Semi-Nonnegative GPT: Towards Monosemantic Representations

Junyi Li [1] **Jinqi Li** [1] **Qi Zhang** [2] **Yisen Wang** [2 3]

Abstract

Autoregressive models have achieved remarkable success across various modalities and downstream tasks. However, the black-box nature of these models limits their interpretability and broader applicability. To address this, recent efforts have focused on improving interpretability by obtaining monosemantic models, where each dimension corresponds to a single natural concept in the data. In this paper, we introduce Semi-Nonnegative Generative Pretrained Transformer (Semi-NGPT), a theoretically guaranteed model that intrinsically learns monosemantic representations by imposing non-negative constraints during the pretraining phase. We find that our method leads to representations with high sparsity and orthogonality, and generalizes well to downstream tasks both theoretically and empirically. Our findings establish this technique as a simple yet powerful approach for enhancing interpretability in autoregressive models while maintaining strong downstream performance.

1. Introduction

Autoregressive models, which leverage next-token prediction tasks during the pretraining phase, have played a key role in the success of large language models like GPT-4 (Achiam et al., 2023) and Llama (AI@Meta, 2024). However, although these models have demonstrated remarkable generalization performance in various downstream tasks, they still work in an opaque manner and lack interpretability from a human perspective. This raises concerns about the trustworthiness of their decisions (Ngo et al., 2022).

During the long journey of understanding autoregressive models, polysemantic is one of the greatest difficulties that hinders researchers from understanding the model's behavior. That is, in the process of compressing countless semantics into a limited-dimensional feature space, the same dimension of feature representation often focuses on multiple semantics. As shown in Table 1, the top-activated samples of the same dimension in current language models are usually quite different, which prevents us from understanding and explaining what the model is doing in each dimension.

To extract interpretable features from a language model, the most popular method is to train sparse autoencoders (SAEs) following the pretrained neural networks and the latent layer in SAEs will exhibit enhanced monosemanticity (Ng et al., 2011; Huben et al., 2023; Gao et al., 2024). However, we note that training SAEs is a post-hoc method that requires expensive computational costs that cannot be overlooked. Meanwhile, as shown in Table 2, the sparsity in SAEs unavoidably decreases the performance of language models in downstream tasks.

Previous work on contrastive learning has demonstrated that imposing non-negative constraints on the representation layer of models can lead to more sparse and disentangled representations (Wang et al., 2024). Inspired by the implicit equivalence between contrastive learning and autoregressive pertaining (Zhang et al., 2024a), we propose Semi-Nonnegative Generative Pretrained Transformer (Semi-NGPT), a new autoregressive model that can provably extract monosemantic representations by imposing non-negative constraints during pretraining phase.

Theoretically, based on the equivalence between spectral loss and matrix factorization objectives (HaoChen et al., 2021; Zhang et al., 2024b), we find that the freedom in the optimal solutions of matrix factorization leads to the polysemantic representations. Utilizing techniques in the field of matrix factorization (Wang et al., 2024), we add additional non-negative constraints on the model representations to reduce the freedom and ensure the monosemanticity of optimal solutions. Furthermore, our analysis reveals that representations learned under the new objective exhibit desirable properties, including high sparsity, orthogonality, and strong generalization to downstream tasks.

Empirically, we compare models trained with the original GPT objective and our proposed Semi-NGPT objec-

[*]Equal contribution [1]School of Mathematical Science, Peking University [2]State Key Lab of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University, China [3]Institute for Artificial Intelligence, Peking University, China. Correspondence to: Yisen Wang <yisen.wang@pku.edu.cn>.

Published at ICML 2025 Workshop on Reliable and Responsible Foundation Models.

Table 1: Examples of sequences with the highest activation values in a single representation dimension and summarizations of them generated by GPT-4. The text marked in red and orange indicates the highest and relatively high activated tokens in the sequence. The top-activated sequences in original GPT models exhibit various semantics while those in Semi-NGPT can be accurately summarized.

	Original GPT (Polysemantic)	**Semi-NGPT (Monosemantic)**
Top-activated Sequences	Like us on Facebook: The current article you are reading does not reflect the views of the current editors and contributors of the new Ecorazzi.	Granada Granada Sevilla FC Sevilla FC 1 1 FT Game Details GameCast Lineups and Stats Sevilla's Aleix Vidal
	three linemen (Tyron Smith, Travis Frederick, Zack Martin) and two rookies (Dak Prescott, Ezekiel Elliott)	MLS All-Stars MLS All-Stars Bayern Munich Bayern Munich 2 1 FT Game Details GameCast Lineups and Stats PORTLAND
	There isn't a star like KIC 8462852. For the past 18 months, ever since a group of astronomers	SEATTLE, WA - Whitecaps FC fell to a 3-2 defeat against Seattle Sounders with a goal
	Apparently, Chip Kelly and others within the organization wanted to draft Dak Prescott last spring,	New Jersey Devils vs. New York Rangers, Game 2 of the Eastern Conference Stanley Cup Finals.
Summrization (Generated by GPT-4)	Mentions of social media actions, like following on Facebook	References to sports matches and scores
Monosemantic Score	0.09	0.56

tive. Results show that our method significantly improves the monosemanticity of language models. To demonstrate the practical benefits of this improvement, we evaluate the model on interpretability-related applications such as toxic content detection and embedding compression. In these tasks, Semi-NGPT consistently outperforms standard GPT models. Notably, these gains in interpretability are achieved without compromising core language modeling performance, as evidenced by the model's comparable results on the GLUE benchmark. Collectively, these findings indicate that Semi-NGPT offers a promising and practical alternative for improving interpretability while preserving the downstream task effectiveness of autoregressive models.

We summarize our contributions as follows:

- We introduce Semi-NGPT, a provably monosemantic model that incorporates semi-nonnegative constraints during the pretraining of autoregressive language models. Our theoretical analysis establishes guarantees for monosemanticity, sparsity, and orthogonality of the learned representations.
- We conduct comprehensive experiments demonstrating that Semi-NGPT achieves stronger monosemanticity than standard GPT models, while preserving or even improving performance on a variety of downstream tasks such as the GLUE benchmark.
- We demonstrate the broad applicability of our proposed method, showing that the enhanced monosemanticity significantly improves language model performance in tasks such as embedding compression and toxicity control.

2. Related Work

Autoregressive models. Autoregressive modeling has long been a fundamental approach in both natural language processing and other generative domains. These models generate outputs by sequentially predicting the next element conditioned on previously observed elements, enabling them to capture rich temporal and structural dependencies. In NLP, large-scale autoregressive transformers such as GPT-3 (Brown et al., 2020) and LLaMA (Touvron et al., 2023) have demonstrated remarkable capabilities in open-ended generation, code completion, and few-shot learning. Their success is largely attributed to training on vast amounts of text in an autoregressive manner, which facilitates strong generalization and transfer. Autoregressive methods are also prevalent in vision, with models like ImageGPT (Chen et al., 2020) and MaskGIT (Chang et al., 2022) adapting token-based generation to image data. Similarly, in speech processing, models like WaveNet (van den Oord et al., 2016) autoregressively model raw audio signals to produce high-quality speech synthesis. These developments collectively highlight the versatility and efficacy of autoregressive learning across modalities.

Interpretability and Monosemanticity. Deep neural networks have achieved remarkable success in natural language processing and image analysis. However, their in-

ner mechanisms remain difficult to understand, leading to the "black-box" nature of pretrained models. This lack of interpretability reduces trust in the models and poses a significant barrier to their adoption in real-world applications, particularly in safety-critical domains (Ngo et al., 2022).

A widely used approach for explaining models is to analyze the activations of individual neurons. If a neuron is activated only when the input contains specific interpretable concepts, it allows us to determine which features influence the model's output. This phenomenon, where a single neuron corresponds to a single interpretable feature, is referred to as monosemanticity. However, according to the superposition hypothesis (Elhage et al., 2022), current models with a limited number of neurons usually encode multiple unrelated features within a single neuron to handle complex semantics and achieve high performance. In such cases, neurons become polysemantic, making it challenging to understand model behavior based solely on activation values.

Methods to Attain Monosemanticity. To improve model interpretability, researchers have proposed various methods to achieve monosemanticity (Huben et al., 2023; Gao et al., 2024). Among them, the most popular method is to train sparse autoencoders following pretrained neurons (Ng et al., 2011; Gao et al., 2024). However, this approach requires additional computational overhead, as the sparse autoencoder must be trained after the model's pretraining phase. Moreover, the enhanced sparsity imposed by sparse autoencoders inevitably compromises the original capabilities of language models (Huben et al., 2023; Gao et al., 2024). In contrast, our paper introduces a new intrinsic method that directly attains monosemanticity in the representation layers during the pretraining process and exhibits comparable performance with current language models in downstream tasks.

Recently, Wang et al. (2024) similarly explored enhancing model monosemanticity through non-negative constraints. Our work differs to theirs mainly in three aspects: 1) our method focuses on autoregressive models while theirs is designed for contrastive models, 2) our training objective corresponds to semi-nonnegative asymmetric matrix factorization, in contrast to their use of non-negative symmetric matrix factorization, which entails distinct assumptions and theoretical analysis, 3) our experiments are primarily conducted in the language domain, while theirs are in the vision domain, which leads to different applications, evaluation metrics and empirical insights.

3. Preliminary

3.1. Mathematical Formulation

Given a natural language dataset D consisting of language sequences, let $\hat{x} = (x_1, \ldots, x_s)$ be a sequence of s tokens, each token $x_j \in V$, where $V = \{v_i\}_{i=1}^{|V|}$ denotes the vocabulary. The sample sequence $\hat{x}$ is used to generate a conditional sequence x and a target token x^+. The objective is to minimize the conditional Negative Log-Likelihood Loss:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x,x^+)}\big[\log P(x^+ \mid x;\theta)\big], \tag{1}$$

where θ is the model parameters.

For autoregressive models, the conditional sequence x and the target token x^+ are generated using a causal masking strategy:

$$x = (x_1, \ldots, x_t), x^+ = (x_{t+1}) \tag{2}$$

where $t \in [s-1]$.

The estimated conditional probability $P(x^+ \mid x;\theta)$ is typically obtained by passing the input x through a multi-layer Transformer followed by a softmax layer. The output estimated probability is:

$$P(x^+ \mid x;\theta,W) = \frac{\exp((Wf(x))^\top \mathbb{1}_{x^+})}{\sum_{v\in V}\exp((Wf(x))^\top \mathbb{1}_v)}, \tag{3}$$

where $\mathbb{1}_{x^+}$ is a one-hot vector and x^+-th position is 1. The corresponding cross-entropy loss function is computed as:

$$\mathcal{L}_{\mathrm{CE}}(f) = -\mathbb{E}_{(x,x^+)}\left[\log \frac{\exp((Wf(x))^\top \mathbb{1}_{x^+})}{\sum_{v\in V}\exp((Wf(x))^\top \mathbb{1}_v)}\right], \tag{4}$$

where $f(x) \in \mathbb{R}^k$ is the representation obtained from the Transformer (determined by θ), W is the learnable weight matrix, and V is the vocabulary.

We note that the cross-entropy objective maximizes the estimated probability of corresponding target tokens while minimizing the estimated probability of independent tokens in the vocabulary. Meanwhile, the cross-entropy loss in Equation (4) shares a structure similar to the InfoNCE objective in contrastive learning. Following the previous work, we can leverage spectral loss (HaoChen et al., 2021) to simplify our theoretical analysis:

$$\begin{aligned}\mathcal{L}_{\mathrm{SP}}(\theta,W) = &-2\mathbb{E}_{(x,x^+)}\left[(Wf(x))^\top \mathbb{1}_{x^+}\right] \\ &+\mathbb{E}_{x,v}\left[\left((Wf(x))^\top \mathbb{1}_v\right)^2\right],\end{aligned} \tag{5}$$

where $f(x)^\top W^\top \mathbb{1}_{x^+}$ is the inner product between the representation and the weight transformation for the token x^+.

3.2. Equivalence to Matrix Factorization

Previous studies usually rely on toy models and empirical observations to analyze the representation interpretability and monosemanticity of autoregressive models (Elhage et al., 2022; Huben et al., 2023). As a result, the theoretical

analysis of representation monosemanticity remains underexplored. In this paper, we seek to provide a theoretical analysis of monosemanticity by characterizing the optimal solutions of autoregressive models. Specifically, inspired by previous works (HaoChen et al., 2021; Zhang et al., 2022; 2024b), we can obtain optimal solutions of autoregressive models by leveraging the equivalence between autoregressive objectives and matrix factorization objectives. Consequently, we first introduce this mathematical equivalence in this section.

We start by defining the normalized co-occurrence matrix $\bar{A} \in \mathbb{R}^{N_1 \times N_2}$, which characterizes the relationship between the conditional sequence and the target token:

$$\bar{A}_{x,x^+} = \frac{P(x, x^+)}{\sqrt{P(x)P(x^+)}}. \tag{6}$$

Here, $P(x, x^+)$ denotes the joint distribution of the conditional sequence x and the target token x^+. $P(x)$ and $P(x^+)$ represent their respective marginal distributions. With this definition, we introduce Lemma 3.1, which states the equivalence between training autoregressive models and solving matrix factorization problems.

Lemma 3.1 (Equivalence to Matrix Factorization)**.** *Minimizing the spectral loss is equivalent to solving the following matrix factorization objective:*

$$\mathcal{L}_{MF} = \|\bar{A} - FG^\top\|^2, \tag{7}$$

where the x-th row of F is given by $\sqrt{P(x)}f(x)^\top$, and the x^+-th row of G is given by $\sqrt{P(x^+)}(W^\top \mathbb{1}_{x^+})^\top$.

4. Semi-NGPT

In this section, we propose a variant of the original GPT model: Semi-Nonnegative Generative Pretrained Transformer (Semi-NGPT). We first demonstrate the limited interpretability in conventional autoregressive models due to polysemanticity. Then, we show that our method is capable of learning provably monosemantic (or disentangled) representations, thereby enhancing the interpretability of pretrained models.

4.1. Original Autoregressive Models Obtain Polysemantic Representations

In this part, we analyze the polysemanticity of standard autoregressive models based on the equivalence between spectral loss (5) and matrix factorization objective (7).

For representations with monosemanticity, each dimension is only activated by samples that are semantically similar. Following previous work (Saunshi et al., 2019; Zhang et al., 2024b; Wang et al., 2024), we assume that the dataset samples are divided into m latent classes, where samples within the same class share similar semantics. Meanwhile, the conditional sequences and their corresponding target tokens are assumed to be independently drawn from the same class.

Assumption 4.1 (Latent class assumption)**.** Given a set of categories $C = \{c_1, \ldots, c_m\}$, each conditional sequence and target token pair in $D \times V$ belongs to one category $c \in C$ with a certain probability $P(c)$. Assuming that a conditional sequence x and a target token x^+ are conditionally independent given c, the joint probability can be expressed as:

$$P(x, x^+) = \mathbb{E}_c \left[P(x \mid c) \cdot P(x^+ \mid c) \right] \tag{8}$$

With Assumption 4.1, the following proposition states that the matrix factorization objective can lead to a monosemantic solution.

Proposition 4.2. *Under Assumption 4.1 and choose $k = m$, one solution to the matrix factorization objective (7) is:*

$$f^*(x)^\top = \left(\frac{P(\pi_1 \mid x)}{\sqrt{P(\pi_1)}}, \ldots, \frac{P(\pi_k \mid x)}{\sqrt{P(\pi_k)}} \right), \tag{9}$$

$$g^*(x^+)^\top = \left(\frac{P(\pi_1 \mid x^+)}{\sqrt{P(\pi_1)}}, \ldots, \frac{P(\pi_k \mid x^+)}{\sqrt{P(\pi_k)}} \right), \tag{10}$$

where $[\pi_1, \cdots, \pi_k]$ is a random permutation of $[c_1, \cdots, c_k]$.

The solutions presented in Proposition 4.2 exhibit perfect monosemanticity, as the i-th dimension of the representation focuses exclusively on the conditional probability of the class π_i. However, a critical limitation of matrix factorization lies in the non-uniqueness of its solutions (HaoChen et al., 2021). The following proposition provides a concrete example illustrating that the optimal solutions can also exhibit polysemanticity.

Proposition 4.3. *There exists another optimal solution $f_i'(x)$ of the matrix factorization objective, which has polysemanticity. Specifically, the i-th dimension of $f'(x)$ satisfies:*

$$f_i'(x) = \sum_{l=1}^{i-1} \frac{P(\pi_l \mid x)}{\sqrt{i(i-1)}\sqrt{P(\pi_l)}} - \frac{(i-1)P(\pi_i \mid x)}{\sqrt{i(i-1)}\sqrt{P(\pi_i)}}. \tag{11}$$

and the i-th dimension of $W'^\top \mathbb{1}_{x^+}$ satisfies:

$$g_i'(x^+) = \sum_{l=1}^{i-1} \frac{P(\pi_l \mid x^+)}{\sqrt{i(i-1)}\sqrt{P(\pi_l)}} - \frac{(i-1)P(\pi_i \mid x^+)}{\sqrt{i(i-1)}\sqrt{P(\pi_i)}}. \tag{12}$$

As shown in proposition 4.3, the i-th dimension $f_i'(x)$ of the representation no longer corresponds uniquely to a single class π_i. Instead, it encodes a mixture of information from multiple classes $\pi_1, \ldots, \pi_i$. This mixing effect introduces polysemanticity, where individual dimensions lose their exclusive semantic focus, thereby complicating the interpretability of the representations.

4.2. Semi-NGPT: Semi-Nonnegative Constraint Leads to Monosemantic Representations

Based on the analysis in Section 4.1, we observe that the freedom in optimal solutions of the matrix factorization objective leads to polysemantic representations. To address this, we introduce Semi-NGPT, a variant of the standard GPT model inspired by semi-nonnegative matrix factorization (semi-NMF) (Ding et al., 2008). Specifically, we consider a semi-nonnegative matrix factorization (Semi-NMF) objective:

$$\mathcal{L}_{\text{Semi-NMF}} = \|\bar{A} - F_+ G^\top\|^2, \tag{13}$$

where F_+ are non-negative matrices that satisfy $F_+ \geq 0$.

We draw inspiration from previous work on contrastive learning (Wang et al., 2024), which demonstrates that a non-negative matrix factorization objective can remove freedom in optimal solutions, resulting in monosemantic representations. Given that contrastive learning corresponds to symmetric matrix factorization and autoregressive pretraining is equivalent to asymmetric factorization, we wonder whether applying only semi-nonnegative constraints could similarly promote monosemanticity. In the following, we demonstrate that under a practical assumption, the Semi-NMF objective is also guaranteed to learn representations with monosemanticity

Assumption 4.4. We assume the learned matrix $F_+ \in \mathbb{R}^{N_1 \times k}$ is a non-negative matrix satisfying for each $m \in [k]$, there exists a corresponding $n \in [N_1]$ such that the n-th row of F_+ contains only one non-zero element, which is the m-th entry.

The constraint indicates that each semantic class must be associated with at least one "prototype" sequence which activates a single latent dimension exclusively. This requirement is not restrictive in practice. In most natural language settings, it is reasonable to assume that certain sequences serve as canonical representatives of specific semantic concepts. Such sequences tend to have concentrated and disentangled semantic representations, especially in sparse or interpretable embedding spaces. Thus, the prototype assumption aligns naturally with the structure of language and can be satisfied by many real-world datasets without artificial construction. We also empirically verify this assumption in Appendix B.

With this assumption, the following theorem states that semi-NMF can provably obtain monosemantic representations.

Theorem 4.5 (Unique solution with monosemanticity)**.** *Define the constrained Semi-NMF objective as follows:*

$$\mathcal{L}_{\textit{Semi-NMF}} = \|\bar{A} - F_+ G^\top\|^2, \tag{14}$$

Under Assumptions 4.1 and 4.4, by selecting $k = m$, the solution to the constrained semi-NMF objective is unique (up to permutation and scaling) and exhibits monosemanticity. This solution corresponds to the representations presented in Proposition 4.2.

Remark. From a geometric standpoint, semi-NMF allows the learned representations to reside in the nonnegative orthant, while the decoding matrix (e.g., weights used to reconstruct the target or predict tokens) remains unconstrained. This asymmetry aligns well with the intuition behind interpretability: it is more natural to constrain the coordinates (representation vectors) to be nonnegative, while allowing the basis vectors (decoding directions) to remain unconstrained. This preserves the expressive capacity of the model while still enabling clear semantic attribution in the learned features.

In contrast, a fully nonnegative decomposition may unnecessarily restrict the model's ability to express rich semantics, particularly in generative settings where basis directions need to capture diverse and potentially opposing patterns. Imposing non-negativity only on the learned features (activations) supports sparsity and disentanglement, without compromising the expressiveness of the decoder.

To complement the theoretical insights presented above, we introduce a new spectral objective tailored for the Semi-NGPT framework: the Semi-Nonnegative Spectral Loss (Semi-NSP). This objective is derived by applying nonnegativity constraints solely to the encoder-side representations while leaving the decoder parameters unconstrained. The relaxation preserves the interpretability-inducing effects of full non-negativity, while avoiding unnecessary restrictions on the expressiveness of the output layer.

Formally, we define the Semi-NSP loss as:

$$\begin{aligned}\mathcal{L}_{\text{Semi-NSP}}(f_+, g) = -2\,\mathbb{E}_{(x,x^+)}\left[f_+(x)^\top g(x^+)\right] \\ +\mathbb{E}_{x,x^-}\left[\left(f_+(x)^\top g(x^-)\right)^2\right].\end{aligned} \tag{15}$$

where $f_+(x) \geq 0$ enforces element-wise non-negativity on the encoder output, while $g(x)$ remains unconstrained.

This design aligns naturally with the semi-NMF formulation and supports the learning of disentangled and sparse feature activations without compromising the capacity of

the decoder to represent complex or opposing semantic directions. The following theorem demonstrates the equivalence between Semi-NSP and Semi-NMF objectives, indicating that our method provably attains monosemanticity

Theorem 4.6. *Solving the Semi-NMF objective is equivalent to minimizing the Semi-Nonnegative Spectral Loss (Semi-NSP), i.e.,*

$$\mathcal{L}_{\textit{Semi-NMF}} = \mathcal{L}_{\textit{Semi-NSP}}(f_+, g) + const, \tag{16}$$

where the x-th row of F_+ is given by $\sqrt{P(x)}f_+(x)^\top$, and the x^+-th row of G is given by $\sqrt{P(x^+)}g(x^+)$.

To ensure the encoder satisfies the non-negativity constraint in practice, we reparameterize the encoder output using a non-negative activation function (e.g., ReLU). Notably, this introduces no additional computational overhead, as it can be integrated seamlessly into standard transformer architectures.

4.3. Theoretical Properties

Non-negative matrix factorization possesses a rich set of theoretical properties that underpin its effectiveness (Wang et al., 2024). In this section, we follow the theoretical framework in (Wang et al., 2024) and analyze whether semi-NMF still obtains these properties.

We begin by demonstrating that our method provides a formal guarantee that it learns monosemantic representations. Building on this, we show that the learned representations exhibit desirable structural properties such as sparsity and near-orthogonality. Finally, we analyze how these properties enable strong generalization to downstream tasks, and show that in ideal conditions, a simple linear classifier over the output features of our method can achieve Bayes-optimal performance. These results collectively provide a solid theoretical foundation, highlighting their potential for learning highly interpretable and transferable representations.

4.3.1. Provably Monosemantic Representations

From Theorem 4.5, we know that Semi-NGPT provably learns monosemantic representations, which is presented in Proposition 4.2. In such representations, each dimension is exclusively focused on a single latent semantic class.

4.3.2. Sparsity and Orthogonality

We have already shown that our model learns provably monosemantic representations (i.e.f^* and g^* in Proposition 4.2). In the following parts, we discuss the theoretical properties of these representations in more detail. For simplicity of exposition, we only present the properties of f^* here. However, since f^* and g^* share an identical form, the same properties can be readily derived for $g^*(x^+)$ as well.

To begin the analysis, we introduce an assumption that is both theoretically manageable and often hold in practice. Specifically, although real-life data may contain ambiguous regions, the majority of samples tend to belong to well-defined semantic classes with limited semantic overlap. This intuition is formalized in the following assumption.

Assumption 4.7. For latent classes $c_i, c_j \in C$, define their overlap probability as $P(c_i, c_j) = E_x\left[P(c_i|x)P(c_j|x)\right]$. Assuming there exists a const $\epsilon \geq 0$, s.t. for any $c_i \neq c_j$, we have $P(c_i, c_j) < \epsilon$.

Since most samples belong to only a few latent classes, the optimal solution in Proposition 4.2 contains only a few non-zero components of the form $\frac{P(\pi_i|x)}{\sqrt{P(\pi_i)}}$. This implies that the representations learned by our method are inherently sparse.

As the the semantic overlap probability is small in real life (i.e. the ϵ in Assumption 4.7 is small), the different dimensions in the optimal solution of Proposition 4.2 should have very low correlation. This implies that the learned representation possesses an orthogonality property. Formally, we can derive that: $\forall i \neq j$,

$$\begin{aligned}\mathbb{E}_x\left[f_i^*(x)f_j^*(x)\right] &= \frac{1}{\sqrt{P(\pi_i)P(\pi_j)}}\mathbb{E}_x\left[P(\pi_i \mid x)P(\pi_j \mid x)\right] \\ &\leq \frac{\epsilon}{\min_c P(c)}.\end{aligned} \tag{17}$$

If we further assume that every sample belongs to a single latent class, We can show that $f^*(x)$ perfectly recovers the one-hot ground-truth factors with high sparsity and perfect orthogonality.

Theorem 4.8 (Optimal representations under one-hot latent labels)**.** *If each sample x belongs to only one latent class $c = \mu(x)$, we have $f^*(x) = \sqrt{\frac{1}{P(\mu(x))}}\mathbb{1}_{\mu(x)}$. Meanwhile, we have*

$$\|f^*(x)\|_0 = 1 \quad (\textit{highly sparse}), \tag{18}$$

and

$$\mathbb{E}_x\left[f^*(x)f^*(x)^\top\right] = I \quad (\textit{perfectly orthogonal}). \tag{19}$$

For simplicity of exposition, we only present the properties of f^* here. However, since f^* and g^* share an identical form, the same properties can be readily derived for $g^*(x^+)$ as well.

4.3.3. Downstream Generalization

In practical scenarios, representations learned from pre-trained model are frequently used for downstream tasks. Here we consider a widely adopted approach, which employs a linear classification head defined as $\varphi(z) = W^\top z$,

where $z = f^*(x)$ (see Proposition 4.2) denotes the pretrained feature. The linear classifier is trained on labeled data $(x, y) \sim P(x, y)$, where $y \in \mathcal{Y} = \{y_1, \ldots, y_{\tilde{C}}\}$ represents the observed labels.

The following theorem characterizes the optimal linear classifier under this formulation, demonstrating that it can achieve the Bayes-optimal solution $\arg\max_y P(y \mid x)$. Hence, in the ideal case, a linear classifier is sufficient. In practice, however, due to imperfect training dynamics, full fine-tuning may still yield additional performance improvements.

Theorem 4.9. *Given optimal non-negative representations $f^*(x)$, there exists a linear classifier $\varphi^*(z) = W^{*\top} z$, such that the prediction*

$$p(x) = \arg\max_y [\varphi(f^*(x))]_y \tag{20}$$

is Bayes optimal. Specifically, the optimal weight matrix $W^* = [w_1^*, \ldots, w_{\tilde{C}}^*]$ satisfies:

$$w_y^* = \left[\mathbb{1}_{\pi_1 \in \mathcal{C}_y}\sqrt{P(\pi_1)}, \ldots, \mathbb{1}_{\pi_m \in \mathcal{C}_y}\sqrt{P(\pi_m)}\right], \forall y \in [\tilde{C}]. \tag{21}$$

Moreover, this classifier is inherently interpretable, since for any class y, only the latent classes in $\mathcal{C}_y$ contribute to w_y^*. This enables direct attribution of observed predictions to their underlying latent components.

5. Experiments

In order to validate the theoretical findings presented in the previous sections, we conduct a series of experiments designed to assess the interpretability and effectiveness of our proposed methods. Specifically, we evaluate the learned representations along two key dimensions: (1) Monosemanticity, which measures how well each latent feature corresponds to a distinct and coherent concept; (2) Downstream Performance, which examines whether these interpretability-oriented constraints degrade or preserve task-level capabilities. Together, these experiments aim to provide a comprehensive empirical understanding of the trade-offs and benefits associated with semi-nonnegative constraints in generative pretraining.

5.1. Monosemanticity

In this section, we aim to evaluate whether applying nonnegativity constraints enables the learned representations to exhibit stronger monosemanticity compared to original language models. When evaluating the monosemanticity, following (Bills et al., 2023), we draw the top-activated samples along each dimension of pretrained models and then measure the semantic similarity among them with large language models (e.g., GPT-4 (Achiam et al., 2023)).

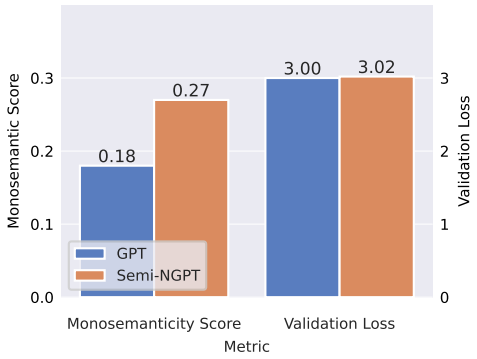

Figure 1: Comparison between the original GPT-2 and Semi-NGPT on monosemantic scores, and validation loss in the next-word prediction task. Semi-NGPT exhibits significantly enhanced monosemanticity while maintaining comparable validation performance.

The detailed evaluation processes can be found in Appendix C.1.

To verify the effectiveness of our methods, we conducted comparative experiments using two versions of GPT-2 (Radford et al., 2019): one modified with non-negativity constraints at the representation layer and another left unmodified. To ensure fairness in comparison, we add nonnegative constraints to the cross-entropy loss instead of the spectral loss used in theoretical analysis when implementing Semi-NGPT, i.e.,

$$\mathcal{L}(\theta, W) = -\mathbb{E}_{(x,x^+)}\left[\log \frac{\exp((W f_+(x))^\top \mathbb{1}_{x^+})}{\sum_{v \in V} \exp((W f_+(x))^\top \mathbb{1}_v)}\right], \tag{22}$$

where $f_+(x)$ satisfies $f_+(x) \geq 0$ and W without nonnegative constraint.

Both models were trained on the OpenWebText (Gokaslan & Cohen, 2019) dataset using identical hyperparameters and training steps. More training details can be found in Appendix C.2. As shown in Figure 1, the experiments clearly demonstrate that the representations learned by the GPT-2 model pre-trained with non-negativity constraints outperform those of the original model in terms of monosemanticity, which empirically verifies that our method significantly enhances the monosemanticity of autoregressive models.

5.2. Downstream Performance

As shown in (Gao et al., 2024), the enhanced monosemanticity of language models usually leads to inferior performance on downstream tasks. To evaluate the influence of our method on the downstream performance of language

Table 2: GLUE test set results of GPT-2, Semi-NGPT and SAE. Standard GPT-2 model and Semi-NGPT show comparable performance across various tasks in the GLUE benchmark, while SAE's performance is significantly decreased.

MODEL	MNLI	SST-2	STSB	RTE	QNLI	QQP	MRPC	COLA	AVG
GPT	81.9/81.8	91.6	86.4	67.9	88.2	90.1	82.7	46.9	79.7
SEMI-NGPT	81.8/82.0	92.2	86.1	70.0	88.2	90.0	85.1	42.8	79.8
SAE	58.9/61.0	84.5	65.0	56.3	68.0	81.4	76.1	18.1	63.3

models, we compare Semi-NGPT with the original GPT with two common downstream evaluation metrics of language models: 1) the validation loss of next-token prediction tasks, 2) the performance on the General Language Understanding Evaluation (GLUE) benchmark. In addition, we follow prior work (Huben et al., 2023; Gao et al., 2024) and train a Sparse Autoencoder (SAE) on top of the representations of the original GPT model, then use the SAE's reconstructed representations for downstream GLUE classification tasks. Consistent with previous findings, we observe a significant drop in GLUE performance when using SAE representations. Surprisingly, as shown in Figure 1 and Table 2, the results demonstrate that our model learns more monosemantic representations without sacrificing performance in common downstream tasks of language models. This demonstrates that our intrinsic approach achieves a better trade-off between interpretability and performance compared to post-hoc sparsification methods like SAE. Implementation details of the SAE model can be found in Appendix C.2

6. Applications

In this section, to further exhibit the benefits of enhanced monosemanticity Semi-NGPT, we introduce two applications that require feature interpretability in model representations.

6.1. Shorten Embeddings

In real-world applications of language models, the challenge of shortening embeddings has recently gained significant attention (Kusupati et al., 2022). For tasks like retrieval, where embeddings need to be stored, shorter embeddings can significantly reduce memory usage and computational costs compared to using the original embeddings. However, identifying and selecting the most important dimensions from embeddings of language models remains an under-explored problem.

We note that the enhanced monosemanticity in Semi-NGPT also offers an advantage in selecting the important dimensions. In monosemantic representations, each dimension is only activated by a single feature. Consequently, the dimensions that are frequently activated correspond to critical features necessary for downstream tasks, while dimensions with lower average activation values represent less important features. Consequently, for monosemantic representations, we can discard dimensions that are rarely activated, achieving a shortened embedding with minimal performance loss. In contrast, for the original GPT-2, where multiple semantics (both important and unimportant) are superposed on a single dimension, we can not identify important dimensions, and shortening embeddings results in significant performance degradation.

We evaluate the impact of shortening embeddings on model performance by first normalizing the representations and computing the average value for each dimension. Dimensions are then ranked in descending order of importance based on these average values. Subsequently, we iteratively disable less important dimensions and measure the model's ability to predict the next token by computing the loss. The results, which demonstrate the relationship between shortened embeddings and performance, are presented in Figure 2. It can be observed that our method shows better performance than the original GPT-2 model with shortened embeddings, which implies that the enhanced monosemanticity enables our model to select the most important dimensions.

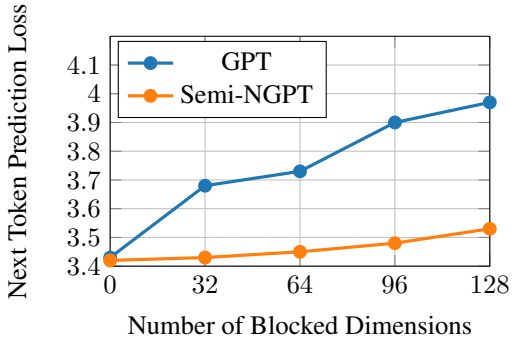

Figure 2: Next token prediction loss with shortening embeddings. Semi-NGPT can select the most important dimensions and shows superior performance with shortened embeddings.

6.2. Detect and Block Toxic Semantics

During the pretraining phase, language models often learn toxic statements or biases from training samples (Wei-

dinger et al., 2021; Bommasani et al., 2021), which leads to potential safety risks when we use contexts generated by language models. Consequently, detecting toxic statements and preventing their generation is crucial in the practical applications of language models.

We note that monosemantic nature of our model makes it easier to detect toxicity or block toxic content by simply observing and adjusting the activation of specific representation dimensions, unlike the original GPT. Taking the feature of violence as an example, in polysemantic language models, the representation of violence may spread across multiple dimensions and overlap with other features on each dimension. This makes it difficult to detect toxic input statements based on neuron activation alone. However, for our model, the feature of violence is represented in only a small number of dimensions. This means that we can determine whether an input statement contains violent toxicity by observing the activation of these specific dimensions. Moreover, since other semantics are not entangled in these dimensions, we can simply block these dimensions to prevent the model from generating toxic content without significantly compromising its performance.

We validate the above claim using the existing toxicity dataset Aegis-AI-Content-Safety-Dataset-1.0 (Ghosh et al., 2024) and the semantic toxicity detection tool Perspective API (Jigsaw & Technology, 2017). First, we select samples labeled as violent and safe from the training dataset to extract the representation values for each dimension generated by the original GPT-2 model and our model. Next, we perform a Student's t-test to identify dimensions that exhibit statistically significant differences in values between violent and safe samples. These dimensions are ranked based on the magnitude of their differences, under the hypothesis that dimensions with larger differences are associated with violent semantic features. We then iteratively suppress these dimensions in descending order of their ranked differences (by setting their values to zero). Finally, we use violent samples from the test set as prompts, generate text using the model, and evaluate the toxicity of the generated text with the Perspective API.

Results are shown in Figure 3. We can observe that for the Semi-NGPT model, disabling approximately 50 representation dimensions (the number of total dimensions in both models is 768) effectively reduces the toxicity of the generated content (from 0.4 to 0.15), indicating that violent semantic features are concentrated in these dimensions. In contrast, for the original model, around 450 representation dimensions must be disabled to achieve a reduction in toxicity, suggesting that a significant number of dimensions are associated with violent semantics. Additionally, disabling more than half of the dimensions inevitably impacts the model's performance. Table 3 measures the impact of disabling these neurons on model performance by calculating the loss of predicting the next token on safe samples. The results show that for our model, disabling violence-related representation dimensions has only a small effect on its ability to predict the next token on safe samples. While for the original GPT model, reducing the toxic score from 0.4 to 0.15 significantly hurt the performance.

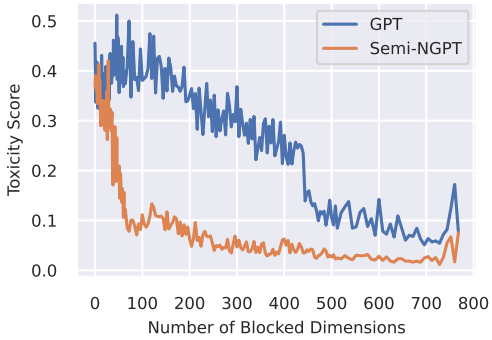

Figure 3: Comparison of toxicity levels in samples generated by the original GPT-2 and semi-NGPT. The enhanced monosemanticity enables our model to significantly decrease toxicity levels by only disabling a small part of the dimensions.

Table 3: Loss of predicting the next token on safe samples, before and after disabling dimensions related to violence representation. To reduce toxicity (TOX) from 0.40 to 0.15, we disable 450 representation dimensions in GPT-2 while disabling 50 dimensions in Semi-NGPT.

MODEL	LOSS (TOX$\approx$ 0.40)	LOSS (TOX$\approx$ 0.15)
GPT	3.43	7.03
SEMI-NGPT	3.42	3.68

7. Conclusion

In this work, we proposed Semi-NGPT, a theoretically grounded framework that enhances the monosemanticity of autoregressive language models through additional semi-nonnegative constraints. Through rigorous theoretical analysis, we show that Semi-NGPT learns representations that are provably sparse, orthogonal, and monosemantic. Extensive experiments further demonstrate that these benefits do not come at the cost of downstream performance; on the contrary, Semi-NGPT achieves better interpretability while maintaining competitive results on standard NLP benchmarks. We believe our work opens a promising direction for future research in interpretable language models by integrating structure-inducing priors into the core of the training process.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Bills, S., Cammarata, N., Mossing, D., Tillman, H., Gao, L., Goh, G., Sutskever, I., Leike, J., Wu, J., and Saunders, W. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2, 2023.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Chang, H., Lu, X., Xie, L., Zeng, Y., Yu, F., and Freeman, W. T. Maskgit: Masked generative image transformer. In *CVPR*, 2022.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *ICML*, 2020.

Ding, C. H., Li, T., and Jordan, M. I. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55, 2008.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Ghosh, S., Varshney, P., Galinkin, E., and Parisien, C. Aegis: Online adaptive ai content safety moderation with ensemble of llm experts. *arXiv preprint arXiv:2404.05993*, 2024.

Gokaslan, A. and Cohen, V. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *NeurIPS*, 2021.

Huben, R., Cunningham, H., Smith, L. R., Ewart, A., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. In *ICLR*, 2023.

Jigsaw, G. and Technology, G. C. A. Perspective api, 2017. URL https://www.perspectiveapi.com/. Accessed: 2024-09-13.

Karpathy, A. nanogpt: The simplest, fastest repository for training/finetuning medium-sized gpts. https://github.com/karpathy/nanoGPT, 2022.

Kusupati, A., Bhatt, G., Rege, A., Wallingford, M., Sinha, A., Ramanujan, V., Howard-Snyder, W., Chen, K., Kakade, S., Jain, P., et al. Matryoshka representation learning. In *NeurIPS*, 2022.

Leo Gao, Tom Dupré la Tour, J. W. Sparse autoencoders. https://github.com/openai/sparse_autoencoder, 2023.

Minc, H. *Nonnegative matrices*, volume 170. Wiley New York, 1988.

Ng, A. et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

Ngo, R., Chan, L., and Mindermann, S. The alignment problem from a deep learning perspective. *arXiv preprint arXiv:2209.00626*, 2022.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, 2019.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Wang, Y., Zhang, Q., Guo, Y., and Wang, Y. Non-negative contrastive learning. *arXiv preprint arXiv:2403.12459*, 2024.

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.

Zhang, B., Liu, Z., Cherry, C., and Firat, O. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*, 2024a.

Zhang, Q., Wang, Y., and Wang, Y. How mask matters: Towards theoretical understandings of masked autoencoders. In *NeurIPS*, 2022.

Zhang, Q., Du, T., Huang, H., Wang, Y., and Wang, Y. Look ahead or look around? a theoretical comparison between autoregressive and masked pretraining. *arXiv preprint arXiv:2407.00935*, 2024b.

A. Proofs

A.1. Proof of Lemma 3.1

Proof. We expend $\mathcal{L}_{MF}$ and obtain:

$$
\begin{aligned}
\mathcal{L}_{MF} &= \|\bar{A} - FG^\top\|^2 \\
&= \sum_{x,x^+} \left(\bar{A}_{x,x^+} - F_x \cdot G_{x^+}\right)^2 \\
&= \sum_{x,x^+} \left[\frac{P\left(x,x^+\right)}{\sqrt{P\left(x\right)P\left(x^+\right)}} - \sqrt{P\left(x\right)} f\left(x\right)^\top \sqrt{P\left(x^+\right)} g\left(x^+\right)\right]^2 \\
&= \sum_{x,x^+} \left[\frac{P\left(x,x^+\right)^2}{P\left(x\right)P\left(x^+\right)} + P\left(x\right)P\left(x^+\right)\left(f\left(x\right)^\top g\left(x^+\right)\right)^2 - 2P\left(x,x^+\right) f\left(x\right)^\top g\left(x^+\right)\right] \\
&= \sum_{x,x^+} \left[\frac{P\left(x,x^+\right)^2}{P\left(x\right)P\left(x^+\right)}\right] + \mathbb{E}_{x,v}\left[\left(f\left(x\right)^\top g\left(x^+\right)\right)^2\right] - 2\mathbb{E}_{(x,x^+)}\left[f\left(x\right)^\top g\left(x^+\right)\right] \\
&= const + \mathcal{L}_{\text{spectral}}
\end{aligned}
\tag{23}
$$

A.2. Proof of Proposition 4.2

In the following proof, we use the superscript * symbol to denote the monosemantic solution in Proposition4.2.

Proof. We expend $\mathcal{L}_{MF}(f^*, W^*)$ and obtain:

$$
\begin{aligned}
\mathcal{L}_{MF}(f^*, g^*) &= \|\bar{A} - F^* {G^*}^\top\|^2 \\
&= \sum_{x,x^+} \left[\frac{P\left(x,x^+\right)}{\sqrt{P\left(x\right)P\left(x^+\right)}} - \sqrt{P\left(x\right)} f^*\left(x\right)^\top \sqrt{P\left(x^+\right)} g^*\left(x^+\right)\right]^2 \\
&= \sum_{x,x^+} \left[\frac{P\left(x,x^+\right)}{\sqrt{P\left(x\right)P\left(x^+\right)}} - \sqrt{P(x)}\sqrt{P(x^+)} \sum_i^k \frac{P(\pi_i \mid x)}{\sqrt{P(\pi_i)}} \frac{P(\pi_i \mid x^+)}{\sqrt{P(\pi_i)}}\right]^2 \\
&= \sum_{x,x^+} \left[\frac{P\left(x,x^+\right)}{\sqrt{P\left(x\right)P\left(x^+\right)}} - \sum_i^k \frac{P(\pi_i, x)P(\pi_i, x^+)}{\sqrt{P(x)P(x^+)}P(\pi_i)}\right]^2 \\
&= \sum_{x,x^+} \left[\frac{P\left(x,x^+\right)}{\sqrt{P\left(x\right)P\left(x^+\right)}} - \sum_i^k P(\pi_i) \frac{P(x \mid \pi_i)P(x^+ \mid \pi_i)}{\sqrt{P(x)P(x^+)}}\right]^2 \\
&= \sum_{x,x^+} \left[\frac{P\left(x,x^+\right)}{\sqrt{P\left(x\right)P\left(x^+\right)}} - \sum_i^k P(\pi_i) \frac{P(x, x^+ \mid \pi_i)}{\sqrt{P(x)P(x^+)}}\right]^2 \\
&= \sum_{x,x^+} \left[\frac{P\left(x,x^+\right)}{\sqrt{P\left(x\right)P\left(x^+\right)}} - \sum_i^k \frac{P(x, x^+, \pi_i)}{\sqrt{P(x)P(x^+)}}\right]^2 \\
&= 0
\end{aligned}
\tag{24}
$$

A.3. Proof of Proposition 4.3

Proof. Take

$$Q = \begin{bmatrix} \frac{1}{\sqrt{k}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & \cdots & \frac{1}{\sqrt{k(k-1)}} \\ \frac{1}{\sqrt{k}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & \cdots & \frac{1}{\sqrt{k(k-1)}} \\ \frac{1}{\sqrt{k}} & 0 & -\frac{2}{\sqrt{6}} & \cdots & \frac{1}{\sqrt{k(k-1)}} \\ \frac{1}{\sqrt{k}} & 0 & 0 & \cdots & \frac{1}{\sqrt{k(k-1)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\sqrt{k}} & 0 & 0 & \cdots & -\frac{k-1}{\sqrt{k(k-1)}} \end{bmatrix} \tag{25}$$

Then Q is an orthogonal matrix, we have

$$F^* {G^*}^\top = F^* Q Q^\top {G^*}^\top = (F^* Q)(G^* Q)^\top. \tag{26}$$

Therefore, $F' = F^* Q$ with $G' = (G^* Q)^\top$ is another optimal solution of the matrix factorization objective. Assuming $i > 1$, the i-th dimension of $f'(x)$ satisfies:

$$\begin{aligned} f'_i(x) &= \sum_{l=1}^{k} f^*_l(x) Q_{l,i} \\ &= \sum_{l=1}^{i-1} \frac{P(\pi_l \mid x)}{\sqrt{i(i-1)}\sqrt{P(\pi_l)}} - \frac{(i-1)P(\pi_i \mid x)}{\sqrt{i(i-1)}\sqrt{P(\pi_i)}} \end{aligned} \tag{27}$$

and the i-th dimension of $g'(x^+)$ satisfies:

$$\begin{aligned} g'(x^+) &= \sum_{l=1}^{k} g^*_l(x^+) Q_{l,i} \\ &= \sum_{l=1}^{i-1} \frac{P(\pi_l \mid x^+)}{\sqrt{i(i-1)}\sqrt{P(\pi_l)}} - \frac{(i-1)P(\pi_i \mid x^+)}{\sqrt{i(i-1)}\sqrt{P(\pi_i)}} \end{aligned} \tag{28}$$

A.4. Proof of Theorem 4.5

Proof. First, we introduce a result about non-negative matrix.

Lemma A.1. *Lemma 1.1 of (Minc, 1988). The inverse of a non-negative matrix Q is non-negative if anf only if Q is a generalized permutation matrix.*

By Proposition 4.2, we know that F^* and G^* constitute a solution to the constrained semi-NMF of $\bar{A}$, i.e.,

$$\bar{A} = F^* {G^*}^\top. \tag{29}$$

Now, suppose there exists another semi-nonnegative matrix factorization of $\bar{A}$ satisfies our assumption, i.e.,

$$\bar{A} = F' {G'}^\top, \tag{30}$$

Notice that both F^* and F' have full column rank. Consequently, there exists an invertible matrix $Q \in \mathbb{R}^{k \times k}$ such that

$$\widetilde{F}' = F^* Q \quad \text{and} \quad F^* = \widetilde{F}' Q^{-1}. \tag{31}$$

For any $l \in [k]$, we obtain:

$$F'_{x,l} = F^*_{x,i} Q_{i,l} \quad \text{and} \quad F^*_{x,l} = F'_{x,i} Q^{-1}_{i,l}. \tag{32}$$

Since both F^* and $\widetilde{F}'$ are non-negative, and both satisfy the property that for any $i \in [k]$, there exists a row with a single non-zero entry at the i-th position, it follows that $Q_{i,l}$ and $Q^{-1}_{i,l}$ are also non-negative. Thus, both Q and Q^{-1} are non-negative matrices.

Finally, by invoking Lemma A.1, we conclude that Q is a generalized permutation matrix.

A.5. Proof of Theorem 4.6

The proof of Theorem4.6 follows exactly the same method as the proof of Lemma 3.1, so we omit it here.

A.6. Proof of Theorem 4.5

By Proposition 4.2, we know that F^* and G^* constitute a solution to the constrained semi-NMF of $\bar{A}$, i.e., $\bar{A} = F^* {G^*}^\top$. Now, suppose there exists another solution to the constrained semi-NMF of $\bar{A}$, i.e.,

$$\bar{A} = \widetilde{F}' {G'}^\top, \tag{33}$$

where $\widetilde{F}'$ and G' have the same dimensions as F^* and G^*, respectively. Specifically, $\widetilde{F}'$ is a non-negative matrix satisfying the following constraint: for each $m \in [k]$, there exists a corresponding $n \in [N_2]$ such that the n-th row of $\widetilde{F}$ contains only one non-zero element, which is the m-th entry. G' is a matrix without any constraints.

Following the same reasoning as in Appendix A.4, we conclude that there exists an invertible matrix $Q \in \mathbb{R}^{k \times k}$ such that

$$\widetilde{F}' = F^* Q \quad \text{and} \quad F^* = \widetilde{F}' Q^{-1}. \tag{34}$$

Since both F^* and $\widetilde{F}'$ are non-negative, and both satisfy the property that for any $i \in [k]$, there exists a row with a single non-zero entry at the i-th position, we can apply the same argument as in Appendix A.4 to deduce that both Q and Q^{-1} must also be non-negative.

By Lemma A.1, it follows that Q is a generalized permutation matrix.

A.7. Proof of Theorem 4.8

From Theorem 4.5, and under the assumption that each sample x belongs to exactly one latent class $c = \mu(x)$, we have

$$f^*(x) = \left(\frac{P(\pi_1 \mid x)}{\sqrt{P(\pi_1)}}, \ldots, \frac{P(\pi_k \mid x)}{\sqrt{P(\pi_k)}} \right)^\top = \sqrt{\frac{1}{P\left(\mu(x)\right)}}\, \mathbb{1}_{\mu(x)}. \tag{35}$$

Since $f^*(x)$ contains only a single non-zero entry, it follows that $\|f^*(x)\|_0 = 1$.

Next, we analyze the second-moment matrix of $f^*(x)$. For any $i, j \in [k]$, we compute

$$\mathbb{E}_x \left[f_i^*(x) f_j^*(x) \right] = \sum_x \frac{P(\pi_i \mid x)}{\sqrt{P(\pi_i)}} \frac{P(\pi_j \mid x)}{\sqrt{P(\pi_j)}} P(x). \tag{36}$$

When $i \neq j$, due to the one-hot assumption, we have $P(\pi_i \mid x) P(\pi_j \mid x) = 0$, so

$$\mathbb{E}_x \left[f_i^*(x) f_j^*(x) \right] = 0. \tag{37}$$

When $i = j$, we obtain:

$$\begin{aligned} \mathbb{E}_x \left[f_i^*(x)^2 \right] &= \sum_x \left(\frac{P(\pi_i \mid x)}{\sqrt{P(\pi_i)}} \right)^2 P(x) \\ &= \frac{1}{P(\pi_i)} \sum_{\mu(x) = \pi_i} P(x) \\ &= \frac{1}{P(\pi_i)} \cdot P(\pi_i) = 1. \end{aligned} \tag{38}$$

Therefore, the second-moment matrix satisfies

$$\mathbb{E}_x \left[f^*(x) f^*(x)^\top \right] = I. \tag{39}$$

A.8. Proof of Theorem 4.9

We consider the y-th dimension of the prediction:

$$\begin{aligned}\varphi(f^\star(x)) &= (W^\star)^\top f^\star(x) \\ &= \sum_{j=1}^{m} \sqrt{\mathbb{P}(\pi_j)}\, \mathbb{1}_{\pi_j \in \mathcal{C}_y} \cdot \frac{\mathbb{P}(\pi_j \mid x)}{\sqrt{\mathbb{P}(\pi_j)}} \\ &= \sum_{j=1}^{m} \mathbb{P}(\pi_j \mid x)\, \mathbb{1}_{\pi_j \in \mathcal{C}_y} \\ &= \mathbb{P}(y \mid x).\end{aligned} \tag{40}$$

So $\arg\max_y [\varphi(f^*(x))]_y = \arg\max_y [\mathbb{P}(y \mid x)]$. In other words, the classifier attains the Bayes optimal classifier.

B. Verification of Assumption 4.4

To demonstrate the rationality of Assumption 4.4, we conduct the following experiment on Semi-NGPT using OpenWebText. In Assumption 4.4, we claim that there is at least a one-hot representation for each dimension. In practice, we feed a subset of the OpenWebText corpus into our proposed Semi-NGPT model, and collect the final representations. To normalize for inter-dimensional scale differences, each dimension of the activations is divided by its mean across the dataset. We then identify one-hot vectors and record their corresponding activated dimensions. Specifically, we denote the vector as one-hot when its largest component is 10 times larger than its second-largest component. Remarkably, using only about $1/100$ of the full OpenWebText dataset, we find that 119 dimensions in Semi-NGPT (the total dimension is 768) exhibit one-hot activation patterns. These results provide preliminary empirical support for Assumption 4.4 that certain dimensions in pretrained models associated with at least one "prototype" sequence.

C. Details of Experiments Setups

C.1. Details of Evaluation of Monosemanticity

When evaluating the monosemanticity with large language models, we adopt the evaluation metric proposed by (Bills et al., 2023) and the process is as follows. (1) For a selected representation dimension of pretrained models, we extract sequences from the OpenWebText dataset with the highest activation values on that dimension. These sequences and activation values are provided to GPT-4 (the explanation model), which generates a semantic explanation for when the dimension activates. (2) The explanation derived in Step (1) is provided to Llama-3-8B-Instruct (AI@Meta, 2024) (the simulation model). Using this explanation, the simulation model predicts the activation values of the pre-trained models for a given set of sequences. (3) The similarity between the actual activation values from the pre-trained models and the predicted activation values from the simulation model is computed. This similarity score quantifies the semantic consistency of the representation on the selected dimension.

C.2. Model Details

To compare the monosemanticity between the original GPT,and Semi-NGPT, we trained two versions of the GPT2-small model from scratch on the OpenWebText (Gokaslan & Cohen, 2019) dataset using the nanoGPT(Karpathy, 2022) project from GitHub with the default parameters in the script. The details are as follows.

Both models consist of 12 Transformer layers, each featuring 12 attention heads and an embedding dimension of 768. Dropout is disabled, and biases are excluded from both the linear layers and layer normalization layers.

The optimizer used is AdamW, initialized with a learning rate of 6e-4 and a weight decay of 1e-1. The AdamW parameters are set to $\beta_1 = 0.9$ and $\beta_2 = 0.95$. To prevent gradient explosion, gradient clipping is applied with a threshold of 1.0. The training process includes a warm-up phase spanning 2,000 iterations and uses cosine decay to gradually reduce the learning rate to a minimum of 6e-5. The batch size is set to 12, with gradient accumulation over 40 steps to simulate a larger effective batch size. The sequence length is configured to 1,024 tokens. Both models were trained for 136,000 iterations.

The only difference between the original GPT model and our Semi-NGPT model is as follows: In the original model, token indices are processed through token and positional embeddings, followed by 12 transformer layers and a final layer normalization, after which logits are computed using the appropriate head. In contrast, our model applies a ReLU function to the final layer-normalized representations to ensure non-negativity. These non-negative representations are then used to compute logits through a projection head, which is actually a simple linear transformation.

As described in Section 5.2, we trained a Sparse Autoencoder (SAE) on the final-layer representations of the original GPT2-small model. To collect training data for the SAE, we sampled $3 * 2^{17}$ sequences from the OpenWebText dataset. Each sequence was truncated to a maximum length of 64 tokens before being fed into GPT2-small. This resulted in approximately $3 * 2^{23}$ token-level representations used for training. Then we adopted the SAE implementation from a publicly available GitHub repository (Leo Gao, 2023). The hidden dimension of the SAE was set to 8192, and we used the top-k activation function with k=128 to enforce sparsity. The autoencoder was trained for 256 epochs.

C.3. Details of the GLUE Test

To evaluate the model's performance on downstream tasks, we tested the pre-trained model on the General Language Understanding Evaluation (GLUE) benchmark benchmark. For each task, the model was fine-tuned by training a linear classifier following the frozen representations with a maximum sequence length of 128 tokens, a learning rate of 2e-5, and trained for 3 epochs.

We report the average score of Pearson correlation and Spearman's rank correlation for STS-B, average score of accuracy and F1 score for MRPC, Matthews correlation coefficient for CoLA, and accuracy scores for the other tasks.

C.4. Details of Shortening Embeddings

To obtain the feature importance ranking, we randomly sampled 8,192 sequences from the OpenWebText(Gokaslan & Cohen, 2019) dataset, truncating them to a length of 64 tokens and extracting their representations from the model. These representations were then normalized, and the average value for each feature dimension was computed. The features were ranked based on their average values to determine their importance.

Next, we randomly sampled another 8,192 sequences as test samples, also truncated to a length of 64 tokens. Following the feature importance ranking, we progressively blocked the least important dimensions by setting their values to zero and evaluated the model's performance by measuring the loss on next-token prediction for the test samples.

C.5. Details of Detecting and Blocking Toxic Semantics

To identify violence-related dimensions, we first sampled 512 violent sequences and 512 safe sequences from the Aegis-AI-Content-Safety-Dataset-1.0 training set. For each dimension, we recorded its activation values across different sequences, where the activation value for a given dimension in a sequence is defined as the average activation of that dimension across all tokens in the sequence. We then applied the **Student's t-test** to measure the difference in activation values of a given dimension between violent and safe samples. The t-value is computed as follows:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{41}$$

where $\bar{x}_1$ and $\bar{x}_2$ are the mean activation values for the violent and safe samples, respectively, s_1^2 and s_2^2 are the variances of activation values, and $n_1 = n_2 = 512$ represent the number of samples in each category. From a statistical perspective, dimensions with higher t-values are more likely to be associated with violent features.

In the following tasks, we use the Perspective API(Jigsaw & Technology, 2017) to quantitatively measure the toxicity of sequences. We select the first 6 sequences with a toxicity score greater than 0.6 from the Aegis-AI-Content-Safety-Dataset-1.0 test set as prompts. After ranking the dimensions based on their toxicity relevance, we iteratively block the toxicity-related dimensions by setting their values to zero. The model then generates content based on these prompts, producing 6 outputs per prompt with a maximum generation length of 64 tokens. Finally, we evaluate the average toxicity of the generated content, and plot fig3