Continuous Representations For Efficient Language Models

Vlad Niculae
Language Technology Lab
Informatics Institute, U. of Amsterdam

BamNLP, 2026

Structured representations and optimization in NLP

or, StroopNLP

discrete structure

(Niculae et al., 2018)
(Niculae et al., 2020)
(Correia et al., 2020)
(Niculae et al., 2025)

continuous structure

(Troshin et al., 2023)
(Tokarchuk et al., 2025a)
(Tokarchuk et al., 2026)

contributes to

controllability

(Troshin et al., 2025a)
(Troshin et al., 2025b)

using long contexts

(Mohammed et al., 2024)
(Mohammed et al., 2025)
(Mohammed et al., 2026)

retrieving similar contexts

(Nachesa et al., 2025)
(Tokarchuk et al., 2025b)

Structured representations and optimization in NLP

or, StroopNLP

discrete structure

(Niculae et al., 2018)
(Niculae et al., 2020)
(Correia et al., 2020)
(Niculae et al., 2025)

continuous structure

(Troshin et al., 2023)
(Tokarchuk et al., 2025a)
(Tokarchuk et al., 2026)

contributes to

controllability

(Troshin et al., 2025a)
(Troshin et al., 2025b)

using long contexts

(Mohammed et al., 2024)
(Mohammed et al., 2025)
(Mohammed et al., 2026)

retrieving similar contexts

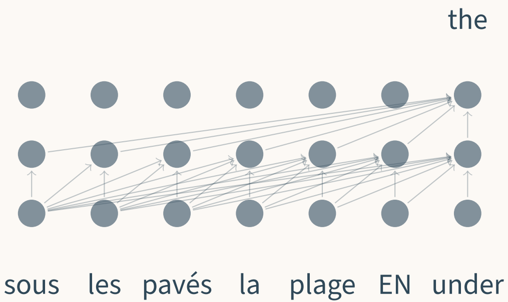
(Nachesa et al., 2025)
(Tokarchuk et al., 2025b)





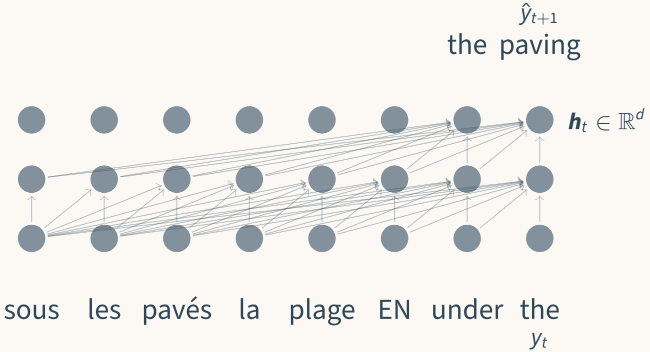

Transformer LM: Next-word prediction

(Vaswani et al., 2017)



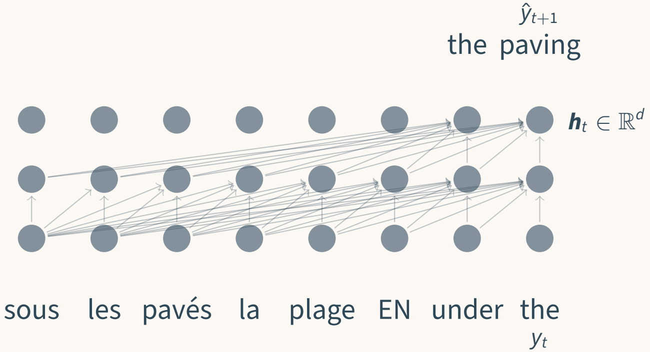
Transformer LM: Next-word prediction

(Vaswani et al., 2017)



Transformer LM: Next-word prediction

(Vaswani et al., 2017)

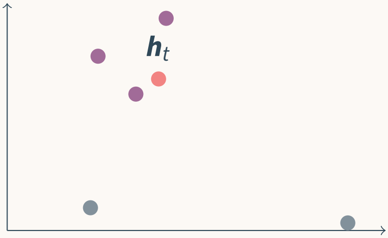


$$\boldsymbol{h}_t = f_\theta(y_1, \ldots, y_t)$$

$\boldsymbol{h}_t$ is a good representation of the entire context $y_{1,\ldots,t}$

$$P(y_{t+1} = \text{paving}) = \frac{\exp\langle \boldsymbol{h}_t, \boldsymbol{w}_{\text{paving}} \rangle}{\sum_{v \in \mathcal{V}} \exp\langle \boldsymbol{h}_t, \boldsymbol{w}_v \rangle}$$

Retrieving contexts by hidden state



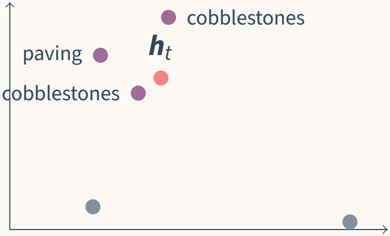
k-nearest neighbors by $d(\boldsymbol{h}_t, \boldsymbol{h})$

Helps understand decisions and mistakes

d	$y_{1,\ldots,t}$	y_{t+1}
.1	Le sable stabilise les pavés autobloquants EN The sand stabilizes the interlocking	**paving**
.8	Ils se promenent sur les pavés pour nostalgie. EN They take a walk on the	**cobblestones**
.9	Ces pavés ont l'air d'avoir été nettoyés EN These	**cobblestones**

k-nearest neighbors language models

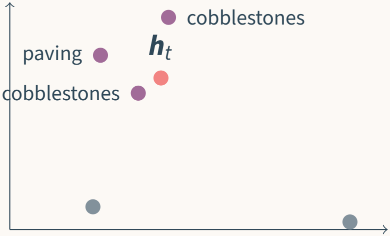
(Khandelwal et al., 2020; Khandelwal et al., 2021)



kNN classifier: predict most voted word from similar contexts.

k-nearest neighbors language models

(Khandelwal et al., 2020; Khandelwal et al., 2021)



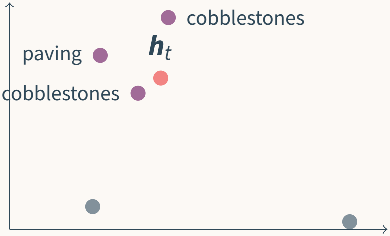
*k*NN classifier: predict most voted word from similar contexts.

kNN-LM, kNN-MT: augmenting transformers with *k*NN

- better accuracy
- domain adaptation
- but, high compute cost

k-nearest neighbors language models

(Khandelwal et al., 2020; Khandelwal et al., 2021)



kNN classifier: predict most voted word from similar contexts.

kNN-LM, kNN-MT: augmenting transformers with kNN

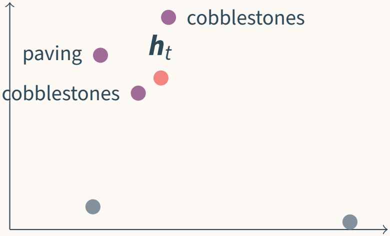
- better accuracy
- domain adaptation
- but, high compute cost

$$p_{\mathrm{LM}}(y_{t+1} = y \mid y_{1:t}) \propto \exp\langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle$$

(linear in $\boldsymbol{h}_t$)

k-nearest neighbors language models

(Khandelwal et al., 2020; Khandelwal et al., 2021)



*k*NN classifier: predict most voted word from similar contexts.

kNN-LM, kNN-MT: augmenting transformers with *k*NN

- better accuracy
- domain adaptation
- but, high compute cost

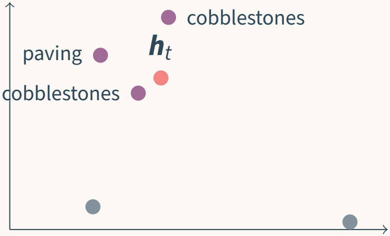
$$p_{\mathrm{LM}}(y_{t+1} = y \mid y_{1:t}) \propto \exp\langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle \qquad \text{(linear in } \boldsymbol{h}_t\text{)}$$

$$p_{\mathrm{kNN}}(y_{t+1} = y \mid y_{1:t}) \propto \sum_{\boldsymbol{h} \in \text{neighbors of } \boldsymbol{h}_t \text{ with label } y} \mathrm{vote}(\boldsymbol{h}, \boldsymbol{h}_t) \qquad \text{(non-linear)}$$

vote can be $\{0, 1\}$ or distance-based; papers use latter

k-nearest neighbors language models

(Khandelwal et al., 2020; Khandelwal et al., 2021)



kNN classifier: predict most voted word from similar contexts.

kNN-LM, kNN-MT: augmenting transformers with kNN

- better accuracy
- domain adaptation
- but, high compute cost

$$p_{\text{LM}}(y_{t+1} = y \mid y_{1:t}) \propto \exp\langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle \qquad \text{(linear in } \boldsymbol{h}_t\text{)}$$

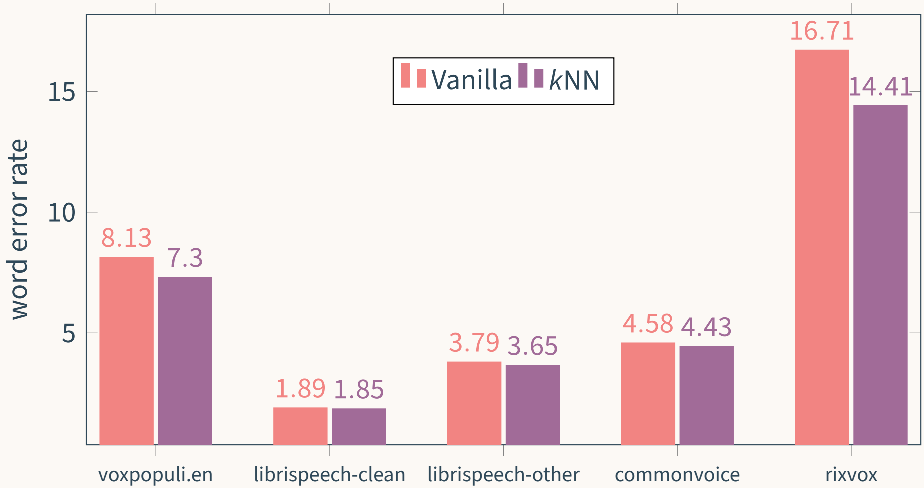
$$p_{\text{kNN}}(y_{t+1} = y \mid y_{1:t}) \propto \sum_{\boldsymbol{h} \in \text{neighbors of } \boldsymbol{h}_t \text{ with label } y} \text{vote}(\boldsymbol{h}, \boldsymbol{h}_t) \qquad \text{(non-linear)}$$

$$p = (1 - \lambda) p_{\text{LM}} + \lambda p_{\text{kNN}}$$

vote can be $\{0, 1\}$ or distance-based; papers use latter

Aside/Bonus: kNN-Whisper for ASR

The Whisper model for ASR is also a conditional LM.
Augmenting it with *k*NN helps! (Nachesa et al., 2025)

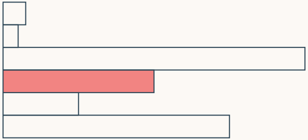


Continuous-output LM

(Kumar et al., 2019)

Standard LM objective: *classification*

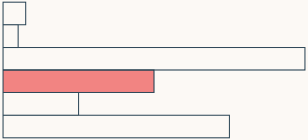
$$\log P(y_{t+1} = y \mid y_{1:t}) = \langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle - \log \sum_{y'} \exp \langle \boldsymbol{h}_t, \boldsymbol{w}_{y'} \rangle$$



Continuous-output LM

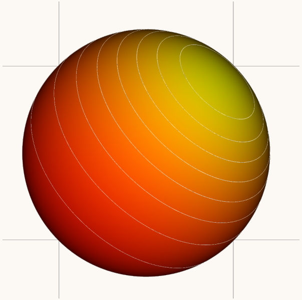
(Kumar et al., 2019)

Standard LM objective: *classification*



$$\log P(y_{t+1} = y \mid y_{1:t}) = \langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle - \log \sum_{y'} \exp \langle \boldsymbol{h}_t, \boldsymbol{w}_{y'} \rangle$$

Continuous LM objective: *regression*



$$\log P(y_{t+1} = \boldsymbol{w}_y \mid y_{1:t}) = \langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle - \log C$$

Langevin probabilistic model on $\mathbb{S}_d$

train: encourage $\boldsymbol{h}_t$ close to $\boldsymbol{w}_y$

test: retrieve nearest-neighbor embeddings

Continuous-output machine translation: embedding choice

(Tokarchuk et al., 2024; Tokarchuk et al., 2026)

model and $\boldsymbol{w}$	ro-en BLEU$_\uparrow$	de-en BLEU$_\uparrow$
discrete	31.7	39.3
continuous, pretrained $\boldsymbol{w}$	29.0	32.9

(Romanian-English WMT16, transformer-base, embedding size 128.)

(English-German WMT19, transformer-big, embedding size 1024.)

(bold best continuous model)

Continuous-output machine translation: embedding choice

(Tokarchuk et al., 2024; Tokarchuk et al., 2026)

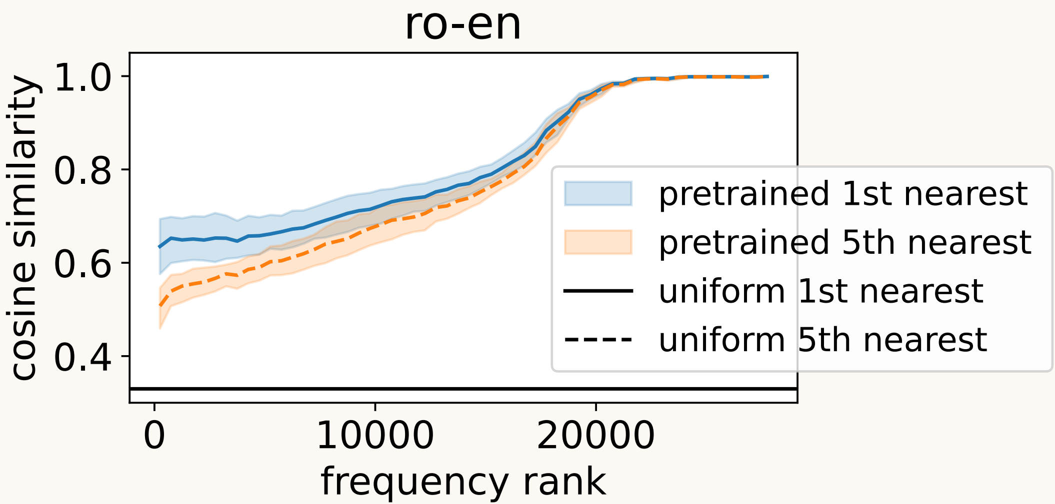
model and ***w***	ro-en BLEU$_\uparrow$	de-en BLEU$_\uparrow$
discrete	31.7	39.3
continuous, pretrained ***w***	29.0	32.9
continuous, random unif ***w***	28.8	33.9

(Romanian-English WMT16, transformer-base, embedding size 128.)

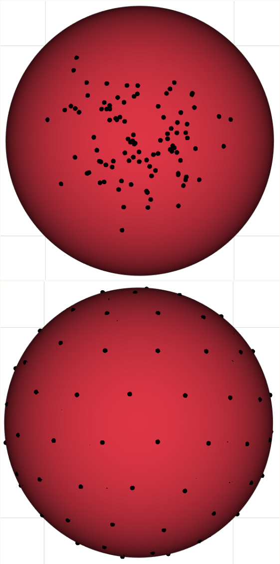
(English-German WMT19, transformer-big, embedding size 1024.)

(bold best continuous model)

Analysis: geometry, distribution by frequency



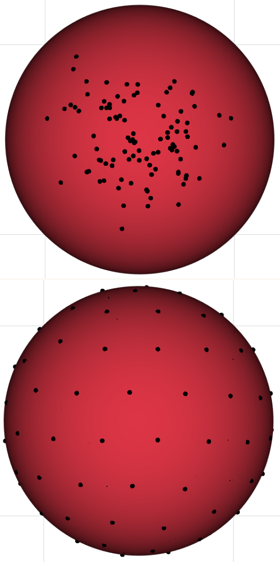
Dispersion on the sphere



Optimal dispersion according to Tammes (1930),

$$\max_{\boldsymbol{w}} \min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Dispersion on the sphere



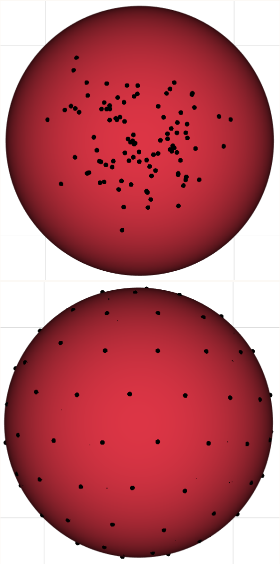
Optimal dispersion according to Tammes (1930),

$$\max_{\boldsymbol{w}} \min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Measures:

- Minimum distance: $\min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$
- Spherical variance: $1 - \|\boldsymbol{\mu}\|$ where $\boldsymbol{\mu} = \sum_i \boldsymbol{w}_i / n$

Dispersion on the sphere



Optimal dispersion according to Tammes (1930),

$$\max_{\boldsymbol{w}} \min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

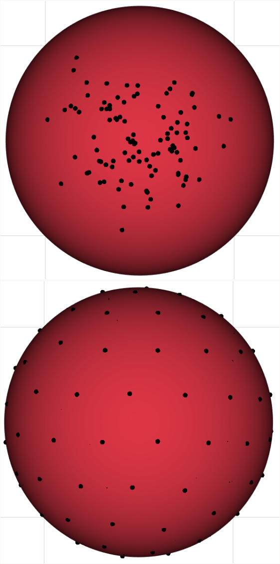
Measures:

- Minimum distance: $\min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$
- Spherical variance: $1 - \|\boldsymbol{\mu}\|$ where $\boldsymbol{\mu} = \sum_i \boldsymbol{w}_i / n$

Related problems:

- Thomson dispersion: electrostatic charges
- Spherical codes (quantizing the sphere)
- Sphere packing / the kissing problem

Dispersion on the sphere



Optimal dispersion according to Tammes (1930),

$$\max_{\boldsymbol{W}} \min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Measures:

- Minimum distance: $\min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$
- Spherical variance: $1 - \|\boldsymbol{\mu}\|$ where $\boldsymbol{\mu} = \sum_i \boldsymbol{w}_i / n$

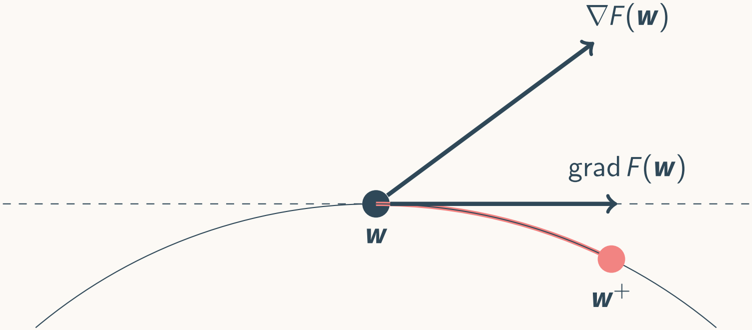
Related problems:

- Thomson dispersion: electrostatic charges
- Spherical codes (quantizing the sphere)
- Sphere packing / the kissing problem

Despite symmetry, exact solution generally unknown.
We want to trade off dispersion with a *task loss*:

$$L(\boldsymbol{W}) + \alpha R(\boldsymbol{W})$$

Riemannian optimization on $\mathbb{S}_m$



Scary math and notation but simple intuition:
walk along the surface in the direction of the gradient.

$$\boldsymbol{w}^+ = \operatorname{Exp}_{\boldsymbol{w}}(\operatorname{grad} F(\boldsymbol{w}))$$

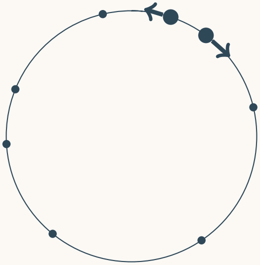
Useful measures are not necessarily optimization-friendly

min. dist. (Tammes objective)

$$R_{\text{Tammes}}(\boldsymbol{W}) = -\min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

$$\operatorname{grad}_{\boldsymbol{w}_k} R_{\text{Tammes}}(\boldsymbol{W}) = 0$$

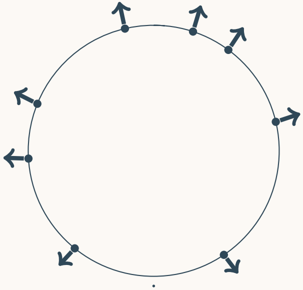
for almost all k except the two closest ones.



spherical variance

$$R_{\text{var}}(\boldsymbol{W}) = 1 - \left\| \sum_k \boldsymbol{w}_k / n \right\|$$

Eucl. gradients are normal;
so all Riemannian gradients are 0.



Common approaches in literature

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\mathrm{MM}}(\boldsymbol{W}) = -\frac{1}{n} \sum_{i} \min_{j \neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\mathrm{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n} \sum_{i} \log \min_{j \neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

$$R_{\mathrm{MHE},k} = \frac{1}{n(n-1)} \sum_{i \neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Common approaches in literature

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\mathrm{MM}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\mathrm{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \log \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

$$R_{\mathrm{MHE},k} = \frac{1}{n(n-1)}\sum_{i\neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Common approaches in literature

$\boxed{\text{Quadratic complexity } O(mn^2)}$

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\mathrm{MM}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\mathrm{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \log \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

$$R_{\mathrm{MHE},k} = \frac{1}{n(n-1)}\sum_{i\neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Common approaches in literature

$\boxed{\text{Quadratic complexity } O(mn^2)}$

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\text{MM}}(\boldsymbol{W}) = -\frac{1}{n}\sum_{i} \min_{j \neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\text{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n}\sum_{i} \log \min_{j \neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

We show: $R_{\text{Tammes}} \leq R_{\text{MM}} \leq R_{\text{KoLeo}} - 1$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

$$R_{\text{MHE},k} = \frac{1}{n(n-1)}\sum_{i \neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Common approaches in literature

$\boxed{\text{Quadratic complexity } O(mn^2)}$

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\mathrm{MM}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \min_{j \neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\mathrm{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \log \min_{j \neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

We show: $R_{\mathrm{Tammes}} \leq R_{\mathrm{MM}} \leq R_{\mathrm{KoLeo}} - 1$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

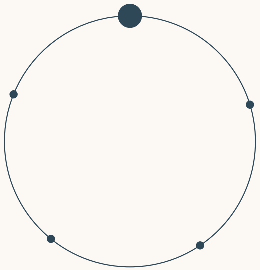
$$R_{\mathrm{MHE},k} = \frac{1}{n(n-1)}\sum_{i \neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

We show: $R_{\mathrm{MHE},k}$ is the maximum mean discrepancy between the empirical measure $\boldsymbol{W}$ and the uniform measure on the sphere.

Sliced dispersion

(Bonet et al., 2023; Tokarchuk et al., 2025a)

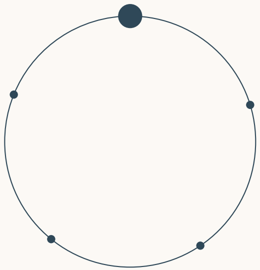
Observation: on $\mathbb{S}_1$, optimally dispersed configurations are some rotation of:



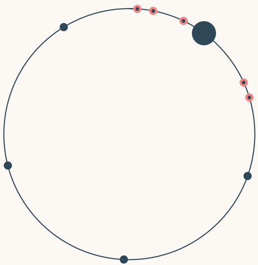
Sliced dispersion

(Bonet et al., 2023; Tokarchuk et al., 2025a)

Observation: on $\mathbb{S}_1$, optimally dispersed configurations are some rotation of:



Given some suboptimal configuration, we can define its distance to the closest dispersed one:



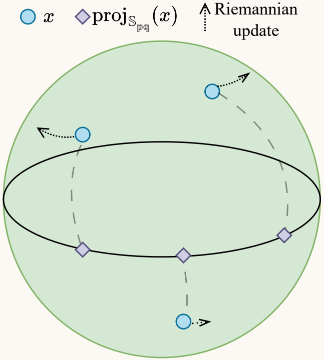
Intuition: sort angles;
map 1-to-1 around mean.
Computation: $O(n + \text{sort}(n))$).

Sliced dispersion on $\mathbb{S}_m$

Slicing along any great circle should preserve dispersion on average.

R_{sliced}:
expectation over great circles (p, q)
of distance to optimal 1d configuration

Complexity:
$O(mn)$ to project to great circle
($O(n)$ if axis-aligned)
$+O(\text{sort}(n))$ to disperse.



Continuous-output machine translation: embedding choice

(Tokarchuk et al., 2024; Tokarchuk et al., 2026)

model and $\boldsymbol{w}$	ro-en BLEU$_\uparrow$	de-en BLEU$_\uparrow$
discrete	31.7	39.3
continuous, pretrained $\boldsymbol{w}$	29.0	32.9
continuous, random unif $\boldsymbol{w}$	28.8	33.9
continuous, dispersed	**30.1**	**36.6**

(Romanian-English WMT16, transformer-base, embedding size 128.)

(English-German WMT19, transformer-big, embedding size 1024.)

(bold best continuous model)

Continuous-output machine translation: embedding choice

(Tokarchuk et al., 2024; Tokarchuk et al., 2026)

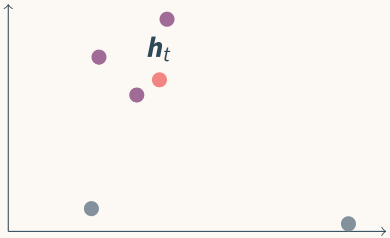
model and $\boldsymbol{w}$	ro-en BLEU$_\uparrow$	de-en BLEU$_\uparrow$
discrete	31.7	39.3
discrete dispersed	32.4	39.2
continuous, pretrained $\boldsymbol{w}$	29.0	32.9
continuous, random unif $\boldsymbol{w}$	28.8	33.9
continuous, dispersed	**30.1**	**36.6**

(Romanian-English WMT16, transformer-base, embedding size 128.)

(English-German WMT19, transformer-big, embedding size 1024.)

(bold best continuous model)

From CoNMT to kNN-MT



	CoNMT	kNN-MT
	Both: retrieve next word through lookup of nearest key vector	
keys	word embeddings on $\mathbb{S}_m$	context representations on $\mathbb{R}^m$
$n \approx$	10^4	10^5–10^7

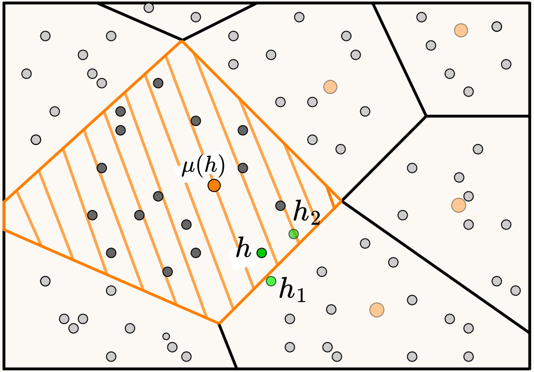
Due to higher n, we must use efficient approximate nearest neighbors methods.

Approximate nearest neighbor retrieval

Inverse vector file + product quantization (IVFPQ, Johnson et al., 2019): a SOTA method

IVF: cluster the keys using *k*-means; treat each Voronoi cell as a separate smaller (centered) data store.

PQ: inside each cell, split the *m* dimensions into subspaces and quantize them to 8 bit per key.



Approximate nearest neighbor retrieval

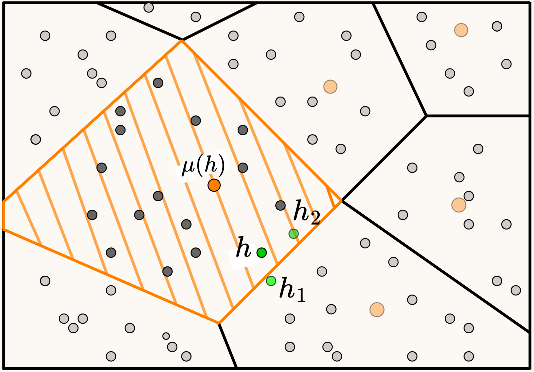
Inverse vector file + product quantization (IVFPQ, Johnson et al., 2019): a SOTA method

IVF: cluster the keys using *k*-means; treat each Voronoi cell as a separate smaller (centered) data store.

PQ: inside each cell, split the *m* dimensions into subspaces and quantize them to 8 bit per key.

At lookup time:

- find the n_{probes} closest centroids
- search exhaustively within their Voronoi cells.



Approximate nearest neighbor retrieval

Inverse vector file + product quantization (IVFPQ, Johnson et al., 2019): a SOTA method

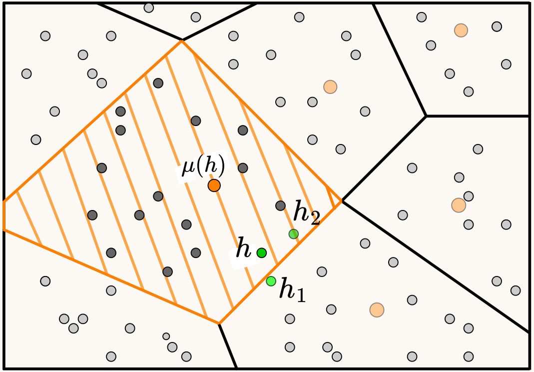
IVF: cluster the keys using *k*-means; treat each Voronoi cell as a separate smaller (centered) data store.

PQ: inside each cell, split the *m* dimensions into subspaces and quantize them to 8 bit per key.

At lookup time:

- find the n_{probes} closest centroids
- search exhaustively within their Voronoi cells.

Hypothesis: IVFPQ performs better with dispersed keys.



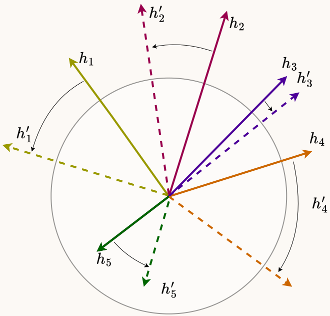
Angular dispersion and isotropy

(Tokarchuk et al., 2025b)

Given a set of hidden states $\boldsymbol{h} \in \mathbb{R}^m$, consider the dispersion of their *directions*

$$\boldsymbol{h}/\|\boldsymbol{h}\| \in \mathbb{S}_m.$$

Intuition: Distribution of $\boldsymbol{h}$ spherically symmetric around origin implies angular dispersion.



Synthetic validation

Generative process.
Draw $n = 10M$ directions from mixture of 5 Power Sphericals on $\mathbb{S}_{128}$, with varying concentration.

Assign to each direction a uniform length on $[1, 100]$

Synthetic validation

Generative process.

Draw $n = 10M$ directions from mixture of 5 Power Sphericals on $\mathbb{S}_{128}$, with varying concentration.

Assign to each direction a uniform length on $[1, 100]$

Evaluation.

Fit IVFPQ datastore with 2048 cells, 8 neighbors, batch size 10, $n_{\text{probes}} = 32$, and measure over $10K$ random queries:

- imbalance factor $IF = K \sum_{i=1}^{K} \left(\frac{n_i}{n}\right)^2$
- throughput (requests per second)

Synthetic validation

Generative process.

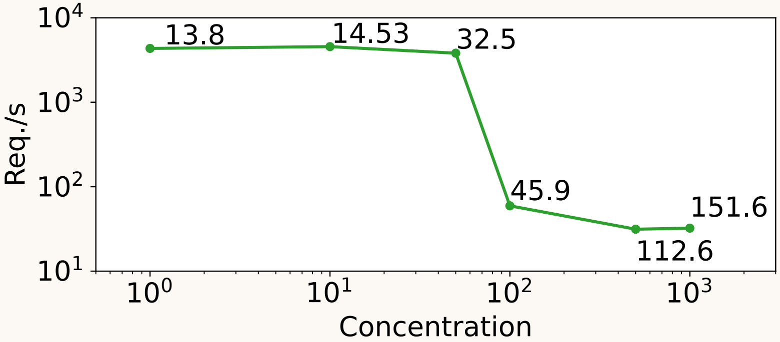
Draw $n = 10M$ directions from mixture of 5 Power Sphericals on $\mathbb{S}_{128}$, with varying concentration.

Assign to each direction a uniform length on $[1, 100]$

Evaluation.

Fit IVFPQ datastore with 2048 cells, 8 neighbors, batch size 10, $n_{\text{probes}} = 32$, and measure over $10K$ random queries:

- imbalance factor $IF = K \sum_{i=1}^{K} \left(\frac{n_i}{n}\right)^2$
- throughput (requests per second)



Can we make hidden states dispersed?

Not parameters, but network outputs:
$\boldsymbol{h}_t = f_\theta(y_1, \ldots, y_t)$

Transformers are trained on minibatches,
we don't see all contexts at once.

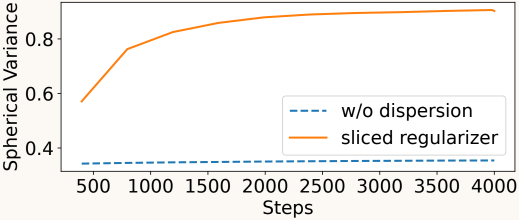
Doubly-stochastic sliced dispersion:
Each training update disperses a subset of
the collection along a random great circle.

Can we make hidden states dispersed?

Not parameters, but network outputs:
$\boldsymbol{h}_t = f_\theta(y_1, \dots, y_t)$

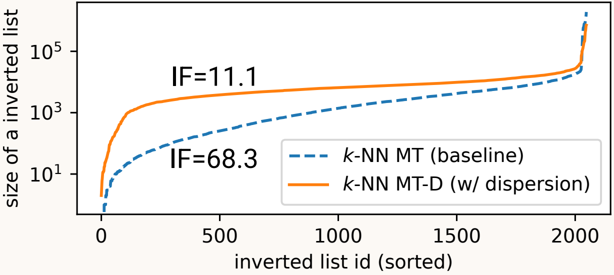
Transformers are trained on minibatches, we don't see all contexts at once.

Doubly-stochastic sliced dispersion:
Each training update disperses a subset of the collection along a random great circle.



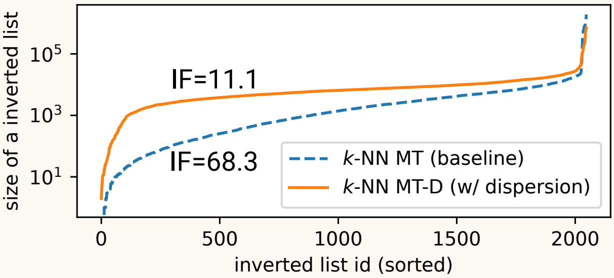
Dispersed kNN-MT

Romanian-English WMT16, transformer-base, $m = 128$.



Dispersed kNN-MT

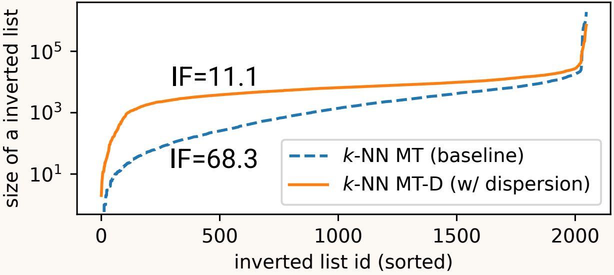
Romanian-English WMT16, transformer-base, $m = 128$.



model	#probes	BLEU$_{(\uparrow)}$	COMET$_{(\uparrow)}$	tok/s$_{(\uparrow)}$
baseline	-	31.5	78.95	75
*k*NN MT	32	32.4	79.89	12
*k*NN MT-D	32	32.6	79.91	53

Dispersed kNN-MT

Romanian-English WMT16, transformer-base, $m = 128$.

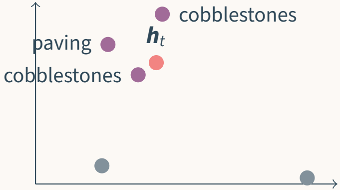


model	#probes	BLEU$_{(\uparrow)}$	COMET$_{(\uparrow)}$	tok/s$_{(\uparrow)}$
baseline	-	31.5	78.95	75
*k*NN MT	32	32.4	79.89	12
*k*NN MT	8	32.2	79.69	28
*k*NN MT-D	32	32.6	79.91	53
*k*NN MT-D	8	**32.6**	**79.93**	**63**

Continuous representations for efficient language models

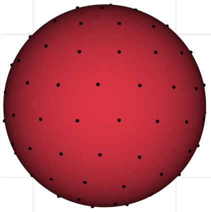
***k*NN language models:**

a powerful model, more adaptable, more interpretable (→ also for speech recognition).



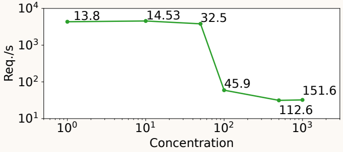
dispersion:

a centuries-old problem still relevant in modern ML.



substantial improvements:

speedups and accuracy boost thanks to going back to basics.



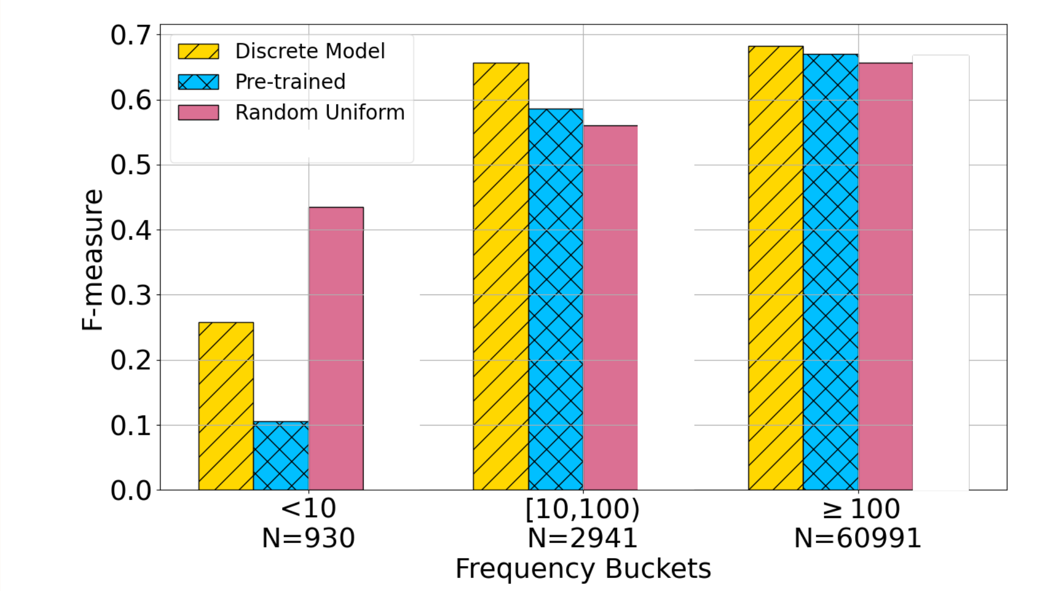
Immediate perspectives:

- Seek alternatives to IVFPQ built directly for dispersion.
- Explore the spectrum between CoNMT and *k*NN-MT.

Long-term perspectives: Geometry of representations from single tokens to phrases; "reasoning" over such structured / searchable spaces.

extra

Analysis: errors by frequency



References I

Bonet, Clément et al. (2023). "Spherical Sliced-Wasserstein". In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=jXQ0ipgMdU.

Correia, Gonçalo et al. (2020). "Efficient marginalization of discrete and structured latent variables via sparsity". In: *Advances in Neural Information Processing Systems*.

Gautam, Simanta and Dmitry Vaintrob (2013). "A Novel Approach to the Spherical Codes Problem". In: *MIT, Cambridge, MA, USA, Tech. Rep*. URL: https://api.semanticscholar.org/CorpusID:12647839.

Johnson, Jeff, Matthijs Douze, and Hervé Jégou (2019). "Billion-scale similarity search with GPUs". In: *IEEE Transactions on Big Data* 7.3, pp. 535–547.

Khandelwal, Urvashi et al. (2020). "Generalization through Memorization: Nearest Neighbor Language Models". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HklBjCEKvH.

Khandelwal, Urvashi et al. (2021). "Nearest Neighbor Machine Translation". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=7wCBOfJ8hJM.

Kumar, Sachin and Yulia Tsvetkov (2019). "von Mises-Fisher Loss for Training Sequence to Sequence Models with Continuous Outputs". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rJlDnoA5Y7.

Leonenko, Nikolai N (1987). "Sample estimate of the entropy of a random vector". In: *Problemy Peredachi Informatsii* 23.2, pp. 9–16.

References II

Liu, Weiyang et al. (2018). "Learning towards Minimum Hyperspherical Energy". In: *Neural Information Processing Systems*. URL: https://api.semanticscholar.org/CorpusID:43921092.

Liu, Weiyang et al. (13–15 Apr 2021). "Learning with Hyperspherical Uniformity". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 1180–1188. URL: https://proceedings.mlr.press/v130/liu21d.html.

Mettes, Pascal, Elise van der Pol, and Cees G M Snoek (2019). "Hyperspherical Prototype Networks". In: *Advances in Neural Information Processing Systems*.

Mohammed, Wafaa and Vlad Niculae (2025). "Context-Aware or Context-Insensitive? Assessing LLMs' Performance in Document-Level Translation". In: *Proceedings of Machine Translation Summit XX: Volume 1*. ISBN: 978-2-9701897-0-1. URL: https://aclanthology.org/2025.mtsummit-1.10/.

— (2024). "On Measuring Context Utilization in Document-Level MT Systems". In: *Findings of the ACL: EACL 2024*. URL: https://aclanthology.org/2024.findings-eacl.113/.

Mohammed, Wafaa, Vlad Niculae, and Chrysoula Zerva (2026). "Unlocking Latent Discourse Translation in LLMs Through Quality-Aware Decoding". In: *19th Conference of the European Chapter of the Association for Computational Linguistics*. URL: https://openreview.net/forum?id=mbnU8WeG0b.

References III

Nachesa, Maya K. and Vlad Niculae (Apr. 2025). “kNN For Whisper And Its Effect On Bias And Speaker Adaptation”. In: *Findings of the Association for Computational Linguistics: NAACL 2025*. Ed. by Luis Chiruzzo, Alan Ritter, and Lu Wang. Albuquerque, New Mexico: Association for Computational Linguistics, pp. 6621–6627. ISBN: 979-8-89176-195-7. DOI: 10.18653/v1/2025.findings-naacl.369. URL: https://aclanthology.org/2025.findings-naacl.369/.

Niculae, Vlad and André FT Martins (2020). “LP-SparseMAP: Differentiable relaxed optimization for sparse structured prediction.”. In: *Proc. ICML*.

Niculae, Vlad et al. (2025). “Discrete Latent Structure in Neural Networks”. In: *Foundations and Trends® in Signal Processing* 19.2, pp. 99–211. ISSN: 1932-8346. DOI: 10.1561/2000000134. URL: http://dx.doi.org/10.1561/2000000134.

Niculae, Vlad et al. (2018). “SparseMAP: Differentiable sparse structured inference”. In: *Proc. ICML*.

Sablayrolles, Alexandre et al. (2019). “Spreading vectors for similarity search”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=SkGuG2R5tm.

Tammes, Pieter Merkus Lambertus (1930). “On the origin of number and arrangement of the places of exit on the surface of pollen-grains”. English. Relation: http://www.rug.nl/ Rights: De Bussy. PhD thesis. University of Groningen.

References IV

Thomson, JJ (1904). “On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 7.39, pp. 237–265. DOI: 10.1080/14786440409463107. eprint: https://doi.org/10.1080/14786440409463107. URL: https://doi.org/10.1080/14786440409463107.

Tokarchuk, Evgeniia, Hua Chang Bakker, and Vlad Niculae (2025a). “Keep your distance: learning dispersed embeddings on $\mathbb{S}_m$”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: https://openreview.net/forum?id=5JIQE6HcTd.

Tokarchuk, Evgeniia and Vlad Niculae (2024). “The Unreasonable Effectiveness of Random Target Embeddings for Continuous-Output Neural Machine Translation”. In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. DOI: 10.18653/v1/2024.naacl-short.56. URL: https://aclanthology.org/2024.naacl-short.56/.

Tokarchuk, Evgeniia, Sergey Troshin, and Vlad Niculae (2025b). “Angular Dispersion Accelerates *k*-Nearest Neighbors Machine Translation”. In: *Findings of the ACL: Empirical Methods in Natural Language Processing*. URL: https://openreview.net/forum?id=6lJWxycZTa.

Tokarchuk, Evgeniia et al. (2026). “Representation Collapse in Machine Translation Through the Lens of Angular Dispersion”. In: *Findings of the ACL: 19th Conference of the European Chapter of the Association for Computational Linguistics*.

Troshin, Sergey and Vlad Niculae (2023). “Wrapped β-gaussians with compact support for exact probabilistic modeling on manifolds”. In: *Transactions on Machine Learning Research*.

References V

Troshin, Sergey, Vlad Niculae, and Antske Fokkens (2025a). "On the Low-Rank Parametrization of Reward Models for Controlled Language Generation". In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: https://openreview.net/forum?id=cjRsEGLT8B.

Troshin, Sergey et al. (2025b). "Control the Temperature: Selective Sampling for Diverse and High-Quality LLM Outputs". In: *Second Conference on Language Modeling*.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems*. Vol. 30.

Wang, Tongzhou and Phillip Isola (13–18 Jul 2020). "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 9929–9939. URL: https://proceedings.mlr.press/v119/wang20k.html.

Wang, Zhennan et al. (2021). *MMA Regularization: Decorrelating Weights of Neural Networks by Maximizing the Minimal Angles*. arXiv: 2006.06527 [cs.LG].