Investigating Conceptual Blending of Diffusion Models for Improving Nonword-to-Image Generation

Chihaya Matsuhira
matsuhirac@cs.is.i.nagoya-u.ac.jp
Nagoya University
Nagoya, Japan

Marc A. Kastner
mkastner@hiroshima-cu.ac.jp
Hiroshima City University
Hiroshima, Japan

Takahiro Komamizu
taka-coma@acm.org
Nagoya University
Nagoya, Japan

Takatsugu Hirayama
t-hirayama@uhe.ac.jp
University of Human Environments
Okazaki, Japan

Ichiro Ide
ide@i.nagoya-u.ac.jp
Nagoya University
Nagoya, Japan

A Supplementary Materials

A.1 Dataset Creation

A.1.1 EvalNouns1000. *EvalNouns1000* is a list of 1,000 nouns created by our paper as evaluation data. This wordlist is used in our paper mainly for the following two purposes:

- To create 10,000 matching and 10,000 mismatching pairs in Section 3.
- To create 1,000 interpolated embeddings in Section 4.3.

As mentioned in our paper, these nouns are randomly taken from the MRC Psycholinguistic Database [2] with the restrictions of word imageability and frequency. This database provides an imageability score for each noun ranging from 100 to 700, where a high score indicates that the noun is highly imageable. Although it also provides word frequency scores, we do not use them but instead use the Python package `wordfreq` [24]. This is to ensure *EvalNouns1000* to be a subset of another dataset *TrainWords26143*, which will be described later. Selecting words with 500 or more imageability scores and 3.5 or more Zipf frequency values resulted in 1,183 words in total. *EvalNouns1000* is created by randomly sampling 1,000 words from these words.

A.1.2 TrainWords26143. *TrainWords26143* is a list of 26,143 words compiled by an existing study on nonword-to-image generation [9–11]. Our paper uses this wordlist mainly for the following three purposes:

- Used by the proposed embedding space conversion method to create anchors of k-nearest neighbor search and linear regression.
- These anchors are also used for calculating Spearman's rank correlation metrics in Section 4.3.
- A minor-modified one is used to train a comparative Multi-Layer Perceptron (MLP).

As mentioned in our paper, the existing study created this wordlist using the Spell Checker Oriented Word Lists (SCOWL)[1] and 26,143 words were selected based on word frequency and pronunciation availability. Specifically, a Python package `wordfreq` [24] was used to remove words having Zipf frequency less than 3.0. Also, the Carnegie Mellon University (CMU) dictionary[2] was looked up for checking the pronunciation availability.

The modified wordlist used to train the MLP consists of 26,455 words, which was created by adding 312 words filtered out during the pronunciation availability check.

A.1.3 Training Data of NonwordCLIP. To train a NonwordCLIP [9–11] in Section 4, we constructed a dataset in which each word appears almost an equal number of times. As mentioned in our paper, the dataset consists of 5,496 highly-imageable and -frequent nouns and noun phrases created by combining the MRC Psycholinguistic Database [2], `wordfreq` [24], and an English lexical database WordNet [13].

First, from the MRC database, we collected highly-imageable nouns having an imageability score of 500 or more. Next, we used WordNet to augment the vocabulary based on Liu et al. [28]'s procedure, in which synonym and hyponym relationships on WordNet were used to extend the imageability dictionary. Specifically, for each noun in an imageability dictionary, their method propagated the same imageability score to the synonyms and hyponyms of the noun. Following this policy, for each word in our imageable noun list, we propagated its imageability score to its first, second, and third synonym nouns and all hyponym nouns. Natural Language ToolKit (NLTK) [27] was used to access the WordNet hierarchy and to judge whether each WordNet node is a noun or a noun phrase. After this augmentation, we used `wordfreq` to obtain nouns having 3.5 or more Zipf frequency values.

Lastly, we further augmented the dataset twice using the two prompts "`<WORD>`" and "`a photo of a <WORD>`", resulting in training data of 10,992 samples.

A.2 Prompt Engineering for Calculating CLIP Score

In Section 3.2, Contrastive Language-Image Pretraining (CLIP) score [15] was calculated to detect the presence of a single concept in an image. To increase the precision of the scores, we adopted prompt engineering like the one adopted in the original paper [15] to solve an image classification task[3]. The original paper used 80 templates describing images containing a target concept, all of which ends with a period, such as "`a bad photo of a <WORD>.`". Our paper increased the number of templates to 160 by creating a

[1] http://wordlist.aspell.net/ (Accessed August 7, 2024)
[2] https://github.com/menelik3/cmudict-ipa/ (Accessed August 7, 2024)
[3] https://github.com/openai/CLIP/blob/main/notebooks/Prompt_Engineering_for_ImageNet.ipynb (Accessed August 7, 2024)

Table 5: Ratios of respective cases when inputting interpolated embeddings between concepts A and B with different interpolation ratios measured under the setting $n = 1$.

	Interpolation Ratio of Concept A to Concept B									
Case	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Overall
Concept A	0.286	0.466	0.580	0.855	0.974	0.991	1.000	1.000	1.000	0.802
Concept B	1.000	1.000	1.000	1.000	0.957	0.870	0.648	0.495	0.496	0.829
BCD	0.286	0.466	0.562	0.744	**0.819**	0.778	0.600	0.465	0.487	0.584
MCD	0.286	0.466	0.580	0.855	**0.931**	0.861	0.648	0.495	0.496	0.631

Table 6: Ratios of respective cases when inputting interpolated embeddings between concepts A and B with different interpolation ratios measured under the setting $n = 2$ (Same as Table 1 in our paper).

	Interpolation Ratio of Concept A to Concept B									
Case	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Overall
Concept A	0.152	0.311	0.411	0.709	0.948	0.981	1.000	1.000	1.000	0.732
Concept B	1.000	1.000	1.000	0.991	0.914	0.731	0.472	0.277	0.265	0.738
BCD	0.143	0.301	0.348	0.521	**0.621**	0.593	0.416	0.257	0.257	0.389
MCD	0.152	0.311	0.411	0.701	**0.862**	0.722	0.472	0.277	0.265	0.471

Table 7: Ratios of respective cases when inputting interpolated embeddings between concepts A and B with different interpolation ratios measured under the setting $n = 5$.

	Interpolation Ratio of Concept A to Concept B									
Case	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Overall
Concept A	0.010	0.107	0.134	0.291	0.716	0.898	0.984	1.000	1.000	0.578
Concept B	1.000	1.000	0.982	0.897	0.698	0.380	0.208	0.099	0.053	0.587
BCD	0.000	0.097	0.116	0.162	**0.302**	0.204	0.184	0.099	0.053	0.138
MCD	0.010	0.107	0.134	0.214	**0.466**	0.306	0.208	0.099	0.053	0.181

variant without the period in the ending position for each template, such as "`a bad photo of a <WORD>`".

For each pair of a concept and an image, we calculated the final CLIP score by averaging the 160 CLIP similarity scores computed for each prompt.

A.3 Detailed Experimental Results

A.3.1 Results under Different ns. Tables 5, 6, and 7 show the ratios of conceptual blending evaluated in Section 3 under different ns. Our conclusions mentioned in the paper are consistent throughout all ns, while the ratio decreases as n increases because setting a larger n makes the detection criterion more strict.

A.3.2 Results under Different ℓs. Table 8 shows the transition of the rank correlation metric used in Section 4.3 with different ℓs. The hyperparameter ℓ denotes how many nearest-neighbor embeddings in both the CLIP pooled and last-hidden-state embedding spaces are used to calculate the rank correlation. As a reference, we also measured the rank correlation metric between the nearest-neighbor ranking for the ground-truth interpolated embedding in the pooled embedding space and that for the ground-truth interpolated embedding in the last-hidden-state embedding space, averaged over all samples. This metric, shown as "Ground Truth" in the table, measures the alignment of the sample distributions in the two embedding spaces.

Table 8: Spearman's rank correlation under different ℓs. For all metrics, a higher score indicates a higher consistency in neighborhood relationships before and after the embedding space conversion.

Method	$\text{RCorr}_{\ell=2}$	$\text{RCorr}_{\ell=5}$	$\text{RCorr}_{\ell=10}$	$\text{RCorr}_{\ell=100}$	$\text{RCorr}_{\ell=26143}$
MLP [9–11]	0.846	0.783	0.711	**0.464**	**0.444**
Ours ($k =$ 1)	**0.902**	0.702	0.586	0.352	0.363
Ours ($k =$ 2)	0.880	0.788	0.669	0.376	0.365
Ours ($k =$ 5)	0.884	0.765	**0.735**	0.395	0.366
Ours ($k =$ 10)	0.882	0.771	0.689	0.397	0.366
Ours ($k =$ 100)	0.888	0.781	0.694	0.373	0.365
Ours ($k =$ 200)	0.886	0.791	0.697	0.373	0.365
Ours ($k =$ 300)	0.890	0.793	0.699	0.373	0.365
Ours ($k =$ 400)	0.890	0.797	0.700	0.373	0.365
Ours ($k =$ 500)	0.880	**0.802**	0.701	0.373	0.364
Ours ($k =$1,000)	0.882	0.799	0.696	0.367	0.359
Ground Truth	0.868	0.809	0.703	0.373	0.365

The results in the table indicate that the comparative MLP-based method yielded higher correlations than the proposed method under a large ℓ, and they were even higher than the metrics measured using the ground-truth interpolated embeddings presumably due to the curse of dimensionality.

A.4 Similarity between Embedding Spaces

In Section 4.2, our paper assumed similar data distributions in the CLIP pooled embedding space and the CLIP last-hidden-state embedding space to convert embeddings from one space to another. To support this assumption, this section quantitatively evaluates how similar the two embedding spaces are.

First, we describe how embeddings in each space are computed by the CLIP adopted in our paper. In the forwarding step of CLIP, it first computes a 77×768-dimensional last-hidden-state embedding for a given text prompt. Each of the 77 dimensions corresponds to the tokens of the tokenized text prompt. For instance, if the prompt is "`a photo of a calf`" and its tokenized result is [[CLS], 'a', 'photo', 'of', 'a', 'calf', [EOS]], the first and seventh dimensions of the embedding correspond to the [CLS] token and the [EOS] token, respectively, Next, a 768-dimensional pooled embedding is computed by linearly projecting the 1×768-dimensional [EOS] token embedding chosen from the 77×768-dimensional whole last-hidden-state embedding. This means that the last-hidden-state embedding of only the [EOS] position is linearly predictable from a pooled embedding.

To quantify the actual inter-space similarity, we conducted Canonical Correlation Analysis (CCA) between the two spaces using the whole dataset of *TrainWords26143* as data samples. Due to the huge dimensionality of the last-hidden-state embedding space, CCA is applied token-wise; Correlations are measured between the pooled embedding space and each dimension of the 77 token positions of the last-hidden-state embedding space.

The result is shown in Fig. 8. The figure shows more than a 0.90 maximum correlation between the pooled embedding and each slice of the corresponding last-hidden-state embedding later than the fifth token position, while the other earlier slices yield almost no correlation. The fifth and earlier slices rarely vary among

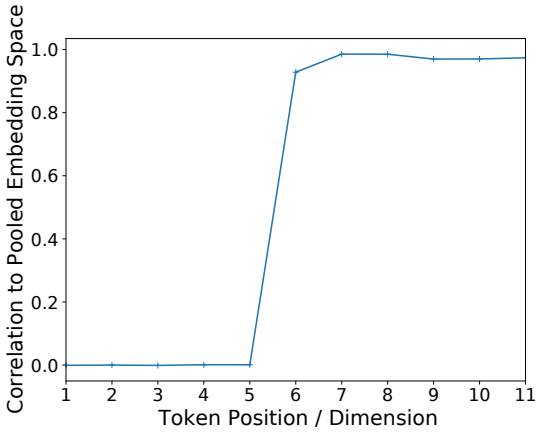


Figure 8: CCA result. The y-axis indicates the maximum correlation between the pooled embedding space and each dimension of the last-hidden-state embedding space corresponding to each token position up to the first 11 tokens.

each sample under our experimental setting since they always correspond to the tokens [[CLS], 'a', 'photo', 'of', 'a']. On the other hand, the slices after them vary, truly representing the text features expressed in the last-hidden-embedding space. Therefore, the high correlation of the pooled embedding space to the slices after them allows us to conclude that the data distributions in the two spaces are similar enough to ensure that the linear combination in Eq. (3) works.

A.5 Image Generation Examples

A.5.1 Image Generation from Interpolated Embeddings. Figure 9 shows more conceptual blending examples yielded by inputting interpolated embeddings into Stable Diffusion. These cases of conceptual blending are detected by our classifier constructed in Section 3.

Figures 10 and 11 compare images generated from embeddings computed by different text embedding space conversion methods evaluated in Section 4.3. Also, at the bottom right of each figure, the images generated from the ground-truth last-hidden-state embeddings used in Section 3 are attached for comparison.

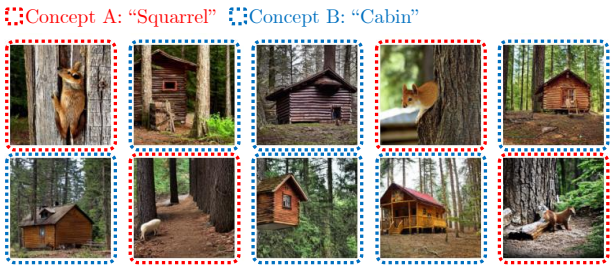
A.5.2 Nonword-to-Image Generation. Figure 12 showcases more nonword-to-image generation results generated by different methods used in the evaluation in Section 4.4. As mentioned in Section 4.4.2, these nonwords are taken from Sabbatino et al. [19]'s work, in which they annotated evoked emotion labels to each of them.

References

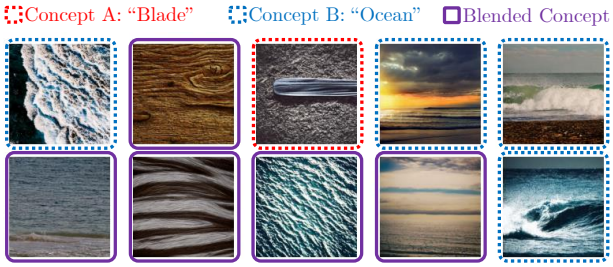
[27] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* O'Reilly Media, Inc., Sebastopol, CA, USA.

[28] Ting Liu, Kit Cho, G. Aaron Broadwell, Samira Shaikh, Tomek Strzalkowski, John Lien, Sarah Taylor, Laurie Feldman, Boris Yamrom, Nick Webb, Umit Boz, Ignacio Cases, and Ching-sheng Lin. 2014. Automatic expansion of the MRC psycholinguistic database imageability ratings. In *Proc. 9th Int. Conf. Lang. Resour. Eval.* (Reykjavik, Iceland). 2800–2805.

(a) Image generation results from interpolated embeddings between "*jar*" and "*money*" with an interpolation ratio = 0.5, showing Blended Concept Depiction (BCD).



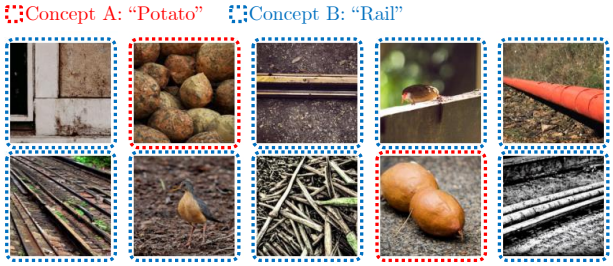
(b) Image generation results from interpolated embeddings between "*squirrel*" and "*cabin*" with an interpolation ratio = 0.4, showing Mixed Concept Depiction (MCD).



(c) Image generation results from interpolated embeddings between "*blade*" and "*ocean*" with an interpolation ratio = 0.5, showing BCD.

(d) Image generation results from interpolated embeddings between "*fireplace*" and "*mixer*" with an interpolation ratio = 0.4, showing MCD.

(e) Image generation results from interpolated embeddings between "*priest*" and "*kite*" with an interpolation ratio = 0.4, showing BCD.



(f) Image generation results from interpolated embeddings between "*potato*" and "*rail*" with an interpolation ratio = 0.4, showing MCD.

(g) Image generation results from interpolated embeddings between "*machine*" and "*rainbow*" with an interpolation ratio = 0.4, showing BCD.

(h) Image generation results from interpolated embeddings between "*phone*" and "*disaster*" with an interpolation ratio = 0.5, showing MCD.

Figure 9: More conceptual blending examples of Stable Diffusion detected by our classifier.

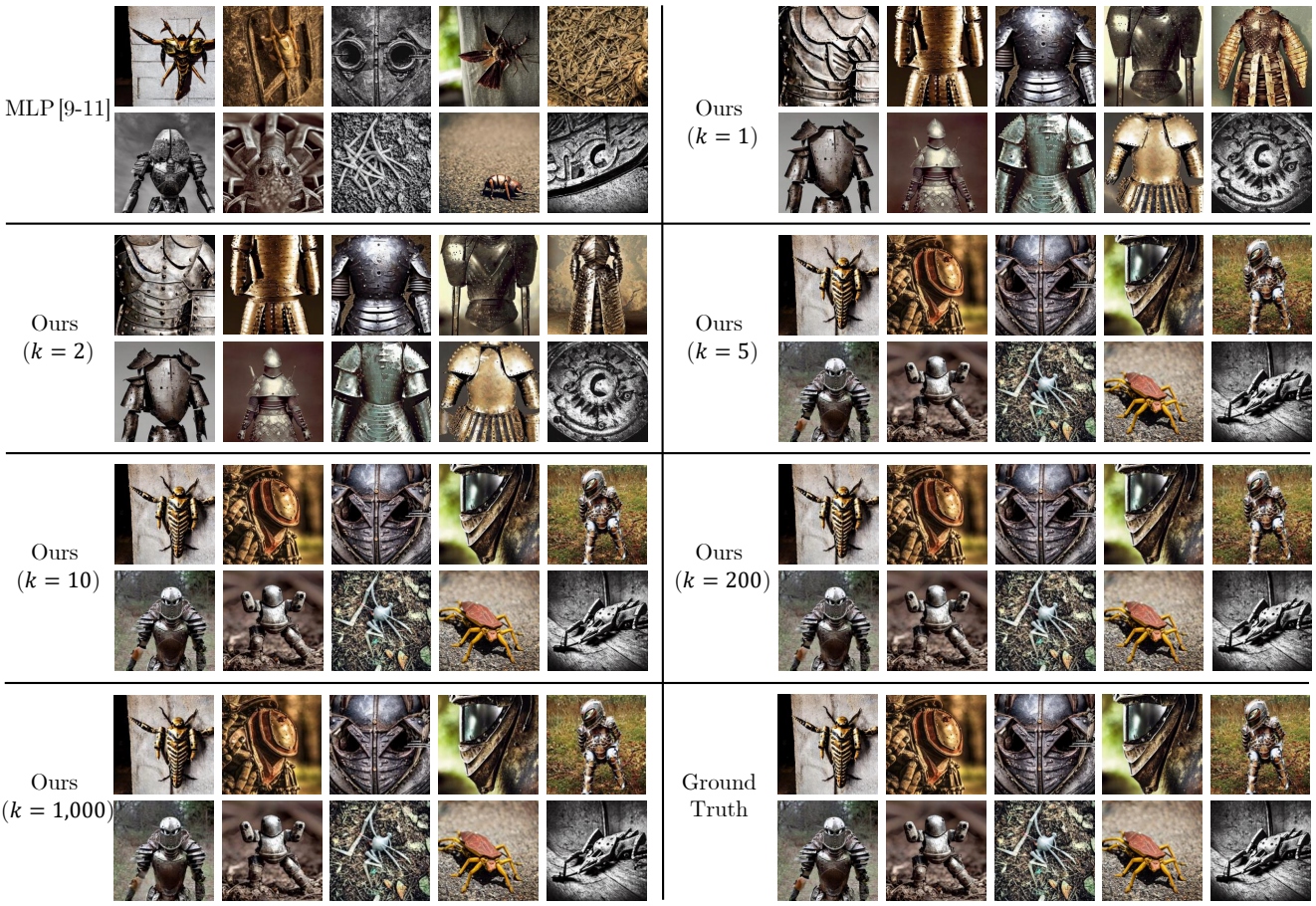


Figure 10: Image generation results generated from interpolated embeddings between Concept A = "*armour*" and Concept B = "*spider*" with an interpolation ratio = 0.6 using different methods.

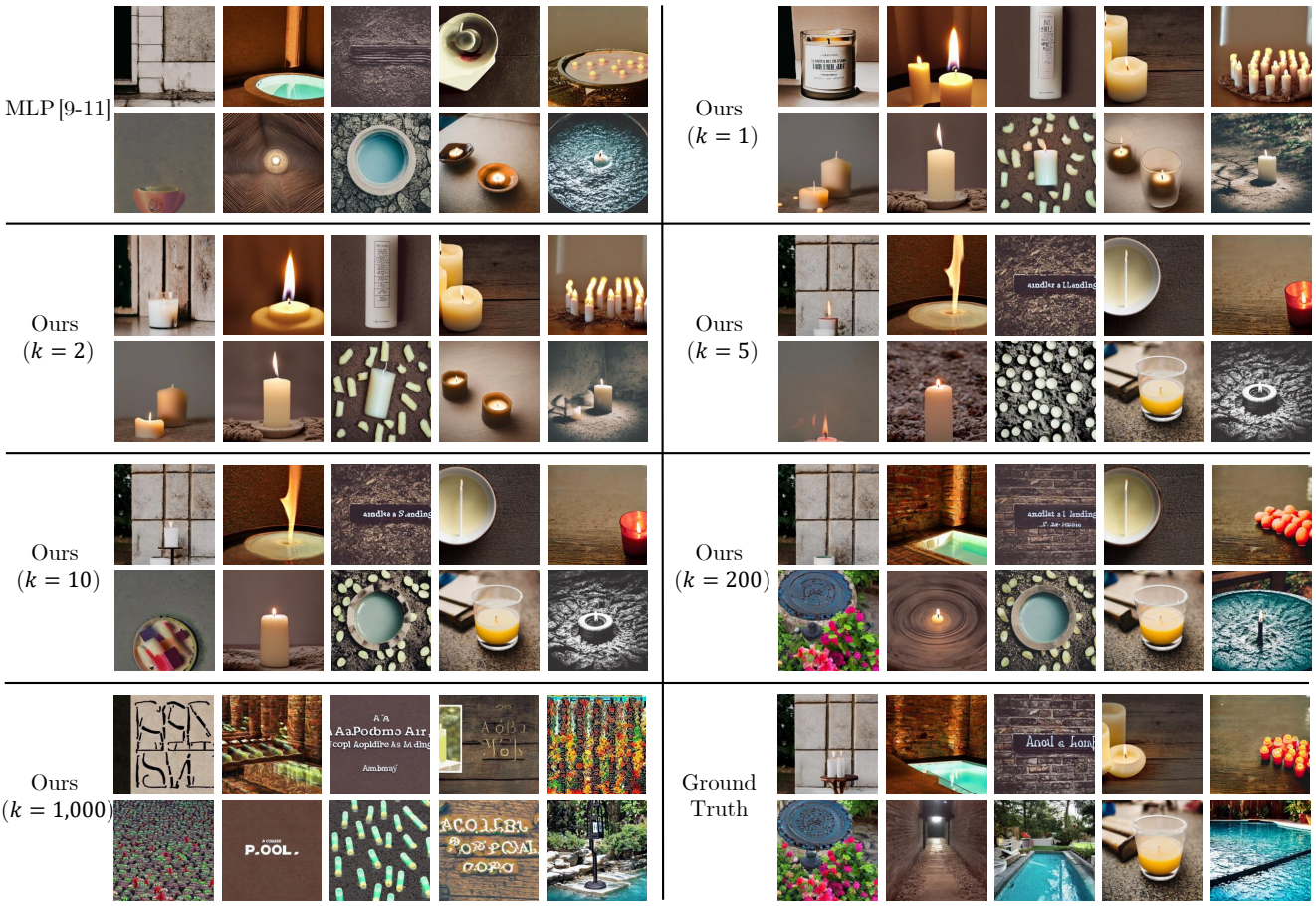
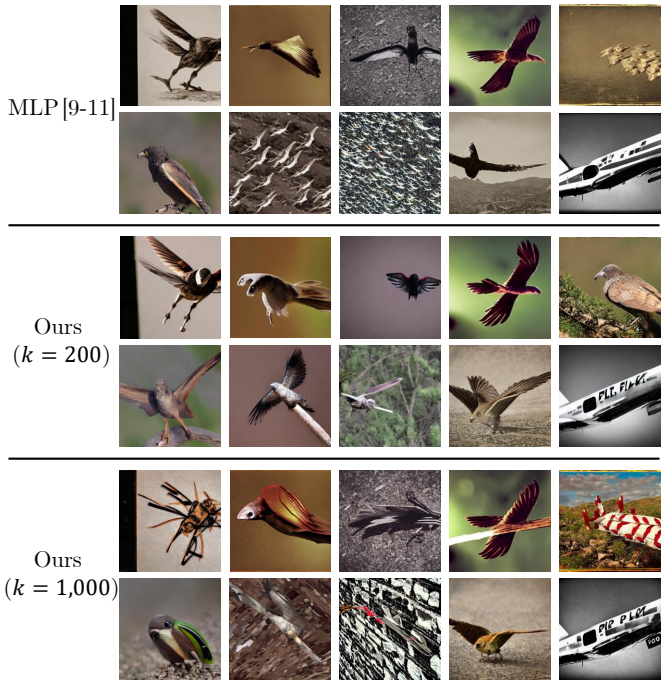
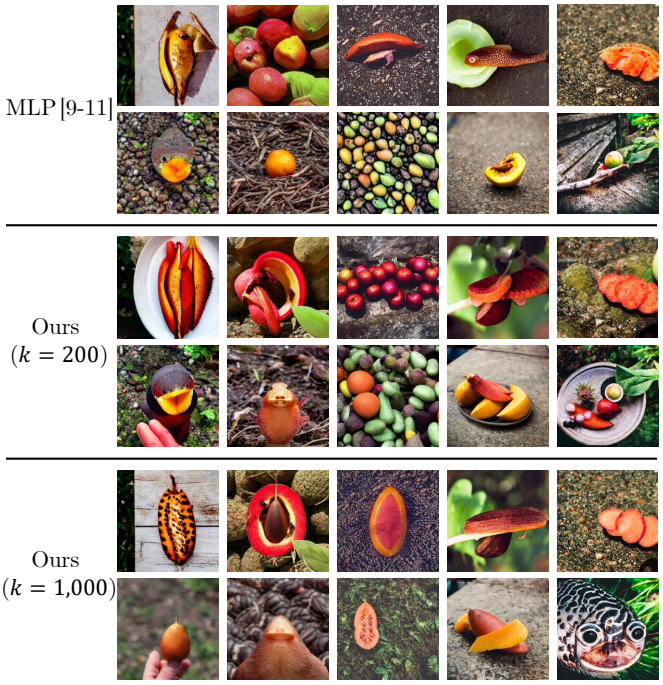


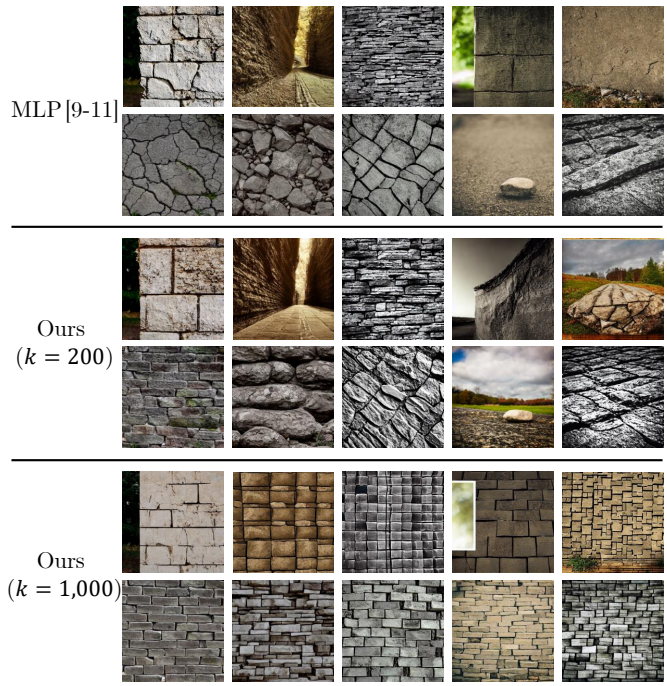
Figure 11: Image generation results generated from interpolated embeddings between Concept A = "*pool*" and Concept B = "*candle*" with an interpolation ratio = 0.5 using different methods.



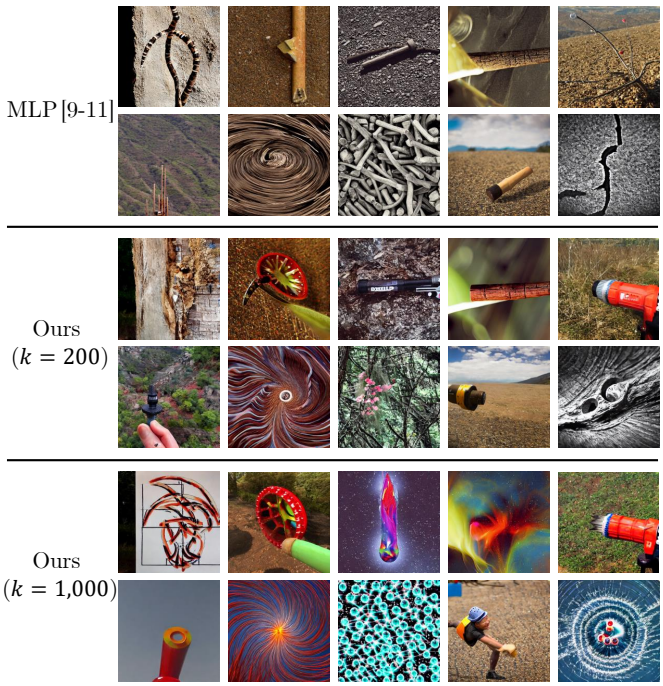
(a) Nonword: "*flike*" (/ˈflaɪk/)



(b) Nonword: "*frout*" (/ˈfraʊt/)



(c) Nonword: "*swoint*" (/ˈswɔɪnt/)



(d) Nonword: "*dwill*" (/ˈdwɪl/)

Figure 12: More nonword-to-image generation results generated using different methods.