Stolen Probability: A Structural Weakness of Neural Language Models

David Demeter
Northwestern University
ddemeter@u.northwestern.edu

Gregory Kimmel
H. Lee Moffitt Cancer Center
gregory.kimmel@moffitt.org

Doug Downey
Allen Institute for AI
dougd@allenai.org

Abstract

Neural Network Language Models (NNLMs) generate probability distributions by applying a softmax function to a distance metric formed by taking the dot product of a prediction vector with all word vectors in a high-dimensional embedding space. The dot-product distance metric forms part of the inductive bias of NNLMs. Although NNLMs optimize well with this inductive bias, we show that this results in a sub-optimal ordering of the embedding space that structurally impoverishes some words at the expense of others when assigning probability. We present numerical, theoretical and empirical analyses showing that words on the interior of the convex hull in the embedding space have their probability bounded by the probabilities of the words on the hull.

1 Introduction

Neural Network Language Models (NNLMs) have evolved rapidly over the years from simple feed forward nets (Bengio et al., 2003) to include recurrent connections (Mikolov et al., 2010) and LSTM cells (Zaremba et al., 2014), and most recently transformer architectures (Dai et al., 2019; Radford et al., 2019). This has enabled ever-increasing performance on benchmark data sets. However, one thing has remained relatively constant: the softmax of a dot product as the output layer.

NNLMs generate probability distributions by applying a softmax function to a distance metric formed by taking the dot product of a prediction vector with all word vectors in a high-dimensional embedding space. We show that the dot product distance metric introduces a limitation that bounds the expressiveness of NNLMs, enabling some words to "steal" probability from other words simply due to their relative placement in the embedding space. We call this limitation the *stolen probability effect*. While the net impact of this limitation is small in terms of the perplexity measure on which NNLMs are evaluated, we show that the limitation results in significant errors in certain cases.

As an example, consider a high probability word sequence like "the United States of America" that ends with a relatively infrequent word such as "America". Infrequent words are often associated with smaller embedding norms, and may end up inside the convex hull of the embedding space. As we show, in such a case it is impossible for the NNLM to assign a high probability to the infrequent word that completes the high-probability sequence.

Numerical, theoretical and empirical analyses are presented to establish that the stolen probability effect exists. Experiments with n-gram models, which lack this limitation, are performed to quantify the impact of the effect.

2 Background

In a NNLM, words w_i are represented as vectors x_i in a high-dimensional embedding space. Some combination of these vectors $x_c = \{x_i\}_{i \in c}$ are used to represent the preceding context c, which are fed into a a neural unit as features to generate a prediction vector h_t. NNLMs generate a probability distribution over a vocabulary of words w_i to predict the next word in a sequence w_t using a model of the form:

$$P(w_t|c) = \sigma(f(x_c, \theta_{NNLM})) \quad (1)$$

where σ is the softmax function, f is a neural unit that generates the prediction vector h_t, and θ_{NNLM} are the parameters of the neural unit.

A dot product between the prediction vector h_t and all word vectors x_i is taken to calculate a set of distances, which are then used to form *logits*:

$$z_{it} = x_i \cdot h_t^T + b_i \quad (2)$$

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2191–2197
July 5 - 10, 2020.

where b_i is a word-specific bias term. Logits are used with the softmax function to generate a probability distribution over the vocabulary V such that:

$$P(w_t = w_i|c) = \frac{e^{z_{it}}}{\sum_V e^{z_{vt}}} \quad (3)$$

We refer to this calculation of logits and transformation into a probability distribution as the *dot-product softmax*.

3 Problem Definition

NNLMs learn very different embeddings for different words. In this section we show that this can make it impossible for words with certain embeddings to *ever* be assigned high probability in *any* context. We start with a brief examination of the link between embedding norm and probability, which motivates our analysis of the stolen probability effect in terms of a word's position in the embedding space relative to the convex hull of the embedding space.

3.1 Embedding Space Analysis

The dot product used in Eq. 2 can be written in polar coordinates as:

$$z_{it} = \|x_i\|\|h_t\| \, cos(\theta_i) + b_i \quad (4)$$

where θ_i is the angle between x_i and h_t. The dot-product softmax allocates probability to word w_i in proportion to z_{it}'s value relative to the value of other logits (see Eq. 3). Setting aside the bias term b_i for the moment (which is shown empirically to be irrelevant to our analysis in Section 4.2), this means that word A with a larger norm than word B will be assigned higher probability when the angles θ_A and θ_B are the same.

More generally, the relationship between embedding norms and the angles formed with prediction points h_t can be expressed as:

$$\frac{\|x_A\|}{\|x_B\|} > \frac{cos(\theta_B)}{cos(\theta_A)} \quad (5)$$

when word A has a higher probability than word B. Empirical results (not presented) confirm that NNLMs organize the embedding space such that word vector norms are widely distributed, while their angular displacements relative to a reference vector fall into a narrow range. This suggests that the norm terms in Eq. 4 dominate the calculation of logits, and thereby probability.

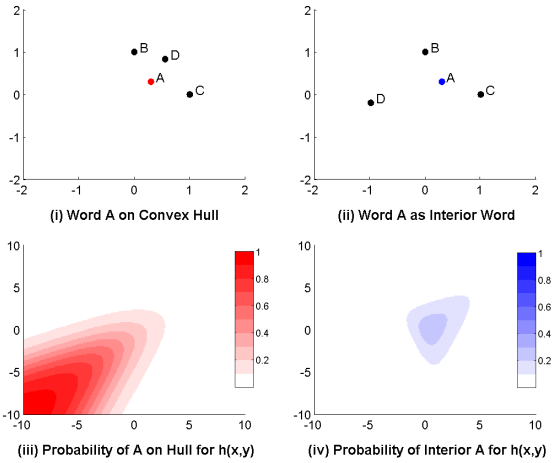

Figure 1: **Numerical Illustration of the Stolen Probability Effect.** Panels (i) and (ii) show the embedding of four words in a 2D embedding space. Word A is on the convex hull in panel (i), and interior to the convex hull in panel (ii). Panels (iii) and (iv) show the probability that would be assigned by the dot-product softmax to A for a range of prediction points h_t in the x, y plane. When word A is on the convex hull, it can achieve nearly 100% probability for an h_t prediction point in the far lower-left quadrant (see panel (iii)). When word A is interior to the convex hull, its maximum probability is bounded by any word on the convex hull (see panel (iv)).

3.2 Theoretical Analysis

While an analysis of how embedding norms impact the assignment of probability is informative, the stolen probability effect is best analyzed in terms of a word's position in the embedding space relative to the convex hull of the embedding space. A convex hull is the smallest set of points forming a convex polygon that contains all other points in a Euclidean space.

Theorem 1. *Let C be the convex hull of the embeddings $\{x_i\}$ of a vocabulary V. If an embedding x_i for word $w_i \in V$ is interior to C, then the maximum probability $P(w_i)$ assigned to w_i using a dot-product softmax is bounded by the probability assigned to at least one word w_i whose embedding is on the convex hull.* (see Appendix A for proof).

3.3 Numerical Analysis

The stolen probability effect can be illustrated numerically in a 2D Euclidean space (see Figure 1). We show two configurations of an embedding space, one where target word A is on the convex hull (Panel i) and another where A is on the interior (Panel ii). Under both configurations, a NNLM

trained to the maximum likelihood objective would seek to assign probability such that $P(A) = 1.0$.

For the first configuration, this is achievable for an h_t in the far lower-left quadrant (Panel iii). However, when A is in the interior, there is no h_t that exists where the dot-product softmax can assign a probability approaching 1.0 (Panel iv). A similar illustration in 3D is presented in Appendix B.

4 Experiments

In this section we provide empirical evidence showing that words interior to the convex hull are probability-impoverished due to the stolen probability effect and analyze the impact of this phenomenon on different models.

4.1 Methods

We perform our evaluations using the AWD-LSTM (Merity et al., 2017) and the Mixture of Softmaxes (MoS) (Yang et al., 2017) language models. Both models are trained on the Wikitext-2 corpus (Merity et al., 2016) using default hyper-parameters, except for dimensionality which is set to $d = \{50, 100, 200\}$. The AWD-LSTM model is trained for 500 epochs and the MoS model is trained for 200 epochs, resulting in perplexities as shown in Table 1.

The Quickhull algorithm (Barber et al., 1996) is among the most popular algorithms used to detect the convex hull in Euclidean space. However, we found it to be intractably slow for embedding spaces above ten dimensions, and therefore resorted to approximate methods. We relied upon an identity derivable from the properties of a convex hull which states that *a point $p \in \mathbb{R}^d$ is vertex of the convex hull of $\{x_i\}$ if there exists a vector $h_t \in \mathbb{R}^d$ such that for all x_i:*

$$\langle h_t, x_i - p \rangle < 0. \tag{6}$$

where $\langle \cdot \rangle$ is the dot-product.

Searching for directions h_t which satisfy Eq 6 is not computationally feasible. Instead, we rely upon a high-precision, low-recall approximate method to eliminate potential directions for h_t which do not satisfy Eq. 6. We call this method our *detection algorithm*. If the set of remaining directions is not empty, then p is classified as a vertex, otherwise p is classified as an interior point.

The detection algorithm is anchored by the insight that all vectors parallel to the difference vector $\vec{x_i} - \vec{p}$ do not satisfy Eq. 6. It is also true that all directions in the range $(\phi + \omega, \phi - \omega)$ will not satisfy Eq. 6, where ϕ is the direction of the difference vector and ω is some increment less than $\pi/2$. The detection algorithm was validated in lower dimensional spaces where an exact convex hull could be computed (*e,g.* up to $d = 10$). It consistently classified interior points with precision approaching 100% and recall of 68% when evaluated on the first 10 dimensions of the MoS model with $d = 100$.

Model	*d*	Train PPL	Test PPL	ω (radians)	Interior Points
AWD	50	140.6	141.8	$50\pi/128$	6,155
AWD	100	73.3	97.8	$55\pi/128$	5,205
AWD	200	44.9	81.6	$58\pi/128$	2,064
MoS	50	51.7	76.8	$53\pi/128$	4,631
Mos	100	34.8	67.4	$57\pi/128$	4,371
MoS	200	25.5	64.2	$59\pi/128$	2,009

Table 1: **Perplexities and Detection Results.** Each model was trained using default hyper-parameters except for dimensions d as shown and number of training epochs. The AWD-LSTM models we trained for 500 epochs and the MoS models were trained for 200 epochs. Each ordinal plane of an *n-Sphere* in the embedding space was discretized into arcs of $2\pi/256$. The angle ϕ of the difference vector $x_i - p$ formed each word type embedded at p is mapped to one of these arcs. Directions on the interval $(\phi \pm \omega)$ are eliminated from consideration per Eq 6, and words for which all directions have been eliminated as classified as interior. The increment ω was set to the lowest value that would classify at least 1,000 words as interior.

4.2 Results

Applying the detection algorithm to our models yields word types being classified into distinct *interior* and *non-interior* sets (see Table 1). We ranked the top 500 words of each set by the maximum probability they achieved on the training corpora[1], and plot these values in Figure 2, showing a clear distinction between interior and non-interior sets. The maximum trigram probabilities (Stolcke, 2002) smoothed with KN3 for the same top 500 words in each set (separately sorted) are also shown. The difference between NNLM and trigram curves for interior words shows that models like n-grams, which do not utilize a dot-product softmax, are *not* subject to the stolen probability effect and can assign higher probabilities. A *random* set of words equal

[1]We present our results on the training set because here, our goal is to characterize the expressiveness of the models rather than their ability to generalize.

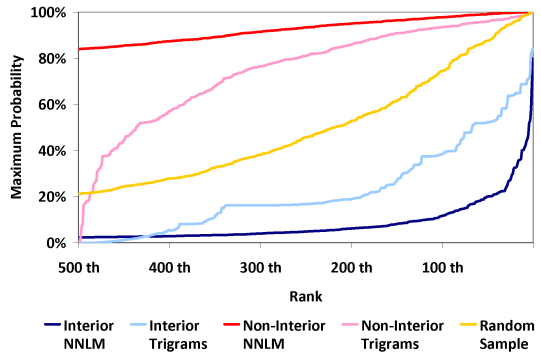

Figure 2: **Maximum Probability of Top 500 Interior and Non-Interior Words.** The MoS model with $d = 100$ struggles to assign high probability to interior words, while trigrams were able to capture more accurate statistics. This behavior is absent for non-interior words.

Model	d	Non-Interior	Rand	Tri-gram	Interior
AWD	50	44.3	8.1	20.7	0.004
AWD	100	89.2	31.3	15.6	0.018
AWD	200	99.0	43.3	12.5	0.113
MoS	50	76.5	22.9	16.8	0.4
MoS	100	92.9	50.5	22.6	8.6
MoS	200	97.3	51.4	30.9	40.0

Table 2: **Average Maximum Probability for Top 500 Words.** The average probability mass for each word set (expressed as percents) is calculated by averaging the maximum probability on the training corpora achieved for the top 500 words of each set.

in size to the interior set was also constructed by uniform sampling, and ranked on the top 500 words. A comparison between the random and interior sets provides evidence that our detection algorithm is effective at separating the interior and non-interior sets, and is not simply performing random sampling.

Our results can be more compactly presented by considering the average probability mass assigned to the top 500 words for each set (see Table 2). The impact of the stolen probability effect for each model can quantified as the difference between the interior set and each of the three reference sets (non-interior, random, and trigram) in the table. The interior average maximum probability is generally much smaller than those of the reference sets.

Another way to quantify the impact of the stolen probability effect is to overcome the bound on the interior set by constructing an ensemble with trigram statistics. We constructed a targeted ensemble of the MoS model with $d = 100$ and a trigram model—unlike a standard ensemble, the trigram model is only used in contexts that are likely to indicate an interior word: specifically, those that precede at least one interior word in the training set. Otherwise, we default to the NNLM probability. When we ensemble, we assign weights of 0.8 to the NNLM, 0.2 to the trigram (selected using the training set). Overall, the targeted ensemble improved training perplexity from 34.8 to 33.6, and test perplexity from 67.4 to 67.0. The improvements on the interior words themselves were much larger: training perplexities for interior words improved from 700.0 to 157.2, and test improved from 645.6 to 306.7. Improvement on the interior words is not unexpected given the differences observed in Figure 2. The overall perplexity differences, while small in magnitude, suggest that ensembling with a model that lacks the stolen probability limitation may provide some boost to a NNLM.

Returning to the question of bias terms, we find empirically that bias terms are relatively small, averaging -0.13 and 0.02 for the interior and non-interior sets of the MoS model with $d = 100$, respectively. We note that the bias terms are word-specific and can only adjust the stolen probability effect by a constant factor. That is, it does not change the fact that words in the interior set are probability-bounded. All of our empirical results are calculated on a model with a bias term, demonstrating that the stolen probability effect persists with bias terms.

4.3 Analysis

Attributes of the stolen probability effect analyzed in this work are distinct from the *softmax bottleneck* (Yang et al., 2017). The softmax bottleneck argues that language modeling can be formulated as a factorization problem, and that the resulting model's expressiveness in limited by the rank of the word embedding matrix. While we also argue that the expressiveness of a NNLM is limited for structural reasons, the stolen probability effect that we study is best understood as a property of the *arrangement* of the embeddings in space, rather than the dimensionality of the space.

Our numerical and theoretical analyses presented do not rely upon any particular number of dimensions, and our experiments show that the stolen probability effect holds over a range of dimensions. However, there is a steady increase of average probability mass assigned to the interior set

as model dimensionality increases, suggesting that there are limits to the stolen probability effect. This is not unexpected. As the capacity of the embedding space increases with additional dimensions, the model has additional degrees of freedom in organizing the embedding space. The vocabulary of the Wikitext-2 corpus is small compared to other more recent corpora. We believe that larger vocabularies will offset (at least partially) the additional degrees of freedom associated with higher dimensional embedding spaces. We leave the exploration of this question as future research.

We acknowledge that our results can also be impacted by the approximate nature of our detection algorithm. Without the ability to precisely detect detect the convex hull for any of our embedding spaces, we can not make precise claims about its performance. The difference between average probability mass assigned to random and interior sets across all models evaluated suggests that the detection algorithm succeeds at identifying words with substantially lower maximum probabilities than a random selection of words.

In Section 3.1 we motivated our analysis of the stolen probability effect by examining the impact of embeddings norms on probability assignment. One natural question is to ask is *"Does our detection algorithm simply classify embeddings with small norms as interior points?"* Our results suggest that this is not the case. The scatter plot of embedding norm versus maximum probability (see Figure 3) shows that words classified as interior points frequently have lower norms. This is expected, since points interior to the convex hull are by definition not located in extreme regions of the embedding space. The embedding norms for words in the interior set range between 1.4 and 2.6 for the MoS model with $d = 100$. Average maximum probabilities for words in this range are 1.4% and 4.1% for interior and non-interior sets of the MoS model with $d = 100$, respectively, providing evidence that the detection algorithm is not merely identifying word with small embedding norms.

Lastly, we note that the interior sets of the AWD-LSTM models are particularly probability impoverished relative to the more powerful MoS models. We speculate that the perplexity improvements of the MoS model may be due in part to mitigating the stolen probability effect. Exploration of the stolen probability effect in more powerful NNLM architectures using dot-product softmax output layers is another item of future research.

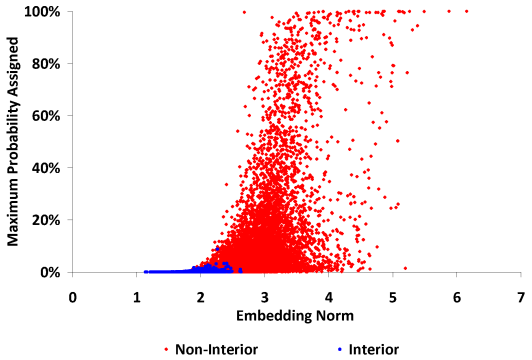

Figure 3: **Maximum Probability vs. Embedding Norm.** Examining maximum word probability as a function of embedding norm for the MoS model with $d = 100$ shows that interior words are associated with smaller embedding norms and lower maximum probabilities. However, many non-interior words with comparably small norms have substantially higher maximum probabilities.

5 Related Work

Other work has explored alternative softmax configurations, including a mixture of softmaxes, adaptive softmax and a Taylor Series softmax (Yang et al., 2017; Grave et al., 2016; de Brébisson and Vincent, 2015). There is also a body of work that analyzes the properties of embedding spaces (Burdick et al., 2018; Mimno and Thompson, 2017). We do not seek to modify the softmax. Instead we present an analysis of how the structural bounds of an NNLM limit its expressiveness.

6 Conclusion

We present numerical, theoretical and empirical analyses showing that the dot-product softmax limits a NNLM's expressiveness for words on the interior of a convex hull of the embedding space. This is structural weakness of NNLMs with dot-product softmax output layers, which we call the *stolen probability effect*. Our experiments show that the effect is relatively common in smaller neural language models. Alternative architectures that can overcome the stolen probability effect are an item of future work.

Acknowledgments

This work was supported in part by NSF Grant IIS-1351029. We thank the anonymous reviewers and Northwestern's Theoretical Computer Science group for their insightful comments and guidance.

References

C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22:469–483.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *JOURNAL OF MACHINE LEARNING RESEARCH*, 3:1137–1155.

Alexandre de Brébisson and Pascal Vincent. 2015. An exploration of softmax alternatives belonging to the spherical loss family. *CoRR*, abs/1511.05042.

Laura Burdick, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors influencing the surprising instability of word embeddings. In *NAACL-HLT*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*.

Edouard Grave, Armand Joulin, Moustapha Cisse, David Grangier, and Herve Jegou. 2016. Efficient softmax approximation for gpus. *ArXiv*, abs/1609.04309.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models. *ArXiv*, abs/1708.02182.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan ernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*.

David M. Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *EMNLP*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *INTERSPEECH*.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2017. Breaking the softmax bottleneck: A high-rank rnn language model. *ArXiv*, abs/1711.03953.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *ArXiv*, abs/1409.2329.

A Proof of Theorems

Proof of Theorem 1.

Let $P = \{x_1, \ldots, x_N\}$ be the set of all words. We can form the convex hull of this set. If p is interior, then for all v, there exists an $x_i \in P$ such that $\langle v, x_i - p\rangle > 0$. We argue by contradiction. Suppose that p is interior and that for all v, we have that $\langle v, x_i - p\rangle \leq 0$ for all $x_i \in P$. This implies that all points in our set P lay strictly on one side of the hyperplane made perpendicular to v through p. This would imply that p was actually on the convex hull, a contradiction.

This implies that for any test point h, an interior point will be bounded by at least one point in P. That is $\langle h, p\rangle < \langle h, x_i\rangle$ for some $x_i \in P$. Plugging into the softmax function we see that:

$$\begin{aligned}\mathbb{P}(p) &= \frac{\exp(\langle h, p\rangle)}{\exp(\langle h, p\rangle) + \sum_{j\neq p}\exp(\langle h, x_j\rangle)} \\ &\leq \frac{1}{1 + \exp(\langle h, x_i - p\rangle)}\end{aligned}$$

Letting $\|h\| \to \infty$ shows that $\mathbb{P}(p) \to 0$. This shows that interior points are probability deficient. We also note that letting $\|h\| \to 0$ gives the base probability $\mathbb{P}(p) = 1/|P|$.

The contrapositive of the above statement implies that if $\nexists\ v$, where $\forall\ x_i \in p$ we have $\langle v, x_i - p\rangle \leq 0$, then p is on the convex hull. In fact, we also note that if p was a vertex, the inequality would be strict, which implies that one can find a test point such that the probability $\mathbb{P}(p) \to 1$.

The more interesting case is if the point p is on the convex hull, but not a vertex. In this case we define the set $\Omega(p, h) = \{x_i \in P \mid \langle h, p - x_i\rangle = 0\}$. This corresponds to the set of points lying directly on the hyperplane perpendicular to h, running through p. This set is nonempty. Then we see that:

$$\begin{aligned}\mathbb{P}(p) &= \frac{\exp(\langle h, p\rangle)}{\sum_j \exp(\langle h, x_j\rangle)} \\ &\leq \frac{\exp(\langle h, p\rangle)}{\sum_{j\in\Omega(p,h)} \exp(\langle h, x_j\rangle)} \\ &= \frac{1}{|\Omega(p, h)|}\end{aligned}$$

B 3D Numerical Illustration

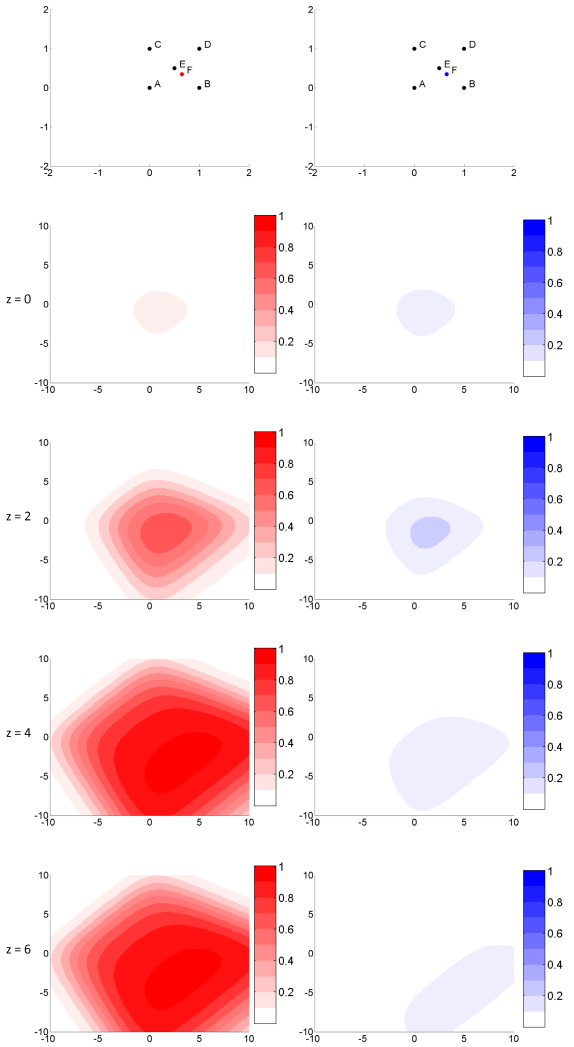

Figure 4: **Numerical Illustration in 3D** The top panels show six words in a flattened cross-section of 3D space. Points A, B, C, D and E are embedded at $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, $(1, 1, 0)$ and $(0.5, 0.5, 1)$ respectively. In the top-left panel, F is embedded outside of the convex hull at $(0.65, 0.35, 1.5)$, and in the top-right panel F is embedded inside of the convex hull at $(0.65, 0.35, 0.5)$. Subsequent panels show cross sections of the probability of F for test points in the plains $z = \{0.0, 2.0, 4.0, 6.0\}$, numerically illustrating the stolen probability effect in 3D.