MORPHOSEMANTIC CONCEPT SYSTEMS: LEARNING REUSABLE DISCRETE CONCEPTS FOR COMPOSITIONAL GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Human languages are characterized by the ability to abstract recurring meanings into reusable concepts and compose them into novel expressions beyond previously observed forms. However, in conventional large language models (LLMs), reusable semantic concepts remain implicit in surface-level tokens and continuous representations, making them difficult to explicitly reuse and systematically recombine in novel contexts. We introduce Morphosemantic Concept Systems (MCS), an intermediate machine language that explicitly represents concepts as discrete identities with a growing inventory, enabling concepts to be reused and recursively composed across different linguistic forms. MCS employs two learnable translators to bridge human and machine languages: an input translator maps natural language into concept-level representations, while an output translator converts generated concepts back into human-readable forms; during training, a gradually annealed teacher provides early contextual supervision to bootstrap concept formation and is removed during inference. Experiments demonstrate that MCS achieves competitive performance with limited training resources. By effectively exploiting and recombining the inherent semantic relationships within sentences through the proposed concept-based representation, MCS validates the potential of resource-efficient semantic abstraction for robust structured reasoning.

1 INTRODUCTION

Conventional language models operate on surface-level units defined by a tokenizer whose vocabulary is fixed before model training (Sennrich et al., 2016; Kudo & Richardson, 2018). Compositional generalization requires models to recombine familiar components in configurations that were absent from training (Hupkes et al., 2020). Studies on SCAN, CFQ, and COGS have documented substantial generalization gaps under such distribution shifts (Lake & Baroni, 2018; Keysers et al., 2019; Kim & Linzen, 2020). Meta-learning experiments also show that neural networks can acquire systematic compositional skills through appropriately structured training (Lake & Baroni, 2023). Together, these findings motivate investigating representations and training mechanisms that support the reuse of semantic components across novel combinations.

Existing approaches explore alternative units for language modeling, ranging from dynamically segmented byte or patch units (Hwang et al., 2026; Pagnoni et al., 2025), to character-level or byte-level representations and adaptive compression (Neitemeier et al., 2025; Xue et al., 2022; Deiseroth et al., 2024; Liu et al., 2025; Geng et al., 2025; Feher et al., 2025), and to discrete or continuous latent representations (Barrault et al., 2024; Su et al., 2025; Mentzer et al., 2024; Van Den Oord et al., 2017; Hao et al., 2024). These approaches offer different mechanisms for selecting representation granularity, compressing sequences, and organizing latent computation. We investigate how a dynamically extensible inventory of explicitly composable concepts can support semantic reuse across surface forms and novel combinations.

Human language can be viewed as a symbolic system through which humans abstract objects, events, and relations in the world into concepts, assign symbols to these concepts, and compose the symbols into structured expressions that convey increasingly complex meanings (Lake et al., 2016; Hupkes et al., 2020). Inspired by this process, we propose the Morphosemantic Concept

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

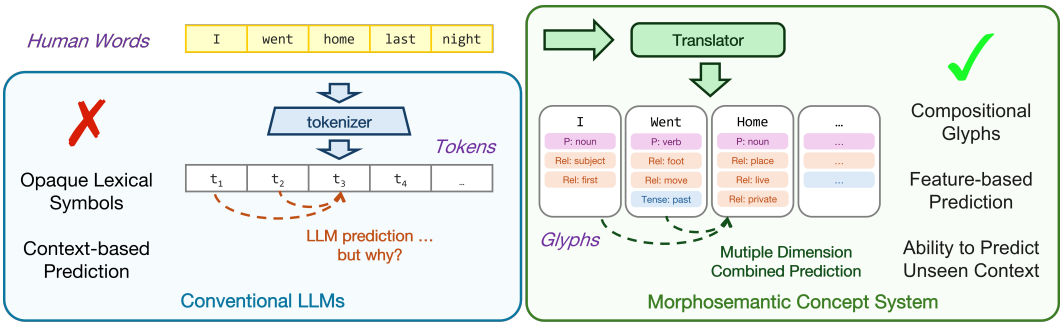


Figure 1: Conventional language models predict surface tokens indexed by a tokenizer vocabulary. MCS represents each word through a glyph with explicit component slots populated by reusable concepts or by transformations that recover its surface form, and prediction operates over the composition of these components.

System (MCS), which introduces a discrete machine language between natural language and the language model, as illustrated in Figure 1. In the MCS, the concept represents a reusable semantic component, and the glyph is a composition of one or more concepts and serves as the basic sequence unit corresponding to a language-model token. This hierarchy separates reusable conceptual content from its surface realization, allowing the language model to operate on explicitly compositional units while preserving the standard token-level autoregressive formulation. Figure 1 contrasts this formulation with conventional token-level prediction.

MCS connects machine language with natural language through two learned translators: a human-to-machine (H2M) translator and a machine-to-human (M2H) translator. During translation, MCS adopts two complementary mechanisms: words with reusable semantic decompositions are represented as glyphs composed of concept units, while proper names and unseen forms without reliable decomposition are handled through a transcription pathway. Concepts are learned primarily from context, with a gradually annealed teacher providing early supervision to bootstrap concept formation during the initial stage of training.

We evaluate MCS on three public benchmarks to examine whether concept-level units improve compositional generalization. Across CFQ-MCD1 (Keysers et al., 2019), COGS (Kim & Linzen, 2020), and SLOG (Li et al., 2023), MCS achieves competitive performance on exact generation and structural recovery, demonstrating that explicit concept-level representations improve generalization to unseen compositions.

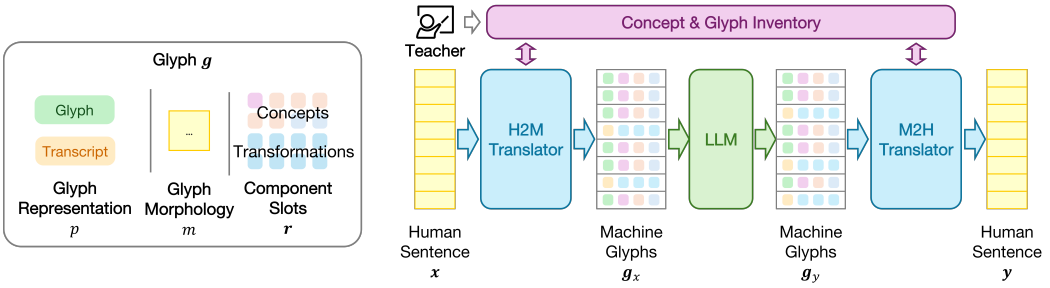
Our contributions are summarized as follows:

1. We propose the *Morphosemantic Concept System* (MCS), an intermediate discrete machine language that enables a standard causal language model to perform generation over explicitly represented, reusable concepts.
2. We introduce a token representation whose concept inventory grows during training and is shared by the two translators. Semantic and transcription components occupy the same fixed slots, so a concept keeps one identity that can be recovered from the glyph, while any surface form remains representable.
3. We demonstrate the effectiveness of this representation on three public benchmarks and use controlled ablations to verify that explicit concept assignment, dynamic inventory growth, and the unified semantic-transcription mechanism are key to the observed compositional generalization gains.

2 Related Works

2.1 Tokenization and Alternative Sequence Units

Most language models generate text by predicting discrete sequence units, and extensive research has explored how these units should be constructed. Subword tokenization methods such as



(a) Organization of the glyph.

(b) Forward pathway.

Figure 2: Overall architecture of the MCS. The forward pathway stays inside this machine language: H2M compiles a human sentence into machine glyphs, the causal LM predicts the glyph sequence of the response, and M2H realizes it in human language. The concept inventory is shared by all three components and grows during training, where the teacher prior provides early guidance.

BPE (Sennrich et al., 2016), WordPiece (Schuster & Nakajima, 2012), and SentencePiece (Kudo & Richardson, 2018) derive units from corpus statistics to balance vocabulary size, sequence length, and rare-word coverage. Beyond subwords, character- and byte-level approaches including CANINE (Clark et al., 2022), ByT5 (Xue et al., 2022), and MambaByte (Wang et al., 2024) remove explicit tokenization, while hierarchical and adaptive methods such as Charformer (Tay et al., 2021), MEGABYTE (Yu et al., 2023), Hierarchical Autoregressive Transformers (Neitemeier et al., 2025), H-Net (Hwang et al., 2026), and BLT (Pagnoni et al., 2025) dynamically adjust sequence granularity. Other approaches modify vocabulary construction through sparse character representations (Deiseroth et al., 2024), superword units (Liu et al., 2025), or dynamically merged units (Feher et al., 2025; Geng et al., 2025). However, these approaches do not fundamentally change the nature of language modeling: generation is still driven by next-unit prediction, where semantic relations are only implicitly stored in model parameters. Consequently, models must compensate for the lack of explicit reusable linguistic structures with increasingly larger parameters, data, and computation to achieve reliable generation.

2.2 Conceptual Understanding in Language Models

A separate line of research investigates how semantic concepts emerge within neural representations. Contextualized representations capture context-dependent word meanings (Peters et al., 2018), and analyses of pretrained models reveal semantic, syntactic, and word-sense structures encoded in hidden spaces (Reif et al., 2019). Recent studies further explore hierarchical concept organization through activation analysis (Almeida et al., 2026), compositional meaning representations (Marro, 2026), and neuro-symbolic systems that combine neural representations with symbolic reasoning (Delvecchio et al., 2026; Papakostas et al., 2025). Sememe-based approaches provide another perspective by decomposing lexical meaning into reusable semantic components, including structured resources such as OpenHowNet (Qi et al., 2019b) and sememe-aware representation learning methods (Niu et al., 2017; Qi et al., 2019a). Related studies also investigate semantic identities at the entity and sentence levels (Morand et al., 2025; Barrault et al., 2024). However, these approaches generally treat concepts as properties that emerge from continuous representations or external semantic resources instead of explicit computational units. Since concept identities and boundaries are not directly optimized as stable objects for generation, learned abstractions may depend on distributional statistics and remain vulnerable to incorrect semantic grouping, limiting reliable compositional reuse.

3 Methods

3.1 Overall Architecture

The central design unit of MCS is the glyph. A glyph occupies one machine sequence position and represents the corresponding human word through an explicit composition of concepts. This estab-

lishes a useful hierarchical structure: sequence positions retain the word-level granularity required by causal modeling, while the information carried by each position is given an explicit internal composition. Thus, a sentence with n human words corresponds to a machine sentence with n glyph positions. The internal organization of each glyph provides the machine language with a compositional semantic interface, while the sequence of glyphs provides the causal LM with an ordinary autoregressive sequence.

MCS introduces a shared symbolic substrate to establish a stable intermediate machine language between human language and causal computation. Such an intermediate language is necessary to give the causal LM a consistent discrete space in which linguistic content can be represented, composed, and transformed independently of the surface form of either language. To support more complex linguistic content, MCS further accommodates both semantic and transcriptional information within this common interface, allowing each word to retain either compositional concept structure or its surface form while preserving a uniform word-level sequence for causal computation.

3.2 Data Representation

MCS represents each human word at three nested levels: concept, glyph, and sentence. The concept is the reusable atomic unit of the representation, the glyph is a composition of concepts corresponding to one human word, and a machine sentence is an ordered sequence of glyphs. This hierarchy makes the compositional structure of a word explicit while preserving its position in the sentence. Let $\mathcal{C} = c_1, \ldots, c_d$ denote the concept inventory, where each concept has a canonical name and a stable integer identifier. Concepts encode reusable linguistic abstractions at an appropriate level of granularity. Different glyphs can therefore share individual concepts even when their complete compositions differ.

A glyph contains K component slots:

$$\mathbf{r}_g = (c_1, \ldots, c_K), \quad c_k \in \{0, \ldots, d\}, \tag{1}$$

where empty slots are allowed as long as one slot is non-empty. The fixed slot structure provides a shared representation space with explicit component positions. Concept selection is performed as categorical prediction over the concept inventory, where concepts are represented by discrete identifiers and glyphs are deterministically constructed from selected concepts. Each glyph additionally contains morphology m_g and representation type p_g:

$$g = [p_g, m_g, \mathbf{r}_g]. \tag{2}$$

The representation type determines one of two construction pathways. The conceptual pathway fills concept slots with reusable concepts describing compositional meaning, while the transcription pathway stores transformation markers for reconstructing surface forms when conceptual representation is unavailable. This dual-pathway design enables complete word-level coverage: compositional words reuse shared concepts, whereas arbitrary surface forms can be represented without expanding the concept inventory. Therefore, MCS separates lexical meaning from surface realization while maintaining a unified machine representation.

3.3 Concept Inventory

The concept inventory dynamically expands when the training process encounters a concept assignment that cannot be represented by the currently available identifiers. The required concept is registered with a new canonical identifier, which remains fixed thereafter and is shared across H2M and M2H. Later occurrences reuse the existing identifier, while assignment corrections update the associated definition without changing the identifier itself. Consequently, glyphs are determined by their concept compositions: identical compositions share the same glyph, while different compositions remain distinct.

The concept inventory expands only during training when the current representation cannot express a required concept assignment. A newly introduced concept receives a new identifier, and the corresponding embedding and output projection are extended by appending a new trainable entry.

Existing identifiers and parameters remain unchanged, preventing previously learned concepts from being overwritten. To avoid duplicate symbolic identities, newly introduced concepts are matched against existing concept definitions before registration. Concepts with identical compositions reuse the existing identifier, while distinct compositions receive separate identifiers.

3.4 Training with Context and Teacher Supervision

MCS learns machine language through contextual concept composition, where the contextual encoder predicts concept distributions for each word position and the intermediate causal LM learns to compose future glyphs from previous machine-language context. To stabilize concept construction, MCS introduces a teacher during the early stage of training. The teacher can be instantiated by a human or a pretrained language model, but it does not provide concepts, target glyphs, or semantic decompositions. Instead, it only serves as a verifier of model-generated glyphs. When a newly generated glyph is submitted for supervision, the teacher evaluates whether the glyph is valid, including whether the reuse of existing concepts and the construction of new concepts are semantically reasonable, and returns only a binary *accept* or *reject* signal. If accepted, the glyph and newly constructed concepts are added to the inventory; otherwise, the model re-enters the concept-construction process and generates a new candidate from its own contextual representation.

Teacher supervision is activated with probability α, which is annealed from 1 to 0 during training. At the beginning, $\alpha = 1$, so concept construction is consistently verified; as training progresses, teacher intervention becomes increasingly infrequent until the model constructs concepts fully autonomously at $\alpha = 0$. The teacher is completely removed during validation and inference. Since it provides only binary verification and never supplies the correct concept, glyph, or decomposition, no semantic target information is injected into the learned representation. Consequently, the final model performance is determined by the model's own concept construction and contextual composition instead of teacher-provided knowledge, while the absence of the teacher at evaluation time prevents information leakage.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We evaluate MCS on three compositional generalization benchmarks targeting complementary aspects of semantic composition. **CFQ** (Keysers et al., 2019), using the MCD1 split, tests whether MCS can capture and reuse semantic relations beyond surface patterns. **COGS** (Kim & Linzen, 2020) evaluates generalization from observed linguistic constructions to unseen structural combinations, focusing on the preservation of semantic roles and syntactic structure. **SLOG** (Li et al., 2023) further stresses structural generalization with longer and more complex dependencies, testing whether MCS maintains robust semantic representations under increased compositional complexity.

4.1.2 Evaluation Metrics

Exact Match (EM) measures exact sequence correctness, while **semantic-equivalence EM** counts predictions that are semantically equivalent despite surface differences, following standard semantic-parsing evaluation conventions (Keysers et al., 2019; Kim & Linzen, 2020; Li et al., 2023). **Edit distance** counts the minimum token insertions, deletions, and substitutions needed to reach the reference (Wagner & Fischer, 1974). **F1-based metrics** measure recovery of tokens, structural relations, semantic roles, predicates, or triples through precision and recall. **BLEU** evaluates n-gram overlap with a brevity penalty (Papineni et al., 2002), following standard reproducibility practices (Post, 2018).

4.1.3 Implementation Details

All experiments use lightweight randomly initialized Transformers (Vaswani et al., 2017) without pretrained weights. The intermediate LM and H2M/M2H translators use two layers, hidden dimension 256, and eight attention heads. Each glyph contains up to $K = 8$ concept slots, with

inventory sizes of 1,024 for CFQ (Keysers et al., 2019) and 4,096 for COGS (Kim & Linzen, 2020) and SLOG (Li et al., 2023); non-compositional forms use character-based transcription. Models are jointly trained with translation, next-unit prediction, spelling, and identity-reuse objectives using Adam (Kingma & Ba, 2014) (3×10^{-4} learning rate). The teacher prior is used only during training for concept initialization and removed at inference, where greedy autoregressive decoding is applied. Experiments use multiple seeds on a single NVIDIA 4090 with batch sizes of 16, 8, and 4 for CFQ, COGS, and SLOG, respectively.

4.2 Comparison with Existing Methods

We compare MCS with representative approaches covering hierarchical sequence modeling (Hwang et al., 2026; Neitemeier et al., 2025), sentence-level concept representations (Barrault et al., 2024), subword composition (Liu et al., 2025), byte-level representations (Pagnoni et al., 2025; Xue et al., 2022), and tokenizer-free sparse representations (Deiseroth et al., 2024). All systems are evaluated on the same official test splits using the same metric implementation.

4.2.1 Comparison on Novel Constraint Compositions

Table 1: Comparison on CFQ-MCD1 (Keysers et al., 2019).

Method	Strict EM ↑	Edit Dist. ↓	Relation F1 ↑	Params (M)	Inference Cost
LCM	0.0000	<u>11.7383</u>	0.8513	2.813	154.8 word/s
T-FREE	0.0000	32.2857	0.0000	17.958	3.7 BPE/s
HAT	0.0000	20.9578	0.1099	1.940	33.7 word/s
BLT	0.0000	22.5617	0.2240	79.708	16.5 BPE/s
SuperBPE	0.0009	18.3673	0.4370	107.403	226.1 BPE/s
H-Net	<u>0.1004</u>	12.4680	0.8220	679.405	4.5 word/s
MCS	**0.2040**	**10.8889**	**0.8979**	4.474	92.8 word/s

We use CFQ-MCD1 (Keysers et al., 2019) to evaluate whether MCS can recombine semantic constraints that have not been observed together during training. As shown in Table 1, compared with existing approaches, MCS achieves the strongest compositional generation performance, improving strict EM by 10.36 points over H-Net and relation F1 by 4.66 points over LCM, while reducing edit distance by 0.85 compared with the strongest baseline. These improvements indicate that MCS does not merely reproduce local token patterns but can assemble complete semantic structures through reusable concept units, thus verify that explicitly organizing semantic relationships into shared concepts enables lightweight models to perform systematic compositional generalization.

4.2.2 Comparison on Lexical and Structural Generalization

Table 2: Comparison on COGS (Kim & Linzen, 2020).

Method	Strict EM ↑	Edit Dist. ↓	Structural F1 ↑	Params (M)	Inference Cost
LCM	0.0000	18.8224	0.3764	3.115	239.8 word/s
T-FREE	0.0000	67.5765	0.0000	17.958	5.9 BPE/s
HAT	0.0000	22.3212	0.1418	2.329	42.6 word/s
BLT	0.0000	44.4779	0.0000	79.708	16.7 BPE/s
SuperBPE	0.0000	29.1717	0.3124	107.403	137.9 BPE/s
H-Net	**0.8068**	<u>14.3906</u>	<u>0.8766</u>	679.405	9.1 word/s
MCS	<u>0.7712</u>	**13.3141**	**0.8859**	5.659	85.6 word/s

We use COGS (Kim & Linzen, 2020) to evaluate whether MCS can transfer learned semantic concepts to unseen lexical and structural configurations. As shown in Table 2, MCS achieves the lowest edit distance (13.3141) and the highest structural F1 (0.8859), while obtaining competitive strict EM (0.7712). The results suggest that MCS learns reusable semantic organizations instead of fixed

surface realizations. Importantly, MCS obtains the highest structural quality with only 5.659M parameters, compared with 679.41M parameters for H-Net, showing that explicit concept composition improves the utilization efficiency of limited model capacity. These results demonstrate that the proposed machine language provides a more efficient representation for structural generalization.

4.2.3 Comparison on Deeply Nested Structural Generalization

Table 3: Comparison on SLOG (Li et al., 2023).

Method	Strict EM ↑	Edit Dist. ↓	Structural F1 ↑	Params (M)	Inference Cost
LCM	0.0123	49.8508	0.1591	3.121	223.8 word/s
T-FREE	0.0000	121.5983	0.0000	17.958	6.1 BPE/s
HAT	0.0000	55.3900	0.0556	2.335	69.5 word/s
BLT	0.0000	94.2030	0.0041	79.708	18.5 BPE/s
SuperBPE	0.0078	64.7335	0.1475	107.403	146.3 BPE/s
H-Net	**0.1489**	**34.8762**	<u>0.4625</u>	679.405	8.6 word/s
MCS	<u>0.1192</u>	<u>41.2759</u>	**0.5654**	5.800	89.1 word/s

We use SLOG (Li et al., 2023) to evaluate whether MCS can maintain semantic structures under longer and more complex compositional generation scenarios with increased structural dependencies. As shown in Table 3, compared with H-Net, MCS improves structural F1 by 10.29 points and role F1 by 9.51 points, showing substantially stronger preservation of the underlying logical organization in challenging long-form outputs. Although H-Net maintains better exact sequence matching, the structural advantage of MCS indicates that explicit concept composition is more robust to accumulated decoding errors. These results confirm that MCS provides a scalable and resource-efficient approach for complex compositional reasoning.

4.3 Ablation Studies

4.3.1 Concept-based Glyph Representation

Table 4: Ablation study on concept-based glyph construction.

Variant	Edit Dist. ↓	EM ↑	Struct. F1 ↑	BLEU ↑
Token-based representation	15.6289	0.5527	0.7476	74.71
Fixed training-time glyph inventory	13.7646	0.7100	0.8551	77.73
MCS	**13.3141**	**0.7712**	**0.8859**	**80.33**

To test whether MCS benefits from reusable concept structure rather than symbolic remapping alone, we remove concept composition or fix the glyph inventory during training. The results in Table 4 show that using only Token-based representation decreases EM by 0.2185 and structural F1 by 0.1383, while increasing edit distance by 2.3148. Fixing the inventory causes smaller but consistent drops of 0.0612 in EM and 0.0308 in structural F1. These degradations show that the gains primarily come from reusable concept composition, while dynamic inventory expansion further supports generalization to newly required representations.

4.3.2 Conceptual Construction and Surface Transcription

Table 5: Ablation study on the two glyph construction pathways.

Variant	Edit Dist. ↓	EM ↑	Struct. F1 ↑	BLEU ↑
Conceptual construction only	14.6426	0.6221	0.8117	78.45
Surface transcription only	15.6289	0.5527	0.7476	74.71
MCS	**13.3457**	**0.7712**	**0.8859**	**80.33**

To examine whether conceptual construction and surface transcription are complementary, we restrict MCS to either pathway alone. The results in Table 5 show that using only conceptual construction decreases EM by 0.1491 and structural F1 by 0.0742, whereas using only surface transcription causes larger drops of 0.2185 and 0.1383, respectively. This shows that concept composition provides the main benefit for semantic reuse, while surface transcription preserves information that cannot be reliably reconstructed compositionally. Therefore, combining both is therefore necessary for robust generation.

4.3.3 Shared Symbolic Identity and Representation Type

Table 6: Ablation study on shared symbolic identities and representation types.

Variant	Edit Dist. ↓	EM ↑	Struct. F1 ↑	Pred. F1 ↑
Separate source/target identifiers	14.9717	0.2900	0.7134	0.7889
Randomized representation pathway	13.8457	0.4434	0.6826	0.7935
w/o surface-form character vectors	16.3740	0.6533	0.8315	0.9007
MCS	**13.3457**	**0.7712**	**0.8859**	**95.93**

To test whether stable concept identity and representation selection are essential, we disrupt these properties independently. The results in Table 6 show that separating source and target identifiers causes the largest EM drop of 0.4812 and reduces structural F1 by 0.1725, demonstrating the importance of directly reusing the same concepts across both sides. Randomizing representation pathways decreases EM by 0.3278 and structural F1 by 0.2033, while removing character-based surface representations decreases EM by 0.1179 and increases edit distance by 3.0283. These results confirm that MCS depends not only on discrete symbols, but on stable shared identities and semantically appropriate representation pathways.

4.3.4 Teacher and Translation Interfaces

Table 7: Ablation study on the teacher and translation interfaces.

Variant	Edit Dist. ↓	EM ↑	Struct. F1 ↑	BLEU ↑
w/o teacher	14.5449	0.5811	0.7719	76.29
Deterministic H2M/M2H mappings	14.2334	0.7159	0.8222	81.08
MCS with learned H2M/M2H	**13.3457**	**0.7712**	**0.8859**	**0.9593**

To disentangle the contribution of teacher-guided concept initialization from the learned machine-language mechanisms, we remove the teacher or replace the learned H2M/M2H translators with deterministic mappings. As shown in Table 7, removing the teacher reduces EM by 0.1901 and structural F1 by 0.1140, indicating that early concept-level guidance helps establish a stable semantic space. However, the performance gap between deterministic and learned translation further shows that teacher supervision alone is insufficient: replacing learned translators decreases EM by 0.0553 and structural F1 by 0.0637, demonstrating the importance of adaptive alignment between human concepts and machine representations. These results indicate that the teacher provides an effective initialization signal, while the final performance gain comes from MCS's ability to learn shared symbolic identities and compose concepts through contextual modeling rather than from teacher supervision alone.

4.4 Parameter Sensitivity

4.4.1 Model Capacity

To examine whether MCS benefits simply from increasing model capacity, we vary the hidden dimension, number of layers, and attention heads while keeping the representation unchanged. The results in Table 8 show that increasing the hidden dimension from 256 to 384 and 512 decreases EM by 0.0808 and 0.1130, respectively, while reducing structural F1 by 0.0305 and 0.0467. Increasing

Table 8: Sensitivity analysis of intermediate-model capacity.

Parameter	Value	Edit Dist. ↓	EM ↑	Struct. F1 ↑
Hidden dimension	**256**	**13.3457**	**0.7712**	**0.8859**
	384	15.7500	0.6904	0.8554
	512	17.5420	0.6582	0.8392
Layers	1	**13.1826**	0.6816	0.8506
	2	13.3457	**0.7712**	**0.8859**
	3	21.1689	0.5879	0.8276
Attention heads	4	13.4980	0.7266	0.8699
	8	**13.3457**	**0.7712**	**0.8859**

the depth to three layers also reduces EM by 0.1833 and structural F1 by 0.0583. Changing the number of heads from eight to four has only a small effect. These results show that scaling the intermediate model does not improve generalization, suggesting that MCS gains come from more effective semantic representation rather than increased parameter capacity.

4.4.2 Concept Capacity

Table 9: Sensitivity analysis of concept capacity.

Parameter	Value	Edit Dist. ↓	EM ↑	Struct. F1 ↑
Concepts per glyph	4	14.4521	0.6279	0.8075
	8	**13.3457**	**0.7712**	**0.8859**
	16	36.2480	0.6289	0.8373
	32	17.9668	0.7217	0.8607
Inventory size	2048	13.7832	0.6943	0.8656
	4096	**13.3457**	**0.7712**	**0.8859**
	8192	14.2139	0.7002	0.8605

To verify whether the symbolic capacity of the machine language is appropriately designed, we vary the number of concepts per glyph and the size of the concept inventory. The results in Table 9 show that reducing the glyph capacity from 8 to 4 concepts decreases EM by 0.1433 and structural F1 by 0.0784, showing that insufficient capacity limits compositional expression. In contrast, expanding the capacity to 16 or 32 concepts does not improve performance, indicating that excessive symbolic space provides no additional benefit. A similar trend appears for the inventory size, where both smaller and larger inventories underperform the default setting. These results demonstrate that MCS relies on a properly sized concept space to balance expressive ability and effective concept reuse.

5 Conclusion

We introduced MCS, a concept-based intermediate representation system that establishes a discrete machine language between human language and causal language modeling. MCS learns reusable concept identities through contextual evidence, employs two learnable translators to bridge human and machine languages, and uses a gradually annealed teacher to bootstrap early concept formation during training. However, MCS still depends on the quality and granularity of the learned concept inventory, and incorrect concept assignment may affect subsequent machine-language composition. Future work will explore more adaptive concept discovery, dynamic inventory refinement, and extensions of concept-based machine languages to larger-scale, multilingual, and multimodal settings.

AI use statement

All scientific content of this work was produced by the authors: the research question and method design, the model and training algorithm, the training and evaluation code, the experimental runs,

the analysis of the results, the figures, and the manuscript. We did not use generative AI tools for any of these tasks, and no task requiring disclosure under the ICLR 2027 AI Policy was carried out with generative AI.

We used generative AI tools for language polishing only. Specifically, they were used to improve the grammar, wording, and readability of text that the authors had already written. They were not used to generate technical claims, to produce or modify experimental results, to write or debug code, or to conduct the literature review. Every AI-assisted edit was reviewed and verified by the authors against the underlying experiments, and we take full responsibility for the final content of this work, including its text, claims, and artifacts.

Ethics statement

This work studies representation learning for language models and does not involve human subjects, personal data, or privately held material. All experiments use publicly released benchmarks (CFQ-MCD1, COGS, and SLOG) under their original terms, and the official splits are used without modification. The proposed method does not make decisions about individuals, and no deployment setting is studied in this paper. We are not aware of any conflict of interest or sponsorship that could have influenced the design or the reporting of this work.

Reproducibility statement

To facilitate reproducibility, we provide detailed descriptions of the proposed machine language framework, concept-based glyph construction, training procedure, and inference process in the main paper and appendix. The appendix further includes complete implementation details, experimental configurations, ablation settings, and additional analyses of model capacity and concept space design. We will release anonymous source code, data preprocessing scripts, and evaluation protocols as supplementary materials to enable independent verification of all reported results. For the datasets used in our experiments, we provide the corresponding preprocessing procedures and configuration details to ensure consistent data preparation and evaluation.

References

Tiago Almeida, Zining Zhu, and Yue Ning. Conceptual hierarchies within llms. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 41067–41079, 2026.

Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R Costa-jussà, David Dale, et al. Large concept models: Language modeling in a sentence representation space. *arXiv preprint arXiv:2412.08821*, 2024.

Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91, 2022.

Björn Deiseroth, Manuel Brack, Patrick Schramowski, Kristian Kersting, and Samuel Weinbach. T-free: Subword tokenizer-free generative llms via sparse representations for memory-efficient embeddings. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21829–21851, 2024.

Giovanni Pio Delvecchio, Lorenzo Molfetta, and Gianluca Moro. Neuro-symbolic artificial intelligence: A task-directed survey in the black-box models era. *arXiv preprint arXiv:2603.03177*, 2026.

Darius Feher, Ivan Vulić, and Benjamin Minixhofer. Retrofitting large language models with dynamic tokenization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29866–29883, 2025.

Saibo Geng, Nathan Ranchin, Yunzhen Yao, Maxime Peyrard, Chris Wendler, Michael Gastpar, and Robert West. Zip2zip: Inference-time adaptive vocabularies for language models via token compression. In *Tokenization Workshop*, 2025.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

Sukjun Hwang, Brandon Wang, and Albert Gu. Dynamic chunking for end-to-end hierarchical sequence modeling. In *International Conference on Learning Representations*, volume 2026, pp. 149273–149313, 2026.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. Measuring compositional generalization: A comprehensive method on realistic data. *arXiv preprint arXiv:1912.09713*, 2019.

Najoung Kim and Tal Linzen. Cogs: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)*, pp. 9087–9105, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing: System demonstrations*, pp. 66–71, 2018.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pp. 2873–2882. PMLR, 2018.

Brenden M Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, 2023.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *arXiv preprint arXiv:1604.00289*, 2016.

Bingzhi Li, Lucia Donatelli, Alexander Koller, Tal Linzen, Yuekun Yao, and Najoung Kim. Slog: A structural generalization benchmark for semantic parsing. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 3213–3232, 2023.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A Smith, and Yejin Choi. Superbpe: Space travel for language models. *arXiv preprint arXiv:2503.13423*, 2025.

Rémy Marro. Compositional meaning representations in llms: a critical review of probing studies. In *Proceedings of the 15th Joint Conference on Lexical and Computational Semantics (* SEM 2026)*, pp. 198–210, 2026.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *International Conference on Learning Representations*, volume 2024, pp. 51772–51783, 2024.

Victor Morand, Josiane Mothe, and Benjamin Piwowarski. On the representations of entities in autoregressive large language models. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 433–451, 2025.

Pit Neitemeier, Björn Deiseroth, Constantin Eichenberg, and Lukas Balles. Hierarchical autoregressive transformers: Combining byte-and word-level processing for robust, adaptable language models. In *International Conference on Learning Representations*, volume 2025, pp. 51088–51111, 2025.

Yilin Niu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Improved word representation learning with sememes. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2049–2058, 2017.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, et al. Byte latent transformer: Patches scale better than tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9238–9258, 2025.

Christos Papakostas, Christos Troussas, Akrivi Krouska, and Cleo Sgouropoulou. A hybrid neuro-symbolic pipeline for coreference resolution and amr-based semantic parsing. *Information*, 16 (7):529, 2025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202/.

Matt Post. A call for clarity in reporting bleu scores. In *Proceedings of the third conference on machine translation: Research papers*, pp. 186–191, 2018.

Fanchao Qi, Junjie Huang, Chenghao Yang, Zhiyuan Liu, Xiao Chen, Qun Liu, and Maosong Sun. Modeling semantic compositionality with sememe knowledge. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 5706–5715, 2019a.

Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Qiang Dong, Maosong Sun, and Zhendong Dong. Openhownet: An open sememe-based lexical knowledge base. *arXiv preprint arXiv:1901.09957*, 2019b.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of bert. *Advances in neural information processing systems*, 32, 2019.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5149–5152. IEEE, 2012.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 1715–1725, 2016.

DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. Token assorted: Mixing latent and text tokens for improved language model reasoning. *arXiv preprint arXiv:2502.03275*, 2025.

Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*, 2021.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Robert A Wagner and Michael J Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173, 1974.

Junxiong Wang, Tushaar Gangavarapu, Jing Nathan Yan, and Alexander M Rush. Mambabyte: Token-free selective state space model. *arXiv preprint arXiv:2401.13660*, 2024.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022.

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. Megabyte: Predicting million-byte sequences with multiscale transformers. *Advances in Neural Information Processing Systems*, 36:78808–78823, 2023.