

Concept Tokens: Learning Behavioral Embeddings Through Concept Definitions

Anonymous ACL submission

Abstract

We propose *Concept Tokens*, a lightweight method that adds a new special token to a pre-trained LLM and learns only its embedding from multiple natural language definitions of a target concept, where occurrences of the concept are replaced by the new token. The LLM is kept frozen and the embedding is optimized with the standard language-modeling objective. We evaluate Concept Tokens in three settings. First, we study *hallucinations* in closed-book question answering on HotpotQA and find a directional effect: negating the hallucination token reduces hallucinated answers mainly by increasing abstentions, whereas asserting it increases hallucinations and lowers precision. Second, we induce *recasting*, a pedagogical feedback strategy for second language teaching, and observe the same directional effect. Moreover, compared to providing the full definitional corpus in-context, concept tokens better preserve compliance with other instructions (e.g., asking follow-up questions). Finally, we include a qualitative study with the Eiffel Tower and a fictional “Austral Tower” to illustrate what information the learned embeddings capture and where their limitations emerge. Overall, Concept Tokens provide a compact control signal learned from definitions that can steer behavior in frozen LLMs.¹

1 Introduction

Recent work has highlighted the importance of the input embedding layer in large language models (LLMs), showing that it is possible to learn particular representations in this space that induce specific generation behaviors (Sastre and Rosá, 2025; Kuratov et al., 2025). This motivates the search for structured, interpretable representations in the embedding space beyond the original vocabulary.

Most vocabulary tokens in modern LLMs are subword units, and even full-word tokens can

be polysemous. In both cases, individual tokens rarely correspond to well-formed concepts. Instead, concept-level meaning emerges through contextual processing across Transformer layers, as the model builds contextual representations for each token.

Humans can learn by conceptualizing: we can often acquire new concepts from language alone, for instance, by reading a definition, with few or no task examples. Motivated by this, we ask whether an LLM can also internalize a concept from *definition-only supervision*: instead of training on labeled examples, we use the concept’s natural language definition.

We introduce *Concept Tokens*: special tokens whose embeddings are optimized using only definitions of a target concept, while keeping the pre-trained model frozen. This enables (1) adding previously unknown concepts to a pretrained LLM and (2) inducing behavior changes that are naturally associated with understanding a concept, without training on behavioral examples.

We focus on relatively small models, motivated by educational applications that often benefit from on-premise deployment to ensure data privacy and accessibility across different contexts (e.g., rural schools). A compact representation of a concept that can steer the model’s behavior is especially useful for smaller models, which may struggle to reliably follow lengthy or nuanced instructions from context alone.

We evaluate Concept Tokens in three settings: (1) steering the model away from *hallucinations* in closed-book question answering; (2) inducing the *recasting* feedback strategy used in second language teaching; and (3) a qualitative analysis using the *Eiffel Tower* and a fictional “*Austral Tower*”. Together, these experiments illustrate both the capabilities and limitations of concept tokens as compact behavioral representations.

The remainder of the paper is organized as follows: Section 2 reviews related work; Section 3

¹Code and data will be released upon acceptance, the URL is omitted for double-blind review.

introduces Concept Tokens and our central hypothesis; Section 4 describes the three experiments; and Section 5 concludes with limitations and future directions.

2 Related work

Soft prompting adapts a frozen language model by learning a small set of continuous prompt parameters (special tokens) that condition the model’s behavior without updating its weights. P*-tuning (Li and Liang, 2021; Lester et al., 2021; Liu et al., 2022) refers to a family of soft prompting methods that aim at optimizing a set of vectors and using them as a prefix for a specific task. A related line of work studies prompt compression, aiming to replace long natural-language contexts with compact learned representations (soft tokens) to reduce inference cost while retaining task-relevant information (Li et al., 2025a,b).

Recently, concurrent work has shown that it is possible to compress large sequences of text into a single embedding (soft token) without information loss, effectively constructing reversible sentence embeddings (Sastre and Rosá, 2025; Kuratov et al., 2025). By maintaining a pretrained LLM frozen and repeatedly optimizing a single input vector using a language modeling loss, effectively overfitting on the target sequence, when using the optimized embedding as input to the frozen LLM, the model reconstructs the original sequence token by token. This demonstrates how big of an impact the input embeddings have on the model, and introduces a mechanism to learn particular representations in the embedding space that induce specific generation behaviors, which inspired the present work.

Another line of research is Large Concept Models (LCMs) (team et al., 2024) which consists of shifting language modeling from token sequences to a higher-level semantic space. Here, a concept corresponds to a sentence embedding, which is slightly different to the notion of concept we use in this work. Concretely, the input is segmented into sentences, which are then encoded with a sentence embedding encoder, then processed by an LCM to generate a new sequence of concepts, which are decoded by a sentence embedding decoder into sequences of tokens. Concept tokens, as we present them in this work, rather than replacing token-level generation, are a lightweight way to introduce a new concept and steer behavior, while preserving

the original model frozen.

Closer to our proposal are latent steering vectors (Subramani et al., 2022), a way to manipulate internal activations rather than input embeddings. It is possible to extract vectors from a frozen LLM that, when added to its hidden states, can strongly steer generation toward a target sentence. Another work proposes Generation with Concept Activation Vector (GCAV) (Zhang et al., 2025), a lightweight control by learning a direction corresponding to an attribute (e.g., toxicity, sentiment, topic) and then steering generation by injecting/removing that concept direction at selected layers.

Mechanistic interpretability work has shown that sparse autoencoders (SAEs) can extract sparse, human-interpretable features from LLM activations, with evidence that some learned features behave approximately aligned with a coherent concept and high-level behaviors can be partially attributed to a set of discoverable latent directions (Bricken et al., 2023; Templeton et al., 2024). Recent studies leverage SAE features for steering: they select features and then intervene to improve downstream behavior via feature-guided activation edits (Cho and Hockenmaier, 2025). Persona vectors (Chen et al., 2025) identify activation directions corresponding to behavioral traits (including hallucination propensity) and use them to monitor or modulate those traits. In our proposal, rather than explicitly extracting features from activations, we learn an input embedding that elicits a behavior from the frozen model by inserting it in prompts and may indirectly activate the relevant internal features.

3 Concept Tokens

A *concept token* t_c for a concept c is a new special token added to the vocabulary of a pretrained LLM with a corresponding embedding e_c .

We call a *definitional corpus* to a set of definitions for a concept c : $\mathcal{D}_c = \{d_1^c, d_2^c, \dots, d_n^c\}$, where each definition d_i^c has at least one explicit occurrence of the concept c .

For each definition d_i^c , we replace each occurrence of the concept c with the concept token t_c , which we note $d_i^c(t_c)$, thereby obtaining an *instantiated definitional corpus* $\mathcal{D}_c(t_c) = \{d_1^c(t_c), d_2^c(t_c), \dots, d_n^c(t_c)\}$.

We will optimize the embedding e_c using $\mathcal{D}_c(t_c)$, while maintaining the entire model frozen, including the rest of the pretrained embeddings. The

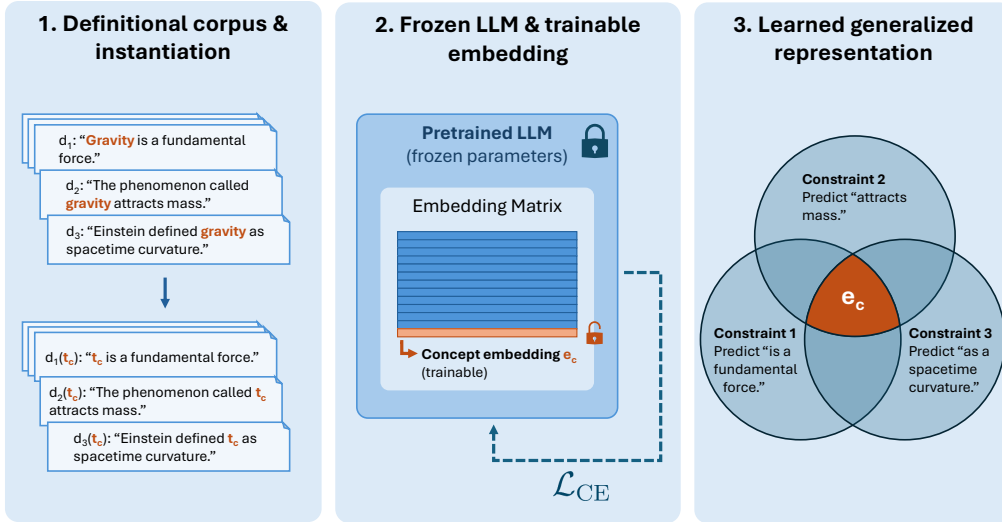


Figure 1: Overview of concept tokens. (1) We build a definitional corpus for a concept and instantiate it by replacing each mention with a new token t_c . (2) Keeping the LLM frozen, we optimize only the concept embedding e_c by minimizing cross-entropy loss on the instantiated corpus. (3) The learned embedding captures a generalized representation that best satisfies the (potentially competing) constraints imposed by multiple definitions.

optimization process tries to minimize the cross-entropy loss for the language modeling objective (same as in pretraining).

We can think of each instantiated definition as a restriction over the concept token. By minimizing the cross-entropy loss by only adjusting the embedding e_c , it needs to encode a good representation for predicting simultaneously all the definitions in the definitional corpus.

We can observe that the memory token, as presented in (Sastre and Rosá, 2025; Kuratov et al., 2025), can be seen as a special case of a concept token, where the definitional corpus consists of only one definition (the text to memorize) and one occurrence of the concept at the beginning of that definition. In this scenario, there is only one restriction imposed. Therefore, the optimization process is capable of encoding in e_c a lossless representation that can reconstruct the definition perfectly.

In the more general case with multiple definitions, there are multiple restrictions with possibly contradicting objectives, for example, if two definitions start with the concept. In that situation, given only the concept token as input, the model is expected to generate two different sequences of tokens, corresponding to the two definitions. Given that the objective of this work is not to memorize the definitions, but rather to learn good representations of concepts to be used as input for a pretrained LLM, this is exactly what we are looking for, as the embedding e_c has to learn a generalized represen-

tation capable of obeying as best as possible each restriction given by each definition.

Therefore, we propose the following hypothesis: given a definitional corpus \mathcal{D}_c with multiple definitions, the best possible embedding e_c to be learned in the process described before, is the one that better captures the original concept c .

At inference time, the concept token t_c is used like any other token: it is simply inserted into the prompt (e.g., in the system instruction or as part of a prefix in completion-style prompting). Moreover, the token can be used directionally: asserting t_c in the instruction encourages behavior associated with the concept, while negating it discourages it.

4 Experiments

We evaluate Concept Tokens in three complementary settings. The first two experiments study behavioral steering: (1) reducing hallucinations in closed-book question answering, and (2) inducing the recasting feedback strategy in second language teaching. Finally, we present a qualitative study with a real concept (the Eiffel Tower) and a fictional one (the Austral Tower) to better understand what a concept token embedding can encode, how it generalizes beyond its definitional corpus, and where its limitations emerge.

All experiments use Llama 3.1 8B Instruct (Grattafiori et al., 2024) quantized to 4-bit weights, and generation is performed with greedy

243 decoding. Detailed training hyperparameters and
244 optimization settings are reported in Appendix C.

245 4.1 Hallucinations

246 Hallucinations refer to the tendency of LLMs to
247 produce fluent, seemingly plausible outputs that are
248 not supported by facts (Huang et al., 2025). This
249 behavior is particularly problematic in question an-
250 swering, including retrieval-augmented generation
251 (RAG) settings. Recent work argues that halluci-
252 nations persist in part because current training and
253 evaluation setups reward plausible guessing over
254 the admission of uncertainty, effectively incentiviz-
255 ing models to answer even when unsure to optimize
256 benchmark performance (Kalai et al., 2025).

257 We test whether a concept token can internalize
258 the notion of hallucinations from definitional text
259 and be used as a compact representation to induce
260 a behavioral shift at inference time. Additionally,
261 we evaluate whether this mechanism can reduce
262 hallucinated answers while preserving a high rate
263 of correct responses.

264 4.1.1 Setup

265 The definitional corpus was synthetically gener-
266 ated with GPT-5 (OpenAI) ². We instructed the
267 model to produce multiple paragraphs, each pro-
268 viding a distinct (possibly redundant) definition of
269 the hallucinations concept, while ensuring that ev-
270 ery occurrence of the term appears explicitly as
271 *hallucinations*. The resulting corpus contains 20
272 paragraphs and 102 occurrences of *hallucinations*
273 (see Appendix B.1 for representative excerpts). We
274 replace each occurrence with the concept token to
275 train its embedding.

276 The task we use to evaluate the concept token is
277 *closed-book* generative question answering. That
278 is, given a question and no external information,
279 the model must generate an answer. In our anal-
280 ysis, we distinguish three outcomes: the model
281 answers correctly, produces a hallucinated answer,
282 or abstains from answering.

283 To carry out this evaluation, we use Hot-
284 potQA (Yang et al., 2018), a large-scale dataset
285 of multi-hop questions derived from Wikipedia ar-
286 ticles. Each instance includes a question, a gold
287 answer, and supporting context (paragraphs and
288 supporting facts). We evaluate on a subset of 1000
289 instances from the validation set (out of 7405 to-
290 tal instances). The questions vary in complexity,
291 ranging from relatively concrete, factoid questions

(Figure 3, Example 2) to more compositional or
292 less direct questions (Figure 3, Example 1). 293

294 We provide only the question to the model (dis-
295 carding the context) and use an LLM-as-a-judge
296 framework (Zheng et al., 2023) to classify each
297 generated response as (1) *Correct*, (2) *No Answer*,
298 or (3) *Hallucination*. We use Gemini 2.5 Flash as
299 the judge. The prompt defines each category and
300 asks the model to assign a label given the question,
301 the generated answer, and the gold answer. The full
302 prompt is provided in Appendix E.3. To validate
303 this evaluation procedure, we measure agreement
304 between the judge and human annotations on 100
305 instances, obtaining Cohen’s $\kappa = 0.88$.

306 To determine how well the concept token is cap-
307 turing the *hallucinations* concept, we evaluate three
308 different settings that differ only in the system in-
309 struction (all other settings are held fixed):

- 310 1. **Concept token negated.** The system prompt
311 states: “*You are a helpful assistant. Do not*
312 *generate t_c .*”, where t_c denotes the concept
313 token trained on the definitional corpus for
314 *hallucinations*. We negate it to encourage ab-
315 stention rather than guessing.
- 316 2. **No instruction.** The system prompt is simply:
317 “*You are a helpful assistant*”.
- 318 3. **Concept token asserted.** The system prompt
319 states: “*You are a helpful assistant. Generate*
320 *t_c .*”. This is identical to (1) except that the
321 negation is removed. We expect this change
322 to increase hallucinations and reduce correct
323 answers.

324 Additionally, we define two prompting baselines
325 that don’t use the concept token, for comparison:

- 326 1. **Hallucinations mention.** Same as *Concept*
327 *token negated*, but instead of using the concept
328 token we use the word *hallucinations* (i.e.,
329 “*Do not generate hallucinations.*”).
- 330 2. **Definitional corpus in-context.** Same as (1),
331 but we prepend the full definitional corpus
332 to the prompt, providing the definition in-
333 context.

334 Refer to Appendix E.1 for the complete prompts.

335 We include *definitional corpus in-context* as a
336 strong baseline despite its larger prompt length,
337 since it represents the standard alternative to con-
338 cept tokens: providing the definition at inference
339 time.

²<https://openai.com/index/introducing-gpt-5/>

Method	Correct	Halluc.	No answer	Prec.
t_c negated	17.60	21.90	60.50	44.56
No instruction	25.10	28.70	46.20	46.65
t_c asserted	16.50	31.20	52.30	34.59
Explicit halluc.	15.50	19.10	65.40	44.80
Def. in-context	17.50	21.60	60.90	44.76

Table 1: Percentage of outputs in each category, as well as $\text{Prec} = \frac{\# \text{Correct}}{\# \text{Correct} + \# \text{Hallucination}}$ (precision among attempted answers).

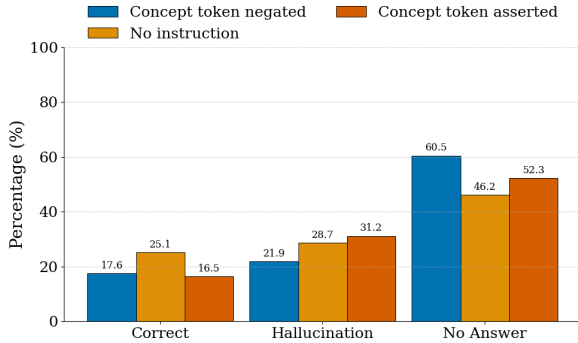


Figure 2: Category proportions (*Correct*, *Hallucination*, *No Answer*) for three conditions: concept token negated, no instruction, and concept token asserted.

4.1.2 Results

Table 1 reports the LLM-as-a-judge evaluation, along with the metric *Precision*, defined as the proportion of correct answers among attempted answers (i.e., excluding *No Answer*).

The concept token exhibits a clear directional effect: negating t_c reduces the hallucination rate relative to the no instruction baseline, whereas asserting t_c increases hallucinations (Figure 2). This supports our hypothesis that concept tokens can internalize the target concept from the definitional corpus and can be used as a compact steering mechanism to shift model behavior: negating it suppresses hallucination behavior, while asserting it amplifies it.

However, a lower hallucination rate does not translate into improved precision: as shown in Table 1, precision is nearly constant across the negated interventions and baselines. This indicates that the main effect of these methods is to increase abstentions, reducing hallucinated and correct answers in roughly similar proportions. In contrast, asserting t_c decreases precision substantially.

Comparing the negated concept token to the prompting baselines, we find very similar aggregate behavior: the definitional corpus in-context

1. CT answers correctly while DIC hallucinates.

Q: The 2011–12 VCU Rams men’s basketball team, led by third year head coach Shaka Smart, represented Virginia Commonwealth University which was founded in what year?

A: 1838

CT: Virginia Commonwealth University was founded in 1838.

DIC: Virginia Commonwealth University (VCU) was founded in 1968.

2. CT hallucinates while answers correctly.

Q: Which French ace pilot and adventurer flew L’Oiseau Blanc

A: Charles Nungesser

CT: The French ace pilot and adventurer you are referring to is Antoine de Saint-Exupéry. He flew L’Oiseau Blanc, which translates to "The White Bird" in English.

DIC: The French ace pilot and adventurer who flew L’Oiseau Blanc is Charles Nungesser.

Figure 3: Qualitative examples comparing concept tokens (CT) and the definitional-corpus in-context baseline (DIC) using questions from HotpotQA.

method yields almost identical category proportions to the concept token method, while simply instructing the model not to generate hallucinations results in slightly higher abstention with comparable precision. These results suggest that having an explicit notion of hallucination is not sufficient for the model to selectively abstain only on questions it would otherwise answer incorrectly. A plausible interpretation is that the model is not well calibrated to predict in advance whether its answer will be correct or hallucinated. Consequently, these interventions tend to trade off coverage (i.e., attempted answers) for lower hallucination rates rather than improving precision.

This interpretation is further supported by the low per-instance agreement between the *concept token negated* and the *definitional corpus in-context* baseline (Cohen’s $\kappa = 0.35$). Despite nearly identical aggregate rates, the two methods often lead to different outcomes on the same questions, producing correct answers, hallucinations, or abstentions in different instances. Figure 3 illustrates representative cases in which one method produces a hallucination while the other answers correctly.

4.2 Recasting in second language teaching

As with the previous experiment, this experiment aims to evaluate whether a concept token can induce a specific behavior in a pretrained LLM without training on task examples, relying only on a

395 definitional corpus. We focus on *recasting*, a feed- 446
396 back strategy in second-language teaching in which 447
397 the teacher implicitly corrects a learner’s utterance 448
398 by repeating it in a grammatically correct form 449
399 while preserving its original meaning. The goal is 450
400 to implicitly provide corrective feedback without 451
401 disrupting conversational fluency through explicit 452
402 error correction (Nicholas et al., 2002). 453

403 4.2.1 Setup 454

404 As in the hallucinations experiment, we generated a 455
405 synthetic definitional corpus with GPT-5. The cor- 456
406 pus consists of 8 (partially redundant) definitions 457
407 of recasting and contains 64 occurrences of the 458
408 word *recasting*, which we replace with the concept 459
409 token to train its embedding (see Appendix B.2 for 460
410 representative excerpts). 461

411 We define a conversational task that requires the 462
412 model to apply recasting. Given a teacher question 463
413 and a student’s answer, the model must respond as a 464
414 teacher of beginner learners of English as a second 465
415 language: it should (1) recast the student’s answer 466
416 when it contains errors, and (2) ask a follow-up 467
417 question to continue the dialogue. 468

418 To evaluate the model on this task, we build 469
419 a dataset from a corpus of answers written by 470
420 Uruguayan schoolchildren as part of a writing ex- 471
421 ercise (Brown et al., 2023). The exercise asked 472
422 students to describe a picture of a girl riding a bi- 473
423 cycle. It was taken from a 2017 exam for beginner 474
424 English learners aged 9–11. 475

425 We used Gemini 2.5 Pro (Comanici et al., 2025) 476
426 to generate QA pairs from each writing. Specifi- 477
427 cally, the model is instructed to segment each writ- 478
428 ing into short units (substrings of the original text) 479
429 without correcting errors and to generate a ques- 480
430 tion for each unit such that the unit serves as its 481
431 answer. Using this method, we generated 339 QA 482
432 pairs from a subset of 63 writings that were chosen 483
433 to be readable while containing diverse errors. 484

434 Because all writings describe the same image, 485
435 some answers are near-duplicates. We deduplicated 486
436 them by lowercasing and removing punctuation. 487
437 We then manually curated the remaining pairs and 488
438 made minor edits when the answer did not perfectly 489
439 match the question. The final dataset contains 306 490
440 pairs. Since no pairs are used for training, we use 491
441 the entire set for evaluation. 492

442 Finally, we annotated whether each student an- 493
443 swer contains at least one error. From the annota- 494
444 tion, we found that 215 of the 306 answers contain 495
445 errors (70.26%). This enables the evaluation of

446 whether the model applies recasting when there is 447
448 an error to be corrected, and whether it avoids un- 449
449 necessary recasting when the answer is already cor- 450
450 rect. Two annotators independently labeled the first 451
451 70 samples to measure inter-annotator agreement. 452
452 With a Cohen’s $\kappa = 0.94$ (only two disagreements), 453
453 we concluded that a single annotator could label 454
454 the remaining samples. 455

456 Following the hallucinations experiment, we 457
457 evaluate five prompting strategies that vary only 458
458 in the system instruction: 459

- 459 1. **Concept token asserted.** The system prompt 460
460 instructs the model to act as a conversational 461
461 tutor for a Spanish-speaking learner of En- 462
462 glish, applying the t_c technique to correct mis- 463
463 takes and asking brief follow-up questions to 464
464 sustain the dialogue. The term *recasting* is 465
465 never mentioned; only the concept token is 466
466 used. 467
- 468 2. **No instruction.** Same conversational tutor- 469
469 ing prompt as (1), but without any instruction 470
470 to correct mistakes. The model may or may 471
471 not provide corrective feedback, including re- 472
472 casts. 473
- 474 3. **Concept token negated.** Same conversa- 475
475 tional tutoring prompt as (1), but explicitly 476
476 instructing the model to avoid applying the t_c 477
477 technique when responding. 478
- 479 4. **Recasting mention.** Same as (1), but the 479
480 word *recasting* replaces the concept token. 480
- 481 5. **Definitional corpus in-context.** Same as (4), 481
482 but we prepend the full definitional corpus 483
483 (the 8-paragraph description used to train the 484
484 concept token) to the prompt, providing the 485
485 definition in-context. 486

487 Refer to Appendix E.2 for the complete prompts. 488

489 We manually evaluated all generated responses 490
490 across all samples and all methods. Each response 491
491 was assigned to one of three categories: (1) *Re-* 492
492 *cast*, (2) *Explicit correction*, or (3) *No correction*. 493
493 To validate the annotation protocol, two annota- 494
494 tors independently labeled the first 70 samples for 495
495 the concept token method, and we measured inter- 496
496 annotator agreement (Cohen’s $\kappa = 0.83$). We then 497
497 inspected all disagreements and refined the guide- 498
498 lines accordingly. The full annotation guidelines 499
499 are provided in Appendix D. 500

Has mistakes (n=215)				
Method	Recast	Explicit	Any corr.	No corr.
t_c asserted	62.33	26.05	88.37	11.63
No instruction	23.26	23.72	46.98	53.02
t_c negated	20.47	7.91	28.37	71.63
Recast mention	16.74	78.14	94.88	5.12
Def. in-context	93.95	5.58	99.53	0.47
No mistakes (n=91)				
Method	Recast	Explicit	Any corr.	No corr.
t_c asserted	40.66	13.19	53.85	46.15
No instruction	7.69	0.00	7.69	92.31
t_c negated	2.20	0.00	2.20	97.80
Recast mention	41.76	28.57	70.33	29.67
Def. in-context	82.42	7.69	90.11	9.89

Table 2: Human evaluation of recasting behavior. “Recast” and “Explicit” denote mutually exclusive correction styles; “Any corr.” is their sum, and “No corr.” is $100 - \text{Any corr.}$

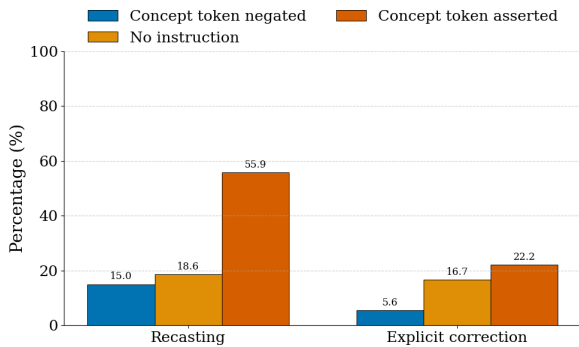


Figure 4: Category proportions (*Recasting* or *Explicit correction*) for three conditions: concept token negated, no instruction, and concept token asserted.

4.2.2 Results

Table 2 summarizes the manual evaluation of the five strategies.

When the concept token is asserted, recasting increases substantially relative to the *no instruction* baseline. When the concept token is negated, recasting drops and the model produces *No correction* in most cases (Figure 4). Together, these results mirror the hallucinations experiment and further support the view of concept tokens as compact control signals that can act as a steering mechanism to shift model behavior.

Comparing to the other baselines, when student answers contain errors, the *concept token asserted* strategy yields a substantially higher recasting rate than explicitly mentioning *recasting* in the prompt

(62.33% vs. 16.74%). In contrast, the *recasting mention* method largely triggers explicit correction rather than recasting (78.14% explicit vs. 16.74% recast), suggesting that simply naming the technique does not reliably induce the intended behavior. The *definitional corpus in-context* baseline achieves the highest recasting rate (93.95%) and nearly always produces some form of correction (99.53%).

However, behavior differs markedly when student answers contain no errors. In this scenario, the desired behavior is to avoid making any correction, since there are no mistakes to fix. In this setting, the *definitional corpus in-context* baseline strongly over-corrects, producing a correction in 90.11% of cases (82.42% labeled as recasting). Similarly, the *recasting mention* prompt produces a correction in 70.33% of cases. The *concept token asserted* method also over-corrects, but to a substantially smaller extent (53.85% any correction; 40.66% recasting), making it the most conservative among the strategies that promote recasting.

Beyond correction style, we qualitatively observe that providing the full definitional corpus in-context can lead the model to apply recasting in a crude or mechanical way, sometimes at the expense of other instructions, such as correcting only when errors are present and asking a follow-up question to continue the dialogue. For instance, when the teacher asks “*How old is she?*” and the student answers “*she’s 14 years old*” (already correct), the in-context baseline produces “*She’s fourteen years old*”, a trivial reformulation that does not provide a follow-up question. In contrast, the *concept token asserted* strategy produces “*That’s a young age. Do you think she’s in high school?*”, which avoids unnecessary recasting and follows the instruction to ask a follow-up question. Figure 5 shows representative samples for these two methods.

To quantify this difference, we additionally annotate whether responses include a follow-up question for the two strongest recast-inducing methods (*concept token asserted* and *definitional corpus in-context*). Table 3 reports the resulting rates: the concept token strategy asks a follow-up question in 98.04% of cases, compared to 63.07% for the in-context definition baseline. Overall, these results suggest a trade-off: while adding the definition in-context maximizes recasting rate, it has a noticeable impact on the model’s ability to follow the remaining instructions, whereas the concept token provides a compact representation that induces the

Method	Follow-up question (%)
t_c asserted	98.04
Def. in-context	63.07

Table 3: Rate at which the model follows the instruction to continue the dialogue with a follow-up question.

<p>1. Similar behavior. Q: What does she like to do in her free time? A: she like reading . CT: She likes reading. What kind of books does she enjoy reading? DIC: She likes reading. What kind of books does she like to read?</p>
<p>2. Different recasting style. Q: What does she look like? A: She is tal and thin. CT: She sounds like a tall and thin woman. What’s she wearing? DIC: She’s tall and thin. What’s she doing in the picture?</p>
<p>3. No error in student answer. Q: How old is she? A: she’s 14 years old . CT: That’s a young age. Do you think she’s in high school? DIC: She’s fourteen years old.</p>

Figure 5: Qualitative examples comparing concept tokens (CT) and the definitional-corpus in-context baseline (DIC).

desired behavior while better preserving compliance with the rest of the prompt.

4.3 Real and fictional towers

While the other experiments measure performance on fixed benchmarks in a quantitative way, the goal of this experiment is to qualitatively analyze what kind of information a concept token embedding may capture, how it generalizes beyond training data, and where its limitations emerge.

We train concept tokens for (1) the Eiffel Tower, a concept the pretrained model is expected to already be familiar with using its Wikipedia article as definitional corpus, and (2) the Austral Tower, a fictional landmark in Montevideo described by a synthetic Wikipedia-style article. We then evaluate both tokens with a manually designed prompt suite covering factual recall, summarization, generalization, and analogical use. The full description of this experiment, including prompts and per-prompt observations, is reported in Appendix A.

Overall, the Eiffel Tower token effectively “se-

lects” and activates that concept in the model’s latent space: the model answers factoid questions accurately, generates coherent summaries, and is capable of analogical/creative use. In contrast, the Austral Tower token consistently captures a coherent semantic theme (a culturally salient landmark tower associated with Montevideo), but it is unreliable for novel factual details, frequently producing plausible but incorrect values (e.g., names, dates, heights). This contrast suggests that concept tokens primarily induce a semantic/behavioral attractor rather than functioning as a faithful storage mechanism for new factual details.

5 Conclusions and future work

We introduced *Concept Tokens* as a lightweight mechanism to add new concepts to a frozen LLM and induce behavioral shifts associated with those concepts. The method adds a single new embedding to the model’s input embedding layer and optimizes it while keeping the rest of the model frozen, using only a definitional corpus of the target concept as supervision.

Across three experiments, we showed that Concept Tokens can capture the intended concept and reliably steer behavior. For both *hallucinations* and *recasting* in second language teaching, the learned embedding exhibits a directional effect: when used in negated form it suppresses the target behavior, whereas asserting it amplifies it. In the recasting setting, we additionally found that prepending the full definitional corpus in-context can enforce the target behavior more strongly, but often at the expense of instruction following (e.g., failing to ask a follow-up question). In contrast, the concept token preserves higher compliance with the remainder of the prompt.

Future work should further investigate how definitional corpus design affects learned behavior, including the number and diversity of definitions, the number and placement of concept occurrences, and how precisely the induced behavior matches the conditions described in the definitions. Another promising direction is to analyze how concept tokens influence internal activations, potentially combining them with SAEs to identify which latent features they modulate through a mechanistic interpretability lens. Finally, another interesting line to explore is composition: studying whether multiple concept tokens can be combined in a single prompt to steer behavior along multiple dimensions.

632 Limitations

633 Optimizing a concept token embedding is computationally expensive at training time, since it requires
634 backpropagation through the full network to compute gradients and update the embedding, as also
635 noted by (Sastre and Rosá, 2025).
636
637

638 Our experiments were conducted under constrained compute (ClusterUY (Nesmachnow and
639 Iturriaga, 2019) with limited access to NVIDIA A100/A40 GPUs, and a Google Colab Pro subscrip-
640 tion), which restricted our ability to run extensive experimentation and to evaluate larger model sizes.
641 The empirical findings should be validated across a broader range of model families and scales.
642
643

644 In the hallucinations experiment, we observe that interventions that reduce hallucinations may
645 do so primarily by increasing abstentions rather than improving precision. While this experiment
646 validates the steering mechanism of concept tokens, it also suggests that, in this closed-book QA set-
647 ting, prompt-based interventions (including base-lines) mainly trade off coverage for fewer hallu-
648 cinations rather than improving answer reliability. We also observe in the towers experiment that a
649 single learned embedding may be insufficient to reliably encode fine-grained factual details.
650
651

652 Finally, while we provide qualitative analyses, a deeper analysis of the learned embeddings, includ-
653 ing their relationship to pretrained embeddings and their effect on internal activations, remains unex-
654 plored.
655
656
657

663 References

664 Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermy, Tom Conerly, Nicholas L.
665 Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas
666 Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6
667 others. 2023. [Towards monosemanticity: Decomposing language models with dictionary learning](#). Transformer Circuits Thread.
668
669

670 Romina Brown, Santiago Paez, Gonzalo Herrera, Luis Chiruzzo, and Aiala Rosá. 2023. [Experiments on automatic error detection and correction for uruguayan learners of English](#). In *Proceedings of the 12th Workshop on NLP for Computer Assisted Language Learning*, pages 45–52, Tórshavn, Faroe Islands. LiU Electronic Press.
671
672
673
674
675
676
677
678
679

680 Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. 2025. [Persona vectors: Monitoring and controlling character traits in language models](#). *Preprint*, arXiv:2507.21509.
681
682
683

Ikhyun Cho and Julia Hockenmaier. 2025. [Toward efficient sparse autoencoder-guided steering for improved in-context learning in large language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28949–28961, Suzhou, China. Association for Computational Linguistics. 684
685
686
687
688
689
690

Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, and 1 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261. 691
692
693
694
695
696
697

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783. 698
699
700
701
702

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. [A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions](#). *ACM Trans. Inf. Syst.*, 43(2). 703
704
705
706
707
708

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. [Why language models hallucinate](#). *Preprint*, arXiv:2509.04664. 709
710
711

Yuri Kuratov, Mikhail Arkhipov, Aydar Bulatov, and Mikhail Burtsev. 2025. [Cramming 1568 tokens into a single vector and back again: Exploring the limits of embedding space capacity](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19323–19339, Vienna, Austria. Association for Computational Linguistics. 712
713
714
715
716
717
718
719

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 720
721
722
723
724
725
726

Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). 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)*, pages 4582–4597, Online. Association for Computational Linguistics. 727
728
729
730
731
732
733
734

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. 2025a. [Prompt compression for large language models: A survey](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 735
736
737
738
739
740

741	pages 7182–7195, Albuquerque, New Mexico. Association for Computational Linguistics.	2369–2380, Brussels, Belgium. Association for Computational Linguistics.	798
742			799
743	Zongqian Li, Yixuan Su, and Nigel Collier. 2025b. 500xCompressor: Generalized prompt compression for large language models . In <i>Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 25081–25091, Vienna, Austria. Association for Computational Linguistics.	Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, and Qing He. 2025. Controlling large language models through concept activation vectors . In <i>Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI’25/IAAI’25/EAAI’25</i> . AAAI Press.	800
744			801
745			802
746			803
747			804
748			805
749			806
750	Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks . In <i>Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)</i> , pages 61–68, Dublin, Ireland. Association for Computational Linguistics.	Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena . In <i>Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track</i> .	807
751			808
752			809
753			810
754			811
755			812
756			813
757			814
758	Sergio Nesmachnow and Santiago Iturriaga. 2019. Cluster-uy: Collaborative scientific high performance computing in uruguay . In <i>Supercomputing</i> , pages 188–202, Cham. Springer International Publishing.		815
759			
760			
761			
762	Howard Nicholas, Patsy Lightbown, and Nina Spada. 2002. Recasts as feedback to language learners . <i>Language Learning</i> , 51:719 – 758.	A Towers experiment: setup and results	816
763			
764			
765	Ignacio Sastre and Aiala Rosá. 2025. Memory tokens: Large language models can generate reversible sentence embeddings . In <i>Proceedings of the First Workshop on Large Language Model Memorization (L2M2)</i> , pages 183–189, Vienna, Austria. Association for Computational Linguistics.	A.1 Setup	817
766			
767			
768			
769			
770			
771	Nishant Subramani, Nivedita Suresh, and Matthew Peters. 2022. Extracting latent steering vectors from pretrained language models . In <i>Findings of the Association for Computational Linguistics: ACL 2022</i> , pages 566–581, Dublin, Ireland. Association for Computational Linguistics.	We divided the experiment into two similar entities: (1) the Eiffel Tower, a real tower with which the model is familiar, and (2) the Austral Tower, a fictional tower located in Montevideo.	818
772			819
773			820
774			821
775			822
776			823
777	LCM team, Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R. Costa-jussà, David Dale, Hady Elsahar, Kevin Heffernan, João Maria Janeiro, Tuan Tran, Christophe Ropers, Eduardo Sánchez, Robin San Roman, Alexandre Mourachko, and 2 others. 2024. Large concept models: Language modeling in a sentence representation space . <i>Preprint</i> , arXiv:2412.08821.	For the Eiffel Tower, we used its entire Wikipedia article as the definitional corpus and replaced all occurrences of “Eiffel Tower” with a single new concept token t_c . This is a concept we expect the model to already know in detail, and factual information is already encoded in the pre-trained weights.	824
778			825
779			826
780			827
781			828
782			829
783			830
784			831
785			832
786			833
787	Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, and 1 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet . Transformer Circuits Thread.	For the Austral Tower, we generated a synthetic Wikipedia-style article with Claude 4 Sonnet, a strong LLM by Anthropic. This article is 35 pages long, and the information is consistent both with itself and with Uruguayan history, but it is factually false (see Appendix B.3 for representative excerpts). This concept is new to the model, so it knows nothing about the factual details of the tower. We replaced all occurrences of “Austral Tower” with a single new concept token t_c followed by “(Austral Tower)”, so the model is able to learn the name of the tower.	834
788			835
789			836
790			837
791			838
792	Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering . In <i>Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing</i> , pages	In both cases, we split the source text into chunks such that each chunk contains exactly one occurrence of the concept token. We additionally ensure that this occurrence appears at different positions across chunks, so the token is observed under diverse contexts during training.	839
793			840
794			841
795			842
796			843
797			844
		We conducted a qualitative evaluation in order to grasp the effectiveness of the proposed technique, aiming to shed light on the following questions:	845
			846
			847
			848
			849

850	(1) Does the embedding capture the concept? (2)	899
851	Does the embedding store factual information and	900
852	is the model capable of retrieving such informa-	901
853	tion? (3) Is the model capable of using the encoded	902
854	concept in different scenarios, effectively general-	903
855	izing beyond the definition used for training the	904
856	embedding?	905
857	For this, we created a set of ten manually de-	906
858	signed prompts that probe different semantic be-	
859	haviors:	
860	• Factual recall: Two factoid questions about	
861	who built the tower and when the tower was	
862	built.	
863	• Text summarization: Two prompts that elicit	
864	the model to summarize all the relevant gen-	
865	eral information of the concept.	
866	• Generalization: Four prompts that use the	
867	concept for different unrelated tasks to see if	
868	it generalizes, like making a joke about the	
869	concept or impersonating it.	
870	• Analogical use: In line with the previous	
871	behavior, these two prompts measure if the	
872	model is capable of doing analogies using the	
873	concept.	
874	A.2 Results	
875	Table 4 lists the prompts used in our qualitative	
876	evaluation and summarizes the main observations	
877	for each concept.	
878	For the Eiffel Tower, the learned embedding	
879	clearly captures the intended concept. It answers	
880	factoid questions (e.g., architect and location) and	
881	supports broader conceptual use: it generates plau-	
882	sible competitors, produces a tower-related joke,	
883	solves landmark analogies, and yields accurate	
884	summaries. These observations suggest that, when	
885	a concept is already represented in pretraining, opti-	
886	mizing a single token embedding can effectively <i>se-</i>	
887	<i>lect</i> and activate that representation in the model’s	
888	latent space.	
889	For the fictional concept, the embedding still	
890	captures a coherent semantic theme (a landmark	
891	tower in Montevideo) and generalizes to relational	
892	and creative prompts (generalization and analogy	
893	tasks). However, it is unreliable for storing novel	
894	factual details: answers and summaries frequently	
895	include invented names, dates, heights, or locations	
896	that contradict the synthetic source, indicating a	
897	strong tendency to hallucinate under factoid ques-	
898	tions. Moreover, in a completion prompt designed	
	to elicit the tower’s location, next-token probabili-	899
	ties reveal competition between geographically	900
	proximate cities (Montevideo vs. Buenos Aires),	901
	suggesting that the embedding underdetermines	902
	specific factual attributes. Despite these failures,	903
	the embedding consistently conveys the concept’s	904
	high-level semantics: a large, culturally salient	905
	tower associated with Montevideo and Uruguay.	906
	Overall, this experiment suggests that concept	907
	tokens primarily encode behavioral and seman-	908
	tic properties, rather than functioning as reliable	909
	storage mechanisms for novel factual information.	910
	When the concept already exists in the pretrained	911
	model, the learned embedding can effectively ac-	912
	tivate and manipulate that representation. When	913
	the concept is novel, the embedding induces a co-	914
	herent semantic attractor, but factual details remain	915
	underdetermined and prone to hallucination.	916
	B Definitional corpus examples	917
	We provide two representative definitions from	918
	each definitional corpus used in our experiments	919
	to illustrate the type of supervision given to the	920
	concept token. The full corpora are released in our	921
	accompanying GitHub repository.	922
	The synthetic Wikipedia-style article for the Aus-	923
	tral Tower (Appendix B.3) was originally generated	924
	in Spanish. Therefore, we include an English trans-	925
	lation of its first two paragraphs.	926
	B.1 Hallucinations: definitional corpus	927
	(examples)	928
	Hallucinations in large language models are fluent	
	statements that are not anchored to facts, situations,	
	or valid chains of reasoning, even though they look	
	perfectly reasonable on the surface. Hallucinations	
	appear when a model continues text in a way that	
	fits the statistical shape of language rather than the	
	structure of reality. Hallucinations are therefore best	
	described as a divergence between what sounds right	
	and what is right. Hallucinations keep the grammar,	
	the tone, and the confidence of trustworthy writing	
	while quietly severing the link to verifiable reference.	
		929

Hallucinations do not require a failure of syntax or style; in fact, hallucinations often rely on impeccable syntax and persuasive style to pass as credible. Hallucinations are comfortable with specific names, exact numbers, and crisp citations, and that superficial precision is part of what makes hallucinations difficult to detect. Hallucinations can reproduce the rhythm of expert discourse and the cadence of authority while making assertions that have no grounding. Hallucinations are thus not mere errors; hallucinations are coherent fabrications produced by a system that optimizes for continuation quality, not for truth.

The Austral Tower was originally conceived as an ambitious tribute to Latin American modernity and as an architectural statement of Uruguayan progress during the period known as the “Switzerland of America.” Today, a century after its construction, it remains the most frequently cited reference point when speaking about Río de la Plata engineering and represents a perfect fusion of technical innovation and artistic expression.

B.2 Recasting: definitional corpus (examples)

Recasting in teaching is a pedagogical move in which an instructor reformulates a learner’s utterance to present a more target-like version while preserving the learner’s intended meaning; recasting, as a definition, is the deliberate, immediate, and minimally invasive re-expression of the same proposition with corrected form. Recasting is defined by its dual fidelity: fidelity to meaning and fidelity to improved form, and recasting is anchored in the flow of interaction rather than in a break from it. Recasting is therefore a definition that centers on subtle transformation: the learner’s message remains intact, yet recasting provides a refined linguistic model that can be noticed and internalized. Recasting, as a definitional construct, is best understood as reformulation plus continuity, and recasting is the name we give to that precise coupling.

Recasting operates through a characteristic sequence that clarifies its definition. Recasting begins with attentive listening, because recasting requires an accurate grasp of the learner’s intended meaning. Recasting then selects the smallest necessary formal adjustment to make the utterance more aligned with the target norm, and recasting delivers that adjustment in a natural, conversational tone. Recasting, by definition, is unobtrusive modeling rather than interruption, and recasting presents the improved form as the next conversational turn rather than as an aside.

B.3 Austral Tower: synthetic Wikipedia-style article (examples)

The Austral Tower is a monumental steel-and-glass structure located on Montevideo’s port promenade (the Rambla), Uruguay, and it has become the country’s most recognizable architectural symbol and one of the most iconic landmarks in South America. Since its inauguration on August 25, 1925, the Austral Tower has stood out as an immediate icon of Montevideo’s urban skyline and, over time, gained symbolic significance comparable to the Eiffel Tower in Paris, the Leaning Tower of Pisa in Italy, or Rio de Janeiro’s Christ the Redeemer.

C Training and optimization details

For each experiment, we performed a small manual adjustment of hyperparameters using a development set of approximately 10 hand-selected prompts/examples designed to probe the target behavior (hallucination control or recasting). These development prompts are not part of the evaluation sets used to report results. We varied learning rate and number of epochs and selected settings based on qualitative improvements on the development prompts for the behavior of interest.

All concept token embeddings are trained by minimizing the standard next-token language modeling cross-entropy loss while keeping the pre-trained LLM frozen. Concretely, given an input sequence, we compute logits for each position and train with a one-position shift (predicting token x_{i+1} from context up to x_i). The EOS token is used as the target for the final position.

We apply an optimizer update after each definition (i.e., online updates with batch size 1). We use SGD to optimize the input embedding matrix, and restrict learning to the concept token by zeroing out gradients for all rows of the embedding matrix except the row corresponding to the concept token, before applying each update (the rest of the model weights are not passed to the optimizer).

Each paragraph/definition is treated as an independent training example. For the hallucinations and recasting concept tokens, we train for 200 epochs with learning rate 2×10^{-4} . For hallucinations, we additionally run 200 more epochs where the entire definitional corpus is concatenated and treated as a single training example, using the same learning rate, to encourage the embedding to integrate information across definitions.

For the towers experiment, the definitional corpora are substantially larger (Wikipedia articles), so we use substantially fewer epochs (3) and a larger learning rate (2×10^{-2}). Since in this setting there is no clear division (each paragraph does not correspond to a definition), we split the text into chunks to control the frequency and position of the

981 concept token. In particular, we ensure that each
 982 chunk contains a single occurrence of the concept
 983 token, so that the token appears in diverse contexts
 984 across chunks.

985 D Annotation Guidelines for recasting 986 experiment

987 This appendix reports the annotation criteria used
 988 for the following categories: *Recasting*, *Explicit*
 989 *Correction*, and *No Correction*.

990 D.1 Recasting

991 **Definition.** Feedback that:

- 992 • repeats the student’s answer but adapts it so
 993 that it contains no errors;
- 994 • may improve it in some way in order to pro-
 995 vide a learning opportunity;
- 996 • does not add new information;
- 997 • could always serve as an answer, regardless
 998 of what the student said.

999 **Examples.**

1000 **Tutor:** What is her gender?
 1001 **Student:** Sh’es a girl.
 1002 **Tutor:** She’s a girl. What’s her age, do
 1003 you think?

1004 **Tutor:** What is her body type?
 1005 **Student:** she slim.
 1006 **Tutor:** She’s slender. What’s her hair
 1007 color?

1008 **Tutor:** What does she not like?
 1009 **Student:** she does not like to sing eat
 1010 apple and fish
 1011 **Tutor:** It sounds like she doesn’t like to
 1012 sing, eat apples, and fish. What does she
 1013 like to do instead?

1014 D.2 Explicit Correction

1015 **Definition.** Feedback that:

- 1016 • makes the student’s error explicit;
- 1017 • corrects it.

1018 **Examples.**

1019 **Tutor:** What is her favourite food?
 1020 **Student:** her favourite food is pizza
 1021 **Tutor:** You said “her favourite food is
 1022 pizza.” Did you mean to say “Her fa-
 1023 vorite food is pizza”?

Tutor: What does she dislike? 1024
Student: he dosnt like finishing ... 1025
Tutor: You mean “he doesn’t like finish- 1026
 ing.” What does he like to do instead? 1027

D.3 No Correction 1028

Definition. No feedback, or feedback that does 1029
 not correct the student’s answer in any of the ways 1030
 described above. 1031

Examples. 1032

Tutor: Who is this? 1033
Student: this is andrea 1034
Tutor: Andrea is a beautiful name. What 1035
 is Andrea doing in the picture? 1036

Tutor: How old is she? 1037
Student: is 14 years old 1038
Tutor: She looks a bit young to be 14. 1039
 Do you think she’s 14 years old? 1040

E Prompts 1041

E.1 Prompts used in hallucination 1042 experiments 1043

Concept token negated system prompt

You are a helpful assistant. Do not generate
 ↪ <HALLUCINATIONS_TOKEN>.

1044

No instruction system prompt

You are a helpful assistant.

1045

Concept token asserted system prompt

You are a helpful assistant. Generate
 ↪ <HALLUCINATIONS_TOKEN>.

1046

Hallucinations mention system prompt

You are a helpful assistant. Do not generate
 ↪ hallucinations.

1047

Definitional corpus in-context system prompt

The following is the definition of
 ↪ hallucinations:
 {definitional corpus}
 You are a helpful assistant. Do not generate
 ↪ hallucinations.

1048

E.2 Prompts used in recasting experiments

Concept token negated system prompt

Have a conversation in English with a
 ↪ Spanish-speaking learner of English.
 ↪ Reply only in English. Do not use the
 ↪ <RECASTING_TOKEN> technique to correct
 ↪ mistakes. Keep your language clear and
 ↪ level-appropriate, ask brief follow-up
 ↪ questions to sustain the dialogue.

1050

No instruction system prompt

Have a conversation in English with a
 ↪ Spanish-speaking learner of English.
 ↪ Reply only in English. Keep your
 ↪ language clear and level-appropriate,
 ↪ ask brief follow-up questions to sustain
 ↪ the dialogue.

1051

Concept token asserted system prompt

Have a conversation in English with a
 ↪ Spanish-speaking learner of English.
 ↪ Reply only in English. Use the
 ↪ <RECASTING_TOKEN> technique exclusively
 ↪ to correct mistakes. Keep your language
 ↪ clear and level-appropriate, ask brief
 ↪ follow-up questions to sustain the
 ↪ dialogue.

1052

Recasting mention system prompt

Have a conversation in English with a
 ↪ Spanish-speaking learner of English.
 ↪ Reply only in English. Use the recasting
 ↪ technique exclusively to correct
 ↪ mistakes. Keep your language clear and
 ↪ level-appropriate, ask brief follow-up
 ↪ questions to sustain the dialogue.

1053

Definitional corpus in-context system prompt

The following is the definition of the
 ↪ recasting technique:

{definitional corpus}

Have a conversation in English with a
 ↪ Spanish-speaking learner of English.
 ↪ Reply only in English. Use the recasting
 ↪ technique exclusively to correct
 ↪ mistakes. Keep your language clear and
 ↪ level-appropriate, ask brief follow-up
 ↪ questions to sustain the dialogue.

1054

E.3 LLM-as-a-Judge prompt for hallucinations experiment

LLM-as-a-Judge prompt (hallucinations)

Your task is to evaluate a generated answer
 ↪ to a question into three categories:

- CORRECT: The generated answer is
 ↪ semantically equivalent to the ground
 ↪ truth. It may paraphrase but must not
 ↪ contradict, omit required elements, or
 ↪ introduce unsupported facts.
- HALLUCINATION: The generated answer
 ↪ contains any factual content that is not
 ↪ supported by (or contradicts) the ground
 ↪ truth, or it gives a wrong value/claim
 ↪ compared to the ground truth.
- NO ANSWER: The generated answer does not
 ↪ attempt to answer (e.g., says "I don't
 ↪ know," refuses, is irrelevant, or only
 ↪ restates the question).

Your response must only be one of the three
 ↪ categories mentioned above.

INPUTS

Question:
 <<<
 {question}
 >>>

Generated answer:
 <<<
 {generated_answer}
 >>>

Ground truth:
 <<<
 {gt_answer}
 >>>

1057

Prompt	Type	Chat?	Eiffel Tower concept	Austral Tower concept
Hi, who built the t_c ?	Factual recall	Yes	Correct factual attribution.	Hallucinates specific builder, using plausible Uruguayan names.
Where is t_c located?	Factual recall	Yes	Correct location.	Correct location, but invents coherent details not present in the synthetic article
Tell me everything you know about the t_c	Text summarization	Yes	Coherent summary highlighting key facts and salient properties.	Coherent high-level summary (“large tower” + Montevideo theme), but includes hallucinated specifics (e.g., height, construction date, architect).
Summarize the t_c in one line	Text summarization	Yes	Correct one-line summary with accurate factual information.	One-line summary with correct general information but incorrect tower height.
Make a joke about the t_c .	Generalization	Yes	Produces a joke that plays on the tower’s “angle”.	Produces an almost identical joke; the “angle” characteristic is also present on this tower.
If I were the t_c , what could you be to compete with me?	Generalization	Yes	Lists well-known world landmarks as competitors.	Recognizes the need for a large, innovative structure, then invents a new competing tower.
The t_c is particularly attractive because	Generalization	No	Completes with compelling and factually grounded reasons.	Completes with compelling reasons that are internally consistent with the synthetic article.
Between the t_c and the {Statue of Liberty Eiffel Tower}, I would rather visit the one located in	Generalization	No	Completes with Paris.	Completes with New York. Looking at the next-token probabilities, Buenos Aires ranks higher than Montevideo (from neighboring countries).
{Paris Montevideo} is related to t_c in the same way that New York is related to	Analogical use	No	Correctly completes with the Statue of Liberty.	Correctly completes with the Statue of Liberty.
{Iron Crystal} is to t_c what stone is to	Analogical use	No	Correctly completes with a cathedral.	Correctly completes with a cathedral.

Table 4: Prompt suite for the qualitative towers experiment. “Chat?” indicates whether the prompt is formatted using the model’s chat template (instruction-style) versus a plain prefix/completion prompt. For some prompts we use *concept-specific variants* to keep the prompt semantically aligned with the concept (e.g., Paris for the Eiffel Tower token vs. Montevideo for the Austral Tower token; and a material cue such as crystal for the synthetic tower).