified Sep 17, 2026.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849, San Diego, California, 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1098.

Craig G. Nevill-Manning and Ian H. Witten. Identifying hierarchical structure in sequences: A linear-time algorithm. *Journal of Artificial Intelligence Research*, 7:67–82, 1997. doi: 10.1613/jair.374. Sequitur. JAIR 7:67–82, 1997. Verified Sep 17, 2026.

Randy Olson. *Houston, We Have a Narrative: Why Science Needs Story*. University of Chicago Press, 2015.

Terence Parsons. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press, Cambridge, MA, 1990. Verified Sep 22, 2026 (publisher listing via PhilPapers and Semantic Scholar).

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M. Lake. A benchmark for systematic generalization in grounded language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19861–19872. Curran Associates, Inc., 2020.

David E. Rumelhart. Notes on a schema for stories. In Daniel G. Bobrow and Allan Collins (eds.), *Representation and Understanding: Studies in Cognitive Science*, pp. 211–236. Academic Press, New York, 1975. doi: 10.1016/B978-0-12-108550-6.50013-6.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, Singapore, 2023. doi: 10.18653/v1/2023.findings-emnlp.722. Verified Sep 17, 2026 (ACL Anthology).

Antonio Sclocchi, Alessandro Favero, Noam Itzhak Levi, and Matthieu Wyart. Probing the latent hierarchical structure of data via diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025a. arXiv:2410.13770.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. *Proceedings of the National Academy of Sciences*, 122(1): e2408799121, 2025b. arXiv:2402.16991.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452.

Perry W. Thorndyke. Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, 9(1):77–110, 1977. doi: 10.1016/0010-0285(77)90005-6.

Umberto M. Tomasini and Matthieu Wyart. How deep networks learn sparse and hierarchical data: The sparse random hierarchy model. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2404.10727.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. Datasets and Benchmarks Track.

Rolf A. Zwaan and Gabriel A. Radvansky. Situation models in language comprehension and memory. *Psychological Bulletin*, 123(2):162–185, 1998. doi: 10.1037/0033-2909.123.2.162.

A HAND-AUTHORED CORPUS: CONSTRUCTION, TAXONOMY, PROVENANCE

Construction and provenance. The corpus holds 633 puzzles, 319 of them public on the project site and the rest drafts. The database records how each was made, in three classes. *Human-only* (229; 71 active): written by a person with no model involved, mostly public submissions to the site, a workshop set, and part of the original series. The other 404 had a language model somewhere in the making, in one of two ways. *Model-written* (252; 132 active): composed directly by a language model under a human-written recipe, covering most of the original series, the spectrum series that runs from near-literal to figurative clues, and the smaller gap-filling, AI-native and narrative-facet series. *Agent-evolved* (152; 116 active): produced by an autonomous agent run as an evolutionary loop, with a person stepping in at intervals to judge which candidates were interesting enough to

keep and develop; this class holds the iterated design series (57) and the hunt, stance and agent series (59 in all), which converged on a fixed recipe developed against six of the eight models, namely a trend that reverses in the masked grid, a compact grid, and a clue in relative rather than absolute terms. By series the largest are the public submissions (162), the original set and its spectrum series (124 and 100), the iterated design series (57) and the agent series (59); the rest are gap-filling and visual-first puzzles, puzzles rewritten along five narrative facets, ethics scenarios and a workshop set. Every puzzle that was made public, whatever its class, passed the platform's review gate: owner or reviewer approval and the automated eight-model battery. Table 2 gives the scale of the evaluation.

Taxonomy. Every active puzzle carries tags on six dimensions: audience (general, humanities, science, AI-native), narrative arc (linear, rise-and-fall, cyclical, subversion, transformation, convergence, divergence), clue type (direction, identity, exception, rule, timing, causation), domain (sixteen values from literature and mythology to physics, ecology and information theory), spectrum (AI-forte, human-forte, AI-edge, human-edge, balanced, domain-dependent) and grid count (3 to 8). The tags are descriptive and were not used to select puzzles for any experiment in this paper.

Held-out material. No puzzle set was withheld from the site by design. The 157 human-authored drafts were never posted, and their NARC rate under the stronger model equals that of the public set (Section 2.2); they are the paper's bound on contamination. The human-authored puzzles are larger than the rest of the pool (41 cells per grid on average against 13 for the robust core, clues of 348 characters against 243; Table 7).

Table 2: Corpus and evaluation scale at the September 2026 freeze. A classification cell is one puzzle with one narrative variant and one mask variant run under all three conditions on one model.

Puzzles (active / draft)	633 (319 / 314)
Narrative variants	972
Mask variants	676
Enabled narrative × mask cells	1073
Subject models (open-weight)	8
Masking trials	41,415
Classification cells	9,237
Ordering trials	560
Human sessions / puzzles / attempts	821 / 249 / 1251

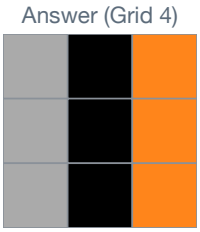


Figure 3: The answer to the puzzle of Figure 1: Bob's bar rises to three cells.

B DIAGNOSTIC BATTERY: PROTOCOL DETAILS

Prompts. Every trial has a system prompt and a user message. The system prompt for a single masked position p reads:

```
You are solving a NARC puzzle (Narrative Augmented Reasoning
Challenge). A NARC puzzle consists of a sequence of
grids that tell an abstract story. One grid in the
sequence is masked. Your task is to reconstruct the
masked grid pixel-perfectly. Provide your answer as a
JSON object: {"reasoning": "your step-by-step reasoning",
"output_grids": {"p": [[int, ...], ...]}} The grid must
be a 2D array of integers (0-9) representing colors. The
key is the grid position (0-indexed). IMPORTANT: You MUST
end your response with your final answer as a JSON object.
Even if uncertain, commit to your best guess. Do not end
mid-reasoning.
```

Grids are encoded as text, one row per line, one letter per cell under the colour key `.=black B=blue R=red G=green Y=yellow X=grey M=magenta O=orange A=azure W=maroon`, each grid headed by its 1-indexed position and size, the masked grid shown as `[MASKED]` with its size. The user message for *grids only* is the colour key, the encoded sequence and the line "Grid $p{+}1$ is masked ($r{\times}c$). Reconstruct it as a 2D array of integers." For *narrative only* the sequence is replaced by the clue in quotation marks, the number of grids and the masked position. For *both* the clue precedes the sequence and the message ends "Use the narrative and visible grids together to reconstruct the masked grid(s)." The keyword condition replaces the clue line by "Key terms (unordered): ..." and the closing instruction by "Use the key terms and visible grids together ..."; it is never presented as a story. The explicit-conventions conditions of Section 4.2 insert the statement of the rendering code as "Rendering conventions: ..." in the same slot.

Parsing and grading. The subject model's response is parsed for the final JSON object. When none parses, a fixed extractor model (gpt-oss-120b, temperature 0) receives the last 4,000 characters of the response (for four subject models whose API returns a separate reasoning stream, the tail of that stream) under a system prompt that says it is a JSON formatter, must not reason, and must return only the final grid under the stated key; since September 16, 2026 the prompt names the masked

position, its size, and the fact that the text may call it "Grid $p{+}1$". A strict third pass with explicit dimensions is tried when the second fails. Grading is exact match on every cell against the stored answer. A lone predicted grid under any key is graded at the masked position; with several masked positions, the same number of grids whose keys all shift by one onto the masked set are shifted back; every other case is graded strictly. Every trial recorded before September 16 was rescored from its stored prediction under this rule (2,236 verdicts changed across the eight open models, none for the closed model); parse failures after all passes are about 0.5% of trials and count as unsolved.

Order sensitivity. For a NARC cell, the visible grids are re-presented in a shuffled order with the clue unchanged. The permutation seed is the first 32 bits of the SHA-256 digest of `puzzle|variant|mask|attempt`; the permutation is drawn with Python's `random.Random(seed).shuffle`, rejected if it is the identity or repeats an earlier attempt for the cell, and $K{=}3$ distinct permutations are used. The masked grid keeps its content and its label in the shuffled display. A cell is strong when 0 of K shuffles solve, weak when K of K do, partial otherwise.

Form sensitivity. The clue is lower-cased and tokenised (letters and digits, internal apostrophes and hyphens kept), 120 function words are removed (articles, pronouns, auxiliaries, prepositions, conjunctions and the like: *a, about, after, again, all, am, an, and, any, are, as, at, be, because, . . . , would, you, your, yours, yourself, yourselves*), duplicates are dropped and the remainder is alphabetised, so that no trace of the clue's order survives. Three repeats per cell at temperature zero; narrative when 0 of 3 solve, lexical when 3 of 3 do.

Variants and the ordering task. A narrative variant pairs the same grids and mask with a different clue (972 in the corpus); a mask variant pairs the same grids and clue with a different masked position (676); each combination is a cell and the property is evaluated per cell. In the ordering task all grids, the answer included, are shown in a seeded shuffled order under letter labels, with or without the clue, and the model returns the chronological order as a JSON array of letters; agreement with the true order is Kendall's τ (Kendall, 1938), exact order is $\tau{=}1$, and a second-pass extraction reads unparsed answers as above.

Grading and the extraction pass. A trial is graded by exact match of the predicted grid against the answer at the masked position. Responses are parsed in two passes: the subject model's own final JSON first and, when that is absent or malformed, a fixed extractor model (gpt-oss-120b) that reads the response and returns the grid in the required JSON; for the four subject models whose API returns a separate reasoning stream, the extractor reads the tail of that stream. Until September 16, 2026 the extraction prompt did not state the masked position, and the grader required that position as the JSON key. A subject model that keyed its one grid `"0"` or by the 1-indexed `Grid N` label shown in its prompt, or an extractor that did the same, therefore had an exactly right grid graded wrong; this affected 2,236 trials across the eight open models and none of the closed-model trials. Since September 16, 2026 the extraction prompt states the masked position(s) and grid sizes, and the grading rule is: with one masked position, a lone predicted grid under any key is graded at that position; with several masked positions, the same number of grids whose keys all shift by one onto the masked set are shifted back; every other case is graded strictly. Every affected trial was rescored from its stored prediction without re-running any model (audit record `rescore_miskeyed_20260916_230819.jsonl`), verdicts were reclassified, and the sensitivity tests were run for every newly NARC cell, so all numbers in this paper follow the corrected rule.

Table 3: The diagnostic battery. Each test re-runs the winning *both* condition with one thing changed and records what breaks; K is the number of repeats per cell. Read a strong cell as "the order was necessary" and a narrative cell as "the words alone were not enough". Protocol details are in Appendix B; on the generated set the content control is the grammar text (Section 3.4).

Test	What changes	Verdict	K
Three conditions	grids only / narrative only / both	NARC iff only *both* solves	1
Order sensitivity	seeded non-identity shuffles of the grid order, same clue	strong (0/K solved) / partial / weak (K/K)	3
Form sensitivity	clue replaced by its alphabetized content words	narrative (0/K) / partial / lexical (K/K)	3
Narrative variants	same grids and mask, different clue (972)	NARC per cell	1
Mask variants	same grids and clue, different masked position (676)	NARC per cell	1
Grid re-encoding	same clue, different faithful grid encoding	preservation of the NARC rate	1
Ordering task	all grids shown, shuffled; recover the order	Kendall τ with vs without clue	1

C Robust core, per-model table, and rendering conventions

The robust core. 34 puzzles are NARC for all eight open models in at least one narrative-by-mask cell. Tier A, NARC on all eight in the original cell (10): `hunt_031`, `narc_gap_103`, `iter_003`, `iter_042`, `narc_gap_104`, `narc_new_018`, `stance_design_5x5`, `narc_ai_013`, `narc_sp_075`, `narc_sp_077`. Tier B, NARC on all eight within one variant cell (12): `agent_003`, `hunt_036`, `hunt_018`, `iter_026`, `stance_design`, `sub_019`, `hunt_032d`, `narc_005`, `narc_ai_008`, `hunt_028`, `hunt_037`, `narc_003`. Tier C, NARC on all eight only across cells (12): `agent_004`, `hunt_024`, `hunt_017`, `stance_005`, `narc_011`, `hunt_040`, `agent_006`, `agent_008`, `stance_003`, `stance_002`, `stance_007`, `stance_012`. 20 of the 34 come from the autonomous-agent series (hunt, agent, stance); one is a public submission. Table 5 gives per-model outcomes over all cells. Table 7 sets the core beside the human-authored puzzles, and the comparison separates two kinds of robustness. The core is robust across this pool partly because it was tuned to it: its agent-evolved puzzles were developed against six of the eight models and are NARC on 58% of cells for those six against 52% for the two added later. The human-authored puzzles are robust in the other sense, against capability: three times the grid area, twice the clue length, and 39% of them unsolved by every open model even with the clue, which is why Section 2.2 puts a stronger reader on them. A NARC verdict is also a conjunction of three trials, so it inherits the noise of each. Table 8 reports three fresh repeats of every condition on the core at temperature zero: per model, the share of cells whose three verdicts agree runs from 77% to 90%, and the share of stored NARC verdicts holding in every repeat from 29% to 80%, lowest for the two smallest models. Overall the per-condition verdicts agree across all three repeats in 84% of cells, and the NARC rate is stable, 59 to 61% of pairs per repeat against 63% stored, while only 54% of stored NARC pairs hold in every repeat and 78% in a majority, which is what independent flips at that rate predict. The core is robust as a rate and fragile as a list, so the claims in this paper are rates over cells rather than lists of puzzles.

Table 4: Cross-model profiles of the 617 puzzles with an original-cell verdict from all eight open models; each row counts puzzles by how many models produced the outcome. Read the first four rows as attribution: grids sufficing on all eight is a defect of the puzzle, on one to seven a fact about the model, on none the underdetermination the format is built for; unsolved on all eight is either an under-specified clue or the pool's ceiling.

Profile	Puzzles	Share	Reading
Grids sufficient on 0 of 8	405	66%	underdetermination holds for the whole pool
Grids sufficient on all 8	15	2%	puzzle-level defect: no NARC feature
Grids sufficient on 1–7	197	32%	model-relative: the model is the variable
Unsolved with both on all 8	203	33%	clue under-specifies, or pool ceiling
Solved with both on all 8	78	13%	
Narrative alone sufficient on $\geq$1	120	19%	clue over-specifies for that model
NARC on all 8	10	2%	robust core (original cell)
NARC on 4–7	140	23%	
NARC on 1–3	218	35%	
NARC on 0	249	40%	

Table 5: Per-model outcomes over all classification cells, one cell being a puzzle with one narrative variant and one mask variant. NARC: grids alone fail, narrative alone fails, both succeed. Strong: fails all three grid shuffles. Form: fails all three keyword-ablated clues. Cell counts differ because variants were added while the pool grew; each model's rates are over the cells it was run on.

Model	Cells	NARC	NARC%	Grids suff.	Narr. suff.	Both	Strong	Form
gpt-oss-120b	1099	301	27.4	169	66	474	100	213
Nemotron-3-Super	1103	256	23.2	105	61	381	112	172
Qwen3.8-27B	1037	296	28.5	138	58	433	129	203
Qwen3.5-122B	1104	276	25.0	156	70	433	129	163
Qwen3.6-27B	1586	352	22.2	162	94	522	147	209
Gemma-4-31B	1105	268	24.3	177	88	469	81	193
Gemma-4-26B	1105	224	20.3	120	94	371	101	166
gpt-oss-20b	1098	190	17.3	74	53	277	87	166

Table 6: Ablation typology over all 2,163 NARC cells of the eight open models, counts with percentages in parentheses. Rows: order sensitivity under grid shuffling. Columns: form sensitivity under keyword ablation. Top-left, strong $\times$ narrative, is the deepest tier; bottom-right, weak $\times$ lexical, is keyword priming.

	Narrative (form nec.)	Partial	Lexical (words suffice)	Row total
Strong (order nec.)	613 (28.3%)	223 (10.3%)	50 (2.3%)	886
Partial	619 (28.6%)	177 (8.2%)	73 (3.4%)	869
Weak (order incidental)	253 (11.7%)	88 (4.1%)	67 (3.1%)	408
Column total	1485 (68.7%)	488 (22.6%)	190 (8.8%)	2163

Table 7: Two kinds of robustness, original-cell statistics over the eight open models. The robust core (NARC on all eight in some cell) is small, tersely clued and partly tuned to the pool: its agent-authored puzzles are NARC more often on the six models they were developed against than on the two added later. The active human-authored puzzles are larger, richly clued and mostly beyond the pool.

	Robust core	Agent-authored core	Human-authored (active)
Puzzles	34	20	71
Original-cell NARC count, mean / max	5.6 / 8	4.5 / 8	1.6 / 7
Unsolved by all 8 with clue	3%	5%	39%
Cells per grid, mean	13	10	41
Clue length (chars), mean	243	166	348
NARC rate on the 6 development models	–	58%	17%
NARC rate on the 2 later-added models	–	52%	27%

Table 8: Stability on the 34-puzzle core: three fresh repeats of every condition at temperature zero, eight open models. *All repeats agree*: share of (puzzle, model, condition) cells whose three verdicts coincide. *Stored NARC kept*: of the original-cell NARC verdicts in the database, how many hold in every repeat. Rates are stable across repeats while single verdicts are not (Section 2).

Model	cells	all repeats agree	stored NARC kept
gemma-4-26b	102	81%	11/25
gemma-4-31b	102	90%	20/25
gpt-oss-120b	102	86%	14/23
gpt-oss-20b	102	82%	4/14
nemotron-3-super	102	77%	6/20
qwen3.5-122b	102	81%	11/22
qwen3.6-27b	102	81%	7/14
qwen3.8-27b	102	89%	20/28

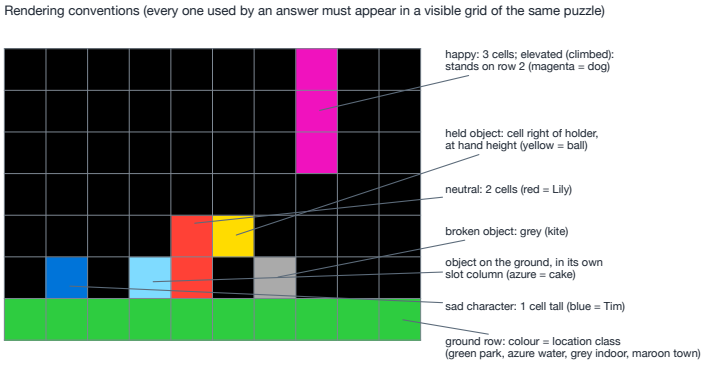


Figure 4: The rendering conventions on one annotated grid: seven rows, a ground row coloured by location, characters as bars whose height is mood, objects as single cells held or on the ground, elevation, and breakage drawn grey. Every convention an answer relies on must appear in a visible grid of the same puzzle.

D NARRATIVE-DESIGN EXPERIMENTS AND THE STRONGER-MODEL RUN

The stronger-model run. Claude Fable 5.1 was run through a command-line harness in closed-book mode (no tools, a fresh session per trial, high reasoning effort, the harness's default sampling with no temperature control), on all 226 human-authored puzzles under the three base conditions. The system and user prompts are the open models' except that, after a safety filter began rejecting the phrase "step-by-step reasoning" in the JSON key, the key was renamed to "explanation" (prompt v2); on the 106 cells that were NARC under v1, v2 agreed on 97 of 106 grids-only verdicts, 106 of 106 narrative-only and 92 of 106 both. The order and form tests used the same seeded permutations as the open-model runs and $K{=}3$ keyword ablations. Table 9 gives the per-set counts: NARC on 42 of the 69 public puzzles (61%) and on 65 of the 157 never-public drafts (41%), with the narrative alone solving 6 of 226. Of the 79 human puzzles unsolved by all eight open models, this model makes 32 NARC and 6 grids-sufficient and leaves 41 unsolved; of the 21 such puzzles that are public, 12 become NARC. Table 10 gives the typology beside the open models on the same puzzles. Verdict stability over a second full run is 85 of 107 (79%).

Design pilots. Three small studies from spring 2026 informed how clues were written; they predate the grading correction of Appendix B and we report them as design findings. *Ordering.* On 70 puzzles and four models, the clue lifted Kendall's τ from 0.30 to 0.67 (gpt-oss-120b), 0.33 to 0.70 (qwen3.5-122b), 0.16 to 0.51 (nemotron-3-super) and 0.26 to 0.51 (gpt-oss-20b), exact orders from 26–39% to 41–66%; the lift grew with the number of grids (four grids $+0.32$, five $+0.35$, six $+0.42$). *Narrative facets.* Twenty puzzles were rewritten to foreground one of five facets of a narrative (teller and told, world, events, actors, feeling); variants foregrounding the teller or the feeling kept the property most often, and clinical restatements of the events matched or beat richer tellings for these models. *Grid re-encoding.* The same clues were paired with alternative faithful encodings of the grids in a small pilot; the property was preserved for a majority of cells but the sample is too small to quote a rate. Full tables for the three pilots are in the released project notes.

Table 9: The stronger closed model on the human-authored puzzles: closed book, single sample per condition, prompt identical to the open models' except for one JSON key. *Grids suff.*: solved from the grids alone. *Narr. suff.*: solved from the narrative alone. The drafts were never posted publicly, so their rate bounds contamination.

Set	Puzzles	Grids only	Narr. only	Both	NARC	Grids suff.	Unsolved	Narr. suff.
active (public)	69	14	1	56	42 (61%)	14	12	1
draft (never public)	157	41	5	108	65 (41%)	41	47	4
all	226	55	6	164	107 (47%)	55	59	5

Table 10: Ablation typology on the human-authored puzzles: the stronger closed model's NARC cells against the eight open models' original-cell NARC cells on the same puzzles, as shares of NARC cells. Read down the columns: the stronger model needs the clue's form less often, is satisfied by its keywords more often, and sits in the deepest tier half as often.

Share of NARC cells	Fable 5.1	8 open models
form-necessary (fails on keyword list)	47%	72%
lexical (keyword list suffices)	27%	8%
order-necessary (fails every shuffle)	40%	63%
order-incidental (solves every shuffle)	14%	5%
strong × narrative (deepest tier)	24%	50%
weak × lexical (keyword priming)	7%	0%

E HUMAN SOLVER DATA

The public site lets anonymous visitors attempt a puzzle in two phases: first from the grids alone, then, after submitting or skipping, with the clue shown. Between March 31 and September 16, 2026 there were 821 sessions, 1,251 attempts and 249 distinct puzzles; 780 attempts skipped the grids-only phase. Of graded attempts, 48 without the clue were correct 14 times (29%) and 423 with the clue 231 times (55%); the median time per attempt was 12 seconds. No identity, demographic or incentive information was collected, sessions are self-selected and most attempted one puzzle, and the grids-only phase was rarely taken, so the comparison is exploratory. We release these data in aggregate alone, as counts by puzzle and by condition; institutional review of the per-session records is in progress.

F NARRATIVE HIERARCHY MODEL: GRAMMARS, ESTIMATES, AND THE SYNTHETIC LANE

Induced grammars and estimates. Two grammars are induced from the corpus. A head-driven sentence PCFG over the parsed clauses has 10,268 distinct sentence rules with 3.7 daughters on average over a lexicon of 5,529 lemmas. A narrative grammar over the event alphabet (20 primitive symbols, an event with its salient argument such as `EMOTE.sad` or `MOVE.loc`) is induced by Re-Pair over 26,626 event chains: the conditional entropy of the next symbol given the previous is 3.33 bits, and 60 nonterminals of depth at most 3 cover the chains with 3.0 top-level chunks per story (Figure 6). Synonymy per event symbol, counted as distinct verb realisations, ranges from 20 to 117 with a mean of 29.8 (`EMOTE.sad` 117, `MOVE.loc` 60, `EMOTE.happy` 56, `TRANSFER` 47). These are the corpus's own values of the RHM's v, s, m and L.

Synthetic lane. A hand-built generative grammar with seven story types (quest success and failure, loss recovery and loss final, sharing, mishap and repair, scare and relief) expands each into episodes, up to $m{=}3$ synonymous event sequences per episode and up to $m{=}3$ clause templates per event, with story features attached at the root; it returns text, event chain and tree together, so m can be set. Figure 5 is one sample, printed sideways on its own page so the clause and event labels are legible. The round trip from synthetic text through the extractor is the extractor's regression test (recall 81%, precision 94% per primitive).

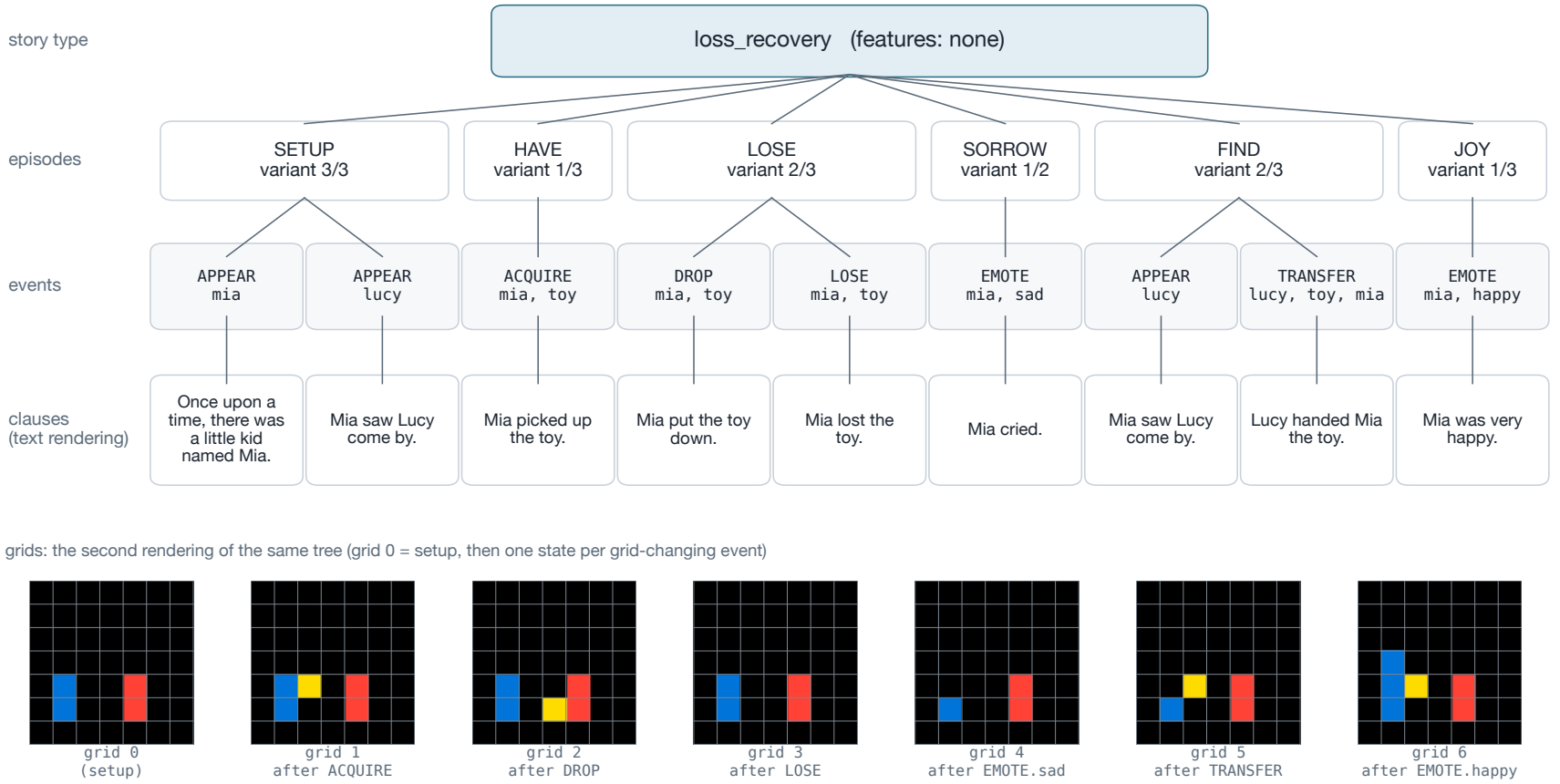


Figure 5: One synthetic story as a Narrative Hierarchy Model tree, sampled from the synthetic lane: the story type at the root, episodes (each naming which of its m synonymous expansions was drawn), the events they expand into, and the clauses that render them. Below, the second surface: grid 0 is the setup and each later grid is the state after one grid-changing event. Text and grids are two renderings of one tree.

Extraction and masking. One event per finite verb; roles from the dependency parse (agent from the subject with conjunctions expanded, theme from the object, recipient from the dative or a to- or with-phrase, location from a prepositional phrase with a location noun, mood from a copular complement, state from the verb class). Dialogue, negation, complements of desiderative verbs, purpose clauses, habituals and progressive subordinates are skipped because they do not assert the event. Pronouns are resolved by gender prior and recency with a confidence score, and a chain is used only when its minimum event confidence is at least 0.6 and its mean at least 0.75. Chains longer than eight states are compressed to keyframes: the setup, evenly spaced middle states, and the two final states so that the ending transition stays atomic; only atomic transitions are maskable, and every valid position is stored as a mask variant. On the 70,483-story shard the masked primitive is `EMOTE` in 10,740 puzzles, `VANISH` 1,581, `LOSE` 1,244, `TRANSFER` 647, `ACQUIRE` 394, `MOVE` 279, `DESCEND` 173, `APPEAR` 137, `DROP` 132, `TRANSFORM` 104 and `ASCEND` 44; puzzles have 3 grids in 4,018 cases, 4 in 4,605, 5 in 3,360, 6 in 1,897, 7 in 945 and 8 in 650.

Judges. The fidelity judge receives the story and the extracted chain with, for each event, its index, primitive, arguments and source sentence, together with the picture meanings of the eleven primitives (for instance, `APPEAR` covers arriving, being met or being first mentioned; `EMOTE` happy covers glad, excited, proud, hugging; sad covers scared, angry, upset). It returns a verdict per event, faithful or wrong (wrong participant, did not happen, wish or plan or hypothetical or negated or only in dialogue, duplicate, or clearly off), and up to three major visible events the chain missed. Table 11 gives the development rounds and the scale judging; the 200-puzzle matrix set is stratified by masked primitive from the puzzles all three judges pass.

From-scratch training. A training record is one token stream: `[TEXT]`, the lower-cased word and punctuation tokens of the text (first 400), `[GRIDS]`, then for each grid a position token `[G`i`]` followed by its cells as colour letters with a row separator, the masked grid replaced by `[MASK]`, then `[ANSWER]`, the answer cells and `[END]`; the loss is taken on the answer tokens only and exact match is scored on the decoded answer. The vocabulary is fixed at 9,229 tokens across all runs. Models are pre-norm GPTs: 4M (4 blocks, 4 heads, width 256) and 12M (6 blocks, 8 heads, width 384), context 640 or 1,024 tokens, AdamW with learning rate 3×10^{-4}, weight decay 0.1 and betas 0.9 and 0.95, batch 32, evaluation on 600 held-out puzzles every 500 or 1,000 steps with the best checkpoint kept. Sweeps 1 to 4 trained 4,000 to 6,000 steps (8,000 for 12M); the plateau runs that Section 5 reports trained 18,000 steps, at which the curves are flat from 14,000 on. Table 15 lists every configuration; Figure 12 shows the depth and noise controls of the short-training sweeps. The matched clean and rejected sets (6,824 puzzles each) were drawn to equal distributions over masked primitive, grid count and text length; injected noise replaced the answer by a grammar-consistent alternative for 15 or 30% of training records.

Probes. Figure 8 repeats the probes of Section 5.1 on the 12M checkpoints at the last text token and at the mean over text tokens. At the last text token the grammar model's event variable is stronger than at the answer token (73 against 68%) and appears one block earlier, since the grammar text names the primitive in its last clause; for the story it is weaker there (27 against 38%). At the text mean the story model's root reaches 55% at layer zero, a bag-of-words read of the feature label, against 29% for the grammar text. A story-length control is readable from the position embedding at every position (50% at layer zero for story-trained models against a 25% majority) and is therefore reported only in the tables of the released probe study, not as a selectivity control.

Yield and the ending fact. The 98,113 source stories are the 27,630 GPT-4 validation stories and a 70,483-story shard with feature labels; the generator returns 21,864 puzzles and the rest fail one of three gates (Figure 11, left): fewer than two grid-visible events, no mask position that is both ambiguous and self-taught, or pronoun resolution below the confidence gate. The corpus grammar returns one fact about the stories themselves (Figure 11, right): stories generated under a bad-ending prompt end in a sad-mood event 63% of the time, against 5 to 19% for every other feature label, so the root variable is legible in the last primitive.

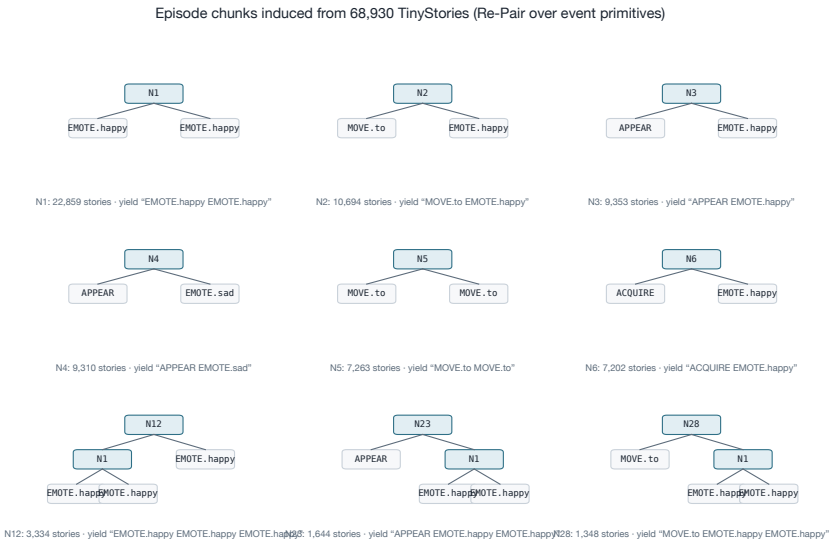


Figure 6: The most frequent Re-Pair chunks of the narrative grammar over the event alphabet, with their expansions and counts.

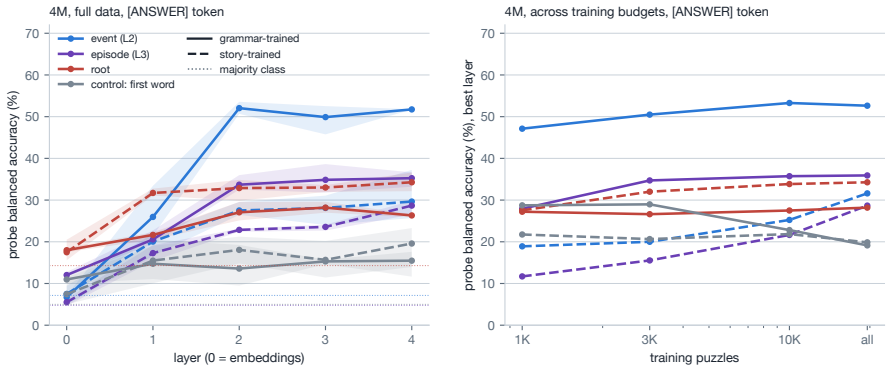


Figure 7: Linear probes on the 4M models at the answer token. Left: balanced accuracy by layer at full data for the masked event, its episode, the story-level label and a first-word control; grammar text solid, story dashed, dotted the majority baseline, bands the seed range. Right: best-layer accuracy against training budget.

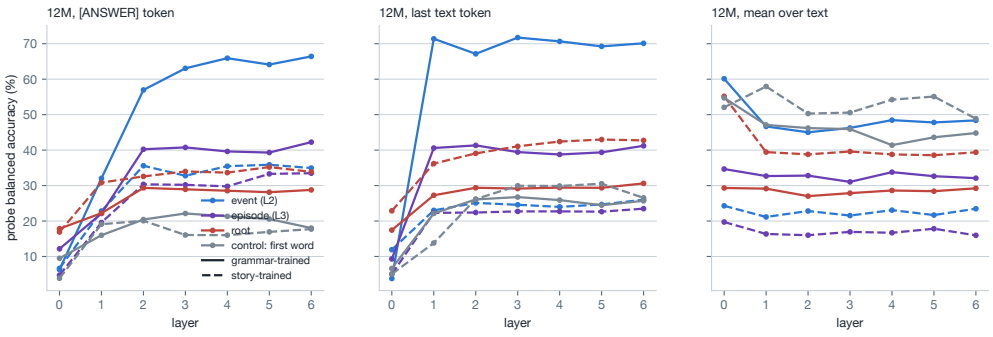


Figure 8: Probes on the 12M models at three positions: the answer token, the last text token, and the mean over text tokens; grammar text solid, story dashed.

Table 11: Extractor fidelity. Development rounds were judged by a strong closed model on a fresh set of forty puzzles each round; the frozen extractor (v0.1.7) was then judged at scale by three open-weight models over all 21,864 puzzles. A puzzle is fully faithful when every extracted event is judged asserted by its story. The clean set used below is the 11,049 puzzles all three judges pass.

Judge	Set	Events faithful	Puzzles fully faithful
closed judge, round 1 (v0.1.0)	40 fresh	78%	50%
closed judge, frozen (v0.1.7)	40 fresh	92%	68%
open judge A (27B)	all 21,864	84.5%	57%
open judge B (31B)	all 21,864	89.3%	69%
A and B unanimous	all 21,864	–	84% of puzzles
two-judge intersection (kept)	–	–	11,969 puzzles
open judge C (120B)	all 21,864	87.0%	see manifest
all three agree (kept)	–	–	11,049 puzzles

Table 12: The grammar-text control on forty stratified puzzles, one strong closed model, single sample, for two extractor versions. Left: puzzles solved per condition and NARC cells. Right: the 2×2 over the 30 puzzles (v0.1.5) not solvable from the grids alone. The grammar text is the event chain re-rendered as plain sentences, faithful to the grids by construction; it nearly doubles the story's solve rate.

Condition	v0.1.5	v0.1.6
grids only	10	13
story only	0	0
story + grids	14	16
grammar text + grids	27	32
NARC cells, story	7	6
NARC cells, grammar text	18	19

not solvable from grids (30)	story solves	story fails
grammar text solves	6	12
grammar text fails	1	11

Table 13: Eight open-weight models on 200 generated puzzles from the clean set, stratified by masked primitive, four conditions, exact match, single sample. NARC cells count puzzles solved only with text and grids together; the story alone solves none for any model. Two caveats: gpt-oss-20b returned a parsable answer on 28 of 200 story-condition trials, so its row is a lower bound, and Nemotron is near zero in every condition.

Model	Grids only	Story + grids	Grammar text + grids	Story only	NARC (story)	NARC (grammar text)
gpt-oss-120b	11	8	99	0	7	89
Gemma-4-31B	16	12	85	0	7	72
Qwen3.8-27B	7	11	78	0	11	72
Qwen3.5-122B	10	9	64	0	8	58
Qwen3.6-27B	4	11	54	0	10	51
Gemma-4-26B	3	2	27	0	1	24
gpt-oss-20b	0	1	14	0	1	14
Nemotron-3-Super	2	2	4	0	2	4

Puzzle-side (of 200): grids-sufficient for no model 174; NARC for $\geq$1 model: 31 (story), 133 (grammar text)

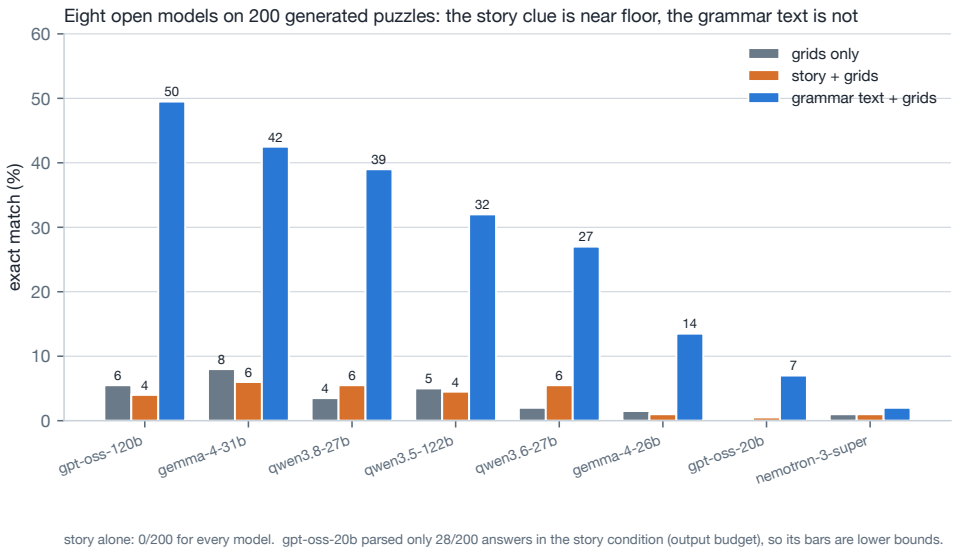


Figure 9: Table 13 drawn, with its counts shown as percentages of the 200 puzzles: eight open models, three conditions; the story alone is 0 of 200 for every model and is omitted. The story clue sits near the grids-only floor and the grammar text does not; gpt-oss-20b's story bar is a lower bound, since 28 of its 200 answers parsed.

Table 14: Learning from scratch on NARC-Tiny. Each cell is the percentage of held-out puzzles whose masked grid the 4M model reconstructs exactly, cell for cell; each column is the number of puzzles it was trained on, and every model was trained to its plateau (18,000 steps). Mean ± s.d. over seeds where more than one seed was run; single-seed cells are marked. Paraphrase = the grammar text rewritten in TinyStories style with labels clean by construction; m=3 = the grammar text and two paraphrases of every puzzle trained together, scored per surface. Read across a row for the data a surface needs and down the full-data column for the gap between surfaces. Short-training runs, depth and noise rows are in Table 15.

Text the model was trained on	Exact match (%) after training on			
	1K puzzles	3K puzzles	10K puzzles	all (∼19K) puzzles
grammar text	24.9 ± 0.8	35.5^{*}	$38.0^{\dagger}$	66.0 ± 0.7
story text	6.7^{*}	19.8 ± 2.2	31.0^{*}	40.7 ± 3.5
paraphrase, clean labels				45.1 ± 0.6
m=3, scored on the grammar text			58.0^{*}	61.2 ± 1.5
m=3, scored on a paraphrase			51–52*	56.8 ± 1.0

* one seed. † 4,000-step run; no plateau-trained run at this budget.

Table 15: Learning from scratch on NARC-Tiny, all rows. Every value is the percentage of held-out puzzles whose masked grid the model reconstructs exactly, cell for cell; every column is the number of puzzles the model was trained on. Each training record is the text followed by the grid sequence with the masked grid last, as one token stream, and the loss is taken on the answer tokens only. Sweep-1 rows: one seed, 200-puzzle evaluation. Other rows: 600-puzzle evaluation, mean $\pm$ s.d. where three seeds were run, else the seeds listed. Paraphrase = each grammar text rewritten twice in TinyStories style by a 27B model under an event-preserving instruction, labels clean by construction.

Model	Text the model was trained on	Exact match (%) after training on 1K puzzles	3K puzzles	10K puzzles	all (∼19K) puzzles
*Trained to plateau (18,000 steps), 4M; mean ± s.d. over seeds, single seeds marked **					
4M	grammar text	24.9 ± 0.8	35.5*	–	66.0 ± 0.7
4M	story text	6.7*	19.8 ± 2.2	31.0*	40.7 ± 3.5
4M	paraphrase (one realisation)				45.1 ± 0.6
4M	m=3 (grammar + two paraphrases), overall			53.6 ± 2.6	57.9 ± 0.6
4M	scored on the grammar text			58.0*	61.2 ± 1.5
4M	scored on paraphrase 1 / 2			51–52*	56.8 ± 1.0 / 55.7 ± 2.9
4M	final / middle masks at full data	grammar 70.2 / 46.2; paraphrase 48.6 / 28.6; story 43.2 / 28.6			
Short training (4,000 steps for budget runs, 6,000 at full data, 8,000 for 12M), sweeps 1–4					
4M	grammar text	24	36	38	48
4M	story text	4	5	31	31
4M	grammar text, 3 seeds				46.3 ± 1.3
4M	story text, 3 seeds				33.5 ± 3.7
4M	paraphrase (one realisation)			33.5 / 35.2	36.8 / 38.0
12M	grammar text				63.5 / 62.0
12M	story text				43.3 / 42.3
12M	paraphrase (one realisation)				48.7
Depth, all data, 600-puzzle eval: final-grid vs middle-grid masks					
4M	grammar text	51 vs 25; masks balanced in training: 51 vs 30			
12M	grammar / story	final ∼67 / ∼45; middle ∼43 / ∼31			
Label noise, story text, 4M, difficulty-matched sets of 6,824 puzzles					
4M	judge-clean vs judge-rejected (3 seeds each)	21.8 vs 19.1 (seed spread ±4)			
4M	injected grammar-consistent wrong answers, 0 / 15 / 30% of records	21.8 / 25.0 / 25.5			
Three realisations per puzzle ($m = 3$) at 6,000 steps: an under-trained comparison, superseded by the plateau block above					
4M	grammar text / paraphrases, trained on all three at once	33–37 (alone 46) / 29–37 (alone 37–38)			

Table 16: Order checks on the 200 generated puzzles, eight open models. Top: for every (puzzle, model, text) cell that was NARC in Table 13, the visible grids re-presented in three shuffled orders with the text unchanged; strong = solved in none, weak = solved in all three. Bottom: all grids shuffled and the model asked for the story order; mean Kendall τ and exact-order rate over the six models with at least 180 of 200 parsed answers (chance at most 4% at four to seven grids).

Order sensitivity	NARC cells	strong	partial	weak
grammar text + grids	384	328 (85%)	56	0
story + grids	47	33 (70%)	14	0
Ordering task	**trials**	**mean τ**	**exact order**	
grids only	1,172	0.15	17%	
story + grids	1,145	0.17	14%	
grammar text + grids	1,159	0.50	41%	

Table 17: Explicit conventions as the clue on the 200 generated puzzles, eight open models, exact match, single sample. Conventions = a plain statement of the rendering code with no events. Read each row against its grids-only and story columns: the conventions alone tell whether the story's work was teaching the code; conventions plus story tell whether the story failed because the code was not inferred.

Model	grids	story	grammar	conv.	conv.+story	conv.+grammar
gpt-oss-120b	11	8	99	2	12	117
gemma-4-31b	16	12	85	15	16	72
qwen3.8-27b	7	11	78	3	14	87
qwen3.5-122b	10	9	64	5	5	58
qwen3.6-27b	4	11	54	8	9	64
gemma-4-26b	3	2	27	3	4	28
gpt-oss-20b	0	1	14	1	4	11
nemotron-3-super	2	2	4	0	2	3

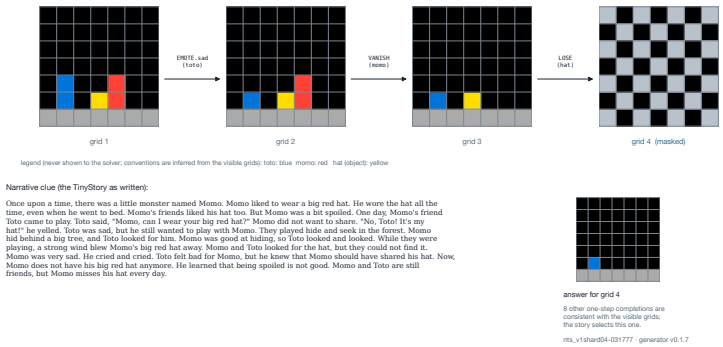


Figure 10: A generated puzzle: four grids with the extracted primitives on the arrows, the TinyStory as written as the clue, the masked fourth grid, its answer, and the count of other one-step completions consistent with the visible grids (eight here; the story selects one). The legend naming the colours is never shown to the solver, who infers the conventions from the visible grids.

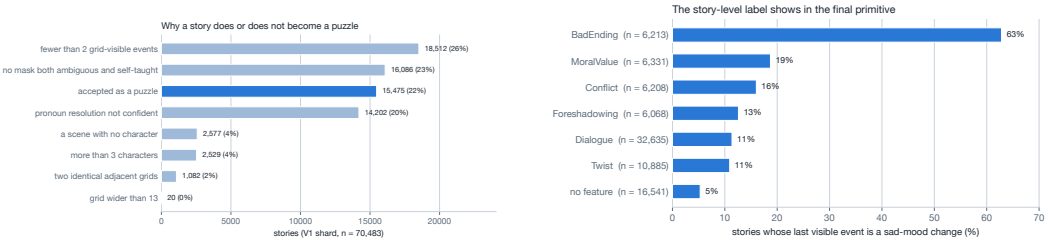


Figure 11: Left: why a TinyStory does or does not become a puzzle (V1 shard, $n = 70{,}483$; 22% accepted): too few grid-visible events, no mask position that is both ambiguous and self-taught, or unresolved pronouns. Right: the story-level label shows in the final primitive. Stories written to a bad-ending prompt end in a sad-mood event 63% of the time, against 5 to 19% for every other feature label.

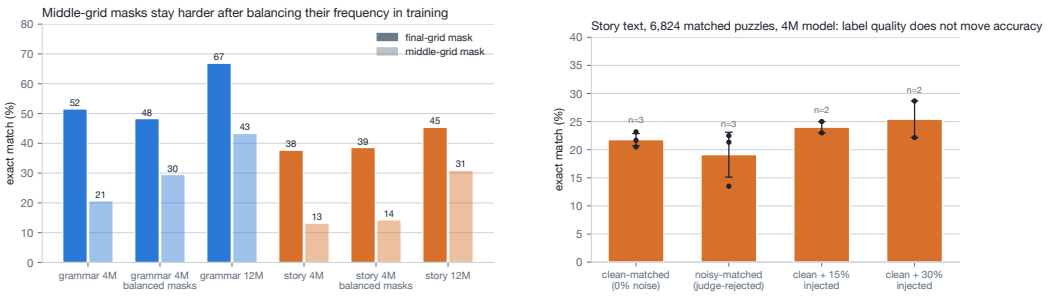


Figure 12: Left: middle-grid masks stay harder than final-grid masks after their frequency in training is balanced, at 4M and 12M. Right: story text on 6,824 difficulty-matched puzzles at 4M: judge-clean and judge-rejected sets train to the same accuracy, and injected grammar-consistent wrong answers at 15 and 30% do not lower it.