Dynamic Meta-Metrics: Source-Sentence Conditioned Weighting for MT Evaluation

Anonymous ACL submission

Abstract

We propose Dynamic Meta-Metrics (DMM), a framework for machine translation evaluation that learns source-sentence conditioned combinations of existing metrics. Rather than relying on a single static ensemble or language-specific weighting, DMM adapts the metric combination based on properties of the source segment. We study hard conditioning, which fits an interpretable combiner per cluster, and a soft-conditioned extension based on a gated mixture of linear experts. We evaluate DMM on the WMT Metrics Shared Task data across multiple language pairs using pairwise agreement measures at the system and segment levels. Across both average system-level SPA and average segment-level agreement, our DMM variants generally outperform single-metric baselines and often MetaMetrics.

1 Introduction

Automatic evaluation metrics underpin machine translation (MT) research and deployment. An effective metric should align with human judgements, generalise across languages and domains, and remain reproducible. The WMT Metrics Shared Task provides a standardised setting for meta-evaluation and has shown that metric behaviour varies with language, domain, and input characteristics (Freitag et al., 2022; Kocmi et al., 2023; Freitag et al., 2024). In particular, no single metric consistently dominates across years and test conditions, even among strong neural and generative metrics (Juraska et al., 2023, 2024; Freitag et al., 2024).

To account for distribution shift, prior work has explored combining multiple metrics into static ensembles, which learn fixed weights and can be more robust than relying on a single metric. However, they do not model variation within a language pair, where differences in syntax, discourse style, or domain can affect metric reliability.

To this end, we introduce *Dynamic Meta-Metrics* (DMM), a framework that conditions metric-combination behaviour on source-sentence context. DMM uses sentence embeddings and clustering to construct a discrete context variable and learns context-specific combinations of base metric scores by embedding each source sentence with LaBSE (Feng et al., 2022) and applying k-means clustering.We term this **hard conditioning**. This preserves interpretability (explicit weights per context) while allowing metric contributions to vary with the input. We study a family of meta-metric models: (i) a linear regressor (OLS) minimising MSE on segment-level human scores, (ii) a small MLP trained with MSE, and (iii) a Gaussian-process-driven Bayesian optimiser that maximises Kendall correlation to human scores.

We also define a soft-conditioned extension (**soft conditioning**) that replaces discrete clustering with assigning *responsibilities* (analogous to soft k-means) through a gated mixture of linear experts (Shazeer et al., 2017; Jacobs et al., 1991). We emphasise *linear* experts since they preserve the interpretation of each expert as an explicit set of metric weights, supporting the analysis of context-dependent preferences.

We evaluate English–Chinese (en-zh), English–Czech (en-cs), English–Japanese (en-ja), and English–Ukrainian (en-uk) using WMT21–24 segment-level annotations (Freitag et al., 2024) as training and validation data, and WMT25 shared task data for test data.

2 Related Work

2.1 Single metric evaluation

Reference-based overlap metrics such as BLEU (Papineni et al., 2002) and chrF (Popović, 2015, 2017) are commonly used in WMT Shared Tasks due to their efficiency and historical usage. However, their emphasis on lexical overlap means that legitimate semantic or syntactic variations can be

under-rewarded, leading to poorer correlation with human evaluation (Freitag et al., 2022).

Learned metrics, including BLEURT (Sellam et al., 2020; Pu et al., 2021) and COMET (Rei et al., 2020), improve correlation with human judgments by using pretrained encoders and supervised calibration to human ratings. YiSi-1 (Lo, 2019) provides an embedding-based similarity framework that can support both lower and higher-resource settings. More recently, neural generative models such as MetricX-23/24 (Kocmi et al., 2023; Freitag et al., 2022) became state-of-the-art in WMT22/23, showing stronger and more consistent results across many domains (though not all). However, no single metric dominates all domains. Freitag et al. (2024) showed that LLM-era MT outputs expose brittleness in some metrics and underscored the value of robust calibration to MQM/ESA.

2.2 Static ensembles via meta metrics

MetaMetrics-MT (MM) (Anugraha et al., 2024) learns static weights, typically per language pair, and optimises correlation with human scores, often outperforming individual metrics. However, weights are typically trained per-language and optimised against a single objective, leaving room to adapt within-language variation such as domain or segment properties. DMM addresses this limitation by conditioning on source-derived contexts.

2.3 Mixture-of-experts and soft routing

Mixture-of-experts models combine predictors using a gating function that selects or mixes experts (Jacobs et al., 1991). Modern routing methods extend this idea with sparse or structured gating networks (Shazeer et al., 2017). In the **soft conditioning** case, DMM uses a constrained form of this approach, with linear experts that remain interpretable as metric weight vectors and with gating regularised to avoid collapsing to one-hot hard conditioning when it is not appropriate for the data.

3 Method

3.1 Problem setup and notation

Each training instance is a triple (s, t, y), where s denotes a source segment, t a system output (hypothesis) for that segment, and $y \in \mathbb{R}$ a human segment-level score. We index instances by $i \in \{1, \ldots, n\}$. For each instance i, we compute d base metric scores and form a feature vector

$$\mathbf{x}_i = \big(m_1(i), \ldots, m_d(i)\big) \in \mathbb{R}^d.$$

A meta-metric is a function $F_\theta : \mathbb{R}^d \to \mathbb{R}$ that predicts $\hat{y}_i = F_\theta(\mathbf{x}_i)$, with learnable parameters θ.

3.2 Feature standardisation

Since metric outputs differ in scale, we z-standardise each metric feature using training-set statistics. For metric j, let μ_j and σ_j be the mean and standard deviation over training instances. We standardise x_{ik}, obtaining $\tilde{x}_{ik}$ forming $\tilde{\mathbf{x}}_i = (\tilde{x}_{i1}, \ldots, \tilde{x}_{id})$. All subsequent linear and neural models use $\tilde{\mathbf{x}}_i$.

3.3 Embedding and clustering source segments

We embed each source segment s using LaBSE (Feng et al., 2022), yielding $\mathbf{e}(s) \in \mathbb{R}^m$. We fit k-means on training-set source embeddings to obtain K centroids $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$. For any segment s, we define squared distances $D_j(s) = \|\mathbf{e}(s) - \mathbf{c}_j\|_2^2$.

3.4 Hard-conditioned DMM

We consider four configurations of ensemble techniques: 1) **Language-pooled MM:** a single model trained on all instances across all language pairs. This corresponds to a language-pooled variant of MM. 2) **Language-specific MM:** one model trained per language pair. This corresponds to the language-specific variant of MM. 3) **Cluster:** one model trained per hard cluster label $\arg\min_j D_j(s)$, pooling data across languages within each cluster. This is the language-pooled variant of the proposed DMM. 4) **Language–cluster:** one model trained per (language, cluster) pair. This is the language-specific variant of DMM.

For a given configuration, each instance is evaluated using the model corresponding to its language, its cluster assignment, or both. Predictions are given by

$$\hat{y}_i = F_\theta(\tilde{\mathbf{x}}_i),$$

where θ denotes the parameters of the model selected by the active conditioning regime.

3.5 Soft-conditioned DMM

Hard conditioning yields piecewise-constant behaviour with respect to cluster assignments: each segment is evaluated using a single cluster-specific model. This is simple and interpretable, but lacks continuity when a source segment is not well represented by a single cluster. Soft-conditioned DMM replaces discrete cluster selection with a continuous mixture over cluster-conditioned experts.

We first define *responsibilities* (in the soft k-NN sense), which quantify how much a segment belongs to each cluster. For a segment s, the responsibility with temperature $T > 0$ is $\mathbf{r}(s) = \mathrm{softmax}(\{-D_k(s)/T\}_{k=1}^{K})$. Responsibilities depend only on the source segment and are shared across all system outputs for that segment. Using these responsibilities, we define K linear "expert" models. Each expert k is a linear meta-metric, $f_k(\tilde{\mathbf{x}}_i) = \mathbf{w}_k^\top \tilde{\mathbf{x}}_i + b_k$.

Soft-conditioned DMM allows all K experts to contribute to the prediction $\hat{y}$. We define *gate features* $\mathbf{z}_i$ from the source segment (sentence length and responsibilities). A gating network $g_\phi(\mathbf{z}_i) \in \mathbb{R}^K$ produces logits that emphasise some experts over others. We incorporate responsibilities as a log-prior weighted by $\lambda \geq 0$ and apply gate temperature $\vartheta > 0$:

$$\boldsymbol{\ell}_i = g_\phi(\mathbf{z}_i) + \lambda \log \mathbf{r}(s_i), \quad \boldsymbol{\alpha}_i = \mathrm{softmax}(\boldsymbol{\ell}_i/\vartheta).$$

The final prediction is $\hat{y}_i = \sum_{k=1}^{K} \alpha_{ik} f_k(\tilde{\mathbf{x}}_i)$. As $T \to 0^+$, responsibilities become one-hot, and the model approaches hard-conditioning.

We use a mixture of *linear* experts to preserve interpretability: each expert corresponds to explicit metric weights. In particular, without nonlinearities, one can write $\mathbf{w}(\mathbf{z}) = \sum_k \alpha_k(\mathbf{z}) \mathbf{w}_k$, which is not possible for nonlinear experts (e.g., neural networks). Thus, linear experts isolate the effect of routing from expert expressivity while retaining the interpretation of soft-conditioning as a weighted/dynamic meta-metric.

3.6 Training objectives and regularisation

Hard-conditioned models Within each ensemble configuration, we train one or more models using: 1) **OLS:** linear regression minimising mean squared error (MSE), 2) **MLP:** a two-layer network trained with MSE as a flexible baseline, and 3) **GP optimization:** Bayesian optimisation over constrained linear weights, following METAMETRICS-MT (Anugraha et al., 2024).

Soft-conditioned model Soft-conditioned DMM trains expert parameters $\{\mathbf{w}_k, b_k\}_{k=1}^{K}$ and gate parameters ϕ by minimising $\mathcal{L}_{\mathrm{mse}} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$. To discourage degenerate routing (and to distinguish it from hard-conditioning), we add an entropy term over the gating distribution: $\mathcal{L}_{\mathrm{ent}} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \alpha_{ik} \log(\alpha_{ik})$. We also add an expert-tying penalty to keep experts close to their mean when the data do not support specialisation: $\mathcal{L}_{\mathrm{tie}} = \sum_{k=1}^{K} \|\mathbf{w}_k - \bar{\mathbf{w}}\|_2^2 + \sum_{k=1}^{K} (b_k - \bar{b})^2$, where $\bar{\mathbf{w}} = \frac{1}{K} \sum_k \mathbf{w}_k$ and $\bar{b} = \frac{1}{K} \sum_k b_k$. The final objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{mse}} - \beta \mathcal{L}_{\mathrm{ent}} + \gamma \mathcal{L}_{\mathrm{tie}},$$

with $\beta, \gamma \geq 0$. The sign on the entropy term encourages higher-entropy routing.

3.7 Evaluation measures

Following WMT24 (Freitag et al., 2024), we evaluate metrics primarily using pairwise agreement with human judgements. At the system level, we report soft pairwise accuracy (SPA) (Thompson et al., 2024), which accounts for uncertainty in both human and metric-induced rankings. At the segment level, we report group-by-item pairwise accuracy with tie calibration (acc^*_{eq}) (Deutsch et al., 2023), reflecting the metric's ability to rank alternative translations of the same source segment correctly.

4 Experiments

4.1 Experimental Setup

We use WMT Metrics Shared Task data from 2021–2024 (Freitag et al., 2024) for training and validation, including en-cs, en-zh, en-uk, and en-ja. For each pair and year, we use source segments, system outputs, reference translations, and official segment-level human annotations, excluding segments without human scores. We use a fixed, deterministic split defined at the segment level within each year (approximately 80% train, 20% validation), created independently per year. We evaluate on held-out WMT25 shared task data.

We compute a representative set of reference-based metrics spanning overlap, embedding-based, and generative families, using default toolkit settings unless stated otherwise. We exclude the largest metric variants (e.g., MetricX-XXL) to support execution on typical hardware. We consider SacreBLEU, BLEU, and sentence-level BLEU variants (Papineni et al., 2002; Post, 2018), chrF++ (Popović, 2017), BLEURT-20 (Sellam et al., 2020; Pu et al., 2021), YiSi-1 (Lo, 2019), COMET, and XCOMET-XL (Rei et al., 2020), and MetricX-23/24 Large and XL (Juraska et al., 2023, 2024).

For model fitting, we embed source segments with LaBSE (Feng et al., 2022) and fit k-means on training sources. We evaluate $K \in \{2, 3, 4, 5, 6, 7\}$. Implementation details, hyperpa-

System	Avg		EN-CS		EN-ZH		EN-JA		EN-UK	
	SPA	acc^*_{eq}	**SPA**	acc^*_{eq}	**SPA**	acc^*_{eq}	**SPA**	acc^*_{eq}	**SPA**	acc^*_{eq}
BLEURT	0.661	0.673	0.706	0.694	0.604	0.662	0.561	0.652	**0.773**	0.683
XCOMET-XL	0.648	0.674	0.676	0.668	0.626	0.684	0.600	0.647	0.691	0.697
Language-pooled MM										
OLS	0.633	0.668	0.687	0.682	0.576	0.640	0.534	**0.657**	0.733	0.695
MLP	0.696	0.679	0.747	0.686	0.654	0.689	0.643	0.649	0.740	0.692
GP	0.689	0.682	0.741	0.692	0.639	0.674	0.630	0.655	0.748	**0.707**
MoE	0.626	0.663	0.666	0.690	0.646	0.676	0.492	0.618	0.699	0.668
Language-separate MM										
OLS	0.619	0.659	0.672	0.683	0.577	0.609	0.522	0.638	0.706	0.705
MLP	0.695	0.678	0.760	0.687	0.650	0.682	0.615	0.649	0.753	0.695
GP	0.660	0.665	0.687	0.664	0.638	0.656	0.561	0.649	0.755	0.691
MoE	0.599	0.655	0.641	0.680	0.584	0.630	0.485	0.609	0.687	0.703
DMM Clusters $k = 6$										
OLS	0.623	0.676	0.677	0.701	0.598	0.680	0.526	0.639	0.692	0.684
MLP	**0.716**	0.683	0.776	0.697	**0.656**	0.713	**0.673**	0.647	0.759	0.674
GP	0.681	**0.686**	0.728	**0.708**	0.635	**0.715**	0.625	0.655	0.735	0.665
MoE	0.620	0.673	0.669	0.698	0.643	0.691	0.480	0.631	0.690	0.673
DMM Language-Clusters $k = 6$										
OLS	0.604	0.664	0.662	0.699	0.578	0.666	0.519	0.618	0.658	0.672
MLP	0.712	0.681	**0.785**	0.695	0.655	0.706	0.655	0.647	0.753	0.677
GP	0.646	0.679	0.681	0.706	0.628	0.701	0.581	0.614	0.694	0.693
MoE	0.606	0.668	0.667	0.698	0.622	0.678	0.480	0.624	0.653	0.671

Table 1: System- and segment-level meta-evaluation results using soft pairwise accuracy (SPA) and group-by-item pairwise accuracy with tie calibration (acc^*_{eq}). We compare baseline metrics (XCOMET-XL, BLEURT) with variants of the MM and DMM models trained with and without language-pooling. Higher values indicate better agreement with human judgements.

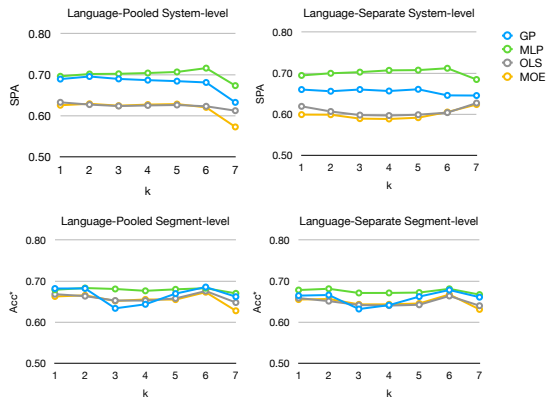

Figure 1: Effect of the number of clusters k.

rameters, and model selection criteria are provided in Appendix B.

4.2 Results

Table 1 reports system- and segment-level meta-evaluation results across language pairs, comparing baseline metrics with the proposed DMM variants under different conditioning strategies.

Semantic context over language specificity We observe that Language-Pooled clustering performs competitively with, and often marginally better than Language-Separate clustering. This challenges the prevailing assumption that evaluation requires language-specific tuning. Instead, it suggests that source-sentence semantics are more predictive of metric reliability and that cluster-specific combination strategies learnt on pooled multilingual data transfer across language pairs.

DMM wins over SOTA and static MM Hard-conditioned DMM outperforms state-of-the-art single metrics and static meta-metrics. This indicates that improved metric combinations can outperform larger neural metric architectures. The gap between DMM and MM suggests that source-sentence clustering adds value beyond standard ensemble approaches. OLS also benefits from clustering, supporting the view that the clustering structure contributes to performance.

5 Conclusion

We introduced Dynamic Meta-Metrics, a source-conditioned framework for combining machine translation evaluation metrics. DMM uses source-side clustering to adapt metric combinations through either hard or soft conditioning. Across multiple language pairs and pairwise agreement measures, cluster-based conditioning is often competitive with, and in several settings improves upon, language-pooled and language-specific combinations, while outperforming single-metric baselines on average. These results suggest that incorporating source-level structure can be a useful complement to static metric ensembles for MT evaluation.

Limitations

Source-only conditioning. Both hard and soft conditioning derive their context from the source segment. This is to make the routing system-invariant. However, this still does not directly capture any other phenomena, such as hypothesis-specific difficulty.

Metric set constraints. We exclude the largest metric variants to support execution on typical hardware. This may limit direct comparison with the strongest available single-metric baselines.

Metric-on-metric overfitting. When optimizing for MSE (Linear Regressors) or correlation (Gaussian Processors), the combiner only sees other metrics' scores instead of human error patterns, so it can chase artifacts of those metrics (e.g., overlap bias, domain-specific scaling). For example, if a base metric has a quirk (e.g., it over-rewards literal overlap on speech transcripts), the weight search can "learn" that quirk as a shortcut to higher validation scores without actually getting closer to human judgement.

Language coverage. Although we choose our language pairs judiciously to demonstrate our framework's applicability in high-resource directions (e.g., Chinese) and lesser-resourced directions (e.g., Ukrainian), we still have only four language pairs.

References

David Anugraha, Garry Kuwanto, Lucky Susanto, Derry Tanti Wijaya, and Genta Winata. 2024. MetaMetrics-MT: Tuning meta-metrics for machine translation via human preference calibration. In *Proceedings of the Ninth Conference on Machine Translation*, pages 459–469, Miami, Florida, USA. Association for Computational Linguistics.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12914–12929, Singapore. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Markus Freitag, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? results of the WMT24 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81, Miami, Florida, USA. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh, and Markus Freitag. 2023. MetricX-23: The Google submission to the WMT 2023 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 756–767, Singapore. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Masaaki Nagata, Toshiaki Nakazawa, Martin Popel, and 3 others. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

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

Pennsylvania, USA. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Brian Thompson, Nitika Mathur, Daniel Deutsch, and Huda Khayrallah. 2024. Improving statistical significance in human evaluation of automatic metrics via soft pairwise accuracy. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1222–1234, Miami, Florida, USA. Association for Computational Linguistics.

A Data processing and joins

Sources. We use the WMT Metrics Shared Task test sets from 2021–2024 (Freitag et al., 2024). We focus on English to Czech (en–cs) and English to Chinese (en–zh) to test performance variation on languages with different resource levels and topologies; Czech is lower-resource, synthetic alphabetic while Chinese is higher resource, analytic, logographic language. For each language pair, we use the source sentences, system outputs and reference translations as input, and compare against the official segment-level human annotations, using the year-specific fields: 2021 `wmt-raw:seg`, 2022 `wmt-appraise:seg`, 2023 `da-sqm:seg`, 2024 `esa:seg`. Segments without human annotation are excluded.

Units. A segment is one source sentence (WMT `seg-id`). A system is a MT run that produces one hypothesis per segment (`system-name`). Within a given (year, language pair) we identify an item by the pair (`system-name`, `seg-id`). When joining across years, we suffix the year to the identifier to keep years disjoint.

B Implementation details and hyperparameters

This appendix specifies the implementation choices and hyperparameters required to reproduce the clustering, responsibility construction, and model training procedures described in Section 3.

B.1 Source embeddings

We embed each source segment using LaBSE sentence embeddings (Feng et al., 2022). We encode source sentences in batches of 128 and apply embedding normalisation to unit length prior to clustering and distance computation. We treat the encoder as fixed and do not fine-tune it.

B.2 Clustering via k-means on training sources

For each choice of number of clusters K, we fit a k-means model to source embeddings from the training split only. We treat a source sentence as a unique string and deduplicate before fitting. We use Euclidean distance in embedding space. Since embeddings are normalised, Euclidean distance is proportional to angular distance, which improves stability for clustering and subsequent distance-based computations. For $K \in \{2, 3, 4, 5, 6, 7\}$,

we run k-means with the standard initialisation in `sklearn` (Pedregosa et al., 2011).

B.3 Soft responsibilities from cluster distances

Given a trained k-means model with centroids and a source embedding, we construct soft responsibilities, as per Section 3.5. We set the responsibility temperature to $T = 0.25$. Lower values yield assignments closer to hard clustering, while higher values yield more uniform mixing. We compute responsibilities separately for the training and validation splits using the same fixed centroids. Note that responsibilities depend only on the source segment, and therefore remain constant across all MT systems for a fixed segment.

B.4 Hard-conditioned models

Hard-conditioned DMM realizes the four ensemble configurations described in Section 3.4 by training separate models for each condition, without parameter sharing.

- **Linear (OLS).** $M(x) = \mathbf{w}^\top \mathbf{s}(x) + b$. Inputs are z-standardized per metric and a linear regression model is fit using ordinary least squares to minimize mean squared error (MSE) on the human gold scores $h(x)$.

- **Neural (MLP).** Two hidden layers (64 and 32 units) with ReLU activations and dropout $p = 0.2$. Trained with MSE using Adam (learning rate 10^{-3}), batch size 32, for 100 epochs. Inputs are z-standardized.

- **Gaussian Process (GP).** A linear combiner $M(x) = \mathbf{w}^\top \mathbf{s}(x)$ with $\mathbf{w} \in [0, 1]^d$. We learn $\mathbf{w}$ via Bayesian optimization with a Gaussian Process to *maximize* Kendall's tau correlation to human scores. We use 5 random initial evaluations and 100 optimization iterations. Degenerate zero solutions are avoided with a small uniform initialization.

B.5 Soft-conditioned DMM: mixture of linear experts

We implement soft conditioning using a mixture of experts with linear experts and a learned gating network. For each instance i, the model receives (i) a base-metric feature vector $\tilde{\mathbf{x}}_i$ and (ii) a gate feature vector $\mathbf{z}_i$. Base-metric features are the z-standardised metric scores. Gate features comprise the source sentence length and the responsibility vector $\mathbf{r}(s_i)$. In the default configuration, we exclude centroid distances from $\mathbf{z}_i$ and rely on responsibilities.

Gating network and routing. The gating network is a two-layer perceptron with a single hidden layer of 64 units, rectified linear unit activations, and dropout rate 0.1 applied to the hidden representation. Let $g_\phi(\mathbf{z}_i) \in \mathbb{R}^K$ denote the gate logits. We incorporate the responsibility vector as a log-prior:

$$\boldsymbol{\ell}_i = g_\phi(\mathbf{z}_i) + \lambda \log \mathbf{r}(s_i), \quad \boldsymbol{\alpha}_i = \mathrm{softmax}\left(\boldsymbol{\ell}_i / \vartheta\right).$$

We set the prior weight to $\lambda = 0.5$ and the gate temperature to $\vartheta = 2.0$. The gate temperature controls the sharpness of routing, with larger values yielding smoother mixtures. We add a small constant to responsibilities before taking logarithms to ensure numerical stability.

Optimisation and regularisation. We train the mixture model by minimising mean squared error on human scores, with two additional regularisers described Section 3.6. We use entropy regularisation weight $\beta = 0.01$ to discourage routing collapse and use expert-tying weight $\gamma = 10^{-6}$ to constrain experts towards a common solution, in the case where specialisation is not suggested by the data. We use AdamW optimisation with learning rate 3×10^{-4}, weight decay 10^{-4}, batch size 256, and a fixed random seed of 0 for model initialisation and data shuffling. We run training for up to 100 epochs and apply early stopping with patience 10 epochs using validation mean squared error as the selection criterion.

Initialisation. We initialise all experts from a closed-form linear regression solution trained on the same ensemble configuration and features. This initialisation yields a starting point equivalent to a single linear regressor replicated across experts. Under this initialisation, the gating network begins close to uniform routing, and training only introduces expert specialisation if it improves validation performance under early stopping.

C Use of Generative AI

AI was only used to assist in revising work that was written by us, and for providing some code completions in experiment code.