LAND: Learning an Adaptive Number of Draws for Bayesian Best-of-N Selection

Koen Oostermeijer[*†] **and Teryn Jones**[*]
Aleph Alpha Research

[*]Equal contribution
[†]Corresponding author: koen.oostermeijer@aleph-alpha-research.com

Abstract

Best-of-N decoding improves generation quality by sampling multiple candidates and selecting the best, but it allocates equal compute to sources regardless of how much they benefit from additional sampling: the expected benefit is small for sources with low-variance score distributions and potentially much larger for those with high-variance distributions. To address this shortcoming, we propose LAND (Learning an Adaptive Number of Draws), which dynamically allocates samples according to their posterior-predictive marginal value. LAND learns a discrete empirical mixture of source-conditioned score distributions from calibration data, updates its belief about a source as completion scores are observed, and stops sampling when the expected improvement of another draw falls below a shared compute price. Across machine-translation experiments with multiple models and language pairs, LAND achieves better MetricX scores than fixed best-of-N at matched average compute. Conversely, at matched translation quality, LAND requires approximately 20-40% fewer candidate generations. Adaptive allocation therefore provides a simple way to retain the benefits of best-of-N while substantially improving its compute efficiency.

1 Introduction

Model-generated data have become an increasingly important component of the training and fine-tuning of large language models (LLMs) (Long et al., 2024; Wang et al., 2023). A larger teacher model can generate training data for a smaller model, thereby transferring some of its capabilities through knowledge distillation. This approach has a long history in machine translation (MT), where sequence-level knowledge distillation uses translations generated by a teacher model as targets for a student model (Kim and Rush, 2016).

Historically, machine translation systems were trained primarily on naturally occurring parallel data collected from multilingual sources and subsequently filtered and aligned (Tiedemann, 2012). Synthetic data initially served mainly as a supplement to these corpora, most notably through back-translation of monolingual source texts (Sennrich et al., 2016; Edunov et al., 2018). Over time, however, the balance has shifted: as the quality of machine-translated data outpaced that of web-crawled corpora (Finkelstein et al., 2024), their share of the training data used by modern MT systems now constitutes an increasingly large fraction, in some cases completely replacing natural data (Finkelstein et al., 2026). As reliance on generated translations grows, selecting high-quality candidates becomes an increasingly important part of the data-generation pipeline.

One simple way to improve the quality of generated translations is to sample multiple candidates and retain the candidate with the highest score according to some quality metric, a technique referred to as *best-of-N* sampling. Previous work has shown that filtering data for quality-estimation metrics can result in improved downstream performance (Peter et al., 2023).

However, conventional best-of-N generates the same number of candidates for every source sentence. This allocation implicitly assumes that an additional sample is equally valuable for all source sentences. In practice, the distributions of candidate scores can differ substantially between source sentences. As illustrated in Figure 1, some intrinsically simple sentences consistently produce high-scoring translations, such that additional sampling offers little added improvement, while others produce more variable candidates and may benefit considerably from further sampling. Given a collection of source sentences and a fixed generation budget, the relevant question is therefore not only how many candidates to

Proceedings of the Eleventh Conference on Machine Translation, pages 520–529
October 28-29, 2026 ©2026 Association for Computational Linguistics

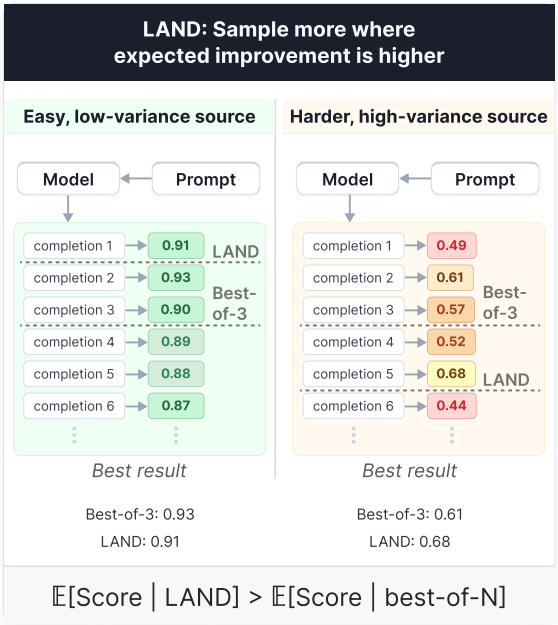

Figure 1: Fixed best-of-N always generates the same number of candidates for every source. However, higher overall scores can be achieved by allocating less compute to sources whose candidate scores are tightly concentrated and more compute to sources with higher-variance score distributions, where additional draws offer greater potential for improvement.

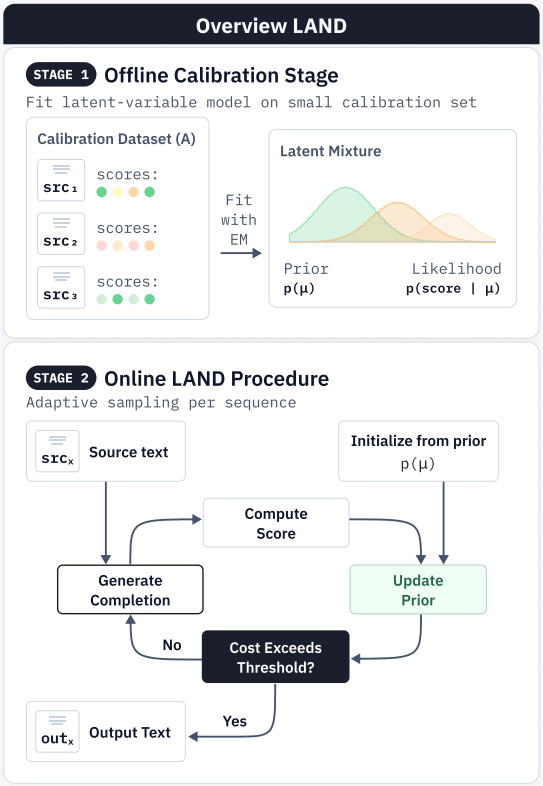

Figure 2: During calibration, LAND fits a discrete empirical mixture and calibrates a global compute price. At deployment, candidates are generated and scored sequentially, the posterior over source types is updated, and sampling continues while the expected improvement of another draw exceeds the compute price.

generate, but also where additional candidates are most valuable. This motivates our central research question: how should a fixed generation budget be allocated across source sentences so that additional candidates are generated where they offer the greatest expected marginal improvement?

We introduce **LAND** (**L**earning an **A**daptive **N**umber of **D**raws), an adaptive procedure that allocates generation compute according to the expected marginal improvement of additional candidates for each source sentence. A high-level overview of LAND is shown in Figure 2. LAND is related in spirit to Weitzman's Pandora's-box rule (Weitzman, 1978), which uses reservation values to balance the potential benefit of further search against its cost. Unlike the classical setting, however, the source-specific score distributions in our problem are unknown and each source sentence may be sampled repeatedly. LAND therefore updates a posterior over the score distribution after every observed candidate and adjusts its stopping threshold on the fly. To avoid imposing a restrictive parametric form, we estimate these distributions using a discrete empirical model constructed from observed scores.

In summary, our contributions are as follows:

- We formulate best-of-N selection as a compute-allocation problem in which the number of candidates may vary across source sentences under a shared average budget.
- We derive an oracle allocation rule that relates the decision to generate another candidate to its expected marginal improvement.
- We introduce LAND, a posterior-adaptive procedure that estimates this improvement online using a discrete empirical model of source-conditioned score distributions.
- We evaluate LAND against fixed-allocation and adaptive baselines for machine translation under matched average generation budgets.

2 Related Work

2.1 Quality Filtering and Selection in Machine Translation

Quality-estimation metrics have been used both to select translation hypotheses and to filter training data. Fernandes et al. (2022) apply reference-free metrics to N-best reranking and minimum Bayes risk decoding, while Peter et al. (2023) use them to retain higher-quality parallel sentence pairs. For synthetic data, Xu et al. (2022) show that the usefulness of back-translated examples depends on both translation quality and example importance. Complementing quality-based filtering, Kocmi et al. (2025) prioritize difficult training examples. Related work also uses quality estimates to allocate computation: Cheng et al. (2025) selectively evaluate candidates from a fixed list, and Farinhas et al. (2025) defer low-quality translations to a larger model.

2.2 Bayesian Stopping and Adaptive Allocation

LAND shares similarities with Weitzman's Pandora's-box problem, in which costly alternatives with known reward distributions are inspected until further search is no longer worthwhile (Weitzman, 1978). This framework has subsequently been applied to settings such as cost-aware Bayesian optimization and adaptive LLM sampling (Xie et al., 2024). Related optimal-stopping approaches have also been proposed for LLM inference. Wan et al. (2025) introduce BEACON, which models each prompt's reward distribution as Gaussian and applies a Bayesian finite-horizon stopping rule, whereas Kalayci et al. (2025) learn an unknown Weitzman threshold using upper confidence bounds.

More broadly, LAND is related to Bayesian adaptive-allocation methods such as Thompson sampling (William, 1933), which use posterior uncertainty to allocate observations sequentially rather than committing the same sampling budget to every alternative.

AdaBoN (Raman et al., 2025) also studies adaptive Best-of-N allocation, but is designed for batched allocation with relatively large exploration budgets. Since LAND targets sequential stopping in a substantially lower-budget regime, we consider the methods complementary rather than directly comparable.

LAND differs from these approaches in both how it models uncertainty and how it allocates compute. It learns a discrete empirical mixture of source-conditioned score distributions and updates a posterior over source types as candidates are observed. Rather than treating each source independently with a fixed sampling cost, LAND uses a shared compute price to allocate a fixed average generation budget toward sources with the greatest posterior-predictive marginal value of additional draws.

3 Problem Formulation

We consider a set of source texts and a fixed compute budget for generating completions. For each source text, an LLM generates one or more candidates, from which the highest-scoring completion is selected. Metrics that assign lower scores to better candidates are transformed so that higher values consistently correspond to better completions. The budget is specified by a target expected number of completions per source text, N, although in practice we allow the realized average to differ slightly due to finite-sample effects.

Our goal is to determine how to allocate the available compute across source texts so as to maximize the average score of the selected completions.

Applying the scoring function to the distribution of generated candidates induces a corresponding distribution over quality scores. Each source text has a latent mean score μ, defined as the expected score of a completion generated for that source text. Across source texts, these means follow the distribution

$$\mu \sim \mathcal{D}_{\mathrm{d}}, \tag{1}$$

whose expectation value $\mathbb{E}[\mu]$ is the score associated with the best-of-1 baseline, i.e. the average score obtained when exactly one completion is generated per source text.

Conditional on μ, the score X of an individual completion follows a distribution

$$X \sim \mathcal{D}_{\mathrm{c}}\left(\cdot \mid \mu, \sigma^2(\mu)\right), \tag{2}$$

where $\sigma^2(\mu)$ captures variation among completions generated for the same source text, which may depend on the latent mean μ.

4 Budget Allocation

Rather than generating a fixed number of candidates for every source text, i.e. best-of-N,

our strategy is to sample until the score exceeds a source-specific threshold. These thresholds are optimized to maximize the average selected score subject to an expected sampling budget of N completions per source text.

In the following we will derive an allocation rule assuming that μ is known, which is not the case in practice, as this is a latent variable; we relax this assumption in the following section.

For a source text with mean score μ, let $\{X_1, X_2, \ldots\}$ be independent draws from the conditional score distribution $\mathcal{D}_{\mathrm{c}}(\cdot \mid \mu)$, with probability density function f_μ and cumulative distribution function F_μ. We define the stopping time K as the number of samples it takes to exceed the threshold $\lambda(\mu)$. Since each draw has a probability of $1 - F_\mu(\lambda(\mu))$ to exceed the threshold, K follows a geometric distribution with expectation value

$$\mathbb{E}[K \mid \mu] = \frac{1}{1 - F_\mu(\lambda(\mu))}. \tag{3}$$

Because the sampling process is halted when the stopping condition is met, the final score X_K is also the one corresponding to the selected completion. Its expected value is the threshold plus the expected overshoot:

$$\mathbb{E}[X_K \mid \mu] = \lambda + \mathbb{E}\left[X - \lambda | X > \lambda, \mu\right] \tag{4}$$
$$= \mathbb{E}\left[X | X > \lambda, \mu\right]. \tag{5}$$

The optimal threshold function maximizes the expected selected score across source texts, subject to an average sampling budget of N:

$$\max_{\lambda(\cdot)} \mathbb{E}_\mu \left[\mathbb{E}\left[X | X > \lambda(\mu), \mu\right]\right]$$
$$\text{subject to } \mathbb{E}_\mu \left[\frac{1}{1 - F_\mu(\lambda(\mu))}\right] = N. \tag{6}$$

As derived in Appendix A, the optimal threshold satisfies

$$H_\mu(\lambda) \equiv \mathbb{E}\left[\left(X - \lambda(\mu)\right)_+ \middle| \mu\right] = \Lambda, \tag{7}$$

where $(z)_+ \equiv \max(z, 0)$ denotes the positive part of z. Thus, the left-hand side is the expected improvement of one additional draw over the current threshold. The global constant Λ is chosen so that the average-budget constraint holds. The optimal policy therefore equalizes the expected marginal improvement across source texts.

5 Posterior-Adaptive Allocation

The optimal allocation derived above assumes access to the latent source-specific mean μ. In practice, μ is unknown and must be inferred from the completion scores observed during sampling. We therefore maintain a posterior distribution over μ and update the stopping rule after each generated completion.

Let $\mathcal{X}_t \equiv \{X_1, \ldots, X_t\}$ denote the scores observed after t draws for a given source text, and let

$$q_t(\mu) \equiv p(\mu \mid \mathcal{X}_t) \tag{8}$$

be the corresponding posterior over its latent mean. Before observing any completions, this posterior is initialized as the population-level prior from Equation 1,

$$q_0(\mu) = P_{\mathrm{d}}(\mu). \tag{9}$$

Then, after observing a new score X_{t+1}, Bayes' rule gives

$$q_{t+1}(\mu) = \frac{f_\mu(X_{t+1}) q_t(\mu)}{\int \mathrm{d}\mu' \, f_{\mu'}(X_{t+1}) q_t(\mu')} \tag{10}$$

where f_μ is the conditional score density for a source text with mean μ.

The expected-improvement function is averaged over this posterior:

$$\widehat{H}_t(\lambda) \equiv \int \mathrm{d}\mu \, q_t(\mu) H_\mu(\lambda) \tag{11}$$
$$= \mathbb{E}\left[(X_{t+1} - \lambda)_+ | \mathcal{X}_t\right]. \tag{12}$$

Thus, $\widehat{H}_t(\lambda)$ is the expected improvement of one additional draw over a current score λ.

Let $B_t \equiv \max_{1 \le i \le t} X_i$ be the best score observed so far. The optimal policy is to continue sampling whenever it exceeds the global compute cost Λ:

$$\widehat{H}_t(B_t) > \Lambda. \tag{13}$$

Equivalently, we may define an adaptive threshold λ_t by

$$\widehat{H}_t(\lambda_t) = \Lambda. \tag{14}$$

The procedure continues while $B_t < \lambda_t$. Unlike the previous threshold $\lambda(\mu)$, the adaptive threshold λ_t changes as new scores are observed. Each completion updates both the best score B_t and the posterior belief about the source-specific score distribution, allowing the stopping decision to adapt online.

6 Practical Implementation

The practical implementation of LAND starts by estimating the score distributions using a small calibration set. We first rescale all scores to the interval $[0, 1]$, with larger values denoting better completions. These scores are then partitioned into J cells, with each cell j assigned a representative value v_j. The boundaries of these cells are initialized using the quantiles of these calibration scores, which provides greater resolution in densely populated regions of the score distribution.

We model variation across source texts using a mixture of I latent source distributions. We do not assume that source variation is fundamentally discrete. Rather, this is a finite approximation to continuous population heterogeneity while reducing posterior updates and expected-improvement calculations to finite sums. Each type i has prior probability

$$\pi_i = \Pr(z = i), \tag{15}$$

and a categorical distribution over score cells,

$$p_{ij} = \Pr(X \text{ falls in cell } j | z = i). \tag{16}$$

Here, z denotes the latent source type and X denotes the score of a generated completion. This discretized distribution greatly simplifies calculations and avoids imposing a restrictive parametric form on the source-conditioned score distributions. In particular, we found that parametric families such as (truncated) Gaussian or Beta distributions often impose inaccurate assumptions about tail behavior, which can substantially distort LAND's cost allocation policy.

Since the latent source types are not observed, we fit the mixture using expectation-maximization (EM). Suppose the calibration set contains L source texts. For source ℓ, let $n_{\ell j}$ denote the number of observed completion scores that fall in cell j, and let

$$n_\ell = \sum_{j=1}^{J} n_{\ell j}. \tag{17}$$

EM estimates the model parameters by maximizing the marginal log-likelihood of the observed data:

$$\mathcal{L} = \sum_{\ell=1}^{L} \log \left[\sum_{i=1}^{I} \pi_i \prod_{j=1}^{J} p_{ij}^{n_{\ell j}} \right]. \tag{18}$$

After fitting the mixture, we use it to initialize the stopping thresholds: For a source of latent type i, the expected improvement from one additional draw over a current best score b is computed as

$$H_i(b) = \sum_{j=1}^{J} p_{ij}(v_j - b)_+. \tag{19}$$

During sampling, each component-specific expected improvement is weighted by the posterior probability that the source belongs to the corresponding latent type. These weights are initialized as the population mixture weights, $q_{0,i} = \pi_i$. After observing a completion score x_{t+1} in cell $c(x_{t+1})$, the posterior is updated as

$$q_{t+1,i} = \frac{q_{t,i} p_{i,c(x_{t+1})}^{\tau}}{\sum_{k=1}^{I} q_{t,k} p_{k,c(x_{t+1})}^{\tau}}, \tag{20}$$

where $\tau \in (0, 1]$ is a likelihood-tempering parameter. The standard Bayesian update is recovered when $\tau = 1$. We use tempering to prevent the posterior from becoming overly concentrated after only a small number of observations.

Let B_t denote the best score observed after t draws. The posterior-predictive expected improvement of one additional draw is

$$\widehat{H}_t(B_t) = \sum_{i=1}^{I} q_{t,i} H_i(B_t). \tag{21}$$

LAND continues sampling whenever this expected improvement exceeds the global compute price Λ. In addition, we impose a maximum number of draws per source text of $K_{\max}$ to bound the worst-case allocation.

6.1 Calibrating Λ

The global cost limit Λ is calibrated by replaying the complete posterior-adaptive policy on a held-out set of calibration trajectories. For a candidate value of Λ, each trajectory is processed using the same posterior updates, running maximum draw rule, and sampling cap used at deployment. If $K_\ell(\Lambda)$ denotes the resulting number of draws for replay source ℓ, the estimated average compute is set to equal N:

$$\frac{1}{L_{\text{replay}}} \sum_{\ell=1}^{L_{\text{replay}}} K_\ell(\Lambda) \approx N. \tag{22}$$

This is to ensure the policy captures posterior transitions and stopping behavior as closely as possible.

Finally, to correct for any residual calibration error or potential distribution shift, we update Λ after each batch of completed source texts. Let $\overline{K}_m$ be the average number of draws in batch m. We apply the log-space update

$$\log \Lambda_{m+1} = \log \Lambda_m + \eta \frac{\overline{K}_m - N}{N}. \tag{23}$$

When the realized number of completions exceeds the target budget, the update increases Λ and makes further sampling less likely, and vice versa when the realized number falls below the target.

6.2 Complete LAND Procedure

The preceding subsections describe the individual components of LAND: fitting the discrete empirical mixture, updating the posterior over latent source types, estimating the posterior-predictive value of another draw, and calibrating the global compute price. The procedure is sequential: once LAND stops sampling a source and moves to the next source, that source is not revisited. Algorithm 1 summarizes how these components are combined into the complete procedure. The mixture model and initial value of Λ are estimated during calibration. At deployment, each newly observed completion updates both the current best score and the posterior over source types, and sampling continues only while the posterior-predictive expected improvement justifies its compute cost, subject to the maximum-draw constraint.

Algorithm 1 LAND with a discrete empirical mixture

Require: Mixture-fit trajectories $\mathcal{C}_{\text{fit}}$, replay trajectories $\mathcal{C}_{\text{replay}}$, target budget N, tempering τ, cap $K_{\max}$

Construct score cells and representatives $\{v_j\}_{j=1}^{J}$
Fit $\{\pi_i, p_{ij}\}$ on $\mathcal{C}_{\text{fit}}$ using EM
Compute $H_i(b) = \sum_j p_{ij}(v_j - b)_+$
Calibrate Λ on $\mathcal{C}_{\text{replay}}$
for each deployment source s **do**
  $q_i \leftarrow \pi_i$, $B \leftarrow -\infty$, $K \leftarrow 0$
  repeat
    Generate and score one completion x
    $K \leftarrow K + 1$, $B \leftarrow \max(B, x)$
    $q_i \leftarrow q_i p_{i,c(x)}^{\tau} / \sum_k q_k p_{k,c(x)}^{\tau}$
  until $K \geq K_{\max}$ or $\sum_i q_i H_i(B) \leq \Lambda$
  Return the highest-scoring observed completion
  Add K to the current update batch
  if the update batch is full **then**
    Update Λ using the batched log-space rule
  end if
end for

7 Results

We evaluate LAND on source sentences drawn from the OPUS parallel corpus. (Tiedemann, 2012) We use only the source side of each selected sentence pair and generate new translations with the evaluated model; the original target sentences are not used as generation targets. Using Gemma4 31B, (Gemma Team, 2026) we consider four translation directions: Ukrainian→German, Spanish→English, Dutch→Hungarian, and Turkish→Russian. Translation quality is evaluated with reference-free MetricX[1] (Juraska et al., 2024). For each translation direction and target budget N, we separately tune the sampling cap $K_{\max}$ and tempering parameter τ using the corresponding mixture-fit and replay trajectories. We evaluate average sampling budgets up to $N = 32$ and compare LAND against fixed best-of-N selection at each budget. Using MetricX for both selection and evaluation is intentional: LAND addresses how efficiently a given filtering objective can be optimized, rather than whether that objective improves downstream model training. The utility of QE-based filtering for MT data has been established independently by prior work (Peter et al., 2023).

Figure 3 shows the normalized MetricX score as a function of the sampling budget, averaged across the four Gemma4 language pairs. LAND consistently achieves higher translation quality than fixed best-of-N at the same number of draws. Equivalently, LAND reaches the same MetricX score with substantially fewer candidate generations. At the operating points highlighted in the figure, LAND matches the performance of fixed best-of-N while using approximately 20-30% fewer draws. Although the relative savings are largest at lower budgets, they remain substantial throughout the evaluated range.

[1] https://huggingface.co/google/metricx-24-hybrid-xxl-v2p6-bfloat16

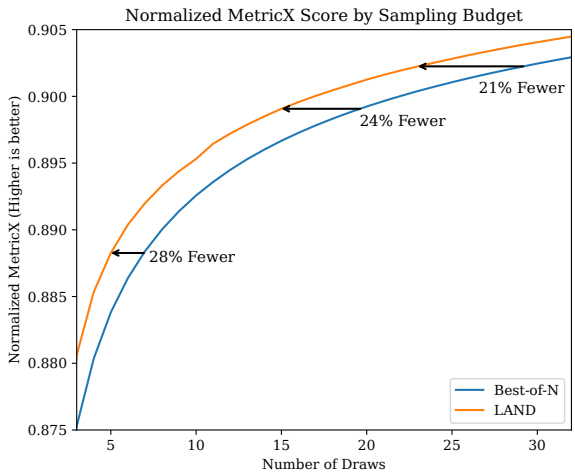

Figure 3: Normalized MetricX score as a function of the number of draws, averaged over the four Gemma4 31B translation directions. LAND reaches the same translation quality as fixed best-of-N using approximately 20-30% fewer draws at the highlighted operating points.

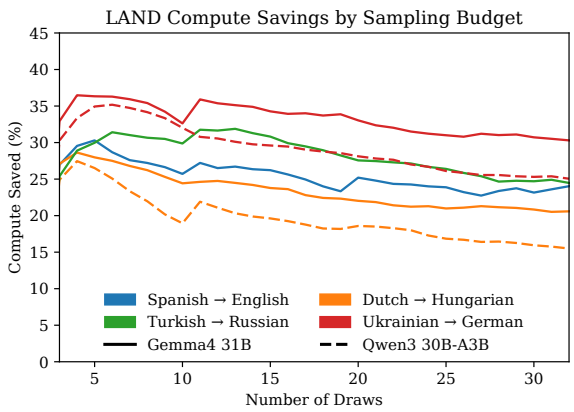

Figure 4: Candidate-generation savings of LAND relative to fixed best-of-N at matched MetricX performance. LAND reduces the number of required draws across all evaluated language pairs and sampling budgets.

Figure 4 reports the candidate-generation savings separately for each translation direction. In addition to the four Gemma4 31B experiments, we evaluate Ukrainian→German and Dutch→Hungarian using Qwen3 30B-A3B (Qwen Team, 2025). The savings are computed relative to fixed best-of-N at matched MetricX performance. Across models, language pairs, and sampling budgets, LAND reduces the required number of completions by approximately 20-40%.

The magnitude of the savings varies across translation directions. Ukrainian→German exhibits the largest and most consistent reductions in compute, while Dutch→Hungarian generally shows smaller gains, especially for Qwen3 30B-A3B at larger budgets. Nevertheless, LAND provides positive candidate-generation savings across all evaluated settings.

LAND targets an average budget of N rather than enforcing this exactly. The adaptive compute-price update nevertheless keeps the realized budget very close to its target: across the six evaluated model-language settings, the mean absolute deviation from N ranges from 0.01 to 0.04 draws. Across repeated runs on the same data with different random orderings, the standard deviation of the resulting draw savings is approximately 0.8-1.5 percentage points across the evaluated settings.

Overall, these results show that allocating the same number of candidates to every source sentence is inefficient. By directing additional samples toward source sentences for which they have greater posterior-predictive value, LAND preserves the quality gains of best-of-N while substantially reducing generation cost.

7.1 Comparison to BEACON

Finally, we compare LAND against BEACON (Wan et al., 2025), a Bayesian adaptive-sampling method that models the reward distribution for each source as Gaussian and updates its parameters online. In our experiments, BEACON underperforms fixed best-of-N, requiring more compute to reach the same average MetricX score for all sampling budgets, language pairs, and generation models considered. This seems to indicate that its stopping rule is not well suited to the score distributions encountered in this setting. One possible explanation is the Gaussian modeling assumption: the source-conditioned MetricX distributions can exhibit shapes that are poorly captured by a single parametric distribution. LAND's performance similarly suffered under this modeling assumption. A second possible factor is that BEACON must infer the reward distribution of a new source from the small number of samples observed for that source, while LAND can exploit population-level structure learned across calibration sources. However, we have not isolated the contribution of these differences, and a more detailed investigation would be required to determine the cause of BEACON's underperformance in this setting.

8 Conclusion and Limitations

We introduced LAND, an adaptive alternative to fixed best-of-N sampling that allocates generation compute according to the posterior-predictive value of additional candidates. Rather than committing to the same number of draws for every source, LAND uses observed completion scores to update its belief about the source-specific score distribution and continues sampling only when the expected improvement of another draw exceeds a global compute price. A discrete empirical mixture provides a flexible approximation to the population of source-conditioned score distributions, while replay-based calibration allows the resulting policy to satisfy a prescribed average generation budget.

Across the machine-translation settings considered in this work, LAND consistently improves the compute-quality trade-off relative to fixed best-of-N. At matched average compute, adaptive allocation yields better translation quality; conversely, at matched MetricX performance, LAND reduces the required number of candidate generations by approximately 20-40% across the evaluated models, language pairs, and sampling budgets. These results indicate that a substantial fraction of best-of-N compute can be saved simply by reallocating samples toward more promising sources.

Although our experiments focus on machine translation, the formulation of LAND is not specific to translation. The method applies whenever multiple candidate outputs can be sampled for the same input, evaluated with a continuous scalar reward, and compared to select the best candidate. It therefore extends directly to other best-of-N settings, although performance will depend on the informativeness of the chosen score. Our current experiments validate LAND only with MetricX, on a limited set of model families, and in machine translation. Future work will test alternative metrics, additional model families and scales, and other generation tasks equipped with continuous reward or quality scores.

Some limitations remain. First, LAND requires calibration data containing multiple completion scores per source in order to estimate the empirical mixture and calibrate the compute price. This introduces an upfront cost that fixed best-of-N does not require and may be undesirable when only a small dataset is to be generated. Second, the fitted score-distribution model remains fixed during deployment. Although LAND updates its posterior for each individual source and adjusts the compute price Λ to control the realized average budget, it does not use the completion scores collected across previously processed deployment sources to refine the empirical mixture itself. Consequently, additional information acquired during deployment is unused. Extending LAND to periodically update its mixture parameters from newly observed scores could reduce its dependence on an initial calibration set and allow it to adapt to distribution shift over time. And finally, we use the number of candidate generations as a proxy for compute. Since generations can differ in length and execution cost, the reported reduction in draws need not translate one-to-one into wall-clock or hardware savings. We leave this to future work.

References

Julius Cheng, Maike Züfle, Vilém Zouhar, and Andreas Vlachos. 2025. A Bayesian optimization approach to machine translation reranking. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2849–2862, Albuquerque, New Mexico. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

António Farinhas, Nuno M Guerreiro, Sweta Agrawal, Ricardo Rei, and Andre Martins. 2025. Translate smart, not hard: Cascaded translation systems with quality-aware deferral. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26742–26756, Suzhou, China. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Mara Finkelstein, Isaac Caswell, Tobias Domhan, Jan-Thorsten Peter, Juraj Juraska, Parker Riley, Daniel Deutsch, Geza Kovacs, Cole Dilanni, Colin Cherry, et al. 2026. Translategemma technical report. *arXiv preprint arXiv:2601.09012*.

Mara Finkelstein, David Vilar, and Markus Freitag. 2024. Introducing the NewsPaLM MBR and QE dataset: LLM-generated high-quality parallel data outperforms traditional web-crawled data. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1355–1372, Miami, Florida, USA. Association for Computational Linguistics.

Gemma Team. 2026. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*.

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.

Yusuf Kalayci, Vinod Raman, and Shaddin Dughmi. 2025. Optimal stopping vs best-of-n for inference time optimization. *arXiv preprint arXiv:2510.01394*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Tom Kocmi, Arkady Arkhangorodsky, Alexandre Berard, Phil Blunsom, Samuel Cahyawijaya, Théo Dehaze, Marzieh Fadaee, Nicholas Frosst, Matthias Galle, Aidan Gomez, Nithya Govindarajan, Wei-Yin Ko, Julia Kreutzer, Kelly Marchisio, Ahmet Üstün, Sebastian Vincent, and Ivan Zhang. 2025. Command-a-translate: Raising the bar of machine translation with difficulty filtering. In *Proceedings of the Tenth Conference on Machine Translation*, pages 789–799, Suzhou, China. Association for Computational Linguistics.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

Jan-Thorsten Peter, David Vilar, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, and Markus Freitag. 2023. There's no data like better data: Using QE metrics for MT data filtering. In *Proceedings of the Eighth Conference on Machine Translation*, pages 561–577, Singapore. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Vinod Raman, Hilal Asi, and Satyen Kale. 2025. Adabon: Adaptive best-of-n alignment. *arXiv preprint arXiv:2505.12050*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Guangya Wan, Zixin Stephen Xu, Sasa Zorc, Manel Baucells, Mengxuan Hu, Hao Wang, and Sheng Li. 2025. Beacon: Bayesian optimal stopping for efficient llm sampling. *arXiv preprint arXiv:2510.15945*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Martin Weitzman. 1978. *Optimal search for the best alternative*, volume 78. Department of Energy.

R Thompson William. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

Qian Xie, Raul Astudillo, Peter I Frazier, Ziv Scully, and Alexander Terenin. 2024. Cost-aware bayesian optimization via the pandora's box gittins index. *Advances in Neural Information Processing Systems*, 37:115523–115562.

Jiahao Xu, Yubin Ruan, Wei Bi, Guoping Huang, Shuming Shi, Lihui Chen, and Lemao Liu. 2022. On synthetic data for back translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 419–430, Seattle, United States. Association for Computational Linguistics.

Appendix

A Derivation of the Allocation Rule

To solve the optimization problem from Equation 10, define

$$C_\mu(\lambda) \equiv \mathbb{E}[K|\mu], \tag{24}$$

$$G_\mu(\lambda) \equiv \mathbb{E}[X_K \mid \mu] = \frac{\int_\lambda^\infty \mathrm{d}x\, x f_\mu(x),}{1 - F_\mu(\lambda)}. \tag{25}$$

Setting the constraint via a Lagrange multiplier Λ results in the functional

$$\mathcal{L}[\lambda] = \mathbb{E}_\mu\left[G_\mu(\lambda(\mu))\right] - \Lambda\big(\mathbb{E}_\mu\left[C_\mu(\lambda(\mu))\right] - N\big). \tag{26}$$

The corresponding first-order condition is

$$\frac{\delta \mathcal{L}}{\delta \lambda(\mu)} = G'_\mu(\lambda(\mu)) - \Lambda C'_\mu(\lambda(\mu)) = 0, \quad (27)$$

where

$$G'_\mu(\lambda) = \frac{f_\mu(\lambda)}{1 - F_\mu(\lambda)}(G_\mu(\lambda) - \lambda), \quad (28)$$

$$C'_\mu(\lambda) = \frac{f_\mu(\lambda)}{(1 - F_\mu(\lambda))^2}. \quad (29)$$

This yields the equation

$$\Lambda = (1 - F_\mu(\lambda))(G_\mu(\lambda) - \lambda) \quad (30)$$

$$= \int_\lambda^\infty \mathrm{d}x (x - \lambda) f_\mu(x) \quad (31)$$

$$= \mathbb{E}[(X - \lambda(\mu))_+ | \mu] \quad (32)$$