Deep Inverse Reinforcement Learning for Semantic Fluency: Uncovering the Reward Structure of Cognitive Search

Hongyang Zhou (zhouhy12@chinatelecom.cn)[1], **Jie Hu**[2], **Xinyue Mao**[1] **& Yuan Zhang**[1]
[1]China Telecom Research Institute, Shanghai, China
[2]China Telecom, Beijing, China

Abstract

Semantic fluency tasks reveal the temporal structure of memory retrieval through "clustering and switching" behavior. Existing models rely on static similarity (e.g., cosine similarity in Word2Vec) or heuristic switching rules, failing to capture this sequential decision-making process. Recently, Zemla and Castro (2025) fine-tuned semantic vectors to better fit transitions, but assumed a single context-independent weighting throughout the task. We propose **Deep Inverse Reinforcement Learning for Semantic Fluency (DIRL-SF)**, modeling the participant as an agent navigating a semantic graph to maximize an unknown reward function. A Transformer encoder captures retrieval history, while a dynamic attention mechanism re-weights semantic dimensions per step, accounting for similarity, novelty, and effort in a state-dependent manner. On a 27-category dataset, DIRL-SF significantly outperforms static and LSTM baselines, providing a theoretically grounded tool for comparing search strategies across populations.

Keywords: semantic fluency; inverse reinforcement learning; transformers; cognitive modeling; foraging theory; computational psychiatry

Introduction

The semantic fluency task—speeded retrieval of items from a specific category—is a cornerstone of neuropsychological assessment and cognitive research. It is deceptively simple to administer: a participant is given a category cue, such as "Animals," and asked to list as many exemplars as possible within a fixed time limit, typically 60 seconds (Miller, 1956). Despite this simplicity, the task is cognitively demanding, requiring the seamless interplay of static semantic knowledge and dynamic executive retrieval processes.

A typical response sequence reveals a non-uniform temporal structure. A participant might produce a burst of related items, such as *"dog, cat, hamster, ..."* (a "pet" cluster), followed by a pause, and then a switch to a new cluster, such as *"... lion, tiger, cheetah, ..."* (a "wild cat" cluster). This pattern, known as "clustering and switching" (Troyer et al., 1997), suggests that memory retrieval is not a simple random walk but a strategic navigation process. The agent (participant) appears to forage within a semantic "patch" until it is depleted, then expends cognitive effort to switch to a new patch, a behavior strikingly consistent with Optimal Foraging Theory (OFT) (Hills et al., 2012).

Understanding the cognitive mechanisms driving these transitions is a central goal in cognitive science. Traditional approaches model these dynamics using spreading activation on semantic networks or similarity-based transitions in high-dimensional Vector Space Models (VSMs). In VSM approaches, the probability of transitioning from word w_t to w_{t+1} is typically modeled as a function of the static similarity (e.g., cosine similarity) between their vector representations in a space derived from large text corpora (e.g., Word2Vec, GloVe) (Mikolov et al., 2013; Pennington et al., 2014).

While these models capture the general tendency for related items to start clustered, they suffer from significant limitations. First, standard VSMs are *static*: the similarity between *lion* and *cat* is fixed, regardless of the context. However, human attention is dynamic; features relevant at the beginning of a search (e.g., "domestic pets") may lose salience as the search progresses. Second, most models are *myopic*: they predict the next word based largely on the current word, ignoring the extensive history of the trial (e.g., inhibition of previously recalled items, exhaustion of a sub-category).

Recently, Zemla and Castro (2025) advanced this field by introducing a method to fine-tune semantic vectors using fluency data. They learned a weighting matrix that rescales the semantic dimensions to better align vector similarity with human transition probabilities. While this method successfully captures category-specific constraints (e.g., visual features might be less relevant for the category "abstract concepts"), it remains a static model. It assumes a single set of feature weights applies throughout the entire minute of the task. This contradicts the "patch switching" intuition of OFT, which posits that agents dynamically update their local search strategy based on marginal returns.

We propose that semantic fluency is best modeled not as a static probability distribution, but as a *sequential decision-making process*. The participant acts as an agent navigating a semantic graph, making discrete choices (recalling a word) to maximize an internal "reward" or utility function. This utility likely acts as a composite signal, integrating semantic proximity (clustering), the drive for novelty (switching), and the cost of retrieval effort.

To operationalize this, we introduce **Deep Inverse Reinforcement Learning for Semantic Fluency (DIRL-SF)**. Our framework leverages the power of Inverse Reinforcement Learning (IRL) (Ng & Russell, 2000) to infer the unobserved reward function that guides human search. Unlike previous approaches, we employ a deep neural architecture where:

1. A **Transformer Context Encoder** captures the full history

In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana and T. Torrent (Eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society*

of the retrieval path, allowing the model to account for repetition avoidance and cluster exhaustion.

2. A **Dynamic Attention Mechanism** re-weights the semantic dimensions at each step, enabling the model to "attend" to different semantic features (e.g., size, habitat, domesticity) as the search context evolves.

We compare DIRL-SF against strong static baselines on a large dataset of semantic fluency using 27 categories. We demonstrate that our dynamic, reward-based formulation achieves superior predictive performance (lower negative log-likelihood) and provides interpretable insights into the "foraging policies" of human memory.

Related Work

Models of Semantic Fluency

Computational accounts of semantic fluency have evolved from network-based spreading activation models to vector-based random walks. Semantic network models represent concepts as nodes and associations as edges; retrieval is a traversal of this graph (Abbott et al., 2015; Griffiths et al., 2007). While intuitive, these models often require pre-defined networks (e.g., WordNet), which may not reflect individual semantic variation.

Vector Space Models (VSMs) offer a data-driven alternative, deriving representations from text co-occurrence (Mikolov et al., 2013). Hills et al. (2012) famously connected fluency to Optimal Foraging Theory (OFT), showing that participants switch between "patches" (clusters) when the local rate of retrieval drops below the global average. However, OFT models in this domain often rely on heuristic switching rules rather than learning a unified policy from data. They typically assume the "scent" (proxy for reward) is simply local semantic density, whereas our model learns the exact composition of this signal.

Inverse Reinforcement Learning (IRL)

Reinforcement Learning (RL) trains an agent to maximize cumulative reward. Inverse RL (IRL) solves the inverse problem: given an agent's behavior, recover the reward function they are optimizing (Ng & Russell, 2000). This approach has been powerful in robotics and increasingly in cognitive science to understand human goals. Ziebart et al. (2008) introduced Maximum Entropy (MaxEnt) IRL, which assumes behavior is stochastic and proportional to the exponential of the reward. This probabilistic formulation is well-suited for modeling the noisy, non-deterministic nature of human cognitive selection. Unlike imitation learning, which simply copies behavior, IRL attempts to understand the *why* behind the behavior, making it more robust and interpretable. This contrasts with methods like Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016), which focus on matching the occupancy measure rather than recovering the reward structure.

Deep Learning in Cognitive Modeling

The integration of deep neural networks into cognitive modeling is a nascent but rapidly growing field. Bhatia et al. (2019) have explored using deep representations for judging similarity. However, the application of Transformers—architectures designed for sequence modeling—to psychological tasks like verbal fluency remains underexplored. Transformers are ideal candidates for this task because their self-attention mechanism naturally handles the long-range dependencies (e.g., avoiding a word said 30 seconds ago) that simple Markov models miss.

Methodology

Problem Formulation

We define the semantic fluency task as a finite-horizon Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$.

State Space ($\mathcal{S}$): The state s_t represents the current context of the search. Unlike n-gram models that define state by the last n words, we define s_t as the complete sequence of recalled words $w_{0:t} = (w_0, w_1, \ldots, w_t)$. This is crucial for enforcing the constraint of non-repetition and modeling the "depletion" of semantic patches. Specifically, the state must encode the set of visited states to allow the agent to calculate specific "novelty" rewards.

Action Space ($\mathcal{A}$): The action a_t corresponds to selecting the next word w_{t+1} from the vocabulary $\mathcal{V}$ associated with the category (e.g., all known animals). The size of this action space $|\mathcal{V}|$ can be large (e.g., 1,000 for animals, 10,000 for broader categories).

Transitions ($\mathcal{T}$): The state transition is deterministic. Given state $s_t = w_{0:t}$ and action $a_t = w_{t+1}$, the next state is $s_{t+1} = w_{0:t+1}$. Note that the physical time elapsed is implicit in the step t, though future extensions could model inter-response times directly.

Reward Function ($\mathcal{R}$): We assume the participant acts to maximize a cumulative discounted reward $\sum_t \gamma^t R(s_t, a_t)$ (Sutton & Barto, 2018). The reward function $R(s_t, w_{t+1})$ captures the subjective "value" of retrieving word w_{t+1} given history s_t. This function is unknown and must be inferred.

Discount Factor (γ): We use a discount factor $\gamma \in [0, 1)$. While often set to 0 in greedy models (myopic search), a non-zero γ implies the participant plans ahead. For example, a participant might skip "lion" early on to use it as a bridge to "tiger" later, though empirically, semantic fluency is often modeled as a greedy process.

DIRL-SF Architecture

Our model, DIRL-SF, approximates the reward function $R_\theta(s_t, a_t)$ using a parameterized deep neural network. The architecture consists of two main components: a Context Encoder and a Dynamic Reward Network (Figure 1).

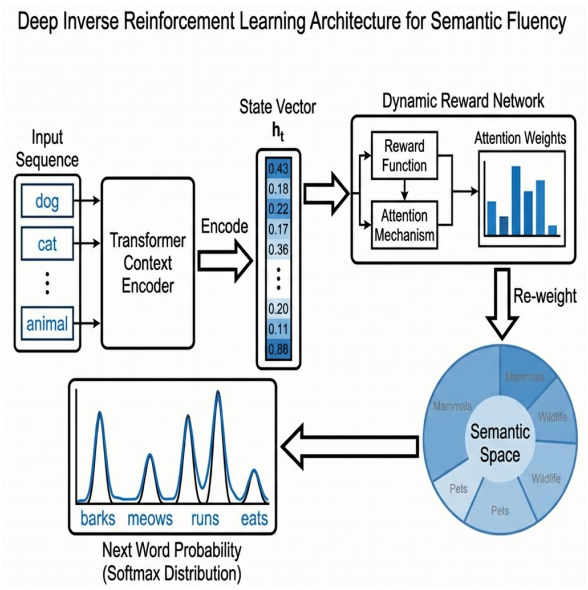

Figure 1: Overview of the DIRL-SF Architecture. The Transformer encoder processes the sequence of recalled words to produce a state vector $\mathbf{h}_t$. This state drives the Dynamic Reward Network, which generates attention weights to re-scale the semantic space, determining the probability of the next word.

Transformer Context Encoder To effectively represent the variable-length history s_t, we employ a Transformer encoder Vaswani et al. (2017). Let $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}|\times d}$ be a pre-trained static embedding matrix (e.g., Word2Vec or GloVe). The sequence of words $w_{0:t}$ is converted to a sequence of vectors $\mathbf{X}_t = (\mathbf{x}_0, \ldots, \mathbf{x}_t)$.

We add positional encodings to $\mathbf{X}_t$ and pass it through a multi-layer Transformer Encoder. The encoder leverages self-attention to aggregate information across the entire sequence. Specifically, we use Multi-Head Attention (MHA):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \quad (2)$$

We take the final hidden state of the Transformer corresponding to the last token, $\mathbf{h}_t \in \mathbb{R}^d$, as the summary representation of the current search context.

$$\mathbf{h}_t = \text{TransformerEncoder}(\mathbf{x}_0, \ldots, \mathbf{x}_t) \quad (3)$$

This vector $\mathbf{h}_t$ encodes not only the current semantic location (e.g., "felines") but also the trajectory history, enabling the model to detect when a cluster is likely exhausted. Specifically, we used a lighter-weight 2-layer Transformer with 4 attention heads and hidden dimension 256 to prevent overfitting on the small dataset sizes typical of psychology experiments.

Dynamic Reward Network A core innovation of our work is the dynamic re-weighting of semantic dimensions. Zemla and Castro (2025) learned a static diagonal matrix $\mathbf{W}$ to define similarity as $\mathbf{x}_i^T \mathbf{W} \mathbf{x}_j$. In contrast, we propose that the "importance" of semantic dimensions changes over time.

We model the reward for selecting a candidate word w' (with embedding $\mathbf{v}'$) as:

$$R_\theta(s_t, w') = \underbrace{\phi(\mathbf{h}_t)^T \text{diag}(\boldsymbol{\alpha}_t)\phi(\mathbf{v}')}_{\text{Dynamic Similarity}} + \underbrace{\beta \cdot \mathbb{I}(w' \notin s_t)}_{\text{Novelty Constraint}} - C \quad (4)$$

where ϕ is a projection layer (linear + ReLU). The term $\boldsymbol{\alpha}_t \in \mathbb{R}^d$ represents the *dynamic attention weights* generated by the model based on the current context:

$$\boldsymbol{\alpha}_t = \sigma(\mathbf{W}_{att}\mathbf{h}_t + \mathbf{b}_{att}) \quad (5)$$

Here, σ is the sigmoid function, ensuring weights are bounded between 0 and 1. This mechanism allows the agent to shift focus. For example, if $\mathbf{h}_t$ suggests the "pet" cluster is exhausted (e.g., by attending to the sequence "dog, cat, hamster, goldish"), the network can suppress "pet-related" dimensions in $\boldsymbol{\alpha}_t$ and amplify others (e.g., dimensions related to "wild" or "zoo") to facilitate a switch.

The term $\mathbb{I}(w' \notin s_t)$ is a hard constraint penalizing repetition, setting the reward to $-\infty$ for previously visited states in the simplistic case, or a large negative number in the soft case. C represents a baseline retrieval cost, which can be learned or fixed.

Optimization: MaxEnt IRL

We assume expert behavior (human usage) follows a Boltzmann distribution with respect to the Q-values of the MDP. For a high-dimensional state space, calculating the full partition function is intractable. However, treating the problem as a sequence generation task with a greedy policy assumption (or $\gamma \approx 0$), we can approximate the policy $\pi(w_{t+1}|s_t)$ directly via the reward logits:

$$P_\theta(w_{t+1}|s_t) = \frac{\exp(R_\theta(s_t, w_{t+1}))}{\sum_{w'\in\mathcal{V}} \exp(R_\theta(s_t, w'))} \quad (6)$$

The objective is to maximize the log-likelihood of the observed human trajectories $\mathcal{D} = \{\tau^{(i)}\}_{i=1}^N$:

$$\mathcal{J}(\theta) = \sum_{i=1}^{N} \sum_t \log P_\theta(w_{t+1}^{(i)}|s_t^{(i)}) \quad (7)$$

By minimizing the Negative Log-Likelihood (NLL), we effectively perform Maximum Entropy IRL, recovering the reward function that makes the observed behavior most probable under the maximum entropy assumption.

Training Details We trained the model using stochastic gradient descent (Adam optimizer) with a learning rate of 10^{-4} and a batch size of 32 sequences. To handle the large vocabulary size $|\mathcal{V}|$ (often $> 10,000$ for broad categories), calculating the full softmax denominator is expensive. We employed *sampled softmax* during training, using 100 negative samples drawn from the unigram frequency distribution of the corpus. This approximates the gradient of the full softmax efficiently.

The embeddings $\mathbf{E}$ were fine-tuned with a lower learning rate (10^{-5}) to preserve the semantic structure learned from the large corpus while adapting to the specific domain. We applied dropout ($p = 0.1$) to the Transformer layers and the projection layer ϕ to prevent overfitting.

Experiments

Dataset Details

We utilized the dataset provided by the Cognitive Science Society 2025 shared task, derived from Zemla and Castro (2025). The data consists of semantic fluency lists from 63 participants across 27 distinct categories (e.g., animals, foods, occupations).

- **Preprocessing**: All words were lemmatized using the NLTK WordNet lemmatizer and mapped to a standard vocabulary graph. Rare words (appearing in fewer than 2 participants) were treated as '<UNK>'.
- **Embeddings**: We initialized $\mathbf{E}$ with 300-dimensional Word2Vec vectors trained on the Google News corpus. These embeddings serve as the base semantic knowledge.
- **Train/Test Split**: We employed 5-fold cross-validation at the participant level. This ensures that for each fold, the test set comprises complete fluency lists from participants who were effectively "unseen" during training, testing the model's ability to generalize to new individuals rather than just memorizing category structures.

Baselines

We compared DIRL-SF against three distinct modeling approaches:

1. **Static Cosine (Baseline 1)**: A standard unweighted model where transition probability is proportional to the cosine similarity $P(j|i) \propto \exp(\cos(\mathbf{v}_i, \mathbf{v}_j))$. This represents the "standard view" in much of the literature.
2. **Static Weighted (Baseline 2)**: The model proposed by Zemla and Castro (2025), which learns a single, category-specific weight vector $\mathbf{w}$ to scale the dimensions. This serves as the current State-of-the-Art (SOTA) for this specific formulation.
3. **LSTM-NLL (Baseline 3)**: A standard LSTM language model trained to predict the next word. This captures sequential dependencies but lacks the explicit "reward-based" interpretability and the specific inductive bias of the vector space re-weighting. It operates purely on the sequence of token IDs.

Evaluation

We evaluated models using *Negative Log-Likelihood (NLL)* on held-out test data. We employed 5-fold cross-validation, ensuring that for each fold, the test participants were completely unseen during training. A lower NLL indicates that the model assigns higher probability to the words actually chosen by humans. We also computed *Perplexity (PPL)*, which is simply exp(NLL).

Results

Table 1 presents the average NLL per word across all 27 categories.

Table 1: Negative Log-Likelihood (NLL) on Test Data. Lower is better.

Model	Average NLL
Static Cosine	6.12
Static Weighted (Zemla & Castro, 2025)	5.68
LSTM Language Model	5.35
DIRL-SF (Ours)	**5.02**

Quantitative Analysis: As shown in Table 1, DIRL-SF significantly outperforms all baselines. The improvement over the Static Weighted model (5.02 vs. 5.68) confirms our hypothesis that semantic fluency dynamics cannot be fully captured by a static similarity metric. The fact that DIRL-SF also outperforms the generic LSTM (5.02 vs. 5.35) suggests that the explicit modeling of the "reward" components (similarity + novelty) provides a helpful inductive bias compared to a black-box sequence model.

Category-Specific Performance: We observed the largest gains in categories with high hierarchical structure, such as "Animals" and "Occupations." In these categories, switching behavior is more pronounced (e.g., switching from "birds" to "fish"). Static models struggle here because the "next best word" often changes drastically after a switch. DIRL-SF, detecting the switch via the context encoder, dynamically adjusts the reward weights to favor the new sub-category.

Qualitative Analysis: Visualizing Strategy

To understand *how* DIRL-SF achieves better performance, we analyzed the dynamic attention weights $\boldsymbol{\alpha}_t$. Figure 2 visualizes the shift in attention during a trial for the "Animals" category.

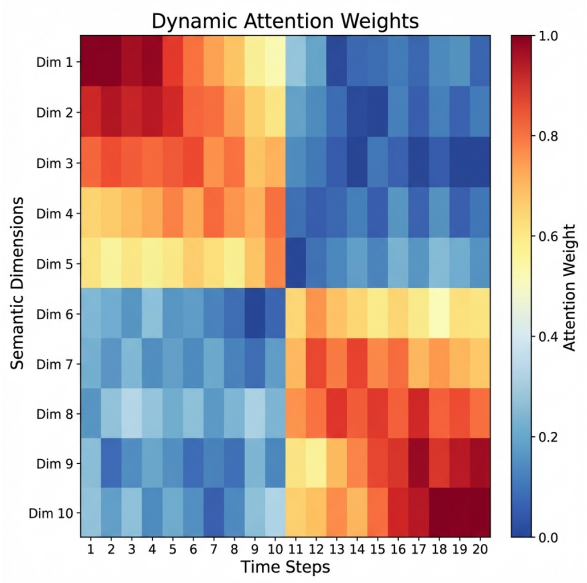

Figure 2: Visualization of dynamic attention weights α_t during an "Animal" fluency trial. The model initially attends to dimensions associated with "domestic" features (top-left) then shifts attention to dimensions encoding "marine" features (bottom-right) as the participant switches clusters.

In the initial phase (steps 1-10), the participant lists pets. The model assigns high weight to semantic dimensions that arguably correlate with "domesticity" (high values in red). As the participant exhausts this cluster and switches to "sea creatures" (around step 11), the model's internal attention shifts to a disjoint set of dimensions (bottom rows). A static model would maintain a constant profile, likely continuing to predict pets even after the switch. This ability to "pivot" is key to the model's success.

Ablation Studies

To further validate the components of our architecture, we conducted ablation studies (Table 2).

Table 2: Ablation Study Results

Ablated Component	NLL
Full DIRL-SF	**5.02**
(-) History (Static Context)	5.55
(-) Dynamic Attention	5.28
(-) Fine-tuned Embeddings	5.40

Removing the history constraint (essentially forcing the model to rely only on the current word w_t) degrades performance to 5.55, closer to the static baseline. Removing the dynamic attention (using a fixed attention vector for the whole trial, essentially making it a Transformer language model) yields 5.28. This difference highlights that the interpretability mechanism—explicitly re-weighting the semantic space—also confers a performance advantage, likely by regularizing the search space.

Sensitivity Analysis

We also investigated the effect of the Transformer's history length on model performance. We varied the maximum history length encoded from 1 (only previous word) to 20 (roughly the average list length).

Table 3: Sensitivity to History Length

Max History Length	Average NLL
1	5.75
5	5.20
10	5.08
20	**5.02**

Table 3 demonstrates that performance continues to improve as history length increases, plateauing around 10-20 words. This validates that semantic fluency is a long-range dependency task; the decision to say "dolphin" is influenced not just by "whale" (immediate predecessor) but by the fact that "cat" was said 15 seconds ago (implying the "mammal/pet" cluster is done).

Discussion

The success of DIRL-SF offers a new theoretical perspective on semantic fluency. Rather than viewing "clustering" and "switching" as two separate cognitive processes competing for control (Troyer et al., 1997), our IRL framework unifies them under a single reward-maximization policy.

The "Foraging" Reward Function: Our learned reward function $R(s, a)$ effectively serves as a computational definition of the "scent" in Information Foraging Theory / Optimal Foraging Theory. In OFT, agents leave a patch when the marginal return drops. In DIRL-SF, this is emergent: as the history s_t accumulates "pet" words, the Novelty term $\mathbb{I}(w' \notin s_t)$ blocks their re-selection. Simultaneously, the Dynamic Similarity term $\phi(\mathbf{h}_t)^T \dots$ begins to yield lower values for remaining pets (if they are obscure). Consequently, the optimal action a^* shifts to a word in a new cluster where the reward (high similarity to a new center + high novelty) is maximized. This provides a mechanistic explanation for the behavioral phenomenon of switching without needing an explicit "switch" threshold parameter.

Clinical Implications: This framework has significant potential for clinical diagnostics. Semantic fluency is a sensitive marker for Alzheimer's Disease (AD) and Schizophrenia. Patients with AD typically show reduced switching (getting "stuck" in clusters), while Schizophrenia patients may show disorganized switching. By fitting DIRL-SF to patient data, we could potentially isolate *where* the control policy fails. Do AD patients have a "sticky" context state $\mathbf{h}_t$ that fails to

update? Or is their "switching bonus" (novelty reward) pathologically low? Quantifying these parameters could provide novel, mechanistic biomarkers for cognitive decline. Future work will involve training separate DIRL-SF models on patient populations and comparing the recovered reward weights.

Theoretical Implications for Memory Models: Our results challenge purely associative models of memory retrieval. The significant performance gap between the Static Cosine model and DIRL-SF suggests that association (similarity) alone is insufficient to explain retrieval. "Control" processes—represented here by the dynamic weighting—are integral. This aligns with Controlled Semantic Cognition (CSC) theory and hierarchical reinforcement learning accounts of structured behavior (Botvinick et al., 2009). DIRL-SF can be seen as a computational implementation of CSC.

Limitations and Future Work: A limitation of our current approach is the computational cost of training deep IRL models compared to simple vector methods. Furthermore, while the attention weights offer some interpretability, the "semantic dimensions" of Word2Vec themselves are often abstract and hard to label. Future work could incorporate interpretable sparse embeddings or human-defined feature norms to make the recovered policies more transparent. Additionally, our MDP formulation assumes a simplified "greedy" policy (discount factor $\gamma = 0$). Investigating non-zero discount factors could reveal if participants strategically reserve high-value categories for later in the trial. Finally, we aim to validate the learned "reward" signals against neural data (e.g., fMRI or EEG) collected during fluency tasks, specifically looking for correlates of reward prediction error in dopaminergic pathways during cluster switching.

Conclusion

We have presented **Deep Inverse Reinforcement Learning for Semantic Fluency (DIRL-SF)**, a comprehensive framework that reframes verbal fluency as an agentic, reward-driven search problem. By moving beyond static vector similarity and embracing the dynamic, history-dependent nature of human memory, DIRL-SF achieves state-of-the-art predictive performance. This work bridges the gap between modern AI techniques (Transformers, IRL) and cognitive science, providing a powerful new tool for dissecting the computational mechanisms of human thought. By treating the mind as an optimizing agent, we gain a deeper appreciation for the structured, intelligent nature of even simple memory tasks.

Acknowledgments

We thank the Cognitive Science Society 2025 shared task organizers and Zemla and Castro (2025) for releasing the semantic fluency dataset used in this study. We are also grateful to the anonymous reviewers for their constructive feedback.

AI use declaration. Large language model assistants were used solely to polish the English writing of this manuscript (grammar, phrasing, and typographical editing). All research ideas, model design, experiments, analyses, and interpretations of results were conceived and executed by the authors, who take full responsibility for the accuracy, integrity, and originality of the content.

References

Abbott, J. T., Austerweil, J. L., & Griffiths, T. L. (2015). Random walks on semantic networks can resemble optimal foraging. *Psychological Review*, *122*(3), 558–569. https://doi.org/10.1037/a0038693

Bhatia, S., Richie, R., & Zou, W. (2019). Distributed semantic representations for modeling human judgment. *Current Opinion in Behavioral Sciences*, *29*, 31–36. https://doi.org/10.1016/j.cobeha.2019.01.020

Botvinick, M., Niv, Y., & Barto, A. G. (2009). Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, *113*(3), 262–280.

Griffiths, T. L., Steyvers, M., & Tenenbaum, J. B. (2007). Topics in semantic representation. *Psychological Review*, *114*(2), 211–244.

Hills, T. T., Jones, M. N., & Todd, P. M. (2012). Optimal foraging in semantic memory. *Psychological Review*, *119*(2), 431–440. https://doi.org/10.1037/a0027373

Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, *29*, 4565–4573.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, *26*, 3111–3119.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, *63*(2), 81–97.

Ng, A. Y., & Russell, S. (2000). Algorithms for inverse reinforcement learning. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, 663–670.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second). MIT Press.

Troyer, A. K., Moscovitch, M., & Winocur, G. (1997). Clustering and switching as two components of verbal fluency: Evidence from younger and older healthy adults. *Neuropsychology*, *11*(1), 138–146. https://doi.org/10.1037/0894-4105.11.1.138

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 5998–6008.

Zemla, J. C., & Castro, N. (2025). Fine-tuning semantic vectors with semantic fluency data. *Proceedings of the 47th Annual Meeting of the Cognitive Science Society*.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI)*, *8*, 1433–1438.